Computer Graphics

COMP3421/9415
2021 Term 3 Lecture 8

What did we learn last lecture?

Scene Graphs

- Organisation for complicated scenes and hierarchical objects

Depth Testing

- Rendering things in the right order
- Seeing only what's in front

Blending

- Also seeing what's behind something if it's transparent!

What are we covering today?

Parametric Equations

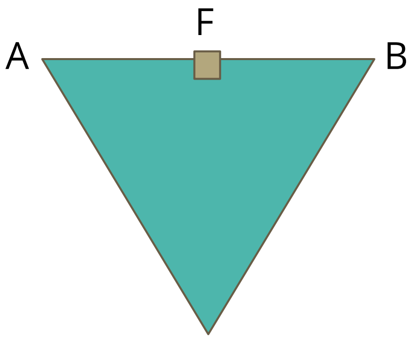
- Linear Interpolation
- Using parameters to control movement and curves
- Also using time as the driver

Linear Interpolation

Maths inside Fragment Shaders

How do we choose a texture coordinate or colour in between two vertices?

- Vertices A and B have texture coordinates A_T and B_T
- What's the texture coord of my fragment F?
- If it's halfway between them:
  - $F_T = A_T * 0.5 + B_T * 0.5$
  - This means both A and B have 50% influence over the texture coordinate



Linear Interpolation

This works for all points between the two vertices

- Not just the halfway point
- If F is at A:
  - $F_T = A_T * 1.0 + B_T * 0.0$
- Or at B:
  - $F_T = A_T * 0.0 + B_T * 1.0$
- You can see a pattern forming:
  - $F_T = A_T * (1.0 - t) + B_T * t$
- If we use a parameter, `t`, we can represent all the possible values between the two points

What can we do with this technique?

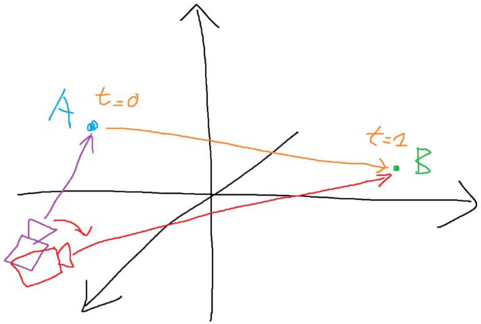
Finding points between two points

- Moving `t` between `0` and `1` can give us any point between `A` and `B`
- This gives us a very simple way of moving between vertices
- We can describe a line or path by only using the end points
- We can also do a smooth transition between properties like colours or texture coordinates

Using Time as a Parameter

Moving in time

- If we change our `t` based on `delta time`, we can animate movement between positions
- We can apply an interpolation to a coordinate or vector, so we could do something like a smooth `delta time` based pan of a camera



`t = elapsed time/5sec`

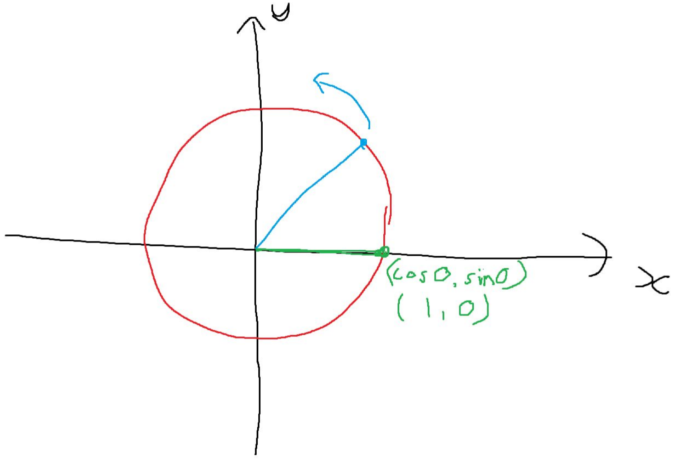
Change the camera's "target" by linear interpolation and recalculate lookAt each frame. In 5 seconds we'll pan from A to B.

Parametric Equations

Straight Lines Only?

Let's add more interesting paths

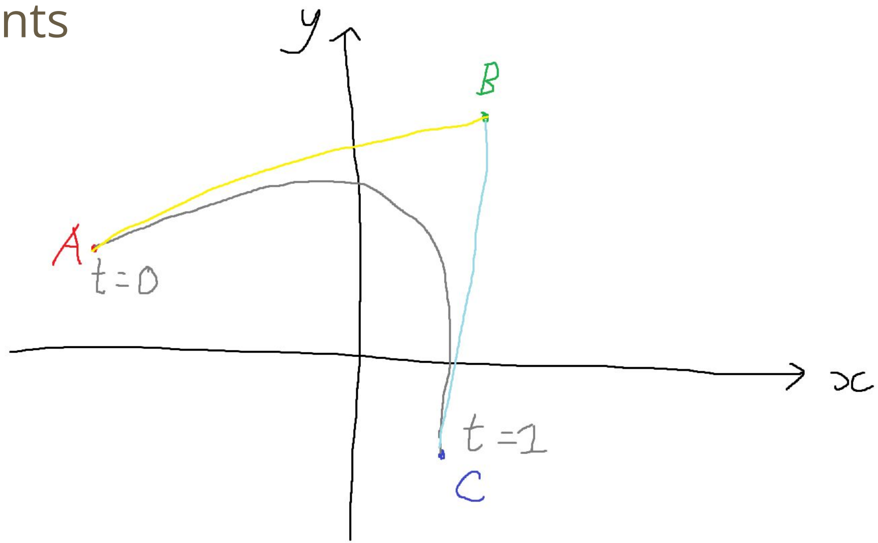
- Linear Interpolation is straight lines between two points or values
- But the idea of parametric equations can do way more than that
- Try this one:
  - `x = cos(t), y = sin(t)`
  - This one even works for any value of `t`



Control Points

Points that influence a line based on a parameter

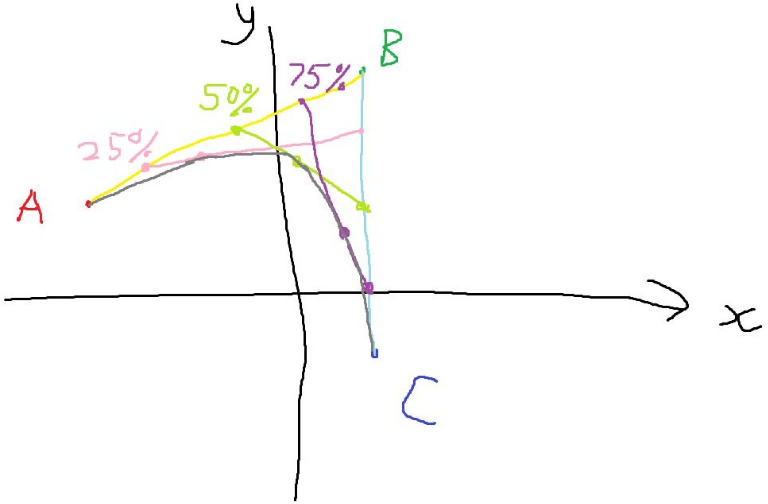
- We've seen a single line with two points
- What about more points?
- Let's look at Bezier Curves
- They're parametric
- And use multiple points



Bezier Curves/Splines

Makes use of Linear Interpolation

- If we have multiple points, we'll linearly interpolate at multiple levels
- Each line we draw is a tangent to the curve



Different types of Bezier Curves/Splines

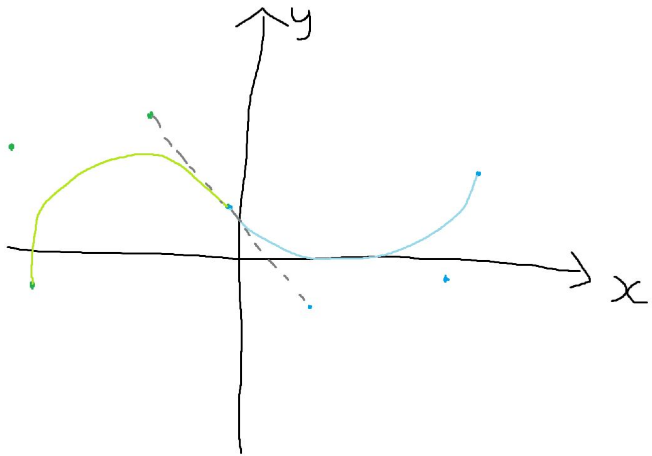
Each curve has a parametric formula

- Two points (Linear Interpolation)
  - $P = (1-t)P_1 + tP_2$
- Three points (Quadratic Curve)
  - $P = (1-t)^2P_1 + 2(1-t)tP_2 + t^2P_3$
- Four points (most commonly used in Graphics)
  - $P = (1-t)^3P_1 + 3(1-t)^2tP_2 + 3(1-t)t^2P_3 + t^3P_4$

Useful Properties

Bezier curves . . .

- Tangents based on control points
  - At either end, the two closest control points form a tangent
- Join multiple curves together smoothly using colinear control points



More Advanced Curves and Splines

We can join them together, but . . .

- While the gradient will be equal at the join
- No guarantees about the second derivative

Why is that an issue?

- If we're using this curve as a path to move on, the speed won't be the same at the join
- Check out B-Splines if you want to know more about possible solutions

Splines in Graphics

- Polygon Rendering works in straight lines
  - Want a lot of points spaced out along a curve?
  - A parametric curve allows us to create an arbitrary number of points
  - We can draw our lines between those points to approximate the curve
- If we are moving an object along a curve
  - We can reach arbitrary positions using our parameter (link delta time to the parameter)
  - If we need a tangent, we can do a simple approximation by creating another nearby point
- Easy to modify
  - Just move control points around to change the nature of the curve
- A downside: Can't quite control size and speed
  - You can't necessarily move along a spline at a fixed speed
  - Parameter based movement is based on how far apart control points are

Break Time

Ed Catmull's interesting career

- Invented Texture Mapping (1974)
- Invented the Catmull Rom Spline (1974)
- Used this and similar techniques in Keyframe Animation (1970s)
- Ended up at Industrial Light and Magic *(owned by George Lucas)* (1979)
- Steve Jobs buys Lucasfilm digital division and creates Pixar (1986)
- Ed Catmull was at one point President of Disney and Pixar

Image credit: Jeff Heusser (VES Awards 2010)

Using Splines

Hello Teapot our old friend

How would we create this object?

- Several tricks in use here!
- We have curves, but how are we creating surfaces?
- We can create something like the teapot with just some simple equations and transform matrices

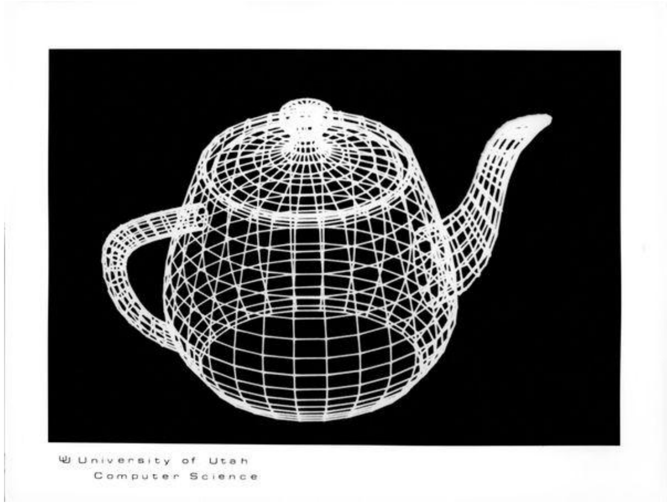


Image credit: School of Computing, University of Utah

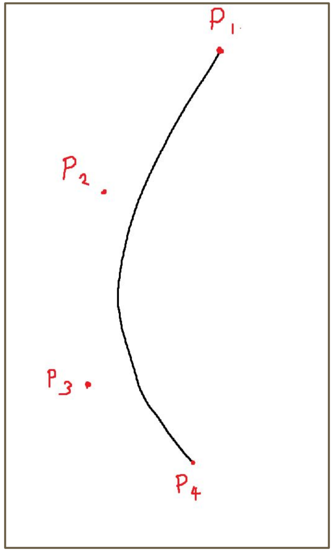
Surfaces of Revolution

The body of the teapot

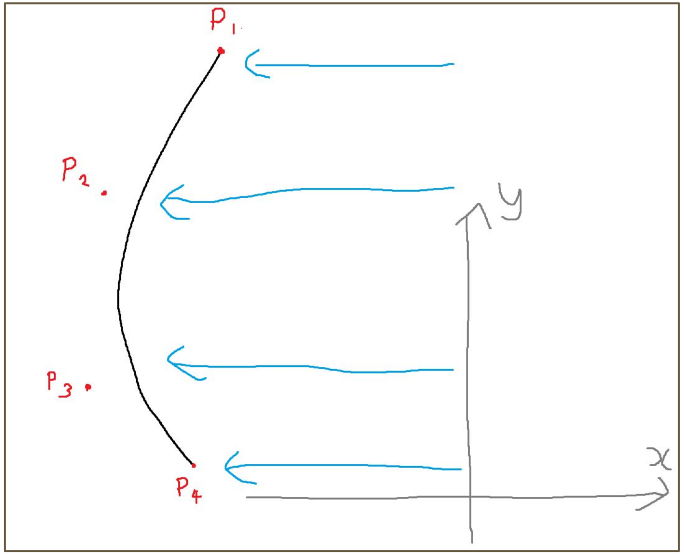
- We can create a curve (side of the pot)
- Translate that curve away from (0,0,0)
- Create a series of points along that curve (using values of t)
- Then we can rotate the curve and its points around the Y axis
- At different rotation angles (the Utah teapot has about 30) we can create vertices
- Make triangles from those vertices and build up buffers

A Surface of Revolution

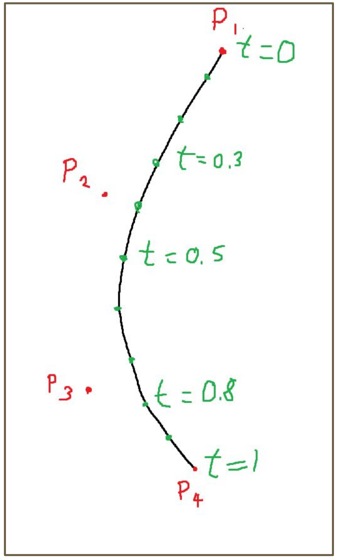
In Images



Create a curve



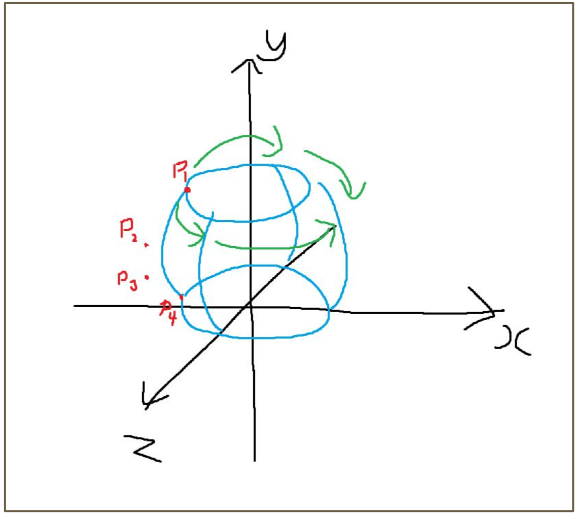
Translate away from y axis



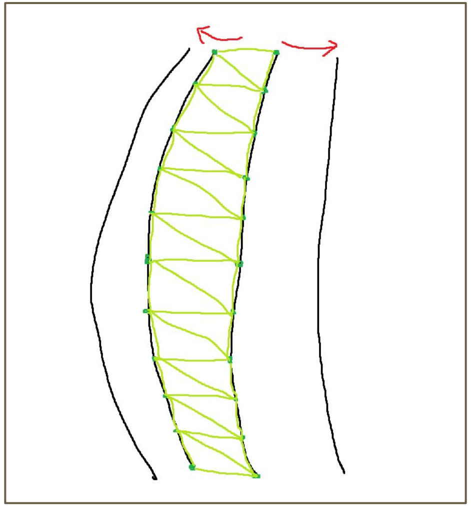
Choose t values
for points

A Surface of Revolution

In Images (continued)



Rotate the curve to different orientations around the y axis



Between two of the close curves, create vertices and triangles

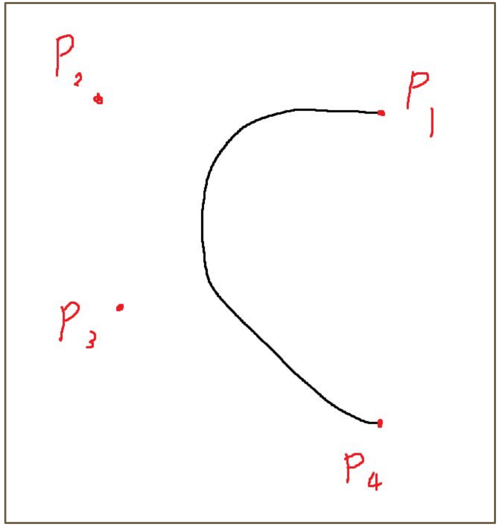
Extrusion

Dragging a shape through space, the handle of the teapot

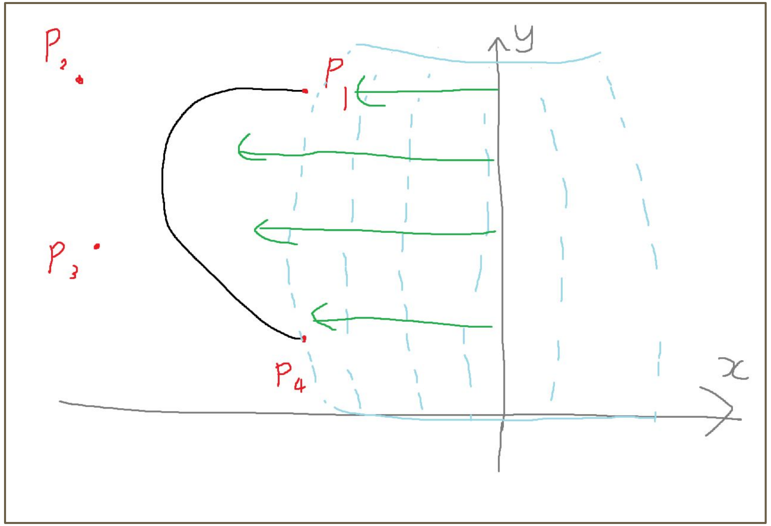
- We can create a curve (the centre of the handle)
- Translate that curve into the correct position at the back of the teapot
- Create a series of points along that curve (using values of t)
- Create a circle (the thickness of the handle)
- Place the circle at each those points
- Use the circle to create vertices

Extrusion

In Images



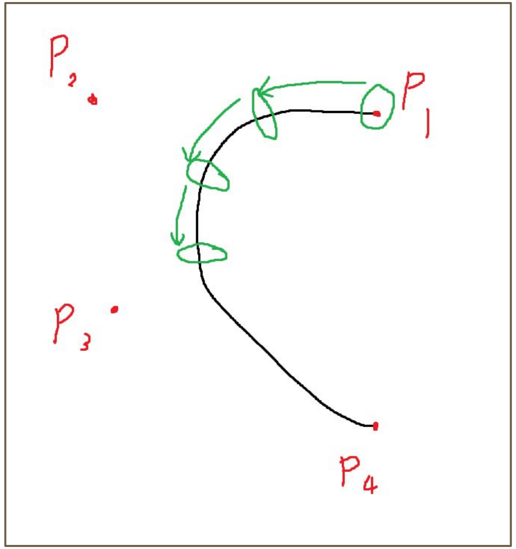
Create a curve



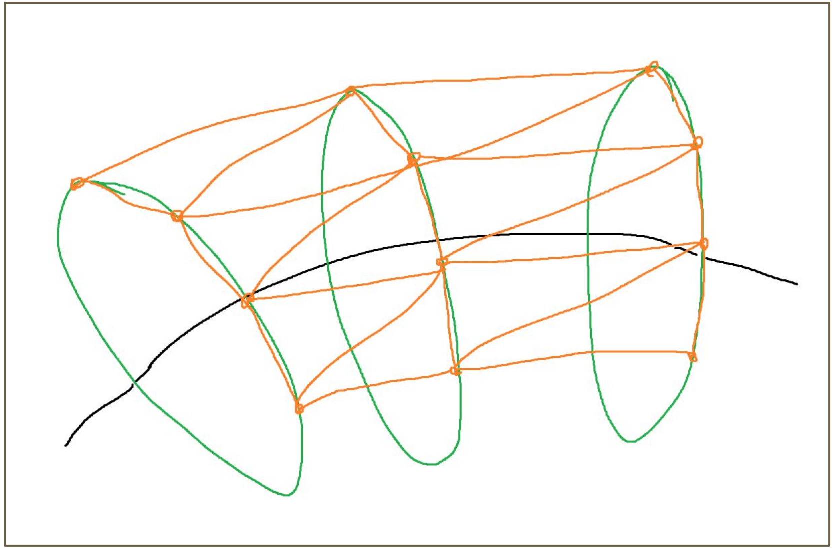
Translate the handle into position

Extrusion

In Images (continued)



Create a circle and place it at different points along the curve



Create vertices around the circles and create triangles between them

Other parts

The lid

- Another surface of revolution, just a slightly more complex curve

The spout

- This gets harder
- An extrusion that is scaled based on distance from the pot?

Techniques for 3D Object Creation

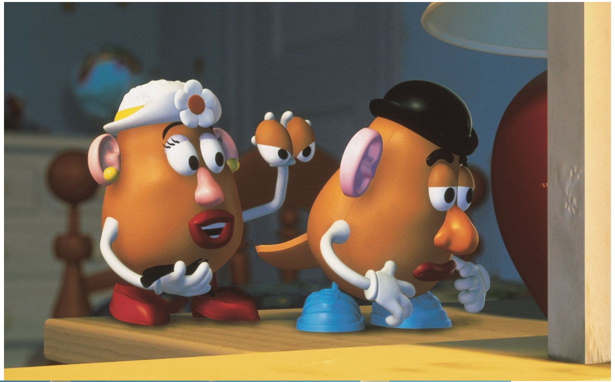
Digital Artists might use a lot of these

- Rotation and extrusion are definitely used
- And simple things like scale, rotate and translate!
- Artists will not usually be expected to be computer scientists and mathematicians
  - So there are tools like Maya and Blender to hide the details
- There's also raw sculpting though
  - Digital Clay! (Zbrush uses this kind of technique)

Let's guess how some things were made

Toy Story is a classic that's historically very important!

- Potato heads
  - Obvious separate objects with scene graph attachments
  - Some simple rotational volumes
- Buzz Lightyear
  - Transparency nightmare!
  - Very simple scene graph with rigid components
- Woody
  - Squishy bits . . . we'll talk about these later!



Images credit: Disney Pixar

What did we learn today?

Parametric Maths and its applications

- Linear Interpolation
- Parametric Curves/Splines
- Using Bezier Curves in 3D
- A small look at how 3D Artists create some game/film assets

Homework

- Watch Toy Story and Monsters, Inc. (Two Pixar films 6 years apart)
- See if you can guess how some things were made and also the technical advancements between the two movies