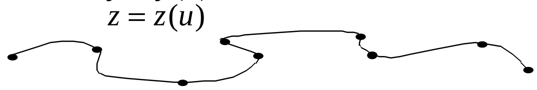
Parametric Curves

Lecture 5

Goal : Better Approximation of a Curve than Piecewise Linear

- Parametric Curve Segments

$$x = x(u)$$
$$y = y(u) \qquad u_1 \le u \le u_2$$
$$z = z(u)$$



- Cubic Polynomial
  - Reasonable Compromise
    - Flexibility (without “ringing”)
    - Speed of Computation
  - Easy Manipulation / Differentiation

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x \qquad 0 \le u \le 1$$

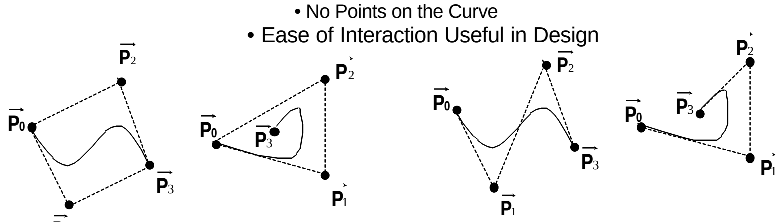
$$\frac{d}{du} x(u) = x'(u) = 3a_x u^2 + 2b_x u + c_x \qquad 0 \le u \le 1$$

1

Splines

Lecture 5

- Spline Curves
  - Defined by a set of Control Points
    - All Points on the Curve
    - Some Points on the Curve
    - No Points on the Curve
  - Ease of Interaction Useful in Design



Convex Hull

Control Graph

- Interpolate => Control points on curve
- Approximate => Control points not on curve

2

Splines

Lecture 5

- Spline Representation
  - Boundary Conditions
  - Basis Matrix
  - Blending Functions

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x \qquad 0 \le u \le 1$$

$$= \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} = [\mathbf{U}][\mathbf{C}]$$

$$\text{Let } [\mathbf{C}] = [\mathbf{M}][\mathbf{G}]$$

3

Splines

Lecture 5

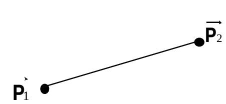
$$[\mathbf{G}] = \text{Geometric Constraints (Boundary Conditions)}$$

$$[\mathbf{M}] = \text{Basis Matrix (Different for each type of spline)}$$

- The Combination of the Basis Matrix with the Geometric Constraints Give Rise to the Polynomial Coefficients

$$x(u) = [\mathbf{U}][\mathbf{C}] = [\mathbf{U}][\mathbf{M}][\mathbf{G}]$$

- The Combination of the Parametric Matrix with the Basis Matrix Gives rise to the Blending Functions



Linear Interpolation

$$\vec{\mathbf{P}}(\mathbf{u}) = \underbrace{(1-u)}\vec{\mathbf{P}}_1 + \underbrace{(u)}\vec{\mathbf{P}}_2$$

Blending Functions

4

Splines

Lecture 5

- Determine Blending Functions for Splines

$$[\mathbf{P(u)}] = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = [\mathbf{U}][\mathbf{M}][\mathbf{G}] = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix}$$

$$\begin{aligned} x(u) = {} & (u^3 m_{11} + u^2 m_{21} + u m_{31} + m_{41}) g_1 + \\ & (u^3 m_{12} + u^2 m_{22} + u m_{32} + m_{42}) g_2 + \\ & (u^3 m_{13} + u^2 m_{23} + u m_{33} + m_{43}) g_3 + \\ & (u^3 m_{14} + u^2 m_{24} + u m_{34} + m_{44}) g_4 \end{aligned}$$

- Spline Curve is a Weighted Sum of Elements in the Geometry Matrix

5

Hermite Splines

Lecture 5

- Boundary Conditions are :
  - Two Endpoints
  - Tangents at Each End

$$\vec{P}(0) = \text{Endpoint @ } u = 0 \qquad \vec{P}'(0) = \text{Tangent @ } u = 0$$
$$\vec{P}(1) = \text{Endpoint @ } u = 1 \qquad \vec{P}'(1) = \text{Tangent @ } u = 1$$

$$[\mathbf{P(u)}] = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}$$

$$[\mathbf{P(0)}] = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix} \qquad [\mathbf{P(1)}] = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{bmatrix}$$

6

Hermite Splines

$$\text{Hermite Basis Matrix} = [\mathbf{M}_H] = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$[\mathbf{P(u)}] = \begin{bmatrix} x(u) \\ y(u) \\ z(u) \end{bmatrix} = [\mathbf{U}][\mathbf{M}_H][\mathbf{G}_H] = [u3 \quad u2 \quad u \quad 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

$$\mathbf{P(u)} = (2u^3 - 3u^2 + 1)\vec{\mathbf{P}}_1 + (-2u^3 + 3u^2)\vec{\mathbf{P}}_2 + (u^3 - 2u^2 + u)\vec{\mathbf{P}}_3 + (u^3 - u^2)\vec{\mathbf{P}}_4$$

where $\vec{P}(0) = \vec{\mathbf{P}}_1 \quad \vec{P}'(0) = \vec{\mathbf{P}}_3$

$\vec{P}(1) = \vec{\mathbf{P}}_2 \quad \vec{P}'(1) = \vec{\mathbf{P}}_4$

Bezier Curves

- Developed by Pierre Bezier
- Automobile Design for Renault

- Arbitrary Number of Control Points
- First and Last Control Points Lie ON the Curve

Bezier Curves

Lecture 5

- Given (n+1) Control Points the Bezier Curve is Determined by :

$$\vec{\mathbf{P}}(u) = \sum_{k=0}^{n} \vec{\mathbf{P}}_k \, B_{k,n}(u) \qquad 0 \le u \le 1$$

where $B_{k,n}(u)$ are Berstein Polynomials

$$B_{k,n}(u) = \frac{n!}{k!(n-k)!} u^k (1-u)^{n-k}$$

Would like the highest order of u to be 3 (u^3), so k=3 means n=3

Note: n=3 implies 4 control points

Bezier Curves

Lecture 5

- Given (n+1) Control Points the Bezier Curve is Determined by :

$$\vec{\mathbf{P}}(u) = \sum_{k=0}^{3} \vec{\mathbf{P}}_k \, B_{k,3}(u) \qquad 0 \le u \le 1$$

$$\vec{\mathbf{P}}(u) = (1-u)^3 \vec{\mathbf{P}}_0 + 3u(1-u)^2 \vec{\mathbf{P}}_1 + 3u^2(1-u) \vec{\mathbf{P}}_2 + u^3 \vec{\mathbf{P}}_3$$

$$[\mathbf{P}(u)] = [\mathbf{U}][\mathbf{M}_B][\mathbf{G}_B]$$

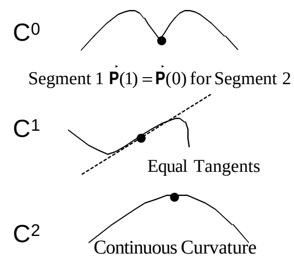
$$[\mathbf{P}(u)] = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

$$\begin{aligned} \vec{\mathbf{P}}_0 &= \vec{\mathbf{P}}(0) \\ \vec{\mathbf{P}}_1 &= \vec{\mathbf{P}}(1) \\ \dot{\mathbf{P}}_2 &= \dot{\mathbf{P}}'(0) = 3(\dot{\mathbf{P}}_1 - \dot{\mathbf{P}}_0) \\ \vec{\mathbf{P}}_3 &= \vec{\mathbf{P}}'(1) = 3(\vec{\mathbf{P}}_3 - \vec{\mathbf{P}}_2) \end{aligned}$$

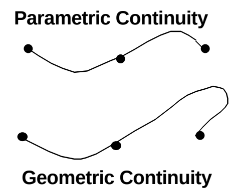
Curve Segment Continuity

Lecture 5

- Parametric Continuity
  - Zero-th Order, C^0
    Curve segments meet (join point)
  - 1st Order, C^1
    1st Derivatives are equal at join point
  - 2nd Order, C2
    2nd Derivatives are equal at join point



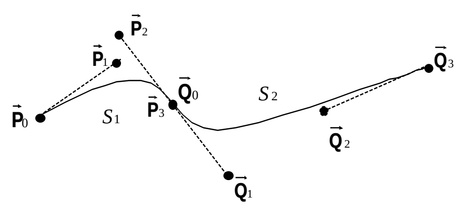
- Geometric Continuity
  - Zero-th Order, G^0
    Curve segments meet (join point)
  - 1st Order, G^1
    1st Derivatives are proportional at join point
  - 2nd Order, G2
    2nd Derivatives are proportional at join point



11

Bezier Curves Segments

Lecture 5



- Zero-th Order Continuity Achieved by : $\vec{\mathbf{Q}}_0 = \vec{\mathbf{P}}_3$
- C^1 Continuity Achieved by : $\vec{\mathbf{Q}}_1 = \vec{\mathbf{P}}_3 + (\vec{\mathbf{P}}_3 - \vec{\mathbf{P}}_2)$
  All Three Points Are Collinear and Equally Spaced
- C^2 Continuity Achieved by : $\vec{\mathbf{Q}}_2 = \vec{\mathbf{P}}_1 + 4(\vec{\mathbf{P}}_3 - \vec{\mathbf{P}}_2)$

C^2 Continuity May Be Too Restrictive Since It Leaves Only Q_3 for Adjustment

12

Splines

Lecture 5

- Blending Functions
  - Everywhere non-zero
  - ALL control points effect curve

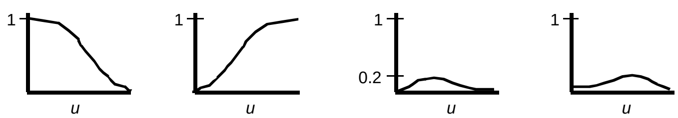
13

Splines

Lecture 5

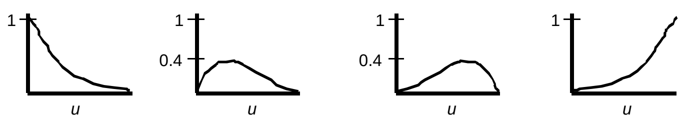
- Blending Functions

Hermite



$$\vec{\mathbf{P}}(u) = (2u^3 - 3u^2 + 1)\vec{\mathbf{P}}_0 + (-2u^3 + 3u^2)\vec{\mathbf{P}}_1 + (u^3 - 2u^2 + u)\vec{\mathbf{P}}_2 + (u^3 - u^2)\vec{\mathbf{P}}_3$$

Bezier



$$\vec{\mathbf{P}}(u) = (1-u)^3\vec{\mathbf{P}}_0 + 3u(1-u)^2\vec{\mathbf{P}}_1 + 3u^2(1-u)\vec{\mathbf{P}}_2 + u^3\vec{\mathbf{P}}_3$$

14

Splines

Lecture 5

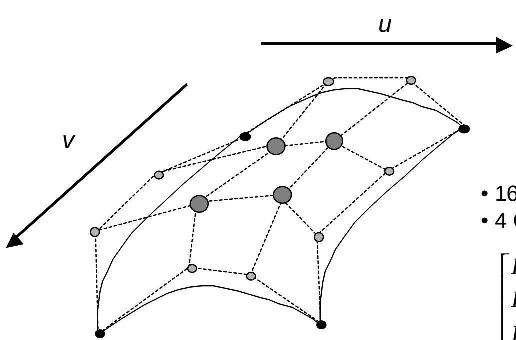
- B-splines
  - Large class of approximating splines
    - Uniform, Non-rational
    - Non-uniform,Non-rational
    - Non-uniform, Rational (NURBS)
  - Weighted sums of polynomial basis functions
  - Local curve control
  - Degree of blending polynomial independent of number of control points
  - Much more complex

15

Bezier Patches

Lecture 5

- Parametric Curve Along Two Dimensions



- 16 Control Points
- 4 Corner Points on Curve

$$\begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix}$$

16

Bezier Patches

Lecture 5

Curve : $\vec{\mathbf{P}}(u) = (1-u)^3\vec{\mathbf{P}}_0 + 3u(1-u)^2\vec{\mathbf{P}}_1 + 3u^2(1-u)\vec{\mathbf{P}}_2 + u^3\vec{\mathbf{P}}_3$

$$[\mathbf{P}(u)] = [u3 \quad u2 \quad u \quad 1][\mathbf{M}_B]\begin{bmatrix}\vec{\mathbf{P}}_0\\ \vec{\mathbf{P}}_1\\ \vec{\mathbf{P}}_2\\ \vec{\mathbf{P}}_3\end{bmatrix} \qquad [\mathbf{M}_B] = \begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 3 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}$$

$$\vec{\mathbf{P}}'(0) = 3(\vec{\mathbf{P}}_1 - \vec{\mathbf{P}}_0)$$
$$\vec{\mathbf{P}}'(1) = 3(\vec{\mathbf{P}}_3 - \vec{\mathbf{P}}_2)$$

Surface : $[\mathbf{P}(u,v)] = [u^3 \quad u^2 \quad u \quad 1][\mathbf{M}_B][\mathbf{P}][\mathbf{M}_B]^T\begin{bmatrix}v^3\\ v^2\\ v\\ 1\end{bmatrix} \qquad [\mathbf{P}] = \begin{bmatrix}P_{00} & P_{01} & P_{02} & P_{03}\\ P_{10} & P_{11} & P_{12} & P_{13}\\ P_{20} & P_{21} & P_{22} & P_{23}\\ P_{30} & P_{31} & P_{32} & P_{33}\end{bmatrix}$

17

Bezier Patches

Lecture 5

- Tangent / Control Point Relationships

$$\vec{\mathbf{P}}(0,0) = \vec{\mathbf{P}}_{00}$$

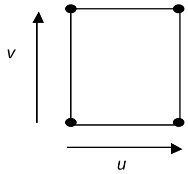
$$\vec{\mathbf{P}}_v(0,0) = 3(\vec{\mathbf{P}}_{01} - \mathbf{P}_{00})$$
$$\vec{\mathbf{P}}_u(0,0) = 3(\vec{\mathbf{P}}_{10} - \mathbf{P}_{00})$$

$$\vec{\mathbf{P}}_{uv}(0,0) = 9(\vec{\mathbf{P}}_{00} - \vec{\mathbf{P}}_{01} - \vec{\mathbf{P}}_{10} + \vec{\mathbf{P}}_{11})$$

18

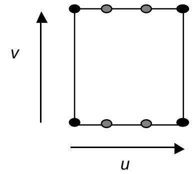
Bezier Patches

Corner Points



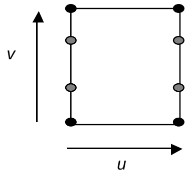
$$\vec{\mathbf{P}}(0,0) = \vec{\mathbf{P}}_{00}$$

Tangent Vectors in *u*



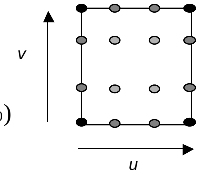
$$\vec{\mathbf{P}}_u(0,0) = 3(\vec{\mathbf{P}}_{01} - \vec{\mathbf{P}}_{00})$$

Tangent Vectors in *v*



$$\vec{\mathbf{P}}_v(0,0) = 3(\vec{\mathbf{P}}_{01} - \vec{\mathbf{P}}_{00})$$

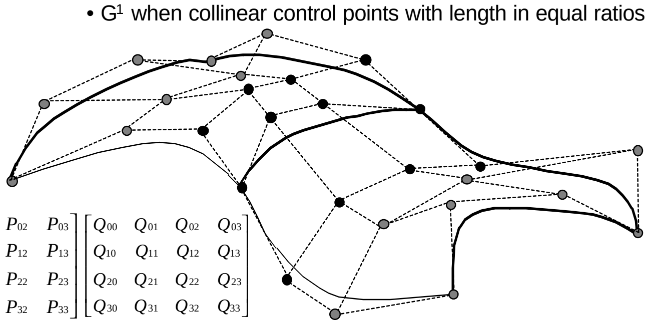
Twist Vectors



$$\vec{\mathbf{P}}_{uv}(0,0) = 9(\vec{\mathbf{P}}_{00} - \vec{\mathbf{P}}_{01} - \vec{\mathbf{P}}_{10} + \vec{\mathbf{P}}_{11})$$

Bezier Patches

- Continuity Between Patches
  - C^0 and G^0 by using common control points along edge
  - G^1 when collinear control points with length in equal ratios



$$\begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \begin{bmatrix} Q_{00} & Q_{01} & Q_{02} & Q_{03} \\ Q_{10} & Q_{11} & Q_{12} & Q_{13} \\ Q_{20} & Q_{21} & Q_{22} & Q_{23} \\ Q_{30} & Q_{31} & Q_{32} & Q_{33} \end{bmatrix}$$

Subdivision Surfaces

Lecture 5

- Create smooth surfaces out of arbitrary meshes
- The need to generalize spline patch model to arbitrary surfaces
- Define a smooth surface as the limit of a sequence of successive refinements

21

Subdivision Surfaces

Lecture 5

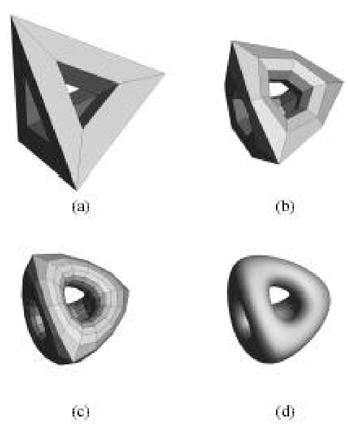


Figure 3: Recursive subdivision of a topologically complicated mesh: (a) the control mesh; (b) after one subdivision step; (c) after two subdivision steps; (d) the limit surface.

22

Subdivision Surfaces

Lecture 5

Many Attractive Features

- Arbitrary topology
- Scalability LOD
- Uniformity
- Numerical Quality
- Code Simplicity

23

Hierachical Modeling

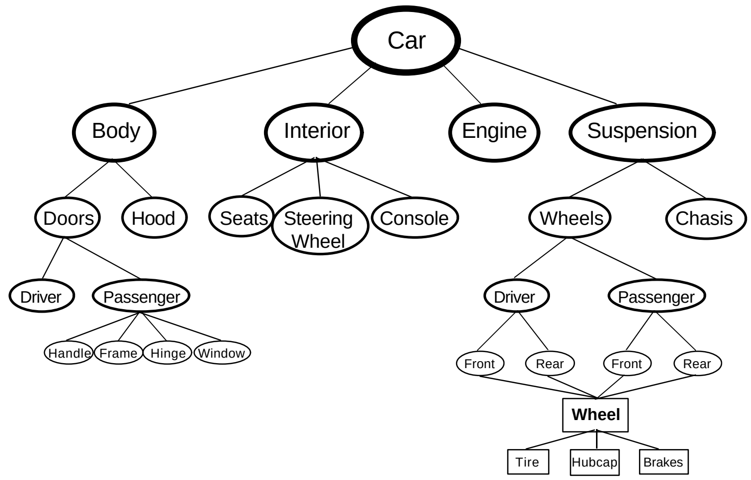
Lecture 5

- Construct complex objects from modular subsystems
- Increase storage economy
  - Subsytems (instances) stored once
  - Transformations matricies stored for each invocation
- Easy object updates
  - Update instance
  - Chahnges propagated to each use of the instance
- Graphical representation of modeled object

24

Hierarchical Modeling

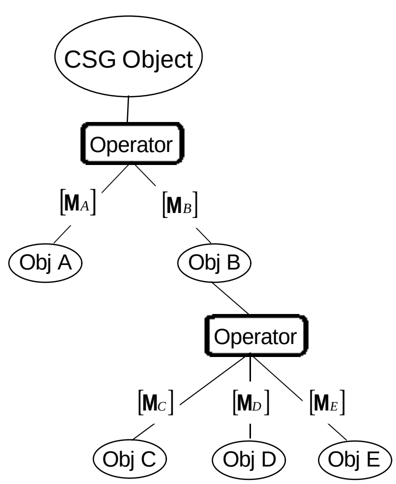
Lecture 5



25

Hierarchical Modeling

Lecture 5



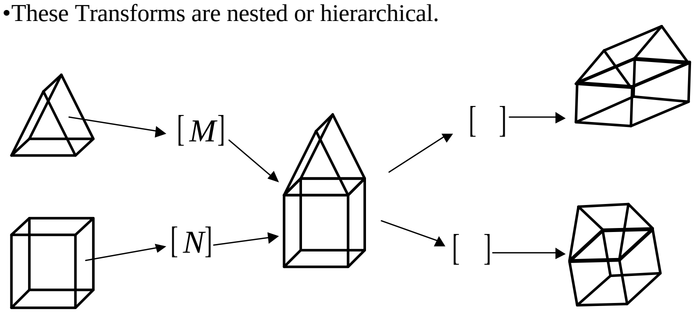
26

Hierarchical Modeling

Lecture 5

Allows the building of complex scenes with a basic set of 3-D primitives

- Each 3-D object can be defined in it's own, convenient, space.
- Transforms will allow us to assemble the primitives
- These Transforms are nested or hierarchical.



27

Hierarchical Modeling

Lecture 5

Example code:

```
Procedure Scene()
    House(B)
    House(D)
end

Procedure House(E)
    Prism(E•M)
    Cube(E•M)
end

Procedure Cube(F)
    …
end

Procedure Prism(F)
    …
end
```

28

Hierarchical Modeling

Lecture 5

Matrix Stack:

Start with identity 4 x 4

Each transform used is multiplied into the 4 x4

Special commands help manipulate the Matrix

- Push - make a copy of the 4 x 4 and save it
- Pop - restore the saved copy of the 4 x 4

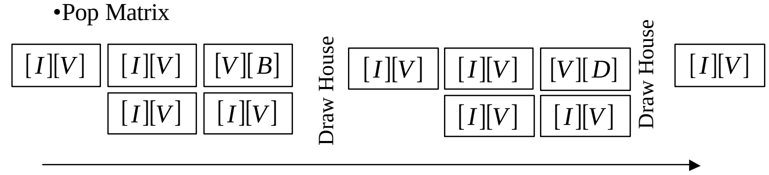
29

Hierarchical Modeling

Lecture 5

Draw Scene:

- First transform should be the viewing Transform
  This sets up the camera - all objects get transformed by this
- Push this matrix
- Multiply in B matrix.
- Draw House
- Pop matrix
- Push this matrix
- Multiply in D matrix
- Draw House
- Pop Matrix



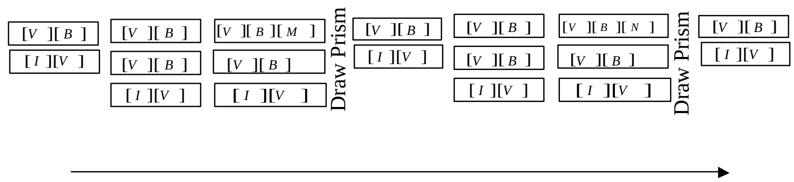
30

Hierarchical Modeling

Lecture 5

Draw House:

- •Push matrix
- •Multiply in M matrix
- •Draw Prism
- •Pop matrix
- •Push matrix
- •Multiply in N matrix
- •Draw Cube
- •Pop matrix



31