LESSON PLAN FOR COMPUTER GRAPHICS & MULTIMEDIA
SEMESTER III (2014-2015)
Class : CSE/IT Paper Code : ETCS-211
Paper Name : Computer Graphics & Multimedia Credits : 4
Total Lecture Classes Available : 40

FIRST TERM SYLLABUS

S. No	Contents	Hours
1.	Introduction, Application Areas, Interactive and Non-Interactive Graphics, Components of Interactive Graphics System. Overview of Input and Output Devices-Keyboard, mouse, joystick, scanner, stylus, light pen, Vector display, Raster Scan CRT display, Random Scan CRT display.	3
2.	Line Drawing Algorithms-DDA and Bresenham's Line drawing Algorithm	2
3.	Circle Drawing Algorithms-Bresnham's and Mid-Point Circle Drawing Algorithm	3
4.	Homogenous Co-ordinate System for 2D and 3D, Need of Homogenous Co-ordinate System, Various 2D and 3D Transformations like translation, rotation, scaling, reflection and shearing.	4
5.	Clipping, Line Clipping Algorithm-Cohen Sutherland Clipping Algorithm	2
6.	Curves and Surfaces-Curved Surfaces, polygonal approximation	1
7.	Bezier Curve, properties of Bezier curve, 4 point and 5-point Bezier curves using Berstein polynomial, conditions for the smooth joining of the curve segments B-Spline curves- cubic B-spline curves using uniform knot vectors, testing first and second order continuities.	3

SECOND TERM SYLLABUS

S. No	Contents	Hours
1.	Define Projection, Types of Projection- Perspective and Parallel projection. Define the term vanishing point. Hierarchy of Perspective Projection-One-point, Two-point and Three- point projection from point O on z-axis, computing the location of vanishing point. Hierarchy of Parallel Projection-Oblique Projection on XY-plane and Isometric Projection.	5
2.	Illumination Model for diffused Reflection, Ambient light, Specular Reflection model, 1 Reflection Vector	1
3.	Shading models - Flat shading , Gourard shading and Phong shading Model	2/
4.	Overview of multimedia: Classification, basic concepts of sound/audio MIDI: devices, messages, software. Authoring tools, Video and Animation : controlling animation, display and transmission of animation	5
5.	Data compression: storage space, coding requirements, basic compression techniques-5 Huffman code, Lempel - Ziv	5

THIRD TERM SYLLABUS

S. No	Contents	Hours
1.	JPEG: Image preparation, Lossy sequential DCT, Expanded Lossy DCT, Lossless mode, Hierachial mode	3
2.	MPEG, Media synchronization, media integration, production standards.	1

MODEL PAPER-I

FIRST TERM EXAMINATION
THIRD SEMESTER (B. TECH) ETCS-211
COMPUTER GRAPHICS & MULTIMEDIA-2014

Time : 1.5 hrs. **M.M. : 25**

Note: ***Attempt any three questions. Q.No. 1 is compulsory.***

Q.1. (a) What do you mean by error term in bresenham's line drawing algorithm. **(3)**

Ans: This algorithm determines an n-dimensional raster which should be plotted in order to form a close approximation to a straight line between two given points. It is commonly used to draw lines on a computer screen, as it uses only integer addition, subtraction and bit shifting, all of which are very cheap operations in standard computer architectures. It is one of the earliest algorithms developed in the field of computer graphics. A minor extension to the original algorithm also deals with drawing circles. the algorithm can track, instead of possibly large y values, a small error value between 0.5 and 0.5: the vertical distance between the rounded and the exact y values for the current x. Each time x is increased, the error is increased by the slope; if it exceeds 0.5, the rasterization y is increased by 1 (the line continues on the next lower row of the raster) and the error is decremented by 1.0.

Q.1. (b) Write the out codes for cohen Sutherland line clipping algorithm with example. **(5)**

Ans. The algorithm includes, excludes or partially includes the line based on where:

- Both endpoints are in the viewport region (bitwise OR of endpoints == 0): trivial accept.
- Both endpoints share at least one non-visible region which implies that the line does not cross the visible region. (bitwise AND of endpoints != 0): trivial reject.
- Both endpoints are in different regions: In case of this nontrivial situation the algorithm finds one of the two points that is outside the viewport region (there will be at least one point outside). The intersection of the outpoint and extended viewport border is then calculated (i.e. with the parametric equation for the line) and this new point replaces the outpoint. The algorithm repeats until a trivial accept or reject occurs.

The numbers in the figure below are called outcodes. An outcode is computed for each of the two points in the line. The outcode will have four bits for two-dimensional clipping, or six bits in the three-dimensional case. The first bit is set to 1 if the point is above the viewport. The bits in the 2D outcode represent: Top, Bottom, Right, Left. For example the outcode 1010 represents a point that is top-right of the viewport. Note that the outcodes for endpoints must be recalculated on each iteration after the clipping occurs.

1001	1000	1010
0001	0000	0010
0101	0100	0110

The Cohen-Sutherland algorithm can be used only on a rectangular clipping area. For other convex polygon clipping windows

Scanned by CamScanner

Q.1. (c) Define virtual reality. **(2)**

Ans. Virtual reality (VR), is a computer-simulated environment that can simulate physical presence in places in the real world or imagined worlds. Virtual reality can recreate sensory experiences, including virtual taste, sight, smell, sound, touch, etc.

Q 2 Discuss the DDA algo for Line drawing. **(10)**

Ans. DDA(x1, y1, x2, y2: integer);

```
var
        dx, dy, steps: integer;
        x_inc, y_inc, x, y: real;
begin
        dx := x2 - x1; dy := y2 - y1;
        if abs(dx) > abs(dy) then
        steps := abs(dx); {steps is larger of dx, dy}
else
        steps := abs(dy);
        x_inc := dx/steps; y_inc := dy/steps;
        {either x_inc or y_inc = 1.0, the other is the slope}
        x:=x1; y:=y1;
        set_pixel(round(x), round(y));
        for i := 1 to steps do
begin
        x := x + x_inc;
        y := y + y_inc;
        set_pixel(round(x), round(y));
end;
end; {DDA}
```

Q 3 Explain parametric and geometric continuity of a curve **(10)**

Ans. An alternate method for joining two successive curve sections is to specify conditions for geometric continuity. In This case, we only require parametric derivatives of the two sections to be proportional to each other at their common boundary instead of equal to each other.

Zero- order geometric continuity, described as G0 continuity, is the same as zero-order parametric continuity. That is, the two curves sections must have the same coordinate position at the boundary point. First order geometric continuity, or G1 continuity, means that the parametric first derivatives are proportional at the intersection on two successive sections. If we denote the parametric position on the curve as P(u), the direction of the tangent vector P'(u), but not necessarily its magnitude, will be the same for two successive curve sections at their joining point under G1 continuity. Second-order geometric continuity, or G2 continuity, means that both the first and second parametric derivatives of the two curve sections are proportional at their boundary. Under G2 continuity, curvatures of two curve sections will match at the joining position.

A curve generated with geometric continuity conditions is similar to one generated with parametric continuity, but with slight differences in curve shape. Figure below provides a comparison of geometric and parametric continuity. With geometric continuity, the curve is pulled toward the section with the greater tangent vector.

Fig. Curves with G^1 continuity

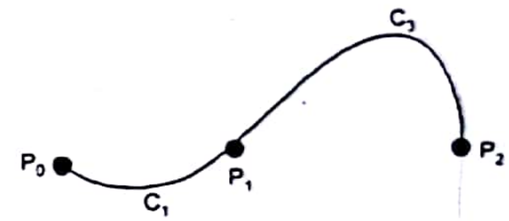


Fig. Curves with C^1 continuity

Q.4. (a) Write a procedure for rotating a given object about any specified rotation axis. **(5)**

Ans. 1.Translate the object to the origin of the coordinate system

2. Perform appropriate rotations to make the axis of rotation coincident with the z axis or any of the three coordinate axis

3. Rotate about z axis by the angle –"

4. Reverse the rotation about y-axis by the angle â. Then reverse the rotation about x-axis by the angle ?

5. Reverse the translation carried out in step 1.

Q.4. (b) Explain midpoint subdivision algo for line clipping **(5)**

Ans. (1) Read two end points of line P1 (x1,y1) and P2 (x2,y2).

(2) Read corners of window (Wx1, Wy1) and (Wx2, Wy2).

(3) Assign region codes for P1 and P2. A region code is a 4 digit bit code which indicates one of nine regions having the end point of line.

Initialize code with bits 0000

Set Bit 1 if (x<Wx1)

Set Bit 2 if (x>Wx2)

Set Bit 3 if (y<Wy2)

Set Bit 4 if (y>Wy1)

(4) Check visibility of line P1P2

(a) If region codes for both end points P1 and P2 are 0 = > Line is completely visible. Draw Line and Go to Step 3.

(b) If region codes for both end points P1 and P2 are not 0 and logical ANDing of them is also not zero = > Line is completely invisible. Reject line and Go to Step 3.

(c) If (a) and (b) both are not satisfied, line is partially visible.

(5) Divide partially visible line segments to equal parts and repeat steps 3 through 5 for both sub divided line segments until you get completely visible and completely invisible line segments.

(6) Stop.

Scanned by CamScanner

MODEL PAPER-I

SECOND TERM EXAMINATION

THIRD SEMESTER (B. TECH) ETCS-211

COMPUTER GRAPHICS & MULTIMEDIA-2014

Time : 1.5 hrs. **M.M. : 25**

Note: ***Attempt any three questions. Q.No. 1 is compulsory.***

Q.1. (*a*)Difference between parallel and perspective projection. **(5)**

Ans. Perspective projection is seeing things larger when they're up close and smaller at a distance. It is a three-dimensional projection of objects on a two-dimensional medium such as paper. It allows an artist to produce a visual reproduction of an object which resembles the real one.

The center of projection in a perspective projection is a point which is at a distance from the viewer or artist. Objects located at this point appear smaller and will appear bigger when they are drawn closer to the viewer. Perspective projection produces a more realistic and detailed representation of an object allowing artists to create scenes that closely resemble the real thing. The other type of projection which is also used aside from perspective projection is parallel projection.

Parallel projection, on the other hand, resembles seeing objects which are located far from the viewer through a telescope. It works by making light rays entering the eyes parallel, thus, doing away with the effect of depth in the drawing. Objects produced using parallel projection do not appear larger when they are near or smaller when they are far. It is very useful in architecture. However, when measurements are involved, perspective projection is best.

It provides an easier way of reproducing objects on any medium while having no definite center of projection. When it is not possible to create perspective projection, especially in cases where its use can cause flaws or distortions, parallel projection is used.

1. Perspective projection is representing or drawing objects which resemble the real thing while parallel projection is used in drawing objects when perspective projection cannot be used.

2. Parallel projection is much like seeing objects through a telescope, letting parallel light rays into the eyes which produce visual representations without depth while perspective projection represents objects in a three-dimensional way.

3. In perspective projection, objects that are far away appear smaller, and objects that are near appear bigger while parallel projection does not create this effect.

4. While parallel projection may be best for architectural drawings, in cases wherein measurements are necessary, it is better to use perspective projection.

Q.1. (*b*) Explain Diffused Light and Specular reflection . **(5)**

Ans. Diffuse reflection is the reflection of light from a surface such that an incident ray is reflected at many angles rather than at just one angle as in the case of specular reflection. An illuminated ideal diffuse reflecting surface will have equal luminance from all directions which lie in the half-space adjacent to the surface (Lambertian reflectance).

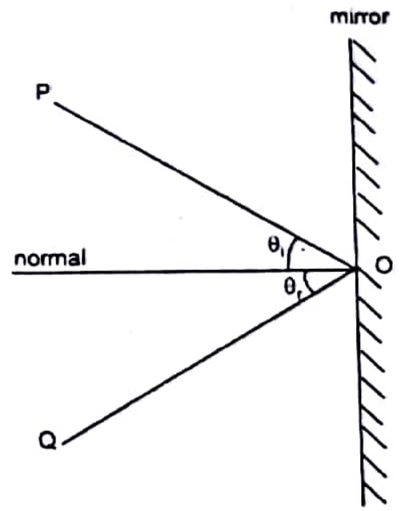
A surface built from a non-absorbing powder such as plaster, or from fibers such as paper, or from a polycrystalline material such as white marble, reflects light diffusely with great efficiency. Many common materials exhibit a mixture of specular and diffuse reflection.

The visibility of objects, excluding light-emitting ones, is primarily caused by diffuse reflection of light: it is diffusely-scattered light that forms the image of the object in the observer s eye.

Diffuse light is produced by extensive areas that emit light. These may be extensive, flat surfaces, such as the sky in the day-time, or, in the field of artificial lighting, luminous ceilings. In interior spaces diffuse light can also be reflected from illuminated ceilings and walls.

This produces very uniform, soft lighting, which illuminates the entire space and makes objects visible, but produces reduced shadows or reflections.

Specular reflection is the mirror-like reflection of light (or of other kinds of wave) from a surface, in which light from a single incoming direction (a ray) is reflected into a single outgoing direction. Such behavior is described by the law of reflection, which states that the direction of incoming light (the incident ray), and the direction of outgoing light reflected (the reflected ray) make the same angle with respect to the surface normal, thus the angle of incidence equals the angle of reflection (in the figure), and that the incident, normal, and reflected directions are coplanar.

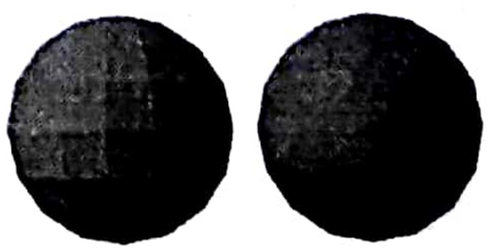


Specular reflection is distinct from diffuse reflection, where incoming light is reflected in a broad range of directions. An example of the distinction between specular and diffuse reflection would be glossy and matte paints. Matte paints have almost exclusively diffuse reflection, while glossy paints have both specular and diffuse reflection. A surface built from a non-absorbing powder, such as plaster, can be a nearly perfect diffuser, whereas polished metallic objects can specularly reflect light very efficiently. The reflecting material of mirrors is usually aluminum or silver.

Even when a surface exhibits only specular reflection with no diffuse reflection, not all of the light is necessarily reflected. Some of the light may be absorbed by the materials. Additionally, depending on the type of material behind the surface, some of the light may be transmitted through the surface. For most interfaces between materials, the fraction of the light that is reflected increases with increasing angle of incidence . If the light is propagating in a material with a higher index of refraction than the material whose surface it strikes, then total internal reflection may occur if the angle of incidence is greater than a certain critical angle.

Q.2. (*a*) Explain gourand shading model **(3)**

Ans. Gourand shading, is an interpolation method used in computer graphics to produce continuous shading of surfaces represented by polygon meshes. In practice, Gouraud shading is most often used to achieve continuous lighting on triangle surfaces by computing the lighting at the corners of each triangle and linearly interpolating the resulting colours for each pixel covered by the triangle.



Gouraud shading works as follows: An estimate to the surface normal of each vertex in a polygonal 3D model is either specified for

Scanned by CamScanner

each vertex or found by averaging the surface normals of the polygons that meet at each vertex. Using these estimates, lighting computations based on a reflection model, e.g. the Phong reflection model, are then performed to produce colour intensities at the vertices. For each screen pixel that is covered by the polygonal mesh, colour intensities can then be interpolated from the colour values calculated at the vertices.

Q.2. (*b*) explain Lempel- Ziv compression method. (7)

Ans. The Lempel-Ziv algorithm is a variable-to-fixed length code. Basically, there are two versions of the algorithm presented in the literature: the theoretical version and the practical version. Theoretically, both versions perform essentially the same. However, the proof of the asymptotic optimality of the theoretical version is easier. In practice, the practical version is easier to implement and is slightly more efficient.

Encoding Algorithm

1.Initialize the dictionary to contain all blocks of length one (D={a,b}).

2.Search for the longest block W which has appeared in the dictionary.

3.Encode W by its index in the dictionary.

4.Add W followed by the first symbol of the next block to the dictionary.

5. Go to Step 2.

The following example illustrates how the encoding is performed.

Data: a b b a a b b a a b a b b a a a a b a a b b a

0 1 1 0 2 4 2 6 5 5 7 3 0

Dictionary			
Index	Entry	Index	Entry
0	a	7	b a a
1	b	8	a b a
2	a b	9	a b b a
3	b b	10	a a a
4	b a	11	a a b
5	a a	12	b a a b
6	a b b	13	b b a

Q.3. Explain any 3 multimedia applications. (10)

Ans. Creative Industries: Creative industries use multimedia for a variety of purposes ranging from fine arts, to entertainment, to commercial art, to journalism, to media and software services provided for any of the industries listed below. An individual multimedia designer may cover the spectrum throughout their career. Request for their skills range from technical, to analytical, to creative.

Commercial uses: Much of the electronic old and new media used by commercial artists is multimedia. Exciting presentations are used to grab and keep attention in advertising. Business to business, and interoffice communications are often developed by creative services firms for advanced multimedia presentations beyond simple slide shows to sell ideas or liven-up training. Commercial multimedia developers may be hired to design for governmental services and nonprofit services applications as well.

Entertainment and fine arts: In addition, multimedia is heavily used in the entertainment industry, especially to develop special effects in movies and animations(VFX, 3D animation, etc.). Multimedia games are a popular pastime and are software programs available either as CD-ROMs or online. Some video games also use multimedia features. Multimedia applications that allow users to actively participate instead of just sitting by as passive recipients of information are called Interactive Multimedia. In the Arts there are multimedia artists, whose minds are able to blend techniques using different media that in some way incorporates interaction with the viewer. One of the most relevant could be Peter Greenaway who is melding Cinema with Opera and all sorts of digital media. Another approach entails the creation of multimedia that can be displayed in a traditional fine arts arena, such as an art gallery. Although multimedia display material may be volatile, the survivability of the content is as strong as any traditional media. Digital recording material may be just as durable and infinitely reproducible with perfect copies every time.

Education: In Education, multimedia is used to produce computer-based training courses (popularly called CBTs) and reference books like encyclopedia and almanacs. A CBT lets the user go through a series of presentations, text about a particular topic, and associated illustrations in various information formats. Edutainment is the combination of education with entertainment, especially multimedia entertainment.

Learning theory in the past decade has expanded dramatically because of the introduction of multimedia. Several lines of research have evolved (e.g. Cognitive load, Multimedia learning, and the list goes on). The possibilities for learning and instruction are nearly endless.

The idea of media convergence is also becoming a major factor in education, particularly higher education. Defined as separate technologies such as voice (and telephony features), data (and productivity applications) and video that now share resources and interact with each other, synergistically creating new efficiencies, media convergence is rapidly changing the curriculum in universities all over the world. Likewise, it is changing the availability, or lack thereof, of jobs requiring this savvy technological skill.

The English education in middle school in China is well invested and assisted with various equipments. In contrast, the original objective has not been achieved at the desired effect. The government, schools, families, and students spend a lot of time working on improving scores, but hardly gain practical skills. English education today has gone into the vicious circle. Educators need to consider how to perfect the education system to improve students' practical ability of English. Therefore an efficient way should be used to make the class vivid. Multimedia teaching will bring students into a class where they can interact with the teacher and the subject. Multimedia teaching is more intuitive than old ways; teachers can simulate situations in real life. In many circumstances teachers do not have to be there, students will learn by themselves in the class. More importantly, teachers will have more approaches to stimulating students' passion of learning.

Q.4. Explain the terms. (5 marks each)

Ans. Multimedia authoring tolls:

Multimedia authoring tools provide the framework for organizing and editing the elements of a multimedia project.

Authoring software provides an integrated environment for combining the content and functions of a project.

It enables the developer to create, edit, and import data.

Scanned by CamScanner

Card- and page-based tools.

Card- and page-based authoring systems provide a simple and easily understood metaphor for organizing multimedia elements.

It contains media objects such as buttons, text fields, and graphic objects.

It provides a facility for linking objects to pages or cards.

Icon-based, event-driven tools.

Icon-based, event-driven tools provide a visual programming approach to organize and present multimedia.

Multimedia elements and interaction cues are organized as objects in a flowchart.

Flowchart can be built by dragging appropriate icons from a library, and then adding the content.

Time-based tools.

Time-based tools are best suited for messages with a beginning and an end.

Some time-based tools facilitate navigation and interactive control.

Macromedia's Director and Flash are time-based development environments

Multimedia authoring tools provide the framework needed for organizing and editing multimedia elements in a project.

The three types of authoring tools are card- or page-based, icon-based or event-driven, and time-based.

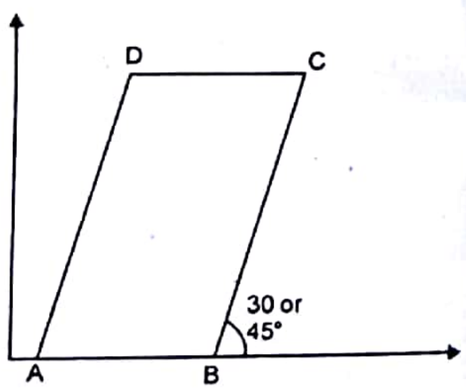
Text and color compatibility must be verified before moving multimedia projects across platforms.

(*b*) Oblique projection

A pictorial view of an object showing its elevation, plan, or section to scale with parallel lines projected from the corners, at 45 degrees or any other angle, indicating the other sides is called oblique projection.

cavalier projection (sometimes cavalier perspective or high view point) a point of the object is represented by three coordinates, x, y and z. On the drawing, it is represented by only two coordinates, x" and y". On the flat drawing, two axes, x and z on the figure, are perpendicular and the length on these axes are drawn with a 1:1 scale; it is thus similar to the dimetric projections, although it is not an orthographic projection, as the third axis, here y, is drawn in diagonal, making an arbitrary angle with the x" axis, usually 30 or 45°.

Cabinet projection (sometimes cabinet perspective) stems from its use in illustrations by the furniture industry. Like cavalier perspective, one face of the projected object is parallel to the viewing plane, and the third axis is projected as going off in an angle (typical 30° or 45°). Unlike cavalier projection, where the third axis keeps its length, with cabinet projection the length of the receding lines is cut in half.



MODEL PAPER-I

END TERM EXAMINATION

THIRD SEMESTER (B. TECH) ETCS-211

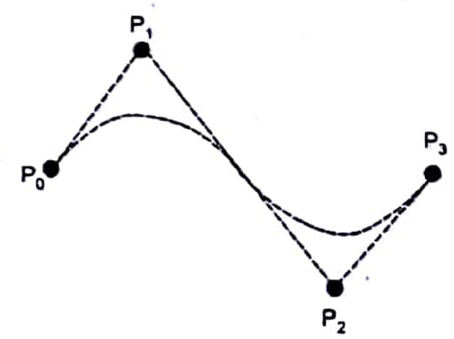
COMPUTER GRAPHICS & MULTIMEDIA-2014

Time : 3 hrs. **M.M. : 75**

Note: ***Attempt any three questions. Q.No. 1 is compulsory.***

Q.1. (*a*)Explain cubic Bezier curve. **(3)**

Ans. A Bezier curve is defined in terms of a number of control points. For example, a **cubic Bezier curve** is defined in terms of 4 control points P_0, P_1, P_2, P_3. To get an idea of the influence of a control point on the Bezier curve.



P_0 and P_3 define that start and endpoints of the curve. P_1 and P_2 determine the initial direction of the curve. Hence, the curve is determined by 2 endpoints and the slope of two lines segments ($P_1 - P_0$ and P_2–P_3).

The parametric formula for the Bezier curve given 4 control points is the following:

$$P_B = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4$$

with $0 <= t <= 1$

In the general case, for n control points, the following applies ($u = t$):

$$P(u) = \sum_{k=0}^{n} \binom{n}{k} (1-u)^{n-k} u^k P_k u \in R$$

Q.1. (*b*) Explain flat shading. **(3)**

Ans. The simplest shading model for a polygon is 'constant shading', also known as 'faceted shading' or 'flat shading'. This approach applies an illumination model once to determine a single intensity value that is then used to shade an entire polygon, and holding the value across the polygon to reconstruct the polygon's shade. This approach is valid if several assumptions are true:

1.The light source is at infinity, so N.L is constant across the polygon face.

2.The viewer is at infinity, so N.V is constant across the polygon face.

3.The polygon represents the actual surface being modelled, and is not an approximation to a curved surface.

If a visible-surface algorithm is used that outputs a list of polygons, such as one of the list-priority algorithms, constant shading can take advantage of the ubiquitous single-color 2D polygon primitive.

If either of the first two assumptions is wrong (actually, all three are wrong for just about every computer game you'll ever see), then, if we are to use constant shading, we

Scanned by CamScanner

need some method to determine a single value for each L and V. For example, values may be calculated for the center of the polygon, or for the polygon's first vertex. Of course, constant shading does not produce the variations in shade across the polygon that should occur in this situation.

Q.1. (c) Explain axonometric projection. **(4)**

Ans. Axonometric projection is a type of parallel projection used to create a pictorial drawing of an object, where the object is rotated along one or more of its axes relative to the plane of projection.

There are three main types of axonometric projection: isometric, dimetric, and trimetric projection.

"Axonometric" means "to measure along axes". Axonometric projection shows an image of an object as viewed from a skew direction in order to reveal more than one side in the same picture. Whereas the term orthographic is sometimes reserved specifically for depictions of objects where the axis or plane of the object is parallel with the projection plane, in axonometric projection the plane or axis of the object is always drawn not parallel to the projection plane.

"In isometric projection, the most commonly used form of axonometric projection in engineering drawing, the direction of viewing is such that the three axes of space appear equally foreshortened, and there is a common angle of 60° between them. As the distortion caused by foreshortening is uniform the proportionality of all sides and lengths are preserved, and the axes share a common scale. This enables measurements to be read or taken directly from the drawing.

"In dimetric projection, the direction of viewing is such that two of the three axes of space appear equally foreshortened, of which the attendant scale and angles of presentation are determined according to the angle of viewing; the scale of the third direction (vertical) is determined separately.

"In trimetric projection, the direction of viewing is such that all of the three axes of space appear unequally foreshortened. The scale along each of the three axes and the angles among them are determined separately as dictated by the angle of viewing.

Q.1. (d) Explain Image processing. **(6)**

Ans. There are two different approaches to image processing:

1. Analog processing. This approach is very fast since the time involved in analog-to-digital (AD) and a digital-to-analog (DA) conversion is saved. But this approach is not flexible since the manipulation of images is very hard.

2. Digital processing. This approach is slower than the analog approach but is very flexible, since manipulation is done very easily. The processing time of this approach is tremendously improved by the advent of parallel processing techniques.

Digital image processing is defined as the processing of two dimensional images by a digital computer". A digital image is represented by an array of regularly spaced and very small quantized samples of the image. Two processes that are related to any digital system are sampling and quantization. When a picture is digitized, it is represented by regularly spaced samples of this picture. These quantized samples are called pixels. The array of pixels that are processed in practice can be quite large. To represent an ordinary black and white television (TV) image digitally, an array of 512 × 512 pixels is required. Each pixel is represented by an 8 bit number to allow 256 gray levels. Hence a single TV picture needs about 2 × 106 bits. Digital image processing encompasses a wide variety of techniques and mathematical tools. They have all been developed for use in one or the other of two basic activities that constitute digital image processing: image preprocessing and image analysis. An approach called the state-space approach has been recently used in modeling image processors. These image processors are made of linear iterative circuits. The state-space model is used efficiently in image processing

and image analysis. If the model of an image processor is known, the realization of a controllable and observable image processor is then very simple.

Image Analysis

Image analysis accepts a digital image as input and produces data or a report of some type. The produced data may be the features that represent the object or objects in the input image. To produce such features, different processes must be performed that include segmentation, boundary extraction, silhouette extraction, and feature extraction. The produced features may be quantitative measures, such as moment invariants, and Fourier descriptors, or even symbols, such as regular geometrical primitives.

Sampling and Quantization

Quantization is the process of representing a very large number (possibly infinite) of objects with a smaller, finite number of objects. The representing set of objects may be taken from the original set (e.g., the common number-rounding process) or may be completely different (e.g., the alphabetical grading system commonly used to represent test results). Further image restoration is done to reconstruct the image.

Q.1. (d) Difference between random and raster display systems **(2)**

Ans.

Random Scan Display	Raster Scan Display
1. In random scan display the beam is moved between the end points of the graphics primitives.	1. In raster scan display the beam is moved all over the screen one scan line at a time from top to bottom and then back to top
2. Picture definition is stored as a set of line drawing instructions in a display file	2. Picture definition is stored as a set of intensity values for all screen points, called pixels in a refresh buffer area
3. Random scan display only draws lines and characters.	3. Raster scan has ability to display areas filled with solid colors or patterns.
4. It don't use interlacing.	4. It uses interlacing.
5. In this resolution is high.	5. In this resolution is less.
6. It is more expensive.	6. It is less expensive.
7. It uses monochrome or beam penetration type.	7. It uses monochrome or shadow mask type.
8. Random scan display draws a continuous and smooth lines.	8. Raster scan display mathematically smooth lines primitives only by approximating them with pixel on the raster grid.
9. Editing is easy.	9. Editing is difficult.
10. Refresh rate depends directly on pictur complexity.	10. Refresh rate independent of picture complexity.
11. Scan conversion is not required.	11. Graphics primitives are specifies in terms of their end points and must be scan converted into their corresponding pixels in the frame buffer.
12. Mathematical functions are used to draw an image.	12. Screen points/pixels are used to draw an image.

Scanned by CamScanner

Q.1. (g) Find the values of Fx, Fy, Fz for isometric projection. **(5)**

Ans. In isometric projection the projection plane makes equal angle with each principal axis. If the projection plane normal is (dx, dy, dz)

Then $|dx| = |dy| = |dz|$

Transforming the unit vector along the x-axis using isometric projectionmatrix yields.

$$[Ux] = [1001] \begin{matrix} \cos\phi & \sin\phi\sin\theta & 0 & 0 \\ 0 & \cos\theta & 0 & 0 \\ \sin\phi & -\cos\phi\sin\theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Thus the forehoretening factors are

$$f_x^2 = \cos^2\phi + \sin^2\phi \sin^2\theta$$
$$f_y^2 = \cos^2\phi$$
$$f_z^2 = \sin^2\phi + \cos^2\phi \sin^2\theta$$

Replacing $\cos^2\phi$ with $1 - \sin^2$ and $\cos^2\phi$ with $1\text{-}\sin^2\phi$

We get

$$\sin^2\phi = \sin^2\phi\, 1 - \sin^2\phi$$

similarly for F_x^2 and F_z^2

Therefore there are four possible isometric projections:

(*i*) $\phi = -45°\ \theta = 35.26°$

(*ii*) $\phi = -45°\ \theta = -35.26°$

(*iii*) $\phi = 45°\ \theta = 35.26°$

(*iv*) $\phi = -45°\ \theta = -35.26°$

For each case the forehortening factor is $\sqrt{2}\sqrt{3} = 0.8165$

Q.1.(*h*) Explain trackball and spaceball mouse. **(2)**

Ans. A spaceball is a "six degrees of freedom" device. It allows full rotation in multiple directions, such as with a CAD modeil. a trackball, however is merely a standard mouse. rather than having the ball moved by the mouse platform, you manually actuate the ball. it just moves the cursor around. the libsball page has a nice image showing the spaceball's range and type of motion. a trackball just does XY axis motion.

Q.2. Explain lossless and lossy compression techniques **(12.5)**

Ans. Lossy image compression and lossy video compression (such as JPEG compression, MPEG compression, and fractal image compression) give much better reduction in size (much higher compression ratio) than we find in almost any other area of data compression.

Lossy compression is best used to reduce the size of video data, where defects in the picture can be hidden as long as the general structure of the picture remains intact. This type of compression is called lossy compression because part of the reason that it compresses so well is that actual data from the video image, is lost and then replaced with some approximation.

Fractal compression is based on the concept of self-similarity in fractals. It uses fractal components that have self-similarity to the rest of the surrounding area in the picture.

The storage is so inaccurate that by replacing some of the fractal data with encrypted text, it is very difficult to see a difference in the picture, which means that important messages can be embedded in video images with little fear of detection.

If we could achieve the same level of compression without loss of data, we wouldn't use Lossy Compression because there is an imperceptible loss of quality in the image.

One of the reasons that there is still some interest in new loss-less compression techniques, is that only very inexact data structures can survive Lossy Compression. It is often the case that loss of a single bit, renders a whole phrase or line of data inaccurate. This is why we attempt to build more and more stable memory systems. The recent shift from RD to SD ram for instance was partially because RD ram needed more interactive maintenance of its data.

If we are so protective of the memory of data, then it makes sense that we must also be protective of the compression scheme we use to store and retrieve data. So the only places Lossy Compression can be used, are places where the accuracy at the bit level, does not materially affect the quality of the data.

Example: Run-length encoding (RLE) is a very simple form of data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs: for example, relatively simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could potentially double the file size.

For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. Let us take a hypothetical single scan line, with B representing a black pixel and W representing white:

WWWWWWWWBWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWW

If we apply the run-length encoding (RLE) data compression algorithm to the above hypothetical scan line, we get the following:

12W1B12W3B24W1B14W

This is to be interpreted as twelve Ws, one B, twelve Ws, three Bs, etc.

The run-length code represents the original 67 characters in only 18. Of course, the actual format used for the storage of images is generally binary rather than ASCII characters like this, but the principle remains the same. Even binary data files can be compressed with this method; file format specifications often dictate repeated bytes in files as padding space. However, newer compression methods such as DEFLATE often use LZ77-based algorithms, a generalization of run-length encoding that can take advantage of runs of strings of characters (such as BWWBWWBWWBWW).

Run-length encoding performs lossless data compression and is well suited to palette-based iconic images. It does not work well at all on continuous-tone images such as photographs, although JPEG uses it quite effectively on the coefficients that remain after transforming and quantizing image blocks.

Common formats for run-length encoded data include Truevision TGA, PackBits, PCX and ILBM.

Run-length encoding is used in fax machines (combined with other techniques into Modified Huffman coding). It is relatively efficient because most faxed documents are mostly white space, with occasional interruptions of black.

Data that have long sequential runs of bytes (such as lower-quality sound samples) can be RLE compressed after applying a predictive filter such as delta encoding.

Scanned by CamScanner

Q.3 Explain window and viewport. Explain window to viewport transformation matrix **(12.5)**

This mapping or transformation involves developing formulas that start with a point in the world window, say (xw, yw). The formula is used to produce a corresponding point in viewport coordinates, say (xv, yv).

Using this proportionality, the following ratios must be equal.

$$\frac{xv - xv_{min}}{xv_{max} - xv_{min}} = \frac{xw - xw_{min}}{xw_{max} - xw_{min}}$$

$$\frac{yv - yv_{min}}{yv_{max} - yv_{min}} = \frac{yw - yw_{min}}{yw_{max} - yw_{min}}$$

By solving these equations for the unkown viewport position (xy, yv) the following becomes true:

$$xv = S_x xw + t_x$$
$$yu = S_y yw + ty$$

The scale factors (Sx, Sy) would be

$$Sx = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$$

$$Sy = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}}$$

And the translation factors (Tx, Ty) would be:

$$tx = \frac{xw_{max}xv_{min} - xw_{min}xv_{max}}{xw_{max} - xw_{min}}$$

$$ty = \frac{yw_{max}yv_{min} - yw_{mm}yv_{max}}{yw_{max} - yw_{min}}$$

The position of the viewport can be changed allowing objects to be viewed at different positions on the Interface Window. Multiple viewports can also be used to display different sections of a scene at different screen positions. Also, by changing the dimensions of the viewport, the size and proportions of the objects being displayed can be manipulated. Thus, a zooming affect can be achieved by successively mapping different dimensioned clipping windows on a fixed sized viewport.

Q.4. (a) Explain Huffman Encoding with Example. **10**

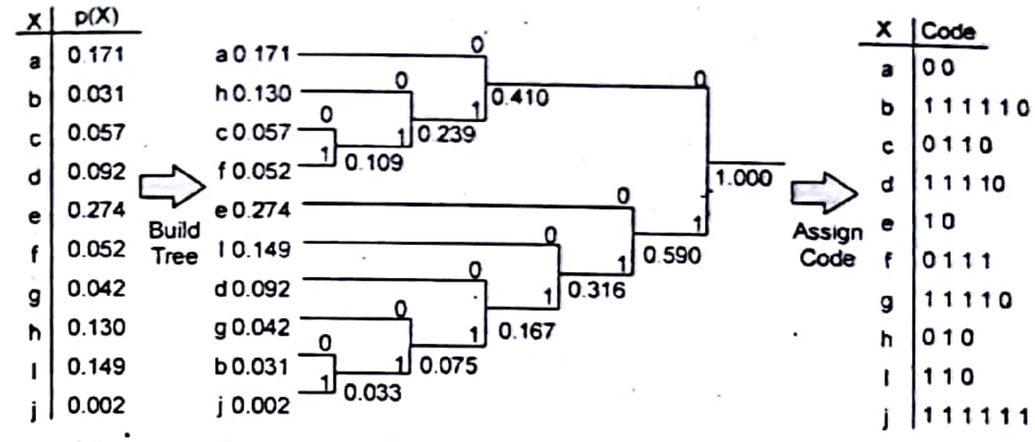
Ans. The basic idea in Huffman coding is to assign short codewords to those input blocks with high probabilities and long codewords to those with low probabilities.

X	p(X)
a	0.171
b	0.031
c	0.057
d	0.092
e	0.274
f	0.052
g	0.042
h	0.130
I	0.149
j	0 002

A Huffman code is designed by merging together the two least probable characters, and repeating this process until there is only one character remaining. A code tree is thus generated and the Huffman code is obtained from the labeling of the code tree. An example of how this is done is shown below.

The final static code tree is given below.

- It does not matter how the characters are arranged. I have arranged it above so that the final code tree looks nice and neat.
- It does not matter how the final code tree are labeled (with 0s and 1s). I chose to label the upper branches with 0s and the lower branches with 1s.
- There may be cases where there is a tie for the two least probable characters. In such cases, any tie-breaking procedure is acceptable.



- Huffman codes are not unique.
- Huffman codes are optimal in the sense that no other lossless fixed-to-variable length code has a lower average rate.
- The rate of the above code is 2.94 bits/character.
- The entropy lower bound is 2.88 bits/character.

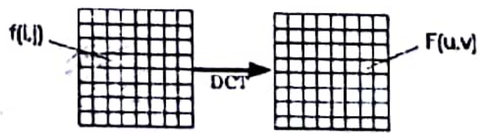
Q.4. (b) Explain dithering **(2.5)**

Ans. Dithering scatters different colored pixels in an image to make it appear as though there are intermediate colors in images with a limited color palette.

Dithering is often used to reduce banding in GIFs with graduated color transitions. Most software provides options that let you control the appearance of the scattered pixels; for example, the dithering can be a rigid pattern, random noise, or diffusion. Keep in mind that dithering often increases the file size of an image, but in many cases the improved appearance is worth the trade-off.

Q.5. Explain DCT **(12.5)**

Ans. The discrete cosine transform (DCT) helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). The DCT is similar to the discrete Fourier transform: it transforms a signal or image from the spatial domain to the frequency domain (Fig 7.8).



DCT Encoding: The general equation for a 1D (N data items) DCT is defined by the following equation.

$$F(u) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} A(i).\cos\left[\frac{\pi.u}{2.N}(2i+1)\right] f(i)$$

and the corresponding inverse 1D DCT transform is simle $F^{-1}(u)$, *i.e.*,

where

$$A(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi=0 \\ 1 & \text{otherwise} \end{cases}$$

Scanned by CamScanner

The general equation for a 2D (N by M image) DCT is defined by the following equation.

$$F(u, v) = \left(\frac{2}{N}\right)^{\frac{1}{2}}\left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} A(i).\Delta(j).\cos\left[\frac{\pi.u}{2.N}(2i+1)\right]\cos\left[\frac{\pi.v}{2.M}(2j+1)\right].f(i,j)$$

and the corresponding inverse 2D DCT transform is simple $F^{-1}(u, v)$, *i.e.*, where

$$A(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

The basic operation of the DCT is as follows:

- The input image is N by M;
- f(i,j) is the intensity of the pixel in row i and column j;
- F(u,v) is the DCT coefficient in row k1 and column k2 of the DCT matrix.
- For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT.
- Compression is achieved since the lower right values represent higher frequencies, and are often small - small enough to be neglected with little visible distortion.
- The DCT input is an 8 by 8 array of integers. This array contains each pixel's gray scale level;
- 8 bit pixels have levels from 0 to 255.
- Therefore an 8 point DCT would be:

where

$$A(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

MODEL PAPER-II

FIRST TERM EXAMINATION

THIRD SEMESTER (B. TECH) ETCS-211

COMPUTER GRAPHICS & MULTIMEDIA-2014

Time : 1.5 hrs. **M.M. : 25**

Note: ***Attempt any three questions. Q.No. 1 is compulsory.***

Q.1. (*a*) Q 1 a) What do you mean by retracing? Define horizontal as well as vertical retracing. **(3)**

Ans. The speed of older circuits was limited. For horizontal deflection, there is also a pause between successive lines, to allow the beam to return from right to left, called the ***horizontal retrace*** or horizontal blanking interval. Modern CRT circuitry does not require such a long blanking interval, and thin panel displays require none. When the beam reaches at the bottom left corner or the screen then the beam is again sent to top left for retracing again a second scene which is known as **vertical retrace or vertical run.**

Q.1. (*b*) What do you mean by Homogeneous coordinates? Give suitable examples. **(5)**

Ans. Homogeneous coordinates are a system of coordinates used in projective geometry, as Cartesian coordinates are used in Euclidean geometry. They have the advantage that the coordinates of points, including points at infinity, can be represented using finite coordinates. Formulas involving homogeneous coordinates are often simpler and more symmetric than their Cartesian counterparts. Homogeneous coordinates have a range of applications, including computer graphics and 3D computer vision, where they allow affine transformations and, in general, projective transformations to be easily represented by a matrix.

If the homogeneous coordinates of a point are multiplied by a non-zero scalar then the resulting coordinates represent the same point. An additional condition must be added on the coordinates to ensure that only one set of coordinates corresponds to a given point, so the number of coordinates required is, in general, one more than the dimension of the projective space being considered. For example, two homogeneous coordinates are required to specify a point on the projective line and three homogeneous coordinates are required to specify a point on the projective plane.

- Any point in the projective plane is represented by a triple (X, Y, Z), called the homogeneous coordinates or projective coordinates of the point, where X, Y and Z are not all 0.
- The point represented by a given set of homogeneous coordinates is unchanged if the coordinates are multiplied by a common factor.
- Conversely, two sets of homogeneous coordinates represent the same point if and only if one is obtained from the other by multiplying by a common factor.
- When Z is not 0 the point represented is the point $(X/Z, Y/Z)$ in the Euclidean plane.
- When Z is 0 the point represented is a point at infinity.

Note that the triple (0, 0, 0) is omitted and does not represent any point. The origin is represented by (0, 0, 1).

Scanned by CamScanner

Q.1. (c)Describe sweep representations for solid modelling. **(2)**

Ans. The basic notion embodied in sweeping schemes is simple. A set moving through space may trace or *sweep* out volume (a solid) that may be represented by the moving set and its trajectory. Such a representation is important in the context of applications such as detecting the material removed from a cutter as it moves along a specified trajectory.

There are two types of sweep representations

(*a*) Translational sweep

(*b*) Rotational sweep

Q.2. Discuss the bresenham's algo for circle drawing. using it generate the points for the circle with center (10,8) and radius 8. **(10)**

Ans. 1. Read the radius r of the circle.

2. Initialize the decision variable

3. d = 3-2*r

4. x=0 , y=r

5. if we are using octant symmetry to plot the pixels then until (x ≤ y) we have to perform following steps.

If (d<0) then

d=d+4x+6

X=x+1

Else

d=d+4(x-y) + 10

y= y-1

x=x+1

6. plot (x,y)

7. determine the symmetry points in other octants also

8. stop

r=8

d= 3-2*r

= 3-2*8

= -13

New point is **(1,8)**

And d2 is d+4x+6

=-13+4*1+6

= -13+10

d= -3

now new x,y are

(2,8)

And D3 is d+4x+6

= -3+4*2+6

=-3+14

=11

Now new X and y coordinates are (3,7)

D4 is d+4(x-y)+10

= 11+4(3-7)+10

= 11+(-16)+10

=11-6 = 5

Now new x and y are (4,6)

And new d is 5+4(x-y)+10

= 5-8+10

=7

Now new values of x and y are (5,5)

Now here x=y

So, stop

Q.3 Explain Mid-point circle drawing algorithm and using it draw a plot of the generated pixel positions in the first quadrant. Given the following data:-

Circle radius (r) =10 units, centre of the circle is at origin, initial point is (x0, y0) = (0,10). **(10)**

Ans. Midpoint Circle Algorithm

1. Input radius r and centre (Xc,Yc) and obtain the first point on the circumference of a circle centred on the origin as (x0,y0) = (0,r) i.e initialize starting position as

X=0

Y=r

2. Calculate the initial value of the decision parameter as

P=1.25 – r

3. Do

{

Plot (x,y)

If(p<0)

{

X=x+1

Y=y

P=p+2x+1

}

Else

{

X=x+1

Y=y-1

P=p+2x-2y+1

}

}

While (x≤ y)

4. Determine symmetry points in the other seven octants

5. Stop.

Given (0,10)

r=10

the initial point (x,y) =(0,10)

Scanned by CamScanner

x=0

y=10

calculate the initial decision parameter p=1-r=1-10=-9

p=-9(<0)

Plot (0,10)

Here p<0 so

X=0+1 =1

Y=10

P=-9+2*1+1=-9+3=-6

X<y i.e 1<10

Plot (1,10)

Now p=-6(<0)

X=1+1=2

Y=10

P=-6 +2*2+1=-6+5=-1

X<y i.e 2<10

Plot (2,10)

P=-1(<0)

X=2+1=3

Y=10

P=-1+2*3+1=6

X<y so

Plot (3,10)

P=6(<0)

X=x+1=3+1=4

Y=y-1=10-1=9

P=p+2x-2y+1 = 6+8-18+1=-3

X<y so

Plot (4,9)

P=-3(<0)

X=x+1=4+1=5

Y=9

P=p+2x+1=-3+10+1=8

X<y

Plot (5,9)

P=8(>0)

X=x+1=5+1=6

Y=y-1=9-1=8

P=p+2x-2y+1=8+12-16+1=5

X<y

Plot (6,8)

P=5(>0)

X=x+1=6+1=7

Y=y-1=8-1=7

P=p+2x-2y+1=5+14-14+1=6

Plot (7,7)

Now stop as x=y

Q.4. (a) Write a procedure for rotating a given object about any specified rotation axis. (5)

1. Translate the object to the origin of the coordinate system
2. Perform appropriate rotations to make the axis of rotation coincident with the z axis or any of the three coordinate axis
3. Rotate about z axis by the angle
4. Reverse the rotation about y-axis by the angle á. Then reverse the rotation about x-axis by the angle á
5. Reverse the translation carried out in step 1.

Q.4 (b) Explain the two principle vanishing point perspective projection. (5)

Two principle or two point vanishing point perspective projection occurs when the plane of projection intersects exactly two of the principle axis

$$[x\ y\ z\ 1]^* \begin{matrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

It has two center of projections, one on x axis at [-1/p 0 0 1] and another on y axis at [0 -1/q 0 1]

And their vanishing points as [1/q 0 0 1] and [0 1/q 0 1]

Scanned by CamScanner

MODEL PAPER-II

SECOND TERM EXAMINATION

THIRD SEMESTER (B. TECH) ETCS-211

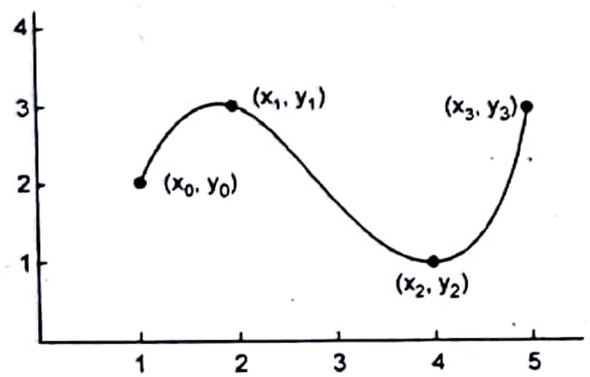
COMPUTER GRAPHICS & MULTIMEDIA-2014

Time : 1.5 hrs. **M.M. : 25**

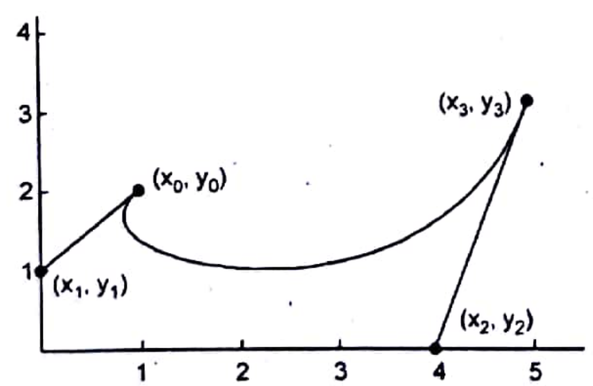
Note: ***Attempt any three questions. Q.No. 1 is compulsory.***

Q.1. (a) How are bezier curves drawn using berstein polynomials? **(5)**

Ans. The Bezier Curves is the original computer generated "French Curve" and it's discovery is attributed to the French engineer, Pierre Bezier. Let us first review ideas connecting the cubic polynomial of degree 3 passaes through the 4 points (x_k, y_k) for k = 0, 1, 2, 3 and is shown in Figure 1.



A cubic Bezier curve is defined by four points. Two of the points are endpoints to the curve, is the starting point and (x_2, y_2) is the destination point. The points (x_1, y_1) and (x_2, y_2) are control points or "handles." A parametric formula for the curve is $R[t] = (x[t], Y[t])$ where $x[t]$ and $Y[t]$ are cubic equations defined in the interval $0 \le t \le 1$ as shown in Figure 3.



Theorem (Bezier Curve): The parametric equations for $x[t]$ and $y[t]$ for the Bezier curve $R[t] = (x[t], y[t])$ are given by

$$x[t] = x_0 + 3t(x_1 - x_0) + 3t^2(x_0 + x_2 - 2x_1) + t^3(x_3 - x_0 + 3x_1 - 3x_2)$$

and $y[t] = y_0 + 3t(y_1 - y_0) + 3t^2(y_0 + y_5 - 2y_1) + t^3(y_3 - y_0 + 3y_1 - 3y_2)$ for $0 \le t \le 1$.

Bernstein Polynomials

Definition (Bernstein Polynomials): The Bernstein polynomials of degree n are

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \text{ for } i = 0, 1, 2, \ldots, n, \text{ where } \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Construction of the Bezier Curve Using Bernstein Polynomials

The coordinate function $x[t]$ and $y[t]$ for the Bezier curve we are using can written as linear combinations of the Bernstein polynomials

$(B_{0,3}[t], B_{1,2}[t], B_{2,3}[t])$;

$$x[t] = \sum_{i=0}^{3} x_i B_{i,3}[t]$$

and

$$y[t] = \sum_{i=0}^{3} Y_i B_{i,3}[t]$$

Verify that the two functions $x[t]$ and $x[t]$ are the same.

$x[t]$

$$x[t] = \sum_{i=0}^{3} x_i B_{i,3}[t]$$

Q.1. (b) List the basic properties of a Bezier curve. **(5)**

Ans. The basic properties are:

The *k-th* derivative at the start (end) of a Bezier curve depends only on the first (last) $(k+1)$ control points. Two obvious special cases:

- k=0: The Bezier curve starts at the first control point and stops at the last control point. (In general, it will not pass through any other control point.)
- k=1: The vector tangent to the Bezier curve at the start (stop) is parallel to the line connecting the first two (last two) control points.

2. A Bezier curve will always be completely contained inside of the Convex Hull of the control points. For planar curves, imagine that each control point is a nail pounded into a board. The shape a rubber band would take on when snapped around the control points is the convex hull. For Bezier curves whose control points do not all lie in a common plane, imagine the control points are tiny balls in space, and image the shape a balloon will take on if it collapses over the balls. This shape is the convex hull in that case. In any event, a Bezier curve will always lie entirely inside its planar or volumetric convex hull.

3. Closely related to the previous is the fact that adjusting the position of a control point changes the shape of the curve in a "predictable manner". Intuitively, the curve "follows" the control point. In the image below, see how a curve defined in terms of four control points (the magenta curve) changes when one of its control points is moved to the right, yielding the modified (cyan) curve.

4. There is no local control of this shape modification. Every point on the curve (with the exception of the first and last) move whenever any interior control point is moved. This property can also be observed in the image shown in the previous item.

5. Also related to property #2 is the fact that Bezier curves exhibit a variation diminishing property. Informally this means that the Bezier curve will not "wiggle" any more than the control polygon does. In other words, the curve will not wiggle unless the designer specifically introduces wiggling in the control polygon. More formally, the variation diminishing property can be stated as follows: any straight line will intersect legs of the control polygon at least as many times as it crosses the Bezier curve itself. See the example below which illustrates the property with a degree 12 Bezier curve:

6. The effect of control point Pi on the curve is at its maximum at parameter value t = i/n. Among other things, this somewhat ameleriorates problems related to the fact that there is no local control (property #4).

7. Bezier curves exhibit a symmetry property: The same Bezier curve shape is obtained if the control points are specified in the opposite order. The only difference will

Scanned by CamScanner

be the parametric direction of the curve. The direction of increasing parameter reverses when the control points are specified in the reverse order.

8. Bezier curves are invariant under affine transformations, but they are not invariant under projective transformations.

9. Bezier curves are also invariant under affine parameter transformations. That is, while the curve is usually defined on the parametric interval [0,1], an affine transformation mapping [0,1] to the interval [a,b], a?b, yields the same curve.

Q.2. (a) Describe the technique of gourand shading. How does it differ from phong shading? (3)

Ans. Phong shading refers to an interpolation technique for surface shading in 3D computer graphics. It is also called Phong interpolation or normal-vector interpolation shading.[2]Specifically, it interpolates surface normals across rasterized polygons and computes pixel colors based on the interpolated normals and a reflection model. Phong shading may also refer to the specific combination of Phong interpolation and the Phong reflection model.

Phong shading improves upon Gouraud shading and provides a better approximation of the shading of a smooth surface. Phong shading assumes a smoothly varying surface normal vector. The Phong interpolation method works better than Gouraud shading when applied to a reflection model that has small specular highlights such as the Phong reflection model.

The most serious problem with Gouraud shading occurs when specular highlights are found in the middle of a large polygon. Since these specular highlights are absent from the polygon's vertices and Gouraud shading interpolates based on the vertex colors, the specular highlight will be missing from the polygon's interior. This problem is fixed by Phong shading.

Unlike Gouraud shading, which interpolates colors across polygons, in Phong shading a normal vector is linearly interpolated across the surface of the polygon from the polygon's vertex normals. The surface normal is interpolated and normalized at each pixel and then used in a reflection model, e.g. the Phong reflection model, to obtain the final pixel color. Phong shading is more computationally expensive than Gouraud shading since the reflection model must be computed at each pixel instead of at each vertex.

In modern graphics hardware, variants of this algorithm are implemented using pixel or fragment shaders.

Q.2 (b) explain the interpolated shading. polygon mesh and gourand shading models. which is best among these? Justify your answer. (7)

Ans. A polygon mesh or unstructured grid is a collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphics and solid modeling. The faces usually consist of triangles, quadrilaterals or other simple convex polygons, since this simplifies rendering, but may also be composed of more general concave polygons, or polygons with holes.

The study of polygon meshes is a large sub-field of computer graphics and geometric modeling. Different representations of polygon meshes are used for different applications and goals. The variety of operations performed on meshes may include Boolean logic, smoothing, simplification, and many others. Network representations, "streaming" and "progressive" meshes, are used to transmit polygon meshes over a network. Volumetric meshes are distinct from polygon meshes in that they explicitly represent both the surface and volume of a structure, while polygon meshes only explicitly represent the surface (the volume is implicit). As polygonal meshes are extensively used in computer graphics, algorithms also exist for Ray tracing, collision detection, and rigid-body dynamics of polygon meshes.

Objects created with polygon meshes must store different types of elements. These include vertices, edges, faces, polygons and surfaces. In many applications, only vertices,

edges and either faces or polygons are stored. A renderer may support only 3-sided faces, so polygons must be constructed of many of these, as shown in Figure 1. However, many renderers either support quads and higher-sided polygons, or are able to triangulate polygons to triangles on the fly, making it unnecessary to store a mesh in a triangulated form. Also, in certain applications like head modeling, it is desirable to be able to create both 3- and 4-sided polygons.

A vertex is a position along with other information such as color, normal vector and texture coordinates. An edge is a connection between two vertices. A face is a closed set of edges, in which a triangle face has three edges, and a quad face has four edges. A polygon is a set of faces. In systems that support multi-sided faces, polygons and faces are equivalent. However, most rendering hardware supports only 3- or 4-sided faces, so polygons are represented as multiple faces. Mathematically a polygonal mesh may be considered an unstructured grid, or undirected graph, with additional properties of geometry, shape and topology.

Gouraud shading, named after Henri Gouraud, is an interpolation method used in computer graphics to produce continuous shading of surfaces represented by polygon meshes. In practice, Gouraud shading is most often used to achieve continuous lighting on triangle surfaces by computing the lighting at the corners of each triangle and linearly interpolating the resulting colours for each pixel covered by the triangle. Gouraud first published the technique in 1971.

An estimate to the surface normal of each vertex in a polygonal 3D model is either specified for each vertex or found by averaging the surface normals of the polygons that meet at each vertex. Using these estimates, lighting computations based on a reflection model, e.g. the Phong reflection model, are then performed to produce colour intensities at the vertices. For each screen pixel that is covered by the polygonal mesh, colour intensities can then be interpolated from the colour values calculated at the vertices.

Comparison: In comparison to Phong shading, Gourand shading's strength and weakness lies in its interpolation. If a mesh covers more pixels in screen space than it has vertices, interpolating colour values from samples of expensive lighting calculations at vertices is less processor intensive than performing the lighting calculation for each pixel as in Phong shading. However, highly localized lighting effects (such as specular highlights, e.g. the glint of reflected light on the surface of an apple) will not be rendered correctly, and if a highlight lies in the middle of a polygon, but does not spread to the polygon's vertex, it will not be apparent in a Gourand rendering; conversely, if a highlight occurs at the vertex of a polygon, it will be rendered correctly at this vertex (as this is where the lighting model is applied), but will be spread unnaturally across all neighboring polygons via the interpolation method.

The problem is easily spotted in a rendering which ought to have a specular highlight moving smoothly across the surface of a model as it rotates. Gourand shading will instead produce a highlight continuously fading in and out across neighboring portions of the model, peaking in intensity when the intended specular highlight passes over a vertex of the model.

Q.3. What are the B-Spline curves? Why are they preferred over the Bezier curves? How do we test for the first order and second order continuities? (10)

Ans. The term "B-spline" was coined by Isaac Jacob Schoenberg and is short for basis spline. B-splines can be evaluated in a numerically stable way by the de Boor algorithm. Simplified, potentially faster variants of the de Boor algorithm have been created but they suffer from comparatively lower stability.

In the computer science subfields of computer-aided design and computer graphics, the term B-spline frequently refers to a spline curve parametrized by spline functions that are expressed as linear combinations of B-splines (in the mathematical sense above). A B-spline is simply a generalisation of a Bézier curve

Given m real values ti, called knots, with

$t_0 \le t_1 \le .... \le t_{m-1}$

Scanned by CamScanner

a B-spline of degree n is a parameteric curve.

$S:[t_n, t_{m-n-1}] \to \mathbb{R}^d$

composed of a linear combination of basis B-splines $b_{i,n}$ of degree n

$$S(t) = \sum_{i=0}^{m-n-2} P_i b_{i,n}(t), t \in [t_n, t_{m-n-1}]$$

The points $P_i \in \mathbb{R}^d$ are called control points or de Boor points. There are $m-n-1$ control points, and the convex hull of the control points is a bounding volume of the curve.

The m-n-1 basis B-splines of degree n can be defined, for $n = 0, 1..., m$-2, using the Cox-de Boor recursion formula.

$$b_{j,0}(t) := \begin{cases} 1 & \text{if } t_j \le t < t_{j+1} \\ 0 & \text{otherwise} \end{cases}, j = 0,..., m-2$$

$$b_{j,n}(t) = \frac{t - t_j}{t_{j+n} - t_j} b_{n,x-1}(t) + \frac{t_{j+r,+1} - t}{t_{j+n+1} - t_{j-1}} b_{n+1,n-1}(t), j = 0,..., m-n-2.$$

Note that $j + n + 1$ can not exceed $m - 1$, which limits both j and n.

When the knots are equidistant the B-spline is said to be uniform, otherwise non-uniform. If two knots t_j are identical, any resulting indeterminate forms 0/0 are deemed to be 0.

Note that when one sums a run of adjacent n-degree basis B-splines one obtains, from this recursion.

$$\sum_{j=j}^{j''} b_{j,n}(t) = \frac{t - t_{j'}}{t_{j'+n} - t_{j'}} b_{j',n-1}(t) + \sum_{j=j'+1}^{j''} b_{n,n-1}(t) + \frac{t_{j''+n+1} - t}{t_{j''+n+1} - t_{j''+1}} b_{j''+1,n-1}(t)$$

for any sum with $0 \le j' < j'' \le m - n - 2$

When $j'' \ge j' + n - 1$ here then this sum is, by this recursion, identically equal to 1, within the limited subrange $t_{j'+n} \le t \le t_{j''+1}$ (since this interval excludes the supports of the two basis B-splines in the separate terms at the ends of this sum).

Uniform B-spline: When the B-spline is uniform, the basis B-splines for a given degree n are just shifted copies of each other. An alternative non-recursive definition for the $m - n - 1$ basis B-splines is.

$$b_{j,n}(t) = b_n(t - t_j), j = 0,..., m-n-2$$

$$b_n(t) = \frac{n+1}{n} \sum_{i=0}^{n+1} \omega_{i,n} (t - t_i)_+^n$$

$$\omega_{i,n} = \prod_{j=0, j \ne i}^{n+1} \frac{1}{t_j - t_i}$$

where

$$(t - t_i)_+^n = \begin{cases} (t - t_i)^n & \text{if } t \ge t_i \\ 0 & \text{if } t < t_i \end{cases}$$

Cardinal B-spline: Define B_0 as the indicator function (or characteristic) function of $\left[-\frac{1}{2}, \frac{1}{2}\right]$, and B_k recursively as the convolution product.

$$B_k := B_{k-1} * B_0, k = 1, 2....$$

then B_k are called (centered) cardinal B-splines. This definition goes back to Schoenberg.

B_k has compact support $\left[-\frac{k+1}{2}, \frac{k+1}{2}\right]$ and is an even function. As $k \to \infty$ the normalized cardinal B-splines tend to the Gaussian function.

Parametric Continuity: Parametric continuity is a concept applied to parametric curves describing the smoothness of the parameter's value with distance along the curve.

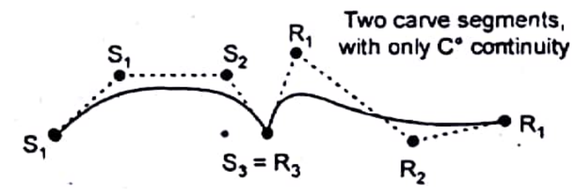
Definition: A curve can be said to have C^n continuity if

$$\frac{d^n s}{dt^n}$$

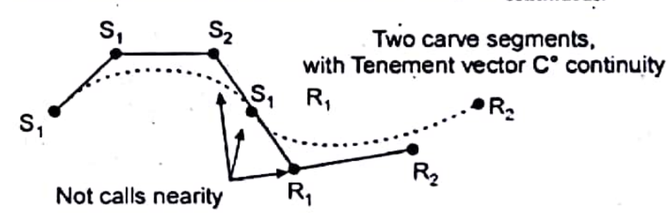
is continuous of value throughout the cuve.

As an example of a practical application of this concept, a curve describing the motion of an object with a parameter of time, must have C1 continuity for the object to have finite acceleration. For smoother motion, such as that of a camera's path while making a film, higher levels of parametric continuity are required.

Order of Continuity



Two Bezier curve segments attached that is only C^0 continuous.



Two Bezier curve segments attached in such a way that they are C^1 continuous.

The various order of parametric continuity can be described as follows:

- C^{-1}: curves include discontinuities
- C^0: curves are joined
- C^1: first derivatives are continuous
- C^2: first and second derivatives are continuous
- C^n: first through n^{th} derivatives are continuous.

The term *parametric continuity* was introduced to distinguish it from *geometric continuity* (G^n) which removes restrictions on the speed with which the parameter traces out the curve.

Geometric continuity

The concept of **geometrical** or **geometric continuity** was primarily applied to the conic sections and related shapes by mathematicians such as Leibniz, Kepler, and Poncelet. The concept was an early attempt at describing, through geometry rather than algebra, the concept of continuity as expressed through a parametric function.

The basic idea behind geometric continuity was that the five conic sections were really five different versions of the same shape. An ellipse tends to a circle as the eccentricity approaches zero, or to a parabola as it approaches one; and a hyperbola tends to a parabola as the eccentricity drops toward one; it can also tend to

Scanned by CamScanner

intersecting lines. Thus, there was *continuity* between the conic sections. These ideas led to other concepts of continuity. For instance, if a circle and a straight line were two expressions of the same shape, perhaps a line could be thought of as a circle of infinite radius. For such to be the case, one would have to make the line closed by allowing the point $x = \infty$ to be a point on the circle, and for $x = +\infty$ and $x = \infty$" to be identical. Such ideas were useful in crafting the modern, algebraically defined, idea of the continuity of a function and of ∞.

curve or surface can be described as having G^n continuity, n being the increasing measure of smoothness. Consider the segments either side of a point on a curve:

- G^0: The curves touch at the join point.
- G^1: The curves also share a common tangent direction at the join point.
- G^2: The curves also share a common center of curvature at the join point.

In general, G^n continuity exists if the curves can be reparameterized to have C^n (parametric) continuity. A re-parametrization of the curve is geometrically identical to the original; only the parameter is affected.

Equivalently, two vector functions $f(t)$ and $g(t)$ have G^n continuity if $f^{(n)}(t) \neq 0$ and $f^{(n)}(t) \equiv kg^{(n)}(t)$, for a scalar $k > 0$ (i.e., if the direction, but not necessarily the magnitude, of the two vectors is equal).

While it may be obvious that a curve would require G^1 continuity to appear smooth, for good aesthetics, such as those aspired to in architecture and sports car design, higher levels of geometric continuity are required. For example, reflections in a car body will not appear smooth unless the body has G^2 continuity.

A *rounded rectangle* (with ninety degree circular arcs at the four corners) has G^1 continuity, but does not have G^2 continuity.

Q.4. Explain the terms **(5 Marks Each)**

(a) MIDI Devices

Ans. Musical Instrument Digital Interface is a technical standard that describes a protocol, digital interface and connectors and allows a wide variety of electronic musical instruments, computers and other related devices to connect and communicate with one another. A single MIDI link can carry up to sixteen channels of information, each of which can be routed to a separate device.

MIDI carries event messages that specify notation, pitch and velocity, control signals for parameters such as volume, vibrato, audio panning, cues, and clock signals that set and synchronize tempo between multiple devices. These messages are sent to other devices where they control sound generation and other features. This data can also be recorded into a hardware or software device called a sequencer, which can be used to edit the data and to play it back at a later time.

Interfaces

A computer MIDI interface's main function is to match clock speeds between the MIDI device and the computer. Some computer sound cards include a standard MIDI connector, whereas others connect by any of various means that include the D-subminiature DA-15 game port, USB, FireWire, Ethernet or a proprietary connection. The increasing use of USB connectors in the 2000s has led to the availability of MIDI-to-USB data interfaces that can transfer MIDI channels to USB-equipped computers. Some MIDI keyboard controllers are equipped with USB jacks, and can be plugged into computers that run music software.

MIDI's serial transmission leads to timing problems. Experienced musicians can detect time differences of as small as 1/3 of a millisecond (ms) (which is how long it takes sound to travel 4 inches), and a three-byte MIDI message requires nearly 1ms for transmission. Because MIDI is serial, it can only send one event at a time. If an event is sent on two channels at once, the event on the higher-numbered channel cannot

transmit until the first one is finished, and so is delayed by 1ms. If an event is sent on all channels at the same time, the highest-numbered channel's transmission will be delayed by as much as 16ms. This contributed to the rise of MIDI interfaces with multiple in- and out-ports, because timing improves when events are spread between multiple ports as opposed to multiple channels on the same port. The term "MIDI slop" refers to audible timing errors that result when MIDI transmission is delayed.

Controllers

There are two types of MIDI controllers: performance controllers that generate notes and are used to perform music, and controllers which may not send notes, but transmit other types of real-time events. Many devices are some combination of the two types.

Performance controllers

Two-octave MIDI controllers are popular for use with laptop computers, due to their portability. This unit provides a variety of real-time controllers, which can manipulate various sound design parameters of computer-based or standalone hardware instruments, effects, mixers and recording devices.

MIDI was designed with keyboards in mind, and any controller that is not a keyboard is considered an "alternative" controller. This was seen as a limitation by composers, who were not interested in keyboard-based music, but the standard proved flexible, and MIDI compatibility was introduced to other types of controllers, including guitars, wind instruments and drum machines.

Keyboards

Keyboards are by far the most common type of MIDI controller. These are available in sizes that range from 25-key, 2-octave models, to full-sized 88-key instruments. Some are keyboard-only controllers, though many include other real-time controllers such as sliders, knobs, and wheels. Commonly, there are also connections for sustain and expression pedals. Most keyboard controllers offer the ability to split the playing area into "zones", which can be of any desired size and can overlap with each other. Each zone can respond to a different MIDI channel and a different set of performance controllers, and can be set to play any desired range of notes. This allows a single playing surface to target a number of different devices. MIDI capabilities can also be built into traditional keyboard instruments, such as grand pianos and Rhodes pianos. Pedal keyboards can operate the pedal tones of a MIDI organ, or can drive a bass synthesizer such as the revived Moog Taurus.

(b) Backface Detection: Back-face detection is a fast object space algorithm based upon the inside-outside test for identifying the backface of a polyhedron. Suppose the point (x,y,z) is inside a polygon surface with plane parameters A,B,C and D.

If $Ax+By+Cz+D < 0$

When an inside point is along the line of sight to the surface, the polygon must be a back face.

The direction of the light face can be identified by examining the result

$N.V > 0$ where

N is the normal vector to the polygon face with Cartesian components A,B,C. and V is a vecto in the viewing direction from the eye position.

We know that the dot product of two vectors, gives the product of the lengths of the two vectors times the cosine of the angle between them. If the vectors are in the same direction i.e. , $0 <= < \Phi < \pi/2$

Then the cosine is positive and the overall dot product is positive however if the directions are opposite then the overall dot product is –ve.

Scanned by CamScanner

MODEL PAPER-II
END TERM EXAMINATION
THIRD SEMESTER (B. TECH) ETCS-211
COMPUTER GRAPHICS & MULTIMEDIA-2014

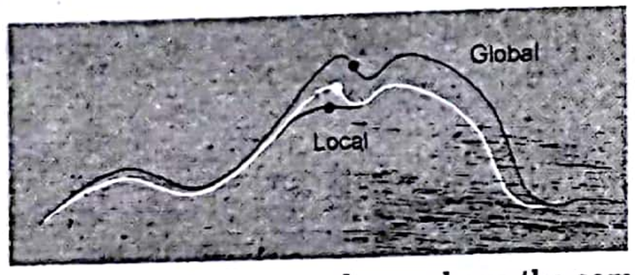
Time : 3 hrs. **M.M. : 75**

Note: ***Attempt any three questions. Q.No. 1 is compulsory.***

Q.1. Give the difference between local and global control property of a curve. **(3)**

Ans. A small change in control point changes the whole curve shape is known as global control of a curve.

Whereas on changing a control point only a curve segment changes is known as local control of a curve.



Q.1. (*b*) What happens when two polygons have the same Z value and Z buffer algorithm is used. **(3)**

Ans. When an object is rendered, the depth of a generated pixel (z coordinate) is stored in a buffer (the z-buffer or depth buffer). This buffer is usually arranged as a two-dimensional array (x-y) with one element for each screen pixel. If another object of the scene must be rendered in the same pixel, the method compares the two depths and overrides the current pixel if the object is closer to the observer. The chosen depth is then saved to the z-buffer, replacing the old one. In the end, the z-buffer will allow the method to correctly reproduce the usual depth perception: a close object hides a farther one. This is called z-culling.

The granularity of a z-buffer has a great influence on the scene quality: a 16-bit z-buffer can result in artifacts (called "z-fighting") when two objects are very close to each other. A 24-bit or 32-bit z-buffer behaves much better, although the problem cannot be entirely eliminated without additional algorithms. An 8-bit z-buffer is almost never used since it has too little precision.

Q.1. (*c*) Prove that any two successive 3D rotations about a given rotation axis commutative. **(4)**

Ans. Successive rotations

Using the trigonometric identities for the sine and cosine of the sum of two angles, we can express the elements of the product matrix for two successive rotations in the xy plane about the coordinate origin as

$$\begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Since the addition of angles is commutative, this composite transformation is commutative. The commutative property of two successive rotations can also be shown by comparing the elements of the composite matrix R1 R2 with the elements of the composite matrix R2 R1.

Q.1. (*d*) write the steps to clip a volume in 3D against canonical view volumes. **(6)**

Ans. 3D world ->Normalize to the canonical view volume -> clip against canonical view volume -> project on to projection plane -> translate into viewport.

Method 1:

1.Extend 3D coordinates to homogeneous coordinates
2.Apply Npar or Nper to normalize the homogeneous coordinates
3.Divide by W to go back to 3D coordinates
4.Clip in 3D against the appropriate view volume (parallel or perspective)
5.Extend 3D coordinates to homogeneous coordinates
6.Perform projection using either Mort or Mper (with d=1)
7.Translate and Scale into device coordinates
8.Divide by W to go to 2D coordinates

Method 2:

1.Extend 3D coordinates to homogeneous coordinates
2.Apply Npar or Nper' to normalize the homogeneous coordinates
3. Clip in homogeneous coordinates
4.Translate and Scale into device coordinates
5.Divide by W to go to 2D coordinates

Q.1. (*d*) What is Z buffer algorithm? Which is better and why, Z-buffer or A-buffer? **(2)**

Ans. In computer graphics, z-buffering, also known as depth buffering, is the management of image depth coordinates in three-dimensional (3-D) graphics, usually done in hardware, sometimes in software. It is one solution to the visibility problem, which is the problem of deciding which elements of a rendered scene are visible, and which are hidden. The A buffer algorithm is more efficient than Z buffer as it adds one more field called depth field which holds a positive or negative real number.by using this depth field it can calculate intensities of various layers as present in a canvas which is not possible by using Z buffer algorithm.

Q.1. (*g*) Find the values of Fx, Fy, Fz for isometric projection. **(5)**

Ans. In isometric projection the projection plane makes equal angle with each principal axis. If the projection plane normal is (dx, dy, dz)

Then $|dx| = |dy| = |dz|$

Transforming the unit vector along the x-axis using isometric projectionmatrix yields.

$$[Ux] = [1001] \begin{matrix} \cos\phi & \sin\phi\sin\theta & 0 & 0 \\ 0 & \cos\theta & 0 & 0 \\ \sin\phi & -\cos\phi\sin\theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Thus the forehoretening factors are

$$f_x^2 = \cos^2\phi + \sin^2\phi\sin^2\theta$$
$$f_y^2 = \cos^2\phi$$
$$f_z^2 = \sin^2\phi + \cos^2\phi\sin^2\theta$$

Replacing $\cos^2\phi$ with $1-\sin^2$ and $\cos^2\phi$ with $1\text{-}\sin^2\phi$

We get

$$\sin^2\phi = \sin^2\phi\, 1 - \sin^2\phi$$

similarly for F_x^2 and F_z^2

Therefore there are four possible isometric projections:

(*i*) $\phi = -45°\ \theta = 35.26°$

Scanned by CamScanner

(ii) $\phi = -45° \; \theta = -35.26°$

(iii) $\phi = 45° \; \theta = 35.26°$

(iv) $\phi = -45° \; \theta = -35.26°$

For each case the forehortening factor is $\sqrt{2}\sqrt{3} = 0.8165$

Q.1. (h) Explain BITMAP, PIXMAP. **(2)**

Ans. Bitmap is a mapping from some domain to bits, that is, values which are zero or one. It is also called a bit array or bitmap index.

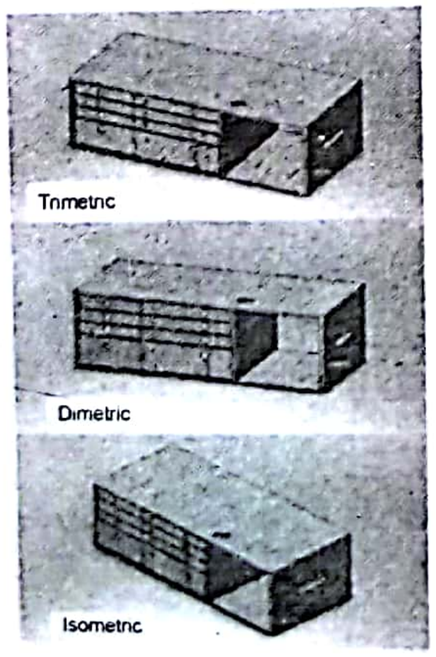
Pixmap refers to a map of pixels, where each one may store more than two colors, thus using more than one bit per pixel.

Q.2. What are the different kinds of projections? explain perspective projection and parallel projection in detail with the help of neat diagrams. **(12.5)**

Ans. Types of Parallel projection: In parallel projection, the lines of sight from the object to the projection plane are parallel to each other. Within parallel projection there is an ancillary category known as "pictorials". Pictorials show an image of an object as viewed from a skew direction in order to reveal all three directions (axes) of space in one picture. Because pictorial projections innately contain this distortion, in the rote, drawing instrument for pictorials, some liberties may be taken for economy of effort and best effect.

Orthographic projection: The Orthographic projection is derived from the principles of descriptive geometry and is a two-dimensional representation of a three-dimensional object. It is a parallel projection (the lines of projection are parallel both in reality and in the projection plane). It is the projection type of choice for working drawings.

Axonometric projection: Within parallel projection there is a subcategory known as Pictorials. Pictorials show an image of an object as viewed from a skew direction in order to reveal all three directions (axes) of space in one picture. Parallel projection pictorial instrument drawings are often used to approximate graphical perspective projections, but there is attendant distortion in the approximation. Because pictorial projections inherently have this distortion, in the instrument drawing of pictorials, great liberties may then be taken for economy of effort and best effect. Parallel projection pictorials rely on the technique of axonometric projection ("to measure along axes").



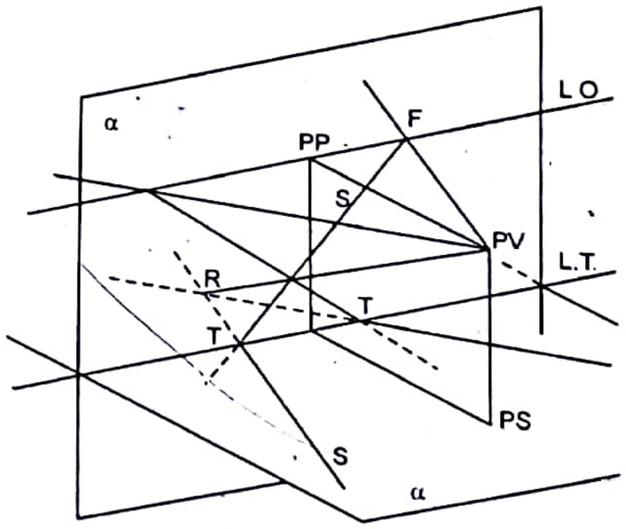
Isometric projection: In isometric pictorials (for protocols see isometric projection), the direction of viewing is such that the three axes of space appear equally foreshortened, of which the displayed angles among them and also the scale of foreshortening are universally known. However in creating a final, isometric instrument drawing, in most cases a full-size scale, i.e., without using a foreshortening factor, is employed to good effect because the resultant distortion is difficult to perceive.

Dimetric projection: In dimetric pictorials (for protocols see dimetric projection), the direction of viewing is such that two of the three axes of space appear equally foreshortened, of which the attendant scale and angles of presentation are determined according to the angle of viewing; the scale of the third direction (vertical) is determined separately. Approximations are common in dimetric drawings.

Trimetric projection: In trimetric pictorials (for protocols see trimetric projection), the direction of viewing is such that all of the three axes of space appear unequally foreshortened. The scale along each of the three axes and the angles among them are determined separately as dictated by the angle of viewing. Approximations in Trimetric drawings are common.

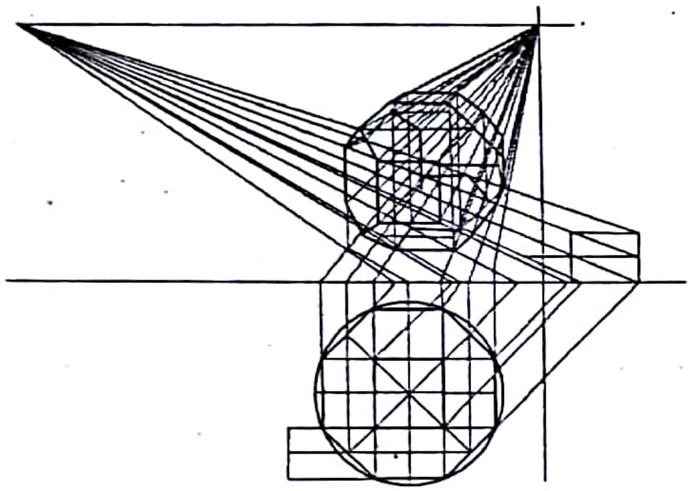
Oblique projection: In oblique projections the parallel projection rays are not perpendicular to the viewing plane as with orthographic projection, but strike the projection plane at an angle other than ninety degrees. In both orthographic and oblique projection, parallel lines in space appear parallel on the projected image. Because of its simplicity, oblique projection is used exclusively for pictorial purposes rather than for formal, working drawings. In an oblique pictorial drawing, the displayed angles among the axes as well as the foreshortening factors (scale) are arbitrary. The distortion created thereby is usually attenuated by aligning one plane of the imaged object to be parallel with the plane of projection thereby creating a true shape, full-size image of the chosen plane. Special types of oblique projections are cavalier projection and cabinet projection.

Perspective projection



Axonometric projection of a scheme displaying the relevant elements of a vertical picture plane perspective. The standing point (P.S.) is located on the ground plane TT, and the point of view (P.V.) is right above it. P.P. is its projection on the picture plane a. L.O. and L.T. are the horizon and the ground lines (linea d'orizzonte andlinea di terra). The bold lines s and q lie on a, and intercept a in Ts and Tq respectively. The parallel lines through P.V. (in red) intercept L.O. in the vanishing points Fs and Fq: thus one can draw the projections s'and q', and hence also their intersection R', the projection of R.

Scanned by CamScanner



Perspective of a geometric solid using two vanishing points. In this case, the map of the solid (orthogonal projection) is drawn below the perspective, as if bending the ground plane.

Perspective (graphical): Perspective projection is a linear projection where three dimensional objects are projected on a picture plane. This has the effect that distant objects appear smaller than nearer objects.

It also means that lines which are parallel in nature (that is, meet at the point at infinity) appear to intersect in the projected image, for example if railways are pictured with perspective projection, they appear to converge towards a single point, called vanishing point. Photographic lenses and the human eye work in the same way, therefore perspective projection looks most realistic.[1] Perspective projection is usually categorized into one-point, two-point and three-point perspective, depending on the orientation of the projection plane towards the axes of the depicted object.

Graphical projection methods rely on the duality between lines and points, whereby two straight lines determine a point while two points determine a straight line. The orthogonal projection of the eye point onto the picture plane is called the principal vanishing point (P.P. in the scheme on the left, from the Italian termpunto principale, coined during the renaissance).

Two relevant points of a line are:

- its intersection with the picture plane, and
- its vanishing point, found at the intersection between the parallel line from the eye point and the picture plane.

The principal vanishing point is the vanishing point of all horizontal lines perpendicular to the picture plane. The vanishing points of all horizontal lines lie on the horizon line, obviously. If, as is often the case, the picture plane is vertical, all vertical lines are drawn vertically, and have no finite vanishing point on the picture plane. Various graphical methods can be easily envisaged for projecting geometrical scenes. For example, lines traced from the eye point at 45° to the picture plane intersect the latter along a circle whose radius is the distance of the eye point from the plane, thus tracing that circle aids the construction of all the vanishing points of 45° lines; in particular, the intersection of that circle with the horizon line consists of two distance points. They are useful for drawing chessboard floors which, in turn, serve for locating the base of objects on the scene. In the perspective of a geometric solid on the right, after choosing the principal vanishing point -which determines the horizon line- the 45° vanishing point on the left side of the drawing completes the characterization of the (equally distant) point of view. Two lines are drawn from the orthogonal projection of each vertex, one at 45° and one at 90° to the picture plane. After intersecting the ground line, those lines go toward

the distance point (for 45°) or the principal point (for 90°). Their new intersection locates the projection of the map. Natural heights are measured above the ground line and then projected in the same way until they meet the vertical from the map.

OR

Q.3. What is line segment clipping? Describe the various clipping categories into which the line segments are categorized. What is the significance of each category? Explain Sutherland and Hodgman polygon clipping algo. (12.5)

Ans. 'line clipping' is the process of removing lines or portions of lines outside of an area of interest. Typically, any line or part thereof which is outside of the viewing area is removed.

There are two common algorithms for line clipping:

1.The Cohen-Sutherland Line-Clipping Algorithm

The more efficient Cohen-Sutherland Algorithm performs initial tests on a line to determine whether intersection calculations can be avoided.

Steps for Cohen-Sutherland algorithm

1.End-points pairs are check for trivial acceptance or trivial rejected using the outcode.

2.If not trivial-accepance or trivial-rejected, divided into two segments at a clip edge.

3.Iteratively clipped by testing trivial-acceptance or trivial-rejected, and divided into two segments until completely inside or trivial-rejected.

Trivial acceptance/reject test

To perform trivial accept and reject tests, we extend the edges of the clip rectangle to divide the plane of the clip rectangle into nine regions. Each region is assigned a 4-bit code deteermined by where the region lies with respect to the outside halfplanes of the clip-rectangle edges. Each bit in the outcode is set to either 1 (true) or 0 (false); the 4 bits in the code correspond to the following conditions:

- Bit 1 : outside halfplane of top edge, above top edge

$Y > Ymax$

- Bit 2 : outside halfplane of bottom edge, below bottom edge

$Y < Ymin$

- Bit 3 : outside halfplane of right edge, to the right of right edge

$X > Xmax$

- Bit 4 : outside halfplane of left edge, to the left of left edge

$X < Xmin$

Sutherland and Hodgman's polygon-clipping Algorithm.

Four Cases of polygon clipping against one edge

The clip boundary determines a visible and invisible region. The edges from vertex i to vertex i+1 can be one of four types:

- Case 1 : Wholly inside visible region - save endpoint
- Case 2 : Exit visible region - save the intersection
- Case 3 : Wholly outside visible region - save nothing
- Case 4 : Enter visible region - save intersection and endpoint

Because clipping against one edge is independent of all others, it is possible to arrange the clipping stages in a pipeline. The input polygon is clipped against one edge and any points that are kept are passed on as input to the next stage of the pipeline.

Q.4.(*a*) Example Run Length Encoding with Example. (10)

Ans. Run-length encoding (RLE) is a very simple form of data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original

Scanned by CamScanner

run. This is most useful on data that contains many such runs: for example, relatively simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could potentially double the file size.

For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. Let us take a hypothetical single scan line, with B representing a black pixel and W representing white:

WWWWWWWWWWWWWWWWBBBWWWWWWWWWWWWWWW

Lossless compression recreates a compressed file as an identical match to its original form. All lossless compression uses techniques to break up a file into smaller segments, for storage or transmission, that get reassembled later. Lossless compression is used for files, such as applications, that need to be reproduced exactly like the original file. Lossy compression, on the other hand, eliminates repeated or "unnecessary" pieces of data, as we discussed above. When such a file is decompressed, you get the compression software's re-interpretation of the original file.

Lossy compression can't be used to compress anything that needs to be reproduced exactly – it can't just toss out redundant pieces and hope to program will still work. Instead, lossy compression will more often be used with data that is open to some level of human interpretation, such as an image, where the results can be "fudged" the tiniest bit so that files can get smaller without, in theory, anyone noticing. The digital cameras we use today, for example, offer varying levels of compression -- in effect reworking the data of an image, throwing out whatever it can to reduce the image size while still retaining the snazzy picture you thought you took.

Q.6 (*b*) Explain Various multimedia authoring Tools **(5.5)**

Ans. The integration of audio, video, graphics and text on the desktop promises to fundamentally challenge the centuries-old model of the printed document as the basis for information exchange. Before this potential can be realized, however, systems must be devised that enable the production and presentation of complex, inter-related media objects. These systems are generically called multimedia authoring tools.

In this article, we consider the development of multimedia authoring tools, examine the current state of the art, and then discuss a set of research challenges that need to be addressed before the full potential of multimedia output technology can be effectively utilized to share information.

Types of Authoring Tools:

1. Card- or page-based tools.
2. Icon-based, event-driven tools.
3. Time-based tools.

Card-based or page-based tools:

1. The elements are organized as pages of a book or a stack of cards.
2. Card-or page-based authoring systems
3. best used when the bulk of your content consists of elements that can be viewed individually
4. the pages of a book or cards in a card file.
5. link these pages or cards into organized sequences.
6. jump, on command, to any page
7. play sound elements and launch animations and digital video.

Icon-or object-based, event-driven tools

1. multimedia elements and interaction cues (events) are organized as objects in a structural framework or process.
2. simplify the organization of your project
3. display flow diagrams of activities along branching paths.
1. In complicated navigational structures, this charting is particularly useful during development

Time-based tools

1. Elements and events are organized along a timeline with resolutions as high as or higher than 1/30 second.
2. Time-based tools
3. best to use when you have a message with a beginning and an end.
4. played back at a speed that you can set
5. Other elements (such as audio events) are triggered at a given time or location in the sequence of events.
6. Jumps to any location in a sequence
1. navigation and interactive control.

COMPUTER GRAPHICS AND MULTIMEDIA–[ETCS-211] THIRD SEMESTER (B.TECH) FIRST TERM EXAMINATION–[2014]

Time : 1 hrs. **M.M. : 30**

Note: *Attempt any three questions in total including Question No. 1 which is compulsory.*

Q.1.(*a*) What is meant by Scan conversion of a line? Mention three disadvantages of it.

Ans. Scan conversion or scan rate converting is a video processing technique for changing the vertical / horizontal scan frequency of video signal for different purposes and applications.

Problem: Approximation of lines by a finite number of pixels

- Costly rounding of operations
- Unnecessary floating point arithmetic
- Error accumulation

Q.1.(*b*) Show that two successive 3D translations are additive i.e. TxTy=TyTx or TyTz=TzTy or TxTy=TyTx.

Ans.

ROTATIONS

Two successive rotation applied to point *P* produce the transformed poition:

$$P' = R(\theta_2) . \{R(\theta_1) . P\}$$
$$= \{R(\theta_2) . R(\theta_1) . P$$

By multiplication the two rotation matrices, we can verify that two successive rotation are additive:

$$R(\theta_2) . R(\theta_1) = R(\theta_1 + \theta_2)$$

So that the final rotated coordinates can be calculated with the composite rotation matrix as:

$$P' = R(\theta_1 + \theta_2) . P$$

Q.1.(*c*) define the terms pixel, bitmap & resolution.

Ans. PIXEL: The pixel (a word invented from "picture element") is the basic unit of programmable color on a computer display or in a computer image.

BITMAP: A representation in which each item corresponds to one or more bits of information, especially the information used to control the display of a computer screen.

RESOLUTION: Resolution is the number of pixels (individual points of color) contained on a display monitor, expressed in terms of the number of pixels on the horizontal axis and the number on the vertical axis. The sharpness of the image on a display depends on the resolution and the size of the monitor.

Q.1.(*d*) what are the differences between the raster scan and random scan display devices? **(2.5×4=10)**

Ans.

Property	Raster scanning System	Random Scanning system
Resolution	It has poor or **less Resolution** because picture definition is stored as a intensity value.	It has **High Resolution** because it stores picture definition as a set of line commands.

Scanned by CamScanner

Electron-Beam	**Electron Beam** is directed from top to bottom and one row at a time on screen, but electron beam is directed to whole screen.	**Electron Beam** is directed to only that part of screen where picture is required to be drawn, one line at a time so also called **Vector Display.**
Cost	It is **less expensive** than Random Scan System.	It is **Costlier** than Raster Scan System.
Refresh Rate	Refresh rate is **60 to 80 frame per second.**	Refresh Rate depends on the number of lines to be displayed i.e **30 to 60 times per second.**
Picture Definition	It Stores **picture definition** in **Refresh Buffer** also called **Frame Buffer.**	It Stores picture definition as a set of **line commands** called **Refresh Display File.**
Line Drawing	**Zig – Zag line** is produced because plotted value are discrete.	**Smooth line** is produced because directly the line path is followed by electron beam .
Realism in display	It contains shadow, advance shading and hidden surface technique so gives the **realistic display** of scenes.	It does not contain shadow and hidden surface technique so it can not give realistic display of scenes.
Image Drawing	It uses **Pixels** along scan lines for drawing an image.	It is designed for line drawing applications and uses various mathematical function to draw.

Q.2.(a) Deduce a matrix to shear an object by unit 5 along X – axis and then by unit 4 along Y-axis. **(3)**

$$\text{shear along } x \text{ axis} = \begin{bmatrix} 1 & \text{shear(x)} \\ 0 & 1 \end{bmatrix}$$

$$\text{shear along y axis} = \begin{bmatrix} 1 & 0 \\ \text{shear(y)} & 1 \end{bmatrix}$$

$$15 \times 10 = 120010404$$

Q.2.(b) Use Cohen-Sutherland algorithm to clip the line P1 (1,2) to P2 (8,5) with respect to the clipping window given by 0<=x<=7, 0<=y<=4. **(7)**

Ans. 1. Cohen-Sutherland Line-Clipping Algorithm: The more efficient Cohen-Sutherland Algorithm performs initial tests on a line to determine whether intersection calculations can be avoided.

Steps for Cohen-Sutherland algorithm

1. End-points pairs are check for trivial acceptance or trivial rejected using the outcode.

2. If not trivial-acceptance or trivial-rejected, divided into two segments at a clip edge.

3.Iteratively clipped by testing trivial-acceptance or trivial-rejected, and divided into two segments until completely inside or trivial-rejected.

Trivial acceptance/reject test

To perform trivial accept and reject tests, we extend the edges of the clip rectangle to divide the plane of the clip rectangle into nine regions. Each region is assigned a 4-bit code deteermined by where the region lies with respect to the outside halfplanes of the clip-rectangle edges. Each bit in the outcode is set to either 1 (true) or 0 (false); the 4 bits in the code correspond to the following conditions:

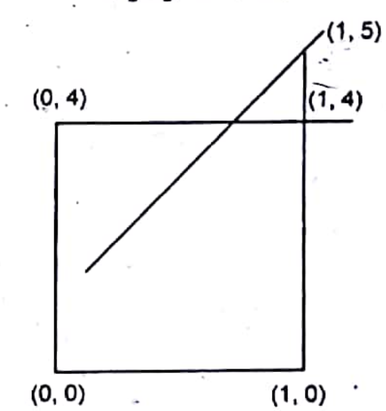
- Bit 1 : outside half plane of top edge, above top edge
  Y > Ymax
- Bit 2 : outside half plane of bottom edge, below bottom edge
  Y < Ymin
- Bit 3 : outside half plane of right edge, to the right of right edge
  X > Xmax
- Bit 4 : outside halfplane of left edge, to the left of left edge
  X < Xmin

Window Size

$$(x_1, y_1) = (0, 0)$$

and

$$(x_2, y_2) = (7, 4)$$



given Point P_1 and P_2 as (1, 2) and (8, 5)

Our codes

1001	1000	1010
0001	0000	0010
0101	0100	0110

Outcode for P_1 = 0000
for P_2 = 1010

Among P_1 and P_2 we get 000
So, the line is partially visible

$$\text{Calculate Slope} = m = \frac{y_2 - y_1}{y_2 - y_1}$$

$$= \left(\frac{5-2}{8-1}\right) = \frac{3}{1}$$

Calculated interection point

$$m = \frac{3}{1} = \frac{y-5}{7-8}$$

Scanned by CamScanner

$$3 = 7(y-5)$$
$$-3 = 7y - 35$$
$$y = \frac{33}{1} \text{ i.e. } (7.33)$$

Q.3.(a) Derive the transformation matrix that reflects an Object w.r.t a line Y=X. **(5)**

Ans. example: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ This transformation matrix will affect a reflection in the line y=x for the point matrix, as shown in the example below:

Transformation Matrix	Point Matrix	Multiplication	New Point Matrix
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 4 \\ 3 \end{bmatrix}$	$\begin{bmatrix} (4x_0)+(3x_1) \\ (4x_1)+(3x_0) \end{bmatrix}$	$\begin{bmatrix} 3 \\ 4 \end{bmatrix}$

Q.3.(b) scan convert a line using Bresenhams' Line Drawing algorithm from (1, 1) to (8, 5). **(5)**

Ans.

$$\Delta x = 8 - 1 = 7$$
$$\Delta y = 5 - 1 = 4$$
$$2\Delta x = 14$$
$$2\Delta y = 8$$
$$P_0 = 2\Delta y - \Delta x = 8 - 7 = 1$$
$$P_k > 0 \text{ so next point is } (2, 2)$$
$$P_{x+1} = P_k + 2\Delta y - 2\Delta x$$
$$P_1 = 1 + 2(4) - 2(7) = -5$$

next point 4 (3, 2)

and

$$P_2 = -5 + 8 = 4$$

next point is (4, 3)

$$P_3 = 4 + 8 - 14 = -2$$

next point is (5, 3)

$$P_4 = -2 + 8 = 6$$

next point is (6, 4)

$$P_5 = 6 + 8 - 14 = 0$$

next point is = (7, 5)

$$P_6 = 0 + 8 - 14 = -6$$

So next point is (8, 5) stop.

Points are (1, 1), (2, 2), (3, 2), (4, 3), (5, 3), (6, 4), (7, 5) and (8, 5)

Q.4. (a) Plot a circle using Mid-Point Circle Drawing Algorithm with center (2, 4) and radius 7. **(5)**

Ans. give radius $r = 7$

$(x, y) = (0, 7)$ (taking 0, 0 as centre)

Initial Decision Parameter $= 1 - r = 1 - 7$

$$P_1 = -6\ (20)$$

Plot, (1, 7)

$$P_1 = P + 2x + 1$$
$$P_1 = -6 + 2(1) + 1 = -3$$

again ($P_1 < 0$)

Plot (2, 7)

$$P_2 = -3 + 2(2) + 1 = 2$$

Now $P_1 > 0$ so plot (3, 6) (3, 6)

$$P_1 = 2 + 2(3) - 2(y) + 1$$
$$P_3 = 2 + 6 - 12 + 1 = -3$$

So plot (4, 6)

$$P_4 = -3 + 2(4) + 1 = 6$$

So plot (5, 5)

Now ($x = y$) so stop.

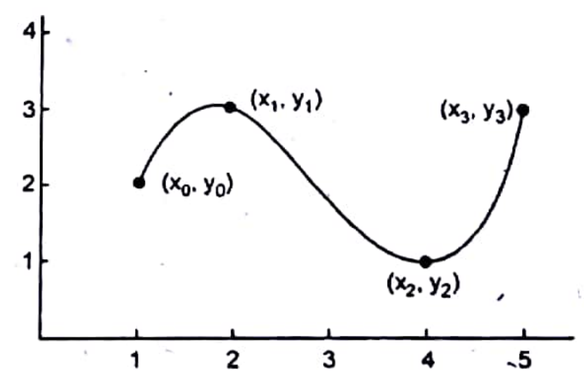
add (2, 4) to each calculated point

(0, 7), (1, 7), (2, 7), (3, 0), (4, 6), (5, 5)

(2, 11), (3, 11), (4, 11), (5, 10), (6, 10), (7, 9) (points acc. to centre 2, 4)

Q.4. (b) how are bezier curves drawn using bersntein polynomials? **(5)**

Ans. Bézier Curve. Let us first review ideas concerning the cubic polynomial of degree 3 passes through the 4 points (x_k, y_k) for $k = 0, 1, 2, 3$ and is shown



A cubic Bézier curve is defined by four points. Two of the points are endpoints to the curve, is the starting point and (x_3, y_3) is the destination point. The points (x_1, y_1) and (x_2, y_2) are control points or "handles." A parametric formula for the curve is

$R[t] = \{x[t], y[t]\}$ where $x[t]$ and $y[t]$ are cubic equations defined in the interval $0 \le t \le 1$ as shown.

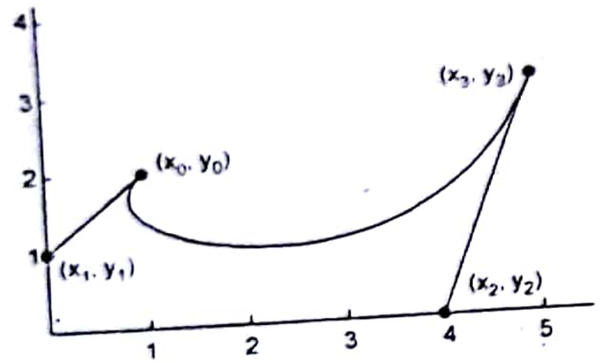
Theorem (Bezier Curve): The parametric equations for $x(t)$ and $y(t)$ for the Bezier curve $R[t] = \{x(t), y(t)\}$ are given by

$$x[t] = x_0 + 3t(x_1 - x_0) + 3t^2(x_0 + x_2 - 2x_1) + t^3(x_3 - x_0 + 3x_1 - 3x_2)$$

and

$$y[t] = y_0 + 3t(y_1 - y_0) + 3t^2(y_0 + y_2 - 2y_1) + t^3(y_3 - y_0 + 3y_1 - 3y_2) \text{ for } 0 \le t \le 1.$$

Scanned by CamScanner



Bernstein Polynomials.

Definition (Bernstein Polynomials). The Bernstein polynomials of degree n are

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad \text{for } i = 0, 1, 2, \ldots n$$

where

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Construction of the Bezier Curve using Bernstein Polynomials.

The coordinate functions $x[t]$ and $y[t]$ for the Bezier curve we are using can written as a linear combinations of the Bernstein polynomials.

$B_{0,3}[t], B_{1,3}[t], B_{2,3}[t], B_{3,3}[t]$):

$$X[t] = \sum_{i=0}^{3} X_i B_{i,3}[t]$$

and

$$Y[t] = \sum_{i=0}^{3} Y_i B_{i,3}[t]$$

Verify that the two functions $x[t]$ and $x[t]$ are the same.

$x[t]$

$$x[t] = \sum_{i=0}^{3} x_i, B_{i,3}[t]$$

COMPUTER GRAPHICS AND MULTIMEDIA–[ETCS-211] THIRD SEMESTER (B.TECH) SECOND TERM EXAMINATION–[2014]

Time : 1 hrs. **M.M. : 30**

Note: *Attempt any three questions in total including Question No. 1 which is compulsory.*

Q.1.(a) Differentiate between perspective and parallel projection. **(2.5)**

Ans. Perspective projection is seeing things larger when they're up close and smaller at a distance. It is a three-dimensional projection of objects on a two-dimensional medium such as paper. It allows an artist to produce a visual reproduction of an object which resembles the real one.

The center of projection in a perspective projection is a point which is at a distance from the viewer or artist. Objects located at this point appear smaller and will appear bigger when they are drawn closer to the viewer. Perspective projection produces a more realistic and detailed representation of an object allowing artists to create scenes that closely resemble the real thing. The other type of projection which is also used aside from perspective projection is parallel projection.

Parallel projection, on the other hand, resembles seeing objects which are located far from the viewer through a telescope. It works by making light rays entering the eyes parallel, thus, doing away with the effect of depth in the drawing. Objects produced using parallel projection do not appear larger when they are near or smaller when they are far. It is very useful in architecture. However, when measurements are involved, perspective projection is best.

It provides an easier way of reproducing objects on any medium while having no definite center of projection. When it is not possible to create perspective projection, especially in cases where its use can cause flaws or distortions, parallel projection is used.

1. Perspective projection is representing or drawing objects which resemble the real thing while parallel projection is used in drawing objects when perspective projection cannot be used.

2. Parallel projection is much like seeing objects through a telescope, letting parallel light rays into the eyes which produce visual representations without depth while perspective projection represents objects in a three-dimensional way.

3. In perspective projection, objects that are far away appear smaller, and objects that are near appear bigger while parallel projection does not create this effect.

4. While parallel projection may be best for architectural drawings, in cases wherein measurements are necessary, it is better to use perspective projection.

Q.1.(b) Differentiate between image space and object space methods of hidden surface removal. **(2.5)**

Ans. Image Space Vs Object Space

In 3D computer animation images have to be stored in frame buffer converting two dimensional arrays into three dimensional data. This conversion takes place after many calculations like hidden surface removal, shadow generation and Z buffering. These calculations can be done in Image Space or Object Space. Algorithms used in image space for hidden surface removal are much more efficient than object space algorithms. But object space algorithms for hidden surface removal are much more

Scanned by CamScanner

functional than image space algorithms for the same. The combination of these two algorithms gives the best output.

Image Space: The representation of graphics in the form of Raster or rectangular pixels has now become very popular. Raster display is very flexible as they keep on refreshing the screen by taking the values stored in frame buffer. Image space algorithms are simple and efficient as their data structure is very similar to that of frame buffer. The most commonly used image space algorithm is Z buffer algorithm that is used to define the values of z coordinate of the object.

Object Space: Space object algorithms have the advantage of retaining the relevant data and because of this ability the interaction of algorithm with the object becomes easier. The calculation done for the color is done only once. Object space algorithms also allow shadow generation to increase the depth of the 3 dimensional objects on the screen. The incorporation of these algorithms is done in software and it is difficult to implement them in hardware.

- Image space algorithms are much more efficient than object space algorithms
- Object space algorithms are much more functional than image space algorithms
- Color calculation in object space algorithms is done only one time and is retained by it but in image space algorithm the calculation once done is over written later.

Q.1.(c) Define the terms multimedia, interactive multimedia and non interactive multimedia. (2.5)

Ans. MULTIMEDIA: - ***Multimedia*** is the field concerned with the computer-controlled integration of text, graphics, drawings, still and moving images (Video), animation, audio, and any other media where every type of information can be represented, stored, transmitted and processed digitally.

INTERACTIVE MULTIMEDIA: Interactive multimedia has been called a "hybrid technology." It combines the storage and retrieval capabilities of computer database technology with advanced tools for viewing and manipulating these materials. Multimedia has a lot of different connotations, and definitions vary depending on the context. For the purposes of this Guide, in the context of upper secondary and postsecondary education, interactive multimedia is defined by three criteria:

- Interactive Multimedia is any package of materials that includes some combination of texts, graphics, still images, animation, video, and audio;
- These materials are packaged, integrated, and linked together in some way that offers users the ability to browse, navigate and analyze these materials through various searching and indexing features, as well as the capacity to annotate or personalize these materials;
- Interactive multimedia is always "reader-centered." In interactive multimedia, the reader contols the experience of reading the material by being able to select among multiple choices, choosing unique paths and sequences through the materials. One of the key features of interactive multimedia is the ability to navigate through material in whatever ways are most meaningful for individual users.

Non Interactive Multimedia: A fixed, self-contained and pre-conceived passive experience, fictional or not, typically featuring a narrative, often with a beginning, a middle and an end. The audience members are observers and not able to interact and participate and therefore incapable of influencing or changing the experience directly. The various forms of non-interactive entertainment have their own immersion and emotional delivery language: For novels it'd be choice of structure, words, perspective,

pace, perception, etc whilst audiovisual media like film and TV use narrative structure, camera framing, editing, lighting, sound, music, actor performance, etc.

Q.1.(d) Explain the file formats – AVI, MOV, MPEG and JPEG? (2.5×4=10)

Ans. AVI: Audio Video Interleaved (also **Audio Video Interleave**), known by its initials **AVI**, is a multimedia container format introduced by Microsoft in November 1992 as part of its Video for Windows software. AVI files can contain both audio and video data in a file container that allows synchronous audio-with-video playback. Like the DVD video format, AVI files support multiple streaming audio and video, although these features are seldom used.

MOV: MOV, an MPEG-4 video container, is a common multimedia format often used in Apple's Quicktime for saving movies and other video files, using a proprietary compression algorithm developed by **Apple Computer**, compatible with both Macintosh and Windows platforms.

MPEG: ***Moving Picture Experts Group***, and pronounced m-peg, is a working group of the ISO. The term also refers to the family of digital video compression standards and file formats developed by the group. MPEG generally produces better-quality video than competing formats, such as Video for Windows, Indeo and QuickTime. MPEG files previously on PCs needed hardware decoders (codecs) for MPEG processing. Today, however, PCs can use software-only codecs including products from Real Networks, QuickTime or Windows Media Player.

JPEG: JPEG compression is used in a number of image file formats. JPEG is the most common image format used by digital cameras and other photographic image capture devices; along with JPEG/JFIF, it is the most common format for storing and transmitting photographic images on the World Wide Web. These format variations are often not distinguished, and are simply called JPEG.

Q.2.(a) What is meant by data compression? Explain any two data compression techniques. (4)

Ans. Huffman Code: Huffman Code assigns shorter encodings to elements with a high frequency, F:e. It differs from block encoding in that it is able to assign codes of different bit lengths to different elements. Elements with the highest frequency, F:e, get assigned the shortest bit length code. The key to decompressing huffman code is a huffman tree.

A huffman tree is a special binary tree called a trie. (pronounced try) A binary trie is a binary tree in which a 0 represents a left branch and a 1 represents a right branch. The numbers on the nodes of the binary trie represent the total frequency, F, of the tree below. The leaves of the trie represent the elements, e, to be encoded. The elements are assigned the encoding which corresponds to their place in the binary trie. Below is an example.

NOTE: Spaces are included in the message and in the encodings for readability purposes only. The space is not considered as a character in the message and therefore is not encoded as an element in the message.

Message to be Encoded

dad ade fade bead ace dead cab bad fad cafe face

Block Encoding

011 000 011 000 011 100 101 000 011 101 001 100 000 011 000 010 100 011 100 000 011 010 000 001 001 000 011 101 000 011 010 000 101 100 101 000 010 100

Scanned by CamScanner

The block encoding above is a fixed length encoding. If a message contains i elements, block encoding requires **log i** bits to encode each element, e.

Spaces have been inserted between the strings of bits which represent each character in both the Block Encoding and the Huffman Encoding.

Huffman Encoding

01 10 01 10 01 111 110 10 01 111 000 111 10 01 10 001 111 01 111 10 01 001 10 000 000 10 01 110 10 01 001 10 110 111 110 10 001 111

element	frequency	block code	huffman code
a	11	000	10
b	3	001	000
c	4	010	001
d	9	011	01
e	7	100	111
f	4	101	110

Entropy: The average information content of a message is called its enropy. The information content is related to uncertainty. The less likely a message is to occur the larger its information content. This makes sense if we think of an example: if a person knows what message is about to be sent to him, how much new information has he learned by receiving that message? None. This is all that the above statement is saying.

Entropy, E, is information content. The entropy of a source is inversely proportional to its probability of occurrence:

$$E = -\log P$$

We use the the log function because we are converting all sources into the binary alphabet, B.

The same rule applies to an element, e, in a message, M. Its entropy can be defined as:

$$E{:}e = -\log (P{:}e)$$

P:e is the probability of an element in a message. It is equal to that element's frequency, F:e, divided by the frequency of the entire message, F:M:

$$P{:}e = F{:}e / F{:}M$$

The average information content or average entropy for a message, E:M, can now be defined. We know the entropy for each element in the message is E:e. We will index the elements, e, in a message, M, by assigning them integer positions, i, according to their place in the message. For example: the first element, e, in message M will be assigned 1, the second will be assigned 2...the i-th will be assigned i. Now we can define the entropy for an entire message, E:M, where there are i elements:

$$E{:}M = P{:}1*E{:}1)+(P{:}2*E{:}2)+(P{:}3*E{:}3)+...+(P{:}i*E{:}i)$$

Entropy is an important concept to data compression. The entropy of an element (E:e) is the minimum number of bits needed to encode that element. The entropy of an entire message (E:M) is the minimum number of bits needed to encode the entire message with a lossless compression. The entropy of a message can be used to determine if data compression is worth attempting. It can also be used to evaluate the effectiveness of a compression. The number of bits in a compressed code can be compared to the entropy for that message (E:M) revealing how close to optimal compression one's compressed code is.

Q.2.(*b*) Explain Z- Buffer method. Which is better and why – Z- Buffer or A-Buffer? (3)

Ans. z-buffering, also known as **depth buffering**, is the management of image depth coordinates in 3D graphics, usually done in hardware, sometimes in software. It is one solution to the visibility problem, which is the problem of deciding which elements of a rendered scene are visible, and which are hidden. The painter's algorithm is another common solution which, though less efficient, can also handle non-opaque scene elements.

When an object is rendered, the depth of a generated pixel (z coordinate) is stored in a buffer (the **z-buffer** or **depth buffer**). This buffer is usually arranged as a two-dimensional array (x-y) with one element for each screen pixel. If another object of the scene must be rendered in the same pixel, the method compares the two depths and overrides the current pixel if the object is closer to the observer. The chosen depth is then saved to the z-buffer, replacing the old one. In the end, the z-buffer will allow the method to correctly reproduce the usual depth perception: a close object hides a farther one. This is called **z-culling**.

Z buffer Vs A buffer: Z buffer and A buffer are two of the most popular visible surface detection techniques used in 3D computer graphics. Visible surface detection (also known as hidden surface elimination) is used to identify what is visible within a scene from a certain viewing point in the 3D world. There are two main categories of surface detection methods known as Object Space Methods and Image Space Methods. Object Space Methods deal with comparing object and/ or parts of objects to determine which surfaces are visible. Image Space Methods deal with deciding visibility on a point-to-point basis at the pixel level. Image Space Methods are the most popular and Z buffer and A buffer belong to that category. Z buffer method computes the surface depth values for each pixel throughout the whole scene. A buffer method is an extension to Z buffer method, which adds transparency.

Q.2.(*c*) Define animation. How animation does play an important role in multimedia applications? (3)

Ans. A simulation of movement created by displaying a series of pictures, or frames. Cartoons on television is one example of animation. Animation on computers is one of the chief ingredients of multimedia presentations. There are many software applications that enable you to create animations that you can display on a computer monitor.

Note the difference between animation and video. Whereas video takes continuous motion and breaks it up into discrete frames, animation starts with independent pictures and puts them together to form the illusion of continuous motion.

Animation is the rapid display of a sequence of images to create an illusion of movement.

Animation plays a huge role in entertainment (providing action and realism) and education (providing visualization and demonstration). Entertainment multimedia titles in general and children's titles specifically, rely heavily on animation. But animation can also be extremely effective in other titles, such as training applications. Animation adds visual impact to the multimedia project. Many multimedia applications for both the Macintosh and Windows provide animation tools, but you should first understand the principles of how the eye interprets the changes it sees as motion.

Scanned by CamScanner

Q.3.(a) What are multimedia authoring tools? Describe the basic characteristics of requirements for selecting a multimedia authority tool. (5)

Ans. The integration of audio, video, graphics and text on the desktop promises to fundamentally challenge the centuries-old model of the printed document as the basis for information exchange. Before this potential can be realized, however, systems must be devised that enable the production and presentation of complex, inter-related media objects. These systems are generically called multimedia authoring tools.

We consider the development of multimedia authoring tools, examine the current state of the art, and then discuss a set of research challenges that need to be addressed before the full potential of multimedia output technology can be effectively utilized to share information.

Types of Authoring Tools:

1. Card- or page-based tools.
2. Icon-based, event-driven tools.
3. Time-based tools.

Card-based or page-based tools:

1. The elements are organized as pages of a book or a stack of cards.
2. Card-or page-based authoring systems
3. best used when the bulk of your content consists of elements that can be viewed individually
4. the pages of a book or cards in a card file.
5. link these pages or cards into organized sequences.
6. jump, on command, to any page
7. play sound elements and launch animations and digital video.

Icon-or object-based, event-driven tools

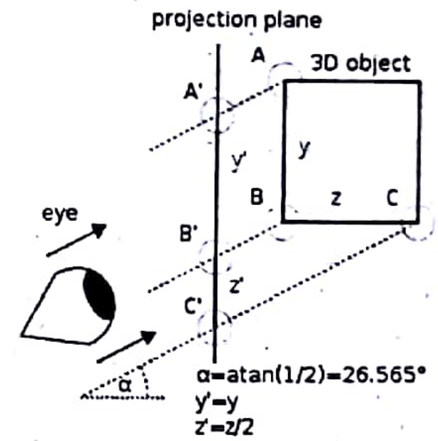
1. multimedia elements and interaction cues (events) are organized as objects in a structural framework or process.
2. simplify the organization of your project
3. display flow diagrams of activities along branching paths.
1. In complicated navigational structures, this charting is particularly useful during development.

Time-based tools

1. Elements and events are organized along a timeline with resolutions as high as or higher than 1/30 second.
2. Time-based tools
3. best to use when you have a message with a beginning and an end..
4. played back at a speed that you can set
5. Other elements (such as audio events) are triggered at a given time or location in the sequence of events.
6. Jumps to any location in a sequence
1. navigation and interactive control.

Q.3.(b) Find the general form of an oblique projection onto the XY plane. (5)

Ans. The term **cabinet projection** (sometimes **cabinet perspective**) stems from its use in illustrations by the furniture industry. Like cavalier perspective, one face of the projected object is parallel to the viewing plane, and the third axis is projected as going off in an angle (typical 30° or 45°). Unlike cavalier projection, where the third axis keeps its length, with cabinet projection the length of the receding lines is cut in half.



Mathematical formula: As formula, if the plane facing the viewer is , and the receding axis is , then a point is projected like this:

$$P\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x + 0.5 \cdot z \cdot \cos\alpha \\ y + 0.5 \cdot z \cdot \sin\alpha \\ 0 \end{pmatrix}$$

Where α is the mentioned angle.

The transformation matrix is:

$$P = \begin{bmatrix} 1 & 0 & 0.5 \cdot \cos\alpha \\ 0 & 1 & 0.5 \cdot \sin\alpha \\ 0 & 0 & 0 \end{bmatrix}$$

Q.4.(a) Derive the simple illumination model. Include the contribution of diffuse, ambient and Specular reflection? (5)

Ans. Illumination models model the interaction of light with the surface and range from simple to very complex. In Computer Graphics, we use physics to derive an illumination model and then fudge it to make the picture look good. More realistic and complex models adhere more closely to the underlying physics. In this section, we will look at only a simple illumination model and consider approximations for two types of light (ambient and point light sources) and two types of light reflection (diffuse and specular). Remember that we have to handle both specular and diffuse reflections and to solve the "pixel equation" as given below:

$$I_r(x, y, z) = \int_{t=-\infty}^{\infty} \int_{\lambda=400}^{700} \int_{\varphi=0}^{\pi/2} \int_{\theta=0}^{\pi} L(t, x, y, z, \varphi, \theta, \lambda)\, R(t, \varphi, \theta, \lambda)\, d\theta\, d\varphi\, d\lambda\, dt$$

$$t = -\infty \; \lambda = 400 \; \varphi = 0 \; \theta = 0$$

Scanned by CamScanner

Where:

x, y, z = the coordinates of the point on the surface

t = time

λ = wavelength

φ = azimuthal angle (from Z axis)

θ = angle about Z axis

A local illumination model must handle diffuse and specular reflections. Diffuse reflections are fairly simple and easily handled. Much research has gone into the specular reflection component. Below, two different models are discussed. The Phong model is not based in physics, but on empirical observation. Despite its lack of a theoretical grounding, it produces quite good results and has been the basis for most computer graphics imagery produced since it was developed in 1975. The Blinn model, developed in 1977, is based on research results from Physics and corrects some of the deficiencies of the Phong model. Both of these models assume that the specular highlights are the color of the light source and make no attempt to estimate the color contribution from the material itself. An even more advanced model by Cook and Torrance does estimate the color contribution from the material and is more asccurate for some substances, especially metals.

Q.4.(*b*) Using the origin as the center of projection, derive the perspective transformation onto the plane passing through the point R0(X0,Y0,Z0) and having normal vector $N = a\hat{i} + B\hat{j} + C\hat{k}$. **(5)**

Ans. $P(x, y, z)$ is projected onto $P'(x', y', z')$

$$x' = \alpha x,$$
$$y = \alpha y$$
$$z = \alpha z$$
$$n_1x' + n_2y' + n_3z' = d \text{ (where } d = n_1 \times 0 + n_2 y_0 + n_3 z_0)$$
$$\alpha = d/(n_1x + n_2y + n_3z)$$
$$d000$$

$$\text{Per N, R0} = \begin{bmatrix} 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 0 & d \end{bmatrix}$$

COMPUTER GRAPHICS AND MULTIMEDIA–[ETCS-211] THIRD SEMESTER (B.TECH) END TERM EXAMINATION–[2014]

Time : 3 hrs. **M.M. : 70**

Note: *Attempt any three questions in total including Question No. 1 which is compulsory.*

Q.1.(*a*) Differentiate between random scan and raster scan techniques. **(10 × 2.5 = 25)**

Ans. Refer to Q.1. (*d*) of 2014 First Term Examination.

Q.1.(*b*) Differentiate between Line clipping and Text Clipping.

Ans. In computer graphics, ***'line clipping'*** is the process of removing lines or portions of lines outside of an area of interest. Typically, any line or part thereof which is outside of the viewing area is removed.

There are two common algorithms for line clipping: Cohen–Sutherland and Liang–Barsky. Text clipping is for strings of text. Text can be clipped in case of overlapping or if it crosses window display area boundaries.

Q.1.(*c*) Distinguish between Spline curves from Bezier curves.

Ans. A Bézier curve is a parametric curve frequently used in computer graphics and related fields. Generalizations of Bézier curves to higher dimensions are called Bézier surfaces, of which the Bézier triangle is a special case.

In vector graphics, Bézier curves are used to model smooth curves that can be scaled indefinitely. "Paths", as they are commonly referred to in image manipulation programs, are combinations of linked Bézier curves. Paths are not bound by the limits of rasterized images and are intuitive to modify.

Bézier curves are also used in the time domain, particularly in animation and user interface design. For example a Bézier curve can be used to specify the velocity over time of an object such as an icon moving from A to B, rather than simply moving at a fixed number of pixels per step.

A **spline** is a numeric function that is piecewise-defined by polynomial functions, and which possesses a sufficiently high degree of smoothness at the places where the polynomial pieces connect (which are known as *knots*).

In interpolating problems, *spline interpolation* is often preferred to polynomial interpolation because it yields similar results to interpolating with higher degree polynomials while avoiding instability due to Runge's phenomenon. In computer graphics, parametric curves whose coordinates are given by splines are popular because of the simplicity of their construction, their ease and accuracy of evaluation, and their capacity to approximate complex shapes through curve fitting and interactive curve design.

The most commonly used splines are **cubic spline**, i.e., of order 3—in particular, cubic B-spline, which is equivalent to C2 continuous composite Bézier curves

Q.1.(*d*) Define Synchronization in multimedia system?

Ans. Synchronization in multimedia systems refers to temporal relationships between media objects in the multimedia systems. In future multimedia systems (based,

Scanned by CamScanner

e.g., on MPEG-4) synchronization may also refer to spatial and content relationships, as well as temporal.

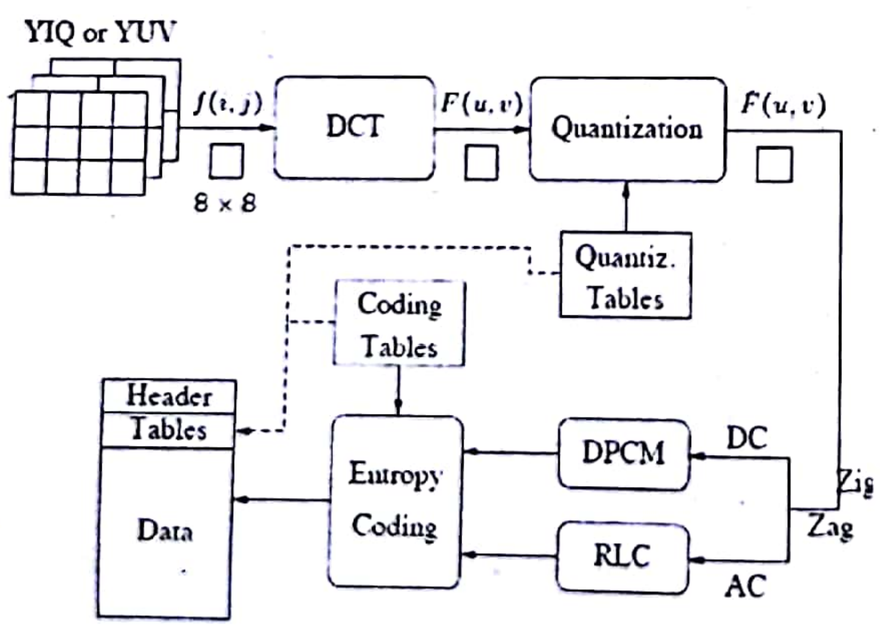
Synchronization between media objects comprises relationships between time-dependent media objects as well as time-independent media objects. Synchronization may need to occur at different levels in a multimedia system, consequently synchronization support is typically found in the operating system, communication system, databases, multimedia documents, and the application. A general scheme might involve a layered approach to achieving synchronization. For example, a Computer-Supported Collaborative Workgroup (CSCW) session might involved a multi-party video conferencing session with audio, and a shared whiteboard. Parties may make reference to objects on the shared whiteboard, using a pointer, to support what they are saying (e.g., saying "This area here..." while indicating the area with a pointer). Here, video and audio are continuous media objects which are highly periodic, whereas the shared whiteboard is a discrete media stream, as changes to it are highly irregular (the content, including the position of the pointer, depends on which participant has control of the object and when they make changes to it). The media streams must be highly synchronized, so that speech remains lip synchronized, and the whiteboard updates are synchronized with audio references to them.

The operating system and lower levels of the communication system are responsible for ensuring that jitter on individual streams does not occur during presentation of the video, audio, and whiteboard streams (intramedia synchronization). At a higher level, the runtime support for the synchronization of multiple multimedia media streams must ensure that the various media streams remain synchronized with respect to each other (intermedia synchronization). Finally, the application(s) are responsible for ensuring synchronicity between application-level events (usually initiated by the users).

Q.1.(*e*) Name image compression standards and explain in brief any one.

Ans. The JPEG Standard

- JPEG is an image compression standard that was developed by the "Joint Photographic Experts Group". JPEG was formally accepted as an international standard in 1992.



Q.1.(*f*) Give the homogeneous transformation matrices for basic 2D transformations.

Ans. For **Translation,** we have

$$[x' \ y' \ 1] = [x \ \ y \ \ 1] . \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$$

or

$$P' = P . T(T_x, T_y)$$

Similarly for **Rotation** transformation, we have

$$[x' \ y' \ 1] = [x \ \ y \ \ 1] . \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

or

$$P' = P . R(\theta)$$

Finally for **Scaling** transformation, we have

$$[x' \ y' \ 1] = [x \ \ y \ \ 1] . \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

or

$$P' = P.S(S_x, S_y)$$

Q.1.(*g*) What is the difference between image and video in terms of storage and processing?

Ans. Most computer images are stored in raster graphics formats or compressed variations, including GIF, JPEG, and PNG, which are popular on the World Wide Web.

Three-dimensional voxel raster graphics are employed in video games such as the Comanche series by Novalogic, and are also used in medical imaging such as MRI scanners

Raster images have a finite set of digital values, called *picture elements* or pixels. The digital image contains a fixed number of rows and columns of pixels. Pixels are the smallest individual element in an image, holding quantized values that represent the brightness of a given color at any specific point.

Typically, the pixels are stored in computer memory as a raster image or raster map, a two-dimensional array of small integers. These values are often transmitted or stored in a compressed form.

Raster images can be created by a variety of input devices and techniques, such as digital cameras, scanners, coordinate-measuring machines, seismographic profiling, airborne radar, and more. They can also be synthesized from arbitrary non-image data, such as mathematical functions or three-dimensional geometric models; the latter

Scanned by CamScanner

being a major sub-area of computer graphics. The field of digital image processing is the study of algorithms for their transformation.

Most users come into contact with raster images through digital cameras, which use any of several image file formats.

Some digital cameras give access to almost all the data captured by the camera, using a raw image format. *The Universal Photographic Imaging Guidelines (UPDIG)* suggests these formats be used when possible since raw files produce the best quality images. These file formats allow the photographer and the processing agent the greatest level of control and accuracy for output. Their use is inhibited by the prevalence of proprietary information (trade secrets) for some camera makers, but there have been initiatives such as OpenRAW to influence manufacturers to release these records publicly. An alternative may be Digital Negative (DNG), a proprietary Adobe product described as "the public, archival format for digital camera raw data". Although this format is not yet universally accepted, support for the product is growing, and increasingly professional archivists and conservationists, working for respectable organizations, variously suggest or recommend DNG for archival purposes

- The most common type of RAM is called dynamic RAM, or DRAM. It is nearly identical to the RAM process, except that before each capacitor is read, its stored value must be validated. An electrical charge is sent to the cell to make sure its charge is correct. Video RAM, or VRAM, is no different, except that it handles the graphical data such as images and videos.

- Video files are made up of strings of images, usually between 30 and 60 images per second. When each image is accessed, the processor reads it and sends it to the VRAM chip. The chip then sends it to the RAM digital-to-analog converter (RAMDAC) to create analog signals that the display uses. Unlike the normal RAM chip, VRAMs are dual-ported, allowing it to simultaneously send the image to the display and write the next one in line to the chip.

- There are several different varieties of video RAM units. One is SGRAM, or Synchronous Graphics RAM, which allows data to be modified in one big chunk instead of a long, inefficient sequence. Another is Multibank Dynamic RAM (MDRAM) is a high-performance chip produced by the company MoSys that splits the memory into 32-kilobyte blocks for more efficient data transfer. MDRAM can also be built with the specific amount of memory needed for different resolutions, which reduces manufacturing cost.

Q.1.(*h*) What is ant aliasing? How it is achieved?

Ans. ***Anti-aliasing*** is a software technique for diminishing ***jaggies*** - stairstep-like lines that should be smooth. Jaggies occur because the output device, the monitor or printer, doesn't have a high enough resolution to represent a smooth line. Antialiasing reduces the prominence of jaggies by surrounding the stairsteps with intermediate shades of gray (for gray-scaling devices) or color (for color devices). Although this reduces the jagged appearance of the lines, it also makes them fuzzier.

Pixel phasing is an antialiasing technique, stair steps are smoothed out by moving the electron beam to more nearly approximate positions specified by the object geometry.

Dithering is the attempt by a computer program to approximate a color from a mixture of other colors when the required color is not available.

Q.1.(*i*) What do you understand by synchronization at content level?

Ans. Refer to Q.1(*d*) of End Term Examination 2014.

Q.1.(*j*) What are the main functions of a multimedia development system?

Ans. Multimedia development system must perform main three functions as follows:

- Input data
- Development
- Data output

Data input from sources such as cameras or musical instruments, aplication development, and data output to some delivery medium such as a videodisk or CD-ROM.

Q.2. Explain DDA Line Drawing Algorithm and its drawbacks with Example. (12.5)

Ans.

```
DDA( x1, y1, x2, y2: integer);
var
    dx, dy, steps: integer;
    x_inc, y_inc, x, y: real;
begin
    dx := x2 - x1; dy := y2 - y1;
    if abs(dx) > abs(dy) then
    steps := abs(dx); {steps is larger of dx, dy}
else
    steps := abs(dy);
    x_inc := dx/steps; y_inc := dy/steps;
    {either x_inc or y_inc = 1.0, the other is the slope}
    x:=x1; y:=y1;
    set_pixel(round(x), round(y));
    for i := 1 to steps do
begin
    x := x + x_inc;
    y := y + y_inc;
    set_pixel(round(x), round(y));
end;
end; {DDA}
```

Draw Backs

- More Calculations
- Jaggies in the line
- No appropriate basis for innement in lines.

Scanned by CamScanner

Q.3. Describe bresenhams' line drawing procedures for lines with slopes >1. Use the procedure to scan with points at (2,5), (5,10). **(12.5)**

Ans.

$$x_1 = 2, y_1 = 5$$
$$x_2 = 5, y_2 = 10$$
$$\Delta x = 5 - 2 = 3$$
$$\Delta y = 10 - 5 = 5$$

Δy is > Δx. So exchange Δx and Δy now $\Delta x = 5$

$$\Delta y = 3$$

Plot (2.5)

$$e = 2 \times 3 - 5 = 1$$
$$e > 0 \text{ so}$$
$$x = x + 1 = 2 + 1 = 3$$
$$y = 5$$

and $e = 1 - 2 \times 5 = -9$

$$e < 0$$

So $y = 5 + 1 = 6\ (3, 6)$

$$e = e + 2 \times \Delta y = 9 + 6 = -3$$

So paid: (3, 7)

$$e = e + 2\ (3) = -3 + 6 = 3$$

plot (4, 7)

similarly Calculate the next point as (5, 8).

1. Algo: Read the endpoints (x1, y1) and (x2, y2) such that they are not equal.
2. Delta x = (x2-x1)

   Delta y=(y2-y1)
3. Initialize starting point x = x1

   Y = y1
4. S1=sign (x2-x1)

   S2=sign(y2-y1)

   [whether sign function returns-1, 0, 1 depending on its agreement < 0, = 0, > 0 respectively]
5. If delta y > delta x then

   Exchange delta x and delta y

   Exchange = 1

   Else exchange = 0
6. e = 2* delta y-delta x
7. i = 1
8. plot (x, y)

9. While (e>=0) {if (exchange = 1 then

   Y = y + s1

   Else

   Y = y+s2

   E=e-2* delta x

   }
10. If exchange = 1 then

    Y = y+s2

    Else

    X=x+s1

    E+2* delta y
11. l = i + 1
12. If (i<=delta x) then go to step (8)
13. Stop

Q.4. Explain Cohen – Sutherland line clipping algorithm. Illustrate with an example. **(12.5)**

Ans. The algorithm includes, excludes or partially includes the line based on where:

• Both endpoints are in the viewport region (bitwise OR of endpoints == 0): trivial accept.

• Both endpoints share at least one non-visible region which implies that the line does not cross the visible region. (bitwise AND of endpoints ! = 0): trivial reject.

• Both endpoints are in different regions: In case of this nontrivial situation the algorithm finds one of the two points that is outside the viewport region (there will be at least one point outside). The intersection of the outpoint and extended viewport border is then calculated (i.e. with the parametric equation for the line) and this new point replaces the outpoint. The algorithm repeats until a trivial accept or reject occurs.

The numbers in the figure below are called outcodes. An outcode is computed for each of the two points in the line. The outcode will have four bits for two-dimensional clipping, or six bits in the three-dimensional case. The first bit is set to 1 if the point is above the viewport. The bits in the 2D outcode represent: Top, Bottom, Right, Left. For example the outcode 1010 represents a point that is top-right of the viewport. Note that the outcodes for endpoints **must** be recalculated on each iteration after the clipping occurs.

$$\begin{bmatrix} 1001 & 1000 & 1010 \\ 0001 & 0000 & 0010 \\ 0101 & 0100 & 0110 \end{bmatrix}$$

The Cohen–Sutherland algorithm can be used only on a rectangular clipping area. For other convex polygon clipping windows (**Ref.** ans 2(*b*) of First term Examination 2014)

Scanned by CamScanner

Q.5. Prove that uniform scaling rotation form a commutative pair of operations. **(12.5)**

Ans. The matrix representation for a two-dimensional transformation sequence consisting of scaling followed by rotation, relative to the coordinate origin, is

$$\begin{bmatrix} s_x \sin\theta & -s_y \sin\theta & 0 \\ s_x \sin\theta & s_y \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

And the matrix representation for a rotation followed by scaling, relative to the coordinate origin, is

$$\begin{bmatrix} s_x \cos\theta & -s_y \sin\theta & 0 \\ s_x \sin\theta & s_y \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

These two matrices are equivalent only when $sx = sy$.

Q.6. Explain the various components of multimedia in audio, video and images and how multimedia data can be stored in database in respect of the audio, video and image files. **(12.5)**

Ans. Text: Text often shapes the content of a multimedia title. On the other hand, large amounts of text data can be a sign of a "converted book", not a true multimedia design. Technically speaking a text medium is easy to handle and store. As a rule text passages should be brief and the font should be easy to read on the computer screen.

Common file formats for text data are .txt, .rtf, and .doc.

Graphics: Graphics shape the design of a multimedia title, adding visual appeal and expressing concepts that text alone cannot easily communicate. Graphics is a more complex medium than text. They are more difficult to create, and you have to deal with a number of colors and screen resolutions. In addition, graphics files tend to be quite large. A single color picture for the computer screen can easily fill up several standard floppy disks. However, applications from the entertainment side of the business have established high quality graphics and photo realistic images as basic elements of all multimedia applications.

Common file formats for graphics are .bmp, .tif, .gif, and .jpg.

Sound: Sound can refer to speech, natural or synthesized sound effects, or music. You can use spoken words to complement written text, and music and sound effects to create a mood, add emphasis or realism, or signal interactivity. Technically, dealing with sound is comparable to dealing with graphics. With recorded (digitized) sound you have to deal with sound quality and storage capacity. Music can mean "recorded songs", or the computer can imitate instruments and produce synthesized music.

Common file formats for digitized sound are .wav, .aif, .snd, and .mid.

Video: Digitized video offers a level of authenticity similar to television or movies - with some performance limitations. With video you can show tasks and events that

words and graphics are inadequate to explain. Producing high quality video clips is a difficult and time consuming job. The current PC performance for video playback is limited by certain technical factors, so if you create a multimedia application that includes video clips you must have a clear idea of the performance of the typical user's PC. Usually you must compromise between the video screen size, video resolution, number of colors used and length of the clip.

Common digital video file formats are .mov, .avi, and .mpg.

Animation: Animation involves graphics that move in some way With animation you can clarify things or create special effects. Digitized video has replaced animation for certain things, but in many cases animations are easier to produce and "lighter" for the PCs performance. Technically speaking, animation can mean either external "video like" media files, or they can be closely integrated (programmed) into the application itself.

Common animation file formats include .flc, .fli, and .swf. Digital video and .gif file formats are also used for animation content.

Control: One final component of multimedia is the amount of control afforded to the user. At a minimum the user should be able to start and stop the program. In more advanced programs there may be a multitude of navigation options available to take users through the program.

The various multimedia elements used in multimedia can be produced by different application programs, such as

- Graphics/Drawing programs
- Animation programs
- Word Processing
- Image capture - digital cameras, scanners
- Digital Video editing
- Sound recording and editing programs.

Because of the use of all of these different types of elements, developing a multimedia program generally requires a multitude of skills:

- Instructional design skills
- Graphic design skills
- Audio production skills
- Video production skills
- Writing skills
- Programming skills

Although the process involves many skills, small-scale multimedia products can be created by individuals through the use of an authoring program.

Q.7. Describe the concepts, frameworks, issues and techniques in multimedia authoring? **(12.5)**

Ans. Multimedia Authoring Systems. Design Issues for Multimedia Authoring. Design Approaches to Authoring. Types of Multimedia Authoring Systems. Multisource

Scanned by CamScanner

Multi-User Authoring Systems. Telephone Authoring Systems. Hypermedia Application Design Considerations. Integration of Applications. Common UI and Application Integration. Data Exchange. Distributed Data Access. Hypermedia Application Design. Structuring the Information. Attaching Sounds and Video Clips to Objects. User Interface Design. Navigation Through the Application. Designing User Interfaces. Special Metaphors for Multimedia Applications. The Organizer Metaphor. The Telephone Metaphor. Aural User Interface. The VCR Metaphor. Audio/Video Indexing Functions. Information Access. Object Display/Playback Issues. Image Display Issues. Audio Quality. Special Features for Video Playback. Video Frame Interleaving. Scene Change Frame Detection. Video Scaling, Panning, and Zooming. Three-Dimensional Object Display and VR.

Multimedia authoring is the process of designing and developing a multimedia product. The author brings all of the different multimedia components together into one program and provides structure, navigation, and interactivity for the user. For educational multimedia it is also important to create an effective instructional environment.

The authoring process can be used to develop many different types of products, including:

- Educational applications
- Computer-based training
- Reference systems
- Simulations
- Virtual environments
- Entertainment programs and games
- Interactive information kiosks
- Retail CD-ROMs
- Training systems

Multimedia applications can be coded directly in a programming language (like C++ or Pascal). These languages are easily understood by a computer's processor and allow programmers to communicate more directly with a computer. They can then make the computer perform complex tasks with a minimum of fuss. The problem is that the programming languages are very complex themselves, and require the programmer to become familiar with an entire new language of strange words and actions.

To simplify the process for those of you who are not fluent in writing programming code, there are applications that will mediate between the designer and the complex programming code. These intervening applications are called authoring programs. An authoring program brings all of the various multimedia elements together into one program, and additionally allows the developer to add control and navigation. In other words, the designer talks to the authoring software, and the authoring software talks to the computer using a language the computer can understand. Authoring programs can be used to create full multimedia productions, and most of these programs have point-and-click features to simplify the development process.

To demonstrate this concept, let's consider a simple authoring program. One simple type of authoring program is a web page editor. These types of programs allow you to

combine text, pictures, sounds, and videos all on one page that can then be displayed on the World Wide Web for others to view. In the early days of the Web all web pages had to be created using a programming language called HTML (HyperText Markup Language). If you wanted to create a web page you had to be familiar with this language. With the advent of web editors such as Netscape Composer, Dreamweaver, and FrontPage almost anyone can now create web pages using an extremely simplified interface (often called a WYSIWYG interface). Pages created in these programs are still composed of HTML, it's just that the web editing program generates the HTML for you and keeps it hidden behind the user interface. When a web browser attempts to display a page it still looks for the HTML code - that is the only way it knows what to do.

Popular authoring programs include ToolBook, Visual Basic, Director, Authorware, and Hyperstudio. Even PowerPoint can be considered a simple authoring program, although it is more commonly referred to as a presentation program. Keep in mind that there is no dividing line between where a presentation program ends and where a multimedia authoring tool starts.

These programs use different metaphors to represent the methodologies by which the programs accomplishes their tasks. For example, PowerPoint and Hyperstudio use a "Card" or "Frame" metaphor in which content is displayed in different frames or screens. Navigation is provided to allow users to jump from one screen to another.

Director uses a "Timeline" metaphor. It displays something like a music score, with the various multimedia elements shown in various horizontal "tracks" with simultaneity shown via the vertical columns. When the program is run it moves through the timeline according to how the author has set things up. This interface allows for the best media synchronization and sequencing, as well as producing excellent sprite and cel animation.

Web applications use a "Tag" metaphor. This involves the use of tags in text files (e.g., HTML) to link pages, provide interactivity, and integrate multimedia elements. This method comes closest to traditional computer programming. Of course, using a web editor allows you to avoid interacting with the language for most tasks.

It is important to understand these different metaphors so that when you go from using one program to another you don't get too overwhelmed by the differences. If you have used PowerPoint and then jump into Director you will need to readjust your thinking as to how a presentation is created.

Q.8. How is multimedia synchronization achieved? Discuss with example the various techniques used. (12.5)

Ans. Synchronization in multimedia systems

- Television systems
  - Sound and video are multiplexed
  - Delay affects both sound and video
  - Synchronization always maintained
- Focus on time-independent media
  - LDU to be put in relationship are independent media objects

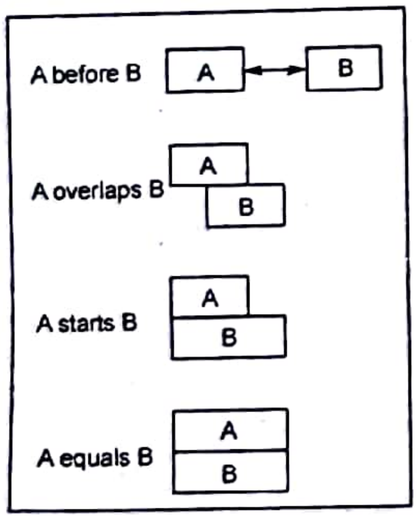
Scanned by CamScanner

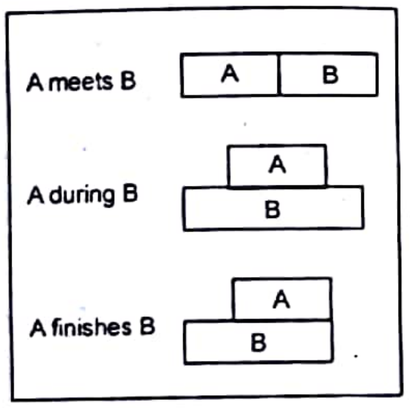
- Quality of Service
  - Hard guarantees
  - Soft guarantees

Multi-step Synchronization

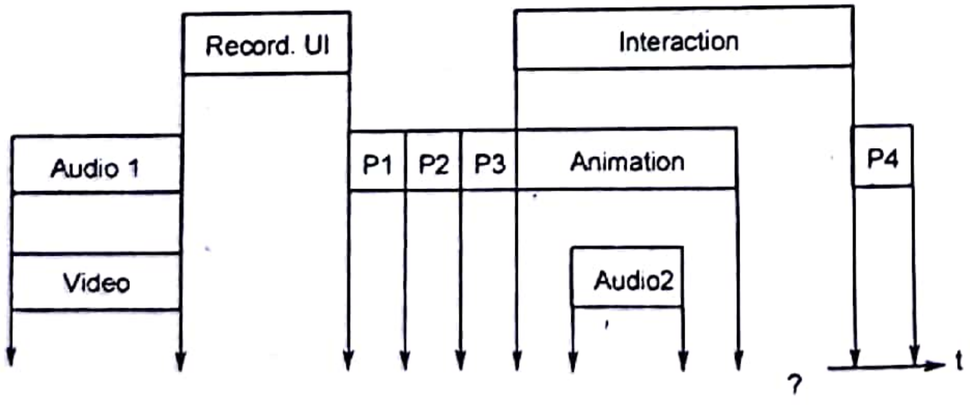
- Synchronization
- during object acquirement
- of retrieval
- during delivery of the LDUs to the network
- during the transport
- at the sink
- within the output device

Interval-based Specification

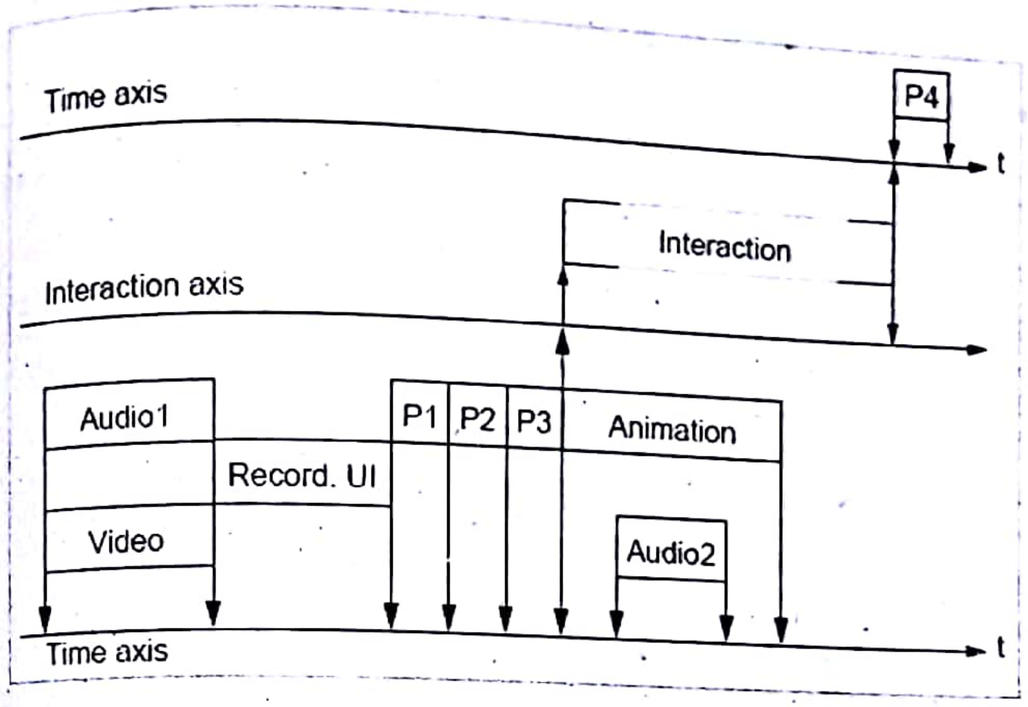




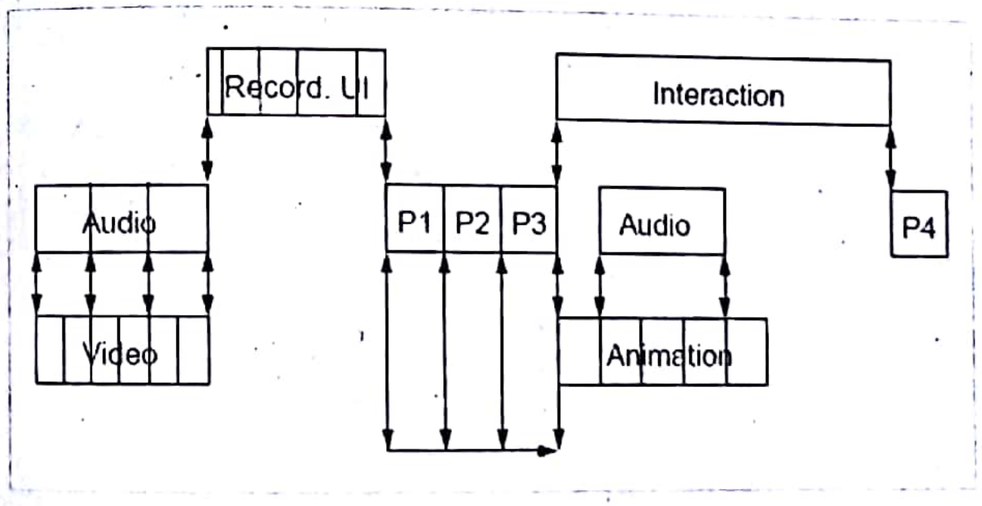
Synchronization Based on a Global Timer



Synchronization Based on Virtual Axes



Synchronization Based on Reference Points



Q.9.(*a*) Explain in detail how MPEG is used to compress both audio and video. Give their respective data streams. **(12.5)**

Ans. The purpose of this report is to inform the audience about video and audio compression standards developed by the Moving Picture Expert Group (MPEG). This report includes general information about different types of MPEG along with its history and development. The following MPEG standards exist:

- MPEG-1, a standard for storage and retrieval of moving pictures and audio on storage media.

Scanned by CamScanner

- MPEG-2, a standard for digital television.
- MPEG-4, a standard for multimedia applications.
- MPEG-7, a content representation standard for information search

ABOUT MPEG

The acronym MPEG stands for Moving Picture Coding Experts Group, and was established in January 1988 with the mandate to develop standards for coded representation of moving pictures, audio and their combination. There were only 25 experts working together when the first MPEG meeting took place. Since then, MPEG has grown to a large committee consisting of 350 experts from 200 companies from about 20 countries. MPEG operates under ISO, the International Organization for Standardization and IEC, the International Electro-technical Commission. MPEG itself is a nickname; the official name is: ISO/IEC JTC1 SC29 WG11. The term also refers to the family of digital video compression standards and file formats developed by the group. MPEG generally produces better-quality video than competing formats, such as Video for Windows, Indeo and QuickTime. MPEG files can be decoded by special hardware or by software.

Presently MPEG audio/video compression is widely used in everyday life in many applications. For example, it is used in DVD players, digital television set-top boxes, HDTV recorders, Internet video, video conferencing, and many others.

MPEG achieves high compression rate by storing only the changes from one frame to another, instead of each entire frame. The video information is then encoded using a technique called DCT, Discrete Cosine Transform. MPEG uses a type of "lossy" compression, since some data is removed. But the diminishment of data is generally imperceptible to the human eye.

As of today, there are three MPEG standards: MPEG-1, MPEG-2, and MPEG-4. MPEG-1 and MPEG-2 provide interoperable ways of representing audiovisual content, used on the air and on digital media. In other words, these standards made interactive video on CD-ROM, DVD and Digital Television possible. MPEG-1 is used for the storage and retrieval of moving pictures and audio on storage media. MPEG-2 is used for digital television; it's the timely response for the satellite broadcasting and cable television industries in their transition from analog to digital formats. MPEG-3 was intended as an extension of MPEG-2 to cater for HDTV but was eventually merged into MPEG-2: not to be confused with MP3 - MPEG-1 layer three: compression of audio files. MPEG-4 is the multimedia standard which considers the growing-together of the three areas TV, computer and communication. MPEG-4 codes content as objects and enables those objects to be manipulated individually or collectively on audio-visual scene.

MPEG 1

In 1991, the Moving Picture Expert Group finalized its first standard for video and audio compression: MPEG-1. This generic standard is independent of a particular application; and thus is better described as mainly a toolbox. Since its' creation, it was noticeable that MPEG-1's quality was superior to other digital video format. This new video and audio encoding standard offered high compression techniques while maintaining moderate quality picture and audio on such popular formats as VideoCD's

(CD-ROM) and MP3's. More importantly, this widely accepted standard became an international standard, which can be adopted universally by different industries (disctronics.co.uk). In 1992, MPEG-1 was officially published by the ISO MPEG committee, and has ever since spawned further advancements in video and audio compressions (as can be seen in MPEG-2). In short, MPEG-1 is a standard in 5 parts:

- Part 1: specifies the combination of video, audio, and data streams into a single data stream
- Part 2: defines video compression
- Part 3: defines audio compression
- Part 4: deals with compliance testing
- Part 5: technically describes/documents Part 1, 2, and 3

Part one of the MPEG-1 standard deals with the problem of combining data streams of video and audio parts to form a single stream. This is an important function that allows a number of different types of inputs to combine into one compressed stream that is well suited for digital storage or transmission (Leonardo Chiriglione). Time stamps are used as the basic coding principle of a MPEG System. This system decodes and displays the times of audio, video, and the time of reception of the multiplexed coded data at the decoder, all in terms of 90 kHz system clock. This method benefits a lot in terms of flexibility in areas such as decoder design, packet lengths, video picture rates, audio sample rates, network performance, etc. (Berkeley.edu).

Part 2 of MPEG-1 is known as MPEG-1 Video. This part presents various techniques that can be used to compress video. It mainly takes advantage of the vast number of redundancies found in series of video frames and the human eye's inability to recognize subtle/discrete changes in between these redundancies. Specifically, part 1 states how to use techniques such as motion compensations and temporal predictions to remove redundant picture frames from a group of pictures, and thereby, reducing the overall size of the video. (A more in depth description of these techniques will be discussed in MPEG-2 Video Compression, since MPEG-2 is an extension of such techniques.)

MPEG-1 Audio is also an important part of the MPEG-1 standard. Here, the same video compression does not apply to audio. The process starts out by placing audio samples into two independent blocks of the encoder. The "mapping block" of the encoder filters and creates 32 equal-width frequency sub-bands, or samples, while the "psychoacoustic block" determines a masking threshold for the audio inputs. By determining such threshold, the psychoacoustic block can output information about noise that is imperceptible to the human ear. The "quantizer and coding" block then uses the sample and masking threshold information to create a set of coding symbols that the "frame packing" block needs. At the end, the "frame packing" block assembles the actual bitstream from the output data of the other blocks, and adds other header information as necessary before sending it out into that single bitstream. (Refer to the diagram)

The MPEG committee also created three compression methods and named them as following: Audio Layer I, II, and III. Layer one is the simplest. It is a 'sub-band coder' with a psychoacoustic model that relies on little processing power to produce what is

Scanned by CamScanner

now obsolete and poor audio quality. Layer II has more advanced bit allocation techniques, and provides greater accuracy, but in today's standard, is just "okay" standard sound. Layer III, on the other hand uses a massive amount of processing power, but is able to produce very high quality CD sound. Layer III is also known as Mp3's, today's most widely used audion compression technique. Important applications include of such standards include: Consumer Recording (DCC), Disc based storage (CD-i, CD-Video), DVD, Disc based Editing, audio broadcasting station automation (Lynezoo).

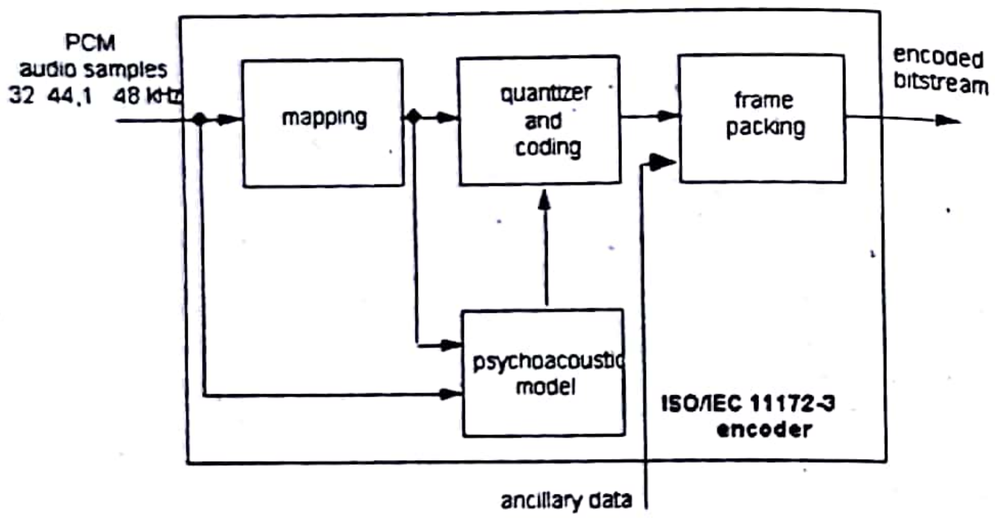


Fig. 1: Example of an Encoder

Part 4 of MPEG-1 verifies whether bitstreams and decoders meet the requirement that was established as essential in parts 1, 2, and 3 of the MPEG-1 standard. Specifically, these tests verify various characteristics of a bit stream, and can be used to make sure that an encoder produces valid bitstreams. In turn, manufacturers of encoders & their customers, or nearly any individual with the appropriate knowledge, can check and verify these parts.

Finally, part 5, while technically not a standard, is a reporting of the full software implementation of the first three parts of the MPEG-1 standard (Chiariglione).

MPEG 2

The Moving Pictures Experts Group-2 (MPEG-2) is an industry standard of delivering digital video on a network. MPEG-2 substantially reduces the bandwidth required to transmit a high-quality digital video signal, and it optimizes the trade-offs between resolution and the required transmission bandwidth. All industries considering digital video service distribution have to make MPEG-2 part of the planning process; it is crucial in digital head-end, broadband distribution, network access equipment, and the associated architectures and operations.

The picture quality through an MPEG-2 codec depends on the complexity and predictability of the source pictures. Real-time coders and decoders have demonstrated generally good quality standard-definition pictures at bit rates around 6 Mbits per second. As MPEG-2 coding increases, the same picture quality may be achievable at lower bit rates.

Q.9.(b) How MIDI devices is used to transmit the information between the components? Explain.

Ans. Serial parallel, and joystick port MIDI

As computer-based studio setups became common, MIDI devices that could connect directly to a computer became available. These typically used the 8-pin mini-DIN connector that was used by Apple for serial and printer ports prior to the introduction of the Blue & White G3 models. MIDI interfaces intended for use as the centerpiece of a studio, such as the Mark of the Unicorn MIDI Time Piece, were made possible by a "fast" transmission mode that could take advantage of these serial ports' ability to operate at 20 times the standard MIDI speed. Mini-DIN ports were built into some late-1990s MIDI instruments, and enabled such devices to be connected directly to a computer. Some devices connected via PCs' DB-25 parallel port, or through the joystick port found in many PC sound cards.

mLAN

Yamaha introduced the mLAN protocol in 1999. It was conceived as a Local Area Network for musical instruments using FireWire as the transport, and was designed to carry multiple MIDI channels together with multichannel digital audio, data file transfers, and time code. mLan was used in a number of Yamaha products, notably digital mixing consoles and the Motif synthesizer, and in third-party products such as the PreSonus FIREstation and the Korg Triton Studio. No new mLan products have been released since 2007.

Ethernet

The computer network implementation of MIDI provides network routing capabilities, and provides the high-bandwidth channel that earlier alternatives to MIDI, such as ZIPI, were intended to bring. Proprietary implementations have existed since the 1980s, some of which use fiber optic cables for transmission.The Internet Engineering Task Force's RTP MIDI open specification is gaining industry support, as proprietary MIDI/IP protocols require expensive licensing fees, or provide no advantage, apart from speed, over the original MIDI protocol. Apple has supported this protocol from Mac OS X 10.4 onwards, and a Windows driver based on Apple's implementation exists for Windows XP and newer versions.

OSC

The Open Sound Control (OSC) protocol was developed at the Center for New Music and Audio Technologies (CNMAT) of the University of California, Berkeley, and is supported by software programs such as Reaktor, Max/MSP and Csound, and some newer hardware controllers, including the Lemur Input Device.OSC can be transported over Ethernet connections at broadband speeds, but is poorly suited for use as a whole-studio solution, as to date it lacks widespread support from hardware and mainstream software. The increased size of OSC messages as compared to MIDI messages make OSC an impractical solution for many mobile devices, and its touted speed advantages over MIDI disappear when both are transmitted over equal media.OSC is not proprietary, but is not maintained by a standards organization.

Scanned by CamScanner

Wireless MIDI

Systems for wireless MIDI transmission have been available since the 1980s. Several commercially available transmitters allow wireless transmission of MIDI and OSC signals over Wi-Fi and Bluetooth. iOS devices are able to function as MIDI control surfaces, using Wi-Fi and OSC. An XBee radio can be used to build a wireless MIDI transceiver as a do-it-yourself project. Android devices are able to function as full MIDI control surfaces using several different protocols over Wi-Fi and Bluetooth.

FIRST TERM EXAMINATION [SEPT. 2015]
THIRD SEMESTER [B. TECH]
COMPUTER GRAPHICS AND MULTIMEDIA
[ETCS–211]

Time: 1.30 Hrs. **MM : 30**

Note: *Q.No.1 which is compulsory and any two questions from the rest.*

Q.1. (a) What is meant by scan conversion? What are the four side effects of scan conversion with raster scan devices. **(2.5)**

Ans. Scan conversion or scan rate converting is a video processing technique for changing the vertical/horizontal scan frequency of video signal for different purposes and applications.

Problem: Approximation of lines by a finite number of pixels

- Costly rounding of operations
- Unnecessary floating point arithmetic
- Error accumulation.

Q.1. (b) To draw a circle, it is sufficient to know the co-ordinates of an octant. Is this statement true, justify?
(2.5)

Ans. Yes, to draw a circle it is sufficient to know the co-ordinates of an octant symmetry as joining all the points of octant symmetry forms various arcs and on joining them a circle can be formed.

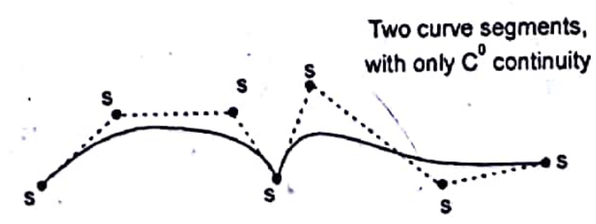
Q.1. (c) What are the various types of parametric continuity conditions and why they are applied in curve drawing? **(2.5)**

Ans. Parametric continuity is a concept applied to parametric curves describing the smoothness of the parameter's value with distance along the curve.

Definition: A curve can be said to have C^n continuity if $\frac{d^n s}{dt^n}$ is continuous of value throughout the curve.

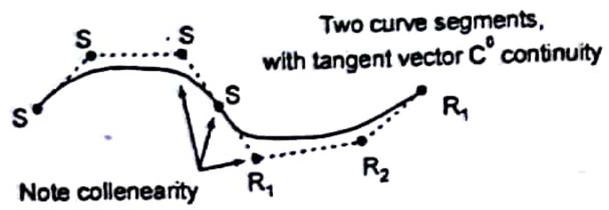
As an example of a practical application of this concept, a curve describing the motion of an object with a parameter of time, must have C^1 continuity for the object to have finite acceleration. For smoother motion, such as that of a camera's path while making a film, higher levels of parametric continuity are required.

Order of continuity:



Two Bézier curve segments attached that is only C^0 continuous.

Scanned by CamScanner



Two Bézier curve segments attached in such a way that they are C^1continuous.

The various order of parametric continuity can be described as follows:

- C^1: curves include discontinuities
- C^0: curves are joined
- C^1: first derivatives are continuous
- C^2: first and second derivatives are continuous
- C^n: first through n^{th} derivatives are continuous

The term *parametric continuity* was introduced to distinguish it from *geometric continuity* (G^n) which removes restrictions on the speed with which the parameter traces out the curve.

Q. 1. (*d*) Derive the rotation matrix for rotating an object through an angle of 45 degree about a line with end points A(0,0,0) and B(3,4,5). **(2.5)**

Ans. Rotation about an arbitrary axis in space.

- Substituting for the respective angles we obtain the following matrices:

$$[T] = [Tr][R_\alpha][R_\beta][R][R_\beta]^{-1}[R_\alpha]^{-1}[Tr]^{-1}.$$

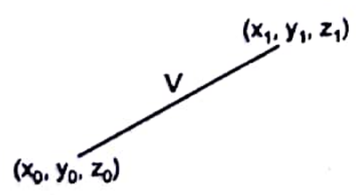
$$[R_\alpha] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_z/d & c_y/d & 0 \\ 0 & -c_y/d & c_z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R_\beta] = \begin{bmatrix} \cos(-\beta) & 0 & -\sin(-\beta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(-\beta) & 0 & \cos(-\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} d & 0 & c_x & 0 \\ 0 & 1 & 0 & 0 \\ -c_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[R] = \begin{bmatrix} \cos\delta & \sin\delta & 0 & 0 \\ -\sin\delta & \cos\delta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about an arbitrary axis in space.

- Direction cosines of a an arbitrary axis when two points on the line are known:



$$[V] = [(x_1 - x_0)(y_1 - y_0)(z_1 - z_0)]$$

$$[C_x C_y C_z] = \left[\frac{(x_1 - x_0)(y_1 - y_0)(z_1 - z_0)}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}}\right]$$

Here alpha is 45 degree and x_0, y_0, z_0 and x_1, y_1, z_1 are A(0, 0, 0) and B(3, 4, 5)

Q.2. (*a*) Scan convert a line using Bresenham's Line drawing Algorithm from A(1, 1) to B(8, 5). **(5)**

Ans.

$$\Delta x = 8 - 1 = 7$$
$$\Delta y = 5 - 1 = 4$$
$$2\Delta x = 14$$
$$2\Delta y = 18$$
$$P_0 = 2\Delta y - \Delta x = 8 - 7 = 1$$
$$P_k > 0 \text{ so next point is } (2,2)$$
$$P_{x+1} = P_k + 2\Delta y - 2\Delta x$$
$$P_1 = 1 + 2(4) - 2(7) = -5$$

next point 4 (3,2)

and

$$P_2 = -5 + 8 = 4$$

next point is (4,3)

$$P_3 = 4 + 8 - 14 = -2$$

next point is (5,3)

$$P_4 = -2 + 8 = 6$$

next point is (6,4)

$$P_5 = 6 + 8 - 14 = 0$$

next point is = (7,5)

$$P_6 = 0 + 8 - 14 = -6$$

So next point is (8,5) stop.

Point are (1,1), (2,2), (3,2), (4,3), (5,3), (6,4), (7,5) and (8,5)

Q.2. (*b*) Clip a line using Cohen-Sutherland algorithm where the window co-ordinates are (–3, 1) and (2, 6) and the line co-ordinates are A(–1, 5) and B(3, 8) respectively. **(5)**

Ans. Line segment AB points lies in (0000) AND (1010) = 0000.

The outcode of B is 1010 *i.e.* it is outside the window. We push the first 1 to 0 by clipping against the line $Y_{max} = 6$.

Here $m = y_2 - y_1 / x_2 - x_1 = 8 - 5/3 + 1 = 3/4$

And $y = 6$

$¾ = 6 - 5/x + 1 = x = 1/3$

So, the resulting intersection point is (1/3, 6) and it's outcode is 0000.

Q.3. (*a*) Reflect the polygon whose vertices are A(–1,0), B(0, –2), C(1, 0) and D(0, 2) about a horizontal line y = 2. **(6)**

Ans. We can represent the given polygon by the homogeneous coordinate matrix as

$$V = [ABCD] = \begin{matrix} -1 & 0 & 1 \\ 0 & -2 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{matrix}$$

Scanned by CamScanner

(*a*) The horizontal line $y = 2$ has an intercept (0, 2) on y axis and makes an angle of 0° degree with the x axis. So $m = 0$ and $c = 2$. Thus, the reflection matrix.

$$ML = T\text{–}v.\ R\text{–}\theta.\ Mx.\ R\theta.\ T\text{–}v,$$

where

$$v = 0I + 2J$$

$$\begin{matrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 4 & 1 \end{matrix}$$

So the new coordinates A'B'C'D' of the reflected polygon ABCD can be found as:

$$[A'B'C'D'] = [ABCD].\ ML$$

$$\begin{matrix} -1 & 0 & 1 \\ 1 & 0 & 0 \\ -1 & 4 & 1 \end{matrix}$$

$$= \begin{matrix} 0 & -2 & 1 \end{matrix} . \begin{matrix} 0 & -1 & 0 \end{matrix} = \begin{matrix} 0 & 6 & 1 \end{matrix}$$

$$\begin{matrix} 1 & 0 & 1 \\ 0 & 4 & 1 \\ 1 & 4 & 1 \end{matrix}$$

$$\begin{matrix} 0 & 2 & 1 \\ 0 & 2 & 1 \end{matrix}$$

Thus, $A' = (-1, 4), B' = (0, 6), C' = (1, 4)$ and $D' = (0, 2)$.

(*b*) The vertical line $x = 2$ has no intercept on y-axis and makes an angle of 90 degree with the x-axis.

So

$$m = \tan 900$$

$$= \infty \text{ and } c=0. \text{ Thus, the reflection matrix}$$

$$ML = T\text{–}v.R\text{–}\theta.My.R\theta.T\text{–}v, \text{ where } v = 2I$$

$$\begin{matrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 0 & 1 \end{matrix}$$

So the new coordinates A'B'C'D' of the reflected polygon ABCD can be found as:

$$[A'B'C'D'] = [ABCD].\ ML$$

$$\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 0 \\ 5 & 0 & 1 \end{matrix}$$

$$= \begin{matrix} 0 & -2 & 1 \end{matrix} . \begin{matrix} 0 & 1 & 0 \end{matrix} = \begin{matrix} 4 & -2 & 1 \end{matrix}$$

$$\begin{matrix} 1 & 0 & 1 \\ 4 & 0 & 1 \\ 3 & 0 & 1 \end{matrix}$$

$$\begin{matrix} 0 & 2 & 1 \\ 4 & 2 & 1 \end{matrix}$$

Thus, $A' = (5, 0), B' = (4, -2), C' = (3, 0)$ and $D' = (4, 2)$

Q.3. (*b*) What are the various types of multimedia authoring tools. (4)

Ans. Refer to Q.7 of End Term Examination 2015.

Q.4. (*a*) Given the four control points A(1, 1), B(2, 3), C(4, 3) and D(6, 4). Draw a bezier curve and show that it always passes through the first and the last control points. (5)

Ans. The equation for the Bezier curve is given as.

$$P(u) = (1-u)^3 P_1 + 3u(1-u)^2 P_2 + 3u^2(1-u)P_3 + u^3 P_4 \text{ for } 0 \le u \le 1$$

Where $P(u)$ is the point on the curve P_1, P_2, P_3, P_4.

Let us take $u = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$

$\therefore \quad P(0) = P_1 = (1, 1)$

$$\therefore \quad P\left(\frac{1}{4}\right) = \left(1-\frac{1}{4}\right)^3 P_1 + 3\frac{1}{4}\left(1-\frac{1}{4}\right)^2 P_2 + 3\left(\frac{1}{4}\right)^2\left(1-\frac{1}{4}\right)P_3 + \left(\frac{1}{4}\right)^3 P_4$$

$$= \frac{27}{64}(1,1) + \frac{27}{64}(2,3) + \frac{9}{64}(4,3) + \frac{1}{64}(6,4)$$

$$= \left[\frac{27}{64}\times 1 + \frac{27}{64}\times 2 + \frac{9}{64}\times 4 + \frac{1}{64}\times 6, \frac{27}{64}\times 1 + \frac{27}{64}\times 3 + \frac{9}{64}\times 3 + \frac{1}{64}\times 4\right]$$

$$= \left[\frac{123}{64}, \frac{139}{64}\right] = (1.9218, 2.1718).$$

$$P\left(\frac{1}{2}\right) = \left(1-\frac{1}{2}\right)^3 P_1 + 3\frac{1}{2}\left(1-\frac{1}{2}\right)^2 P_2 + 3\left(\frac{1}{2}\right)^2\left(1-\frac{1}{2}\right)P_3 + \left(\frac{1}{2}\right)^3 P_4$$

$$= \frac{1}{8}(1,1) + \frac{3}{8}(2,3) + \frac{3}{8}(4,3) + \frac{1}{8}(6,4)$$

$$= \left[\frac{1}{8}\times 1 + \frac{3}{8}\times 2 + \frac{3}{8}\times 4 + \frac{1}{8}\times 6, \frac{1}{8}\times 1 + \frac{3}{8}\times 3 + \frac{3}{8}\times 3 + \frac{1}{8}\times 4\right]$$

$$= \left[\frac{25}{8}, \frac{23}{8}\right] = (3.125, 2.875)$$

$$P\left(\frac{3}{4}\right) = \left(1-\frac{3}{4}\right)^3 P_1 + 3\frac{3}{4}\left(1-\frac{3}{4}\right)^2 P_2 + 3\left(\frac{3}{4}\right)^2\left(1-\frac{3}{4}\right)P_3 + \left(\frac{3}{4}\right)^3 P_4$$

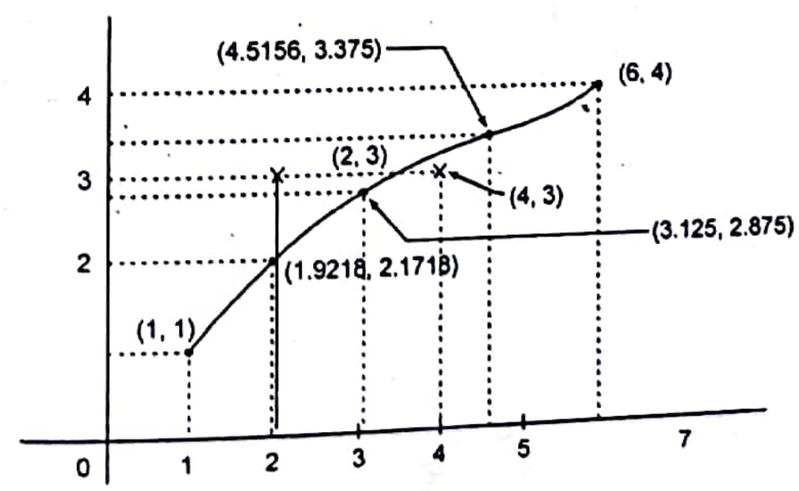
$$= \frac{1}{64}P_1 + \frac{9}{64}P_2 + \frac{27}{64}P_3 + \frac{27}{64}P_4$$

$$= \frac{1}{64}(1,1) + \frac{9}{64}(2,3) + \frac{27}{64}(4,3) + \frac{27}{64}(6,4)$$

$$= \left[\frac{1}{64}\times 1 + \frac{9}{64}\times 2 + \frac{27}{64}\times 4 + \frac{27}{64}\times 6, \frac{1}{64}\times 1 + \frac{9}{64}\times 3 + \frac{27}{64}\times 3 + \frac{27}{64}\times 4\right]$$

$$= \left[\frac{289}{64}, \frac{217}{64}\right] = (4.5156, 3.375)$$

P1=P3=(6,4)



Scanned by CamScanner

Q.4. (b) What is meant by the interactive and non-interactive graphical system. **(2.5)**

Ans. MULTIMEDIA: - Multimedia is the field concerned with the computer-controlled integration of text, graphics, drawings, still and moving images (Video), animation, audio, and any other media where every type of information can be represented, stored, transmitted and processed digitally.

INTERACTIVE MULTIMEDIA: Interactive multimedia has been called a "hybrid technology." It combines the storage and retrieval capabilities of computer database technology with advanced tools for viewing and manipulating these materials. Multimedia has a lot of different connotations, and definitions vary depending on the context. For the purposes of this Guide, in the context of upper secondary and postsecondary education, interactive multimedia is defined by three criteria:

- Interactive Multimedia is any package of materials that includes some combination of texts, graphics, still images, animation, video, and audio;
- These materials are packaged, integrated, and linked together in some way that offers users the ability to browse, navigate and analyze these materials through various searching and indexing features, as well as the capacity to annotate or personalize these materials;
- Interactive multimedia is always "reader-centered." In interactive multimedia, the reader contols the experience of reading the material by being able to select among multiple choices, choosing unique paths and sequences through the materials. One of the key features of interactive multimedia is the ability to navigate through material in whatever ways are most meaningful for individual users.

Non Interactive Multimedia: A fixed, self-contained and pre-conceived passive experience, fictional or not, tvpically featuring a narrative, often with a beginning, a middle and an end. The audience members are observers and not able to interact and participate and therefore incapable of influencing or changing the experience directly. The various forms of non-interactive entertainment have their own immersion and emotional delivery language: For novels it'd be choice of structure, words, perspective, pace, perception, etc whilst audiovisual media like film and TV use narrative structure, camera framing, editing, lighting, should, music, actor, performance, etc.

Q.4. (c) What is meant by composite transformations and how homogeneous co-ordinate system plays an important role in handling it? **(2.5)**

Ans. Homogeneous coordinate system convert 2D images to 3D images and it also removes the problem where after applying number of transformation on object yields similar results.For Translation, we have

$$[x'\ y'\ 1] = [x\ \ y\ \ 1].\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$$

or

$$P' = P.T(T_x, T_y)$$

Similarly for Rotation transformation, we have

$$[x'\ y'\ 1] = [x\ \ y\ \ 1].\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

or

$$P^1 = P.R.(\theta)$$

Finally for Scaling transformation, we have

$$[x'\ y'\ 1] = [x\ \ y\ \ 1].\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

or

$$P' = P.S(S_x, S_y).$$

A composite transformation (or composition of transformations) is two or more transformations performed one after the other. Sometimes, a composition of transformations is equivalent to a single transformation.

Scanned by CamScanner

SECOND TERM EXAMINATION [NOV. 2015]
THIRD SEMESTER [B. TECH]
COMPUTER GRAPHICS AND MULTIMEDIA
[ETCS–211]

Time: 1.30 Hrs. **MM : 30**

Note: *Q.No.1 which is compulsory and any two questions from the rest.*

Q.1. (a) Explain the terms – projection plane, projectors, center of projection. **(3)**

Ans. A projection plane, or plane of projection, is a type of view in which graphical projections from an object intersect. Projection planes are used often in descriptive geometry and graphical representation. A picture plane in perspective drawing is a type of projection plane.

Projection lines traveling between object point and the viewer are called projectors.

The projection from a point onto a plane or central projection: is called the **center of projection**

Q.1. (b) Write three differences between lossless and lossy compression techniques. **(3)**

Ans. File compression software looks for redundancies in a computer file's code, and eliminates those redundancies in order to reduce the size of the file. Most computer files have a lot of the same information listed over and over again, just like a sentence reuses many of the same letters and words — such as vowels or "the" and "and" — to express a complete thought. Compression software uses mathematical algorithms to catalog repeated pieces of information in a file, and then refers back to that information whenever it's needed. The result is a smaller file. Imagine needing to use "and" only once in a paragraph that has eight separate "ands" and you begin to get the idea.

All kinds of files can be compressed: Audio files, video files and still images each can be made smaller using a variety of algorithms. There are, broadly, two forms of compression: lossless and lossy.

Lossless compression recreates a compressed file as an identical match to its original form. All lossless compression uses techniques to break up a file into smaller segments, for storage or transmission, that get reassembled later. Lossless compression is used for files, such as applications, that need to be reproduced exactly like the original file. Lossy compression, on the other hand, eliminates repeated or "unnecessary" pieces of data, as we discussed above. When such a file is decompressed, you get the compression software's re-interpretation of the original file.

Lossy compression can't be used to compress anything that needs to be reproduced exactly — it can't just toss out redundant pieces and hope to program will still work. Instead, lossy compression will more often be used with data that is open to some level of human interpretation, such as an image, where the results can be "fudged" the tiniest bit so that files can get smaller without, in theory, anyone noticing. The digital cameras we use today, for example, offer varying levels of compression — in effect reworking the data of an image, throwing out whatever it can to reduce the image size while still retaining the snazzy picture you thought you took.

Q.1. (c) Explain the terms – illumination model and surface rendering? Which is calculated first and why? **(4)**

Ans. The contribution from the light that goes directly from the light source and is reflected from the surface is called a "local illumination model". So, for a local illumination model, the shading of any surface is independent from the shading of all other surfaces.

A "global illumination model" adds to the local model the light that is reflected from other surfaces to the current surface. A global illumination model is more comprehensive, more physically correct, and produces more realistic images. It is also more computationally expensive. We will first look at some basic properties of light and color, the physics of light-surface interactions, local illumination models, and global illumination models.

Surface rendering involves the careful collection of data on a given object in order to create a three-dimensional image of that object on a computer. It is an important technique used in a variety of industries.

One of the techniques to construct an image using surface rendering is with illumination. An illumination model shines a light on a surface and makes calculations based on the intensity of the light reflected back. This collects enough specific data to create a 3D image later.

Q.2. (a) What is meant by visible surface detection? What are the two types of surface detection methods? Explain any one of the Object Space Method. **(8)**

Ans. Visible Surface Detection is the problem of deciding which elements of a rendered screne are visible, and which are hidden.

Object Space: Space object algorithms have the advantage of retaining the relevant data and because of this ability the interaction of algorithm with the object becomes easier. The calculation done for the color is done only once. Object space algorithms also allow shadow generation to increase the depth of the 3 dimensional objects on the screen. The incorporation of these algorithms is done in software and it is difficult to implement them in hardware.

- Image space algorithms are much more efficient than object space algorithms.
- Object space algorithms are much more functional than image space algorithms.
- Color calculation in object space algorithms is done only one time and is retained by it but in image space algorithm the calculation once done is over written later.

z-buffering, also known as **depth buffering,** is the management of image depth coordinate in 3D graphics, usually done in hardware, sometimes in software. It is one solution to the visibility problem is the problem of deciding which elements of a rendered scene are visible, and which are hidden. The painter's algorithm is another common solution which, though less efficient, can also handle non-opaque scene elements.

When an object is rendered, the depth of a generated pixel (z coordinate) is stored in a buffer (the **z-buffer** or **depth buffer**). This buffer is usually arranged as a two-dimensional array (x–y) with one element for each screen pixel. If another object of the scene must be rendered in the same pixel, the method compares the two depths and overrides the current pixel if theo bject is closer to the observer. The chosen depth is then saved to the z-buffer, replacing the old one. In the end, the z-buffer will allow the method to correctly reproduce the usual depth perception: a close object hides a farther one. This is called **z-culling.**

Scanned by CamScanner

Q.2. (b) What is a vanishing point? How many principal vanishing points exists? **(2)**

Ans. A **vanishing point** is a point in the picture plane that is the intersection of the projections (or drawings) of a set of parallel lines in space on to the picture plane. When the set of parallels is perpendicular to the picture plane, the construction is known as one-point perspective and their vanishing point corresponds to the oculus or eye point from which the image should be viewed for correct perspective geometry.

Any number of vanishing points are possible in a drawing, one for each set of parallel lines that are at an angle relative to the plane of the drawing.

Q. 3. (a) What is drawback of Gourand Shading? How it is rectified by another shading technique. **(5)**

Ans. Phong shading refers to an interpolation technique for surface shading in 3D computer graphics. It is also called Phong interpolation or normal-vector interpolation shading.Specifically, it interpolates surface normals across rasterized polygons and computes pixel colors based on the interpolated normals and a reflection model. *Phong shading* may also refer to the specific combination of Phong interpolation and the Phong reflection model.

Phong shading improves upon Gourad shading and provides a better approximation of the shading of a smooth surface. Phong shading assumes a smoothly varying surface normal vector. The Phong interpolation method works better than Gourand shading when applied to a reflection model that has small specular highlights such as the Phong reflection model.

The most serious problem with Gouraud shading occurs when specular highlights are found in the middle of a large polygon. Since these specular highlights are absent from the polygon's vertices and Gouraud shading interpolates based on the vertex colors, the specular highlight will be missing from the polygon's interior. This problem is fixed by Phong shading.

Unlike Gourand shading, which interpolates colors across polygons, in Phong shading a normal vector is linearly interpolated across the surface of the polygon from the polygon's vertex normals. The surface normal is interpolated and normalized at each pixel and then used in a reflection model, e.g. the Phong reflection model, to obtain the final pixel color. Phong shading is more computationally expensive than Gourand shading since the reflection model must be computed at each pixel instead of at each vertex.

In modern graphics hardware, variants of this algorithm are implemented using pixel or fragment shaders.

Q.3. (b) What is JPEG? Explain the four modes of operation of JPEG briefly. **(5)**

Ans. JPEG is a lossy compression technique for color images. Although it can reduce files sizes to about 5% of their normal size, some detail is lost in the compression.

1. The **sequential** DCT mode of operation, where the picture is scanned in the same way as in *part one* (but not in our zigzag way, column-wise from left to right, but line-wise from top to bottom, just as in reading).

2. The **progressive** DCT mode of operation, where the picture is displayed in its *entirety* concurrently with the transmission of the bitstream, at first imperfect and then gradually improving.

3. The **lossless** mode of operation, where the file is only compressed, with no data lost by cosine transforms or quantization.

4. The **hierarchical** (DCT or lossless) mode of operation, where the picture is stored at multiple resolutions for different uses (low-resolution screen, high-resolution printer, etc.), in such a way that the lower-resolution images are stored with supplementary data which can be added on to produce higher-resolution images as required.

Q. 4. (a) Obtain the projection matrix of an object on Z=d plane and the center of projection is (0,0,d). Also notify the type of perspective projection. **(5)**

Ans. $Z = d$ is parallel to xy plane..

Thus the view plane normal vector N is the same as the normal vector k to the xy plane. *i.e.* N=k

Choosing the reference point as $R_0(0,0,d)$

$$N(n_1,n_2,n_3) = (0,0,1)$$
$$R_0 = (X_0, Y_0, Z_0) = (0, 0, d)$$
$$D_0 = n_1X_0 + n_2Y_0 + n_3Z_0 = d$$

Projection matrix is

$$\begin{matrix} D & 0 & 0 & 0 \\ 0 & D & 0 & 0 \\ 0 & 0 & D & 0 \\ 0 & 0 & 1 & 0 \end{matrix}$$

Q.4. (b) Why do we compress the data? Explain any one of the compression technique. **(5)**

Ans. Huffman Code: Huffman Code assigns shorter encodings to elements with a high frequency, F:e. It differs from block encoding in that it is able to assign codes of different bit lengths to different elements. Elements with the highest frequency, F:e, get assigned the shortest bit length code. The key to decompressing huffman code is a huffman tree.

A huffman tree is a special binary tree called a trie. (pronounced try) A binary trie is a binary tree in which a 0 represents a left branch and a 1 represents a right branch. The numbers on the nodes of the binary trie represent the total frequency, F, of the tree below. The leaves of the trie represent the elements, e, to be encoded. The elements are assigned the encoding which corresponds to their place in the binary trie. Below is an example.

NOTE: Spaces are included in the message and in the encodings for readability purposes only. The space is not considered as a character in the message and therefore is not encoded as an element in the message.

Message to be Encoded

dad ade fade bead ace dead cab bad fad cafe face

Block Encoding

011 000 011 000 011 100 101 000 011 101 001 100 000 011 000 010 100 011 100 000 011 010 000 001 001 000 011 101 000 011 010 000 101 100 101 000 010 100

The block encoding above is a fixed length encoding. If a message contains i elements, block encoding requires **log i** bits to encode each element, e.

Spaces have been inserted between the strings of bits which represent each character in both the Block Encoding and the Huffman Encoding.

Scanned by CamScanner

Huffman Encoding

01 10 01 10 01 111 110 10 01 111 000 111 10 01 10 001 111 01 111 10 01 001 10 000 000 10 01 110 10 01 001 10 110 111 110 10 001 111

element	frequency	block code	huffman code
a	11	000	10
b	3	001	000
c	4	010	001
d	9	011	01
e	7	100	111
f	4	101	110

Entropy: The average information content of a message is called its enropy. The information content is related to uncertainty. The less likely a message is to occur the larger its information content. This makes sense if we think of an example: if a person knows what message is about to be sent to him, how much new information has he learned by receiving that message? None. This is all that the above statement is saying.

Entropy, E, is information content. The entropy of a source is inversely proportional to its probability of occurrence:

$$E = -\log P$$

We use the the log function because we are converting all sources into the binary alphabet, B.

The same rule applies to an element, e, in a message, M. Its entropy can be defined as:

$$E{:}e = -\log (P{:}e)$$

P:e is the probability of an element in a message. It is equal to that element's frequency, F:e, divided by the frequency of the entire message, F:M:

$$P{:}e = F{:}e / F{:}M$$

The average information content or average entropy for a message, E:M, can now be defined. We know the entropy for each element in the message is E:e. We will index the elements, e, in a message, M, by assigning them integer positions, i, according to their place in the message. For example: the first element, e, in message M will be assigned 1, the second will be assigned 2...the i-th will be assigned i. Now we can define the entropy for an entire message, E:M, where there are i elements:

$$E{:}M = P{:}1*E{:}1)+(P{:}2*E{:}2)+(P{:}3*E{:}3)+...+(P{:}i*E{:}i)$$

Entropy is an important concept to data compression. The entropy of an element (E:e) is the minimum number of bits needed to encode that element. The entropy of an entire message (E:M) is the minimum number of bits needed to encode the entire message with a lossless compression. The entropy of a message can be used to determine if data compression is worth attempting. It can also be used to evaluate the effectiveness of a compression. The number of bits in a compressed code can be compared to the entropy for that message (E:M) revealing how close to optimal compression one's compressed code is.

END TERM EXAMINATION [DEC. 2015]
THIRD SEMESTER [B. TECH]
COMPUTER GRAPHICS AND MULTIMEDIA
[ETCS–211]

Time: 3 Hrs. **MM : 75**

Note: *Attempt any five questions including Q.No.1 which is compulsory. Select one question from each unit.*

Q.1. Write short note on the following: **(10 x 2.5 = 25)**

Q.1.(a) Discuss the classification of computer Graphics.

Ans. A broad classification of major subfields in computer graphics:

1. Geometry: studies ways to represent and process surfaces.
2. Animation: studies with ways to represent and manipulate motion.
3. Rendering: studies algorithms to reproduce light transport.
4. Imaging: studies image acquisition or image editing.
5. Topology:studies the behaviour of spaces and surfaces.

Q.1. (b) What is the use of control points in spline curve?

Ans. Moving control points is the most obvious way of changing the shape of a B-spline curve. The position of control point P_i only affects the curve $C(u)$ on interval $[u_i, u_{i+p+1})$, where p is the degree of a B-spline curve. In fact, the shape change is *translational* in the direction of the control point being moved. More precisely, if control point P_i is moved in certain direction to a new position Q_i, then point $C(u)$, where u is in (u_i, u_{i+p+1}), will be moved in the same direction from P_i to Q_i. However, the distance moved is different from point to point.

Q.1 (c) Explain homogeneous coordinates and give its importance in transformation operations.

Ans. Homogeneous coordinates are a system of coordinates used in projective geometry, as Cartesian coordinates are used in Euclidean geometry. They have the advantage that the coordinates of points, including points at infinity, can be represented using finite coordinates. Formulas involving homogeneous coordinates are often simpler and more symmetric than their Cartesian counterparts. Homogeneous coordinates have a range of applications, including computer graphics and 3D computer vision, where they allow affine transformations and, in general, projective transformations to be easily represented by a matrix.

If the homogeneous coordinates of a point are multiplied by a non-zero scalar then the resulting coordinates represent the same point. An additional condition must be added on the coordinates to ensure that only one set of coordinates corresponds to a given point, so the number of coordinates required is, in general, one more than the dimension of the projective space being considered. For example, two homogeneous coordinates are required to specify a point on the projective line and three homogeneous coordinates are required to specify a point on the projective plane.

- Any point in the projective plane is represented by a triple (X, Y, Z), called the homogeneous coordinates or projective coordinates of the point, where X, Y and Z are not all 0.

- The point represented by a given set of homogeneous coordinates is unchanged if the coordinates are multiplied by a common factor.

Scanned by CamScanner

• Conversely, two sets of homogeneous coordinates represent the same point if and only if one is obtained from the other by multiplying by a common factor.

• When Z is not 0 the point represented is the point $(X/Z, Y/Z)$ in the Euclidean plane.

• When Z is 0 the point represented is a point at infinity.

Note that the triple (0, 0, 0) is omitted and does not represent any point. The origin is represented by (0, 0, 1).

Q. 1. (*d*) Differentiate between cavalier and cabinet projection.

Ans. A pictorial view of an object showing its elevation, plan, or section to scale with parallel lines projected from the corners, at 45 degrees or any other angle, indicating the other sides is called oblique projection.

Cavalier projection (sometimes **cavalier perspective** or **high view point**) a point of the object is represented by three coordinates, x, y and z. On the drawing, it is represented by only two coordinates, x'' and y''. On the flat drawing, two axes, x and z on the figure, are perpendicular and the length on these axes are drawn with a 1:1 scale; it is thus similar to the dimetric projections, although it is not an orthographic projection, as the third axis, here y, is drawn in diagonal, making an arbitrary angle with the x'' axis, usually 30 or 45°.

Cabinet projection (sometimes **cabinet perspective**) stems from its use in illustrations by the furniture industry. Like cavalier perspective, one face of the projected object is parallel to the viewing plane, and the third axis is projected as going off in an angle (typical 30° or 45°). Unlike cavalier projection, where the third axis keeps its length, with cabinet projection the length of the receding lines is cut in half.

Q.1. (*e*) What is anti aliasing? how it is achieved?

Ans. Pixel phasing is an antialiasing method where the stair-steps look are removed out by placing the electron beam to an exact position determined by the object geometry.

Dithering scatters different colored pixels in an image to make it appear as though there are intermediate colors in images with a limited color palette.

Dithering is often used to reduce banding in GIFs with graduated color transitions. Most software provides options that let you control the appearance of the scattered pixels; for example, the dithering can be a rigid pattern, random noise, or diffusion. Keep in mind that dithering often increases the file size of an image, but in many cases the improved appearance is worth the trade-off.

Q.1. (*f*) Why is gourand shading also referred to as interpolation shading?

Ans. Gouraud shading, named after Henri Gouraud, is an interpolation method used in computer graphics to produce continuous shading of surfaces represented by polygon meshes. In practice, Gouraud shading is most often used to achieve continuous lighting on triangle surfaces by computing the lighting at the corners of each triangle and linearly interpolating the resulting colours for each pixel covered by the triangle.

Q.1. (*g*) What is the full form of PIXEL? What is the effect on picture quality on increasing or decreasing the n umber of pixel in a computer screen?

Ans. A Pixel (picture element) or pel is the smallest unit of picture that can be represented or controlled. Each pixel has its own address. The address of a pixel corresponds to its coordinates. Pixels are normally arranged in a two-dimensional grid, and are often represented using dots.

Screen resolution is the number of pixels (W× H) displayed by the monitor. More pixels = better image quality, but it also means more strain on the graphics card, which = lower performance. It also depends on the type of monitor you're using. If you're using a CRT, you can safely change resolution without any drop in image quality. LCDs,

on the other hand, are designed to run at a recommended resolution, and any change from that setting – higher or lower

Q.1. (*h*) Differentiate between Cel animation and Path animation.

Ans. A Cel-base animation is made on Photoshop using lots of frames to make a sequence. When you play it it is really jumpy and doesn't run smoothly because its only flashing each picture that you create.

A Path-based animation is an animation that is created on Macromedia Flash. When you creates it you only use 2 frames and when you play it it runs really smoothly and glides from top to bottom.

Q.1. (*i*) What do you mean by data compression? Differentiate between lossless and lossey compression?

Ans. Refer Q.1. (*b*) Second Term Examination 2015.

Q.1. (*j*) For showing a cartoon or animation related content which kind of computer screen will be used, a screen with high or low persistence level? Explain with reason.

Ans. For showing a cartoon or animation related content a high persistence computer screen will be used, with very high refresh rate will be used so as to display the fast change in per pixel intensity.

Q.2. Find the pixel position using bresenham line drawing algorithm when converting from A(2,5) to B(5,10).

Ans.

$$x_1 = 2, y_1 = 5$$
$$x_2 = 5, y_2 = 10$$
$$\Delta x = 5 - 2 = 3$$
$$\Delta y = 10 - 5 = 5$$

Δy *is* $> \Delta x$ so exchange Δx and Δy

now

$$\Delta x = 5$$
$$\Delta y = 3$$

Plot (2, 5)

$$e = 2 \times 3 - 5 = 1$$

$e > o$, so

$$x = x + 1 = 2 + 1 = 3$$
$$y = 5$$

and

$$e = 1 - 2 \times 5 = -9$$

$e < 0$

So

$$y = 5 + 1 = 6 \quad (3, 6)$$
$$e = e + 2 \times \Delta y = -9 + 6 = -3$$

So, Point is (3, 7)

$$e = e + 2(3) = -3 + 6 = 3$$

Plot (4, 7)

similarly calculate the next point as (5, 8)

Q.3. (*a*) Derive the transformation matrix that reflects a object about line Y = X. **(6)**

Ans.

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Thix transformation matrix will affect a reflection in the line $y = x$ for the point matrix, as shown in the example below.

Scanned by CamScanner

Transformation matrix	Point matrix		Multiplication		New Point matrix
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	=	$\begin{bmatrix} (4x0)+(3x1) \\ (4x1)+(3x0) \end{bmatrix}$	=	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Q.3. (*b*) Magnify the triangle with vertices A(0, 0), B(1, 1), C(5, 2) to twice its size while keeping C(5, 2) fixed.

Ans. The transformation has three steps:

1. Translate the center of scaling, C(5, 2), to the origin.
2. Scale the triangle.
3. Translate back to the original location. We apply the series of transformations to triangle ABC as follows:

$$[A'B'C'] = T^{-1}ST[ABC] = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}[ABC]$$

Q. 4. Given a clipping window *A*(20, 20), *B*(60, 20), *C*(60, 40), *D*(20, 40). Using Cohen Sutherland algorithm find the visible portion of the line segment joining the points *P*(40,80) and *Q*(120,30). **(12.5)**

Ans. End point code of *P* is (1000)

End point code of *Q* is (0010)

Both end point codes are not zero, and their logical AND is zero. Hence, line cannot be rejected as invisible.

Slope of PQ = $$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{30-80}{120-40} = \frac{-50}{80} = \frac{-5}{8}$$

$$[(x_1, y_1) = (40, 80); (x_2, y_2) = (120, 30)]$$

and intersections with window edge are-

- Left, $x_L = 20, y = m(x_L - x_1) + y_1$

$$= \frac{-5}{8}(20-40)+80$$

$$= 92.5$$

(which is greater than y_T and hence rejected)

- Right, $x_R = 60, y = m(x_R - x_1) + y_1$

$$= \frac{-5}{8}(60-40)+80$$

$$= 67.5$$

(which is greater than y_T and hence rejected)

- Top, $y_T = 40, x = x_1 + \frac{1}{m}(y_T - y_1)$

$$= 40 + \frac{1}{-\frac{5}{8}}(40-80)$$

$$= 104.$$

(which is greater than x_R and hence rejected)

- Bottom, $y_B = 20, x = x_1 + \frac{1}{m}(y_B - y_1)$

$$= 40 + \frac{1}{-\frac{5}{8}}(20-80)$$

$$= 136$$

(which is greater than x_R and hence rejected)

Q. 5 What are the major differences between Bezier curve B-Spline curve and hermite curves? Make a comparison between them and suggest which curve should be used in computer graphics. **(12.5)**

Ans. Bezier curves are parametric curves used frequently in modeling smooth surfaces in computer graphics and many other related fields. These curves can be scaled indefinitely. Linked Bezier curves contain paths that are combinations that are intuitive and can be modified. This tool is also made use of in controlling motions in animation videos. When programmers of these animations talk about the physics involved, they are in essence talking about these Bezier curves. Bezier curves were first developed by Paul de Castlejau using Castlejau's algorithm, which is considered a stable method to develop such curves. However, these curves became famous in 1962 when French designer Pierre Bezier used them to design automobiles.

The most popular Bezier curves are quadratic and cubic in nature as higher degree curves are expensive to draw and evaluate. An example of the equation of Bezier curve involving two points (linear curve) is as follows

$$B(t) = P_0 + t(P_1 - P_0) = (1-t)P_0 + tP_1, t \in [0, 1]$$

B-Spline curves are considered as a generalization of Bezier curves and as such share many similarities with it. However, they have more desired properties than Bezier curves. B-Spline curves require more information such as degree of the curve and a knot vector, and in general involve a more complex theory than Bezier curves. They however possess many advantages that off set this shortcoming. Firstly, a B-Spline curve can be a Bezier curve whenever the programmer so desires. Further B-Spline curve offers more control and flexibility than Bezier curve. It is possible to use lower degree curves and still maintain a large number of control points. B-Spline, despite being more useful are still polynomial curves and cannot represent simple curves like circles and ellipses. For these shapes, a further generalization of B-Spline curves known as NURBS are used.

Bezier vs B-Spline curves

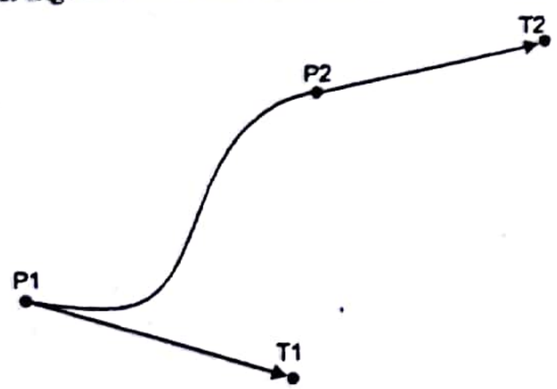
- Both Bezier and B-Spline curves are used for drawing and evaluating smooth curves, especially in computer graphics and animations.
- B-Spline are considered a special case of Bezier curves
- B-Spline offer more control and flexibility than Bezier curves

Hermite curves are very easy to calculate but also very powerful. They are used to smoothly interpolate between key-points (like object movement in keyframe animation or camera control). Understanding the mathematical background of hermite curves will help you to understand the entire family of splines. Maybe you have some experience with 3D programming and have already used them without knowing that (the so called kb-splines, curves with control over tension, continuity and bias are just a special form of the hermite curves).

Scanned by CamScanner

To calculate a hermite curve you need the following vectors:

- P1: the startpoint of the curve
- T1: the tangent (*e.g.* direction and speed) to how the curve leaves the startpoint
- P2: he endpoint of the curve
- T2: the tangent (*e.g.* direction and speed) to how the curves meets the endpoint



These 4 vectors are simply multiplied with 4 hermite basis functions and added together.

h1(s) = 2s^3 – 3s^2 + 1

h2(s) = -2s^3 + 3s^2

h3(s) = s^3 – 2s^2 + s

h4(s) = s^3 – s^2

A free form curve ic mostly used in computer graphics.

Q.6. Explain Z buffer algorithm. How A buffer method removes the drawback of Z- buffer algorithm. **(12.5)**

Ans. In computer graphics, Z-buffering, also known as depth buffering, is the management of image depth coordinates in three-dimensional (3–D) graphics, usually done in hardware, sometimes in software. It is one solution to the visibility problem, which is the problem of deciding which elements of a rendered scene are visible, and which are hidden. The A buffer algorithm is more efficient than Z buffer as it adds one more field called depth field which holds a positive or negative real number.by using this depth field it can calculate intensities of various layers as present in a canvas which is not possible by using Z buffer algorithm.

This method is developed by Cutmull. It is an image-space approach. The basic idea is to test the Z-depth of each surface to determine the closest (visible) surface.

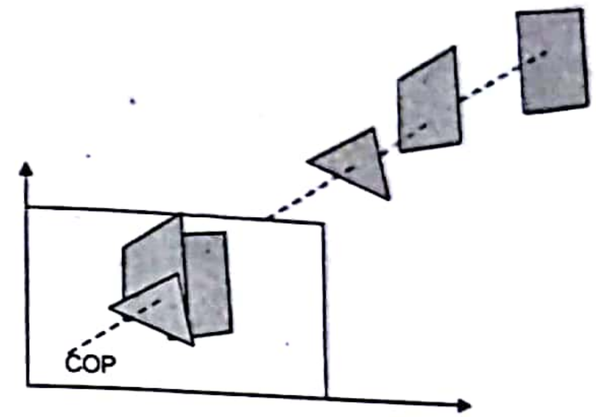
In this method each surface is processed separately one pixel position at a time across the surface. The depth values for a pixel are compared and the closest (smallest z) surface determines the color to be displayed in the frame buffer.

It is applied very efficiently on surfaces of polygon. Surfaces can be processed in any order. To override the closer polygons from the far ones, two buffers named **frame buffer** and **depth buffer,** are used.

Depth buffer is used to store depth values for (x, y) position, as surfaces are processed ($0 \le$ depth ≥ 1).

The **frame buffer** is used to store the intensity value of color value at each position (x, y).

The z-coordinates are usually normalized to the range [0, 1]. The 0 value for z-coordinate indicates back clipping pane and 1 value for z-coordinates indicates front clipping pane.



Algorithm

Step-1 – Set the buffer values –

Depthbuffer $(x, y) = 0$

Framebuffer (x, y) = background color

Step-2 " Process each polygon (One at a time)

For each projected (x, y) pixel position of a polygon, calculate depth z.

If Z > depthbuffer (x, y)

Compute surface color,

set depthbuffer $(x, y) = z$,

framebuffer (x, y) = surfacecolor (x, y)

Advantages

- It is easy to implement.
- It reduces the speed problem if implemented in hardware.
- It processes one object at a time.

Disadvantages

- It requires large memory.
- It is time consuming process.
- Which is overcome by A buffer method.

Q.7. Discuss the various multimedia authoring tools in detail. **(12.5)**

Ans. The integration of audio, video, graphics and text on the desktop promises to fundamentally challenge the centuries-old model of the printed document as the basis for information exchange. Before this potential can be realized, however, systems must be devised that enable the production and presentation of complex, inter-related media objects. These systems are generically called multimedia authoring tools.

In this we consider the development of multimedia authoring tools, examine the current state of the art, and then discuss a set of research challenges that need to be addressed before the full potential of multimedia output technology can be effectively utilized to share information.

Types of Authoring Tools:

1. Card- or page-based tools.
2. Icon-based, event-driven tools.
3. Time-based tools.

Scanned by CamScanner

Card-based or page-based tools:

1. The elements are organized as pages of a book or a stack of cards.
2. Card-or page-based authoring systems
3. Best used when the bulk of your content consists of elements that can be viewed individually
4. The pages of a book or cards in a card file.
5. Link these pages or cards into organized sequences.
6. Jump, on command, to any page
7. Play sound elements and launch animations and digital video.

Icon-or object-based, event-driven tools

1. Multimedia elements and interaction cues (events) are organized as objects in a structural framework or process.
2. Simplify the organization of your project
3. Display flow diagrams of activities along branching paths.

Time-based tools

1. Elements and events are organized along a timeline, with resolutions as high as or higher than 1/30 second.
2. Time-based tools
3. Best to use when you have a message with a beginning and an end.
4. Played back at a speed that you can set
5. Other elements (such as audio events) are triggered at a given time or location in the sequence of events.
6. Jumps to any location in a sequence

Q.8. (*a*) What are the various shading models? Enlist and explain any one in brief.

Ans. In computer graphics, shading refers to the process of altering the color of an object/surface/polygon in the 3D scene, based on its angle to lights and its distance from lights to create a photorealistic effect. Shading is performed during the rendering process by a program called a shader. hading alters the colors of faces in a 3D model based on the angle of the surface to a light source or light sources.

The first image below has the faces of the box rendered, but all in the same color. Edge lines have been rendered here as well which makes the image easier to see.

The second image is the same model rendered without edge lines. It is difficult to tell where one face of the box ends and the next begins.

The third image has shading enabled, which makes the image more realistic and makes it easier to see which face is which. Shading is also dependent on the lighting used. Usually, upon rendering a scene a number of different lighting techniques will be used to make the rendering look more realistic. Different types of light sources are used to give different effects.

Phong shading algorithm: Using highlights avoids surfaces that look dull, lifeless, boring, blah. One cool thing about highlights is that, in addition to being all bright an shiny, they change as the object moves. In this way, highlights provide useful visual information about shape and motion.

The simplest model for approximating surface highlights is the *Phong* model, originally developed by Bui-Tong Phong. In this model, we think of the interaction between light and a surface as having three distinct components:

- Ambient
- Diffuse
- Specular

The *ambient* component is usually given a dim constant value, such as 0.2. It approximates light coming from a surface due to all the non-directional ambient light

that is in the environment. In general, you'll want this to be some tinted color, rather than just gra y. $[r_a, g_a, b_a]$. For example, a slightly greenish object might have an ambient color of [0.1, 0.3, 0.1].

The *diffuse* component is that dot product **n•L** that we discussed in class. It approximates light, originally from light source **L**, reflecting from a surface which is diffuse, or non-glossy. One example of a non-glossy surface is paper. In general, you'll also want this to have a non-gray color value, so this term would in general be a color defined as: $[r_d, g_d, b_d]$ (n•L).

Finally, the Phong model has a provision for a highlight, or *specular*, component, which reflects light in a shiny way. This is defined by $[r_s, g_s, b_s](\mathbf{R}\bullet\mathbf{L})^p$, where **R** is the mirror reflection direction vector we discussed in class (and also used for ray tracing), and where p is a specular power. The higher the value of p, the shinier the surface.

The complete Phong shading model for a single light source is:

$$[r_a, g_a, b_a] + [r_d, g_d, b_d]\max_0(\mathbf{n}\bullet\mathbf{L}) + [r_s, g_s, b_s]\max_0(\mathbf{R}\bullet\mathbf{L})^p$$

If you have multiple light sources, the effect of each light source $\mathbf{L}_i$ will geometrically depend on the normal, and therefore on the diffuse and specular components, but not on the ambient component. Also, each light might have its own [r,g,b] color. So the complete Phong model for multiple light sources is:

$$[r_a, g_a, b_a] + \Sigma_i ([L_r, L_g, L_b] ([r_d, g_d, b_d] \max_0(\boldsymbol{n} \bullet \boldsymbol{L}_i) + [r_s, g_s, b_s] \max_0(\mathbf{R}\bullet\mathbf{L}_i)^p))$$

Q.8. (*b*) How the MIDI files are generated for audio and video information? Explain in brief.

Ans. MIDI (Musical Instrument Digital Interface) is a protocol designed for recording and playing back music on digital synthesizers that is supported by many makes of personal computer sound cards. Originally intended to control one keyboard from another, it was quickly adopted for the personal computer. Rather than representing musical sound directly, it transmits information about how music is produced. The command set includes note-ons, note-offs, key velocity, pitch bend and other methods of controlling a synthesizer. The sound waves produced are those already stored in a wavetable in the receiving instrument or sound card.

Since a MIDI file only represents player information, it is far more concise than formats that the sound directly. An advantage is very small file size. A disadvantage is the lack of specific sound control.

With a program that provides this interface, you can create music using a standard keyboard or other input device. You or others can then play your MIDI-conforming creation with the same or another program and a sound card as a music synthesizer. The MIDI program may come with a graphical user interface that looks like a sound studio control room. Many sound cards come as a package with MIDI software (for example, Media Vision's Pro Audio Studio 16).

The MIDI protocol uses eight-bit serial transmission with one start bit and one stop bit, has a 31.25 Kbs data rate, and is asynchronous. Connection is made through a five-pin DIN plug, of which three pins are used.

Q.9 (*a*) What are the different compression techniques? Discuss any one image compression algorithm in detail. How does it handle different image formats. (6)

Ans. There are two categories of compression techniques used with digital graphics, lossy and **lossless**. Whilst each uses different techniques to compress files, both have the same aim. To look for duplicate data in the graphic and use a much more compact data representation.

Lossy and Lossless each have various methods which are used by different file formats and achieve different results. Therefore not all lossy or lossless formats will use the same methods. It is beyond the scope of this Unit to look at these methods in

Scanned by CamScanner

detail so you will not be assessed on them. The Unit entitled Digital Imaging: Bitmaps covers compression methods in more detail. If you are a bit unclear about this, the following may help.

- Lossy compression methods include DCT (Discreet Cosine Transform), Vector Quantisation and Huffman coding
- Lossless compression methods include RLE (Run Length Encoding), string-table compression, LZW (Lempel Ziff Welch) and zlib.

Compression methods are otherwise known as *algorithms*, which are calculations that are used to compress files. Organisations that create file formats create their own algorithms and compete with each other to create the best format. You will look at this in more detail later in the 'File Format' section of this course.

Bitmaps consist of pixels that have the same values. Think of a photograph of a setting. There will be areas of pixels in the photo that all have the same value. There is no need to store every single pixel. Algorithms look for redundant data, is repeat values that aren't needed, in the graphic and create a new representation of the data. This representation creates a smaller file size but keeps the relevant data for displaying the graphic. Essentially when you look at a compressed file you should see no *or* little difference in the graphic.

Run-length encoding (RLE) is a very simple form of data compression in which *runs* of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs: for example, relatively simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could potentially double the file size.

For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. Let us take a hypothetical single scan line, with B representing a black pixel and W representing white:

WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWW
WWWWWWWWWWWWWBWWWWWWWWWWWWWW

If we apply the run-length encoding (RLE) data compression algorithm to the above hypothetical scan line, we get the following:

12W1B12W3B24W1B14W

This is to be interpreted as twelve Ws, one B, twelve Ws, three Bs, etc.

The run-length code represents the original 67 characters in only 18. Of course, the actual format used for the storage of images is generally binary rather than ASCII characters like this, but the principle remains the same. Even binary data files can be compressed with this method; file format specifications often dictate repeated bytes in files as padding space. However, newer compression methods such as DEFLATE often use LZ77-based algorithms, a generalization of run-length encoding that can take advantage of runs of strings of characters (such as BWWBWWBWWBWW).

Run-length encoding performs lossless data compression and is well suited to palette-based iconic images. It does not work well at all on continuous-tone images such as photographs, although JPEG uses it quite effectively on the coefficients that remain after transforming and quantizing image blocks.

Common formats for run-length encoded data include Truevision TGA, PackBits, PCX and ILBM.

Run-length encoding is used in fax machines (combined with other techniques into Modified Huffman coding). It is relatively efficient because most faxed documents are mostly white space, with occasional interruptions of black.

Data that have long sequential runs of bytes (such as lower-quality sound samples) can be RLE compressed after applying a predictive filter such as delta encoding.

Q.9. (b) Explain reference models for multimedia synchronization with the help of diagram. (6.5)

Ans. A reference model is needed to understand the requirements of multimedia synchronization, identify and structure runtime mechanisms that can support these requirements, identify interfaces between runtime mechanisms, and compare solutions for multimedia synchronization systems. Figure 1 shows a reference model for multimedia synchronization systems. Each layer implements synchronization mechanisms which are provided by an appropriate interface. These interfaces can be used to specify or enforce the temporal relationships. Each interface can be used by the application directly, or by the next higher layer to implement an interface. Higher layers offer higher programming and QoS abstractions.

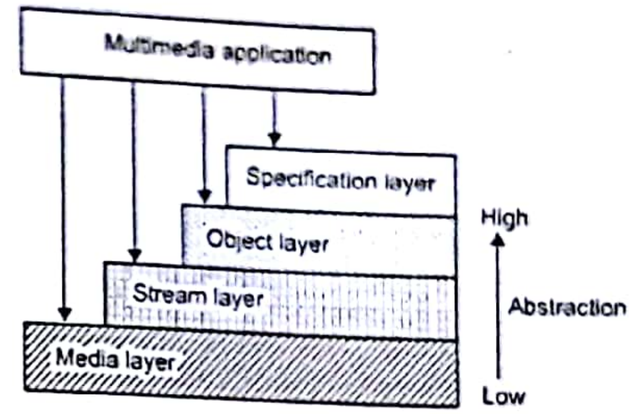


Fig. 1

Media Layer: An application operates on a single continuous media stream, which is treated as a sequence of LDUs. Networking components must be taken into account. Provides access to files and devices.

Stream Layer: The stream layer operates on continuous media streams as well as groups of media streams. In a group, all streams are presented in parallel by using mechanisms for interstream synchronization. QoS parameters will specify intrastream and interstream synchronization requirements. Continuous media is seen as a data flow with implicit time constraints; individual LDUs are not visible. An application using the stream layer is responsible for starting, stopping and grouping the streams, and for the definition of the required QoS in terms of timing parameters supported by the stream layer. It is also responsible for the synchronization with time-independent media objects. Tasks include resource reservation and LDU process scheduling.

Object Layer: The object layer operates on all media streams and hides the differences between continuous and discrete media. An application that interacts with this layer will be presented with a view of a complete, synchronized presentation. This layer takes a complete synchronization specification as its input and is responsible for the correct schedule of the overall presentation.

Specification Layer: This layer contains applications and tools that are allowed to create synchronization specifications (*e.g.*, authoring tools, multimedia document editors). The specification layer is also responsible for mapping user-required QoS parameters to the qualities offered at the object layer interface. Synchronization specifications can be:

- Interval-based: specifications of the temporal relations between the time intervals of the presentation of media objects
- Axes-based: allows presentation events to be synchronized according to shared axes, *e.g.*, a global timer
- Control flow-based: at specified points in presentations, they are synchronized
- Event-based: Events in the presentation trigger presentation actions

Scanned by CamScanner

Synchronization in a Distributed Environment: Synchronization in a distributed environment is complex, because there may be more than one source of multimedia data, and more than one sink consuming it. The synchronization information for the various media stream may also reside at different sources.

Transport of the synchronization specification: The sink needs to have the synchronization information available to correctly display an object. There are three main approaches to delivering the synchronization information to the sink:

- Delivery of the synchronization information before the start of the presentation
- Use of an additional synchronization channel
- Multiplexed data streams

If the multimedia presentation is live and multiple parties are involved, then none of the approaches above is suitable for delivering synchronization information to the sink (s) in a timely fashion. Figure 2 shows typical communication patterns.

Of particular interest here, is that if multiple sinks are involved, then they will receive identical data. It would be inefficient if the data were replicated at the source for separate transmission to each of the sinks. It would also be inefficient if the same operation was carried out at different sinks. *Multicasting* or broadcasting of streams is the responsibility of the stream layer, whereas efficient planning of operation execution in the different communication patterns is a responsibility of the object layer.

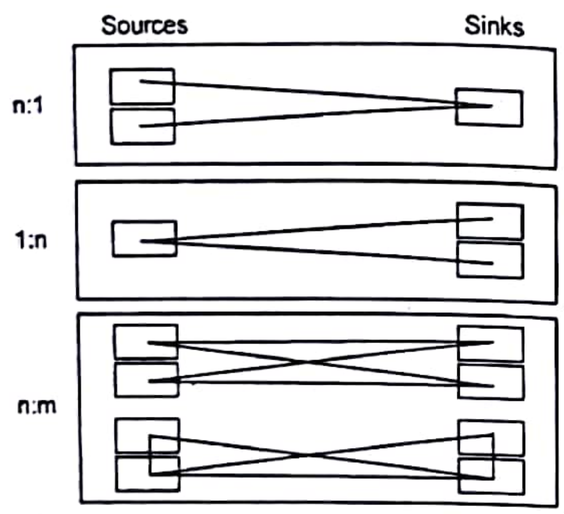


Fig.2

Multi-Step Synchronization: In a distributed environment, synchronization is typically a multi-step process, during which the synchronization must be maintained so as to enable the sink to perform the final synchronization. The synchronization steps are:

- During object acquisition, *e.g.*, during frame digitization
- During retrieval, *e.g.*, synchronized access to frames of a stored video
- During delivery of the LDUs to the network
- During the transport of the LDUs, *e.g.*, using isochronous protocols
- At the sink, *e.g.*, synchronized delivery to the output devices
- Within the output device

With many different points at which synchronization must occur decisions must be made about how to implement it. A first decision is the selection of the type of transport for the synchronization specification. In runtime, decisions must be taken concerning the location of synchronization operations, keeping clocks in synchrony (if used to provide common timing information), and the handling of multicast and broadcast messages. Coherent planning of the steps in the synchronization process, together with the necessary operations of the objects, e.g., decompression, must also be done. In addition, presentation manipulation operations demand additional replanning at runtime.

FIRST TERM EXAMINATION [SEPT. 2016]
THIRD SEMESTER [B.TECH]
COMPUTER GRAPHICS AND MULTIMEDIA
[ETCS–211]

Time : 1.5 Hrs. **M.M. : 30**

Note: *Attempt three questions in all. Question 1 is compulsory. All questions carry equal marks.*

Q.1. Attempt any five parts from the following. Each part carry equal marks. **(10)**

Q.1. (*a*) What is computer Graphics? List four major applications of computer graphics.

Ans. Computer graphics are pictures and films created using computers. Usually, the term refers to computer-generated image data created with help from specialized graphical hardware and software. It is a vast and recent area in computer science. The following are also considered graphics applications

Paint programs: Allow you to create rough frechand drawings. The images are stored as bit maps and can easily be edited

Ilustration/design programs: Supports more advanced features than paint programs, particularly for drawing curved lines. The images are usually stored in vector-based formats.

Animation software: Enables you to chain and sequence a series of images to simulate movement. Each image is like a frame in a movie.

Q.1.(*b*) What do you understand by Transformation?

Ans. Any image in a plane could be altered by using different operations, or transformations. Translation is when we slide a figure in any direction. In applications that work with 3-D graphics, geometrical transformations can be used to do the following.

Express the location of an object relative to another object.

Rotate and size objects.

Change viewing positions, directions, and perspectives.

Q.1.(*c*) What is rasterization?

Ans. Rasterisation (or rasterization) is the task of taking an image described in a vector graphics format (shapes) and converting it into a raster image (pixels or dots) for output on a video display or printer, or for storage in a bitmap file format. It refers to both rasterisation of models and 2D rendering primitives such as polygons, line segments, etc.

Q.1.(*d*) Write name of four input devices?

Ans. Following are some of the important input devices which are used in a computer—

Keyboard

Mouse

Joy Stick

Light pen

Q.1.(*e*) What do you mean by clipping in respect of computer graphics?

Ans. Clipping, in the context of computer graphics, is a method to selectively enable or disable rendering operations within a defined region of interest. Mathematically, clipping can be described using the terminology of constructive geometry.

Q.1.(*f*) What is vertical retrace of electron beam?

Ans. At the end of each scan line, the electron beam returns to the left side of the screen to begin displaying the next scan line. The return to the left of the screen, after

Scanned by CamScanner

refreshing each scan line, is called the horizontal retrace of the electron beam. And at the end of each frame (displayed in 1/80th to 1/60th of a second), the electron beam returns (vertical retrace)to the top left corner of the screen to begin the next frame.

Q.1.(g) Define the following. Aspect Ratio and Resolution

Ans. The aspect ratio of an image describes the proportional relationship between its width and its height. It is commonly expressed as two numbers separated by a colon, as in 16:9. For an x:y aspect ratio, no matter how big or small the image is, if the width is divided into x units of equal length and the height is measured using this same length unit, the height will be measured to be y units.

In computers, resolution is the number of pixels (individual points of color) contained on a display monitor, expressed in terms of the number of pixels on the horizontal axis and the number on the vertical axis.

Q.2. (a) Write and explain techniques for producing color displays with a CRT? **(3)**

Ans. The basic principle behind colored displays is that combining the 3 basic colors –Red, Blue and Green, can produce every color. By choosing different ratios of these three colors we can produce different colors – millions of them in-fact. We also have basic phosphors, which can produce these basic colors. So, one should have a technology to combine them in different combinations.

There are two popular techniques for producing color displays with a CRT are:

1. Beam-penetration method
2. Shadow-mask method

Beam Penetration method

This CRT is similar to the simple CRT, but it makes use of multi coloured phosphorus of number of layers. Each phosphorus layer is responsible for one colour. All other arrangements are similar to simple CRT. It can produce a maximum of 4 to 5 colours

The organization is something like this - The red, green and blue phosphorus are coated in layers - one behind the other. If a low speed beam strikes the CRT, only the red colored phosphorus is activated, a slightly accelerated beam would activate both red and green (because it can penetrate deeper) and a much more activated one would add the blue component also.

But the basic problem is a reliable technology to accelerate the electronic beam to precise levels to get the exact colors - it is easier said than done. However, a limited range of colors can be conveniently produced using the concept.

The Shadow - Mask method

This works, again, on the principle of combining the basic colors - Red, green and Blue - in suitable proportions to get a combination of colors, but it's principle is much more sophisticated and stable.

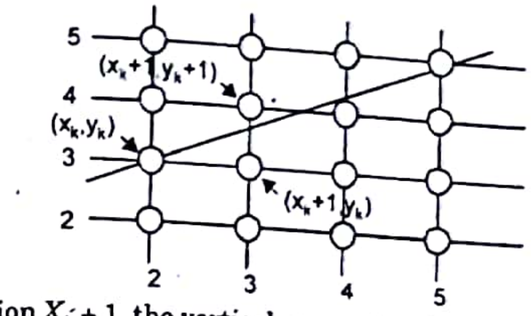
The shadow mask CRT, instead of using one electron gun, uses 3 different guns placed one by the side of the other to form a triangle or a "Delta". Each pixel point on the screen is also made up of 3 types of phosphors to produce red, blue and green colors. Just before the phosphor screen is a metal screen, called a "shadow mask".

Q.2.(b) Compute the infermediate point on the straight line using the Bresenham's algorithm, when the end points are given as input. **(3)**

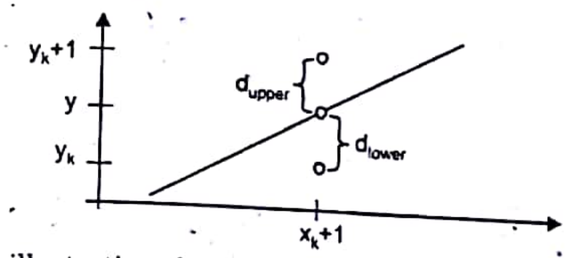
Ans. Bresenham's Line Generation

The Bresenham algorithm is another incremental scan conversion algorithm. The big advantage of this algorithm is that, it uses only integer calculations. Moving across the x axis in unit int ervals and at each step choose between two different y coordinates.

For example, as shown in the following illustration, from position (2, 3) you need to choose between (3, 3) and (3, 4). You would like the point that is closer to the original line.



At sample position $X_k + 1$, the vertical separations from the mathematical line are labelled as dupper and dlower.



From the above illustration, the y coordinate on the mathematical line at $x_k + 1$ is—

$$y = m(X_k + 1) + b$$

So, d_{upper} and d_{lower} are given as follows—

$$d_{lower} = y - y_k$$
$$= m(X_k + 1) + b - Y_k$$

and

$$d_{upper} = (y_k + 1) - y$$
$$= Y_k + 1 - m(X_k + 1) - b$$

You can use these to make a simple decision about which pixel is closer to the mathematical line. This simple decision is based on the difference between the two pixel positions.

$$d_{lower} - d_{upper} = 2m\,(x_k + 1) - 2y_k + 2b - 1$$

Let us substitute m with dy/dx where dx and dy are the differences between the end-points.

$$dx\,(d_{lower} - d_{upper}) = dx(2dydx(x_k + 1) - 2y_k + 2b - 1)$$
$$= 2dy.x_k - 2dx.y_k + 2dy + 2dx\,(2b - 1)$$
$$= 2dy.x_k - 2dx.yk + C$$

So, a decision parameter p_k for the kth step along a line is given by—

$$p_k = dx(d_{lower} - d_{upper})$$
$$= 2dy.x_k - 2dx.y_k + C$$

The sign of the decision parameter p_k is the same as that of $d_{lower} - d_{upper}$.

If p_k is negative, then choose the lower pixel, otherwise choose the upper pixel.

Remember, the coordinate changes occur along the x axis in unit steps, so you can do everything with integer calculations. At step $k + 1$, the decision parameter is given as—

$$p_k + 1 = 2dy.x_k + 1 - 2dx.y_k + 1 + C$$

Subtracting pk from this we get—

$$p_k + 1 - p_k = 2dy(x_k + 1 - x_k) - 2dx(y_k + 1 - y_k)$$

But, $x_k + 1$ is the same as $(x_k) + 1$, So—

$$p_k + 1 = p_k + 2dy - 2dx(y_k + 1 - y_k)$$

Where, $y_k + 1 - y_k$ is either 0 or 1 depending on the sign of p_k.

Scanned by CamScanner

The first decision parameter p_0 is evaluated at (x_0, y_0) is given as—

$$p_0 = 2dy - dx$$

Now, keeping in mind all the above points and calculations, here is the Bresenham algorithm for slope $m < 1$:-

Step 1—Input the two end-points of line, storing the left end-point in (x_0, y_0).

Step 2—Plot the point (x_0, y_0).

Step 3—Calculate the constants dx, dy, $2dy$, and $(2dy - 2dx)$ and get the first value for the decision parameter as—

$$p_0 = 2dy - dx$$

Step 4—At each x_k along the line, starting at $k = 0$, perform the following test—

If $p_k < 0$, the next point to plot is $(x_k + 1, y_k)$ and

$$p_k + 1 = p_k + 2dy$$

Otherwise,

$$(x_k, y_k + 1)$$

$$p_k + 1 = p_k + 2dy - 2dx$$

Step 5—Repeat step 4 $(dx - 1)$ times.

For $m > 1$, find out whether you need to increment x while incrementing y each time.

After solving, the equation for decision parameter p_k will be very similar, just the x and y in the equation gets interchanged.

Q.2.(c) Explain the procedure to clip three-dimensional planes against a regular parallelepiped by extending Sutherland-Hodgeman algorithm for polygon clipping. **(4)**

Ans. A polygon can also be clipped by specifying the clipping window. Sutherland Hodgeman polygon clipping algorithm is used for polygon clipping. In this algorithm, all the vertices of the polygon are clipped against each edge of the clipping window.

First the polygon is clipped against the left edge of the polygon window to get new vertices of the polygon.While processing an edge of a polygon with clipping window, an intersection point is found if edge is not completely inside clipping window and the a partial edge from the intersection point to the outside edge is clipped. The following figures show left, right, top and bottom edge clippings

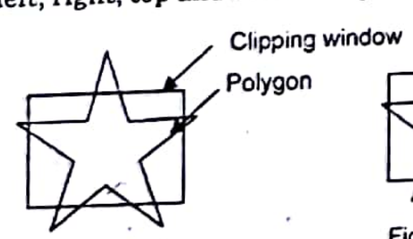


Figure. Clipping left edge

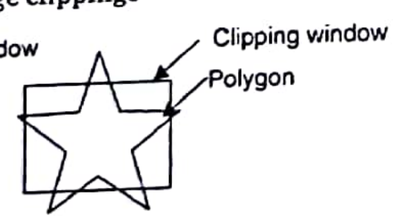


Figure: Clipping right edge

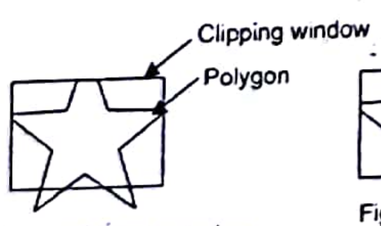


Figure: Clipping top edge

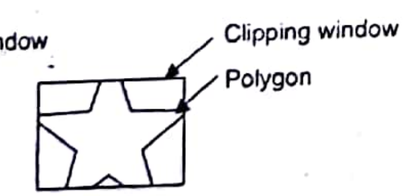


Figure: Clipping bottom edge

Q.3. (a) List the basic transformation techniques. What are their respective mathematical and matrix representations? **(4)**

Ans. Rotation

3D rotation is not same as 2D rotation. In 3D rotation, we have to specify the angle of rotation along with the axis of rotation. We can perform 3D rotation about X, Y, and Z axes. They are represented in the matrix form as below—

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling

You can change the size of an object using scaling transformation. In the scaling process, you either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object with the scaling factor to get the desired result. In 3D scaling operation, three coordinates are used. Let us assume that the original coordinates are (X, Y, Z), scaling factors are (SX, SY, Sz) respectively, and the produced coordinates are (X', Y', Z'). This can be mathematically represented as shown below—

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = P.S$$

$$[X'\ Y'\ Z'\ 1] = [X\ Y\ Z\ 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [X.S_x\ Y.S_y\ Z.S_z\ 1]$$

Shear

A transformation that slants the shape of an object is called the shear transformation. Like in 2D shear, we can shear an object along the X-axis, Y-axis, or Z-axis in 3D.

$$Sh = \begin{bmatrix} 1 & sh_x^y & sh_x^z & 0 \\ sh_y^x & 1 & sh_y^z & 0 \\ sh_z^x & sh_z^y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned by CamScanner

$$P' = P.Sh$$

$$X' = X + Sh_z^y Y + Sh_y^z Z$$

$$Y' = Sh_y^x X + Y + Sh_y^z Z$$

$$Z' = Sh_z^x X + Sh_z^y Y + Z$$

Translation

In 3D translation, we transfer the Z coordinate along with the X and Y coordinates. The process for translation in 3D is similar to 2D translation. A translation moves an object into a different position on the screen. A point can be translated in 3D by adding translation coordinate (tx, ty, tz) to the original coordinate (X, Y, Z) to get the new coordinate (X', Y', Z').

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

$$P' = P.T$$

$$[X'\ Y'\ Z'\ 1] = [X\ Y\ Z\ 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

$$= [X + t_x\ Y + t_y\ Z + t_z\ 1]$$

3D vectors are expressed as coordinate triples that define a relative location.

Defining a coordinate system requires four 3D vectors:

1 to locate the origin

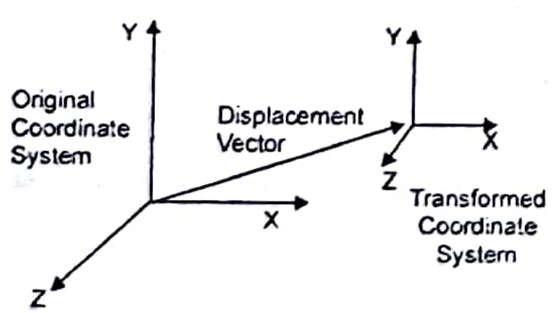
1 X unit vector

1 Y unit vector

1 Z unit vector

A new coordinate system can therfore be expressed via a 4 × 3 transforamtion matrix

Homogeneous transforms or 4 × 4 matrices are sometimes used



Multiple transforms can be expressed by multiplying matrices for the component tranforms together, so that any number of nested transforms can ultimately be collapsed into a single transformation matrix. Hardware can do matrix multiplications very quickly.

Q.3. (b) Distinguish between the Raster Scan System and Random Scan System. **(3)**

Ans.

Raster Scan System	Random Scan System
It has poor to less Resolution because picture definition is stored as a intensity value.	It has High Resolution becuase it stores picture definition as a set of line commands.
Electron Beam is directed from top to bottom and one row at a time on screen, but electron beam is directed to whole screen.	Electron Beam is directed to only that part of screen where pciture is required to be drawn, one line at a time so also called Vector Display.
It is less expensive than Random Scan Sytem.	It is costlier than Raster Scan System.
Refresh rate is 60 to 80 frame per second.	Refresh Rate depends on the number of lines to be displayed i.e. 30 to 60 times per second.
It Stores picture definition in Refresh Buffer also called Frame Buffer.	It Stores picture definition as a set of line commands called Refresh Display File.
Zig-Zag line is produced because plotted value are discrete.	Smooth line is produced because directly the line path is followed by electron beam.
It contains shadow, advance shading and hidden surface technique so gives the realistic display of scenes.	It does not contain shadow and hidden surface technique so it can not give realistic display of scenes.
It uses Pixels along scan lines for drawing an image.	It is designed for line drawing applications and uses various mathematical function to draw.

Q.3. (c) Digitize the line with end-points (20, 10) and (30,18) using Bresenham's Algorithm. **(3)**

Ans. Bresenham Line Algorithm

Decision parameter

$$d_i = (s_i - t_i)$$

Example: Draw a line from (20, 10) to (30, 18)

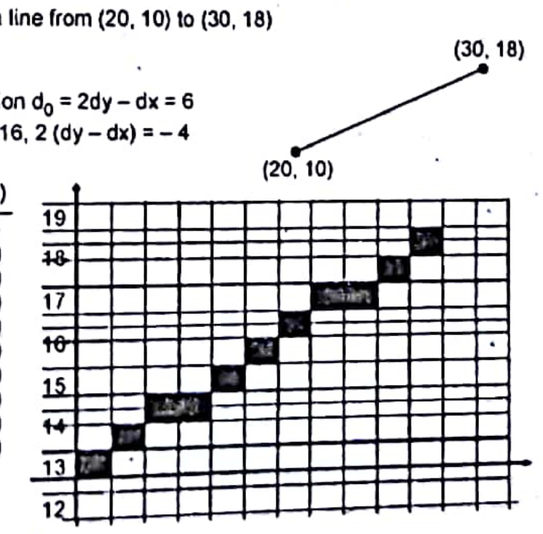
dx = 10

dy = 8

initial decision d_0 = 2dy – dx = 6

Also 2dy = 16, 2 (dy – dx) = – 4

i	d_i	(x_{j+1}, y_{j+1})
0	6	(21, 11)
1	2	(22, 12)
2	–2	(23, 12)
3	14	(24, 13)
4	10	(25, 14)
5	6	(26, 15)
6	2	(27, 16)
7	–2	(28, 16)
8	14	(29, 17)
9	10	(30, 18)



Scanned by CamScanner

- If $d_i < 0$, then closest pixel is below true line (s_i smaller)
- If $d_i \geq 0$, then closest pixel is above true line (t_i smaller)
- We must calculate the new values for d_i as we move along the line.

Q.4. (a) A square with vertices A(1,1),B(3,1) C(3,3), D(1,3) is scaled with scaling factor 0.5 After scaling the resultant geometry is also centered at same center. Find out the transformation matrix and co-ordinates of geometry after transformation. (4)

Ans. This transformation can be carried out in the following steps.

1. Translate the square so that its center coincides with the origin.
2. Scale the square with respect to the origin.
3. Translate the sqaure back to the original position.

Thus, the overall transformation matrix is formed by multiplication of three matrices.

$$\therefore \quad T_1.S.T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & -2 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} A' \\ B' \\ C' \\ D' \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 3 & 1 & 1 \\ 3 & 3 & 1 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1.5 & 1.5 & 1 \\ 2.5 & 1.5 & 1 \\ 2.5 & 2.5 & 1 \\ 1.5 & 2.5 & 1 \end{bmatrix}$$

Q.4. (b) A trangle A(2,3), B(5,5) C(4,3) is rotated by 45° about a point P(1,1). Find out the co-ordinates of resultant geometry.

Ans. We have:

$$T_1.R.T_2 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_p\cos\theta + y_p\sin\theta + x_p & -x_p\sin\theta - y_p\cos\theta + y_p & 1 \end{bmatrix}$$

Here, $\theta = 45°$, $x_p = 1$ and $y_p = 1$. Substituting values we get

$$T_1.R.T_2 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1 & -\sqrt{2}+1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 \\ 5 & 5 & 1 \\ 4 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1 & -\sqrt{2}+1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{1}{\sqrt{2}}+1 & \frac{3}{\sqrt{2}}+1 & 1 \\ 1 & \frac{8}{\sqrt{2}}+1 & 1 \\ \frac{1}{\sqrt{2}}+1 & \frac{5}{\sqrt{2}}+1 & 1 \end{bmatrix}$$

Q.4. (c) What are B-spline Curves? What are advantages of B-Splines over Bezier curves? Differentiate between interpolation and approximation in spline representations. (3)

Ans. In the mathematical subfield of numerical analysis, a B-spline, or basis spline, is a spline function that has minimal support with respect to a given degree, smoothness, and domain partition. Any spline function of given degree can be expressed as a linear combination of B-splines of that degree.

The Advantage of Using B-spline Curves

B-spline curves require more information (*i.e.*, the degree of the curve and a knot vector) and a more complex theory than Bézier curves. But, it has more advantages to offset this shortcoming. First, a B-spline curve can be a Bézier curve. Second, B-spline curves satisfy all important properties that Bézier curves have. Third, B-spline curves provide more control flexibility than Bézier curves can do. For example, the degree of a B-spline curve is separated from the number of control points. More precisely, we can use lower degree curves and still maintain a large number of control points. We can change the position of a control point without globally changing the shape of the whole curve (local modification property). Since B-spline curves satisfy the strong convex hull property, they have a finer shape control. Moreover, there are other techniques for designing and editing the shape of a curve such as changing knots.

However, keep in mind that B-spline curves are still polynomial curves and polynomial curves cannot represent many useful simple curves such as circles and ellipses. Thus, a generalization of B-spline, NURBS, is required

A spline is a section of a curve used to make the final curve. A spline is represented by control points.

Interpolation spline: When the curve passes through all the control points then it is called as Interpolation spline.

Approximation Spline: When the curve does not passes through the control points then it is called as approximation spline.

Scanned by CamScanner

END TERM EXAMINATION [DEC. 2016]
THIRD SEMESTER [B.TECH]
COMPUTER GRAPHICS AND MULTIMEDIA
[ETCS–211]

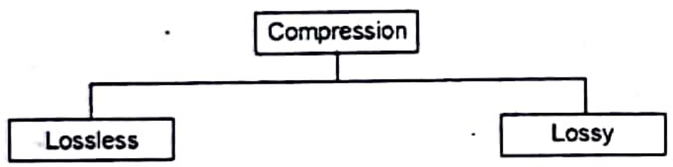
Time : 3 Hrs. **M.M. : 75**

Note: *Attempt any five questions. All questions carry equal marks.*

Q.1.(a) Explain the significance of various data compression techniques in detail with the help of an example for each technique. **(7)**

Ans. Data compression is the function of presentation layer in OSI reference model. Compression is often used to maximize the use of bandwidth across a network or to optimize disk space when saving data. There are two general types of compression algorithms:

1. Lossless compression
2. Lossy compression



Types of compression

Lossless Compression

Lossless compression compresses the data in such a way that when data is decompressed it is exactly the same as it was before compression i.e. there is no loss of data.

A lossless compression is used to compress file data such as executable code, text files, and numeric data, because programs that process such file data cannot tolerate mistakes in the data.

Lossless compression will typically not compress file as much as lossy compression techniques and may take more processing power to accomplish the compression.

Lossless Compression Algorithms

The various algorithms used to implement lossless data compression are :

1. Run length encoding
2. Differential pulse code modulation
3. Dictionary based encoding

1. Run length encoding

• This method replaces the consecutive occurrences of a given symbol with only one copy of the symbol along with a count of how many times that symbol occurs. Hence the names 'run length'.

• For example, the string AAABBCDDDD would be encoded as 3A2BIC4D.

• A real life example where run-length encoding is quite effective is the fax machine. Most faxes are white sheets with the occasional black text. So, a run-length encoding scheme can take each line and transmit a code for while then the number of pixels, then the code for black and the number of pixels and so on.

• This method of compression must be used carefully. If there is not a lot of repetition in the data then it is possible the run length encoding scheme would actually increase the size of a file.

2. Differential pulse code modulation

• In this method first a reference symbol is placed. Then for each symbol in the data, we place the difference between that symbol and the reference symbol used.

• For example, using symbol A as reference symbol, the string AAABBC DDDD would be encoded as AOOOl123333, since A is the same as reference symbol, B has a difference of 1 from the reference symbol and so on.

3. Dictionary based encoding

• One of the best known dictionary based encoding algorithms is Lempel-Ziv (LZ) compression algorithm.

• This method is also known as substitution coder.

• In this method, a dictionary (table) of variable length strings (common phrases) is built.

• This dictionary contains almost every string that is expected to occur in data.

• When any of these strings occur in the data, then they are replaced with the corresponding index to the dictionary.

• In this method, instead of working with individual characters in text data, we treat each word as a string and output the index in the dictionary for that word.

• For example, let us say that the word "compression" has the index 4978 in one particular dictionary; it is the 4978th word is usr/share/dict/words. To compress a body of text, each time the string "compression" appears, it would be replaced by 4978.

Lossy Compression

Lossy compression is the one that does not promise that the data received is exactly the same as data send i.e. the data may be lost.

This is because a lossy algorithm removes information that it cannot later restore.

Lossy algorithms are used to compress still images, video and audio.

Lossy algorithms typically achieve much better compression ratios than the lossless algorithms.

Audio Compression

• Audio compression is used for speech or music.

• For speech, we need to compress a 64-KHz digitized signal; For music, we need to compress a 1.411.MHz signal

• Two types of techniques are used for audio compression:

1. Predictive encoding
2. Perceptual encoding

Techniques of Audio Compression

Predictive encoding

• In predictive encoding, the differences between the samples are encoded instead of encoding all the sampled values.

• This type of compression is normally used for speech.

• Several standards have been defined such as GSM (13 kbps), G. 729 (8 kbps), and G.723.3 (6.4 or 5.3 kbps).

Perceptual encoding

• Perceptual encoding scheme is used to create a CD-quality audio that requires a transmission bandwidth of 1.411 Mbps.

• MP3 (MPEG audio layer 3), a part of MPEG standard uses this perceptual encoding.

• Perceptual encoding is based on the science of psychoacoustics, a study of how people perceive sound.

• The perceptual encoding exploits certain flaws in the human auditory system to encode a signal in such a way that it sounds the same to a human listener, even if it looks quite different on an oscilloscope.

• The key property of perceptual coding is that some sounds can mask other sound. For example, imagine that you are broadcasting a live flute concert and all of a sudden someone starts striking a hammer on a metal sheet. You will not be able to hear the flute any more. Its sound has been masked by the hammer.

Scanned by CamScanner

• Such a technique explained above is called frequency masking-the ability of a loud sound in one frequency band to hide a softer sound in another frequency band that would have been audible in the absence of the loud sound.

• Masking can also be done on the basis of time. For example: Even if the hammer is not striking on a metal sheet, the flute will be inaudible for a short period of time because the ears turn down its gain when they start and take a finite time to turn up again.

• Thus, a loud sound can numb our ears for a short time even after the sound has stopped. This effect is called temporal masking.

MP3

• MP3 uses these two phenomena, i.e. frequency masking and temporal masking to compress audio signals.

• In such a system, the technique analyzes and divides the spectrum into several groups. Zero bits are allocated to the frequency ranges that are totally masked.

• A small number of bits are allocated to the frequency ranges that are partially masked.

• A larger number. of bits are allocated to the frequency ranges that are not masked.

• Based on the range of frequencies in the original analog audio, MP3 produces three data rates: 96kbps, 128 kbps and 160 kbps.

Q.1. (*b*) How is A-Buffer method different from Z-Buffer method? Highlight their roles in back face detection. (8)

Ans. There are two approaches for removing hidden surface problems " Object-Space method and Image-space method. The Object-space method is implemented in physical coordinate system and image-space method is implemented in screen coordinate system.

When we want to display a 3D object on a 2D screen, we need to identify those parts of a screen that are visible from a chosen viewing position.

Depth Buffer (Z-Buffer) Method

This method is developed by Cutmull. It is an image-space approach. The basic idea is to test the Z-depth of each surface to determine the closest (visible) surface.

In this method each surface is processed separately one pixel position at a time across the surface. The depth values for a pixel are compared and the closest (smallest *z*) surface determines the color to be displayed in the frame buffer.

It is applied very efficiently on surfaces of polygon. Surfaces can be processed in any order. To override the closer polygons from the far ones, two buffers named frame buffer and depth buffer, are used.

Depth buffer is used to store depth values for (x, y) position, as surfaces are processed ($0 \le \text{depth} \le 1$).

The frame buffer is used to store the intensity value of color value at each position (x, y).

The *z*-coordinates are usually normalized to the range [0, 1]. The 0 value for *z*-coordinate indicates back clipping pane and 1 value for *z*-coordinates indicates front clipping pane.

Advantages

It is easy to implement.

It reduces the speed problem if implemented in hardware.

It processes one object at a time.

Disadvantages

It requires large memory.

It is time consuming process.

A-Buffer Method

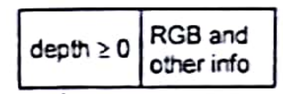
The A-buffer method is an extension of the depth-buffer method. The A-buffer method is a visibility detection method developed at Lucas film Studios for the rendering system Renders Everything You Ever Saw (REYES).

The A-buffer expands on the depth buffer method to allow transparencies. The key data structure in the A-buffer is the accumulation buffer.

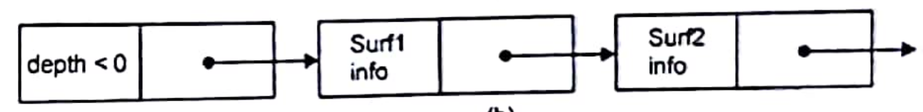
Each position in the A-buffer has two fields—

Depth field—It stores a positive or negative real number

Intensity field—It stores surface-intensity information or a pointer value



(a)



(b)

If depth >= 0, the number stored at that position is the depth of a single surface overlapping the corresponding pixel area. The intensity field then stores the RGB components of the surface color at that point and the percent of pixel coverage.

If depth < 0, it indicates multiple-surface contributions to the pixel intensity. The intensity field then stores a pointer to a linked list of surface data. The surface buffer in the A-buffer includes—

RGB intensity components

Opacity Parameter

Depth

Percent of area coverage

Surface identifier

The algorithm proceeds just like the depth buffer algorithm. The depth and opacity values are used to determine the final color of a pixel.

Back-Face Detection

A fast and simple object-space method for identifying the back faces of a polyhedron is based on the "inside-outside" tests. A point (x, y, z) is "inside" a polygon surface with plane parameters A, B, C, and D if When an inside point is along the line of sight to the surface, the polygon must be a back face (we are inside that face and cannot see the front of it from our viewing position).

We can simplify this test by considering the normal vector N to a polygon surface, which has Cartesian components (A, B, C).

In general, if V is a vector in the viewing direction from the eye (or "camera") position, then this polygon is a back face if

$$V.N > 0$$

Furthermore, if object descriptions are converted to projection coordinates and your viewing direction is parallel to the viewing z-axis, then—

$$V = (0, 0, Vz) \text{ and } V.N = VZC$$

So that we only need to consider the sign of C the component of the normal vector N.

In a right-handed viewing system with viewing direction along the negative ZV axis, the polygon is a back face if $C < 0$. Also, we cannot see any face whose normal has *z* component $C = 0$, since your viewing direction is towards that polygon. Thus, in general, we can label any polygon as a back face if its normal vector has a z component value—

$$C <= 0$$

Q.2. Write short notes on any three the following: (5×3=15)

Ans. (*a*) MIDI

MIDI (Musical Instrument Digital Interface) is a protocol designed for recording and playing back music on digital synthesizers that is supported by many makes of

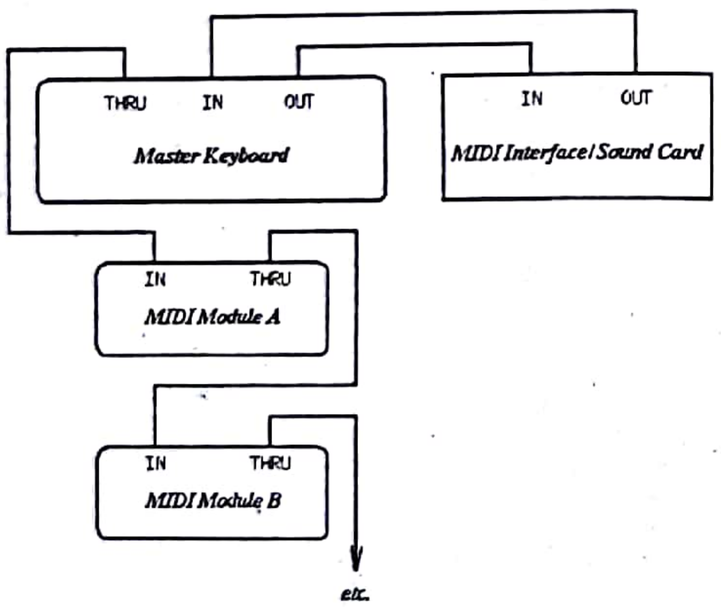
Scanned by CamScanner

personal computer sound cards. Originally intended to control one keyboard from another, it was quickly adopted for the personal computer. Rather than representing musical sound directly, it transmits information about how music is produced. The command set includes note-ons, note-offs, key velocity, pitch bend and other methods of controlling a synthesizer. The sound waves produced are those already stored in a wavetable in the receiving instrument or sound card.

Since a MIDI file only represents player information, it is far more concise than formats that the sound directly. An advantage is very small file size. A disadvantage is the lack of specific sound control.

With a program that provides this interface, you can create music using a standard keyboard or other input device. You or others can then play your MIDI-conforming creation with the same or another program and a sound card as a music synthesizer. The MIDI program may come with a graphical user interface that looks like a sound studio control room. Many sound cards come as a package with MIDI software (for example, Media Vision's Pro Audio Studio 16).

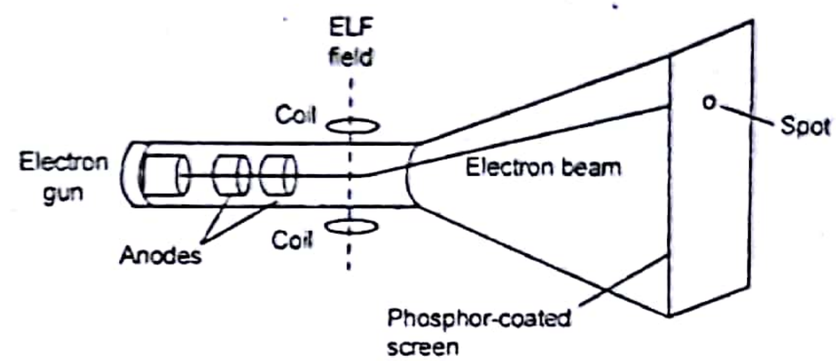
The MIDI protocol uses eight-bit serial transmission with one start bit and one stop bit, has a 31.25 Kbs data rate, and is asynchronous. Connection is made through a five-pin DIN plug, of which three pins are used.



Q.2. (*b*) CRT

Ans. A cathode ray tube consists of several basiccomponents, as illustrated below. The electron gun generates anarrow beam of electrons. The anodes accelerate the electrons.Deflecting coils produce an extremely low frequency electromagnetic field that allowsfor constant adjustment of the direction of the electron beam.There are two sets of deflecting coils: horizontal and vertical.(In the illustration, only one set of coils is shown forsimplicity.) The intensity of the beam can be varied. Theelectron beam produces a tiny, bright visible spot when itstrikes the phosphor-coated screen.

To produce an image on the screen, complexsignals are applied to the deflecting coils, and also to theapparatus that controls the intensity of the electron beam. Thiscauses the spot to race across the screen from right to left, andfrom top to bottom, in a sequence of horizontal lines called theraster. As viewed from the front of the CRT, the spot moves in apattern similar to the way your eyes move when you read asingle-column page of text. But the scanning takes place at sucha rapid rate that your eye sees a constant image over the entirescreen.



This causes the spot to race across the screen from right to left, andfrom top to bottom, in a sequence of horizontal lines called theraster. As viewed from the front of the CRT, the spot moves in apattern similar to the way your eyes move when you read asingle-column page of text. But the scanning takes place at sucha rapid rate that your eye sees a constant image over the entirescreen.

The illustration shows only one electron gun.This is typical of a monochrome, or single-color, CRT. However,virtually all CRTs today render color images. These devices havethree electron guns, one for the primary color red, one for theprimary color green, and one for the primary color blue. The CRTthus produces three overlapping images: one in red (R), one ingreen (G), and one in blue (B). This is the so-called RGB colormodel. today render color images. These devices havethree electron guns, one for the primary color red, one for theprimary color green, and one for the primary color blue. The CRTthus produces three overlapping images: one in red (R), one ingreen (G), and one in blue (B). This is the so-called RGB colormodel.

Q.2. (*c*) Specular reflection

Ans. Specular reflection, also known as regular reflection is the mirror-like reflection of waves, such as light, from a surface. In this process, each incident ray is reflected, with the reflected ray having the same angle to the surface normal as the incident ray. The result is that an image reflected by the surface is reproduced in mirror-like (specular) fashion. The law of reflection states that for each incident ray the angle of incidence equals the angle of reflection, and the incident, normal, and reflected directions are coplanar. This behavior was first described by Hero of Alexandria When light hits a surface, there are three possible outcomes. Light may be absorbed by the material, light may be transmitted through the surface, or light may be reflected back. Materials often show some mix of these behaviors, with the proportion of light that goes to each depending on the properties of the material, the wavelength of the light, and the angle of incidence. For most interfaces between materials, the fraction of the light that is reflected increases with increasing angle of incidence 0 .

Reflected light can be divided into two sub-types, specular reflection and diffuse reflection. Specular reflection reflects all light at the same angle, whereas diffuse reflection reflects in a broad range of directions. An example of the distinction between specular and diffuse reflection would be glossy and matte paints. Matte paints have almost exclusively diffuse reflection, while glossy paints have both specular and diffuse reflection. A surface built from a non-absorbing powder, such as plaster, can be a nearly perfect diffuser, whereas polished metallic objects can specularly reflect light very efficiently. The reflecting material of mirrors is usually aluminum or silver.

Q.2. (*d*) Huffman coding

Ans. Huffman coding is based on the frequency of occurance of a data item (pixel in images). The principle is to use a lower number of bits to encode the data that occurs more frequently. Codes are stored in a Code Book which may be constructed for each image or a set of images. In all cases the code book plus encoded data must be transmitted to enable decoding.

The Huffman algorithm is now briefly summarised:

Scanned by CamScanner

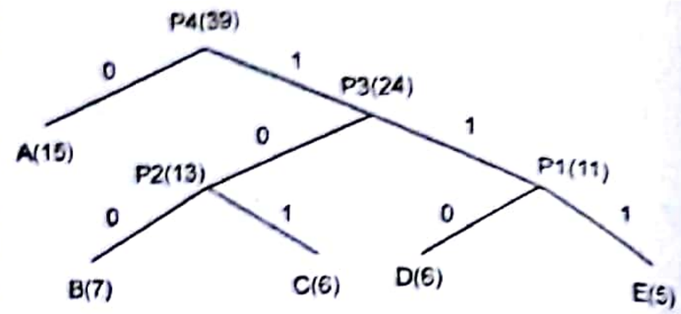
A bottom-up approach

1 Initialization: Put all nodes in an OPEN list, keep it sorted at all times

(*e.g.*, ABCDE).

2 Repeat until the OPEN list has only one node left:

(*a*) From OPEN pick two nodes having the lowest frequencies/probabilities, create a parent node of them.



(*b*) Assign the sum of the children's frequencies/probabilities to the parent node and insert it into OPEN.

(*c*) Assign code 0, 1 to the two branches of the tree, and delete the children from OPEN.

Symbol	Count	log(1/p)	Code	Subtotal (# of bits)
A	15	1.38	0	15
B	7	2.48	100	21
C	6	2.70	101	18
D	6	2.70	110	18
E	5	2.96	111	15
TOTAL (# of bits):				87

The following points are worth noting about the above algorithm:

Decoding for the above two algorithms is trivial as long as the coding table (the statistics) is sent before the data. (There is a bit overhead for sending this, negligible if the data file is big.)

Unique Prefix Property: no code is a prefix to any other code (all symbols are at the leaf nodes) -> great for decoder, unambiguous.

Q.3. (*a*) Highlight the various shading models with the help of an example for each one. (8)

Ans. Shading Models

We often use polygons to simulate curved surfaces. If these cases we want the colours of the polygons to flow smoothly into each other.

flat shading
goraud shading (color interpolation shading)
phong shading (normal interpolation shading)

Flat Shading

each entire polygon is drawn with the same colour
need to know one normal for the entire polygon
fast
lighting equation used once per polygon

Given a single normal to the plane the lighting equations and the material properties are used to generate a single colour. The polygon is filled with that colour.

Goraud Shading

colours are interpolated across the polygon
need to know a normal for each vertex of the polygon
slower than flat shading
lighting equation used at each vertex

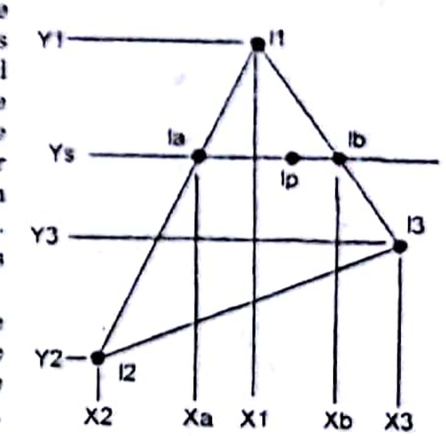
Given a normal at each vertex of the polygon, the colour at each vertex is determined from the lighting equations and the material properties. Linear interpolation of the

colour values at each vertex are used to generate colour values for each pixel on the edges. Linear interpolation across each scan line is used to then fill in the colour of the polygon.

Phong Shading

normals are interpolated across the polygon
need to know a normal for each vertex of the polygon
better at dealing with highlights than Goraud shading
slower than Goraud shading
lighting equation used at each pixel

Where Goraud shading uses normals at the vertices and then interpolates the resulting colours across the polygon, Phong shading goes further and interpolates tha normals. Linear interpolation of the normal values at each vertex are used to generate normal values for the pixels on the edges. Linear interpolation across each scan line is used to then generate normals at each pixel across the scan line.



Whether we are interpolating normals or colours the procedure is the same

To find the intensity of Ip, we need to know the intensity of Ia and Ib. To find the intensity of Ia we need to know the intensity of I1 and I2. To find the intensity of Ib we need to know the intensity of I1 and I3.

$I_a = (Y_s - Y2) / (Y1 - Y2) * I1 + (Y1 - Ys) / (Y1 - Y2) * I2$

$I_b = (Y_s - Y3) / (Y1 - Y3) * I1 + (Y1 - Ys) / (Y1 - Y3) * I3$

$I_p = (X_b - X_p) / (X_b - Xa) * Ia + (Xp - Xa) / (X_b - Xa) * I_b$

Q.3.(*b*) How are the parallel projections different from the perspective projections? Briefly describe their sub types. (7)

Ans. Projections transform points in *n*-space to *m*-space, where $m < n$. In 3D, we map points from 3-space to the projection plane (PP) along projectors emanating from the center of projection (COP)

There are two basic types of projections:

Perspective

• distance from COP to PP finite. Parallel Projection: In this projection, *z* coordinate is eliminated and parallel lines from each vertex on the object are extended until they intersect view plane. These points of intersections are the projections of vertices. Then, these projections of these vertices are joined by line segments. Types: Orthographic and oblique projection.

Parallel

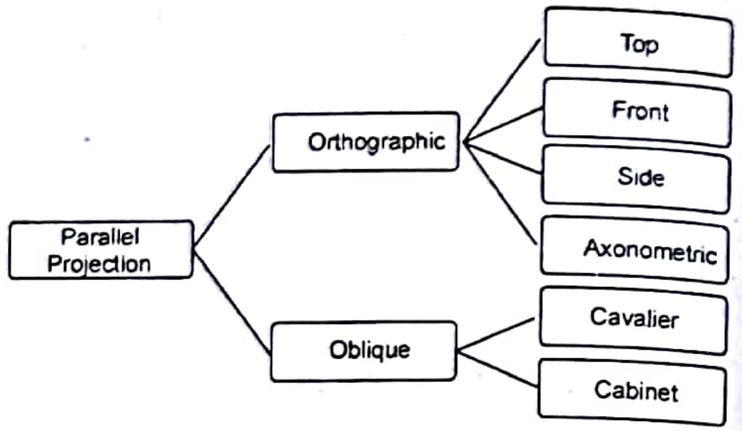
• distance from COP to PP infinite. In this projection lines of projections are not parallel and they converge at a point called as the center of projection. In this projection realistic views are produced but relative proportions are not preserved.

Parallel vs Perspective Projection

Perspective projection is seeing things larger when they're up close and smaller at a distance. It is a three-dimensional projection of objects on a two-dimensional medium such as paper. It allows an artist to produce a visual reproduction of an object which resembles the real one.

The center of projection in a perspective projection is a point which is at a distance from the viewer or artist. Objects located at this point appear smaller and will appear bigger when they are drawn closer to the viewer. Perspective projection produces a more realistic and detailed representation of an object allowing artists to create scenes that closely resemble the real thing. The other type of projection which is also used aside from perspective projection is parallel projection.

Scanned by CamScanner



Parallel projection, on the other hand, resembles seeing objects which are located far from the viewer through a telescope. It works by making light rays entering the eyes parallel, thus, doing away with the effect of depth in the drawing. Objects produced using parallel projection do not appear larger when they are near or smaller when they are far. It is very useful in architecture. However, when measurements are involved, perspective projection is best.

It provides an easier way of reproducing objects on any medium while having no definite center of projection. When it is not possible to create perspective projection, especially in cases where its use can cause flaws or distortions, parallel projection is used.

Orthographic Projection

In orthographic projection the direction of projection is normal to the projection of the plane. There are three types of orthographic projections—

Front Projection

Top Projection

Side Projection

Oblique Projection

In orthographic projection, the direction of projection is not normal to the projection of plane. In oblique projection, we can view the object better than orthographic projection.

There are two types of oblique projections — Cavalier and Cabinet. The Cavalier projection makes 45° angle with the projection plane. The projection of a line perpendicular to the view plane has the same length as the line itself in Cavalier projection. In a cavalier projection, the foreshortening factors for all three principal directions are equal.

The Cabinet projection makes 63.4° angle with the projection plane. In Cabinet projection, lines perpendicular to the viewing surface are projected at ½ their actual length

Isometric Projections

Orthographic projections that show more than one side of an object are called axonometric orthographic projections. The most common axonometric projection is an isometric projection where the projection plane intersects each coordinate axis in the model coordinate system at an equal distance. In this projection parallelism of lines are preserved but angles are not preserved

Perspective Projection

In perspective projection, the distance from the center of projection to project plane is finite and the size of the object varies inversely with distance which looks more realistic.

The distance and angles are not preserved and parallel lines do not remain parallel. Instead, they all converge at a single point called center of projection or projection reference point. There are 3 types of perspective projections which are shown in the following chart.

One point perspective projection is simple to draw.

Two point perspective projection gives better impression of depth.

Three point perspective projection is most difficult to draw.

Q.4.(a) Are any illumination models needed in computer graphics? If yes, then name them and elaborate the same. **(7)**

Ans. Yes, illumination models are required.

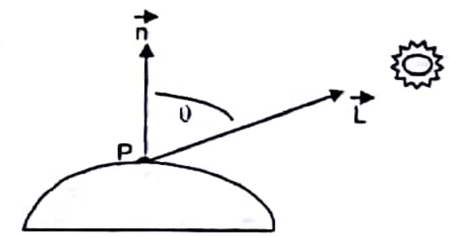
Surfaces in real world environments receive light in 3 ways:

1. Directly from existing light sources such as the sun or a lit candle
2. Light that passes and refracts through transparent objects such as water or a glass vase
3. Light reflected, bounced, or diffused from other exisiting surfaces in the environment

Local Illumination

Material Models

Diffuse illumination



Lambert's cosine law of reflection as shown in the above diagram:

1. n; a normal vector to the surface to be illuminated.
2. L, a vector from the surface position that points towards the light source.
3. Il, an intensity for a point light source.
4. kd, a diffuse reflection constant.

Equation gives the brightness of a surface point in terms of the brightness of a light source and its orientation relative to the surface normal vector, n,

$$I = I_i k_d \hat{n} \cdot \hat{L} = I_i k_d \cos\theta \qquad 0 \le \theta \le \pi/2$$

- I is the reflected intensity

Measures how bright the surface is at that point.

- Surface brightness varies as a function of the angle between n and L

When n and L coincide, the light source is directly overhead.

- I is at a maximum and cos θ = 1.

As the angle increases to 90°, the cosine decreases the intensity to 0.

- All the quantities in the equation are normalized between 0 and 1.
- I is converted into frame buffer intensity values by multiplying by the number of shades available

Ambient illumination

- Simple illuminated model is unable to directly accommodate all scattered light
- It is grouped together as independent intensity, Ia.
- The formula becomes

$$I = I_a k_a + I_i k_d \cos 0 \quad 0 < 0 < \pi/2$$

Iaka is the ambient illumination term, taking into account the additional environmental illumination, Ia, and the ability of the object to absorb it, ka.

Specular Highlights – (Phong Reflection Model)

- Regions of significant brightness, exhibited as spots or bands, characterize objects that specularly reflect light.
- Specular highlights originate from smooth, sometimes mirrorlike surfaces
- Fresnel equation is used to simulate this effect.
- The Fresnel equation states that for a perfectly reflecting surface the angle of incidence equals the angle of reflection.

Ii, specular = $W(0)\, I_i \cos^r\alpha$)

W(0) = specular reflection coefficient

n = specular reflection exponent

Scanned by CamScanner

Q.4. (b) Explain in detail the difference between DDA and Bresenham's line drawing algorithm with example. **(8)**

Ans. Digital Differential Analyzer

Line Drawing Algorithm	Bresenhams Line Drawing Algorithm
DDA algorithm can draw circles and curves but that are not as accurate as Bresenhams algorithm.	Bresenhams algorithm can draw circles and curves with much more accuracy than DDA algorithm.
DDA algorithm round off the coordinates to integer that is nearest to the line.	Bresenhams algorithm does not round off but takes the incremental value in its operation.
DDA algorithm uses an enormous number of floating-point multiplications so it is expensive.	Bresenhams algorithm is less expensive than DDA algorithm as it uses only addition and subtraction.
DDA algorithm uses floating points i.e. Real Arithmetic.	Bresenhams algorithm uses fixed points i.e. Integer Arithmetic.
DDA algorithm uses multiplication and division in its operations.	Bresenhams algorithm uses only subtraction and addition in its operations.
DDA algorithm is rather slowly than Bresenhams algorithm in line drawing because it uses real arithmetic (floating-point operations).	Bresenhams algorithm is faster than DDA algorithm in line drawing because it performs only addition and subtraction in its calculation and uses only integer arithmetic so it runs significantly faster
DDA algorithm is not as accurate and efficient as Bresenham algorithm.	Bresenhams algorithm is more efficient and much accurate than DDA algorithm.

Consider the line from (0, 0) to (-6, -6). Use the simple DDA algorithm line.

Evaluating steps 1 to 5 in the DDA algorithm we have

$X1 = 0, Y1 = 0$

$X2 = -6\ Y2 = -6$

Length $= |X2 - X1|\ |Y2 - Y1| = 6$

$\Delta X = \Delta Y = -1$

Initial values for

$X = 0 + 0.5 * \text{Sign}(-1) = -0.5$

$Y = 0 + 0.5 * \text{Sign}(-1) = -0.5$

Tabulating the results of each iteration in the step 6 we get,

i	Plot	x	y
1	(–1, –1)	–0.5	–0.5
2	(–2, –2)	–1.5	–1.5
3	(–3, –3)	–2.5	–2.5
4	(–4, –4)	–4.5	–4.5
5	(–5, –5)	–5.5	–5.5
6	(–6, –6)	–6.5	–6.5

Example: Consider the line from (0, 0) to (–8, –4), use general Bresenham's line algorithm to rasterize this line. Evaluate and tabulate all the steps involved.

Solution: Given data,

$$(x_1, y_1) = (0, 0)$$
$$(x_2, y_2) = (-8, -4)$$
$$\Delta x = x_2 - x_1 = 8 - 0 = 8$$
$$\therefore \quad S_1 = -1$$
$$\Delta y = y_2 - y_1 = -4 - 0 = 4$$
$$\therefore \quad S_2 = 1$$
$$\text{Decision Variable} = e = 2 \times (\Delta y) - (\Delta x)$$
$$\therefore \quad e = 2 \times (4) - (8)$$
$$= 8 - 8 = 0$$
$$\boxed{\therefore e = 0}$$

∴ By using general Bresenham's algorithm,

The result in tabulated form as,

Pixel	e	x	y
Initially (0, 0)	0	0	0
(–1, 0)	+8	–1	0
(–2, –1)	0	–2	–1
(–3, –1)	–8	–3	–1
(–4, –2)	0	–4	–2
(–5, –2)	+8	–5	–2
(–6, –3)	0	–6	–3

Q.5. (a) What are the various 3D Transformations? Explain their significance. **(7)**

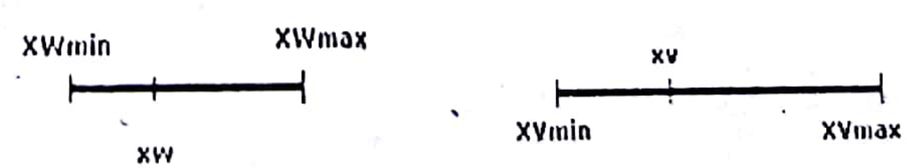
Ans. Refer Q. 3a of First Term Exam. 2016.

Q.5. (b) What do you understand by window to viewport mapping? Derive equations for scaling and translation factors. **(8)**

Ans. Window-to-Viewport mapping is the process of mapping or transforming a two-dimensional, world-coordinate scene to device coordinates. In particular, objects inside the world or clipping window are mapped to the viewport. The viewport is displayed in the interface window on the screen. In other words, the clipping window is used to select the part of the scene that is to be displayed. The viewport then positions the scene on the output device.

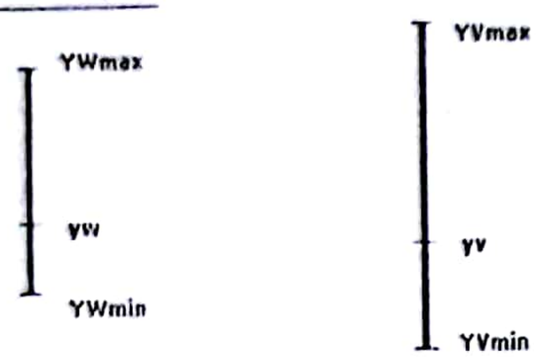
This mapping or transformation involves developing formulas that start with a point in the world window, say (xw, yw). The formula is used to produce a corresponding point in viewport coordinates, say (xv, yv). We would like for this mapping to be "proportional" in the sense that if xw is 30% of the way from the left edge of the world window, then xv is 30% of the way from the left edge of the viewport. Similarly, if yw is 30% of the way from the bottom edge of the world window, then yv is 30% of the way from the bottom edge of the viewport. The picture below shows this proportionality.

For proportionality in x:



Scanned by CamScanner

For proportionality in y:



Using this proportionality, the following ratios must be equal.

$$\frac{xv - xv_{min}}{xv_{max} - xv_{min}} = \frac{xw - xw_{min}}{xw_{max} - xw_{min}}$$

$$\frac{yv - yv_{min}}{yv_{max} - yv_{min}} = \frac{yw - yw_{min}}{yw_{max} - yw_{min}}$$

By solving these equations for the unknown viewport position (xv, yv), the following becomes true:

$$xv = Sx\, xw + tx$$
$$yv = Sy\, yw + ty$$

The scale factors (Sx, Sy) would be:

$$Sx = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$$

$$Sy = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}}$$

And the translation factors (Tx, Ty) would be:

Translation factors

$$tx = \frac{xw_{max}xv_{min} - xw_{min}xv_{max}}{xw_{max} - xw_{min}}$$

$$ty = \frac{yw_{max}yv_{min} - yw_{min}yv_{max}}{yw_{max} - yw_{min}}$$

Q.6.(a) What is Lempel-ziv coding? Explain its functioning with an example? (7)

Ans. The Lempel-Ziv algorithm is a variable-to-fixed length code. Basically, there are two versions of the algorithm presented in the literature: the theoretical version and the practical version. Theoretically, both versions perform essentially the same. However, the proof of the asymptotic optimality of the theoretical version is easier. In practice, the practical version is easier to implement and is slightly more efficient. We explain the practical version of the algorithm as explained in the book by Gersho and Gray and in the paper by Welch.

The basic idea is to parse the input sequence into non-overlapping blocks of different lengths while constructing a dictionary of blocks seen thus far.

Encoding Algorithm

Initialize the dictionary to contain all blocks of length one (D={a,b}).

Search for the longest block W which has appeared in the dictionary.

Encode W by its index in the dictionary.

Add W followed by the first symbol of the next block to the dictionary.

Go to Step 2.

Data a, b, b, a, a b, b a, a b, a b b, a a, a a, b a a, b b, a,

0 1 1 0 2 4 2 6 5 5 7 3 0

Dictionary

Index	Entry
0	a
1	b
2	a b
3	b b
4	b a
5	a a
6	a b b
7	b a a
8	a b a
9	a b b a
10	a a a
11	a a b
12	b a a b
13	b b a

Theoretically, the size of the dictionary can grow infinitely large.

In practice, the dictionary size is limited. Once the limit is reached, no more entries are added. Welch had recommended a dictionary of size 4096. This corresponds to 12 bits per index.

The length of the index may vary. For example, in the n-th block, the current dictionary size is $n + 1$ (assuming that the limit has not been reached). Therefore, we can encode the index of block n using $\lceil \log2(n + 1) \rceil$ bits (rounded up to the nearest integer). For example, the first index can be represented by 1 bit, the 2nd and 3rd by 2 bits each, the 4th, 5th, 6th, and 7th by 3 bits each, and so on. This is the variable-to-variable length version of the Lempel-Ziv algorithm.

For a maximum dictionary size of 2m, variable-length encoding of the indices saves exactly 2m-1 bits compared to fixed-length encoding.

The above example, as most other illustrative examples in the literature, does not result in real compression. Actually, more bits are used to represent the indices than the original data. This is because the length of the input data in the example is too short.

In practice, the Lempel-Ziv algorithm works well (lead to actual compression) only when the input data is sufficiently large and there are sufficient redundancy in the data.

Decompression works in the reverse fashion. The decoder knows that the last symbol of the most recent dictionary entry is the first symbol of the next parse block. This knowledge is used to resolve possible conflict in the decoder.

Q.6. (b) Explain midpoint circle algorithm in detail? Write a short note on the significance of scan conversion algorithms. (8)

Ans. Mid Point Algorithm

Step 1 Input radius r and circle center (x_c, y_c) and obtain the first point on the circumference of the circle centered on the origin as

$$(x_0, y_0) = (0, r)$$

Step 2—Calculate the initial value of decision parameter as

$p_0 = 5/4 - r$ (See the following description for simplification of this equation.)

$f(x, y) = x2 + y2 - r2 = 0$

Scanned by CamScanner

$f(x_i - 1/2 + e, y_i + 1)$
$= (x_i - 1/2 + e)2 + (y_i + 1)2 - r2$
$= (x_i - 1/2)2 + (y_i + 1)2 - r2 + 2(x_i - 1/2)e + e2$
$= f(x_i - 1/2, y_i + 1) + 2(x_i - 1/2)e + e2 = 0$
Let $d_i = f(x_i - 1/2, y_i + 1) = -2(x_i - 1/2)e - e2$
Thus,
If $e < 0$ then $d_i > 0$ so choose point $S = (x_i - 1, y_i + 1)$.
$d_i + 1 = f(x_i - 1 - 1/2, y_i + 1 + 1) = ((x_i - 1/2) - 1)2 + ((y_i + 1) + 1)2 - r2$
$= d_i - 2(x_i - 1) + 2(y_i + 1) + 1$
$= d_i + 2(y_i + 1 - x_i + 1) + 1$
If $e >= 0$ then $d_i <= 0$ so choose point $T = (x_i, y_i + 1)$
$d_i + 1 = f(x_i - 1/2, y_i + 1 + 1)$
$= d_i + 2y_i + 1 + 1$
The initial value of d_i is
$d_0 = f(r - 1/2, 0 + 1) = (r - 1/2)2 + 12 - r2$
$= 5/4 - r$ {$1 - r$ can be used if r is an integer}
When point $S = (x_i - 1, y_i + 1)$ is chosen then
$d_i + 1 = d_i + -2x_i + 1 + 2y_i + 1 + 1$
When point $T = (x_i, y_i + 1)$ is chosen then
$d_i + 1 = d_i + 2y_i + 1 + 1$
Step 3—At each x_k position starting at K = 0, perform the following test—
If $P_K < 0$ then next point on circle (0,0) is $(X_K + 1, Y_K)$ and
$P_K + 1 = P_K + 2X_K + 1 + 1$
Else
$P_K + 1 = P_K + 2X_K + 1 + 1 - 2Y_K + 1$
Where, $2X_K + 1 = 2X_K + 2$ and $2Y_K + 1 = 2Y_K - 2$.
Step 4—Determine the symmetry points in other seven octants.
Step 5—Move each calculate pixel position (X, Y) onto the circular path centered on (XC,YC) and plot the coordinate values.

$$X = X + X_C, Y = Y + Y_C$$

Step 6—Repeat step-3 through 5 until X >= Y.

Scan conversion or scan converting rate is a video processing technique for changing the vertical / horizontal scan frequency of video signal for different purposes and applications. The device which performs this conversion is called a scan converter.

The application of scan conversion is wide and covers video projectors, cinema equipment, TV and video capture cards, standard and HDTV televisions, LCD monitors, radar displays and many different aspects of picture processing.

Scan conversion involves changing the picture information data rate and wrapping the new picture in appropriate synchronization signals. There are two distinct methods for changing a picture's data rate:

Analog Methods (Non retentive, memory-less or real time method)

This conversion is done using large numbers of delay cells and is appropriate for analog video. It may also be performed using a specialized scan converter vacuum tube.In this case polar coordinates (angle and distance) data from a source such as a radar receiver, so that it can be displayed on a raster scan (TV type) display.

Digital methods (Retentive or buffered method)

In this method, a picture is stored in a line or frame buffer with n1 speed (data rate) and is read with n2 speed, several picture processing techniques are applicable when the picture is stored in buffer memory including kinds of interpolation from simple to smart high order comparisons, motion detection and ...to improve the picture quality and prevent the conversion artifacts.

Scanned by CamScanner