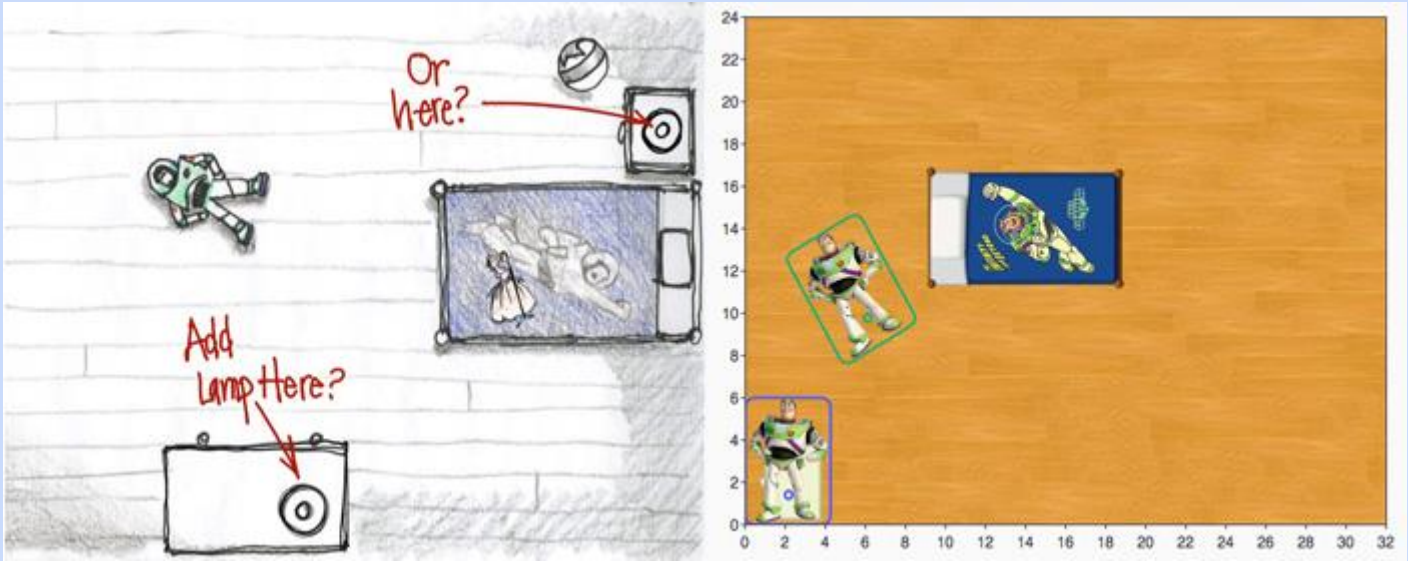
Math in Movies

Giovi Moriarty

Animation

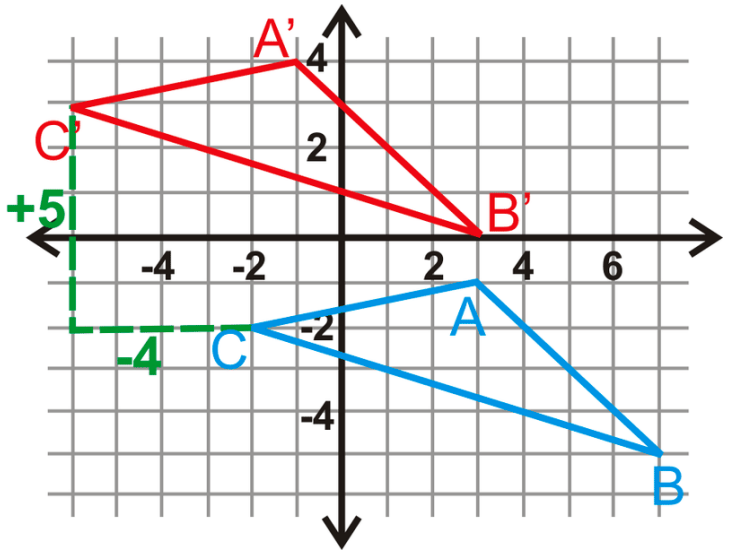
- Geometric transformations
- Curves & surfaces
- Hair simulation
- Patterns & textures

Geometric Transformations



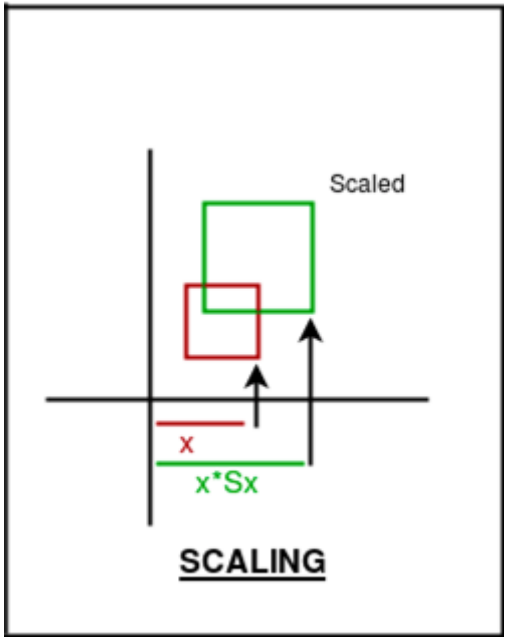
Geometric Transformations

- In the beginning, to determine position of objects, use x-y coordinate system
- Translation: moving an object along a coordinate plane from point A=(x_1, y_1) to point B=(x_2, y_2)
  - $x_1=x_0+a$ and $y_1=y_0+b$



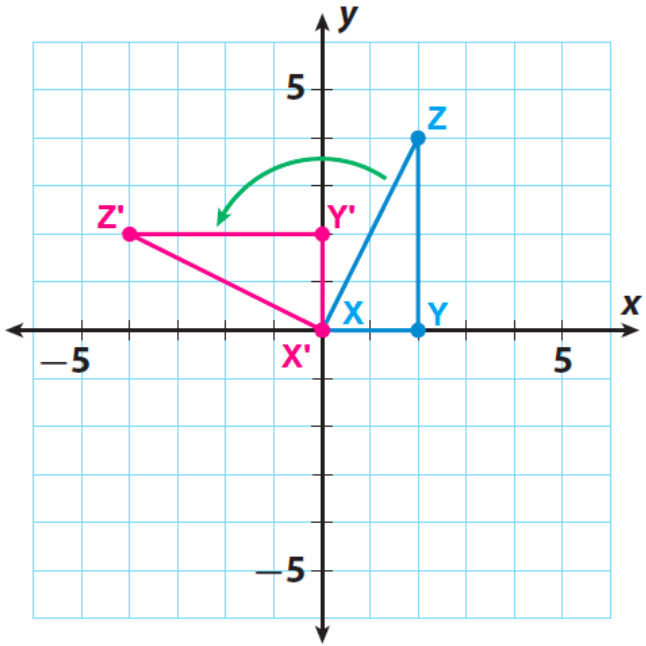
Geometric Transformations

- In the beginning, to determine position of objects, use x-y coordinate system
- Translation: moving an object along a coordinate plane from point A=(x_1, y_1) to point B=(x_2, y_2)
  - $x_1=x_0+a$ and $y_1=y_0+b$
- Scaling: resizing an object
  - $x_1=a*(x_0)$ and $y_1=b*(y_0)$
  - Also used to stretch vertically and horizontally



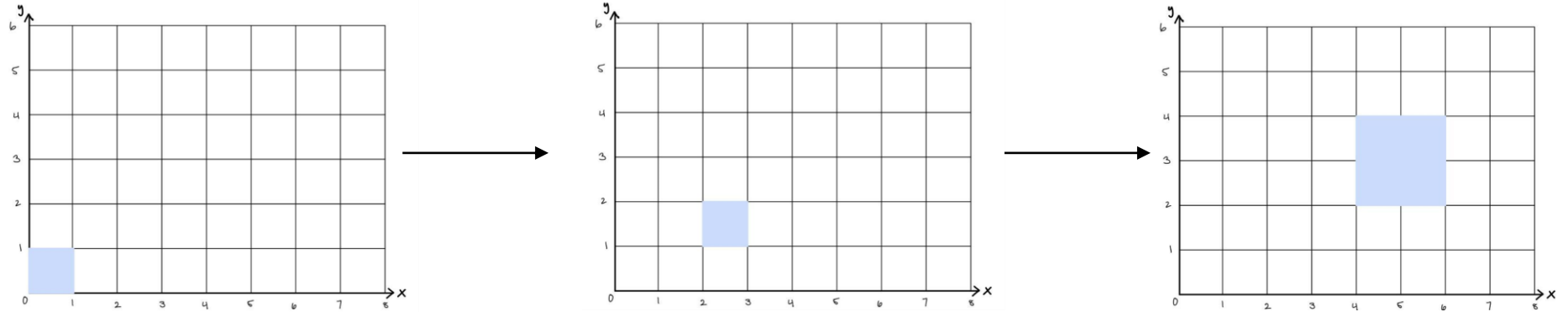
Geometric Transformations

- In the beginning, to determine position of objects, use x-y coordinate system
- Translation: moving an object along a coordinate plane from point A=(x_1, y_1) to point B=(x_2, y_2)
  - $x_1=x_0+a$ and $y_1=y_0+b$
- Scaling: resizing an object
  - $x_1=a*(x_0)$ and $y_1=b*(y_0)$
  - Also used to stretch vertically and horizontally
- Rotation: rotating an object about a point
  - $x_1=\cos(\theta)x_0-\sin(\theta)y_0$ and $y_1=\sin(\theta)y_0+\cos(\theta)x_0$



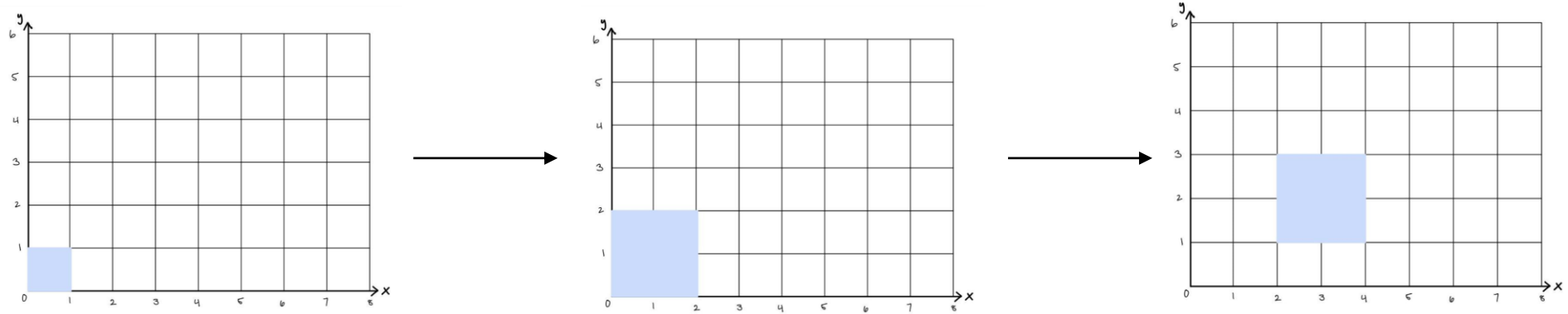
Geometric Transformations: Commutativity

- Commutative operations (order doesn't matter)
  - Combining two translations (essentially an addition operation)
- Non-commutative operations (order does matter)
  - Scaling combined with translations
  - Example:
    - Translate: 2 in X and 1 in Y then scale: factor of 2

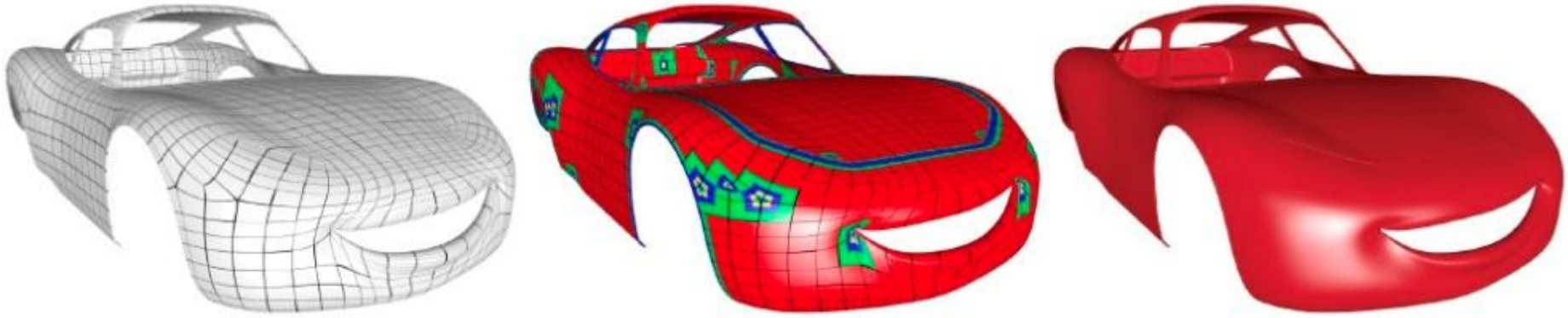


Geometric Transformations: Commutativity

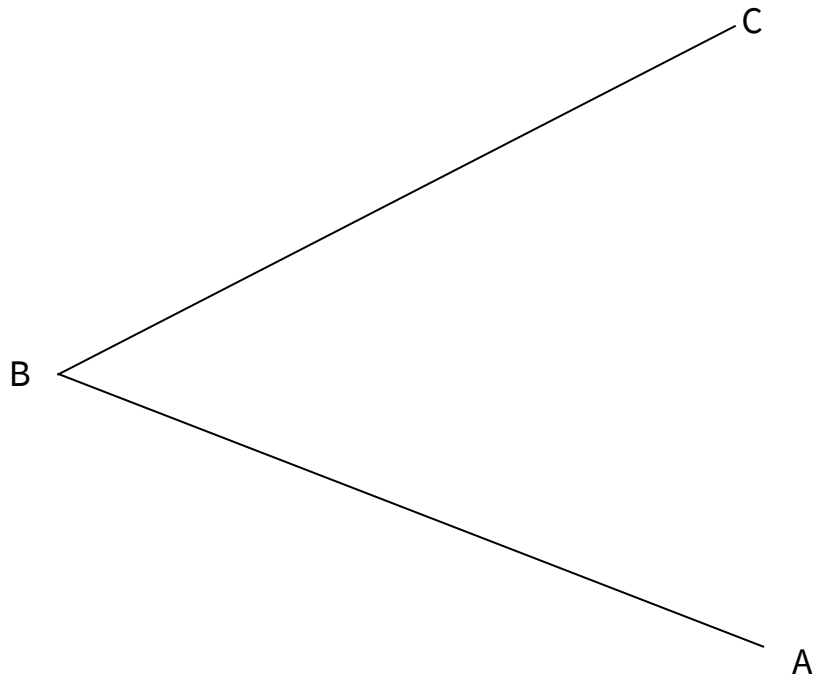
- Commutative operations (order doesn't matter)
  - Combining two translations (essentially an addition operation)
- Non-commutative operations (order does matter)
  - Scaling combined with translations
  - Example:
    - Scale: factor of 2 then translate: 2 in X and 1 in Y



Creating Curves and Surfaces

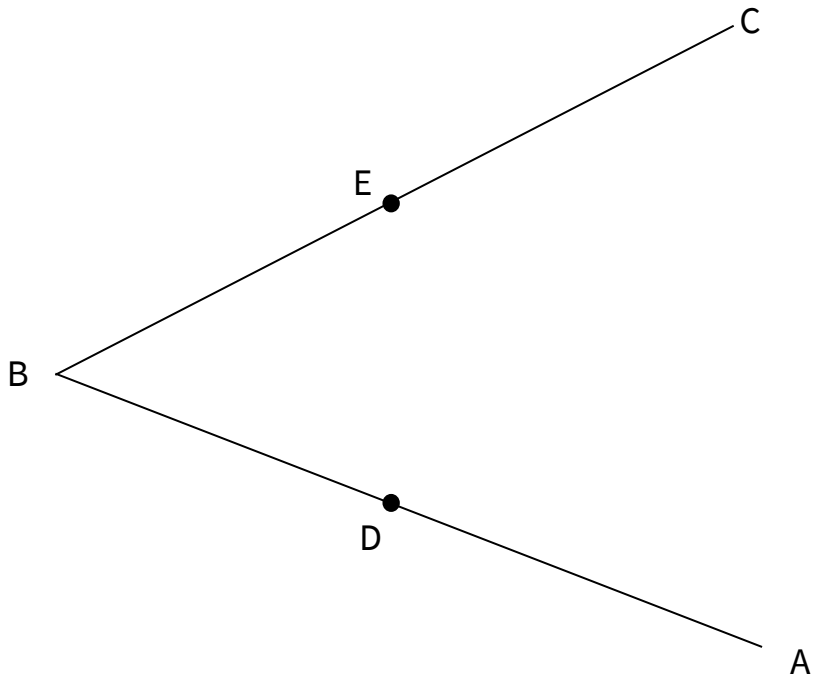


Curves & Surfaces: Midpoint Method



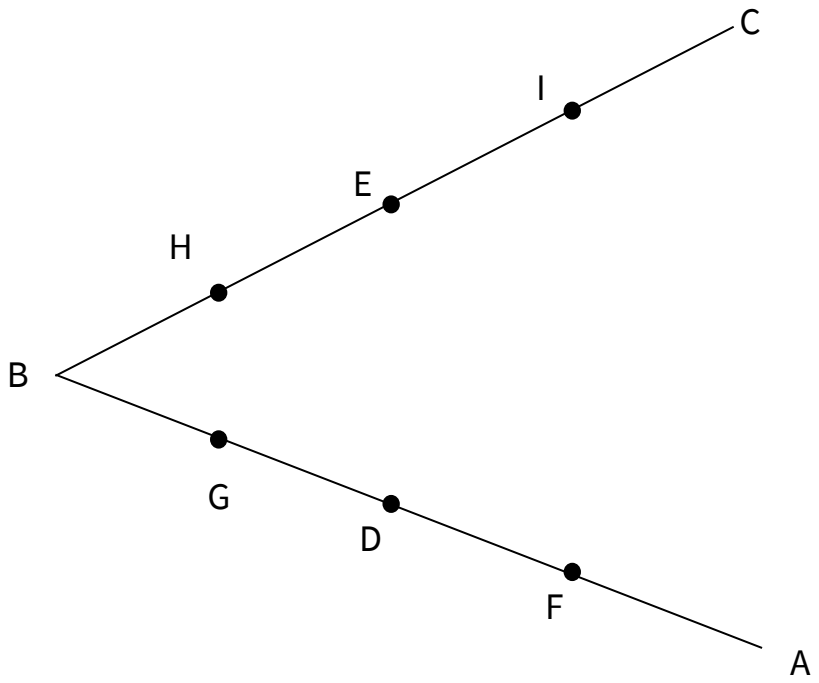
Curves & Surfaces: Midpoint Method

1) Find midpoint of AB (called D) and BC (called E)



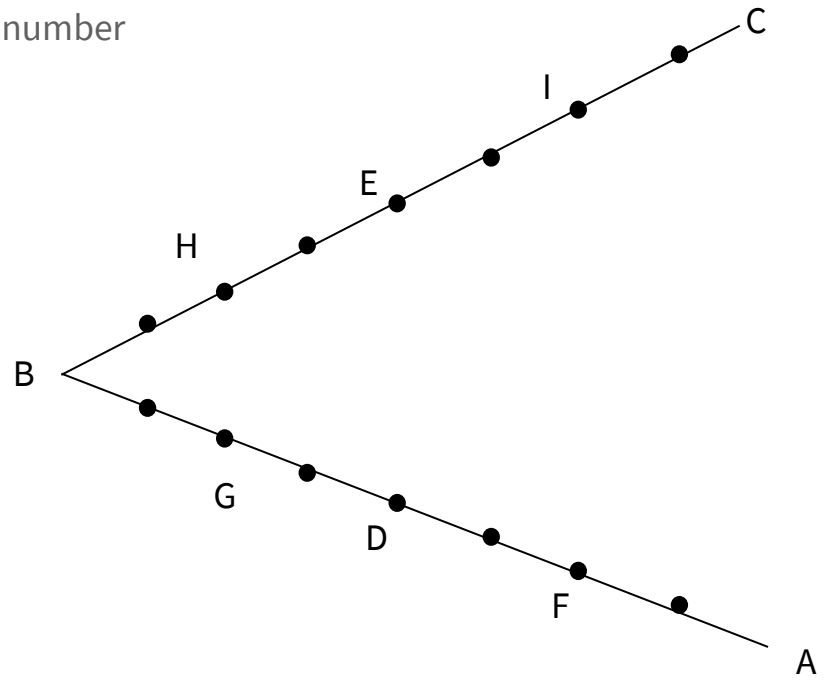
Curves & Surfaces: Midpoint Method

1) Find midpoint of AB (called D) and BC (called E)
2) FInd midpoint of AD, DB, BE, and EC



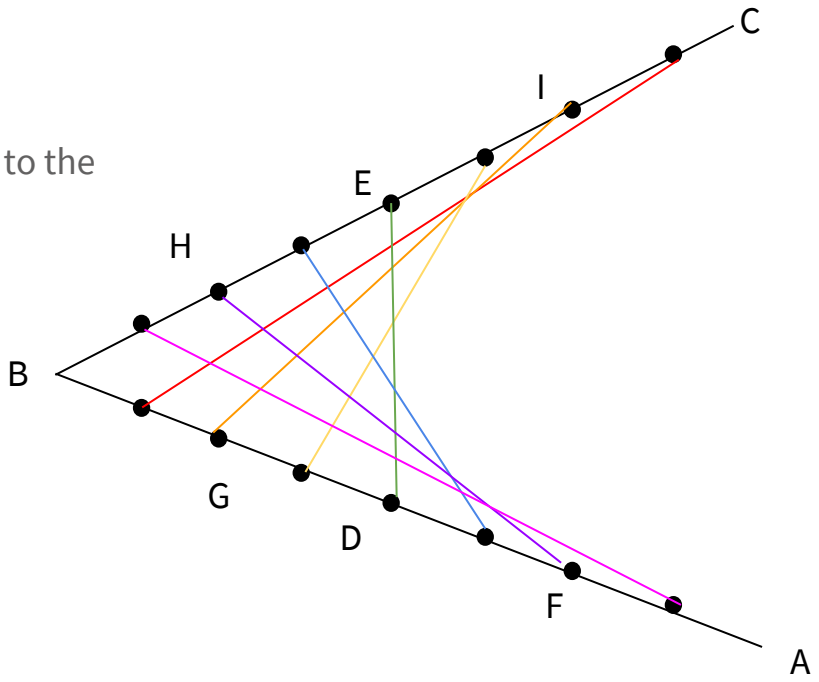
Curves & Surfaces: Midpoint Method

1) Find midpoint of AB (called D) and BC (called E)
2) FInd midpoint of AD, DB, BE, and EC
3) Continue to find midpoints of segments until desired number of points is reached



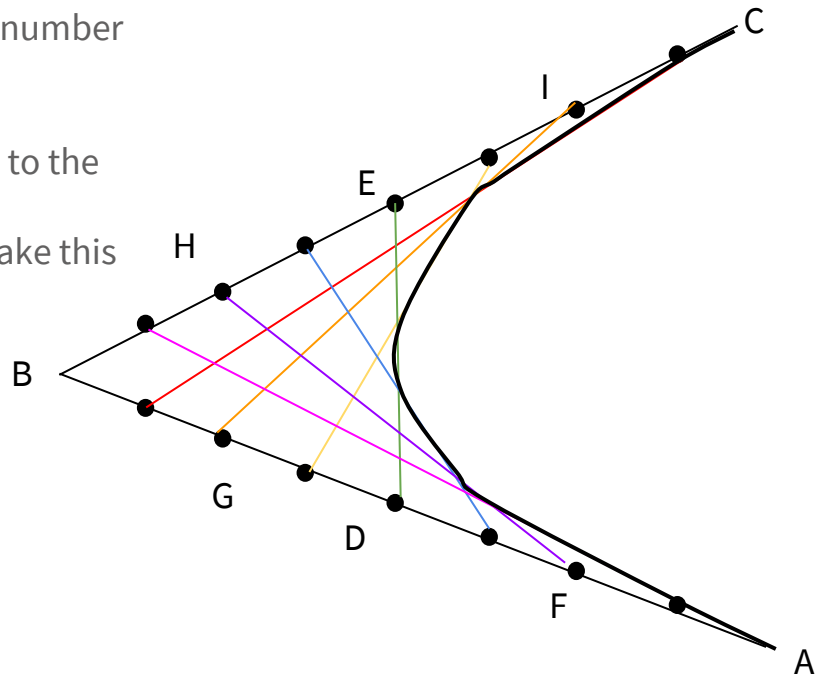
Curves & Surfaces: Midpoint Method

1) Find midpoint of AB (called D) and BC (called E)
2) FInd midpoint of AD, DB, BE, and EC
3) Continue to find midpoints of segments until desired number of points is reached
4) Connect D and E using a line segment, then connect midpoints to the left of D to the corresponding points to the right of E and vice versa, as shown:



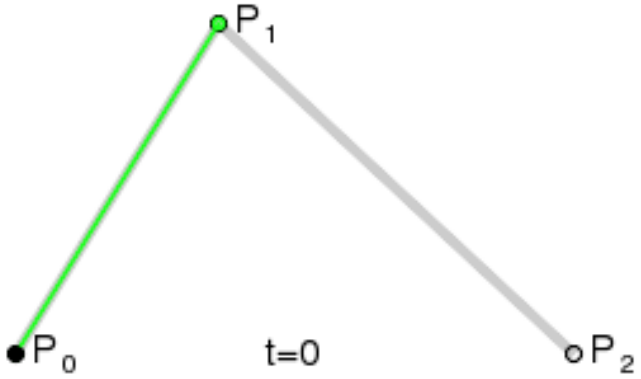
Curves & Surfaces: Midpoint Method

1) Find midpoint of AB (called D) and BC (called E)
2) FInd midpoint of AD, DB, BE, and EC
3) Continue to find midpoints of segments until desired number of points is reached
4) Connect D and E using a line segment, then connect midpoints to the left of D to the corresponding points to the right of E and vice versa, as shown:
5) Now we have a curve! More points can be added to make this curve smoother



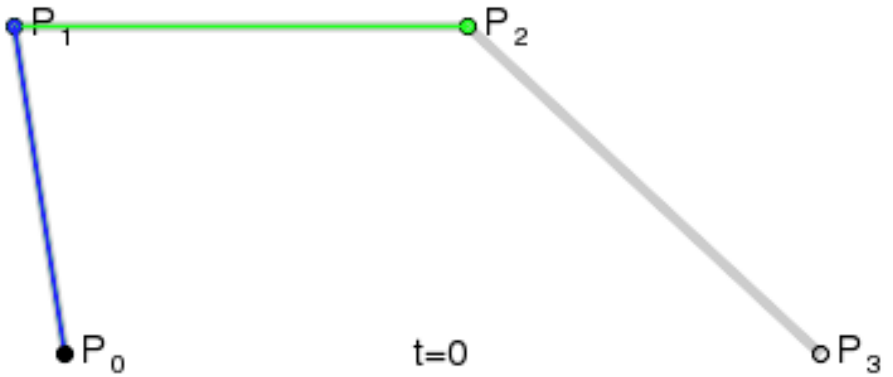
Curves & Surfaces: Bezier Curves

- Realistic and seamless lines and curves of objects is achieved using Bezier curves, which can be applied to computer graphics using DeCasteljau's algorithm
- Let Q be a point on P_0P_1 and t is a parameter that indicates where on the line Q is
  - As t goes from 0 to 1, Q goes from P_0 to P_1
  - Create point R on P_1P_2 and connect Q and P
  - Let U be a point on the line QR
  - As t increases a curve is created



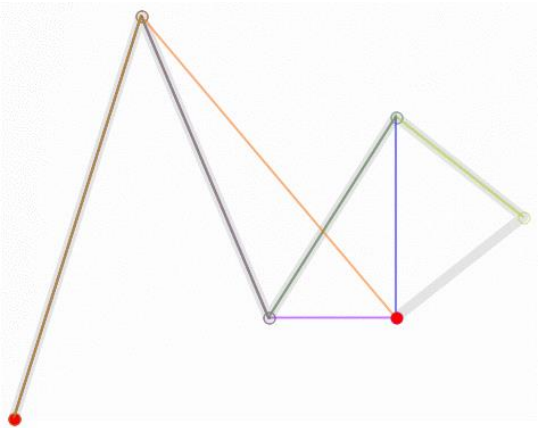
Curves & Surfaces: Bezier Curves

- Realistic and seamless lines and curves of objects is achieved using Bezier curves, which can be applied to computer graphics using DeCasteljau's algorithm
- Let Q be a point on P_0P_1 and t is a parameter that indicates where on the line Q is
  - As t goes from 0 to 1, Q goes from P_0 to P_1
  - Create point R on P_1P_2 and connect Q and R
  - Let U be a point on the line QR
  - As t increases a curve is created
- Usually 4 points are used in animation
- Changing the location of these points in space changes the shape of the curve



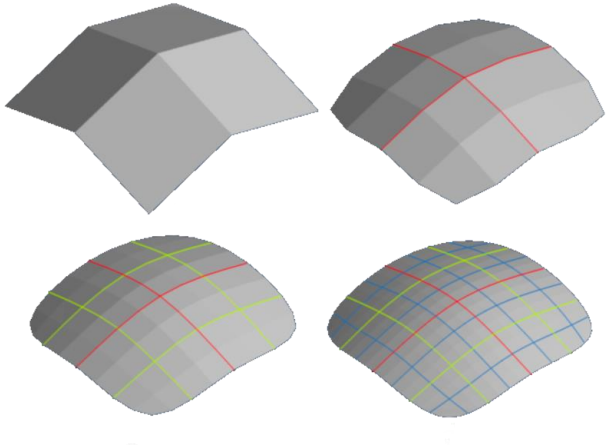
Curves & Surfaces: Finding Equation of Curve

- For a three point polygon, we have $Q=(1-t)P_0+tP_1$, $R=(1-t)P_1+tP_2$, and $U=(1-t)Q+tR$
- Then $U=(1-t)[(1-t)P_0+tP_1]+t[(1-t)P_1+tP_2]=(1-t)^2P_0+2t(1-t)P_1+t^2P_2$
- So, for a three point polygon, we have a degree 2 polynomial
    - A general rule is that for an n point polygon, we get a polynomial of degree n-1



Curves & Surfaces: Subdivision

- Start with a base level polygonal mesh and recursively apply refinement scheme to the mesh
  - Creates new vertices and surfaces that are then used as the base level and further subdivided



Curves & Surfaces: Subdivision

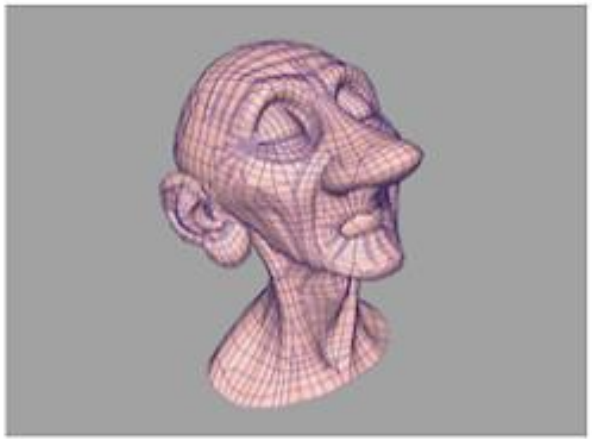


Figure 2: The control mesh for Geri's head, created by digitizing a full-scale model sculpted out of clay.

animated. As a case in point, considerable manual effort was required to hide the seams in the face of Woody, a principal character in *Toy Story*.

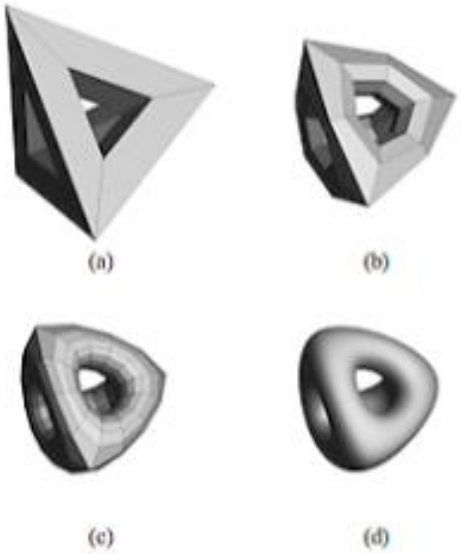
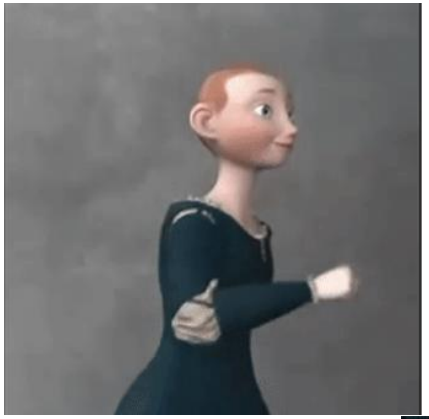


Figure 3: Recursive subdivision of a topologically complicated mesh: (a) the control mesh; (b) after one subdivision step; (c) after two subdivision steps; (d) the limit surface.

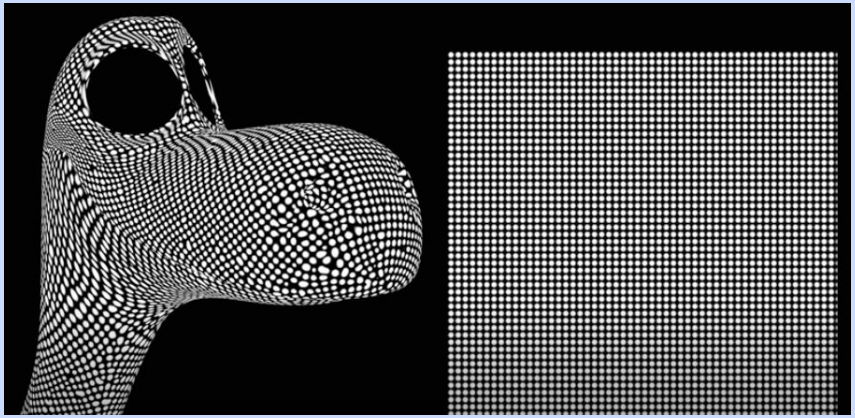
Simulations

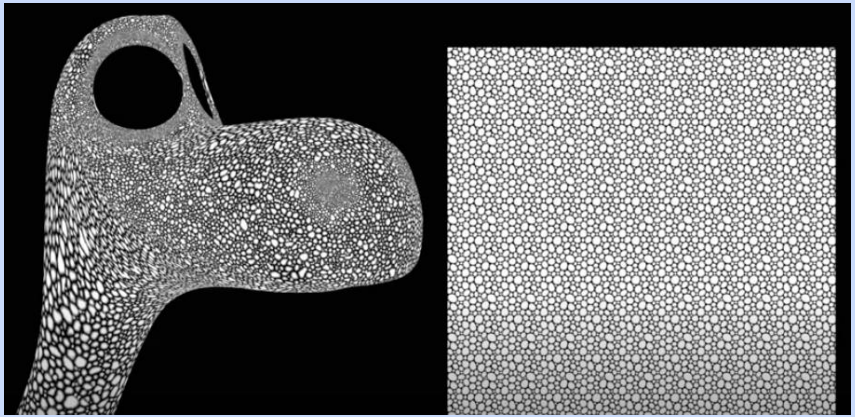
Simulating Hair

- Hair acts like a spring
  - Different parameters for stiffness, damping, and support springs depending on curliness and movement of hair
  - Damper resists fast changes in displacement and makes spring come to rest faster
  - Support springs hold different parts of hair together
  - Use Hooke’s law to calculate spring force
- Simulations also used for water and air using particles



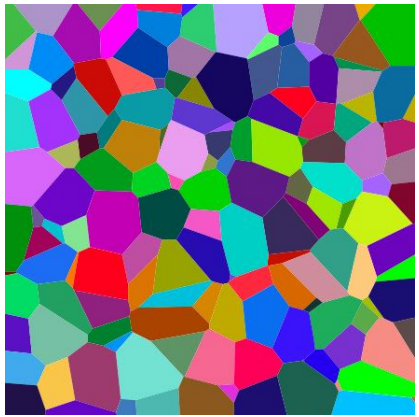
Patterns & Texture





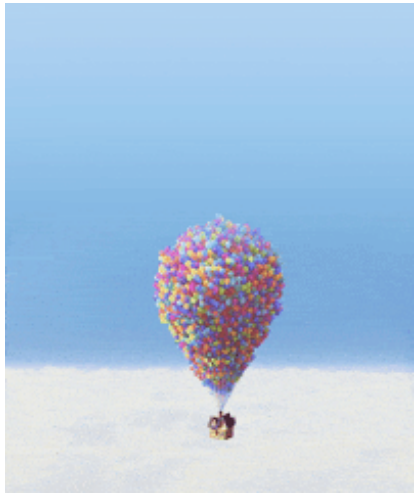
Patterns & texture

- Most skin, hair, scales, fur, etc. is generated semi-randomly so that it looks natural
- Voronoi diagram
  - Partition surface into regions close to each of a given set of objects, usually points on a plane.
  - Poisson disk process: set minimum distance between points and then randomly generate points outside of that distance



And More!

- Virtual camera lens and placement
- Lighting
- Rigging
- Rendering



Questions & Discussion

References

https://en.wikipedia.org/wiki/B%C3%A9zier_curve
https://calculate.org.au/2014/05/01/pixar-animations-maths-make-work/
https://www.kqed.org/mindshift/41879/pixar-in-a-box-teaches-math-through-real-animation-challenges
https://www.khanacademy.org/computing/pixar/animate/parametric-curves/a/lesson-brief-animation
https://en.wikipedia.org/wiki/Voronoi_diagram

Pictures
https://www.ck12.org/geometry/translations/lesson/Geometric-Translations-BSC-GEOM/
https://www.onlinemath4all.com/properties-of-rotations.html
https://www.cinemablend.com/news/2570786/mads-mikkelsen-9-other-celebrities-voiced-disney-characters-other-languages-pixar
https://www.sapeople.com/2015/12/13/sas-pixar-ish-studio-that-made-khumba-announces-new-projects/
https://hifigif.cc/gifs/-priyanka-in-cloth/pixar-brave-animation-tests-merida-fergus-cloth-and-hair
https://www.deviantart.com/joaoppereiraus/art/Bouncing-Ball-Animation-671561657