Animation

COSC450

Key-Frame Animation, Interoplation, and Splines

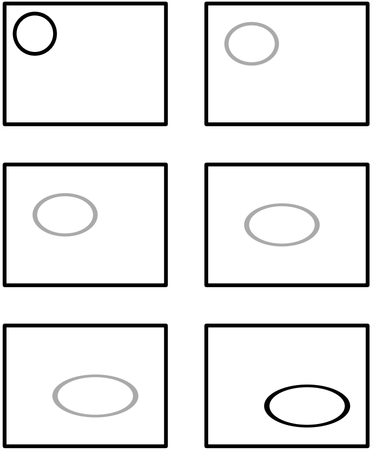
Key-Frame Animation

Key-frame animation

- Animation is a series of images/frames
- Drawing every frame is expensive
- Only draw a few, interpolate the rest

Computer key-frame animation

- Need to interpolate between frames
- Assume numerical values
  - Location (x, y, z) is common
  - Size, colour (r, g, b), pose (angles)...
- Given two values $f(t_1)$ and $f(x_2)$
- Find a general function $f(t)$



Linear Interpolation

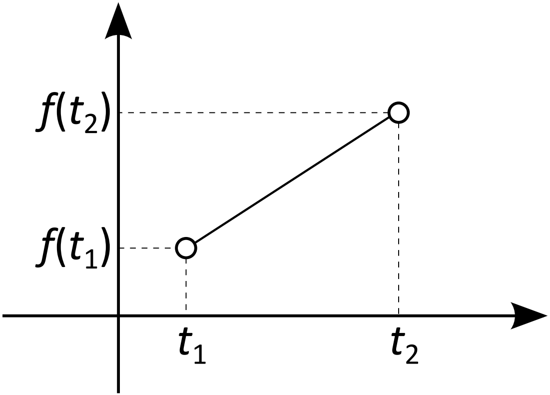
Simplest way to interpolate two values

- Model is $f(t) = at + b$
- We know that

$$f(t_1) = at_1 + b$$
$$f(t_2) = at_2 + b$$

- Easy to show that

$$a = \frac{f(t_2) - f(t_1)}{t_2 - t_1}$$
$$b = x_1 - at_1$$



Cubic Interpolation

Linear interpolation can be jerky

- Higher-order curves are smoother
- Cubics are very commonly used

$$f(t) = at^3 + bt^2 + ct + d$$

- Example: slow start and stop
- Derivative of $f(t)$ is zero at ends

$$f'(t) = 3at^2 + 2bt$$

We have four equations:

$$\begin{aligned} f(t_1) &= at_1^3 + bt_1^2 + ct_1 + d \\ f(t_2) &= at_2^3 + bt_2^2 + ct_2 + d \\ 0 &= 3at_1^2 + 2bt_1 + c \\ 0 &= 3at_2^2 + 2bt_2 + c \end{aligned}$$

Solve for a, b, c, d:

$$\begin{bmatrix} t_1^3 & t_1^2 & t_1 & 1 \\ t_2^3 & t_2^2 & t_2 & 1 \\ 3t_1^2 & 2t_1 & 1 & 0 \\ 3t_2^2 & 2t_2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} f(t_1) \\ f(t_2) \\ 0 \\ 0 \end{bmatrix}$$

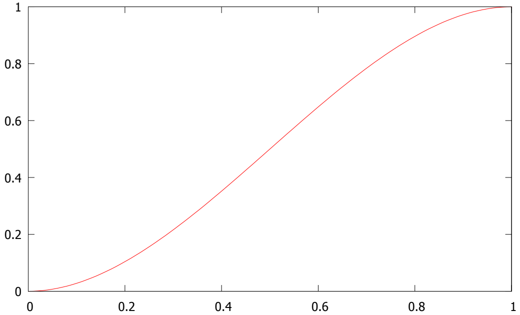
Cubic Interpolation

Example: $f(0) = 0, f(1) = 1$

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Solving gives

$$f(t) = -2t^3 + 3t^2$$



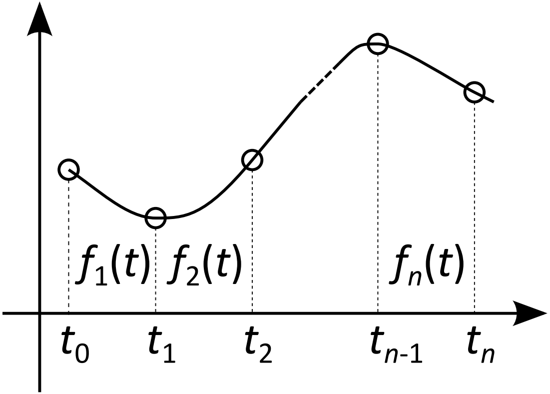
Splines

We often have a series of key-frames

- Interpolate from each one to the next
- Each interpolation is a polynomial
- This creates a *spline*
- Linear splines have sharp corners
- Smoother splines with cubic functions

Given some values $f(t_0), f(t_1), \ldots, f(t_n)$

- Find n functions f_i
- We use f_i to interpolate from t_{i-1} to t_i



Natural Cubic Splines

A 'natural' set of constraints

- Suppose we're given $n+1$ points

$$f(t_0), f(t_1), \ldots, f(t_n)$$

- We need to find n functions,

$$f_i(t) = a_i t^3 + b_i t^2 + c_i + d_i$$

- This gives $4n$ parameters
- We need $4n$ constraints
- Spline should go through the points

$$f_i(t_{i-1}) = f(t_{i-1})$$
$$f_i(t_i) = f(t_i)$$

- This gives $2n$ constraints
- Spline should be smooth at the joins

$$f_i'(t_i) = f_{i+1}'(t_i)$$
$$f_i''(t_i) = f_{i+1}''(t_i)$$

- This gives $2(n-1)$ constraints
- Usual extra constraints are

$$f_1''(0) = 0$$
$$f_n''(t_n) = 0$$

Cubic Hermite Splines

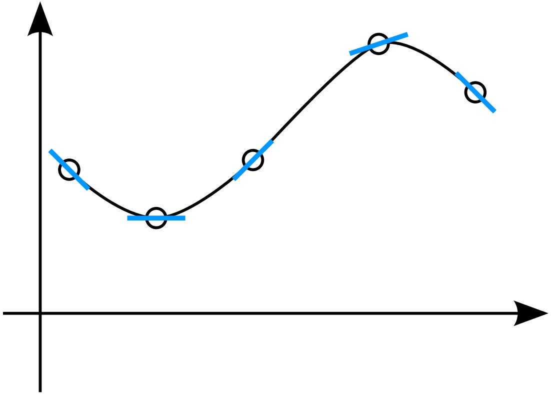
Suppose we know $f'(t_0), f'(t_1), \ldots, f'(t_n)$

- The derivative of $f_i(t)$ is

$$f_i'(t) = 3a_i t^2 + 2b_i t + c_i$$

- This gives four equations in a_i, b_i, c_i, d_i

$$f_i(t_{i-1}) = a_i t_{i-1}^3 + b_i t_{i-1}^2 + c_i t_{i-1} + d_i$$
$$f_i(t_i) = a_i t_i^3 + b_i t_i^2 + c_i t_i + d_i$$
$$f_i'(t_{i-1}) = 3a_i t_{i-1}^2 + 2b_i t_{i-1} + c_i$$
$$f_i'(t_i) = 3a_i t_i^2 + 2b_i t_i + c_i$$



Finding the Derivatives

Cubic Hermite splines require derivatives

- Can ask the user – requires more input, but gives more control
- Can estimate them by finite differences – average from both sides

$$f'(t_i) \approx \frac{1}{2}\left(\frac{f(t_{i+1}) - f(t_i)}{t_{i+1} - t_i} + \frac{f(t_i) - f(t_{i-1})}{t_i - t_{i-1}}\right)$$

- Catmull-Rom splines just use the two neighbouring points

$$f'(t_i) \approx \frac{f(t_{i+1}) - f(t_{i-1})}{t_{i+1} - t_{i-1}}$$

Approximating Splines

Splines so far are *interpolating* – they pass through the control points

- This is not necessary – the alternative is *approximating* splines
- Given n control points $p_1, p_2, \ldots, p_n$
- We define a set of *basis functions* B_i
- The spline location at time t is then

$$f(t) = \sum_{i=1}^{n} B_i(t) p_i$$

- In *cubic B-splines* the basis functions are cubics

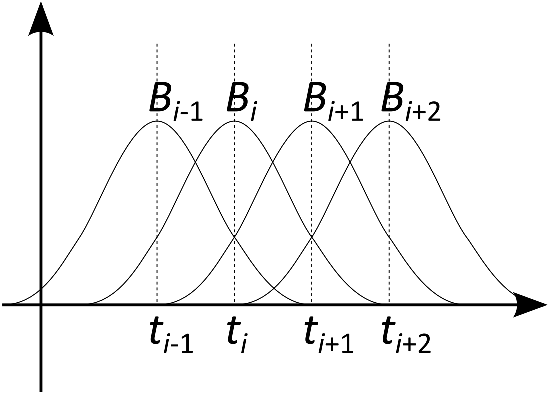
Cubic B-Splines

Consider the time from t_i to t_{i+1}

- Spline based on p_{i-1}, p_i, p_{i+1}, and p_{i+2}
- Our basis functions have the form

$$B_i = a_i t^3 + b_i t^2 + c_i t + d_i$$

- Four of these, so need 16 constraints
- The functions should be smooth
- They should have *limited support*
- They should 'swap over' smoothly



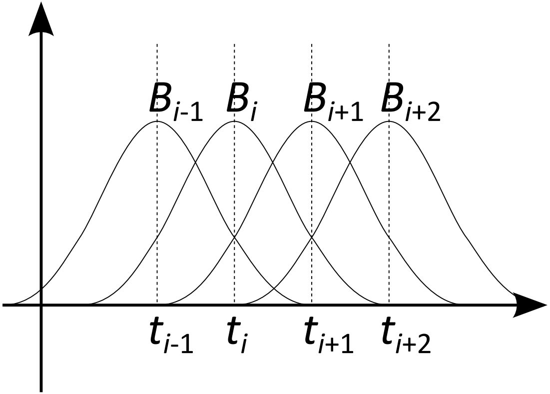
Cubic B-Splines

Consider the time from t_i to t_{i+1}

- Spline based on p_{i-1}, p_i, p_{i+1}, and p_{i+2}
- Our basis functions have the form

$$B_i = a_i t^3 + b_i t^2 + c_i t + d_i$$

- Four of these, so need 16 constraints
- The functions should be smooth
- They should have *limited support*
- They should 'swap over' smoothly



Cubic B-Spline Constraints

The curve should be continuous:

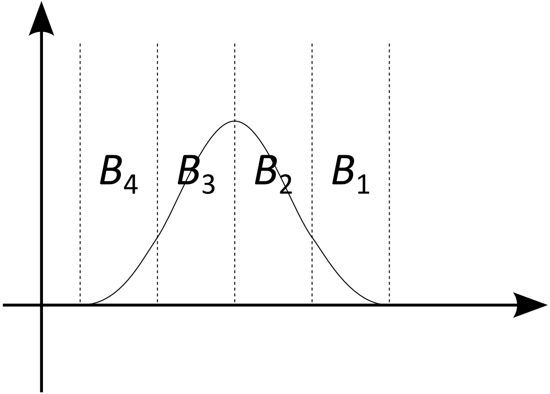
$$B_4(1) = B_3(0) \quad B_3(1) = B_2(0) \quad B_2(1) = B_1(0)$$

The curve should be smooth:

$$B_4'(1) = B_3'(0) \quad B_3'(1) = B_2'(0) \quad B_2'(1) = B_1'(0)$$
$$B_4''(1) = B_3''(0) \quad B_3''(1) = B_2''(0) \quad B_2''(1) = B_1''(0)$$

The curve should stop at the ends:

$$B_4(0) = 0 \quad B_4'(0) = 0 \quad B_4''(0) = 0$$
$$B_1(1) = 0 \quad B_1'(1) = 0 \quad B_1''(1) = 0$$



Cubic B-Spline Constraints

This gives 15 constraints

- But we have 16 parameters
- Need one more constraint
- This is that the cuves add to 1

$$B_1(t) + B_2(t) + B_3(t) + B_4(t) = 1$$

- We can now find the curves

$$B_1(t) = \frac{(1-t)^3}{6}$$

$$B_2(t) = \frac{3t^3 - 6t^2 + 4}{6}$$

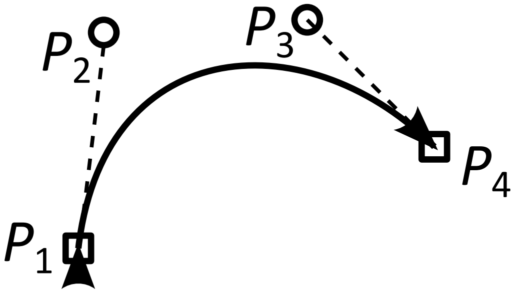
$$B_3(t) = \frac{-3t^3 + 3t^2 3t - 1}{6}$$

$$B_4(t) = \frac{t^3}{6}$$

Bézier Curves

Bézier are defined by 4 points, p_1, p_2, p_3, p_4

- Mix interpolation and approximation
- Curve starts at p_1
- It is heading towards p_2
- $|p_2 - p_1|$ gives it's starting 'speed'
- It ends up at p_4, coming from the direction of p_3
- It's final speed is $|p_4 - p_3|$



Bézier Curves

If we define a curve $B(t)$ for $t \in [0, 1]$

- $B(0) = p_1$ and $B'(0) \propto p_2 - p_1$
- $B(1) = p_4$ and $B'(1) \propto p_4 - p_3$
- This leads to:

$$
\begin{aligned}
B(t) = {} & (1-t)^3 p_1 + 3(1-t)^2 t p_2 + \\
& 3(1-t)t^2 p_3 + t^3 p_4 \\
B'(t) = {} & 3(1-t)^2 (p_2 - p_1) + \\
& 3t^2 (p_4 - p_3) + \\
& 6(1-t)t(p_3 - p_2)
\end{aligned}
$$