COMPUTER GRAPHICS

**Prepared By**

**Sharafat Ibn Mollah Mosharraf**

**CSE, DU**

**12th Batch (2005-2006)**

**Special Edition for CSEDU Students Students**

**TOUCH-*N*-PASS EXAM CRAM GUIDE SERIES**

Table of Contents

[Chapter 3: Basic Raster Graphics Algorithms for Drawing 2D Primitives 1](#_Toc271585473)

[Scan Conversion 1](#_Toc233702955)

[Filling 3](#_Toc233702955)

[Clipping 3](#_Toc233702955)

[Chapter 5: Geometrical Transformations 4](#_Toc271585475)

[Theories 4](#_Toc233702955)

[Excercises 4](#_Toc233702955)

[Chapter 6: Viewing in 3D 5](#_Toc271585477)

[Theories 5](#_Toc233702955)

[Excercises 6](#_Toc233702955)

[Chapter 11: Representing Curves and Surfaces 7](#_Toc271585479)

[Chapter 13: Achromatic and Colored Light 8](#_Toc271585481)

[Chapter 15: Visible Surface Determination 10](#_Toc271585483)

[Z-Buffer 10](#_Toc233702955)

[Ray-Tracing](#_Toc233702955) 10

[Exercises](#_Toc233702955) 10

[Chapter 16: Illumination and Shading 11](#_Toc271585485)

[Chapter: Short Notes on Various Topics 11](#_Toc271585487)

# Chapter 3

# Basic Raster Graphics Algorithms for Drawing 2D Primitives

**Scan Conversion**

|  |  |
| --- | --- |
| **3.1** | **What do you mean by scan conversion? [*2002. Marks: 2*]** |
| **3.2** | **How can you determine whether a point P(x, y) lies to the left or to the right of a line segment joining the points Q(x1, y1) and R(x2, y2)? [*2007, 2004, 2002. Marks: 2*]** |
| **3.3** | **Make a comparative analysis of the following algorithms: [*2005. Marks: 3*]**   1. **Basic incremental algorithm** 2. **Mid-point line drawing algorithm** |
| **3.4** | **Derive the (initial) decision variable and its derivatives in mid-point/Bresenham’s line drawing algorithm while the slope of the line is ≥ 1.0 AND ≤ ∞. [*2005, 2003. Marks: 6*]**  For slope 1.0 ≤ |*m*| ≤ **∞**, the decision to be made is between N and NE. The midpoint to be tested is at (*xp* + ½, *yp* + 1).  Decision variable, *d* = F(*xp* + ½, *yp* + 1)  = *a*(*xp* + ½) + *b*(*yp* + 1) + c …(1)  Initial value of *d*, *d*0 = F(*x*0 + ½, *y*0 + 1)  = *a*(*x*0 + ½) *+ b*(*y*0 + 1) *+ c*  = *ax*0 + *by*0 + *c* + *a* / 2 + *b*  = *F*(*x*0, *y*0) + *a* / 2 + *b*  = 0 + *a* / 2 + *b* [∵ (*x*0, *y*0) is a point *on* the line]  = *a* / 2 + *b*  = *dy* / 2 *– dx*  If N chosen, then *dnew* = *F*(*xp* + ½, *yp* + 2)  = *a*(*xp* + ½) *+ b*(*yp* + 2) *+ c*  = *a*(*xp* + ½) *+ b*(*yp* + 1) *+ c* + *b*  = *d* + *b* [From (1)]  ∴ incN = *b* = –*dx*  If NE chosen, then *dnew* = *F*(*xp* + ), *yp* + 2)  = *a*(*xp* + ) *+ b*(*yp* + 2) *+ c*  = *a*(*xp* + ½) *+ b*(*yp* + 1) *+ c* + *a +* *b*  = *d* + *a +* *b* [From (1)]  ∴ incNE = *a + b* = *dy* – *dx*  [*The algorithm is not needed to answer this question (the question simply didn’t ask for it). However, it is provided below for convenience.*]  To eliminate the fraction in *d*0, we redefine our original *F* by multiplying it by 2; *F*(*x*, *y*) = 2(*ax + by + c*). This multiplies each constant (*a*, *b* and *c*) and the decision variable (*d*) by 2, but does not affect the sign of the decision variable.   |  | | --- | | Listing: The midpoint line scan-conversion algorithm for slope ≥ 1.0 and ≤ ∞. | | 1 /\* Assumes 1 <= |m| <= **∞** \*/  2 void midpointLine(int x0, int y0, int x1, int y1, int value) {  3 int dx = x1 - x0;  4 int dy = y1 - y0;  5 int d = dy - 2 \* dx; // initial value of d  6 int incrN = -2 \* dx; // increment used for move to N  7 int incrNE = 2 \* (dy - dx); // increment used for move to NE  8  9 int x = x0;  10 int y = y0;  11 writePixel(x, y, value); // the start pixel  12  13 while (y < y1) {  14 if (d <= 0) { // choose NE  15 d += incrNE;  16 x++;  17 y++;  18 } else { // choose N  19 d += incrN;  20 y++;  21 }  22 writePixel(x, y, value); // the selected pixel closest to the line  23 }  24 } | |
| **3.5** | **How can you make the mid-point line drawing algorithm slope independent? Explain in detail. [*2006. Marks: 4*]**  **OR, Explain using algorithm. [*2004. Marks: 6*]**  To make the mid-point line drawing line algorithm slope independent, we find out the conditions for a line being placed in a region of the coordinate, and accordingly either transpose the values of *x* and *y* or negate the value of one or both while passing them to the DrawLine() function.  if (abs(*dx*) ≥ abs(*dy*)) { //group0: 0 ≤ |*m*| ≤ 1  if (*x*0 < *x*1 && *y*0 ≤ *y*1) { //op0  DrawLine(*x*0, *y*0, *x*1, *y*1, *value*);  } else if (*x*0 > *x*1 && *y*0 ≤ *y*1) { //op3: change sign of *x*  (*x*0, *y*0)  (*x*1, *y*1)  (*x*1, *y*1)  (*x*0, *y*0)  (*x*1, *y*1)  (*x*0, *y*0)  (*x*0, *y*0)  (*x*0, *y*0)  (*x*0, *y*0)  (*x*0, *y*0)  (*x*0, *y*0)  (*x*1, *y*1)  (*x*1, *y*1)  (*x*1, *y*1)  (*x*1, *y*1)  (*x*1, *y*1)  0  1  2  3  4  5  6  7  *x*0 ≤ *x*1  *y*0 < *y*1  *x*0 < *x*1  *y*0 ≤ *y*1  *x*0 > *x*1  *y*0 < *y*1  *x*0 > *x*1  *y*0 ≤ *y*1  *x*0 > *x*1  *y*0 > *y*1  *x*0 > *x*1  *y*0 > *y*1  *x*0 ≤ *x*1  *y*0 > *y*1  *x*0 < *x*1  *y*0 > *y*1  DrawLine(-*x*0, *y*0, -*x*1, *y*1, *value*);  } else if (*x*0 > *x*1 && *y*0 >*y*1) { //op4: change sign of both *x* and *y*  DrawLine(-*x*0, -*y*0, -*x*1, -*y*1, *value*);  } else if (*x*0 < *x*1 && *y*0 > *y*1) { //op7: change sign of *y*  DrawLine(*x*0, -*y*0, *x*1, -*y*1, *value*);  }  } else { //group1: |*m*| > 1 – swap *x* and *y*  if (*x*0 ≤ *x*1 && *y*0 < *y*1) { //op1  DrawLine(*y*0, *x*0, *y*1, *x*1, *value*);  } else if (*x*0 > *x*1 && *y*0 < *y*1) { //op2: change sign of *x*  DrawLine(*y*0, -*x*0, *y*1, -*x*1, *value*);  } else if (*x*0 > *x*1 && *y*0 >*y*1) { //op5: change sign of both *x* and *y*  DrawLine(-*y*0, -*x*0, -*y*1, -*x*1, *value*);  } else if (*x*0 ≤ *x*1 && *y*0 > *y*1) { //op6: change sign of *y*  DrawLine(-*y*0, *x*0, -*y*1, *x*1, *value*);  }  } |
| **3.6** | **Illustrate the Bresenham algorithm to rasterize a line from (0, 0) to (-8, -4). [*2002. Marks: 5*]** |
| **3.7** | **What steps are required to scan convert an arc of a circle using the polynomial method? [*2001. Marks: 4*]** |
| **3.8** | **Why two sets of “decision variables and its derivatives” are required in mid-point ellipse drawing? Explain the transition/termination criteria from region-1 to region-2 in mid-point ellipse drawing algorithm. [*2007, 2003. Marks: 4*]** |
| **3.9** | **Derive the (initial) decision variable and its derivatives in mid-point ellipse drawing algorithm. [*2006. Marks: 6*]**  **OR, Derive decision parameters for the midpoint ellipse algorithm assuming the starting position at (r, 0) and points are to be generated along the curve path in anticlockwise direction. [*2002. Marks: 9*]** |

**Filling**

|  |  |
| --- | --- |
| **3.1** | **What do you understand by polygon filling? [*2005. Marks: 1*]** |
| **3.2** | **Mention three step process to fill the line span in scan line (polygon filling) algorithm. [*2008. Marks: 4*]** |
| **3.3** | **How are the horizontal edges handled in scan line (polygon filling) algorithm? Explain with proper example. [*2008. Marks: 3*]** |
| **3.4** | **What is Active Edge Table? Write down the processing steps of global edge table in scan line algorithm. [*2008. Marks: 3*]** |

**Clipping**

|  |  |
| --- | --- |
| **3.1** | **What is meant by clipping? [*2007, 2004. Marks: 1*]** |
| **3.2** | **Explain Cohen-Sutherland’s algorithm for line clipping. [*2004. Marks: 3*]** |
| **3.3** | **Explain Sutherland-Hodgeman polygon clipping algorithm. [*2007. Marks: 3*]** |
| **3.4** | **Consider a rectangular window whose lower left hand corner is at L(-3, 1) and upper right hand corner is R(2, 6). Find the region codes for the end points of the following two lines using Cohen-Sutherland line clipping algorithm:**   1. **Line A(-4, 2), B(-1, 7)** 2. **Line C(-2, 3), D(1,2)**   **Also determine the clipping category of the above lines. [*2005. Marks: 3*]** |

# Chapter 5

# Geometrical Transformations

**Theories**

|  |  |
| --- | --- |
| **5.1** | **What are the advantages of using homogeneous coordinate system? [*2008. Marks: 1*]** |
| **5.2** | **What is the difference between geometric transformation and coordinate transformation? [*2006. Marks: 1*]** |
| **5.3** | **How can you make a coordinate transformation matrix using geometric transformation matrix? Explain using proper example. [*2008, 2006. Marks: 5*]** |
| **5.4** | **What do you understand by the composition of geometric/3D transformations? Show (using proper example) that composition of geometric transformations is a must when the center of rotation is not on the origin. [*2003. Marks: 4*]** |
| **5.5** | **Explain how to achieve the transformations so that the line lengths remain unaffected and the motions look realistic. [*2002. Marks: 10*]** |
| **5.6** | **Show that two successive reflections about either of the coordinate axes is equivalent to a single rotation about the coordinate origin. [*2007. Marks: 4*]** |
| **5.7** | **Derive a 4×4 composite rotation matrix while a 3D point rotating across x-, y- and z-axis (assuming center of a rotation at origin). [*2007, 2006, 2005. Marks: 3*]** |
| **5.8** | **Derive a 4×4 3D rotation matrix while rotating across z-axis where the center of a rotation at (a, b, c). [*2004. Marks: 4*]** |
| **5.9** | **Derive a 4×4 3D rotation matrix while rotating across y-axis. [*2003. Marks: 4*]** |
| **5.10** | **Derive a 4×4 3D rotation matrix while rotating across z-axis. [*2002. Marks: 4*]** |
| **5.11** | **Find the transformation for mirror reflection with respect to the xy plane. [*2002. Marks: 6*]** |
| **5.12** | **Compute the tilting matrix for the rotation about the x-axis followed by a rotation about the y-axis. Does the order of performing the rotation matter? [*2002. Marks: 6 + 3*] {Foley p208-209}** |

**Exercises**

|  |  |
| --- | --- |
| **5.1** | **Let *P* be the coordinate of a three-dimensional point on a sphere whose center is at origin.**   1. **What will be the composite rotation matrix for determining the new coordinate of *P* after θx = 30º and θz = 30º rotations?** 2. **What will be the composite rotation matrix to move the point at its original location? [*2008. Marks: 5*]** |
| **5.2** | **Let *P*(12.0, 10.0, 25.0) be the coordinate of a three-dimensional point on a sphere whose center is at origin. Determine the new coordinate of P after θx = 45º and θy = 45º and θz = 30º. (Use the composite matrix) [*2007. Marks: 3*]** |
| **5.3** | **Let *P*(12.0, 0.0, 25.0) be the coordinate of a three-dimensional point on a sphere whose center is at (10.0, 50.0, 30.0). Determine the new coordinate of P after θx = 45º and θy = 45º and θz = 30º. (Use the composite matrix) [*2006. Marks: 3*]** |
| **5.4** | **Let *P*(10.0, 0.0, 30.0) be the coordinate of a three-dimensional point on a sphere whose center is at (10.0, 50.0, 30.0). Determine the new coordinate of P after θx = 30º and θy = 60º and θz = 30º. (Use the composite matrix) [*2006. Marks: 3*]** |
| **5.5** | **Let *P*(12.0, 24.0, 100.0) be the coordinate of a three-dimensional point on a sphere whose center is at (10.0, 50.0, 30.0). Determine the new coordinate of P after Rx(300) and Ry(600). (Use composition of three dimensional transformation rule) [*2004. Marks: 4*]** |
| **5.6** | **Let *P*(12.0, 24.0, 20.0) be the coordinate of a three-dimensional point on a sphere whose center is at (10.0, 50.0, 30.0). Determine the new coordinate of P after Ry(30) and Rz(60). (Use composition of three dimensional transformations rule) [*2003, 2002. Marks: 2/5*]** |

# Chapter 6

# Viewing in 3D

**Theories**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6.1** | **Distinguish between perspective and parallel projection. Mention some anomalies while performing perspective projection. [*2007. Marks: 3*]**   |  |  | | --- | --- | | **Perspective Projection** | **Parallel Projection** | | 1. The distance between the center of projection and the projection plane is finite. | 1. The distance between the center of projection and the projection plane is infinite. | | 1. To determine the perspective projection, its center of projection is explicitly specified. | 1. To define the parallel projection, its direction of projection is specified. | | 1. The size of perspective projection of an object varies inversely with the distance of that object from the center of projection. | 1. The perspective foreshortening (which is human visual system) is lacking. This projection can be used for exact measurement of object. | | 1. Angles are preserved only on those face of the object parallel to the projection plane and do not in general project as parallel line. | 1. The parallel lines remain parallel. | | 1. The center of projection is a point that has a homogeneous coordinate. | 1. The direction of projection is a vector. | |
| **6.2** | **In what way is orthographic projection a special case of perspective projection? Mathematically prove this special case. [*2008. Marks: 3*]** |
| **6.3** | **Derive a perspective projection matrix when the projection plane is at x = d and the center of projection is at x = -d. [*2008. Marks: 4*]** |
| **6.4** | **Derive the general perspective projection onto the view plane z = a where the center of projection is the origin (0, 0, 0). [*2007. Marks: 3*]** |
| **6.5** | **Derive perspective projection matrix where the projection plane is perpendicular to the z-axis, but the center of projection is not on the z-axis. [*2004. Marks: 6*]**  **Show that this can be treated as the standard projection matrix. [*2003. Marks: 7*]** |
| **6.6** | **Derive perspective projection matrices assuming**   1. **Origin at the center of projection** 2. **Origin on the projection plane**   **(Assuming both center of projection and the iso-center of projection plane is on the z-axis) [*2002. Marks: 10*]** |
| **6.7** | **A unit cube is projected onto the XY-plane. Draw the projected image using standard perspective projection with (i) d = 1 and (ii) d = 10, where d is the distance of COP from the projection plane. Assume that the projection plane is at the origin. [*2008. Marks: 3*]** |
| **6.8** | **Show that the normalized perspective to parallel transform preserves the relationship of the original perspective transformation while transforming the normalized perspective view volume onto the unit cube. [*2002. Marks: 4*]** |

**Exercises**

|  |  |
| --- | --- |
| **6.1** | **Let P(40.0, 65.0, 0.0) be the coordinate of a three dimensional point projected on the projection plane. The center of projection of P is (70.0, 10.0, -205.0). The origin of the projection plane is (0.0, 0.0, -150.0). Determine the coordinate of P on the projection plane. [*2007, 2005. Marks: 4*]** |
| **6.2** | **Let P(40.0, 30.0, 200.0) be the coordinate of a three dimensional point projected on the projection plane. The center of projection of P is (70.0, 10.0, 5.0). The origin of the projection plane is (0.0, 0.0, 150.0). Determine the coordinate of P on the projection plane. [*2004. Marks: 4*]** |
| **6.3** | **Let P(27.0, 30.0, 190.0) be the coordinate of a three dimensional point projected on the projection plane. The center of projection of P is (7.0, 60.0, 5.0). The origin of the projection plane is at a distance 150.0. If the projection plane is perpendicular to the z-axis, then determine the coordinate of P on the projection plane. [*2004. Marks: 4*]** |
| **6.4** | **Let P(27.0, 30.0, 190.0) be the coordinate of a three dimensional point on a sphere whose iso-center is at (7.0, 15.0, 200.0). The center of projection is at the origin and the projection place is at a distance 155.0 in z direction from the center of projection.**   1. **Determine the coordinate of P on the projection plane.** 2. **What will be the coordinate of P on the projection plane after Rz(30)? [*2002. Marks: 5 + 5*]** |

# Chapter 11

# Representing Curves and Surfaces

|  |  |
| --- | --- |
| **11.1** | **Write about the different ways of representing polygonal meshes. [*2007, 2005. Marks: 3*]** |
| **11.2** | **Why polynomials of high degree are not useful in curve fitting? [*2005. Marks: 1*]**  {Because it is complex (???)} |
| **11.3** | **What is G1 and C1 continuity? [*2002. Marks: 2*]** |
| **11.4** | **What is a Bezier curve? Mention some properties of Bezier Curve. Find the equations for the control points of a 3rd order Bezier curve. [*2006, 2004. Marks: 4*]** |
| **11.5** | **Show that a 3rd Order Bezier curve can be expressed as**  **Q(t) = (1 – t)3P0 + 3(1 – t)2tP1 + 3(1 – t)t2P2 + t3P3 [*2004. Marks: 3*]** |
| **11.6** | **Write a routine to display a 3rd order Bezier curve using a subdivision method. [*2006, 2004, 2003. Marks: 3*] {11.2.7 p531}** |
| **11.7** | **Derive the basis matrix and blending functions of Bezier cubic curve. [*2007, 2005. Marks: 3*]** |
| **11.8** | **Determine the Bezier blending functions for five control points. Plot each function and label the maximum and minimum values. [*2006. Marks: 3*]** |
| **11.9** | **Explain about Hermite curve with proper mathematical derivations. [*2002. Marks: 8*]** |
| **11.10** | **Write a program in C to display Hermite Curve. [*2002. Marks: 2*]** |
| **11.11** | **How is Hermite geometry and Bezier geometry related? Prove the relation. [*2008. Marks: 3*]** |
| **11.12** | **Find the basis matrices to establish a relationship between Bezier curve, B-spline curve and Hermite curve. [*2003. Marks: 6*]** |
| **11.13** | **Define B-spline curves. Mention some properties of B-spline curves. [*2007, 2002. Marks: 4*]** |
| **11.14** | **Consider a quadratic parametric cubic curve Q(t) = T.M.G, where T = [t2 t 1]. The geometry vector for this curve is defined as G = [P0, P1 P2].**   1. **Find the basic matrix M.** 2. **Find the blending functions for this curve. [*2008. Marks: 2 + 1*]** |
| **11.15** | **A Bezier curve Q has control points P0 = (0, 0, 0), P1 = (0, 1, 0), P2 = (1, 1, 0) and P3 = (2, 0, 0).**   1. **Plot the control points and give a freehand sketch of the curve.** 2. **What point is Q(1/2)?** 3. **What are the values of the derivatives Q′(0) and Q′(1)? [*2008. Marks: 4*]** |

# Chapter 13

# Achromatic and Colored Light

|  |  |
| --- | --- |
| **13.1** | **What is meant by half-toning? [*2004. Marks: 2*]** |
| **13.2** | **Explain half-toning algorithm to convert a grey level image into binary image. [*2005. Marks: 3*]** |
| **13.3** | **Explain the technique of converting a grey image into half-toned binary image using a 3×3 dither matrix using proper flow-chart. [*2003. Marks: 4*]** |
| **13.4** | **Write down the algorithm (with appropriate comments) to convert RGB to HSV color model. [*2007, 2006. Marks: 4*]** |
| **13.5** | **Write down the algorithm (with appropriate comments) to convert RGB to HLS color model. [*2005. Marks: 4*]** |
| **13.6** | **Write down the algorithm to convert HLS to RGB color model and hence describe the red color distribution principle in HLS color model. [*2004. Marks: 5*]** |
| **13.7** | **Write down the algorithm to convert HSV to RGB color model and hence describe the red color distribution principle in HSV color model. [*2003. Marks: 4*]** |
| **13.8** | **Draw RGB and CMY color cube. [*2007, 2006. Marks: 3*]** |
| **13.9** | **Convert the following CMY color to HSV color: C = 1.0, M = 0.5, Y = 0.0. [*2008. Marks: 3*]** |
| **13.10** | **Imagine that you have a CMYK printer where cyan and magenta inks have been swapped. When one tries to print the following colors, what color will actually appear on the paper? [*2008. Marks: 2*]**   1. **Red** 2. **Green** 3. **Blue** 4. **Cyan** 5. **Magenta** 6. **Yellow** 7. **Black** 8. **White** |
| **13.11** | **The color format in BMP file is BGR, whereas the format in OpenGL is RGB. What will happen for the following colors if someone assumes BMP file as RGB format? [*In-course 2008-2009. Marks: 8*]**   1. **Red** 2. **Green** 3. **Blue** 4. **Cyan** 5. **Magenta** 6. **Yellow** 7. **Black** 8. **White** |
| **13.12** | **Convert the following RGB colors into equivalent HLS color model:**   1. **(1.0, 0.7, 0.5)** 2. **(0.0, 0.9, 0.1)** 3. **(0.1, 0.2, 0.7)** 4. **(0.5, 0.4, 0.8)**   **N.B.: Range: Hue = 0 º to 360 º, Others = 0.0 to 1.0 [*In-course 2008-2009. Marks: 16*]** |
| **13.13** | **Convert the following CMY colors into equivalent HSV/HSB color model:**   1. **(1.0, 0.7, 0.5)** 2. **(0.0, 0.9, 0.1)** 3. **(0.1, 0.2, 0.7)** 4. **(0.5, 0.4, 0.8)**   **N.B.: Range: Hue = 0 º to 360 º, Others = 0.0 to 1.0 [*In-course 2008-2009. Marks: 18*]** |

# Chapter 15

# Visible Surface Determination

**Z-Buffer**

|  |  |
| --- | --- |
| **15.1** | **How is the depth of a polygon determined by the painter’s algorithm? [*2006. Marks: 2*] {15.5, p697}** |
| **15.2** | **What is hidden surface problem? Explain different steps involved in Z-buffer algorithm. [*2005. Marks: 4*]**  **ALSO, Explain Z-buffer algorithm. How does it determine which surfaces are hidden and which are not? What happens when two polygons have the same z value? [*2006, 2003, 2002. Marks: 3*] {15.4, p692}** |
| **15.3** | **In a situation A, a scene has 100 polygons and the frame buffer size is 1920×1200 pixels. In situation B, a scene has 100,000 polygons and the frame buffer is 320×240 pixels. Briefly discuss which hidden surface removal method (Back-face Culling or Z-buffer) would be efficient for each scene? [*2008. Marks: 3*]** |

**Ray-Tracing**

|  |  |
| --- | --- |
| **15.1** | **What is meant by ray tracing? [*2002. Marks: 3*]** |
| **15.2** | **Mention some advantages of ray tracing method. [*2006. Marks: 2*] {p725}** |
| **15.3** | **Explain basic ray tracing method with the help of a flowchart. [*2003. Marks: 6*]** |
| **15.4** | **Describe how hidden surface removal and projection are integrated into ray-tracing process. [*2005. Marks: 3*]** |

**Exercises**

|  |  |
| --- | --- |
| **15.1** | **There are three points A(1, 2, 0), B(3, 5, 15) and C(3, 5, 7). Determine which points obscure each other when viewed from point P(0, 0, -15). [*2008. Marks: 2*]**  *Similar to exercise 15.3* |
| **15.2** | **Given points P1(2, 3, 1), P2(3, 7, 15) and P3(3, 7, 9) and viewpoint C(2, 1, -5), determine which points obscure the others when viewed from C. [*2006. Marks: 3*]**  *Similar to exercise 15.3* |
| **15.3** | **Given points P1(1, 2, 0), P2(3, 6, 20) and P3(2, 4, 6) and viewpoint C(0, 0, -10), determine which points obscure the others when viewed from C. [*2005. Marks: 3*]**  Applying the parametricequation of line on points *C* and P1, we get  *x* = t  *y* = 2t  *z* = -10 + 10*t*  To determine whether P2 lies on this line, we see that *x* = 3 when *t* = 3, and then at *t* = 3, *x* = 3, *y* = 6 and *z* = 20. So, P2 lies on the projection line through *C* and P1.  Next, we determine which pointis in frontwith respectto *C*. Now, *C* occurs on the line at *t* = 0, P1 occurs at *t* = 1, and P2 occurs at *t* = 3. Thus, comparing *t* values, P1 is in frontof P2 with respectto *C*; thatis, P1 obscures P2.  We now determine whether P3 is on the line. Now, *x* = 2 when *t* = 2 and then *y* = 4, and *z* = 10. Thus, P3 is noton this projection line and so itneither obscures nor is obscured byP1 and P2. |
| **15.4** | **Given points P1(1, 2, 0), P2(3, 5, 15) and P3(3, 5, 7) and viewpoint C(0, 0, -15), determine which points obscure the others when viewed from C. [*2002. Marks: 8*]**  *Similar to exercise 15.3* |

# Chapter 16

# Illumination and Shading

|  |  |
| --- | --- |
| **16.1** | **What are the differences between local and global light model? [*2008, 2006. Marks: 2*]** |
| **16.2** | **Explain in details about specular reflection in illumination model. [*2002. Marks: 10*]** |
| **16.3** | **Explain diffuse reflection model. [*2007, 2003. Marks: 3*]** |
| **16.4** | **Explain Gouraud shading. Show that this algorithm can easily be integrated with the scan-line visible surface algorithm. [*2008. Marks: 5*]** |
| **16.5** | **How the Phong illumination model can be implemented in OpenGL? [*2008. Marks: 3*]** |

# Chapter

# Short Notes on Various Topics

**Write short notes on the following: [*Marks: 2 to 2.5 each*]**

1. **Recursive ray-tracing [*2007*]**
2. **Line-clipping algorithms [*2007, 2006*]**
3. **Bicubic surface representation techniques [*2007*]**
4. **Antialiasing algorithms [*2007*]**
5. **Specular reflection [*2006*]**
6. **Polygonal surface representation techniques [*2006*]**
7. **Polygon filling algorithm [*2006*]**
8. **Light source attenuation [*2005, 2002*]**
9. **Atmospheric attenuation [*2002*]**
10. **Antialiasing [*2005*]**
11. **Bicubic surfaces [*2005*]**
12. **Video controller [*2005*]**
13. **BMP file format [*2004*]**
14. **B-spline [*2004*]**
15. **Rendering pipeline [*2004*]**
16. **CMY Color Model [*2004*]**
17. **Geometric vector [*2002*]**
18. **Basis matrix [*2002*]**
19. **Blending function in parametric curves [*2002*]**