# 3D Modelling
for Real-Time Applications

An introduction to
3D content production techniques

(focusing on geometry production)

Contents

- Surface modelling
- Modelling strategies
    - Creating geometry using modelling tools
    - Procedural modelling
    - As-built modelling
    - Information visualisation
- Use in a real-time 3D simulation

Polygons

- Real-time (and many animation-) oriented modelling packages are polygon-oriented
- Polygons are multi-sided planar figures
- Polygons consist of vertices and edges
  - The vertices are points in space and the edges are lines between the points

One object built by several polygons
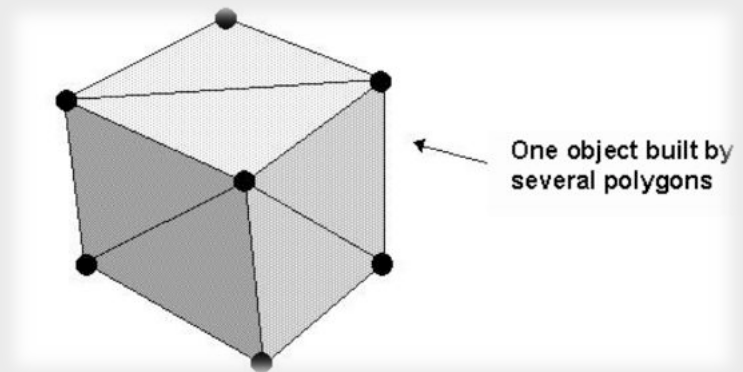
Why Polygons?

- Most 3D graphics hardware works ultimately at the polygon level
  - Usually triangles, as arbitrary polygons can easily be converted (tessellated) into triangles

- Also: 3D graphics hardware usually only supports convex polygons
  - No interior angles larger than 180 degrees
  - None of the edges can cross
 - Don't lose sleep over this!

Curved Surfaces

- Many modelling tools support modelling using mathematical representations of curved surfaces (e.g. NURBS and Bezier surfaces)
  - More intuitive to manipulate in the modelling tool than polygons
  - Very compact representation compared to storing polygons
    - Models stored in polygon form require more RAM/disk

Surface & Solid Modelling

- Polygonal modelling = *surface modelling*
- Emphasises visual appearance
  - A sphere modelled with a polygonal modelling package is hollow and has a faceted surface that looks round if there are enough polygons
- Solid modelling emphasises physical properties
  - A sphere modelled with a CAD package is perfectly round and is solid if you slice it in two

Solid & Surface Modelling
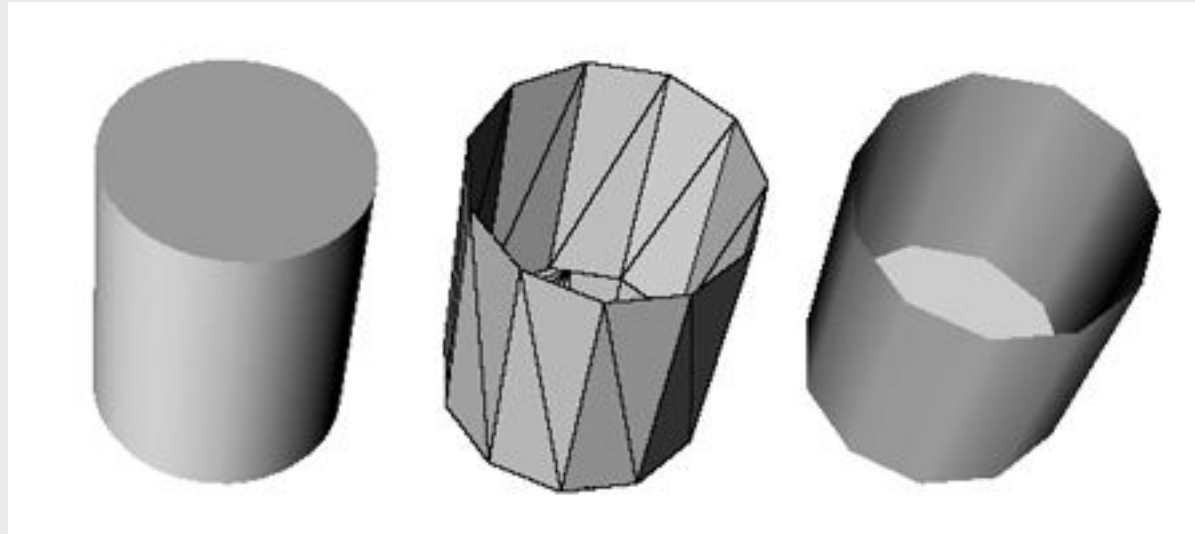
Polygonal Approximation

- Polygonal approximation is used to approximate the curvature of a surface using small flat polygons
- If geometry is stored only in polygonal form, any curved surfaces used to facilitate modelling are converted into polygonal approximations
- If the geometry is stored as curves then this needs to be converted when the model is loaded or displayed
  - Requires a lot of processing if there are a lot of curves
  - "Tessellation"
- CAD models are normally more accurate than polygonal models because *everything* is stored as accurate curves and not as polygons

Polygonal Approximation
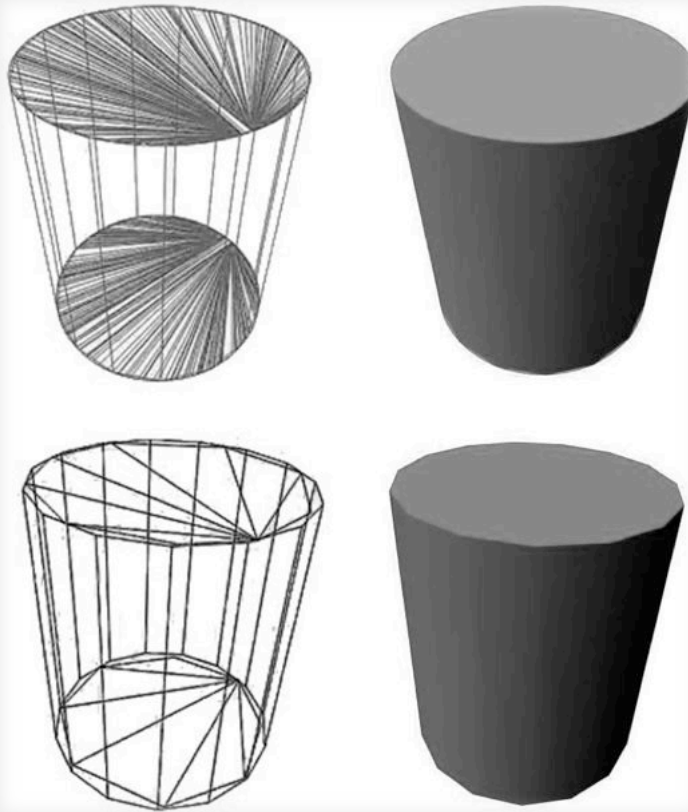
Managing Complexity

- When making models for interactive 3D applications, the aim is to produce objects that look great
  - but use as few polygons as possible ("low polygon count")
  - so that a computer can render them quickly
- So we need to manage a trade-off between visual quality and rendering performance

Managing Complexity

- Need to sustain a *frame rate* that is sufficient to support real-time interaction and smooth motion
  - Frame rate = number of times the graphics frame is redrawn per second
  - The frame rate should *never* drop below 8 – 10 fps (fps = frames per second), with a recommended minimum of 15 fps, and ideally 30 fps or more at all times
  - For acceptable user performance, the frame-rate should not fluctuate significantly if < 30fps
    - *What is comfortable for users in practice depends on the application, however the impact of higher frame rates on measured human performance has been found to be disproportionate to the increase in frame-rate once >15 fps (e.g. Reddy, 1997)*

Primitives

- Primitives are simple shapes generated by a modelling package (or API) using user-specified values for parameters
- Typical primitives include:
  - Spheres
  - Cylinders
  - Cubes
  - Cones
  - Planes (squares/rectangles)
  - Circles and Lines
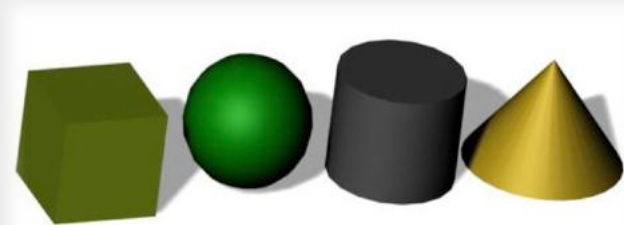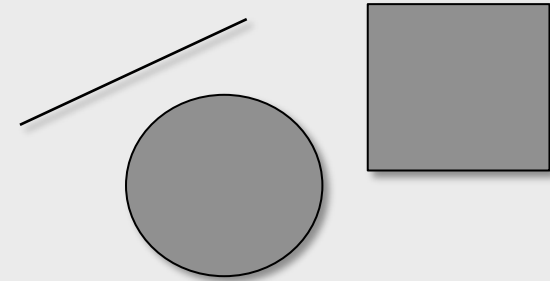
Modelling Strategies

- Could build a model by just defining vertices and edges to build up an object
  - But building a complex model polygon by polygon is very time consuming
- To get results faster, we normally start by creating a primitive shape and then add/delete/move vertices and edges to turn it into the shape we want
  - But this can also be time consuming if the object is complex, especially if it has curved surfaces

Modelling Strategies

- Most modelling tools provide a features that support rapid modelling of complex shapes
  - By learning how to use them, you can save lots of time and produce great results

Lathed Surfaces

- To create a lathed surface, you draw a curve and then rotate it around an axis
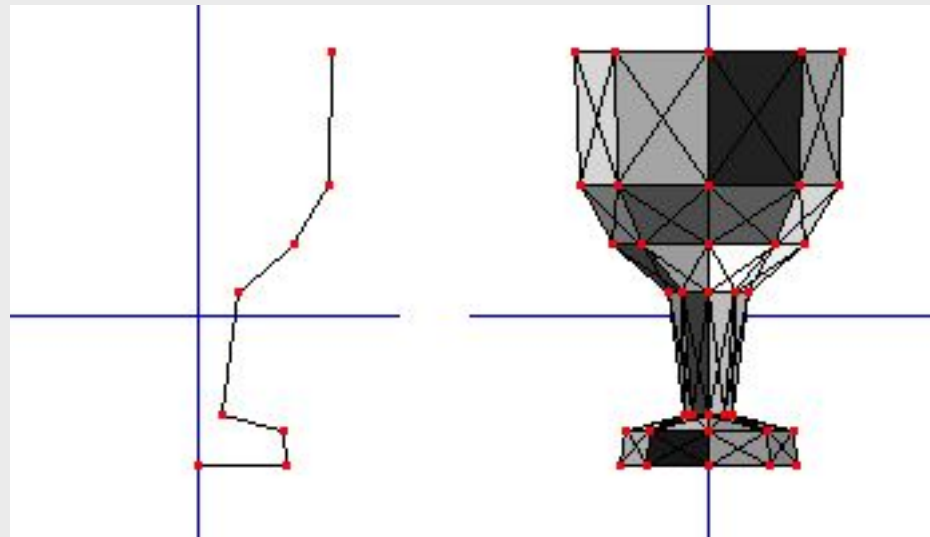
Lathed Surfaces

- You can normally control
  - The number of steps in the revolve
  - The amount the curve is rotate
- Important to achieve the desired form
- Enables control over number of polygons created
  - Tip: In combination with smooth shading, odd numbers of sides give a visually better approximation of a curved surface than even numbers of sides

Extruded Surfaces

- An extruded surface is a curve that is pushed in a straight line through space
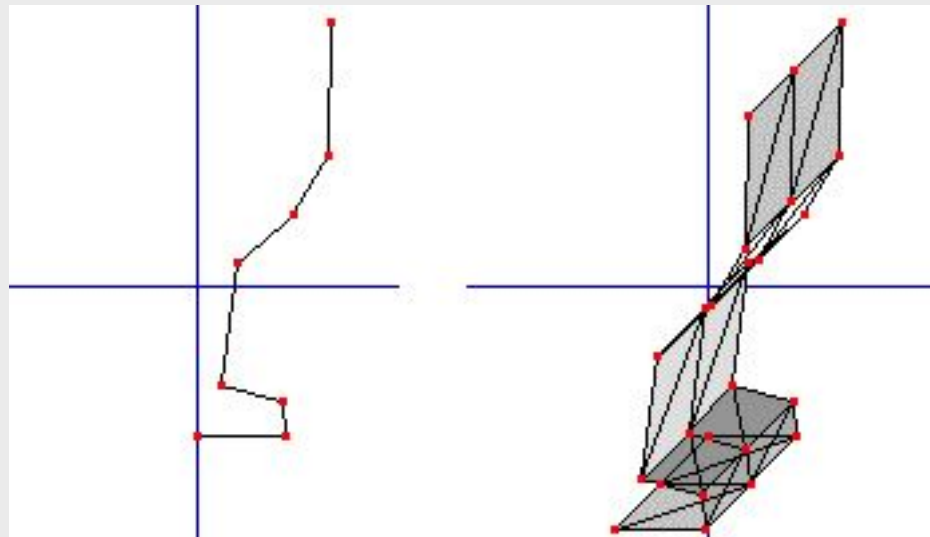
Swept Surfaces

- Swept surfaces are extrusions along a curve. For example, a circle can be swept along a curved path to create a complex pipe
  - Some modelling tools also support transform operations on the curve so that the swept surface can be scaled or rotated along the path.

Lofted Surfaces

- A lofted surface (or "skin") is produced by selecting a series of curves and performing a lofting operation, which joins them together
  - E.g. the hull of a ship or a vaulted roof

Geometry Deformation

- Some modelling programs have tools that enable you to deform objects as if they were made of clay

Geometry Deformation

- Geometry deformation techniques enable complex organic objects to be sculpted relatively quickly
  - BUT care should be taken so that the final object does not have more polygons than necessary

- Simpler deformation techniques that modelling packages typically offer include the ability to bend, twist or taper objects

Procedural Modelling

- Some modelling packages and 3D run-time systems support the definition of *procedural models*

  - Models are generated using small programs (*procedures*)

- Input to procedure can be a logical specification or data set

- Anything that can be defined using an algorithm can be created using procedural modelling

- Particularly useful for create complex organic shapes using genetic or fractal algorithms

  - Landscapes, trees, fur, feathers, clouds, fire, etc.

As-built Modelling

- It may not be possible to create a an accurate model of a real environment or a complex object in a reasonable amount of time 'by hand'

- Techniques and methods have been developed that enable us to create relatively accurate 3D models automatically or semi-automatically

Laser Scanning

- Lasers can be used to scan outer surfaces of building or the interior of rooms
  - Can also be used to accurately scan small objects
  - Can even be used to model large outdoor areas
- The distance a laser beam travels until it hits a surface is registered as a point in space
  - Many points are registered to create a *point cloud*
- Point cloud is then converted into a polygonal model that can be further optimised

Photogrammetry

- Objects can be created using photographs as a starting point to capture geometry and surface characteristics

- Two common modelling methods:
  - Systems interprets multiple images and generates geometry automatically
    - Typically for small objects on a turntable with a blue screen background
  - User models geometry manually using digital photographs as a guide to defining edges and vertices to which the software maps textures

Low-cost: Microsoft Kinect

Information Visualisation

- 3D space can be used to present data to users in informative ways

- Successful visualisation offers the user greater insight into the data set presented

- Can be viewed as an extension of procedural modelling, where data is read and a 3D representation is created and presented to the user

Information Visualisation
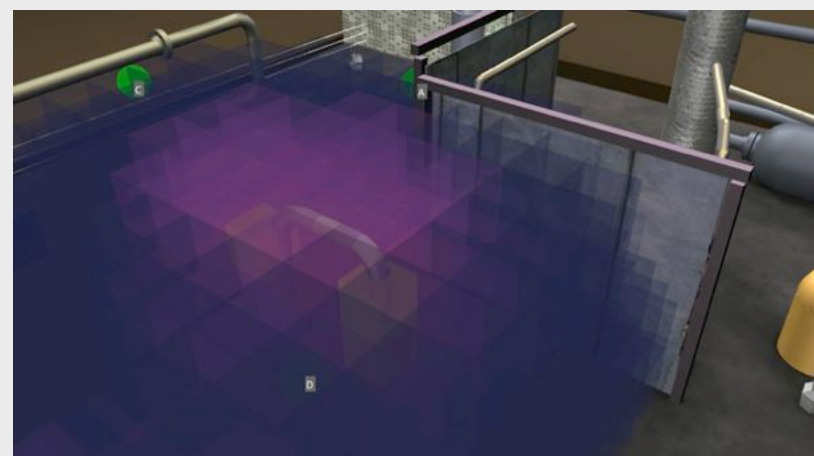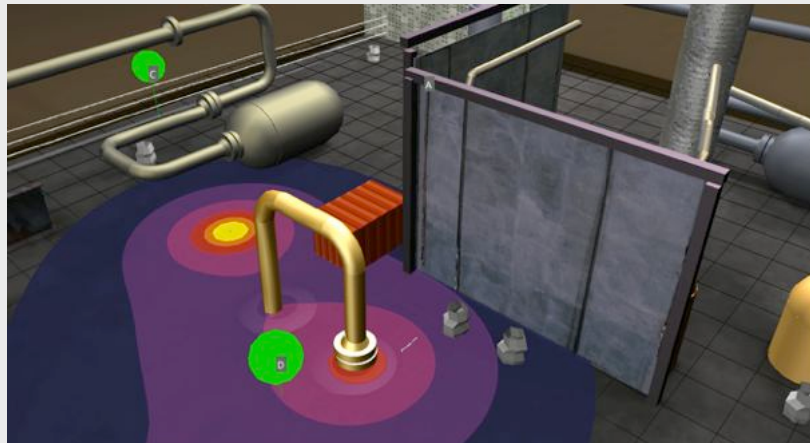
Structure and Appearance

- Geometry in a 3D model file needs to be structured to support the simulation
  - Typically hierarchical structure
  - *Transform* nodes used to specify the position and rotation of *Shapes* in hierarchy
  - *Shapes* themselves comprise of *Geometry* and *Appearance* nodes
  - *Appearance* node specifies the surface characteristics to be applied to the geometry
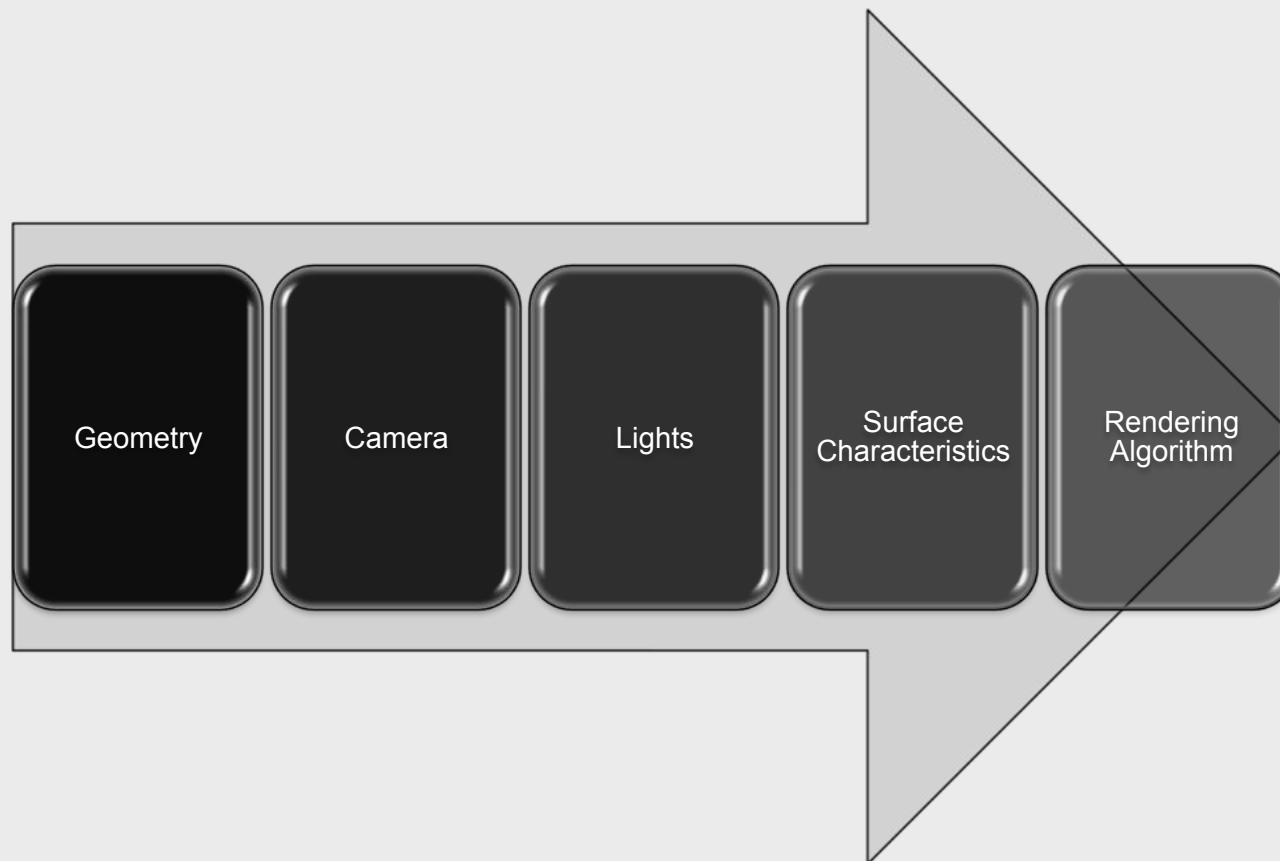
Surface Characteristics

Rendering an image

Geometry → Camera → Lights → Surface Characteristics → Rendering Algorithm
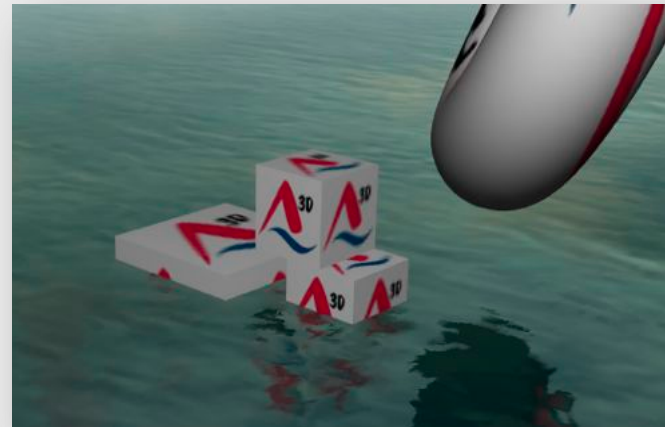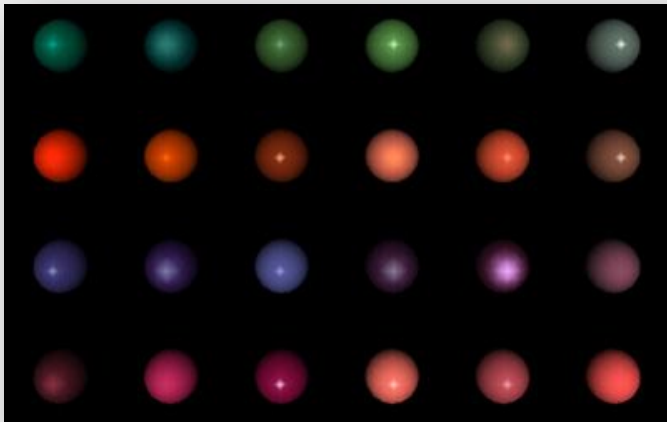
Rendering a scene from file

- Data in 3D model file is converted into a scene graph branch representation when loaded into run-time simulation process
  - For example, into a scene graph using a *Loader*
  - Loaded content can be inserted into scene
- If resulting scene has an associated camera/view and contains lights then we can render it using a renderer

Summary

- 3D content can be pre-modelled using a 3D modelling tools or be generated using as-built geometry capture techniques (or both!)

- 3D content, comprising of structured geometry and surface characteristics, can be loaded into a simulation at run-time

- Content should be structured to support the simulation needs with appropriate level of detail to meet performance requirements