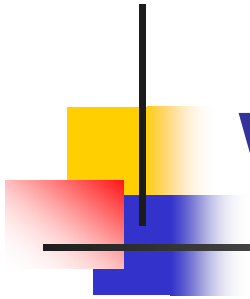
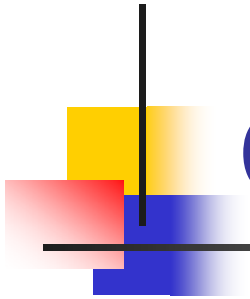
CS 112 - Object Representation



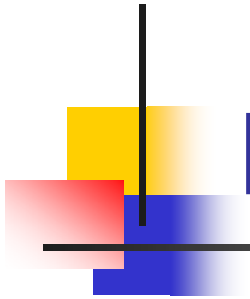
What is Graphics?

- Modeling
  - Computer representation of the 3D world
- Analysis
  - For efficient rendering
  - For catering the model to different applications.....
- Rendering
  - Generating 2D images of the 3D world



Object Representation

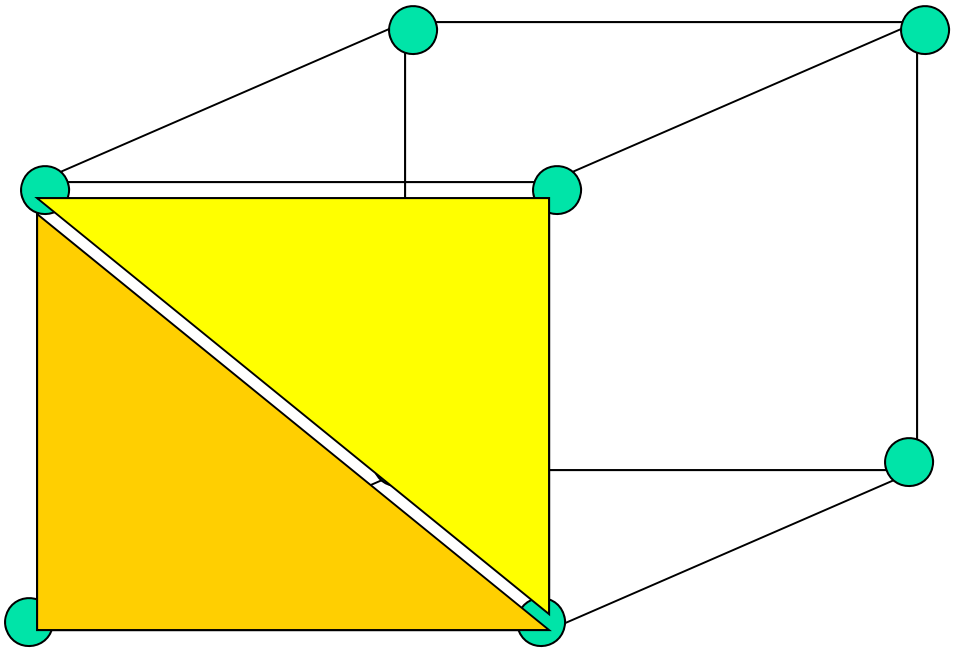
- Polygons, Points, Images, Surfaces
- Representations
  - Explicit : y = mx+c
    - Can draw the line, but cannot represent y axis
  - Implicit: ax+by+c =0
    - Cannot draw the line, but evaluate if points lie on the line
  - Parametric: P1(1-u)+P2(u)
    - Interpolation is simple

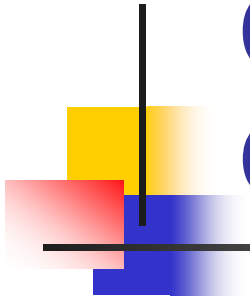


Polygon Representation

- TRIANGLES: most common
- Geometric information
  - Vertices
    - 3D coordinates, Normals
- Topological information
  - Connectivity or Adjacency
- Other attributes **at vertices**
  - Color, texture, transparency....

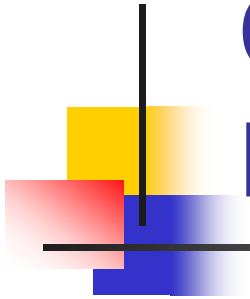
Object Representation: Example





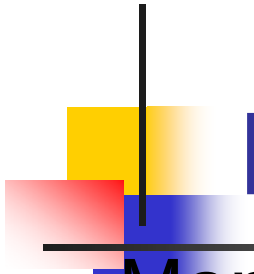
Characteristics of Polygonal Objects

- Geometric properties
  - Position
  - Normals
  - Curvature, continuity



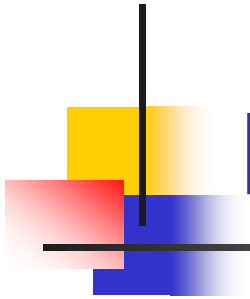
Characteristics of Polygonal Models

- Topological properties
  - Manifolds (w/ boundaries) /Non-manifold
  - Dimension
  - Euler characteristic/Genus
  - Orientability
- *Invariant* with change in geometric properties



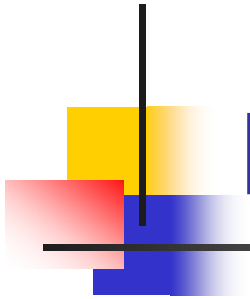
Manifold Definitions

- Manifolds
  - 2D: Every edge has exactly two incident triangles.
  - 3D: Every triangle has exactly two incident tetrahedrons.
- Manifolds with boundaries
  - 2D: Every edge has either one or two incident triangles.
  - 3D: Every triangle has either one or two incident tetrahedrons.
- Non-manifolds
  - That does not have the above restrictions.



Dimension

- Number of parameters you can change and still be in the object
- Point : dimension 0
- Point in a room: dimension 0 embedded in 3D room
  - 3 coordinates, how many can you change and still be on the point? --- None
  - Hence, dimension 0



Dimension

- Line
  - P1(1-u)+P2(u)
- You can change u and still be on the line
- Hence, dimension 1
- But note, these objects can be *embedded* in a higher dimension world

Manifold: Euler Characteristic (e)

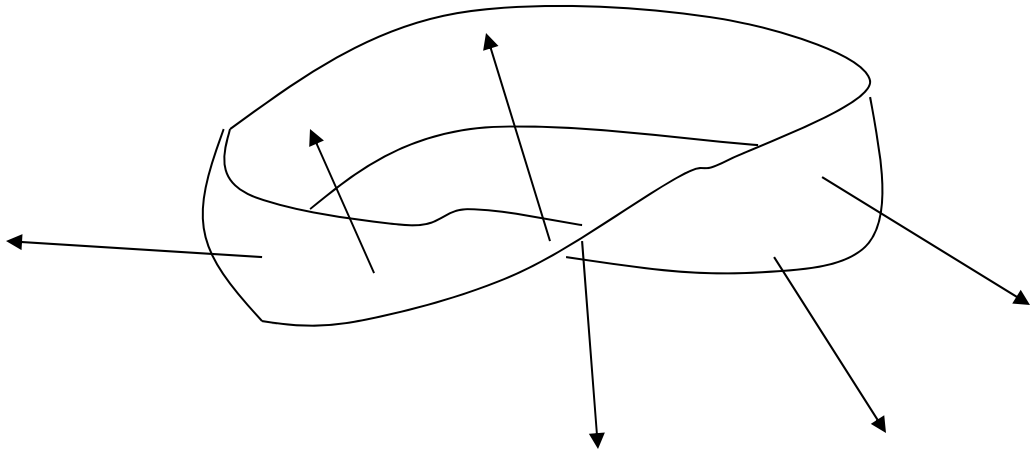
- e = V-E+F (V: Vertices, E: Edges, F: Faces).
- Applicable only for manifolds
- In general
  - e=(0 dim)-(1 dim)+(2 dim)-(3 dim)+(4 dim)...
- Verify: Cube has 8 vertices, 12 edges, 6 faces
  - e = V-E+F = 8-12+6 = 2
- Changing geometric properties keeps euler characteristic invariant
  - Adding edges, vertices or pulling vertices
  - Square approaching sphere

Manifold: Genus (g)

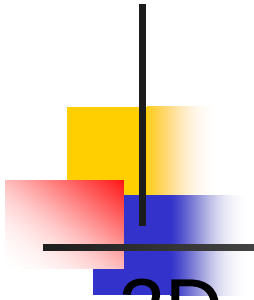
- Applicable only for manifolds
- (Naïve) Number of “handles”.
- Sphere has g=0; cube has g=0; torus has g=1; coffee cup has g=1.
- Going from coffee cup to torus
  - Changing only geometric properties
- Relationship between e and g: e=2-2g
  - Sphere or Cube: e=2-2(0)=2
  - Torus: e=2-2(1)=0

Orientability of an object

- If you have consistent normal direction for a point then the object is orientable. Otherwise, non-orientable.



Mobius Strip

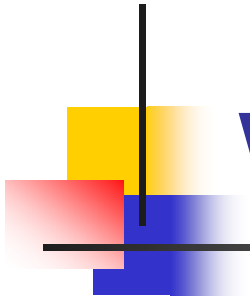


In this course..

- 2D orientable manifolds without boundaries.

Remember: manifolds with boundaries is a superset of manifolds, and non-manifold is a superset of manifolds with boundaries.

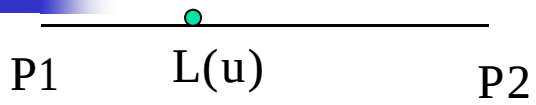
Non-manifold actually means that “need not” be a manifold; not “is not” a manifold.



Why TRIANGLES?

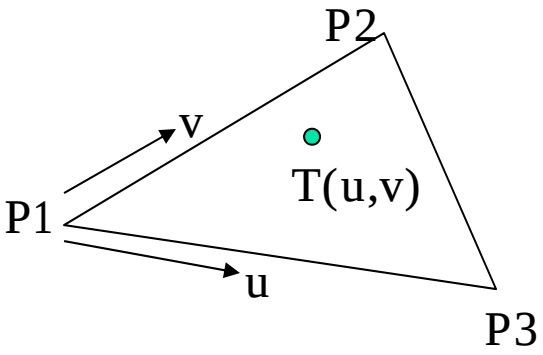
- Consistently planar
- Easy definition of topology and geometry
- Rotationally invariant interpolation of attributes during rasterization
- Easy to implement in hardware

Interpolation



Linear Interpolation

$L(u) = P1(1-u)+P2(u)$

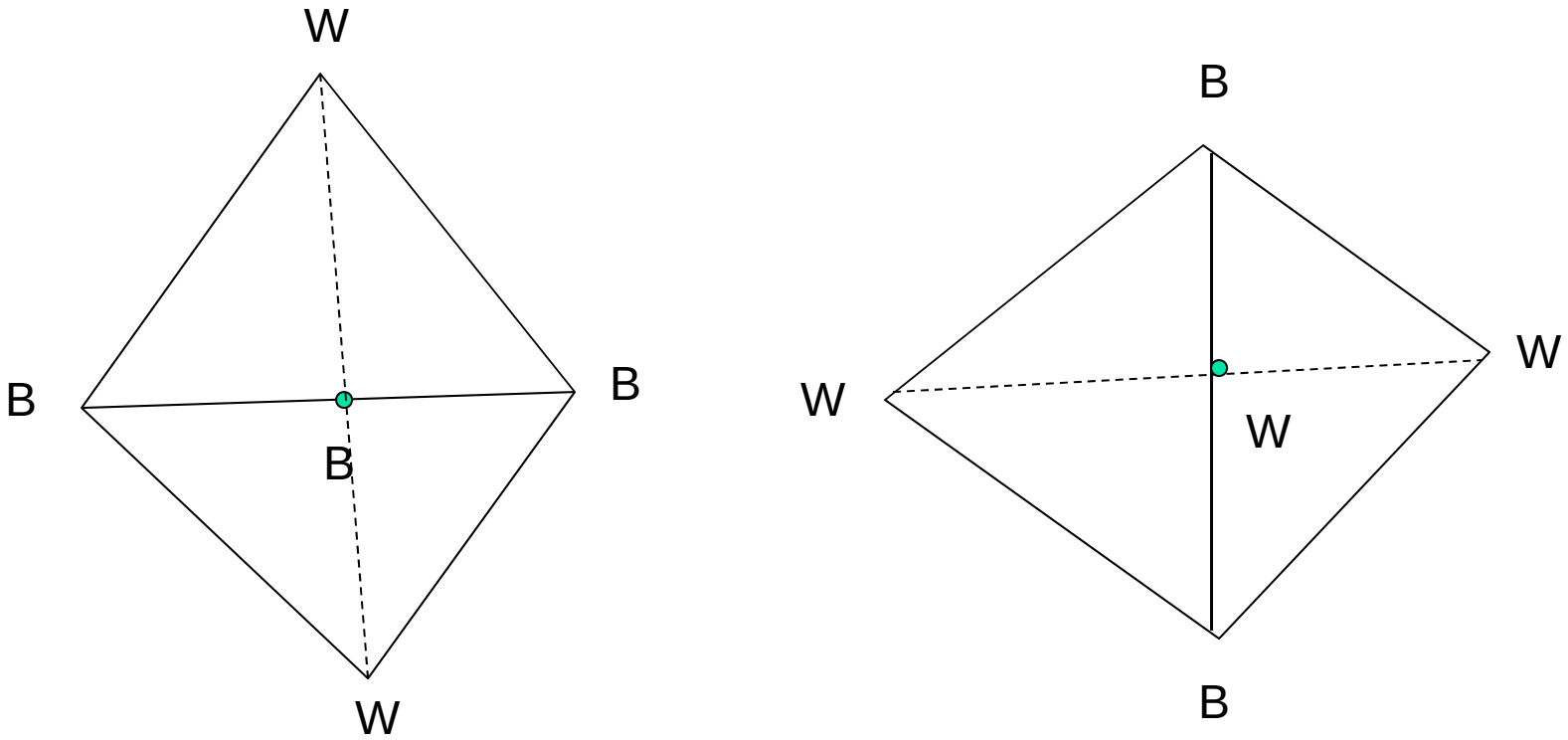


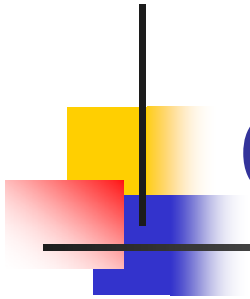
Bilinear Interpolation

$T(u,v) = P1(1-u)(1-v) + P2.u + P3.v$

- *Linear* combination of the points
- *Coefficients*
  - Less than 1.0 - Within the convex hull
  - Greater than 1.0 – Outside the convex hull

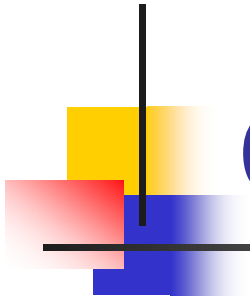
Rotationally Invariant Interpolation





Other Representations

- Point
  - No connectivity
- Quadrilaterals
  - Non-planar, inconsistent interpolation of attributes
- Spline
  - Have to be converted to polygons for rasterization



Common Data Structure

- Sequential List of Vertices
- Indexed List of Vertices
  - No need to be in order
- List of Triangles
  - Defined by indices of vertices