

## **COMPUTER GRAPHICS AND VISUALIZATION**

### **ENCT 201**

**Lecture : 3**  
**Tutorial : 1**  
**Practical : 3**

**Year : II**  
**Part : I**

#### **Course Objectives:**

The objective of this course is to provide basic principles and their applications to computer graphics and visualization. After completion of this course, students will be able to understand geometric transformations and their applications, two-dimensional and three-dimensional object modeling techniques, rendering techniques, animation and get an overview of the latest trends in computer graphics.

#### **1 Introduction and Application (4 hours)**

- 1.1 History of computer graphics
- 1.2 Overview of graphic systems
  - 1.2.1 Video display devices: Raster-scan displays, random-scan displays, flat panel displays, three-dimensional viewing devices
  - 1.2.2 Graphics software and tools: Coordinate representations, graphics functions, software standards, PHIGS workstations, DirectX, OpenGL, WebGL, Maya, Blender, Unity
- 1.3 Graphics pipeline
  - 1.3.1 Two-dimensional (2D) viewing pipeline
  - 1.3.2 Three-dimensional (3D) viewing pipeline
- 1.4 Applications in various fields like medicine, engineering, art, uses in augmented and virtual realism

#### **2 Raster Graphics and Algorithms (9 hours)**

- 2.1 Rasterizing a point
- 2.2 Rasterizing a straight line: DDA line algorithm, Bresenham's line algorithm
- 2.3 Rasterizing a circle and an ellipse: Mid-point circle and ellipse algorithm
- 2.4 Scan-line polygon fill algorithm
- 2.5 Scan-line fill of curved boundary areas
- 2.6 Boundary-fill algorithm
- 2.7 Flood-fill algorithm
- 2.8 Point clipping

2.9	Line clipping	
2.9.1	Cohen-Sutherland line clipping	
2.9.2	Liang-Barsky line clipping	
2.10	Polygon clipping: Weiler-Atherton polygon clipping	
2.11	Text clipping	
<b>3</b>	<b>2D and 3D Coordinate Systems and Viewing Transformations</b>	<b>(9 hours)</b>
3.1	2D transformation: Translation, rotation, scaling, reflection, shear	
3.2	2D composite transformation	
3.3	Window-to-viewport coordinate transformation	
3.4	3D display methods	
3.4.1	Parallel projection	
3.4.2	Perspective projection	
3.5	3D transformation: Translation, rotation, scaling, reflection, shear	
3.6	3D composite transformation	
3.7	Projection and viewing transformation	
<b>4</b>	<b>Curve Modeling and Surface Modelling</b>	<b>(4 hours)</b>
4.1	Introduction to parametric cubic curves, splines, Bezier curves	
4.2	Surface modeling (Polygon surface, vertex table, edge table, polygon table, surface normal and spatial orientation of surfaces)	
<b>5</b>	<b>Visible Surface Determination</b>	<b>(4 hours)</b>
5.1	Image space and object space techniques	
5.2	Back face detection, Z-Buffer, A-Buffer, Scan-Line method	
<b>6</b>	<b>Illumination and Surface Rendering Methods</b>	<b>(4 hours)</b>
6.1	Algorithms to simulate ambient, diffuse and specular reflections	
6.2	Constant, Gouraud, Phong and Fast Phong shading models	
<b>7</b>	<b>Computer Animation and Visualization</b>	<b>(5 hours)</b>
7.1	Computer animation functions	
7.2	Raster animations	
7.3	Key-frame systems	
7.4	Motion specifications	
7.4.1	Direct-motion specifications	
7.4.2	Goal-directed systems	
7.4.3	Kinematics and dynamics	

<b>8</b>	<b>Latest Trends in Computer Graphics</b>	<b>(6 hours)</b>
	<ul style="list-style-type: none"> <li>8.1 Interactive visualization</li> <li>8.2 Distributed scene rendering</li> <li>8.3 Augmented reality (AR), virtual reality (VR) and mixed reality (MR)</li> <li>8.4 Game development and real-time graphics</li> <li>8.5 Applications of AR, VR and gaming</li> </ul>	
<b>Tutorial</b>		<b>(15 hours)</b>
<ul style="list-style-type: none"> <li>1. Computations regarding raster graphics system- frame buffer size, color manipulation techniques, aspect ratio, refresh rate, resolution</li> <li>2. Implementation of studied algorithms to determine points for digitizing lines, circles, and ellipses</li> <li>3. Computational problems related to different clipping algorithms</li> <li>4. Solution of problems related to 2D and 3D transformations and matrix compositions, including fixed-point scaling, pivot-point rotation, and reflection across an arbitrary line, among others</li> <li>5. Transformation of object descriptions from the window coordinate system to the viewport coordinate system and solving problems related to parallel and perspective projection</li> <li>6. Calculation of the points required to construct different curves studied using the specified set of control points and the desired number of line segments</li> <li>7. Calculation of the surface normal of polygons and evaluating visibility using various visible surface determination techniques</li> <li>8. Calculation of average intensity at a point on a polygon using Gouraud shading</li> </ul>		
<b>Assignment</b>		
Appropriate assignment problems are given to students after the completion of each chapter.		
<b>Practical</b>		<b>(45 hours)</b>
<ul style="list-style-type: none"> <li>1. DDA line algorithm</li> <li>2. Bresenham's line algorithm</li> <li>3. Mid-point circle algorithm</li> <li>4. Mid-point ellipse algorithm</li> <li>5. Lab on 2-D transformations</li> <li>6. Lab on 3-D transformations</li> <li>7. Program for viewing and shading the 3D object</li> <li>8. Clipping hidden surface removal</li> <li>9. Implement the discrete techniques</li> <li>10. Basic drawing techniques in OpenGL</li> <li>11. A simple computer animation</li> <li>12. A simple AR/VR scene or application using a framework (e.g., Unity, ARKit, Unreal Engine)</li> </ul>		

By the end of the practical, students required to develop a prototype project to demonstrate their understanding of computer graphics concepts. Students will work in teams and are encouraged to explore new programming languages or platforms (e.g., Unity, Unreal Engine, WebGL) to complete their project.

#### **Final Exam**

The questions will cover all the chapters in the syllabus. The evaluation scheme will be as indicated in the table below:

<b>Chapter</b>	<b>Hours</b>	<b>Marks Distribution*</b>
1	4	5
2	9	13
3	9	13
4	4	5
5	4	5
6	4	5
7	5	7
8	6	7
<b>Total</b>	<b>45</b>	<b>60</b>

\* There may be minor deviation in marks distribution.

#### **References**

1. Hearn D., Baker, M. P. (1997). Computer Graphics C version (2nd edition), Prentice Hall.
2. Theoharis, T., Papaioannou, G., Platis, N., Patrikalakis, N. M. (2008). Graphs and Visualization: Principles & Algorithms. United States: CRC Press.
3. Foley, J. D. (1995). Computer Graphics: Principles and Practice in C (2nd edition). Germany: Addison-Wesley.