

Mesh Processing

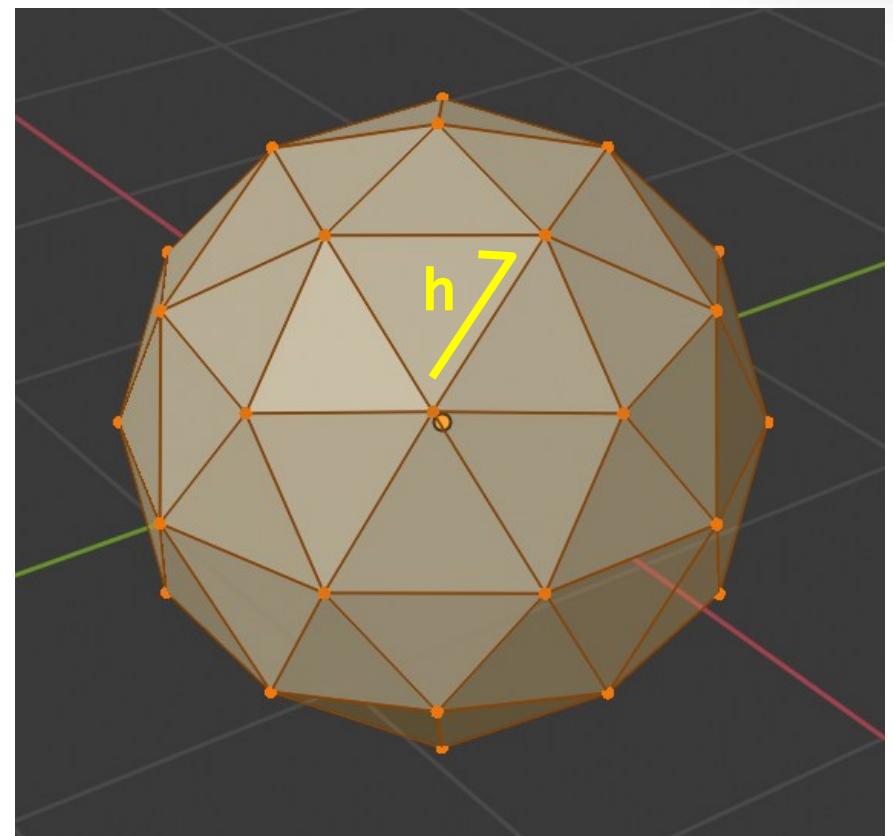
Half Edge data structure is useful...

- To answer local questions about mesh geometry and topology
- Such questions must be answered in many geometry processing algorithms
- Let's look at some example questions where the input can be a half edge to specify
 - An edge in the mesh
 - A vertex in the mesh, at the head of the half edge
 - A face in the mesh, found to the left of the half edge

Midpoint of an Edge?



```
HEdge {  
    HEdge o; // opposite (i.e., twin, pair)  
    HEdge n; // next  
    Vertex v; // vertex (i.e., head)  
    Face f; // left face  
}  
  
Face {  
    HEdge h; // some adjacent h-edge  
    // other per-face data: area, normal, etc.  
}  
  
Vertex {  
    HEdge h; // some incident h-edge  
    Vec3 p; // position data  
}  
  
Vec3 midpoint( HEdge h ) {  
    // TODO:}
```

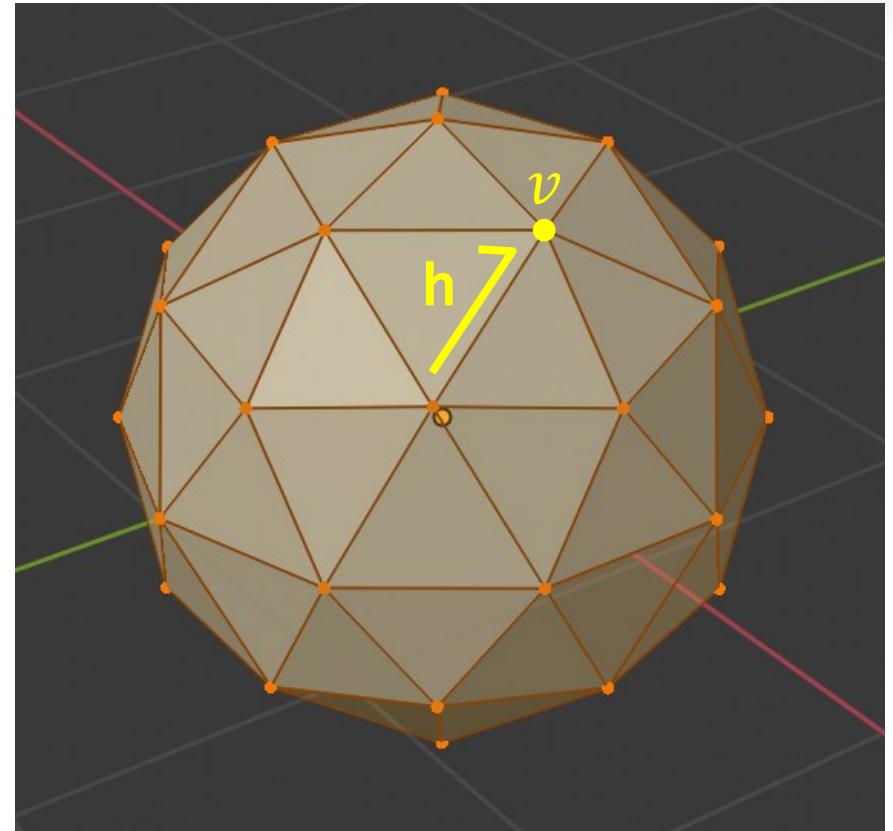


Mesh Processing

How might we use a half edge data structure to...

1. Compute vertex valence (i.e., degree)?

```
HEdge {  
    HEdge o; // opposite (i.e., twin, pair)  
    HEdge n; // next  
    Vertex v; // vertex (i.e., head)  
    Face f; // left face  
}  
  
Face {  
    HEdge h; // some adjacent h-edge  
    // other per-face data: area, normal, etc.  
}  
  
Vertex {  
    HEdge h; // some incident h-edge  
    // other per-vertex data: position, normal, etc.  
}  
  
int valence( Vertex v ) {  
    // TODO  
}
```

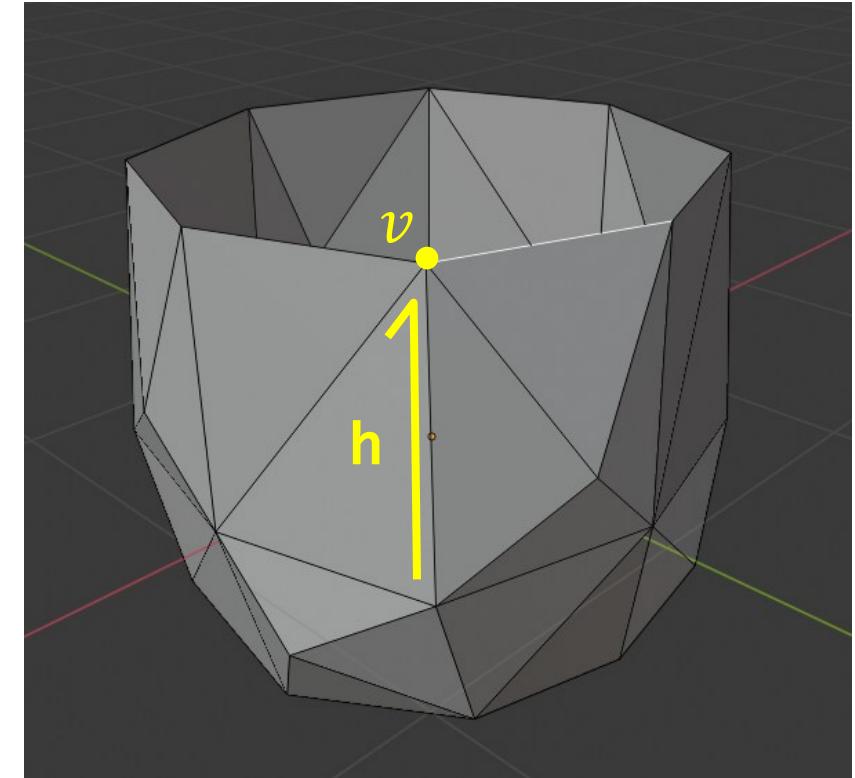


Mesh Processing

How might we use a half edge data structure to...

2. Compute vertex valence in mesh with boundaries?

```
HEdge {  
    HEdge o; // opposite (i.e., twin, pair) can be null  
    HEdge n; // next  
    Vertex v; // vertex (i.e., head)  
    Face f; // left face  
}  
  
Face {  
    HEdge h; // some adjacent h-edge  
    // other per-face data: area, normal, etc.  
}  
  
Vertex {  
    HEdge h; // some incident h-edge  
    // other per-vertex data: position, normal, etc.  
}  
  
int valence( Vertex v ) {  
    // TODO  
}
```



Mesh Processing

How might we use a half edge data structure to compute...

3. Normal of a triangular face? $N_f = \frac{(v_1 - v_0) \times (v_2 - v_0)}{\|(v_1 - v_0) \times (v_2 - v_0)\|}$

4. Compute an approximate vertex normal? $N_v = \frac{\sum_f N_f}{\|\sum_f N_f\|}$

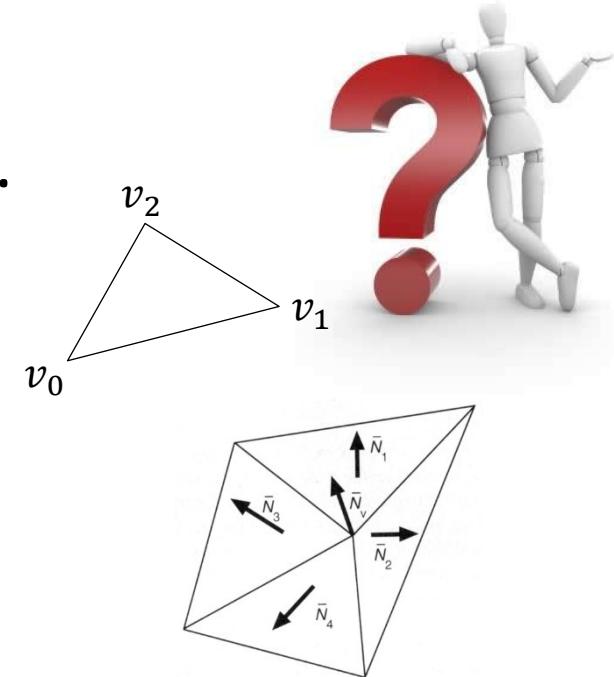
- Other options for vertex normal?

5. Area of a triangular face? $A_f = \frac{1}{2} \|(v_1 - v_0) \times (v_2 - v_0)\|$

- How would you determine a signed area?

6. Area of a polygonal face?

- What assumptions are necessary to compute the area of an n-gon?



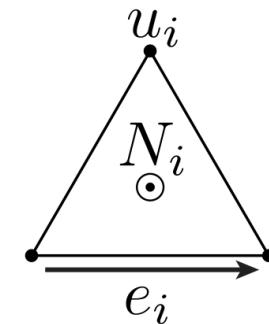
Mesh Processing

How might we use a half edge data structure to compute...



7. Gradient, in a triangle, of a quantity u stored at vertices?
(This gradient is a 3D vector lying in the plane of the triangle)

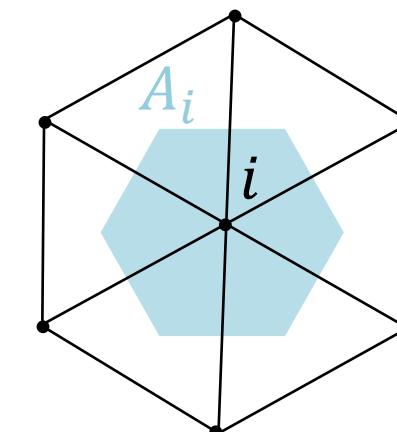
$$\nabla u = \frac{1}{2A_f} \sum_i u_i (N \times e_i)$$



8. Area associated with a vertex?

$$A_i = \frac{1}{3} \sum_f A_f$$

- Other options for vertex area?



Mesh Processing

How might we use a half edge data structure to compute...

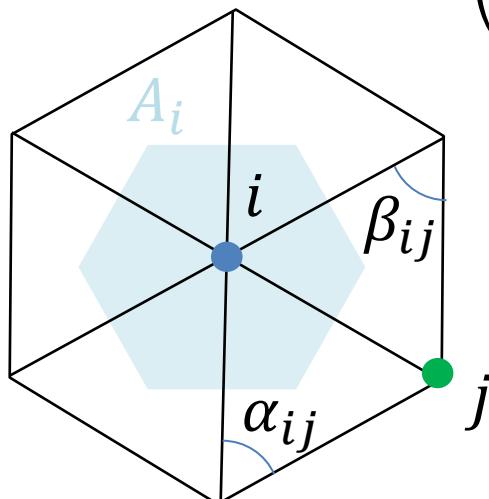


9. Given quantity u_i at vertex i , compute mesh Laplacian?

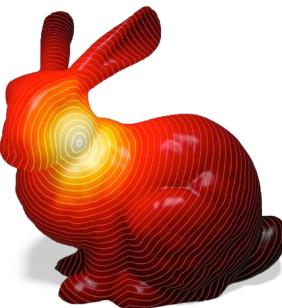
Laplacian is a second derivative $\nabla^2 u \equiv \nabla \cdot \nabla u \equiv \Delta u$, and in the plane, $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$

Think of Laplacian as diffusion operator

$$\begin{aligned}(Lu)_i &= \frac{1}{2A_i} \sum_{j \text{ adj } i} (\cot \alpha_{ij} + \cot \beta_{ij})(u_j - u_i) \\ &= \left(-\frac{1}{2A_i} \sum_{j \text{ adj } i} (\cot \alpha_{ij} + \cot \beta_{ij}) \right) u_i + \sum_{j \text{ adj } i} \left(\frac{1}{2A_i} (\cot \alpha_{ij} + \cot \beta_{ij}) \right) u_j\end{aligned}$$



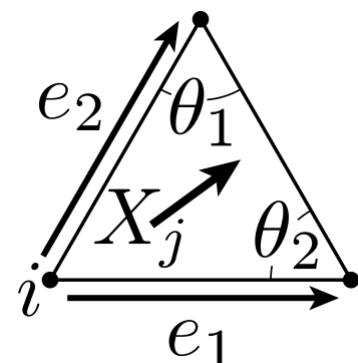
- For intuition in planar equilateral triangular tiling
 - Cotangent of 60 degrees is $\frac{1}{\sqrt{3}}$
 - Laplacian weights are $\frac{1}{A_i\sqrt{3}}$ times -6 on middle vertex and times 1 on adjacent vertices



Mesh Processing

10. Compute the divergence at a vertex of vectors stored on faces?

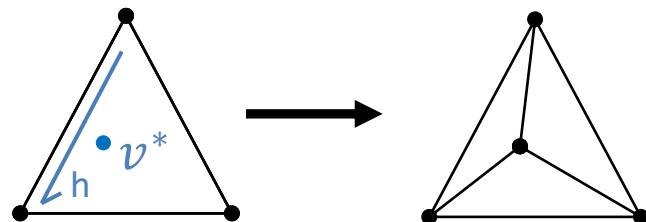
$$\nabla \cdot X = \frac{1}{2} \sum_j \cot \theta_1 (e_1 \cdot X_j) + \cot \theta_2 (e_2 \cdot X_j)$$



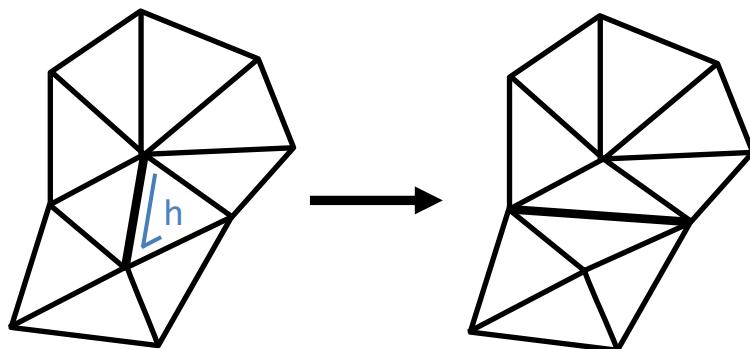
Mesh Processing: Topology Changes



11. Insert a vertex into a face, dividing the face into 3 new faces?



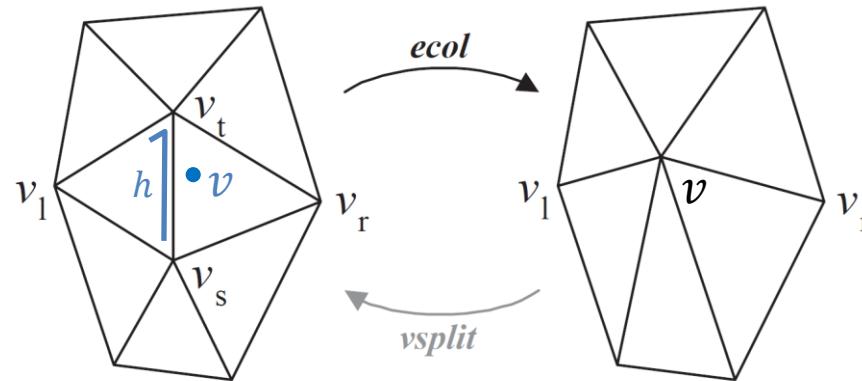
12. Edge flip



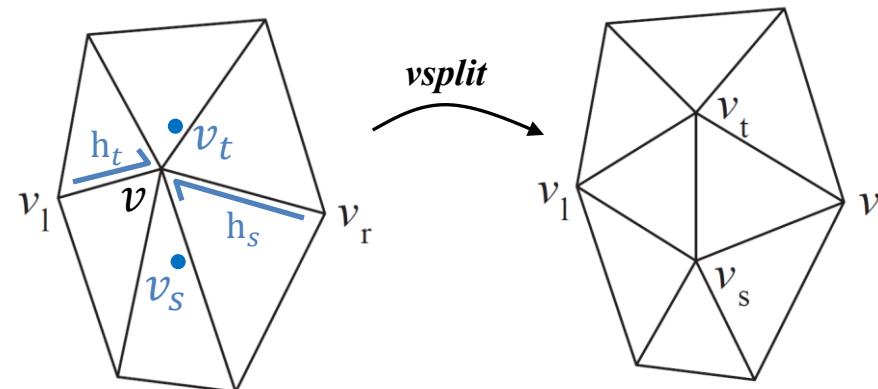
Mesh Processing: Topology Changes



13. Edge collapse

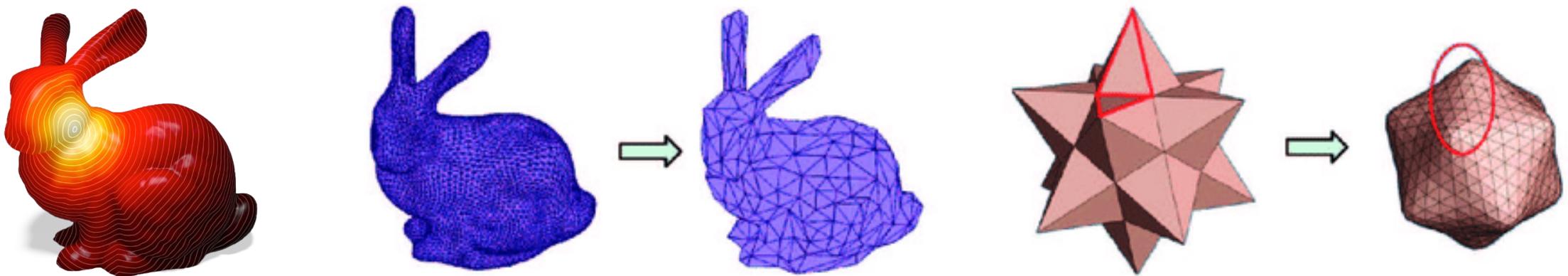


14. Vertex split



Why care about all these quantities?

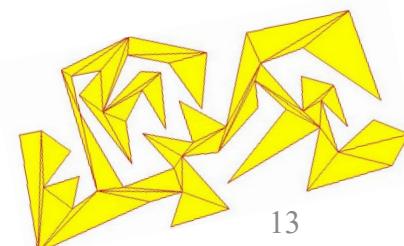
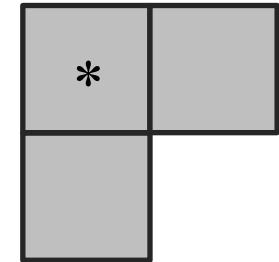
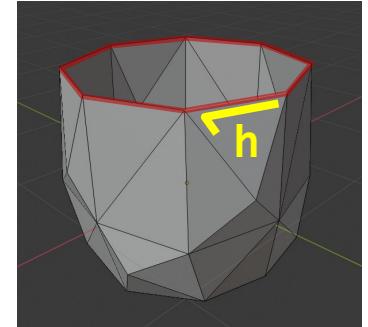
- Three examples
 - Solving PDEs on meshes, e.g., to compute geodesics from heat
 - Mesh simplification, using local metrics and local modifications
 - Mesh subdivision, using local rules





More examples

- a) Given a boundary half edge, create an ordered list of edges forming a simple closed curve
- b) Given a half edge, check if left face can be removed without making the mesh non-manifold (e.g., at right, * can't be removed)
- c) Given a half edge, remove its left face
- d) Given a half edge, repeatedly merge with adjacent co-planar faces to form an n-gon (possibly non-convex)
- e) Given a half edge whose left face is an n-gon (possibly non-convex), return an *ear*, i.e., two edges that form a triangle that lies completely inside the n-gon
- f) Given a half edge, repeatedly find ears and split the n-gon along third edge of triangle to triangulate a planar n-gon



Review and More Information

- Last class: FCG Chapter 12.1
 - basic definitions and mesh data structures
- Mesh Simplification [Zorin Schroder 2000]
<https://mrl.cs.nyu.edu/publications/subdiv-course2000/coursenotes00.pdf>
 - Sections 4.1-4.4 for overview, notation, Loop, and Catmull-Clark (don't worry about Butterfly)
- Surface Simplification Using Quadric Error Metrics [Garland Heckbert 1997] <https://www.cs.cmu.edu/~garland/Papers/quadratics.pdf>
- Not covered this year: Geodesics in Heat [Crane et al. 2017]
<https://www.cs.cmu.edu/~kmcrane/Projects/HeatMethod/>