CS 174C Week 3

Aidan Jan

January 22, 2024

Motion Curves

- The most basic capability of an animation package is to let the user set animation variables in each frame
  - Not so easy - major HCI challenges in designing an effective user interface
  - We will not consider HCI issues
- The next is to support keyframing: Computer automatically interpolates in-between frames
- A motion curve is what you get when you plot an animation variable against time
  - The computer must come up with motion curves that interpolate your keyframe values

Different Forms of Curve Functions

- Explicit: $y = f(x)$
  - Cannot get multiple values for single x or infinite slopes
- Implicit: $f(x, y) = 0$
  - Cannot easily compare tangent vectors at joints
  - In/Out test, normals from gradient
- Parametric: $x = f_x(t), y = f_y(t), z = f_z(t)$
  - Most convenient for motion representation

Describing Curves by Means of Polynomials

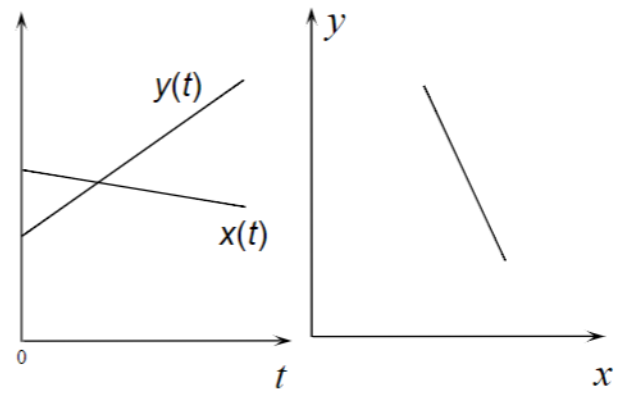
Reminder:

- L^{th} degree polynomial
- $p(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_L t^L$
- $a_0, \ldots, a_L$ are the coefficients
- L is the degree
- $(L + 1)$ is the "order" of the polynomial

Polynomial Curves of Degree 1

Parametric and implicit forms are linear

$$x(t) = at + b$$
$$y(t) = ct + d$$



Polynomial Curves of Degree 2

Parametric

- $x(t) = at^2 + 2bt + c$
- $y(t) = dt^2 + 2et + f$
- For any choice of constants
  - $a, b, c, d, e, f \to$ parabola

Rational Parametric

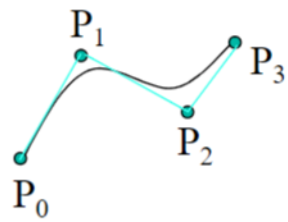
- $P(t) = \frac{P_0(1-t)^2 + 2wP_1 t(1-t) + P_2 t^2}{(1-t)^2 + 2wt(1-t) + t^2}$
- $w < 1$: ellipse
- $w = 1$: parabola
- $w > 1$: hyperbola

Curves From Geometric Constraints

Geometric Approach

- Constraints $\to$ Polynomial $\to$ Curve
- $P_0, \ldots, P_L \to$ (Curve Generation) $\to P(t)$
  - P_i: control points
  - $P_0, \ldots, P_L$: control polygon

Interpolation vs. Approximation



Bezier Curves and the De Casteljau Algorithm

Tweening

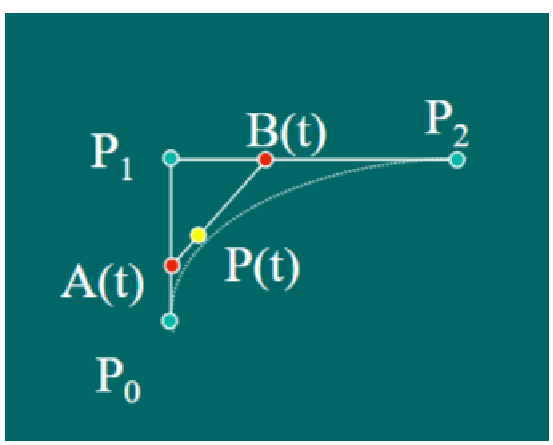
When there are two points:

- $A(t) = (1-t)P_0 + tP_1$
- $P(t) = A(t)$
- Essentially, A is a point on the line between P_0 and P_1

When there are three points:

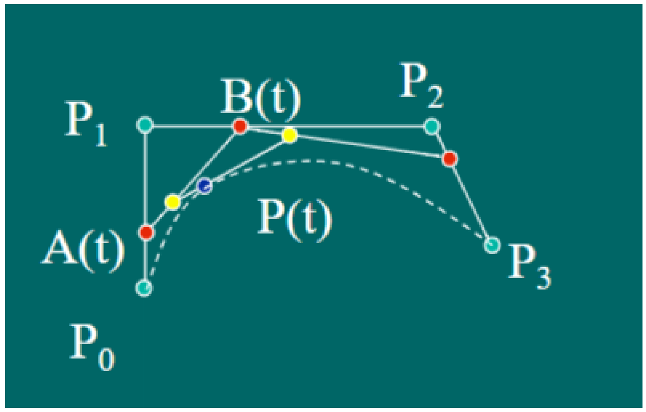
- $A(t) = (1-t)P_0 + tP_1$
- $B(t) = (1-t)P_1 + tP_2$
- $A(t)$ is a point between P_0 and P_1 and $B(t)$ is a point between P_1 and P_2.
- Now, place another point, $P(t)$ on the line between $A(t)$ and $B(t)$.
  - $P(t) = (1-t)A + tB = (1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2$

When we move the value of t from 0 to 1, $P(t)$ would move from P_0 to P_2 along a curved path, defined by the quadratic equation.



If we repeat the same process for $P(t)$ but instead of four points, then

$$P(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3$$



Cubic Berstein Polynomials