MODELLING 3
Low level object modelling

OVERVIEW

Objects in 3D Applications
- The polygon mesh
- What’s in a model
- Procedural models
- How do we create models?

MODELLING PROCESSES

Creation / Acquisition
- Image & 3D Scanners
- 3D Model - extraction (from images/video)
- Procedural Modelling
- Mechanical Digitizers
- Modelling Tools
- From Scratch

Manipulation / Editing
- Rendering
- Restoration
- Procedural Animation
- Animation Tools
- Data Driven
- Direct Manipulation

HIGHLY AUTOMATED
MANUAL

Rendering
Rasterization
Procedural Animation
Animation Tools
Data Driven
Direct Manipulation

POLYGON MESH

This is de facto method for modelling a large range of complex shapes in 3D applications
REPRESENTING OBJECTS WITH LINES

Start with Points/Vertices on the surface of an object
- Defined as positional vectors \(x,y,z\) from the origin

Edges are line segments on the surface, defined by pairs of points.

Closed polygons are made up of a number of co-planar edges.

SURFACE REPRESENTATIONS

Polygon Mesh a.k.a. Wire frame data

SURFACE REPRESENTATIONS

PARAMETRIC SURFACES

http://www.turbosquid.com/FullPreview/Index.cfm/ID/301320

PARAMETRIC SURFACES

MATHEMATICAL MODELS

Implicit models: a point lie on a line/surface/volume? does a line intersect an object?
e.g. in Ray-tracing: does a given Ray hit an object? if so, where?

Parametric models: points are on the line/surface/volume?
e.g. calculate 100 points on the surface of a given object

\[
x = \sin \theta t
\]
\[
y = \cos 2\theta
\]

PARAMETRIC EQUATIONS

Simple 2D Example: \(t\) is the parameter

\[
x = t/t/2
\]
\[
y = t/2
\]

A method of defining a mathematical function using a set of equations with one or more common parameters

For modelling purposes, an evaluator function generates multiple points by calculating \(x, y, z\) values for multiple values of the parameter(s)
**PARAMETRIC EQUATIONS**

Simple 2D Example:

\[ x = t \]
\[ y = t^2 \]

If \( t = 1 \) then \( <x, y> = <1, 1> \)
If \( t = 2 \) then \( <x, y> = <2, 4> \)
If \( t = 3 \) then \( <x, y> = <3, 9> \)
If \( t = 10 \) then \( <x, y> = <10, 100> \)

... 

**PARAMETRIC SURFACES**

Require two parameters

**PARAMETRIC SURFACES**

The evaluator will provide different values for \( u \) and \( v \) to get different vertex positions and connect these up for a complete mesh.

Depending on how many values we give it we get more detail on the cylinder.

**PARAMETRIC SURFACES**

Low detail sphere
High detail sphere (more different input values of \( u \) and \( v \) used to generate more \( <x, y, z> \) values)

**DISCRETIZATION**

In most cases, mathematical models need to be converted to discrete representations before we use them in scenes. We need to find an appropriate resolution for this.

**PARAMETRIC PATCHES**

- Spheres and cylinders are relatively simple to create
- In some cases we may be able to find parameterised functions for a number of shapes e.g. quadrics, super quadrics
- However parameterizing complex/real-world objects can be too difficult
- One alternative is parametric patches, essentially break up complex shapes into smaller regular sub-shapes, and parameterize these
NORMAL VECTORS

RECALL: NORMALS IN ILLUMINATION

Light reflected off a point depends on a number of vectors:

\[ \mathbf{n} \]: Normal to surface
\[ \mathbf{l} \]: Light Direction
\[ \mathbf{v} \]: Viewer direction
\[ \mathbf{r} \]: Direction of reflection
\[ \mathbf{t} \]: Direction of transmission/refraction

DIFFUSE REFLECTION: affected by the angle between light direction and normal direction.

SPECULAR REFLECTION: affected by the angle between light direction and reflected direction (which depends on angle between light and normal)

MIRROR-LIKE REFLECTION (e.g., in ray-tracing): affected by reflected direction (which depends on angle between light and normal)

REFRACTION: affected by transmitted direction (which depends on angle between light and normal)

NORMALS

Normals are directional vectors for each face or vertex:
- ESSENTIAL for proper lighting of the object
- In most cases this is added to the representation of the object
- If not these must be calculated at run-time for lighting purposes

The normal can be calculated by taking the cross product of two edges of a polygon

MACH BANDS

For curved objects polygon meshes are an approximation –
- when rendering the mesh errors can be very apparent

However clever shading can mask these errors e.g., Gouraud shading

SHADING

Approximate normals at every vertex
- By averaging neighbouring face normals

Then interpolate colours between vertices

GOURAUD SHADING
TEXTURE COORDINATES

Two numbers \((u, v)\) coordinate "vector" that
identifies where on the texture the colour of each vertex must come from.
Texture artist must provide these
Some tools for simplifying the process
Also used for bump mapping, normal mapping, etc.

TEXTURE COORDINATES

Given a 3D model and a texture map that needs to be mapped onto it:

- 3D coordinates \(x, y, z\) of each vertex (corner point) used to define the geometry.
- 2D UV coordinates used to define "from where in the texture that point should get its colour".

Each vertex has its own \(u\) and \(v\) coordinates (where the object is transformed is no changed but its orientation constant).

TEXTURES IN INTERACTIVE 3D APPLICATIONS

For more complex shapes the texture won’t necessarily be the same size or shape as the geometry.

TEXTURE COORDINATES

Assigning colours point for point is a lot of work.
These can also be derived using some standard parameterisation:
- E.g. spherical, cylindrical mapping.
For MAX objects you can create mapping coordinates when you create the object.
Or using the UVW Mapping modifier.

SOME STANDARD MAPPINGS

- Planar
- Cylindrical
- Spherical
- Face
Treat us if you were texturing a sphere

SURFACE REPRESENTATIONS

Polygon Mesh a.k.a. Wire frame data

Creating 3D Meshes

Creating Shapes

Spline
Creating Meshes from Scratch

Using Splines (and a few modifiers), you can create fairly general shapes from scratch with a bit of effort.

For details check out the "Modelling a Knight" tutorial: https://goo.gl/z6qmCw

Max Examples

Shapes in 3D Studio Max

Generally 2D shapes – that can be used as a basis for creating meshes.

Extrude

Create a shape
On the Modifiers tab, pull down the modifiers list and Select the Extrude modifier.

Extruded star shape

Lathe

Create any shape
Select the Lathe Modifier.

Sweep

Create a Spline
Choose the Sweep Modifier
Select a sweep cross-section.

YouTube tutorial: http://www.youtube.com/watch?v=IBnpO9wktls
CREATE A SHAPE OF YOUR OWN AND USE IT AS A CROSS SECTION

PRACTICAL EXAMPLE

1. Create the profile of a chess piece (rook)
2. Use the Lathe tool to create a 3D object
3. Use the Edit Mesh tool to select and extrude parts of the top to to create the jagged walls of the turret

EDITING MESSES

Click on a mesh
Select the Modifiers Tab on the right hand pane
Click on Modifier List
Select the Edit Mesh Modifier

SELECTING INDIVIDUAL PARTS OF THE MESH

- Vertex
- Edge
- Face
- Patch
- Element / Object

Some other functions to make selection easier

MODIFYING THE MESH COMPONENT

Now, we can apply any kind of transform to the selected part of the Mesh. Neighbouring vertices will automatically adjust to fit

SOFT SELECTION

Soft Selection gradually applies adjustments to even non-neighbouring vertices
So what can we do with this?

Possibly... edit existing meshes – including basic shapes to generate our own objects.