1. Introduction

R. Mukundan,
Department of Computer Science and Software Engineering
University of Canterbury.

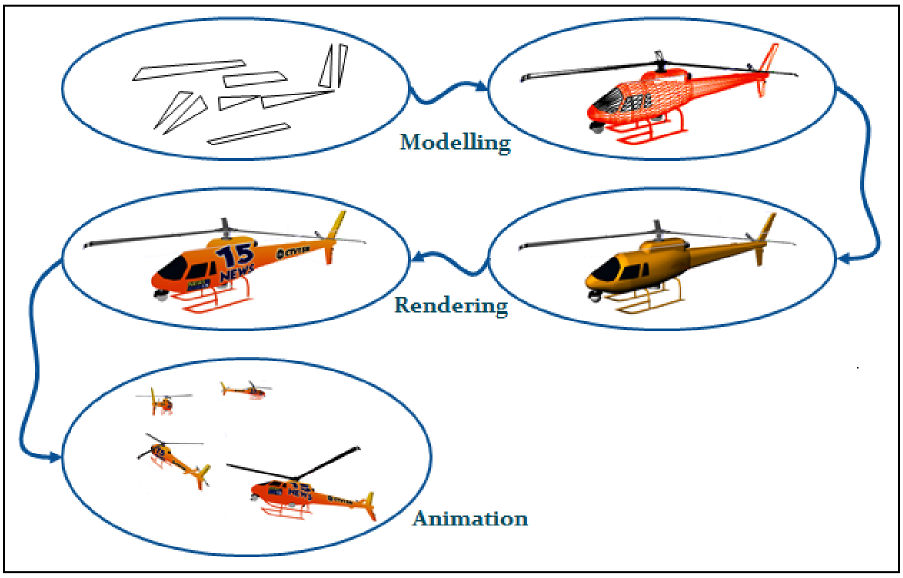
1.1 Computer Graphics

Computer graphics is a rapidly expanding field of computer science, with applications spanning a wide range of areas such as game engine development, mobile graphics technology, cartoon animations, computer-generated commercials, algorithm and information visualization, medical data analysis, process simulation, image processing, user interface design, augmented reality and movie special effects. Until a few decades ago, development of graphics applications required expensive hardware and display systems, and substantial amount of add-on computer resources. Graphics applications have now become ubiquitous, and we encounter them every day on television, laptops and mobile devices. Graphics algorithms are being increasingly used in many scientific and technological areas, with an explosive growth in applications requiring three dimensional scenes, models and animations. On present-day processors, we are able to render highly complex three-dimensional models and scenes with remarkable photorealistic effects and real-time animation rates. The expansion of computer graphics into diverse and interdisciplinary areas is the result of many factors such as decreasing hardware costs, availability of powerful graphics processors and associated software tools, research advancements in the field, and significant improvements in graphics APIs. Additionally, vast amounts of resources including online learning materials, images, 3D models, libraries, programs, data sets and research papers are now easily available to developers of computer graphics applications.

It should be noted here that "computer graphics" and "graphics design" often used as synonymous terms, denote two entirely different fields. While graphics design refers to creative processes (possibly using software packages such as Photoshop) for visual communication, computer graphics can be formally defined as the field of study of methods, algorithms, mathematical techniques and software implementations for digitally synthesizing and manipulating visual content.

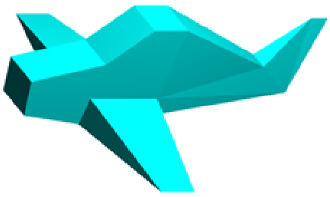
1.2 Graphics Application Development

Modeling, rendering and animation are typically the main processes involved in the development of a graphics application.

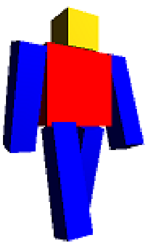


Processes in the development of a graphics application

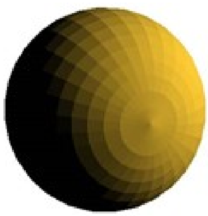
Modelling: Three-dimensional modeling refers to the process of creating scene content using one or many object models. An object might have a simple geometry containing only a few polygons (eg., a cube) or could represent a highly complex shape (eg. a terrain) consisting of several thousand polygons. An object model is made up of *graphics primitives*, usually triangles or quadrilaterals. A modeling process normally involves several transformations (scaling, rotations and translations of polygons). A model can be created using various ways:



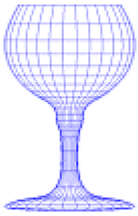
A set of graphics primitives can be selected and carefully positioned and orientated to construct a shape. Simple shapes can be created in this fashion. You may require a graphical sketch of the model for estimating the positions of the polygonal elements.



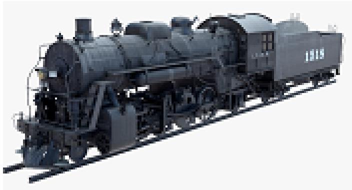
Graphics libraries generally support a few *built-in models* such as the cube, octahedron, sphere, torus, and the most popular teapot. We can create *composite models* by combining a set of built-in models using appropriate transformations. For example, a character model may be created using scaled, rotated and translated versions of a few cubes.



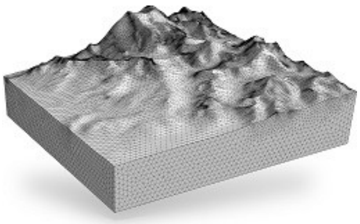
We can procedurally create models using the mathematical equations that represent a shape. Shapes such as spheres, ellipsoids, cones etc., have well-defined mathematical equations that can be used to compute the vertex positions of polygonal surface elements.



We can use surface generation methods such as spline approximations, surface extrusions, surface revolutions and fractal algorithms to create complex shapes using iterative procedures. For example, the revolution of a two-dimensional curve about an axis produces a three-dimensional surface.



There are many types of 3D modelling software which content developers use for creating extremely complex and highly detailed models. Blender, 3DS Max, SketchUp, Maya, MilkShape-3D are examples of such tools.



One could acquire 3D data with the help of laser scanners, range sensors or other distance measurement devices. Modeling algorithms then convert the *point cloud* to polygonal models. Terrain models are often constructed in this manner, and stored as digital elevation maps (DEMs).

Rendering: Rendering refers to the display of a model (or models) as seen from a given view point and along a particular view direction. Rendering involves the use of projective transformations, hidden surface removal algorithms, and illumination models. Textures are mapped to the polygons to generate a photorealistic rendering of objects. Illumination models create variation of shades across a surface, providing depth cues. Shadow generation algorithms are also commonly used to further improve the quality of rendering.

Animation: A series of transformations applied to objects to create motion sequences is collectively called animation. Animations can be defined using functions that modify transformation parameters over time, or with the help of *keyframes* that specify object configurations only at certain points in time. Interpolation between keyframes produces a seemingly continuous motion of objects (or object parts). Animation sequences may also be generated through interactive inputs (mouse motion, keyboard inputs etc.) or through some procedural methods (eg., game logic, simulation outputs).

1.3 Graphics API

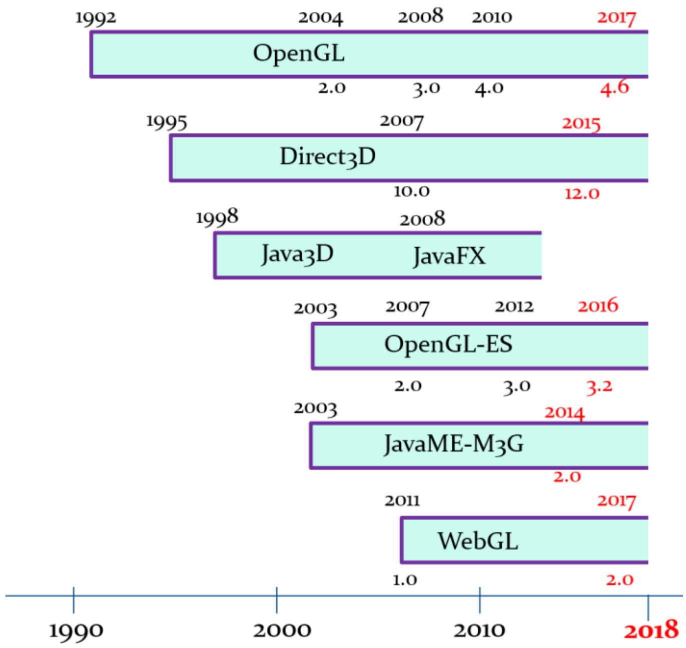
In the 1970s and the 80s, only very few graphics libraries were available to developers. Popular among them were the Graphical Kernel System (GKS) and the Programmable Hierarchical Interactive Graphics System (PHIGS). Some of the processors and display systems (workstations) used in those days were Silicon Graphics (SGI), Tektronix (TEK), and Hewlett Packard (HP).

The Open Graphics Library (OpenGL) was released in 1992.It was developed by Silicon Graphics (SGI) as a platform independent, low-level programmer interface to the graphics hardware. OpenGL is often visualized as a state machine, where the operations performed on the rendering pipeline are based on the current settings of the state machine. OpenGL has now become an industry-standard API for programming 3D graphics and supports a huge range of graphics hardware and features. The current version of OpenGL also includes a powerful shader language designed to run on the graphics processor.

The main competitor for OpenGL has been Direct3D developed by Microsoft as a subsystem component of DirectX for use on Windows machines and the .NET framework. The development of Direct3D saw several version upgrades with changes in Windows operating system. The current version of Direct3D (DirectX-12) has a rich feature set and is supported on Windows-10 machines and Xbox-One game consoles.

WebGL was developed in 2009 as cross-platform 3D graphics API for the web. It uses a set of JavaScript programming interfaces on HTML5 browsers to create a visual layer for 3D applications. The hardware-accelerated 3D functions of WebGL are based on the OpenGL-ES API.

A chart showing important stages in the evolution of graphics APIs and their current versions is given in the following figure.



Developments in Graphics APIs