

# CS181a: Computer Animation

Curves & Curved Surfaces  
Z Sweedyk

2/4/02

CS181a Lec 03

1

## Exercise Policy

- You must submit all exercises to pass the class
- Why?

2/4/02

CS181a Lec 03

2

## Review: Curves

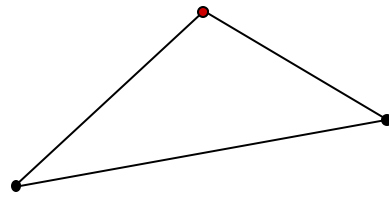
- Simple curves: Parametric cubic polynomials
- Representation
  - Constraints
  - Basis matrix
  - Blending Function
- Computation
  - Direct
  - Forward differencing
  - Recursive

2/4/02

CS181a Lec 03

3

## Compute Quadratic Bezier

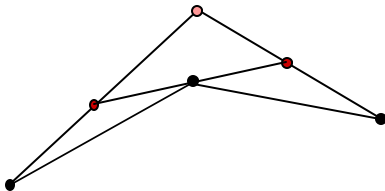


2/4/02

CS181a Lec 03

4

## Compute Quadratic Bezier

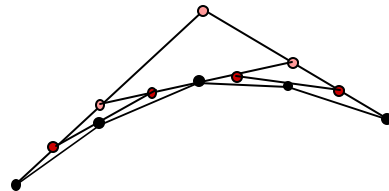


2/4/02

CS181a Lec 03

5

## Compute Quadratic Bezier



2/4/02

CS181a Lec 03

6

Why does this work?



2/4/02

CS181a Lec 03

7

Why does this work?

- **Review: Bezier**
- Review: Barycenters
- Casteljau's algorithm

2/4/02

CS181a Lec 03

8

## Properties of Bezier Curves

- Polynomial curve of degree  $n$  is defined by  $n+1$  (non-collinear) control points,  $p_0, \dots, p_n$ .
- Each  $p_i$  is an  $m$ -dimensional point for  $m \leq n$ .
- The curve lies in an  $m$ -dimensional affine space.
- Curve starts at  $p_0$  and ends at  $p_n$ .
- Etc.

2/4/02

CS181a Lec 03

9

## Bezier Curve

- For each dimension  $x$  define  $x(t)$ :  
$$x(t) = x_0 b_{n,0}(t) + x_1 b_{n,1}(t) + \dots + x_n b_{n,n}(t)$$
  
 $x_i$  is the  $x$ -coordinate of the  $i^{\text{th}}$  control point  
 $b_{n,i}$  is the  $i^{\text{th}}$  Bernstein polynomial of degree  $n$
- $b_{n,i}(t) = C(n,i) t^i (1-t)^{n-i}$

2/4/02

CS181a Lec 03

10

## Bezier Curve

$$p(t) = p_0 b_{n,0}(t) + p_1 b_{n,1}(t) + \dots + p_n b_{n,n}(t)$$

$p_i$  is the  $i^{\text{th}}$  control point

$b_{n,i}$  is the  $i^{\text{th}}$  Bernstein polynomial of degree  $n$

- $b_{n,i}(t) = C(n,i) t^i (1-t)^{n-i}$

2/4/02

CS181a Lec 03

11

## Example

- Write the quadratic polynomial that for the bezier curve with the control following control points:

-  $P_0 = (1,5)$

-  $P_1 = (2,0)$

-  $P_2 = (-1,1)$

$$X(t) = (1-t)^2 + 4t(1-t) - t^2 = -4t^2 + 2t + 1$$

2/4/02

CS181a Lec 03

12

## Exercise 1

- Write  $Y(t)$
- Sketch the curve
- Plots points on the curve using 2 levels of deCasteljau's algorithm

2/4/02

CS181a Lec 03

13

## Why does this work?

- Review: Bezier
- **Review: Barycenters**
- Casteljau's algorithm

2/4/02

CS181a Lec 03

14

## Review: Affine Spaces

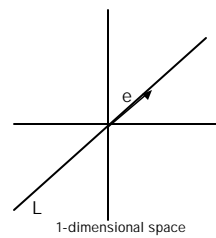
- Points + Vectors
- Vectors act on points
- Frame
- Barycenters

2/4/02

CS181a Lec 03

15

## Linear Space



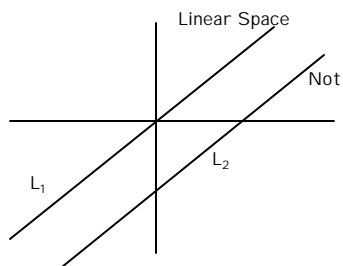
- Specified by basis vectors
- Closed under linear combination
- Contains the 0 vector
- Points  $\equiv$  Vectors

2/4/02

CS181a Lec 03

16

## Linear Spaces



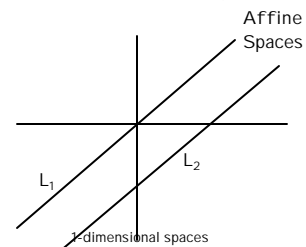
2/4/02

CS181a Lec 03

17

## Affine Space

Translation of a Linear Space



2/4/02

CS181a Lec 03

18

### Frame

Specify origin and basis vectors

2/4/02 CS181a Lec 03 19

### $\mathbb{R}^n$ : an Affine Space

$\mathbb{R}^n$ : points

+

$\mathbb{R}^n$ : vectors

Naming convention:  
 $\langle x_1, x_2, \dots, x_n \rangle$  is the vector defined from the point  $(0, 0, \dots, 0)$  to the point  $(x_1, x_2, \dots, x_n)$

2/4/02 CS181a Lec 03 20

### Vectors Act On Points

$p + \mathbf{v}$  is the point you get to by moving  $\mathbf{v}$  from the point  $p$

2/4/02 CS181a Lec 03 21

### Linear combination of vectors act on a point

2/4/02 CS181a Lec 03 22

### Linear combination of vectors act on a point

2/4/02 CS181a Lec 03 23

### Frame Dependent

2/4/02 CS181a Lec 03 24

### Affine combination of points

$\sum \alpha_i p_i$ , where  $\sum \alpha_i = 1$

2/4/02 CS181a Lec 03 25

### Affine combination of points

- Pick any origin  $p$
- Define the vector  $v_i = p_i - p$
- Move to:  $p + \sum \alpha_i v_i$
- Destination does not depend on the choice of  $p$ .

2/4/02 CS181a Lec 03 26

### Why?

$$p + \sum \alpha_i (p_i - p) = q + (p - q) + \sum \alpha_i [(p_i - q) + (q - p)]$$

$$= q + \sum \alpha_i (p_i - q) + (p - q) + \sum \alpha_i (q - p)$$

2/4/02 CS181a Lec 03 27

### Barycenter

- Barycenter is denoted  $\sum \alpha_i p_i$
- Barycenter is frame independent

2/4/02 CS181a Lec 03 28

### Computation of Barycenter

$(1/4)p + (1/2)q + (1/4)r$

2/4/02 CS181a Lec 03 29

### Exercise 2: Recursive Computation of Barycenter

Compute the barycenter  $p + q - r$

2/4/02 CS181a Lec 03 30

## Recursive Computation Bezier Curve

- For each dimension  $x$  define  $x(t)$ :  

$$x(t) = x_0 b_{n,0}(t) + x_1 b_{n,1}(t) + \dots + x_n b_{n,n}(t)$$
 $x_i$  is the  $x$ -coordinate of the  $i^{\text{th}}$  control point  
 $b_{n,i}$  is the  $i^{\text{th}}$  Bernstein polynomial of degree  $n$
- $b_{n,i}(t) = C(n,i) t^i (1-t)^{n-i}$
- NOTE that for any  $t$ ,  $\sum_{i=0,\dots,n} b_{n,i}(t) = 1$
- $p(t) = \sum_{i=0,\dots,n} b_{n,i}(t) p_i(t)$  is a barycenter

2/4/02

CS181a Lec 03

31

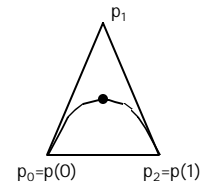
## Recursive computation of Barycenter

$$p(t) = (1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2$$

$$p(0) = p_0$$

$$p(1) = p_2$$

$$p(1/2) = (1/4)p_0 + (1/2)p_1 + (1/4)p_2$$



2/4/02

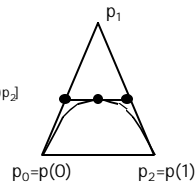
CS181a Lec 03

32

## Recursive computation of Barycenter

$$p(1/2) = (1/4)p_0 + (1/2)p_1 + (1/4)p_2$$

$$p(1/2) = [(1/4)p_0 + (1/4)p_1] + [(1/4)p_1 + (1/4)p_2]$$



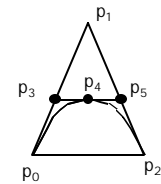
2/4/02

CS181a Lec 03

33

## Recursive computation of Barycenter

Let  $p'(t)$  be the bezier curve defined by  $p_0, p_3, p_4$ .  
Then  $p'(t) = p((1/2)t)$ .



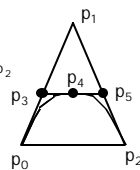
2/4/02

CS181a Lec 03

34

## Recursive computation of Barycenter

- Let  $p'(t)$  be the bezier curve defined by  $p_0, p_3, p_4$ .
- $p'(t) = (1-t)^2 p_0 + 2t(1-t)p_3 + t^2 p_4$
- Note  $p_3 = (1/2)p_0 + (1/2)p_1$  &  $p_4 = (1/4)p_0 + (1/2)p_1 + (1/4)p_2$
- $p'(t) = (1 - (1/2)t)^2 p_0 + t(1 - (1/2)t) p_1 + (1/4)t^2 p_2$
- $p'(t) = p((1/2)t)$



2/4/02

CS181a Lec 03

35

## Why does this happen?

- Deeper mathematics: Polar forms

2/4/02

CS181a Lec 03

36

## Surfaces

- Explicit
- Implicit
- **Parametric**
- Subdivision

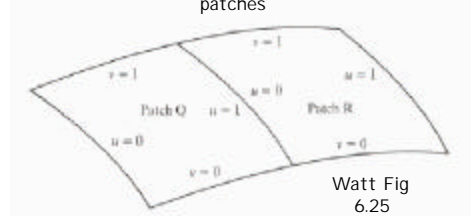
2/4/02

CS181a Lec 03

37

## Bezier Surfaces

Piece together smaller patches



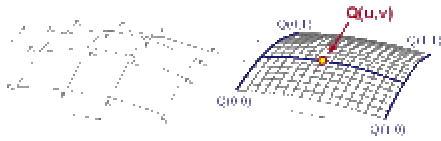
2/4/02

CS181a Lec 03

38

## Bezier Patch

Tensor product of Bezier curves



Watt Figure 6.11

2/4/02

CS181a Lec 03

39

## Properties of Bezier Patch

- Interpolates four corners
- Convex hull property
- Local control
- Boundary conditions are difficult to maintain
- Patches are necessarily quadrilaterals

2/4/02

CS181a Lec 03

40

## Subdivision Surfaces

Smooth surface is the limit of a subdivision process



Zorin & Schroeder  
SIGGRAPH 99  
Camera Notes

2/4/02

CS181a Lec 03

41

## Surface Subdivision

Refine polygon mesh via subdivision rules

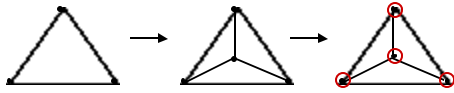
1. Split face into multiple faces
2. Reposition vertices

2/4/02

CS181a Lec 03

42

## Split Vertices

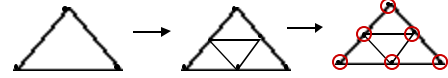


2/4/02

CS181a Lec 03

43

## Split Edges

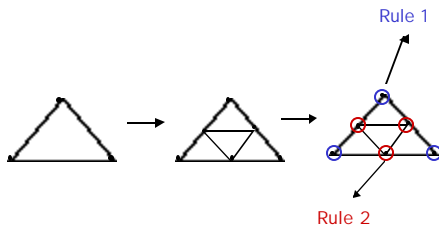


2/4/02

CS181a Lec 03

44

## Loop Algorithm



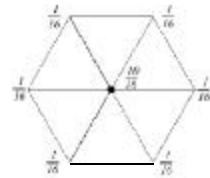
2/4/02

CS181a Lec 03

45

## Loop Subdivision

Rule 1: Existing vertex  
Case: Internal, valence 6



Note: Weights are applied to vertex positions in the old mesh.

2/4/02

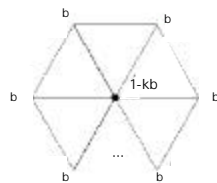
CS181a Lec 03

46

## Loop Subdivision

Rule 1: Existing Vertex  
Case: Internal, valence  $k$

$k > 3$ :  $b = 3/8k$   
 $k = 3$ :  $b = 3/16$



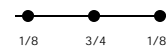
2/4/02

CS181a Lec 03

47

## Loop Subdivision

Rule 1: Existing vertex  
Case: Boundary



2/4/02

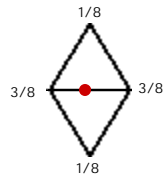
CS181a Lec 03

48

## Loop Subdivision

Rule 2: New vertex

Case: Internal



Note: Weights are applied to vertex positions in the old mesh.

2/4/02

CS181a Lec 03

49

## Loop Subdivision

Rule 2: New vertex

Case: Boundary



2/4/02

CS181a Lec 03

50

## Project 1

- Let's go to course web site for details

2/4/02

CS181a Lec 03

51

## Exercise 1: \_\_\_\_\_

- Write the quadratic polynomial for the bezier curve with the control points:  $P_0=(3,2)$ ,  $P_1=(0,1)$ ,  $P_2=(-1,0)$

2/4/02

CS181a Lec 03

52

## Exercise 1 cont.:

- \_\_\_\_\_
- Sketch the curve

2/4/02

CS181a Lec 03

53

## Exercise 1 cont:

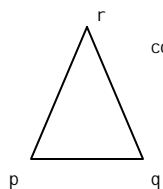
- \_\_\_\_\_
- Plots points on the curve using 2 levels of deCasteljau's algorithm

2/4/02

CS181a Lec 03

54

## Exercise 2: \_\_\_\_\_



Use recursive  
computation to find  
the barycenter  
 $p + q - r$