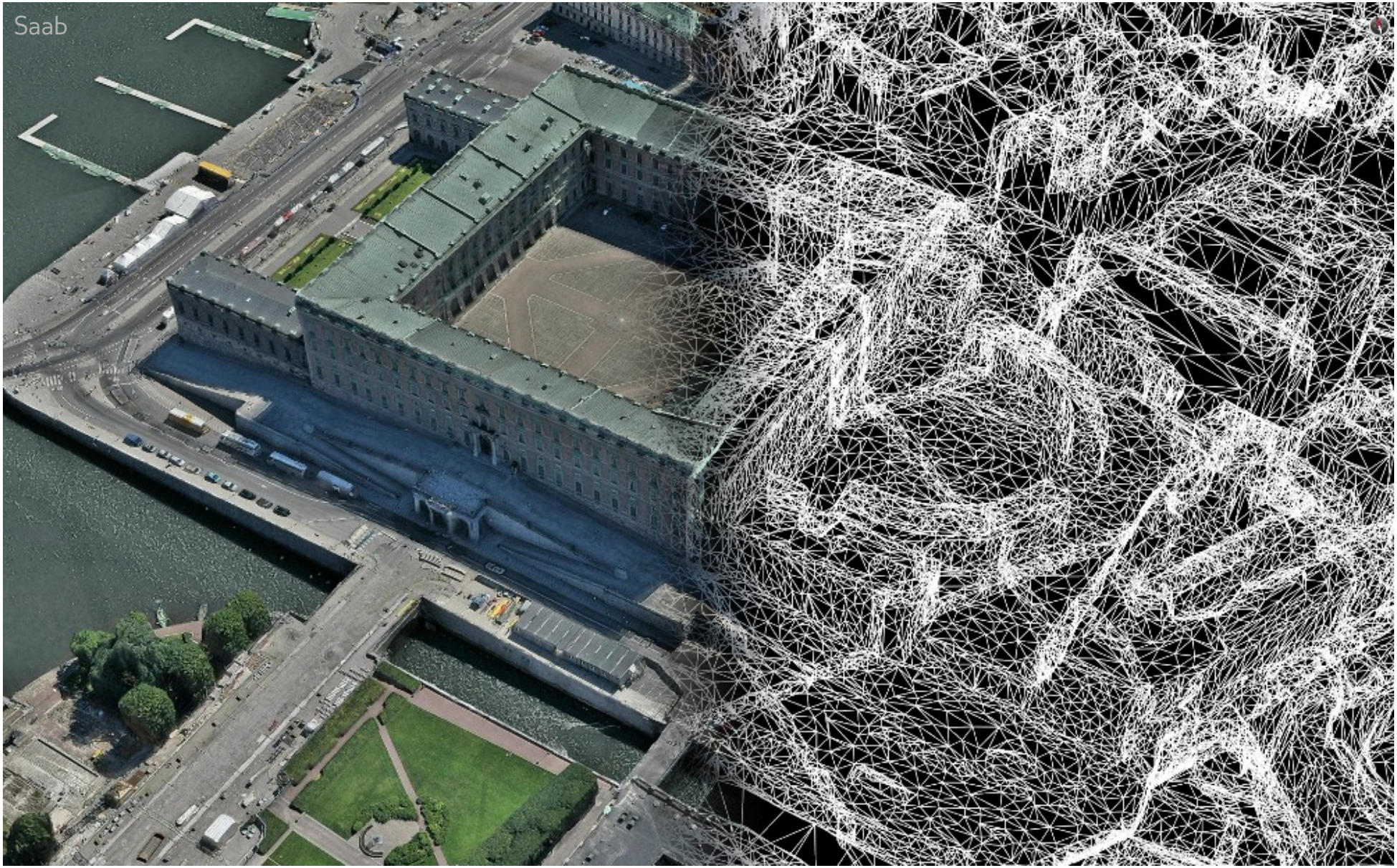
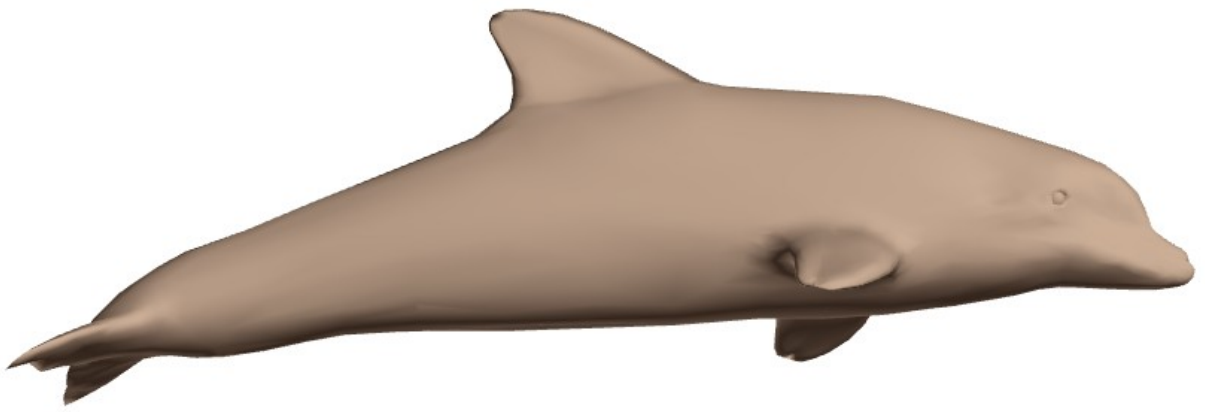



# Polygon Meshes

Siddhartha Chaudhuri http://www.cse.iitb.ac.in/~cs749

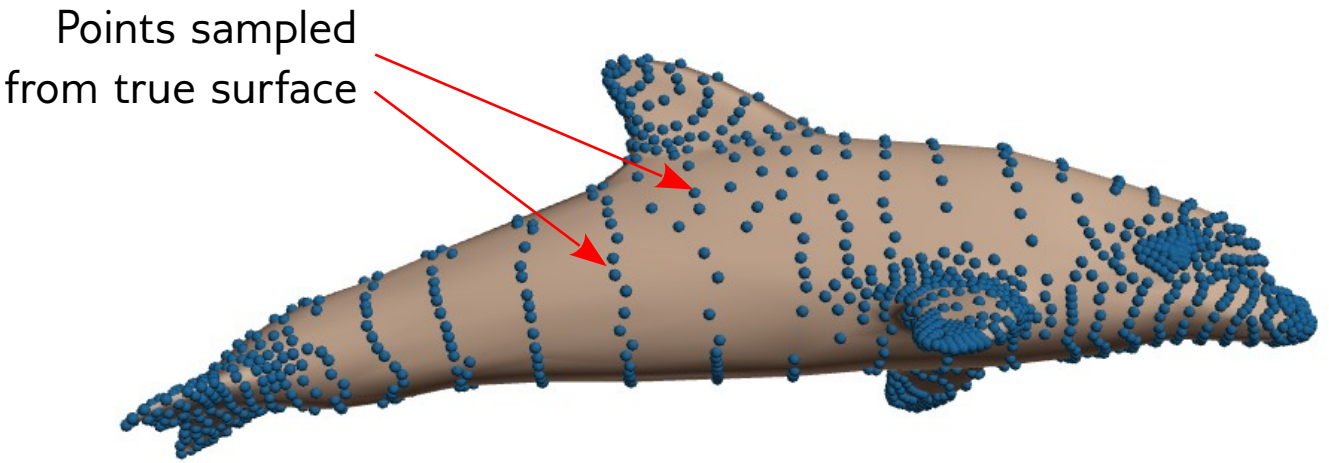
# What is a polygon mesh?

- Like a point cloud, it is a discrete sampling of a surface
- ... but, it adds simple polygons (no holes or self-intersections) as linear (flat) approximations of local regions of the actual underlying surface



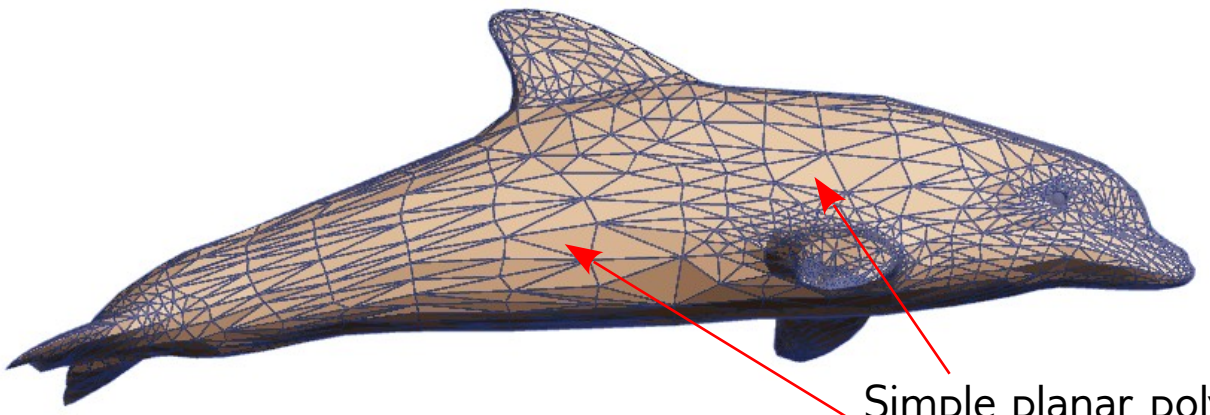
# What is a polygon mesh?

- Like a point cloud, it is a discrete sampling of a surface
- ... but, it adds simple polygons (no holes or self-intersections) as linear (flat) approximations of local regions of the actual underlying surface



# What is a polygon mesh?

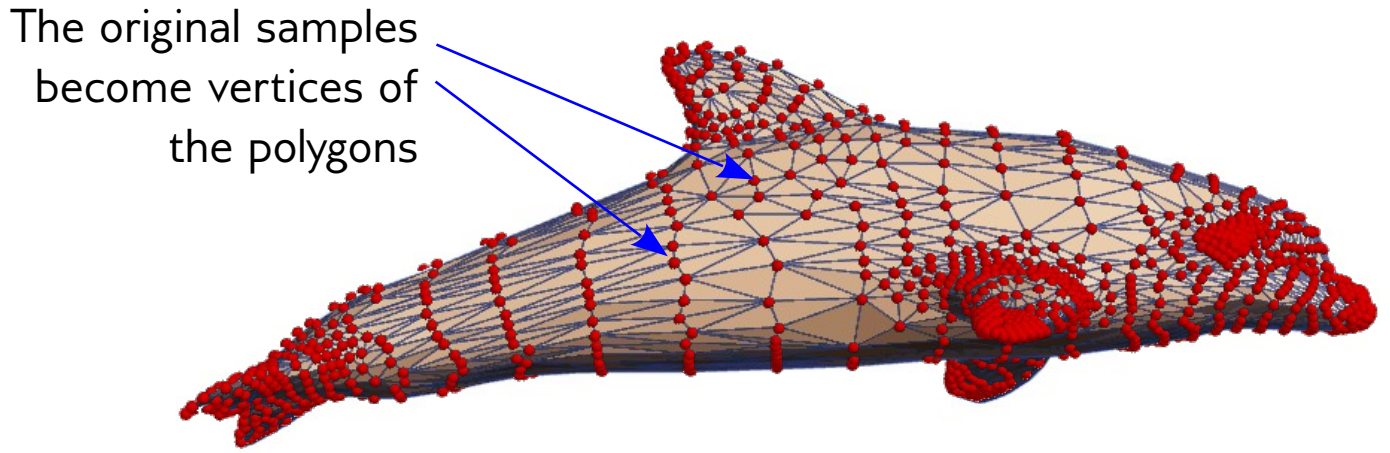
- Like a point cloud, it is a discrete sampling of a surface
- … but, it adds simple polygons (no holes or self-intersections) as linear (flat) approximations of local regions of the actual underlying surface



Simple planar polygons: linear approximations to true surface

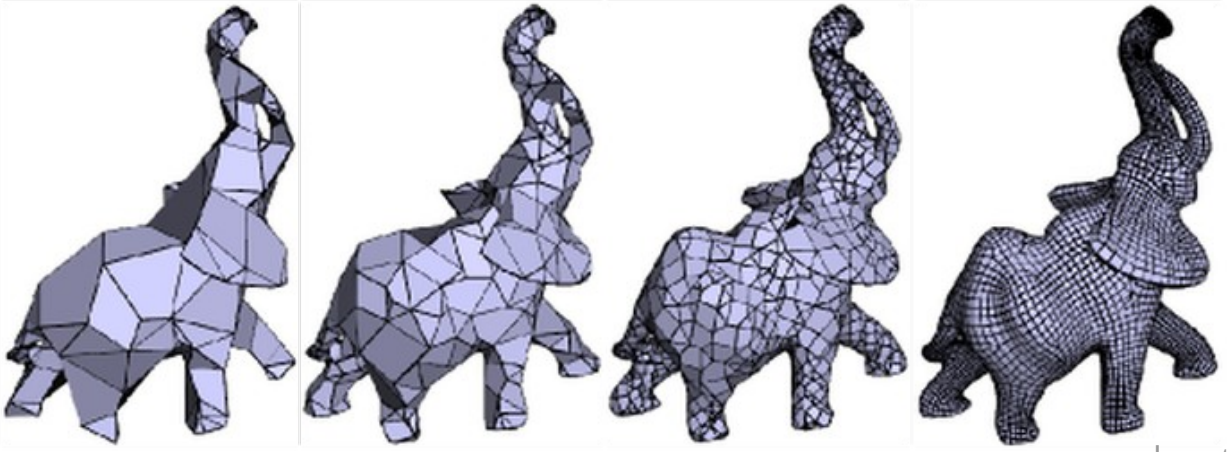
# What is a polygon mesh?

- Like a point cloud, it is a discrete sampling of a surface
- … but, it adds simple polygons (no holes or self-intersections) as linear (flat) approximations of local regions of the actual underlying surface



# What is a polygon mesh?

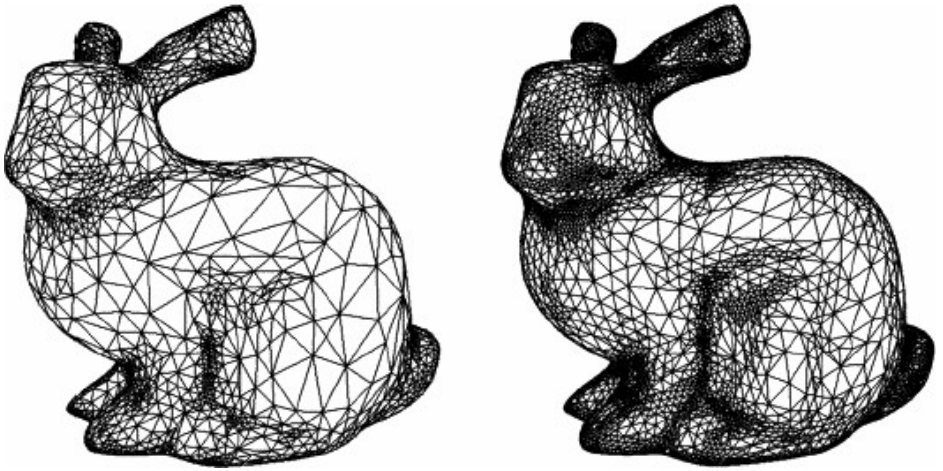
- Like a point cloud, it is based on a discrete sampling of a surface
- ... but, it adds simple polygons (no holes or self-intersections) as linear (flat) approximations of local regions of the actual underlying surface
- Like point clouds, meshes can have different resolutions



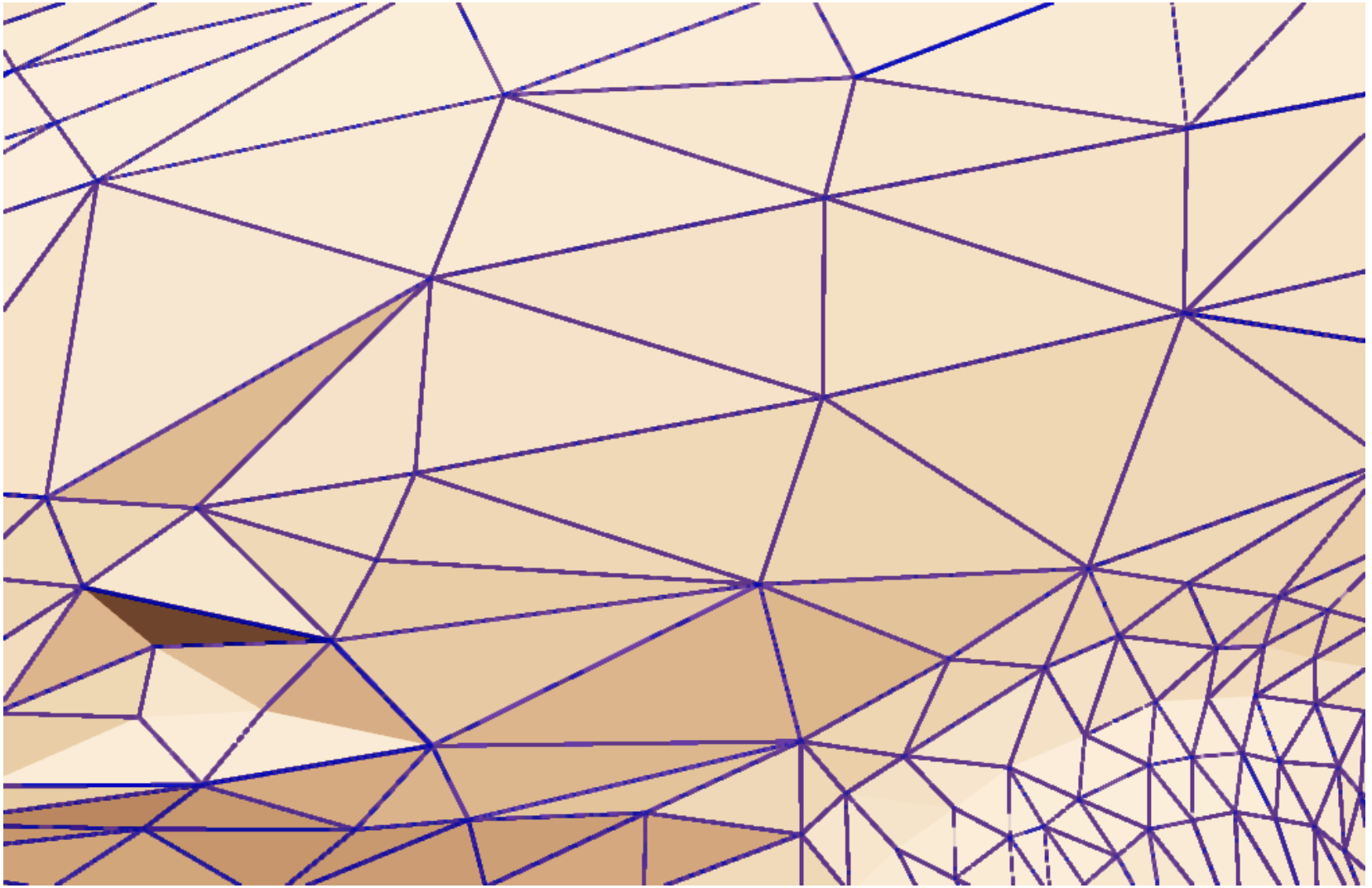
gamedev.stackexchange.com

# What is a polygon mesh?

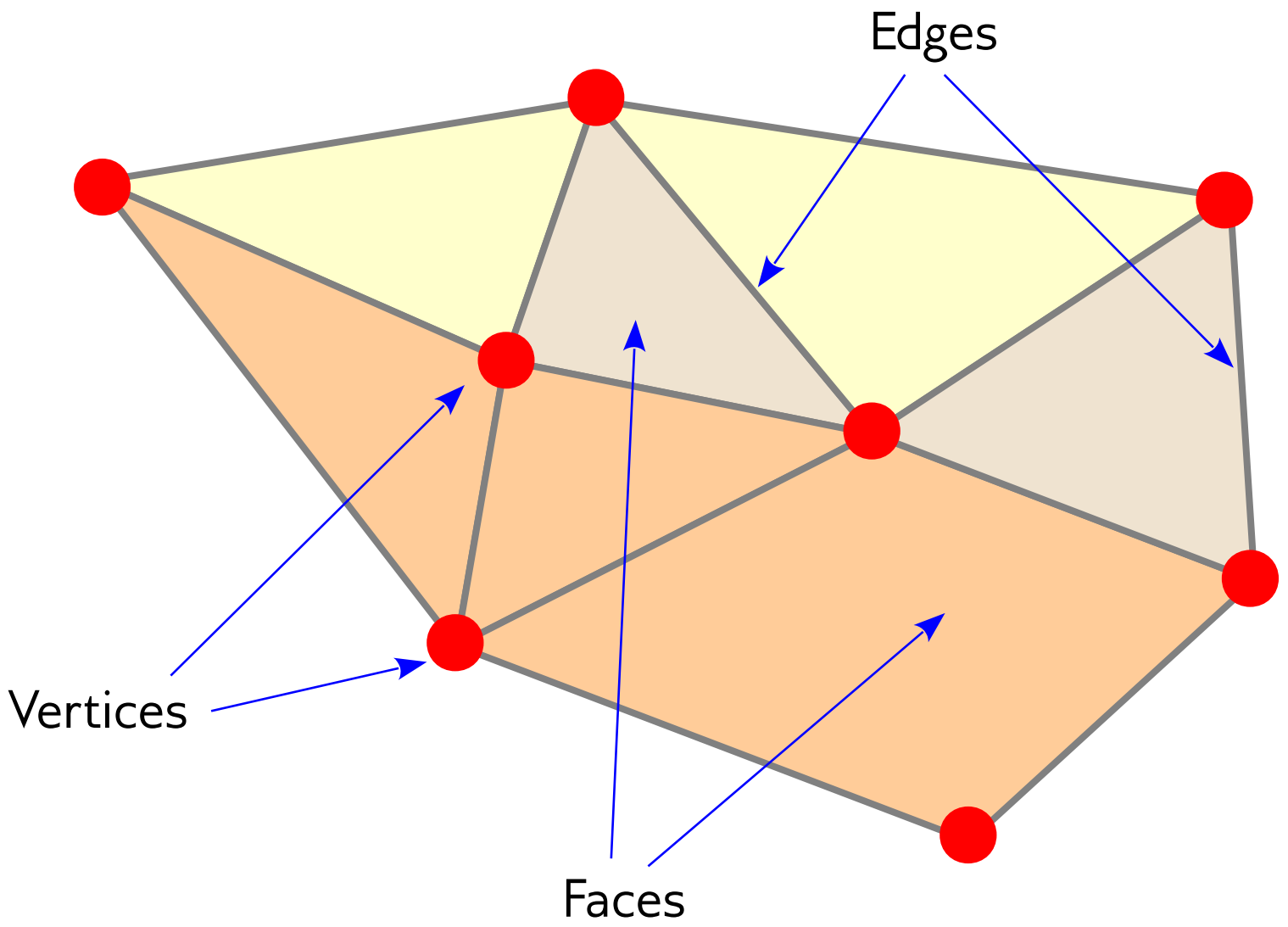
- Like a point cloud, it is based on a discrete sampling of a surface
- … but, it adds simple polygons (no holes or self-intersections) as linear (flat) approximations of local regions of the actual underlying surface
- Like point clouds, meshes can have different resolutions
  - … at different places ("adaptive meshing")



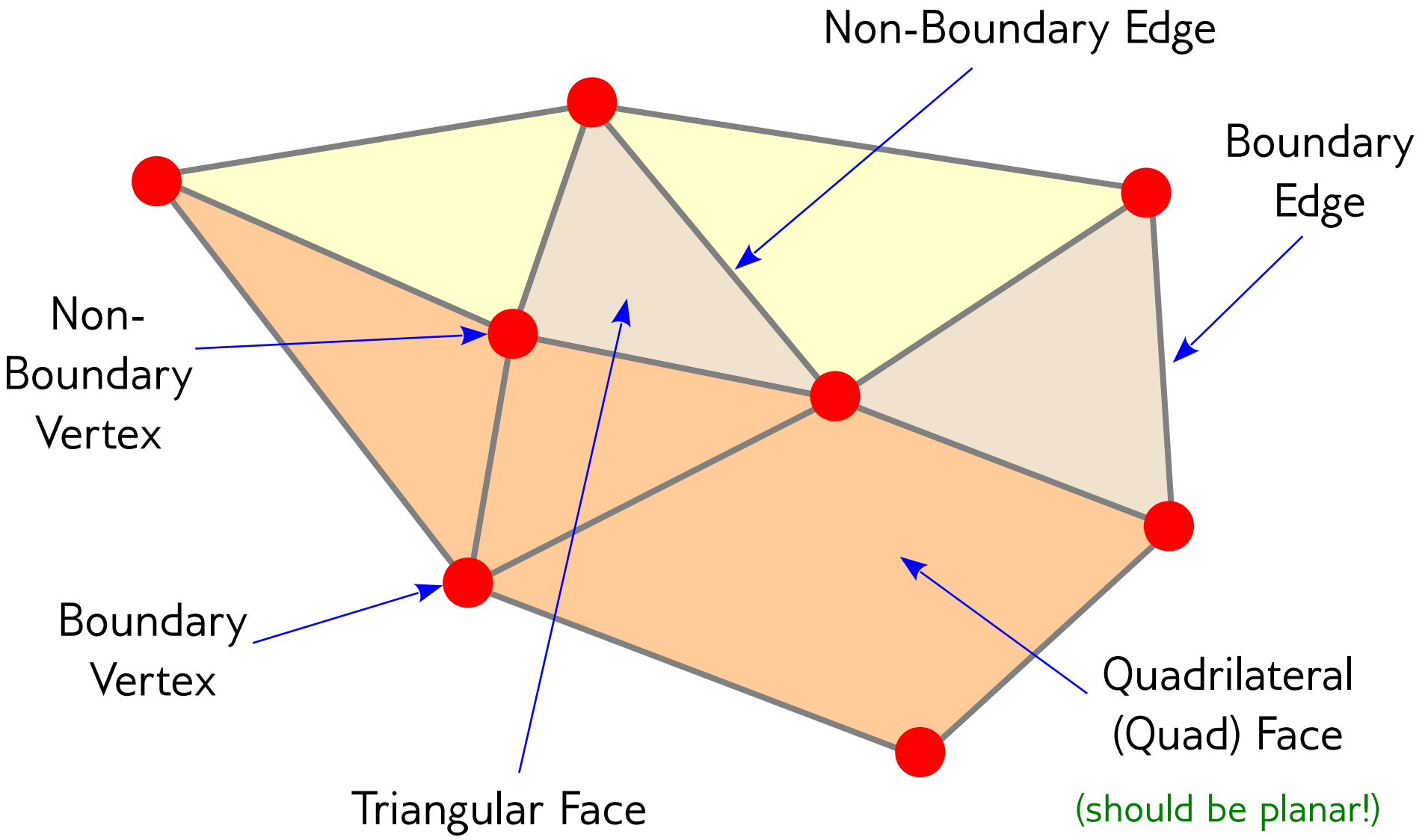
# Elements of a mesh



# Elements of a mesh

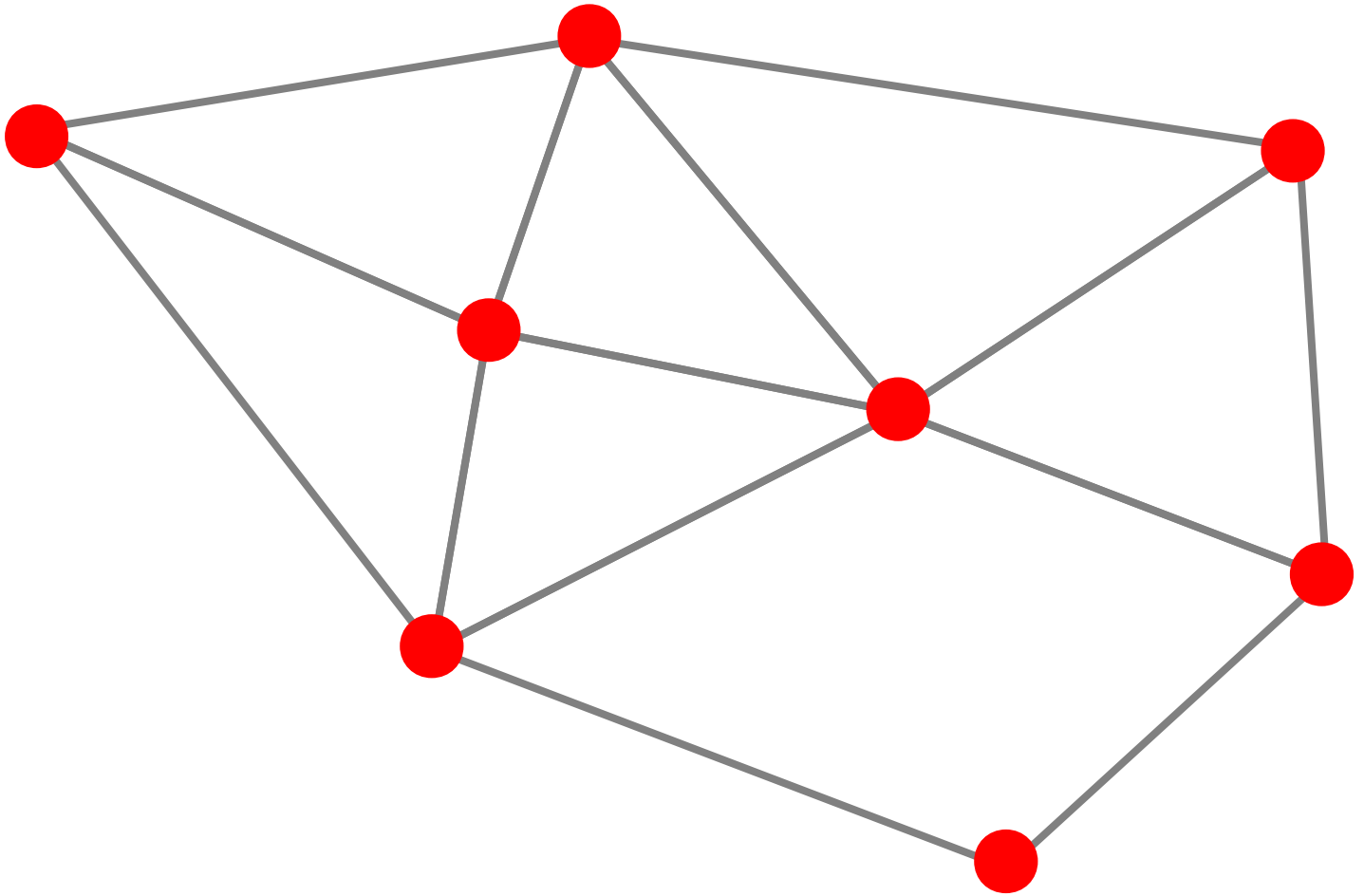


# Elements of a mesh

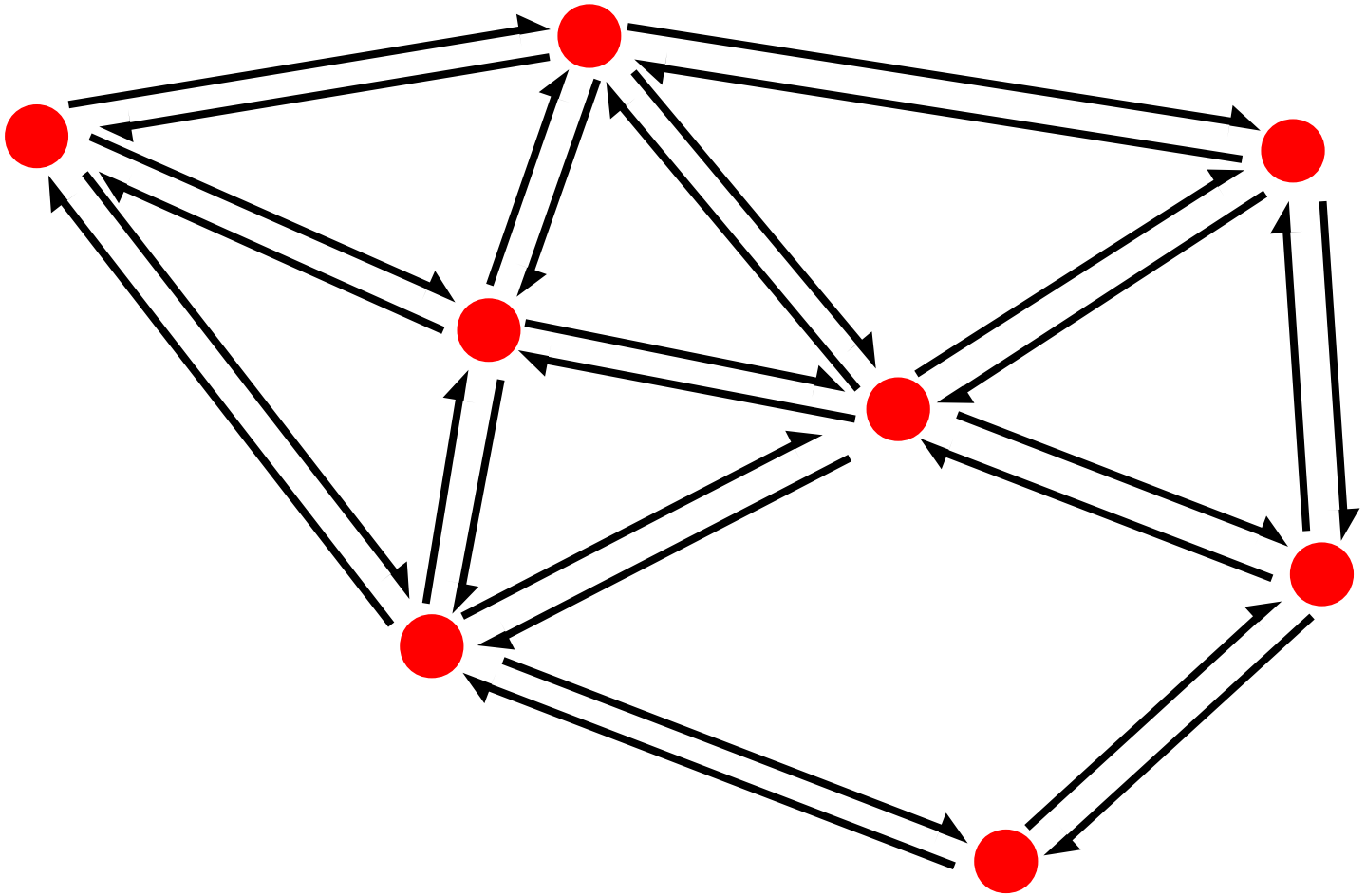


# A mesh is a graph

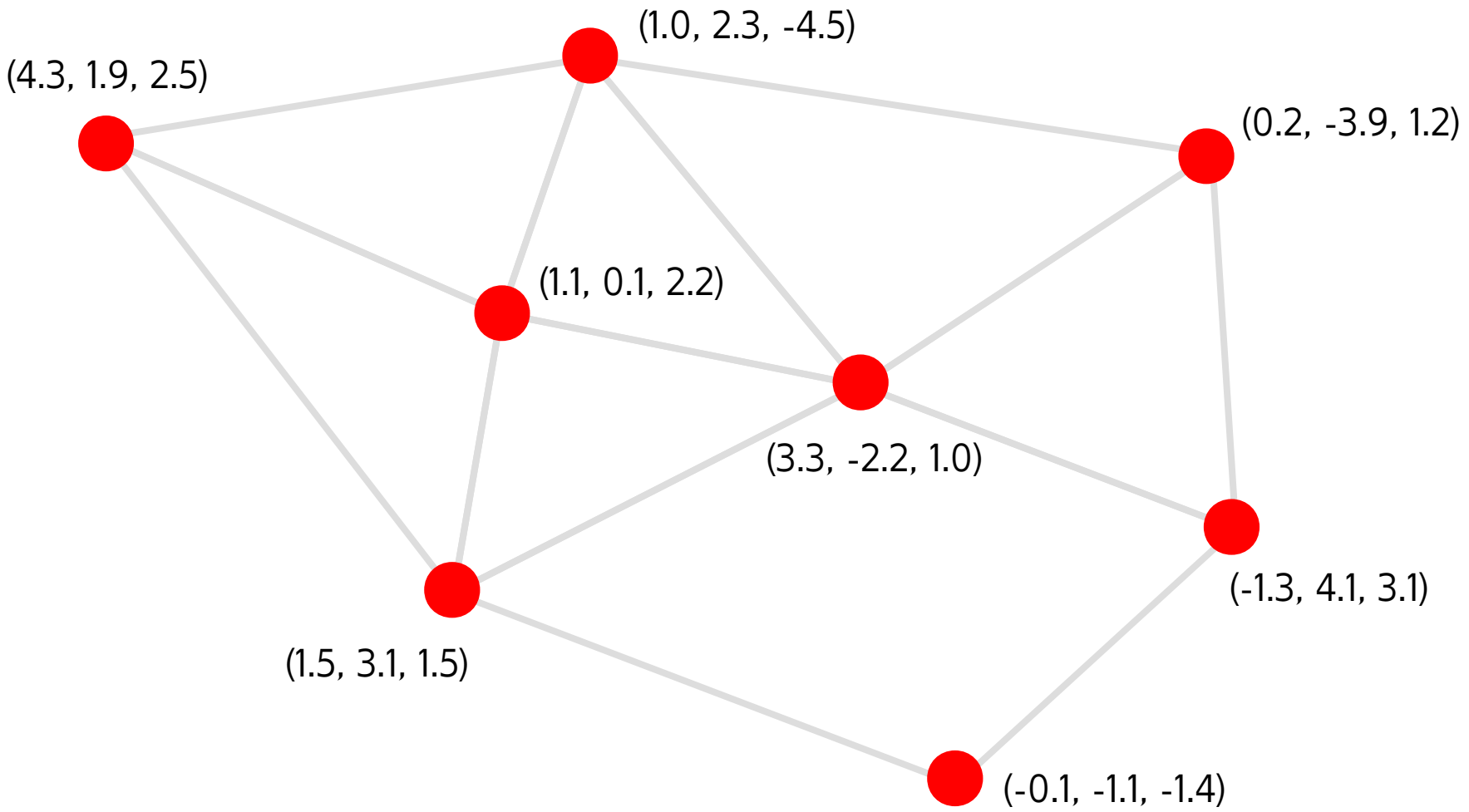
**This cannot be stressed strongly enough!**



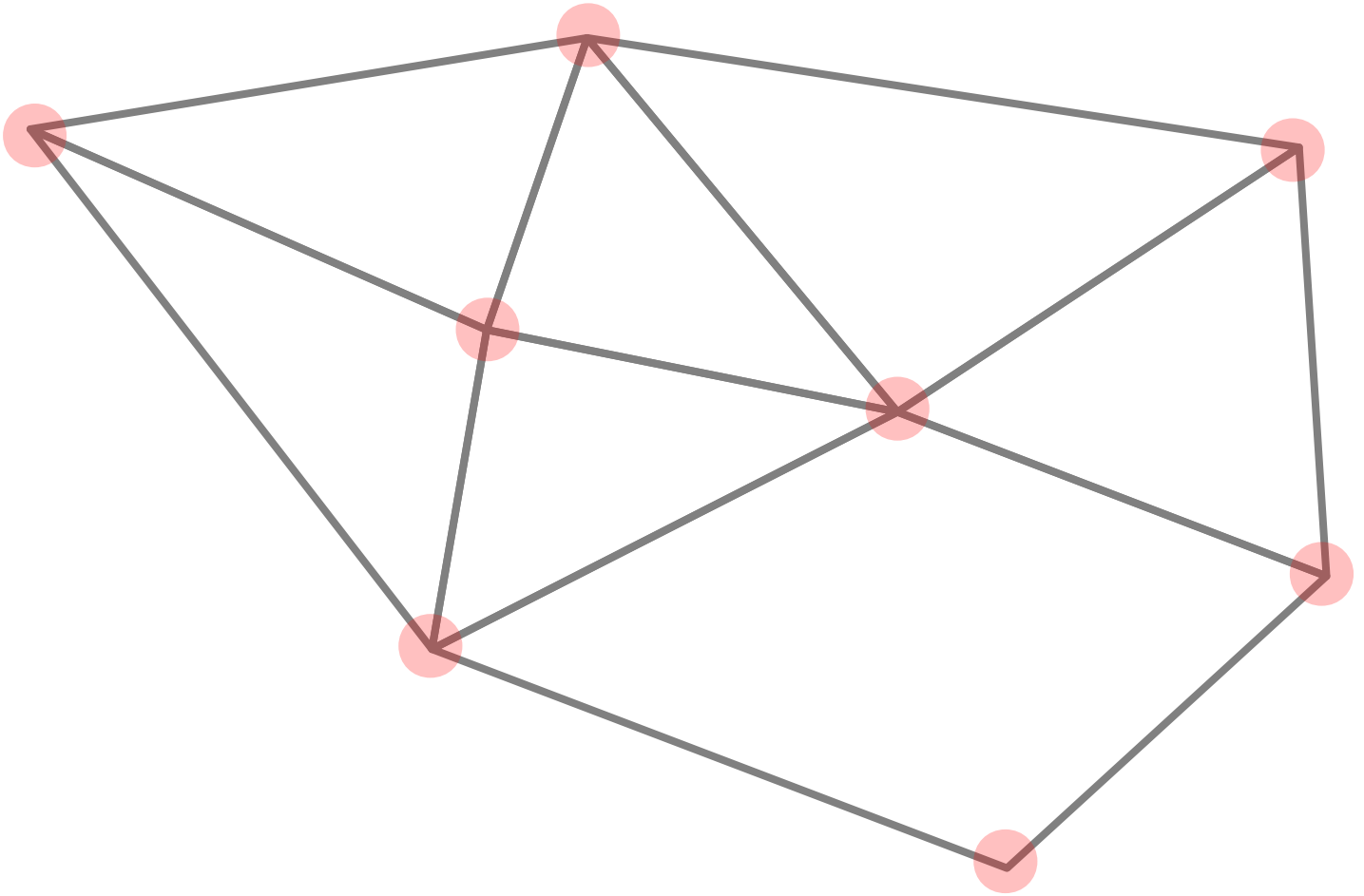
# A mesh is an undirected graph



# The vertex positions capture the **geometry** of the surface

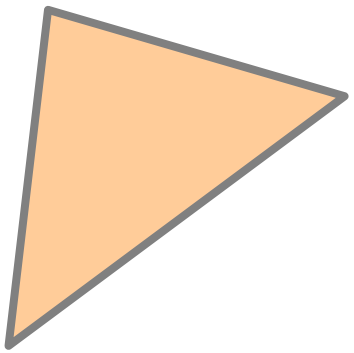


The mesh connectivity captures the **topology** of the surface

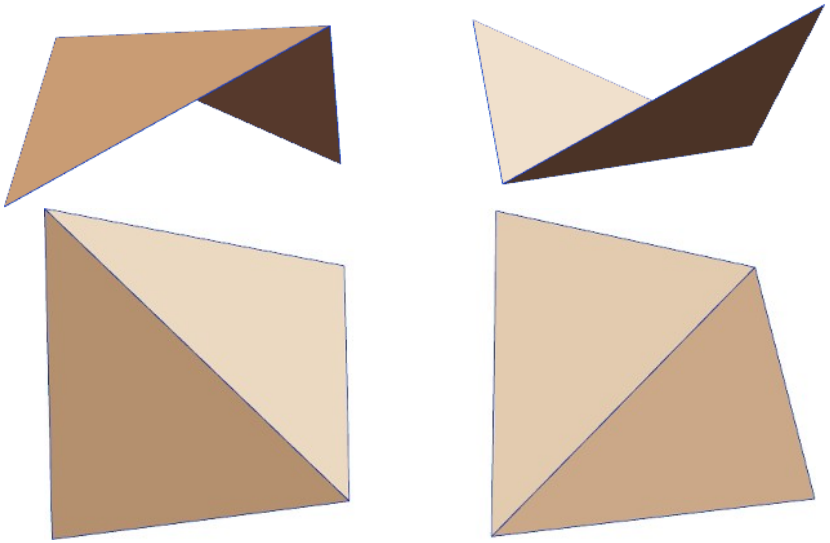


# Mesh Geometry: Planes and Normals

- Each polygon is (assumed to be) planar
  - Triangular faces are always planar
  - Quads and higher degree faces need not be
    - Ambiguity revealed by triangulation
  - Many mesh formats allow non-planar faces, but most algorithms assume planar faces. Caveat emptor.



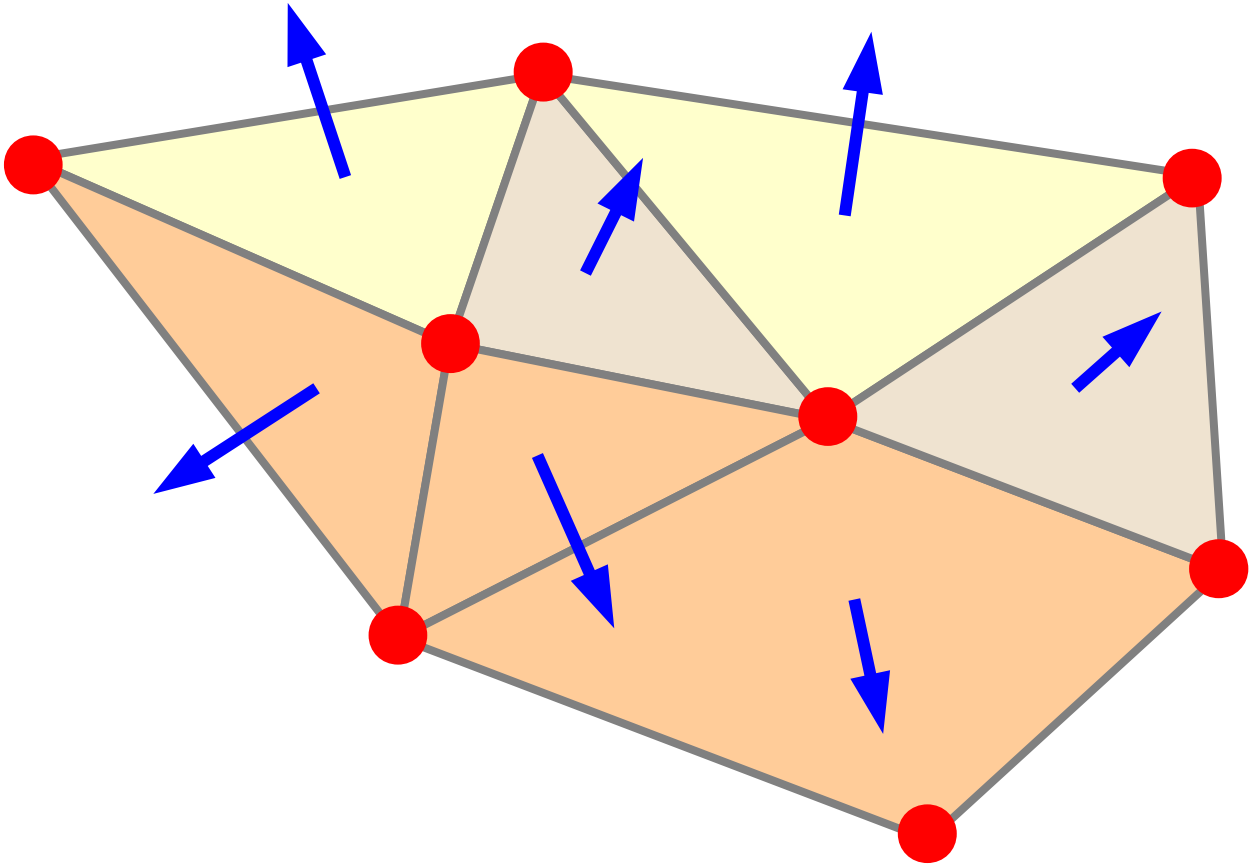
Always planar



Same 4 non-coplanar vertices, different geometry!

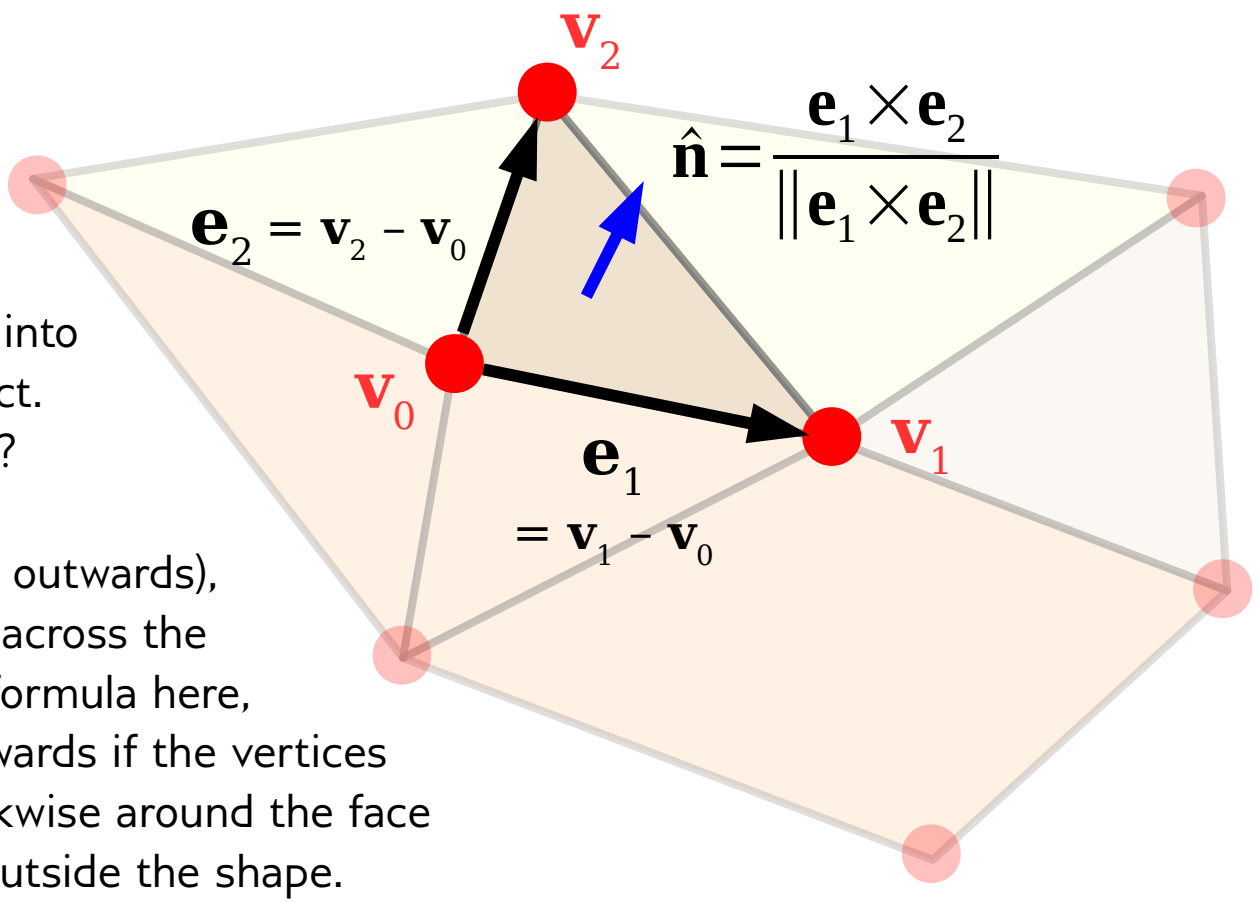
# Mesh Geometry: Planes and Normals

- The plane of each polygon has an associated normal vector



# Mesh Geometry: Planes and Normals

- The plane of each polygon has an associated normal vector

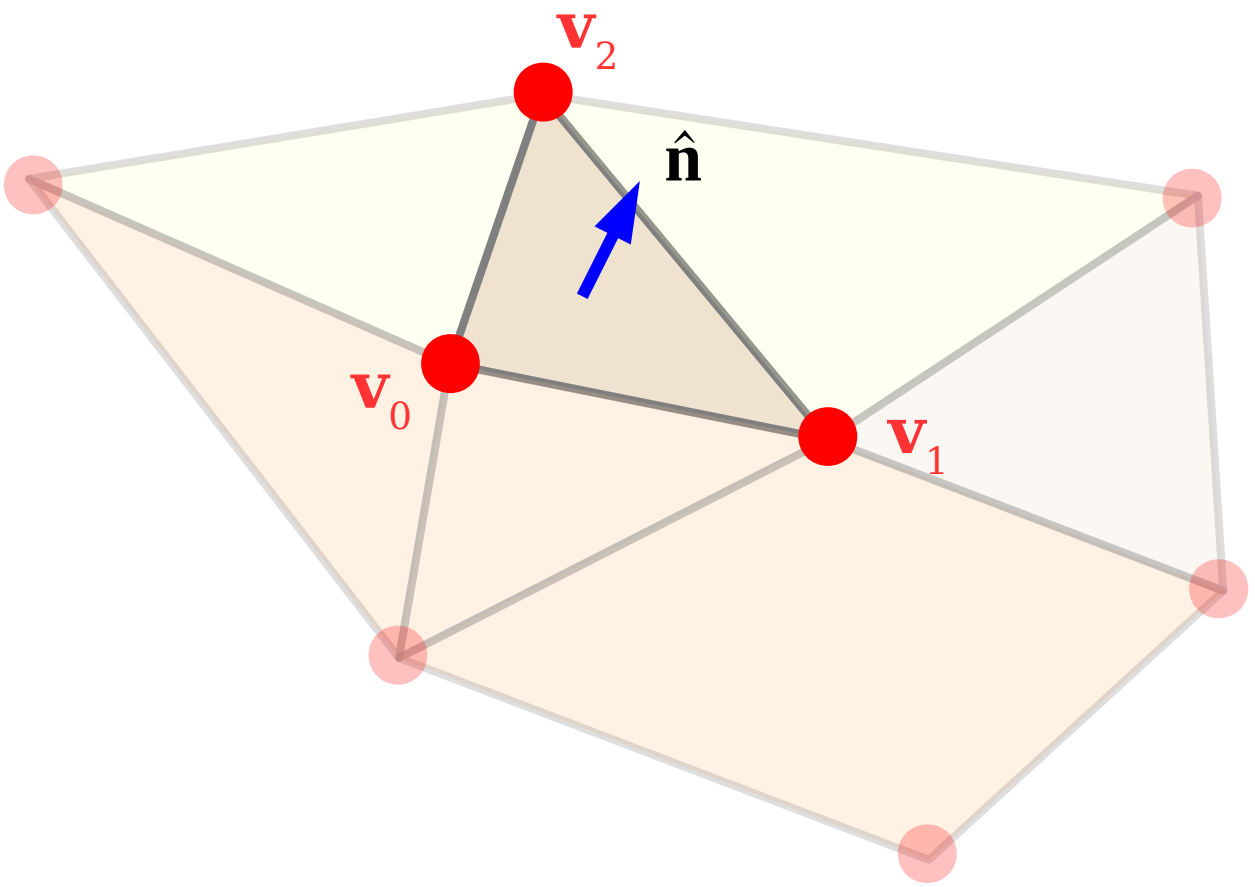



**Q:** The computed normal may point into or out of the object. Which one to pick?

**A:** Either (typically outwards), but be consistent across the shape! Using our formula here, the normal is outwards if the vertices wind counter-clockwise around the face when seen from outside the shape.

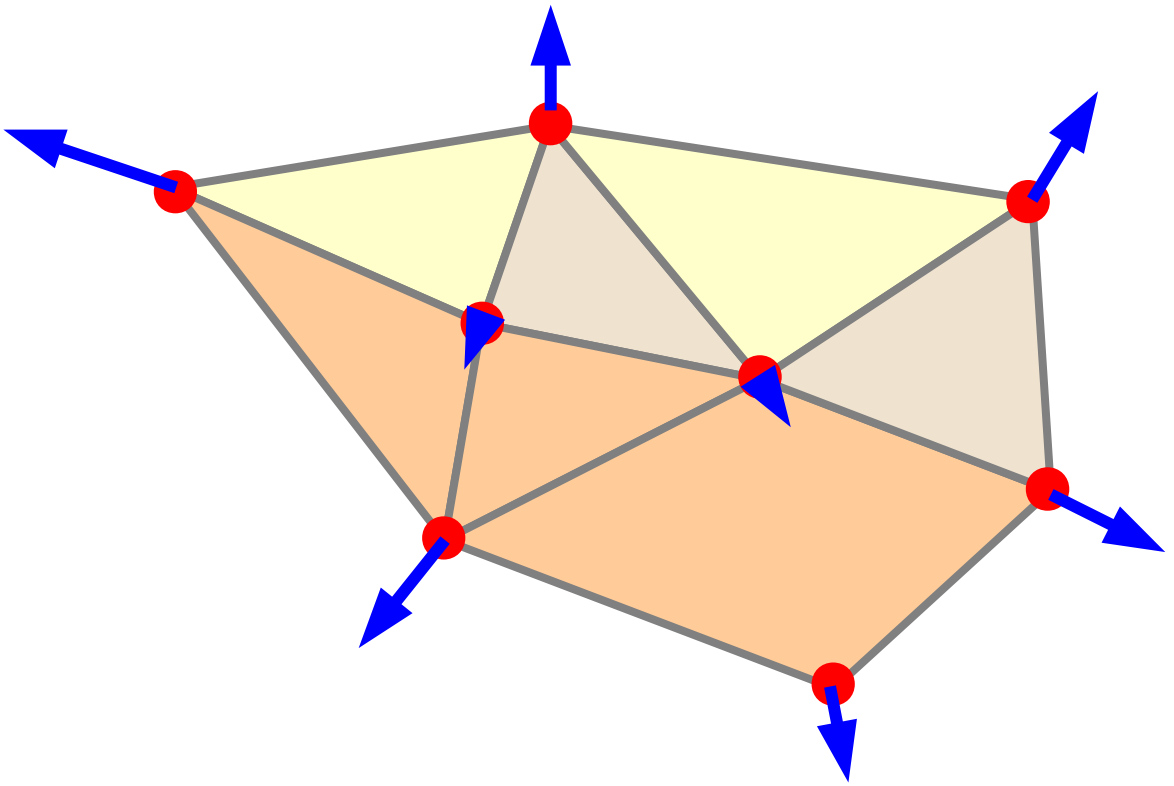
# Mesh Geometry: Planes and Normals

- The plane of each polygon has an associated plane equation: $\hat{\mathbf{n}} \cdot (\mathbf{p} - \mathbf{v}_0) = 0$



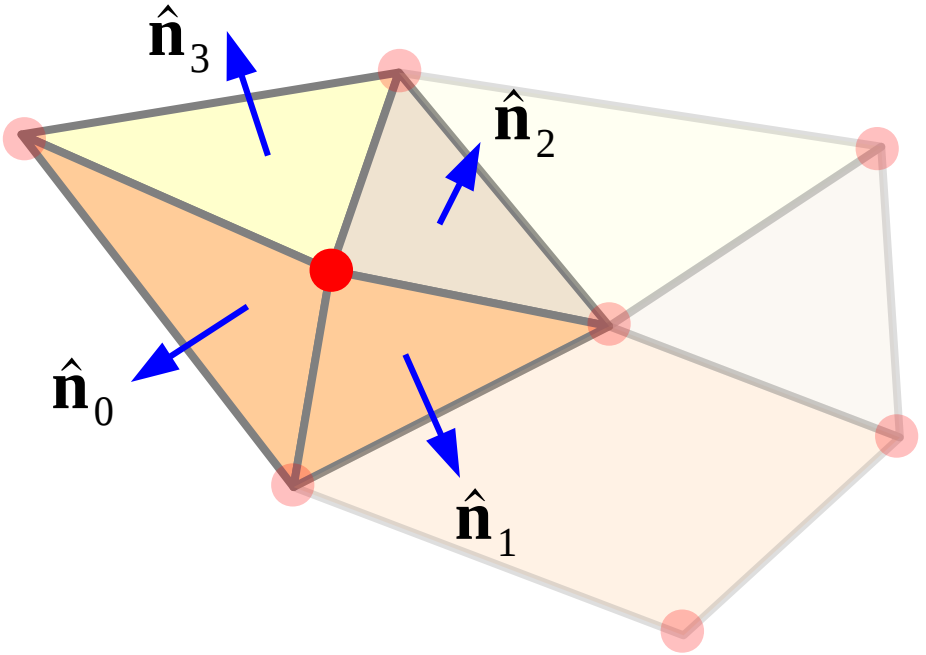
# Mesh Geometry: Planes and Normals

- We can also associate vertices with normals
  - Sometimes they come with the mesh (e.g. if they were estimated when the mesh was constructed from a point cloud)
  - Sometimes we have to estimate them



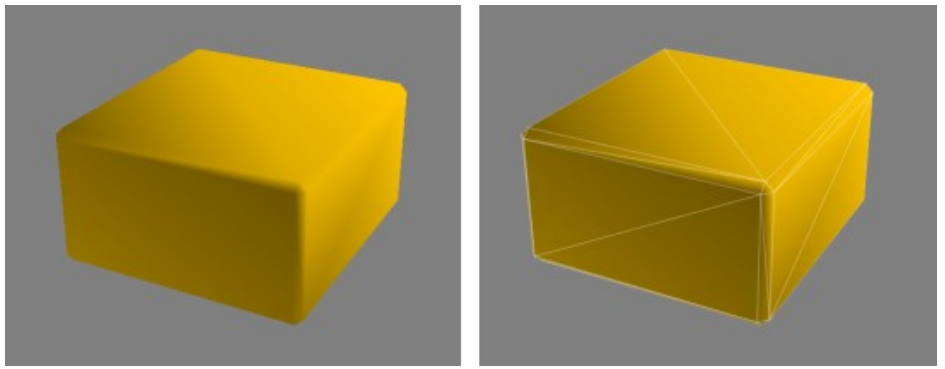
# Estimating vertex normals

- **Simplest:** Add up the normals of adjacent faces and unitize

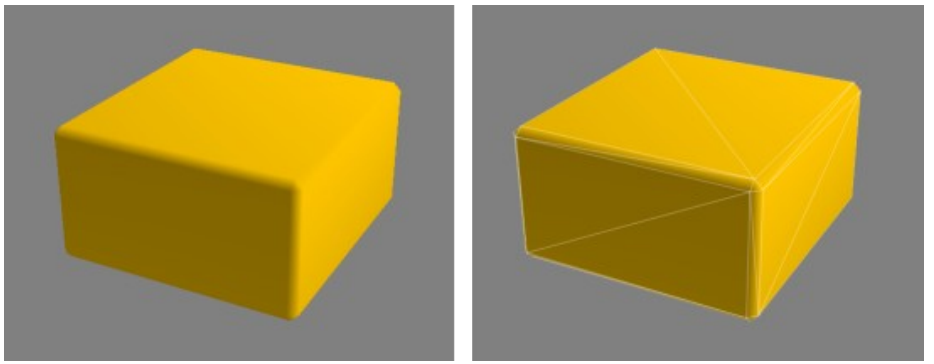


# Estimating vertex normals

- **Simplest:** Add up the normals of adjacent faces and unitize
- **Simple and usually a bit better:** Add up the normals of adjacent faces, weighted by face areas



Without area-weighting

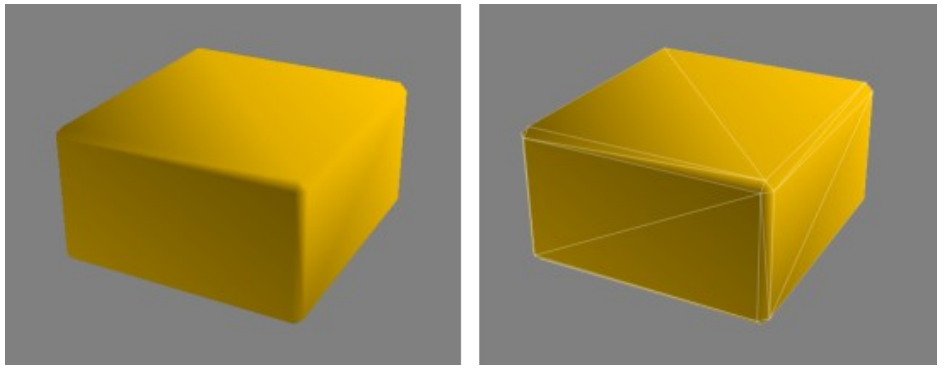


With area-weighting

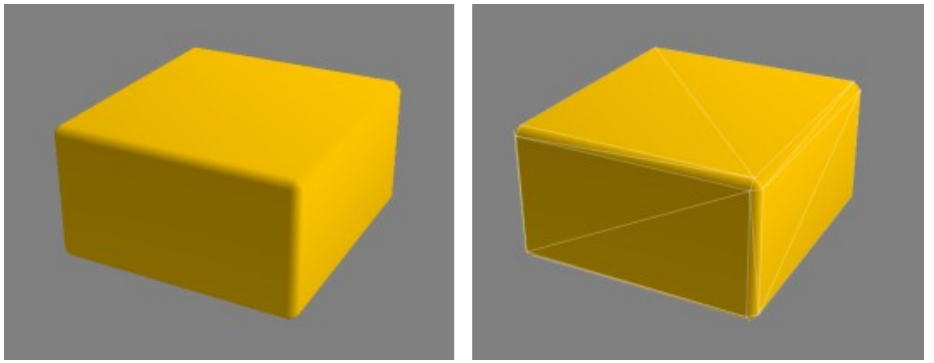
bytehazard.com/articles/vertnorm.html

# Estimating vertex normals

- **Simplest:** Add up the normals of adjacent faces and unitize
- **Simple and usually a bit better:** Add up the normals of adjacent faces, weighted by face areas
- **Complex:** Detect sharp edges



Without area-weighting



With area-weighting

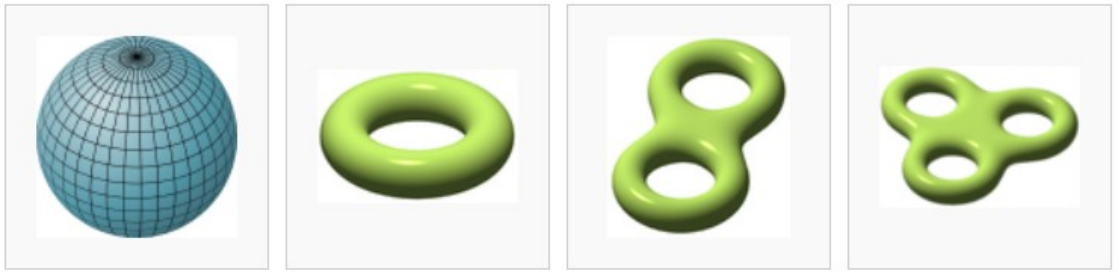
# Mesh Topology

- **Topology** (loosely): The structure of a shape ignoring any measurements of distance, angle etc
  - i.e. the properties invariant to bending, twisting, folding, stretching... (but not tearing)
- E.g. Genus: The number of handles in a shape

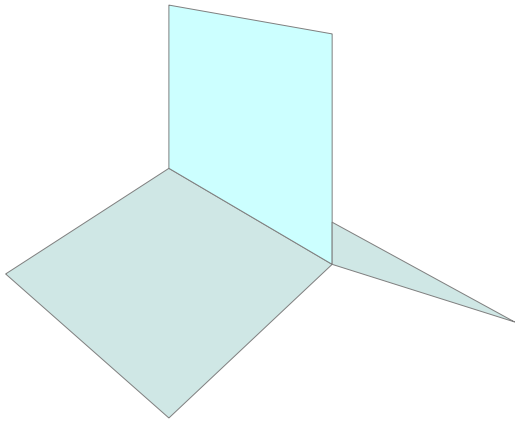

Wikipedia

# Mesh Topology

- **Manifold:** A topological space that is locally Euclidean (neighborhood has the topology of the unit ball)




Some manifold shapes



Not manifold

Manifold structure of a surface is approximated by its mesh connectivity