Lecture 1

Mesh Representations

3D Shapes

The world is 3D

(Euclidean) Geometry is continous

3D Shapes

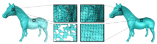
- Problem: computers store discrete quantities
- Geometry is continuous

3D Shapes

We need to choose a representation

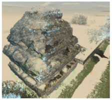
Representation Zoo

PolyMender



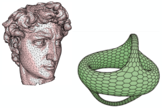
Polygon Soups

eScience Appia

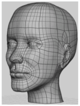


Point Clouds

unstructured

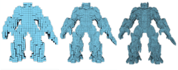


Polygonal Meshes



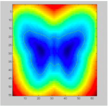
Spline Surfaces

Explicit



Voxels

http://snikolov.weebly.com/uploads/3/9/0/4/3904101/4164909.png?460



Level Sets

Implicit

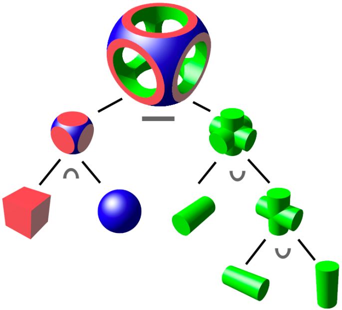
Point Clouds

- Acquired from range scans
- Data: 3D point positions
- No proximity or topological information

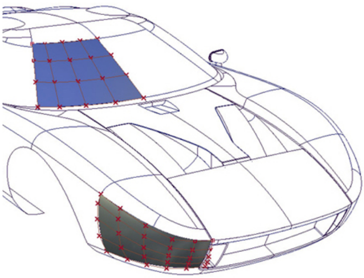
Point Cloud

CAD Models

- Constructive solid geometry (CSG)
- Stereolithography (STL)
- Tensor-product surfaces
- Connected patches
- Descriptive geometry



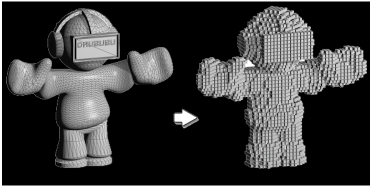
CSG



CAD model

Voxels

- Shape as a subset of a discrete grid
- Used in volumetric rendering
- Minecraft/No Man's Sky



Voxel

Minecraft

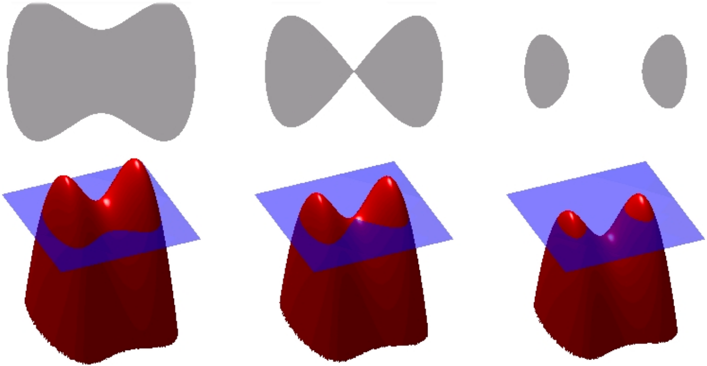
Voxels

Voxel cloud simulation rendered with raymarching

Level Sets

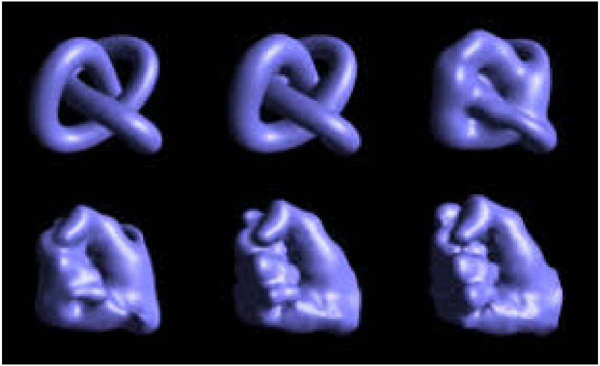
Shape as zero set of implicit distance function

- Easy topological changes
- Surface Evolution



Level Set

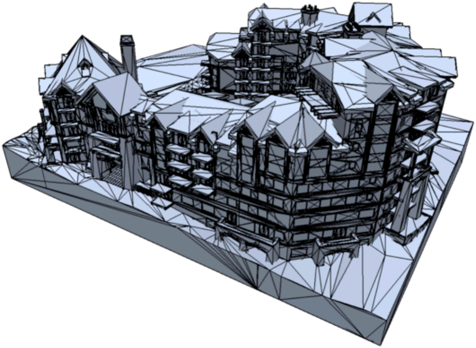
Level Sets



Morphing an implicit surface

Polygon Soups

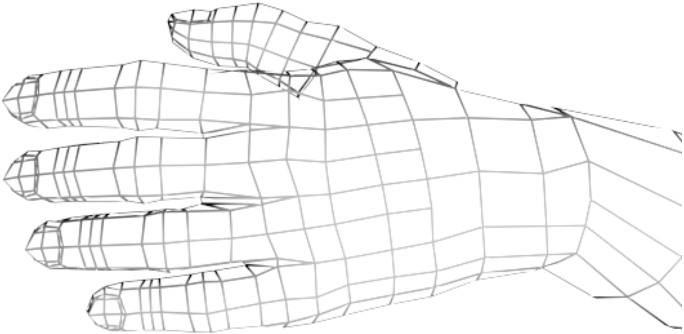
- "Floating" polygons
- No connectivity between them



Polygon Soup

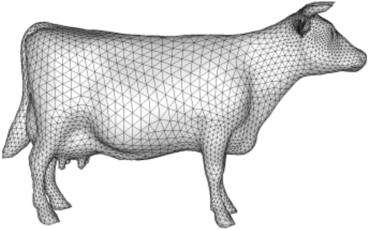
Polygon Meshes

- Most Ubiquitous Representation
- Models the surface only
- Several variants tailored to specific needs

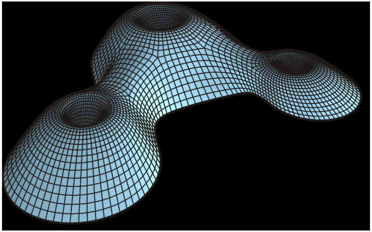


Polygon Mesh

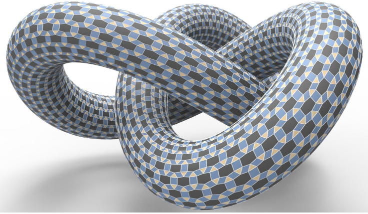
Polygonal Mesh Variants



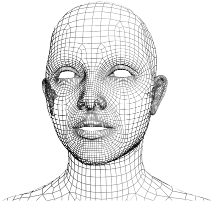
Triangle Meshes



Quad Meshes



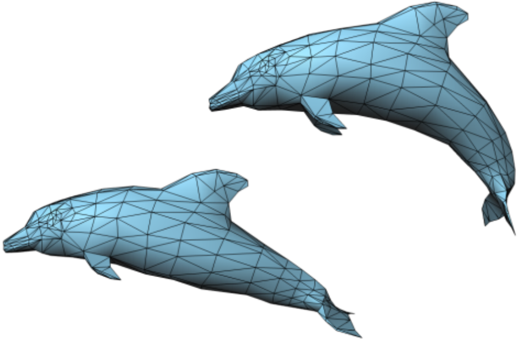
Tilings



Semi-regular Mesh

Triangle Meshes

- The “default” mesh variant
- Faces are always planar
- Any polygonal mesh can be subdivided into a triangle mesh
- Efficient to render
- Maximal amount of edges / superflous information



Triangle Mesh

3D Shapes

- If a shape has a volume, then it is bounded by 2-dimensional patches (faces or surfaces)
- These 2-dimensional boundary patches are bounded by 1-dimensional features (edges or curves)
- These 1-dimensional features are bounded by 0-dimensional objects (vertices or points)

3D Shapes

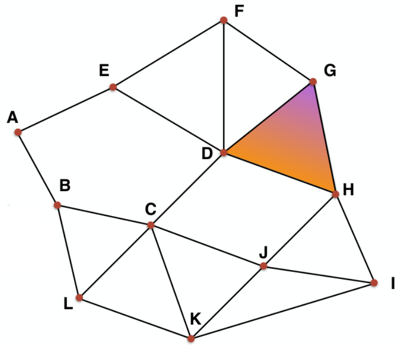
- We often only model the surface, volume is implicit.
- Surface is locally 2-dimensional.
- In practice we have to deal with holes.

Meshes as Graphs

$$V = A, B, C, \ldots, L$$

$$E = (A, B), (B, C), \ldots, (K, L)$$

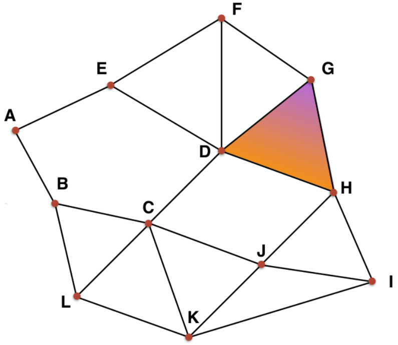
$$F = (A, B, C, D, E), \ldots, (G, D, H)$$



Mesh as a graph

Vertices

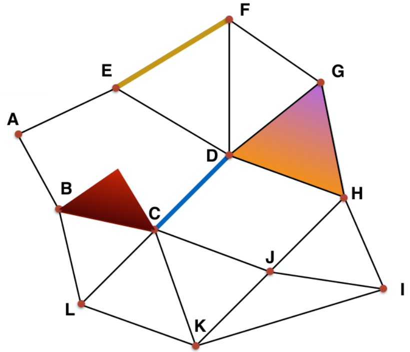
- Each vertex is connected to a number of edges
- Number of edges is called degree or valence
- We say a mesh is regular if all degrees are equal



Mesh as a graph

Edges

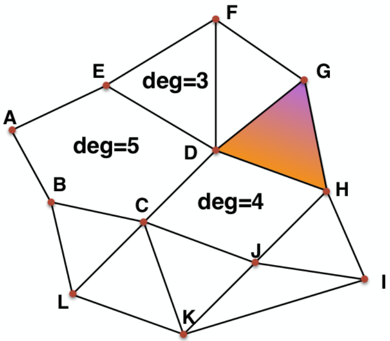
- Not every mesh has a closed surface
- Regular edge: adjacent (connected) to exactly 2 faces
- Boundary edge: adject to only 1 face
- Singular edge: adjacent to more than 2 faces
- Isolated edge: adjacent to no face (not always possible)



Mesh edge cases

Faces

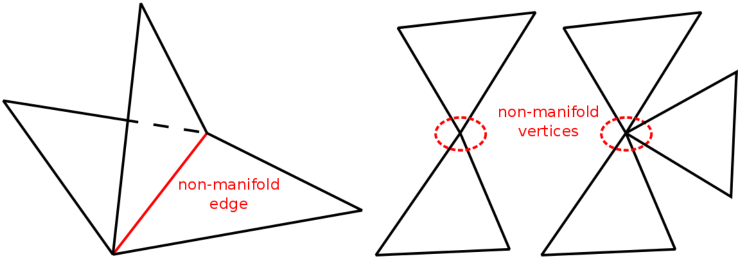
- Faces also have a degree or valence
- Equal to the number of edges and vertices
- Defines a triangle, quad or polygonal mesh



Faces in graph

Manifolds

- We say a 3D shape is a manifold when it is “watertight”.
- It has a uniquely defined volume.
- the interior is separated from the exterior everywhere
- In a 3D mesh this means there are no non-manifold subobjects and no “open” edges.

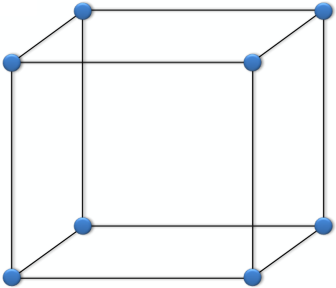


Manifolds

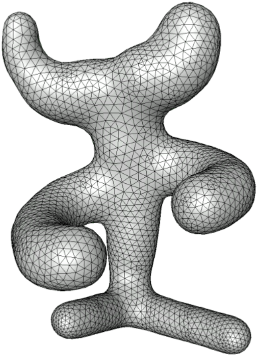
For manifold meshes the Euler-Poincare Formula holds:

$$\mathsf{V} + \mathsf{F} - \mathsf{E} = \chi$$

The χ here is the Euler characteristic.



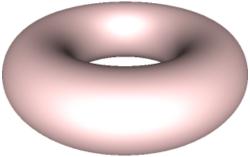
$\mathsf{V} = 8, \mathsf{E} = 12, \mathsf{F} = 6, \chi = 2$



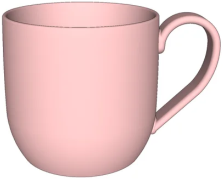
$\mathsf{V} = 3890, \mathsf{E} = 11664, \mathsf{F} = 7776, \chi = 2$

Manifolds

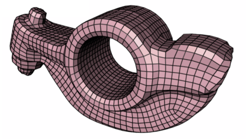
Euler characteristic says something about the type of surface of the manifold.



$\chi = 0$



$\chi = 0$

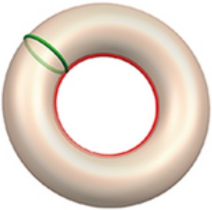


$\chi = 0$

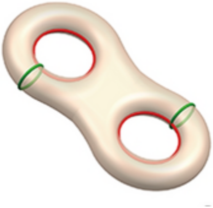
Manifolds

An object has the same topology:

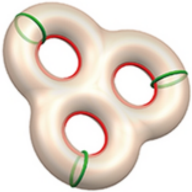
- if it can be transformed into the other object without any cuts.
- The “class” of such an object is called its **genus**.
- Genus corresponds to the number of “holes” the manifold has.



genus = 1



genus = 2



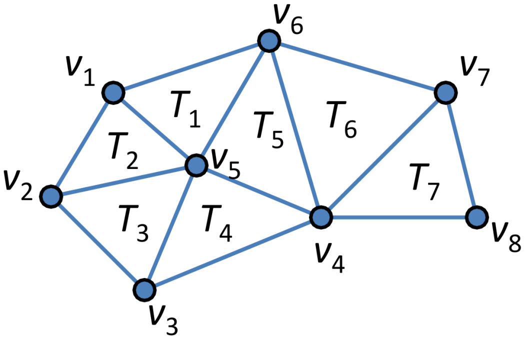
genus = 3

Mesh Representations

Mesh Representations

Triangle List

- Each triangle is an triple that stores the coordinates of its vertices:



$$[(v_1, v_2, v_5), (v_2, v_3, v_5), \ldots, (v_8, v_7, v_4))]$$

Mesh Representations

- Need to choose a face orientation: clockwise (CW) or counter-clockwise (CCW).
- In graphics CCW is most common.
- Determines the direction of the face normal (right-hand rule).

Mesh Representations

Triangle list implementations.

$$[(\mathtt{v}_1, \mathtt{v}_2, \mathtt{v}_5), (\mathtt{v}_2, \mathtt{v}_3, \mathtt{v}_5), \ldots, (\mathtt{v}_8, \mathtt{v}_7, \mathtt{v}_4))]$$

Vertices given as:

- Explicit coordinates, e.g first triangle is:

$$((x_1, y_1, z_1), (x_2, y_2, z_2), (x_5, y_5, z_5))$$

- References if the language supports them.
- Most often: indices. $v_1 = 0, v_2 = 1, \ldots v_8 = 7$ and an array of vertices V.

`V[1][1]` is the y-coordinate of the second vertex.

Mesh Representations

- In a (big enough) triangle mesh:

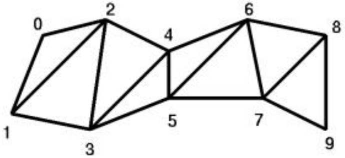
$$\mathsf{E} \approx 3\mathsf{V}, \mathsf{F} \approx 2\mathsf{V}$$

- Degree approaches 6

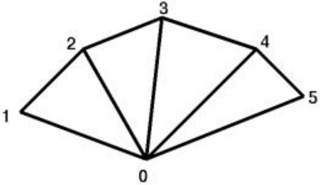
Mesh Representations

Triangle strips and fans maximize throughput.

Only a single vertex is added at each step.



triangle "strip"
vertices: 10 triangles: 8
vertices per triangle: 1.25

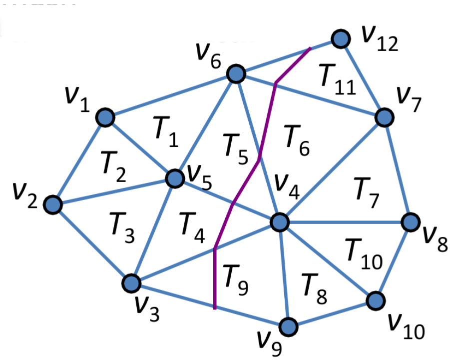


triangle "fan"
vertices: 6 triangles: 4
vertices per triangle: 1.5

Mesh Representations

Triangle lists are not suited for mesh editing.

- Subobjects should have direct access to the adjacent subobjects.
- Allows neighborhoods on the mesh accessed efficiently.
- Allows pathfinding across subobjects.

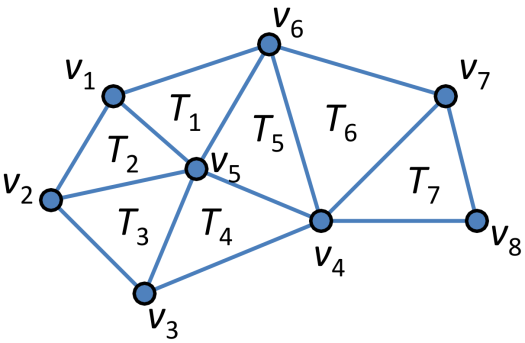


finding an face loop

Mesh Representations

Triangle neighbor structure:

- Adds references to adjacent triangles (often also indices).



$$[\texttt{Triangle}((v_1, v_2, v_5), [T_1, T_3]), \texttt{Triangle}((v_2, v_3, v_5), [T_2, T_4]), \ldots]$$

Mesh Representations

Winged edge structure:

- Stores vertices, edges, and faces as separate objects.
- Looks a bit like the graph representation.
- Very common, you will run into this a lot.
- Highest storage requirement.

Mesh Representations

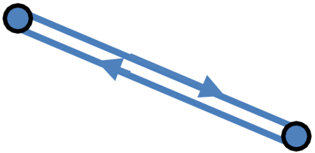
Winged edge structure:

- Each vertex stores references to its:
  - adjacent edges
- Each edge stores references to its:
  - two vertices and
  - two adjacent faces (called left and right).
- Each face stores references to its:
  - vertices (optional)
  - neighboring faces (optional), and
  - adjacent edges.

Mesh Representations

Half-edge structure

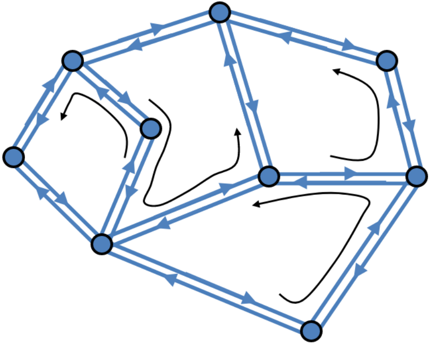
- Uses a doubly-connected edge list (DCEL).
- You can traverse the entire mesh, by just moving forward.



each edge consists of two half-edges

Mesh Representations

- Each halfedge only has a single face to its left (by convention).
- Leads to every face being CCW.
- Fixes the inconsistent edge direction problem.



DCEL Representation

Mesh Representations

Data structure

Mesh:

$$[\mathtt{v}_1, \mathtt{v}_2, \mathtt{v}_3, \mathtt{v}_1, \mathtt{v}_5, \ldots, \mathtt{v}_9]$$

Halfedges:

$$[\mathtt{e}_1, \mathtt{e}_2, \ldots, \mathtt{e}_{24}]$$

size is `len(DCEL) - 1`.

Faces:

$$[\mathtt{f}_1, \mathtt{f}_1, \mathtt{f}_1, \mathtt{f}_1, \mathtt{f}_2, \mathtt{f}_2, \mathtt{f}_2, \ldots, \mathtt{f}_3]$$

Conclusion

- Many 3d geometry representations exist.
- Most common is the mesh representation.
  - Simple discrete representation.
  - Fast to render.
  - Not always suited.

Notice

Exercises will be made available after the lecture.

See you on Thursday.