

Introduction to Computer Graphics

Introduction to Computer Graphics

Key concepts, terminology, and foundational techniques

1. What Is Computer Graphics?

Aspect	Description
Definition	The creation, manipulation, and representation of visual content using computers.
Goal	Transform data (geometric, textual, or procedural) into images that can be displayed on screens, printed, or used in virtual environments.
Fields of Application	Video games, scientific visualization, animation, UI design, CAD, medical imaging, virtual/augmented reality, film production, data analytics.

2. Core Components

- Geometry** – The mathematical description of shapes (points, lines, polygons, curves, surfaces).
- Rendering** – Converting geometry + material data into a 2D image (rasterization, ray tracing, etc.).
- Shading & Lighting** – Simulating how light interacts with surfaces (Phong, Blinn-Phong, Physically-Based Rendering).
- Animation** – Changing geometry or camera parameters over time (keyframe, skeletal, procedural).
- Image Processing** – Post-render operations (anti-aliasing, tone mapping, filters).

3. Graphics Pipeline Overview

Stage	Primary Tasks	Typical Data Structures
Application	Scene setup, input handling, physics, animation	Scene graph, object hierarchies

Introduction to Computer Graphics

Stage	Primary Tasks	Typical Data Structures
Geometry	Transform vertices (model ? world ? view ?	Vertex buffers, index
Processing	projection), clipping	buffers
Rasterization	Convert primitives to fragments (pixels)	Framebuffer, depth buffer
Fragment	Apply textures, lighting, per?pixel operations	Shader programs,
Processing		samplers
Output Merging	Blend fragments, write final color/depth	Color buffer, stencil buffer

Modern GPUs use a highly parallel version of this pipeline, with programmable shaders at the geometry and fragment stages.

4. Fundamental Mathematics

Topic	Relevance
Vectors & Matrices	Represent points, directions, transformations (translation, rotation, scaling).
Homogeneous Coordinates	Enable perspective projection and concatenation of transforms.
Linear Algebra	Basis for vertex shading, normal transformations, skinning.
Coordinate Spaces	Model, World, View (Camera), Clip, Normalized Device Coordinates (NDC), Screen.
Interpolation	Barycentric coordinates for per?pixel attribute interpolation.
Geometry	Dot product (angles, lighting), cross product (normals), plane equations.

5. Rendering Techniques

5.1 Rasterization (Real?time)

- **Pipeline?based** – Fast, works well for interactive rates.
- **Key steps:** Vertex shading ? Primitive assembly ? Scan conversion ? Fragment shading.
- **Common APIs:** OpenGL, Direct3D, Vulkan, Metal.

Introduction to Computer Graphics

5.2 Ray Tracing (Offline & Real-time)

- **Principle:** Cast rays from the eye (or light) and compute intersections with geometry.
- **Advantages:** Accurate reflections, refractions, soft shadows, global illumination.
- **Implementations:** Path tracing, photon mapping, RTX hardware acceleration.

5.3 Hybrid Approaches

- **Deferred Shading** – First store geometry data in G-buffers, then compute lighting in a separate pass.
- **Tile-based Rendering** – Divide screen into tiles to reduce memory bandwidth (common in mobile GPUs).
- **Screen-Space Effects** – Ambient occlusion, reflections,