COS 426
Computer Graphics
Princeton University

Tianqiang Liu (Tim)

Feb 29, 2012

Thanks to Vladimir Kim for providing the slides!

Mesh Processing

- **Half Edge representation**
  - Data structure
  - How to load a shape?
  - How to find faces adjacent to a vertex?
  - How to collapse an edge?
  - How to flip an edge?

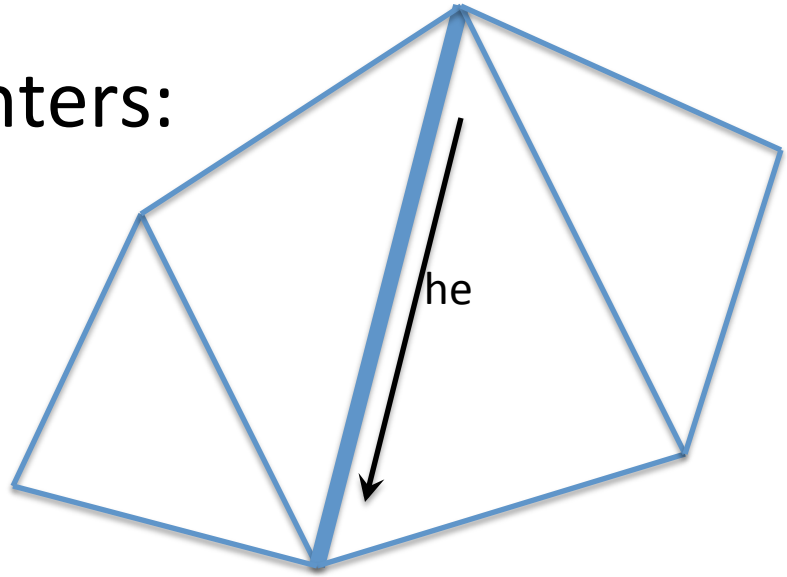
Half Edge

- http://groups.csail.mit.edu/graphics/classes/6.838/S98/meetings/m4/IV.HalfEdge.html

- Mesh Represented by:
  - list of half edges (|HE|)
  - list of vertices (|V|)
  - list of faces (|F|)

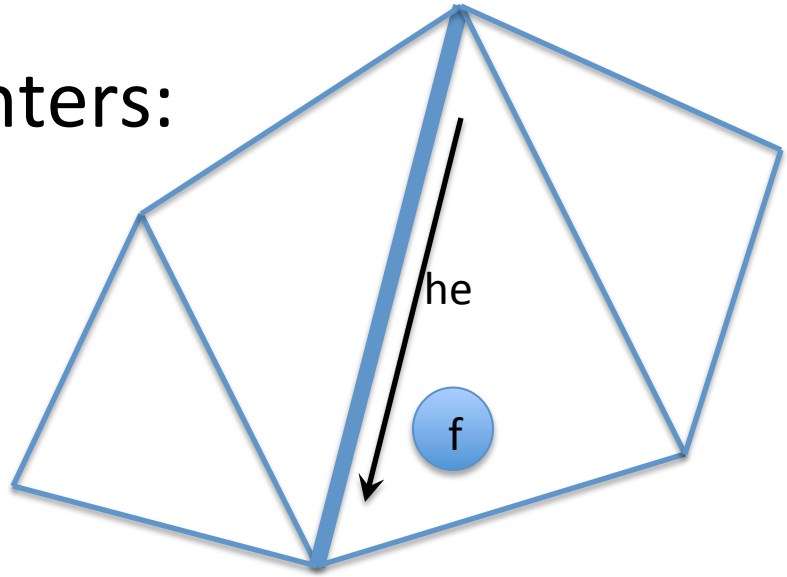
Data structure

- A half edge contains 4 pointers:



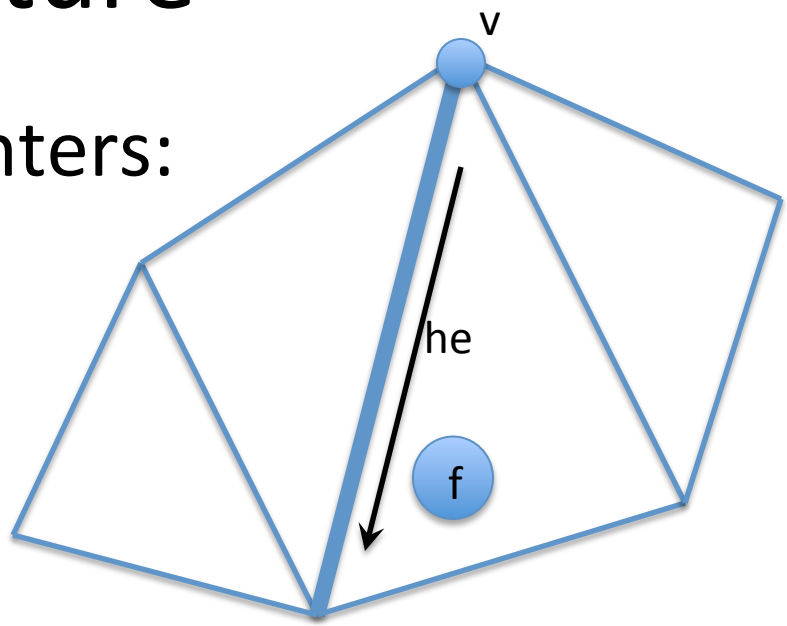
Data structure

- A half edge contains 4 pointers:
  - adjacent face f (to the left)



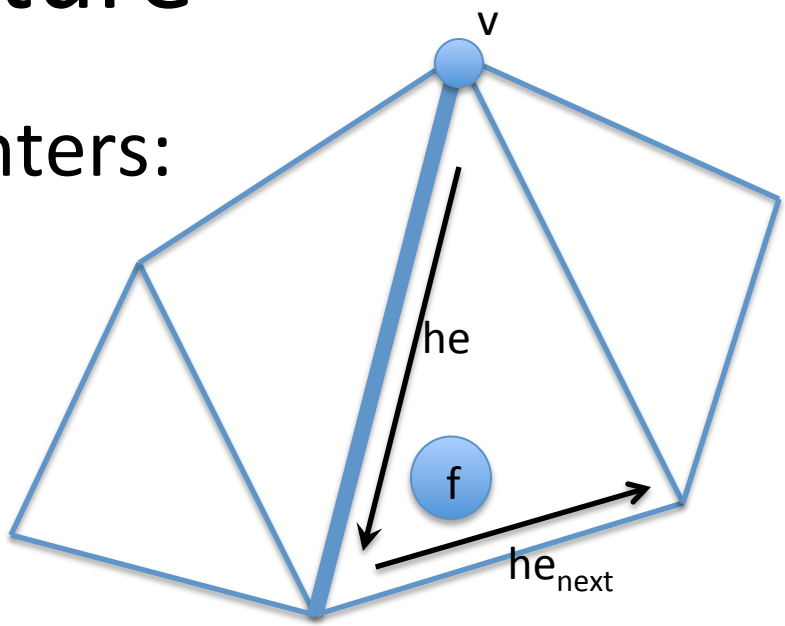
Data structure

- A half edge contains 4 pointers:
  - adjacent face f (to the left)
  - source vertex v



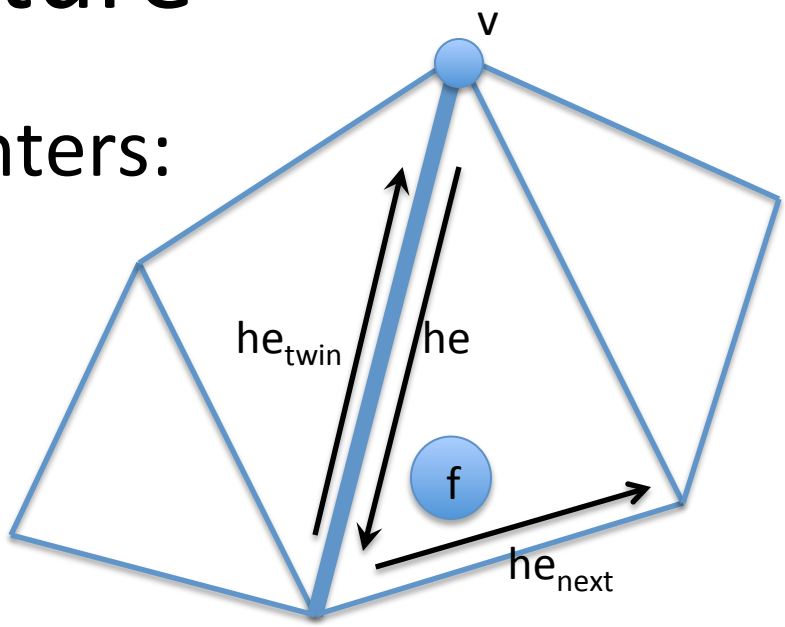
Data structure

- A half edge contains 4 pointers:
  - adjacent face f (to the left)
  - source vertex v
  - next half edge he_{next}



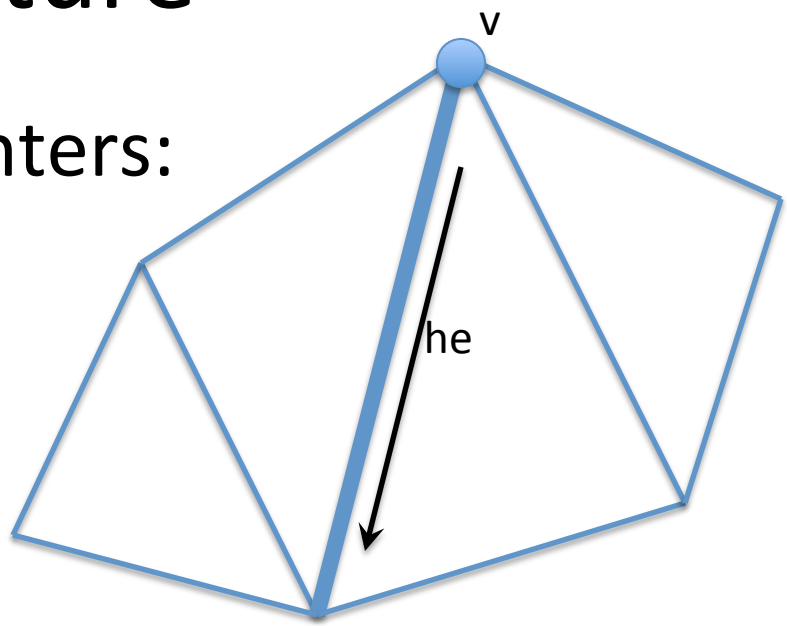
Data structure

- A half edge contains 4 pointers:
  - adjacent face f (to the left)
  - source vertex v
  - next half edge he_{next}
  - 'twin' half edge he_{twin}



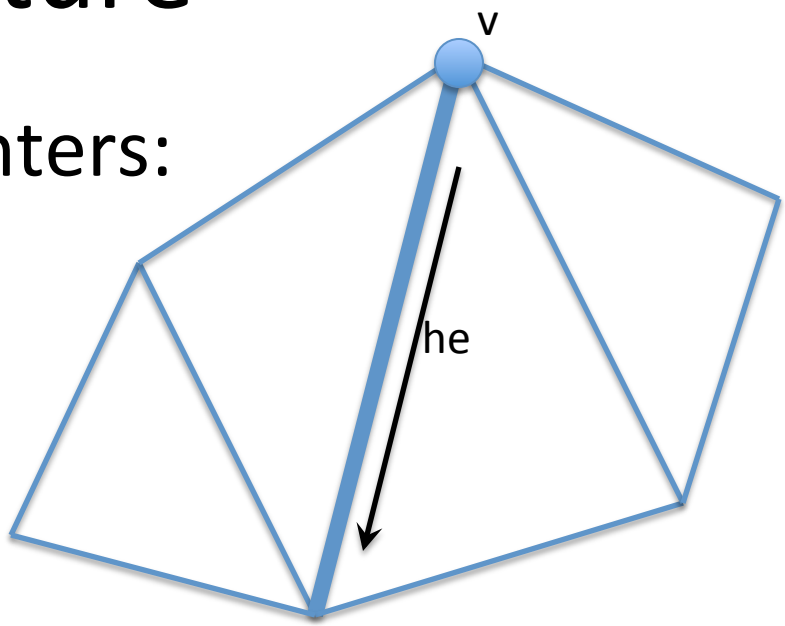
Data structure

- A half edge contains 4 pointers:
  - adjacent face f (to the left)
  - source vertex v
  - next half edge he_{next}
  - 'twin' half edge he_{twin}
- A vertex v has pointer to
  - an outgoing half edge



Data structure

- A half edge contains 4 pointers:
  - adjacent face f (to the left)
  - source vertex v
  - next half edge he_{next}
  - 'twin' half edge he_{twin}
- A vertex has pointer to
  - an outgoing half edge
- A face has pointer to
  - a boundary half edge

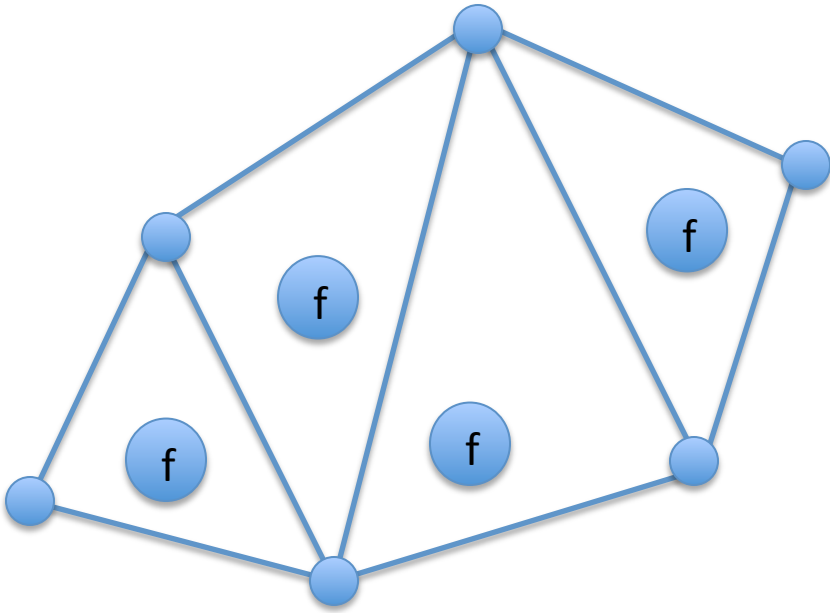


Mesh Processing

- **Half Edge representation**
  - Data structure
  - **How to load a shape?**
  - How to find faces adjacent to a vertex?
  - How to collapse an edge?
  - How to flip an edge?

Example Shape

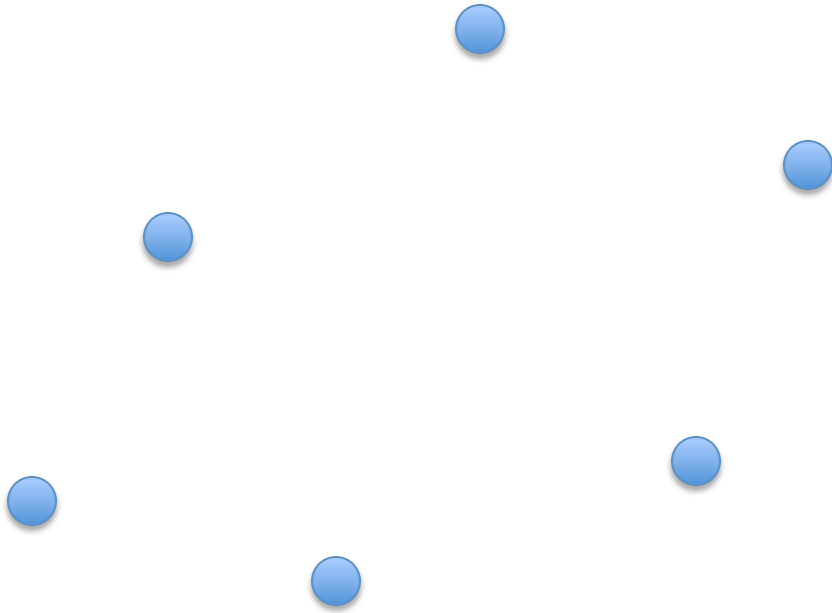
- *.off: vertices + triangles



?

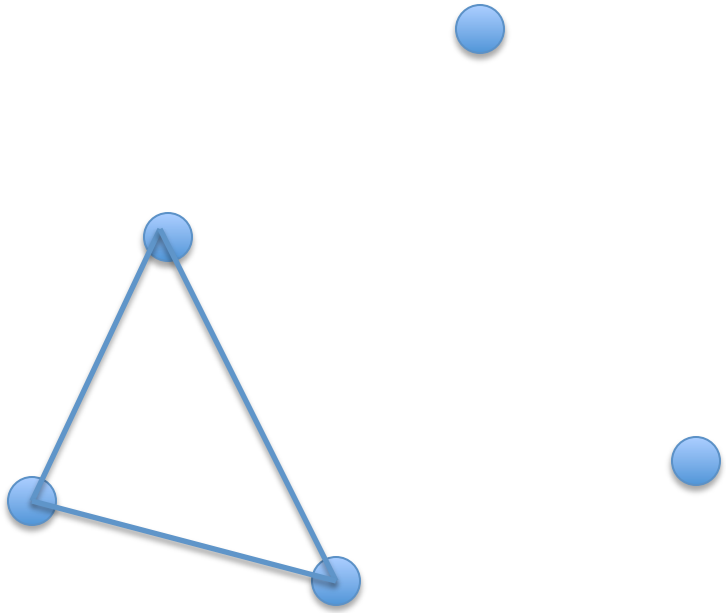
Example Shape

- *.off: vertices + triangles
- Add vertices to the list
  - only coordinates



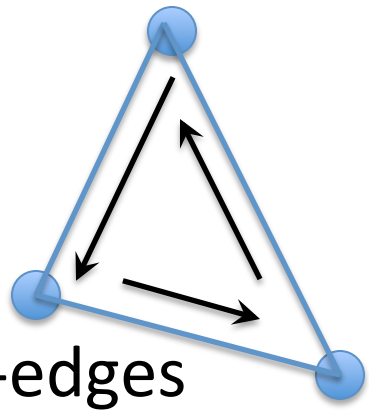
Example Shape

- *.off: vertices + triangles
- Add vertices to the list
  - only coordinates
- Add half-edges with faces



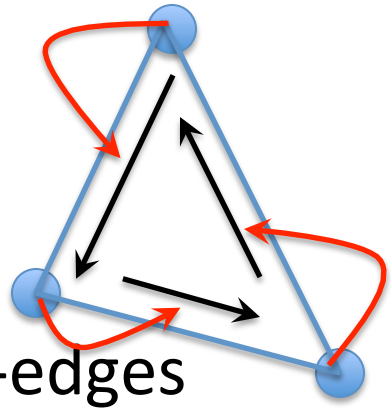
Example Shape

- *.off: vertices + triangles
- Add vertices to the list
  - only coordinates
- Add half-edges with faces
  - sufficient to add inner half-edges



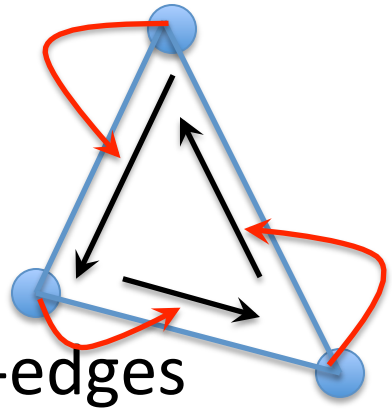
Example Shape

- *.off: vertices + triangles
- Add vertices to the list
  - only coordinates
- Add half-edges with faces
  - sufficient to add inner half-edges
  - if necessary: update vertex pointers to half edges



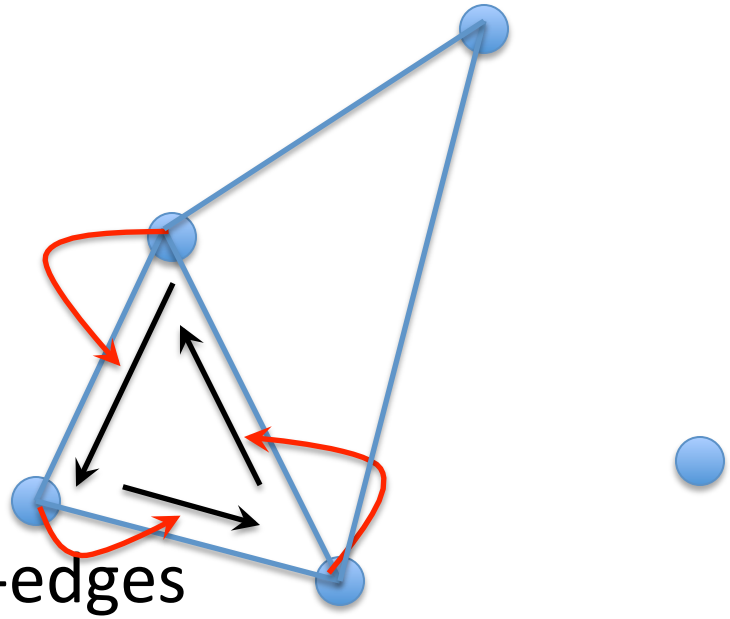
Example Shape

- *.off: vertices + triangles
- Add vertices to the list
  - only coordinates
- Add half-edges with faces
  - sufficient to add inner half-edges
  - if necessary: update vertex pointers to half edges
  - each half-edge: pointer to ‘next’, pointer to ‘face’
  - face: pointer to one of the inner half-edges



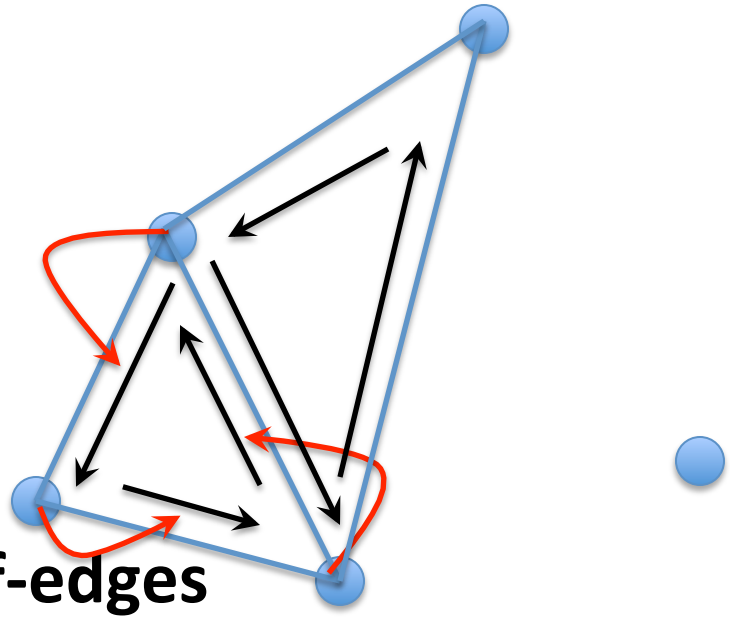
Example Shape

- *.off: vertices + triangles
- Add vertices to the list
  - only coordinates
- Add half-edges with faces
  - sufficient to add inner half-edges
  - if necessary: update vertex pointers to half edges
  - each half-edge: pointer to ‘next’, pointer to ‘face’
  - face: pointer to one of the inner half-edges



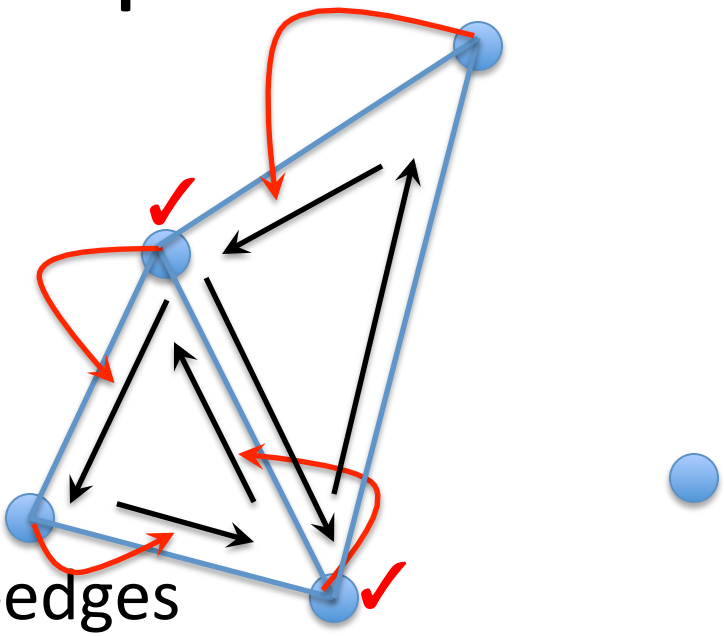
Example Shape

- *.off: vertices + triangles
- Add vertices to the list
  - only coordinates
- Add half-edges with faces
  - **sufficient to add inner half-edges**
  - if necessary: update vertex pointers to half edges
  - each half-edge: pointer to ‘next’, pointer to ‘face’
  - face: pointer to one of the inner half-edges

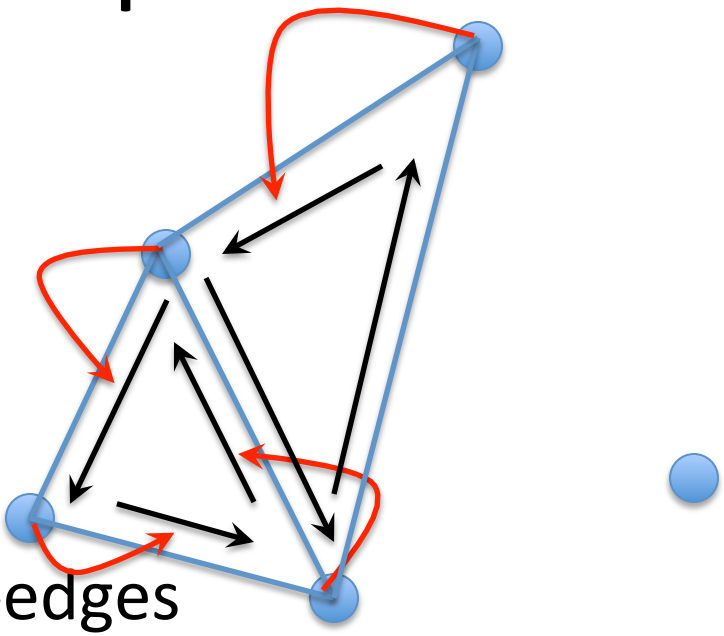


Example Shape

- *.off: vertices + triangles
- Add vertices to the list
  - only coordinates
- Add half-edges with faces
  - sufficient to add inner half-edges
  - **if necessary: update vertex pointers to half edges**
  - each half-edge: pointer to ‘next’, pointer to ‘face’
  - face: pointer to one of the inner half-edges

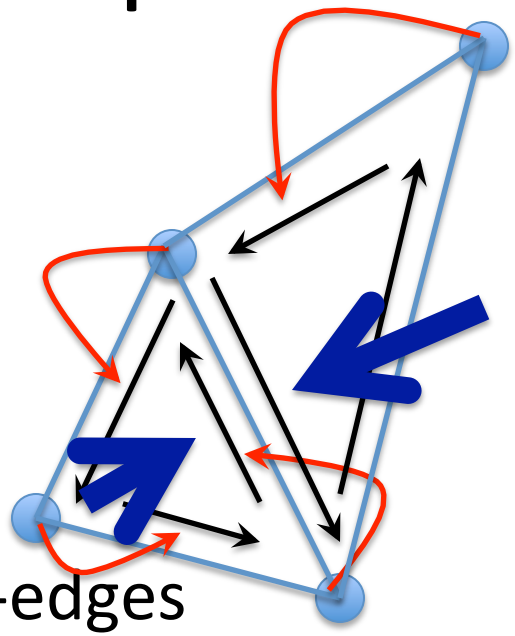


Example Shape



- *.off: vertices + triangles
- Add vertices to the list
  - only coordinates
- Add half-edges with faces
  - sufficient to add inner half-edges
  - if necessary: update vertex pointers to half edges
  - **each half-edge: pointer to ‘next’, pointer to ‘face’**
  - **face: pointer to one of the inner half-edges**

Example Shape



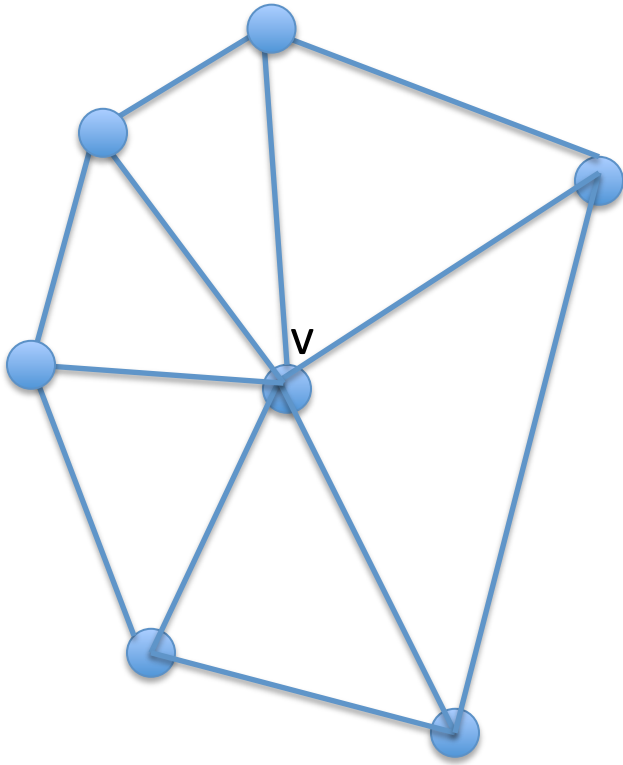
- *.off: vertices + triangles
- Add vertices to the list
  - only coordinates
- Add half-edges with faces
  - sufficient to add inner half-edges
  - if necessary: update vertex pointers to half edges
  - each half-edge: pointer to ‘next’, pointer to ‘face’
  - face: pointer to one of the inner half-edges
  - **pointer to the ‘twin’ half edge**

Mesh Processing

- **Half Edge representation**
  - Data structure
  - How to load a shape?
  - **How to find faces adjacent to a vertex?**
  - How to collapse an edge?
  - How to flip an edge?

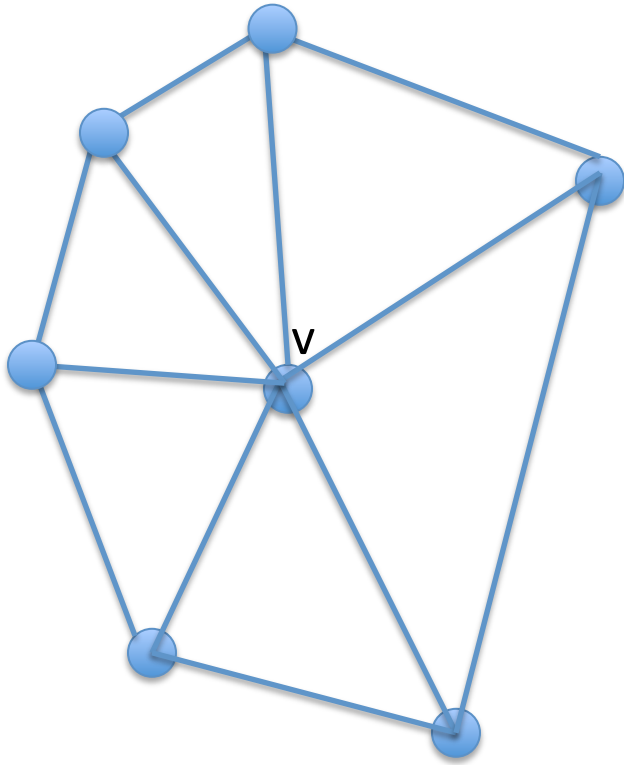
Find Adjacent Faces

?



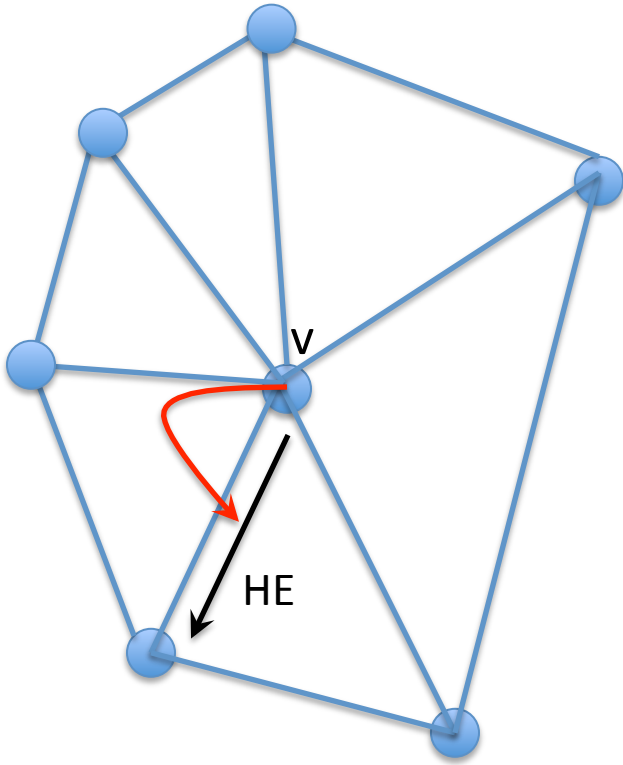
Find Adjacent Faces

- Check all outgoing half edges



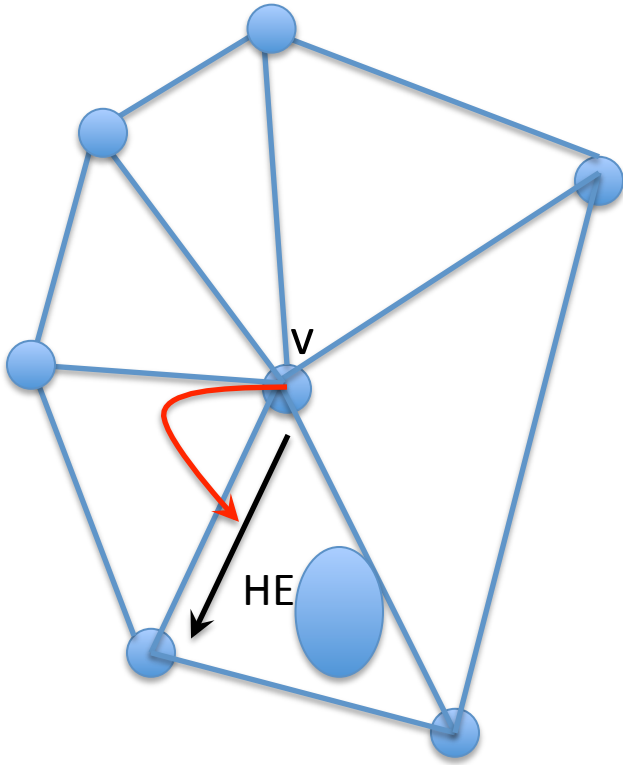
Find Adjacent Faces

- Check all outgoing half edges
  - points to a half edge HE



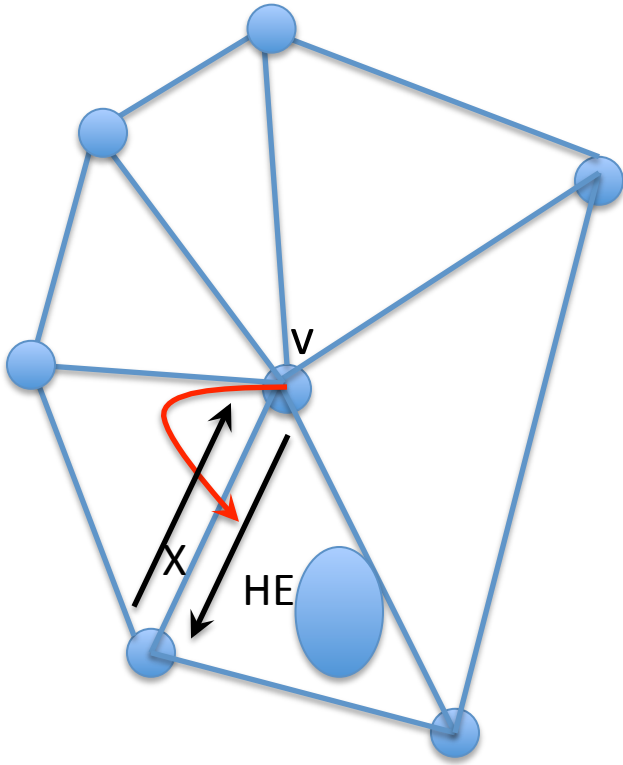
Find Adjacent Faces

- Check all outgoing half edges
  - points to a half edge HE
  - ADD_FACE(HE)



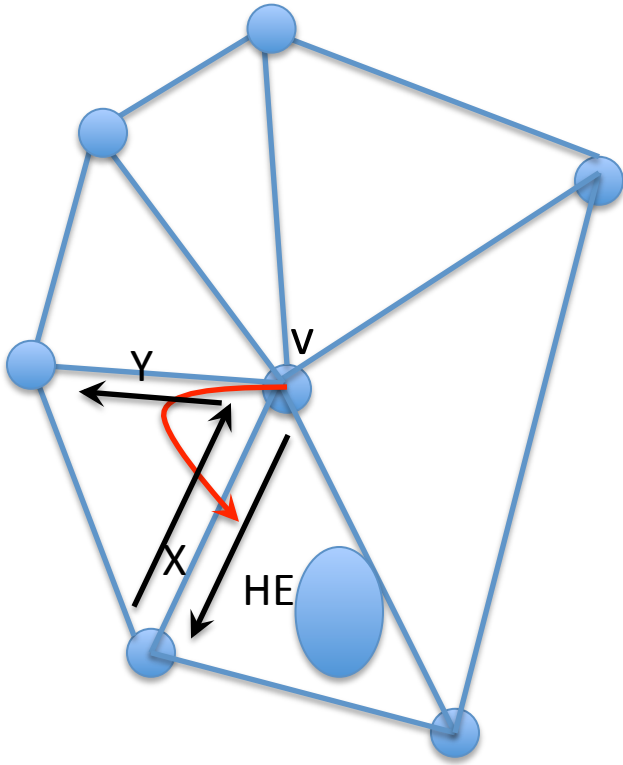
Find Adjacent Faces

- Check all outgoing half edges
  - points to a half edge HE
  - ADD_FACE(HE)
  - Iterate:
    - $X=HE_{twin}$



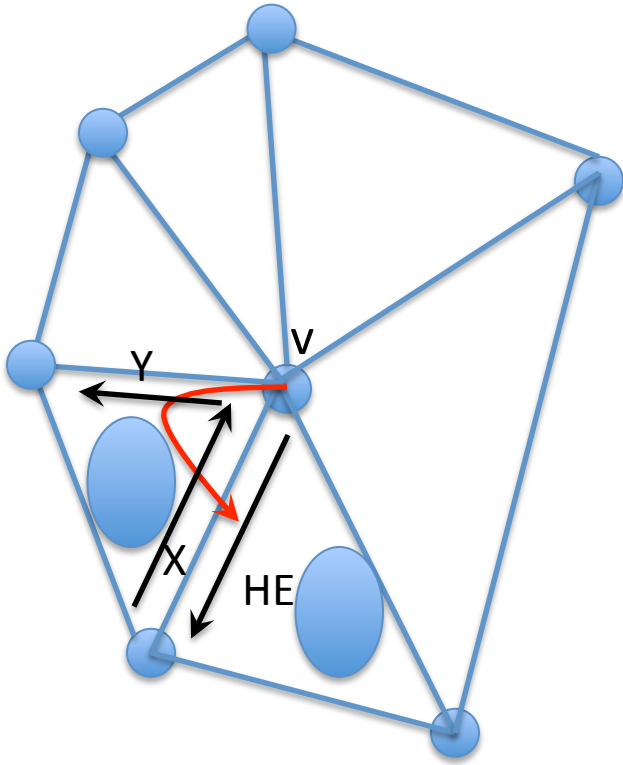
Find Adjacent Faces

- Check all outgoing half edges
  - points to a half edge HE
  - ADD_FACE(HE)
  - Iterate:
    - $X=HE_{twin}$
    - $Y=X_{next}$



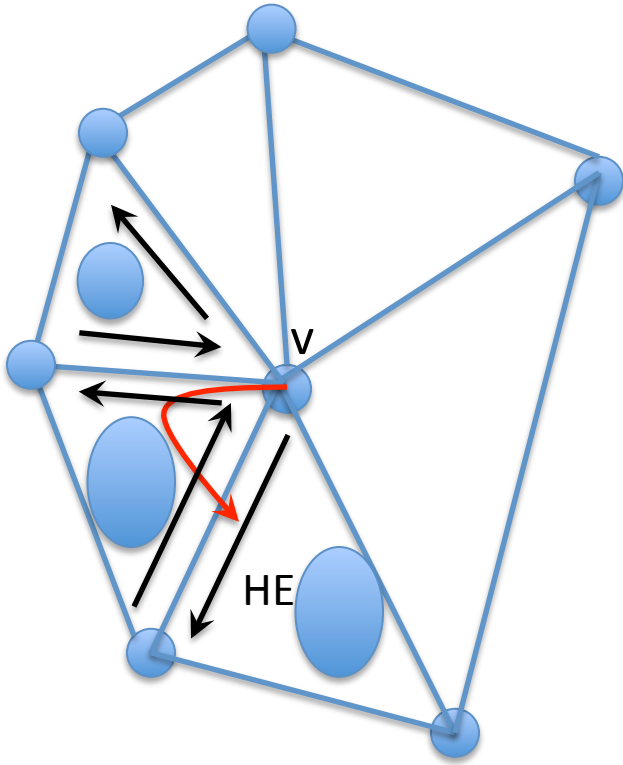
Find Adjacent Faces

- Check all outgoing half edges
  - points to a half edge HE
  - ADD_FACE(HE)
  - Iterate:
    - $X=HE_{twin}$
    - $Y=X_{next}$
    - ADD_FACE(Y)
    - HE:=Y



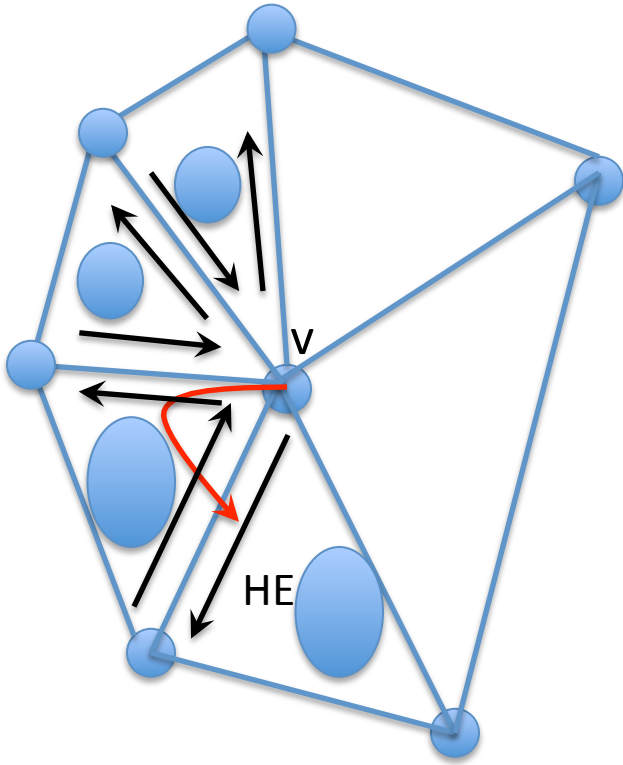
Find Adjacent Faces

- Check all outgoing half edges
  - points to a half edge HE
  - ADD_FACE(HE)
  - Iterate:
    - $X=HE_{twin}$
    - $Y=X_{next}$
    - ADD_FACE(Y)
    - HE:=Y



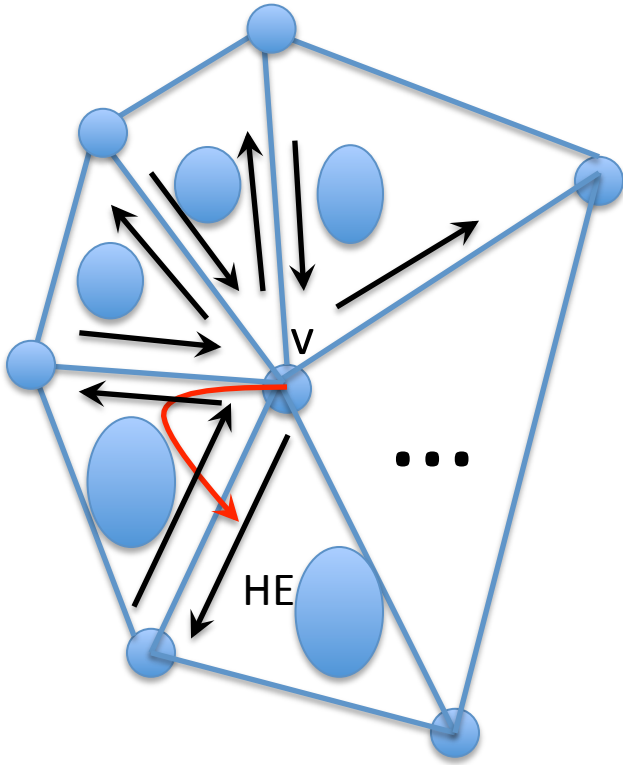
Find Adjacent Faces

- Check all outgoing half edges
  - points to a half edge HE
  - ADD_FACE(HE)
  - Iterate:
    - $X=HE_{twin}$
    - $Y=X_{next}$
    - ADD_FACE(Y)
    - HE:=Y



Find Adjacent Faces

- Check all outgoing half edges
  - points to a half edge HE
  - ADD_FACE(HE)
  - Iterate:
    - $X=HE_{twin}$
    - $Y=X_{next}$
    - ADD_FACE(Y)
    - HE:=Y

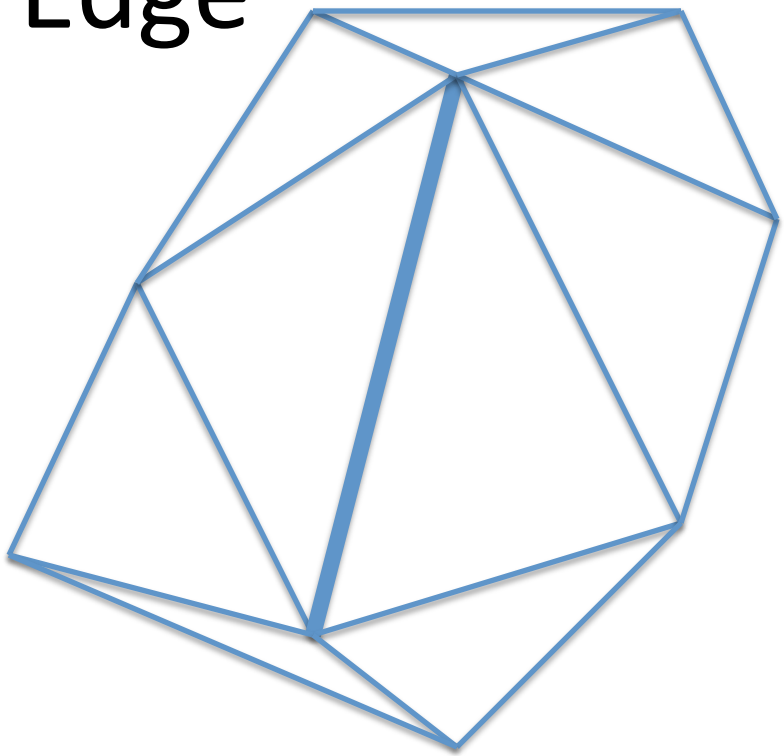


Mesh Processing

- **Half Edge representation**
  - Data structure
  - How to load a shape?
  - How to find faces adjacent to a vertex?
  - **How to collapse an edge?**
  - How to flip an edge?

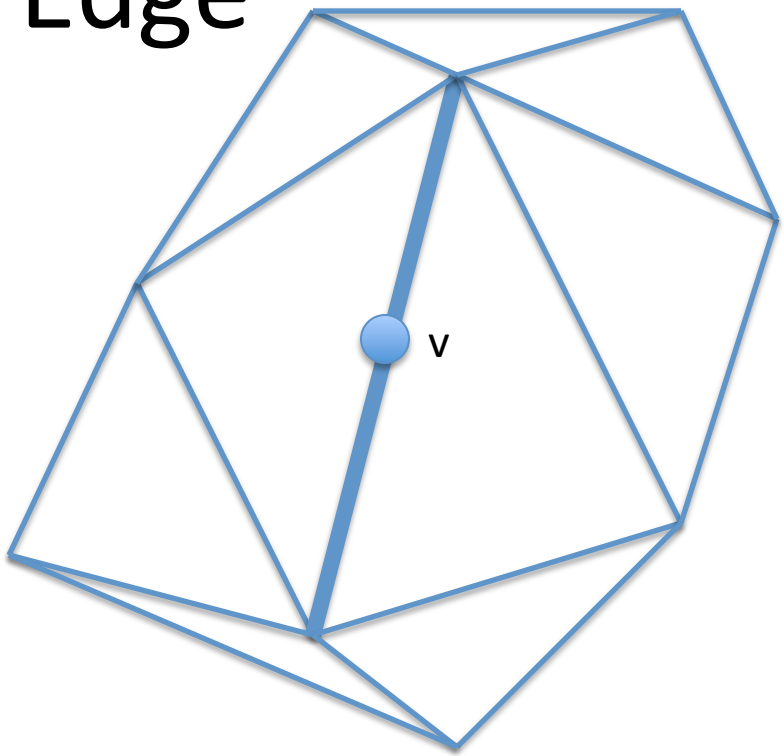
Collapse an Edge

?



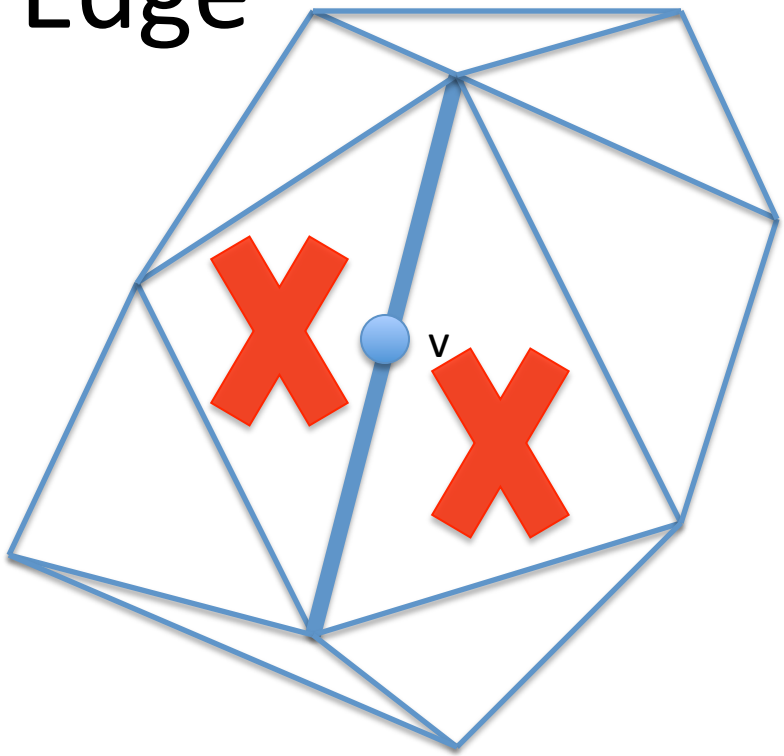
Collapse an Edge

- Create new vertex v



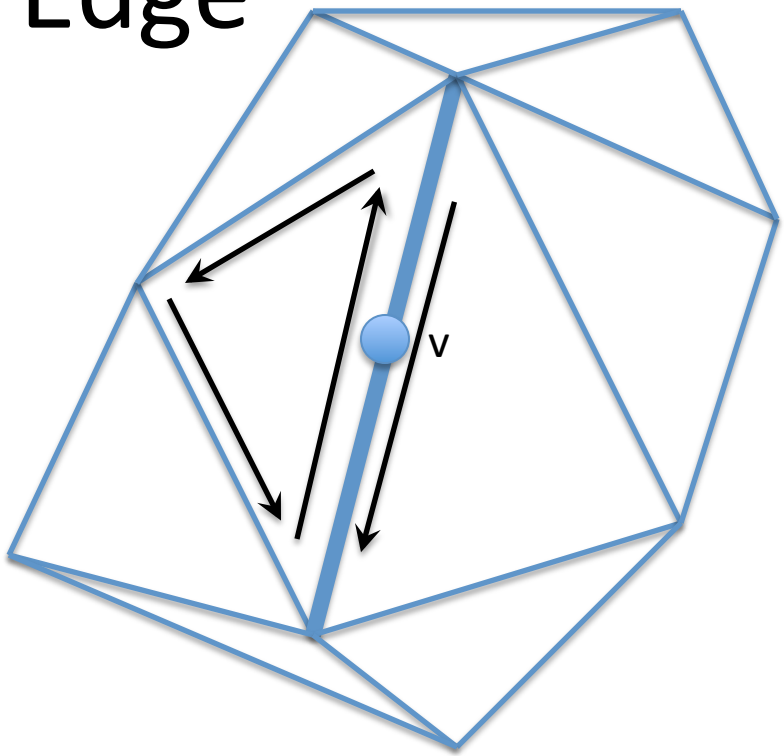
Collapse an Edge

- Create new vertex v
- Remove faces



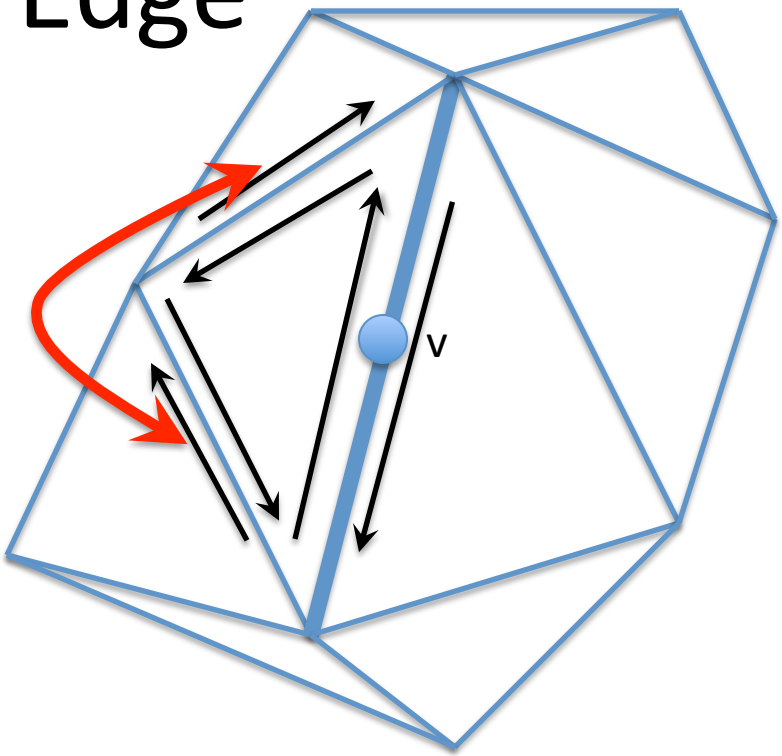
Collapse an Edge

- Create new vertex v
- Remove faces
- Change ‘twin’ pointers



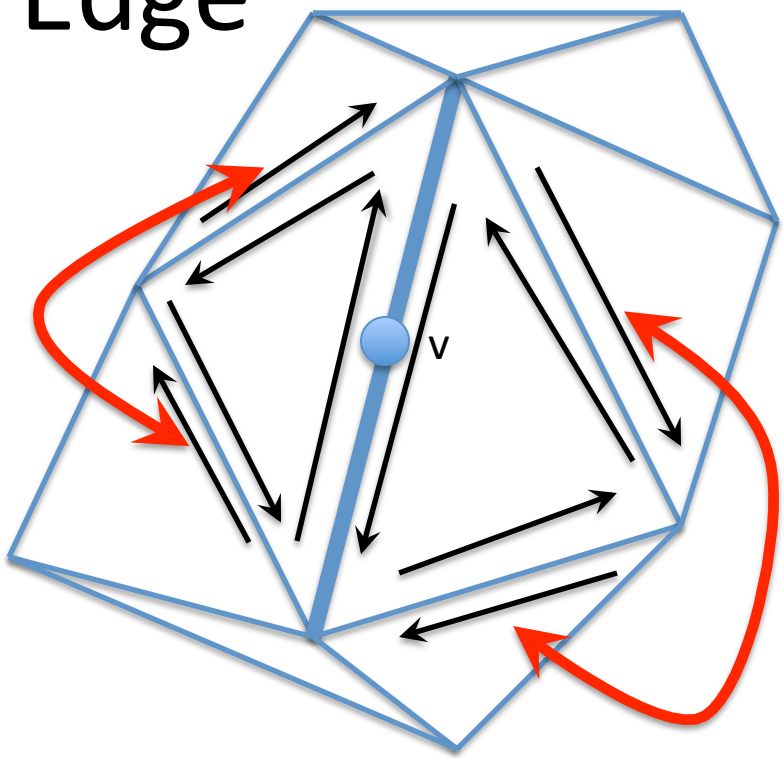
Collapse an Edge

- Create new vertex v
- Remove faces
- Change ‘twin’ pointers



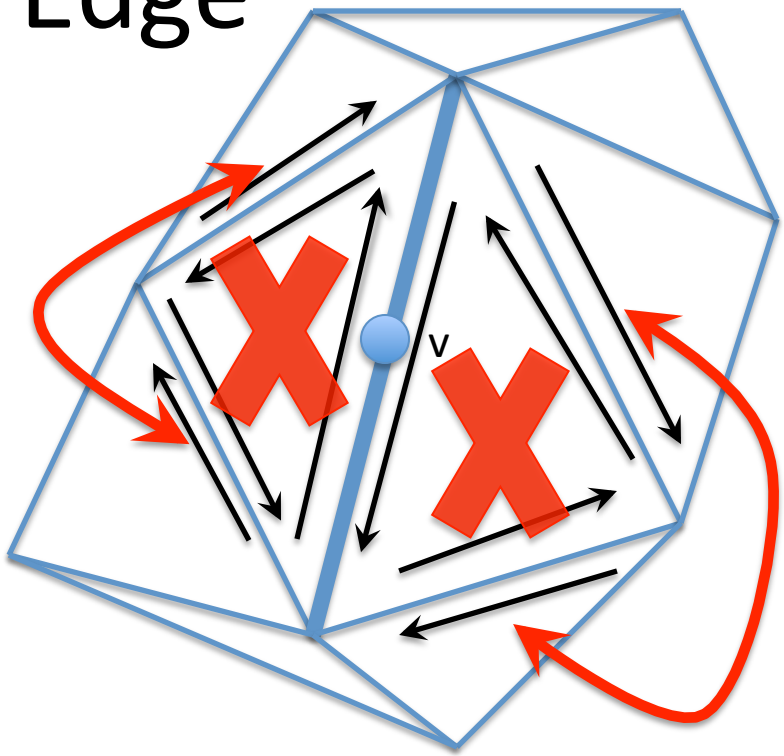
Collapse an Edge

- Create new vertex v
- Remove faces
- Change ‘twin’ pointers



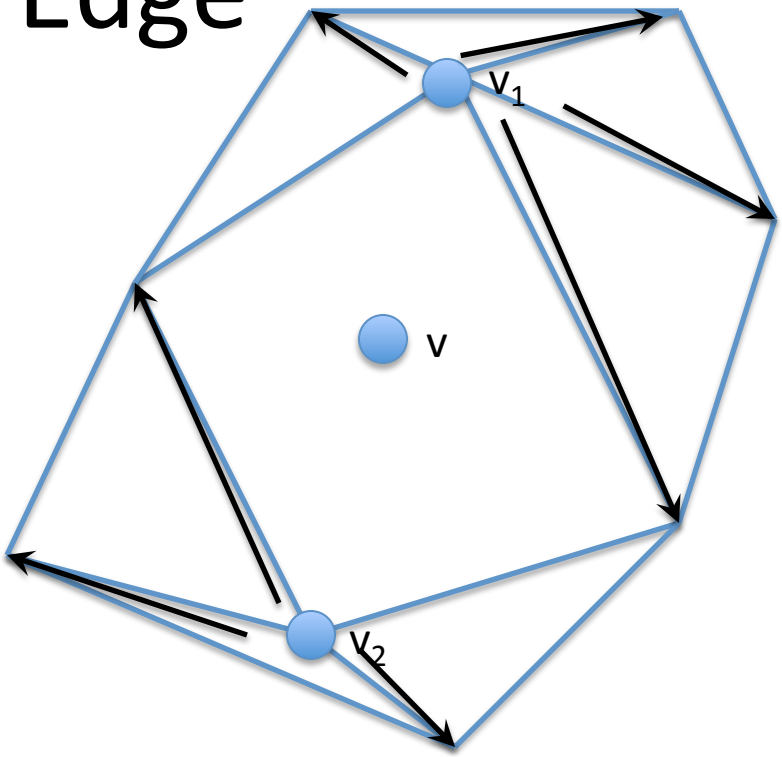
Collapse an Edge

- Create new vertex v
- Remove faces
- Change ‘twin’ pointers
- Remove edges



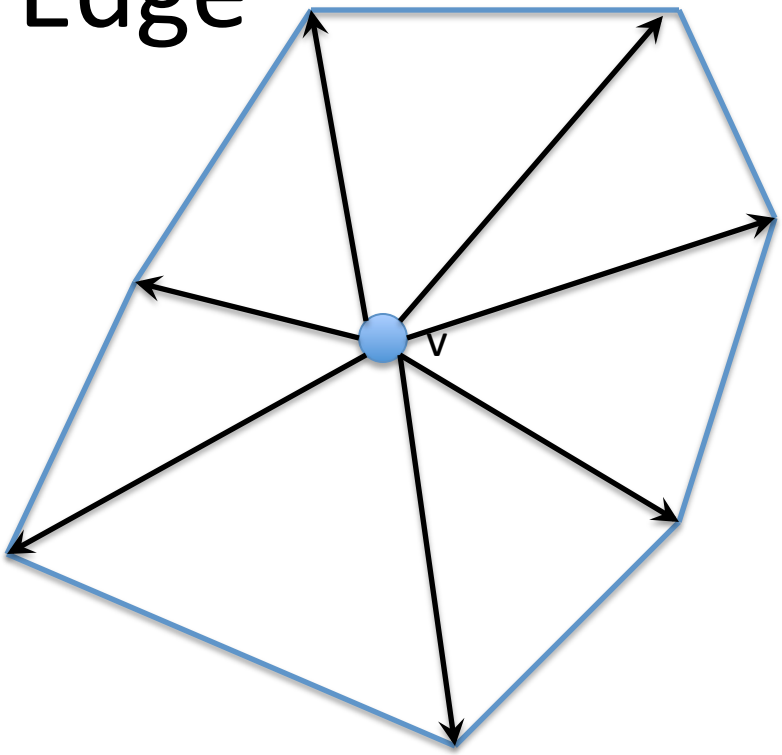
Collapse an Edge

- Create new vertex v
- Remove faces
- Change ‘twin’ pointers
- Remove edges
- Change pointers to v_1, v_2
  - check outgoing edges



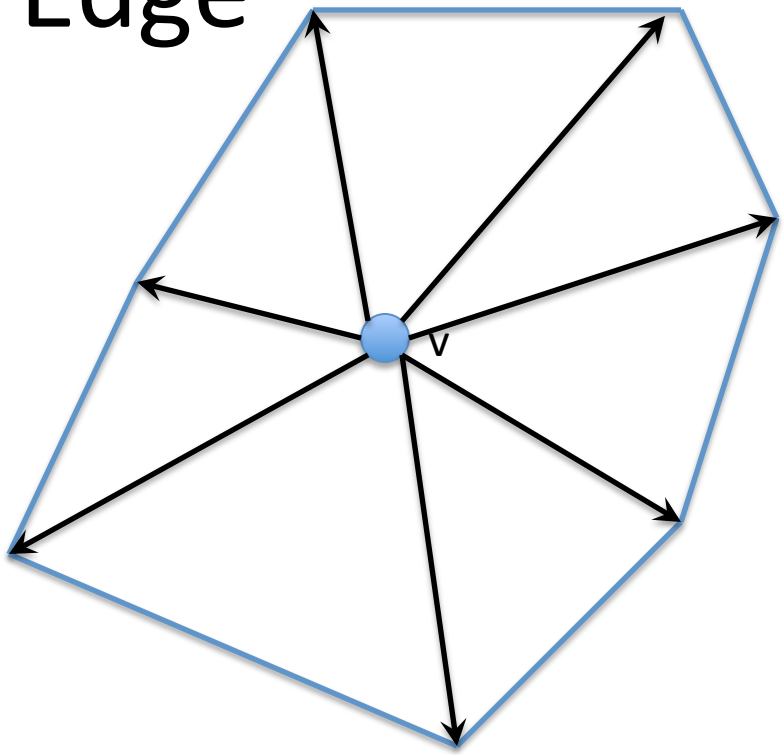
Collapse an Edge

- Create new vertex v
- Remove faces
- Change ‘twin’ pointers
- Remove edges
- Change pointers to v_1, v_2
- Remove v_1, v_2



Collapse an Edge

- Create new vertex v
- Remove faces
- Change ‘twin’ pointers
- Remove edges
- Change pointers to v_1, v_2
- Remove v_1, v_2
- Pick an outgoing edge for v

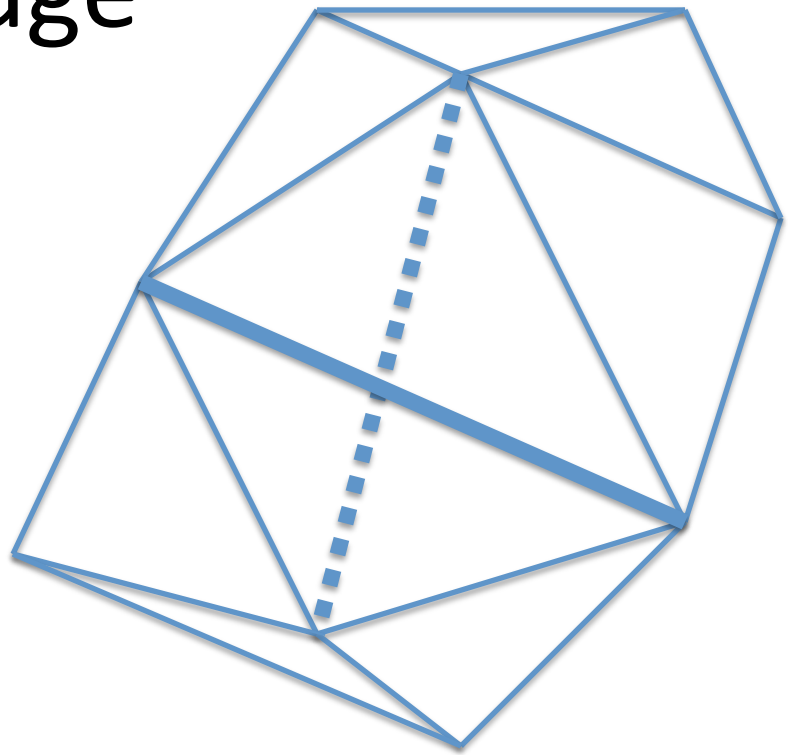


Mesh Processing

- **Half Edge representation**
  - Data structure
  - How to load a shape?
  - How to find faces adjacent to a vertex?
  - How to collapse an edge?
  - **How to flip an edge?**

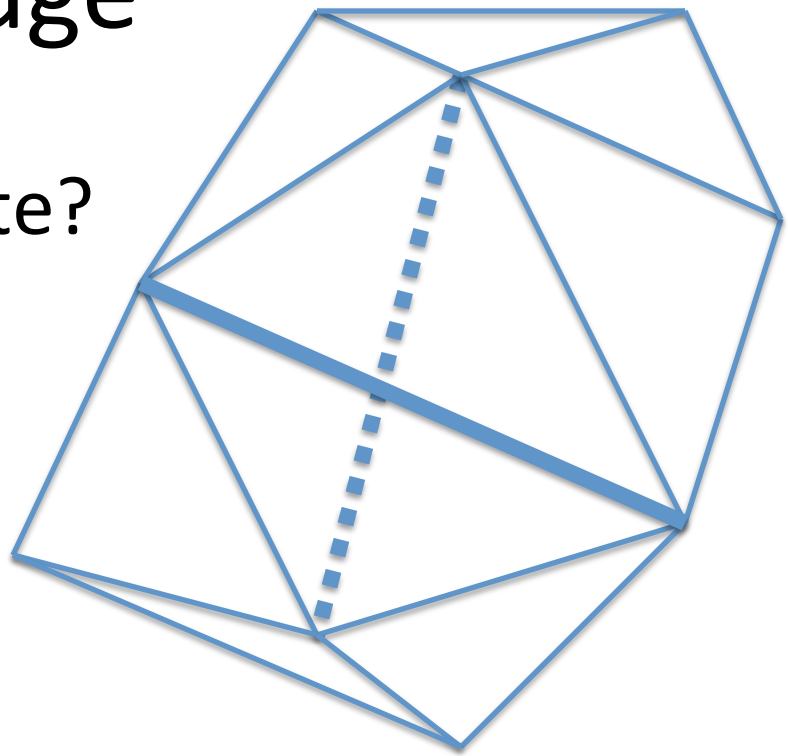
Flip an Edge

?



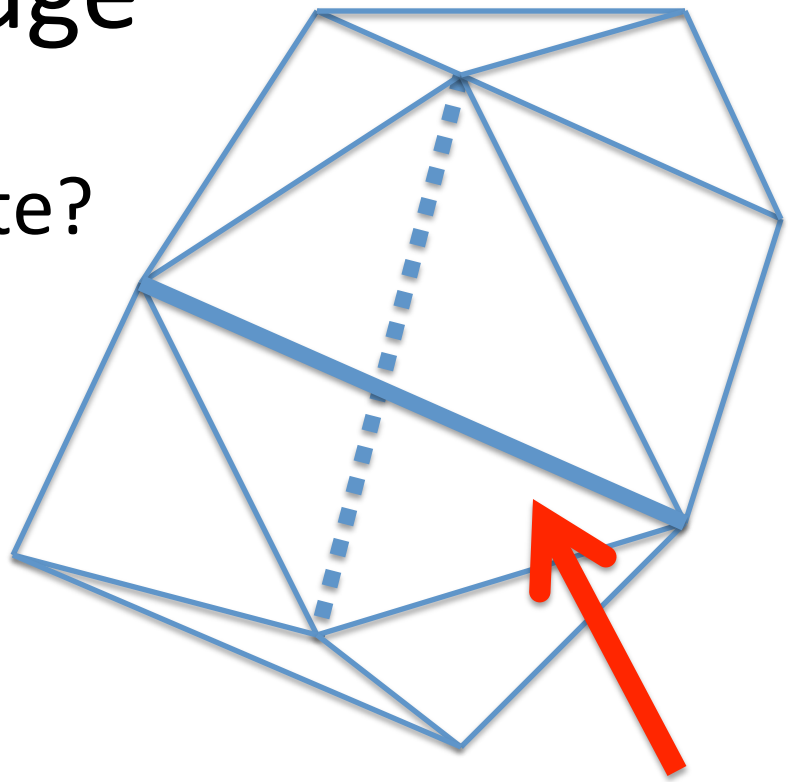
Flip an Edge

- What do we need to update?



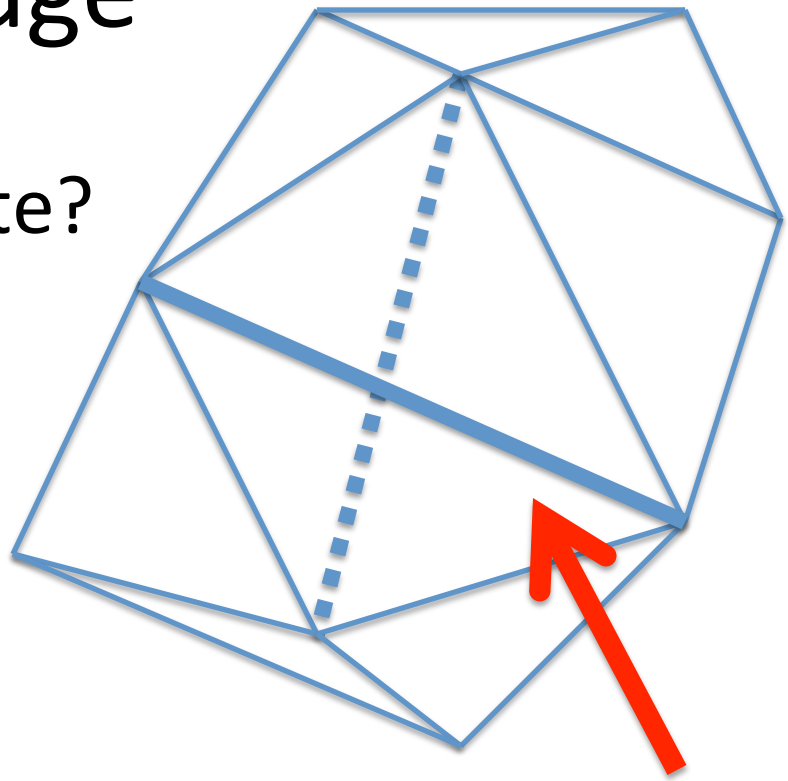
Flip an Edge

- What do we need to update?
  - Half-edges on the edge



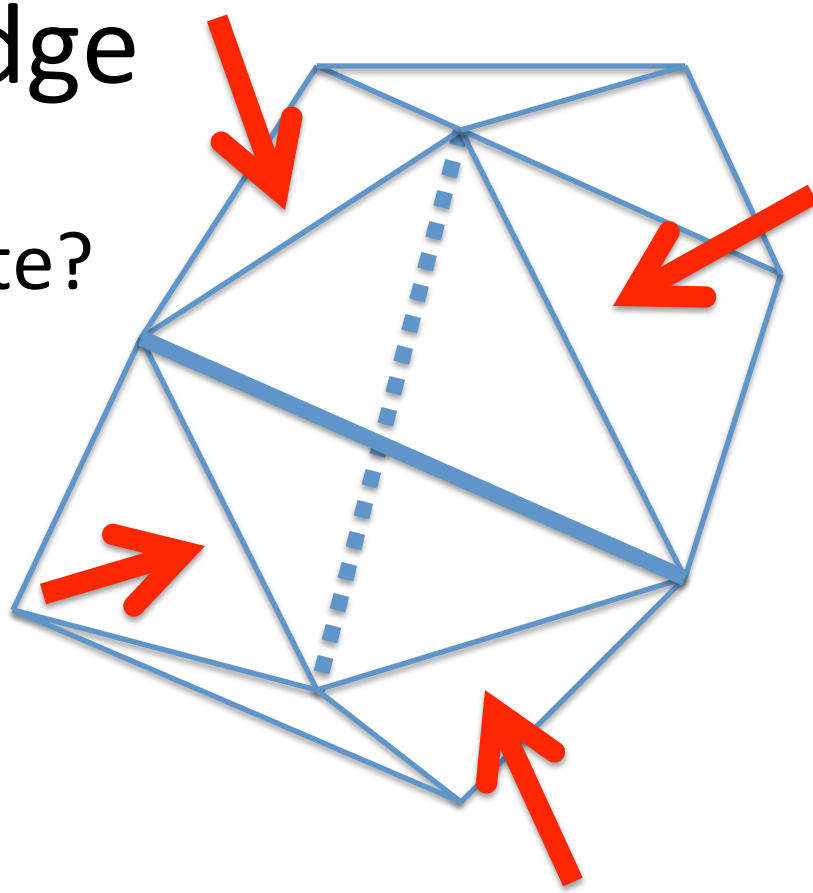
Flip an Edge

- What do we need to update?
  - Half-edges on the edge
    - vertex, next



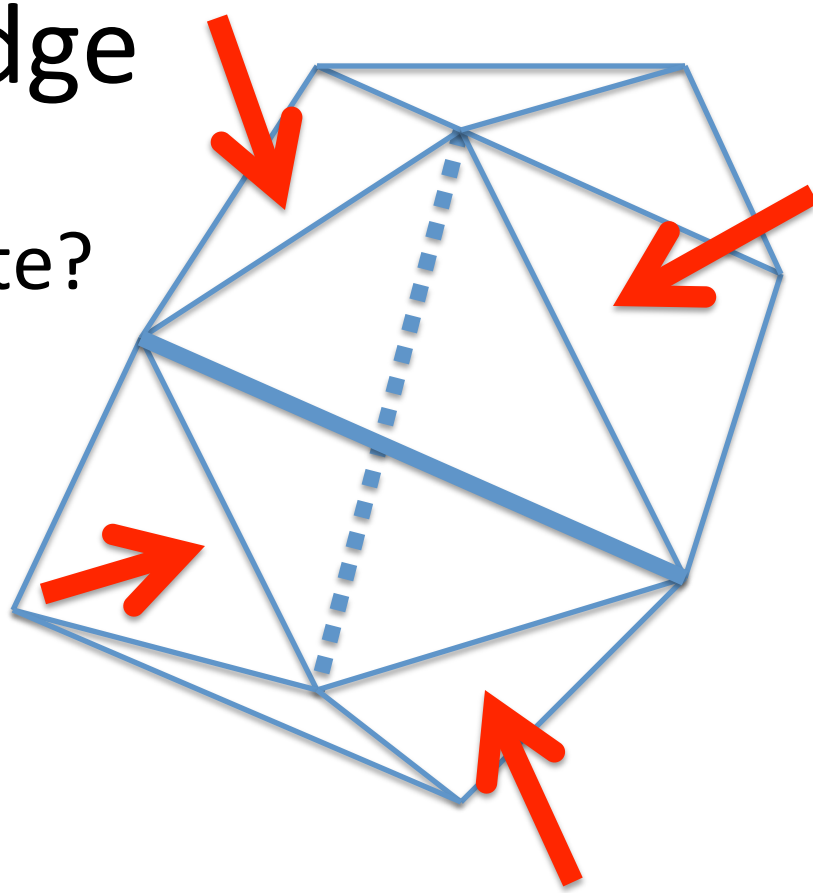
Flip an Edge

- What do we need to update?
  - Half-edges on the edge
    - vertex, next
  - Adjacent half-edges



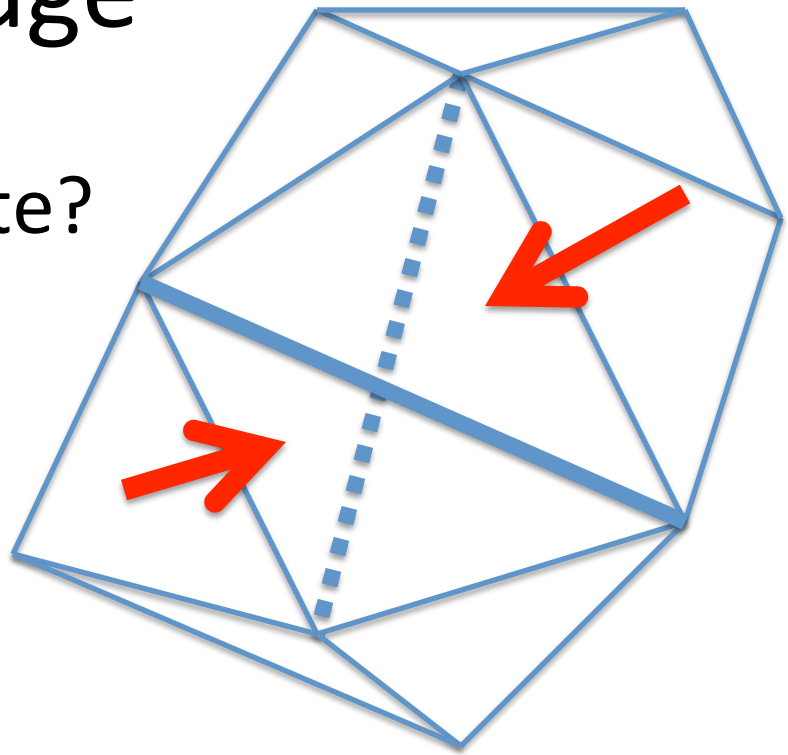
Flip an Edge

- What do we need to update?
  - Half-edges on the edge
    - vertex, next
  - Adjacent half-edges
    - next



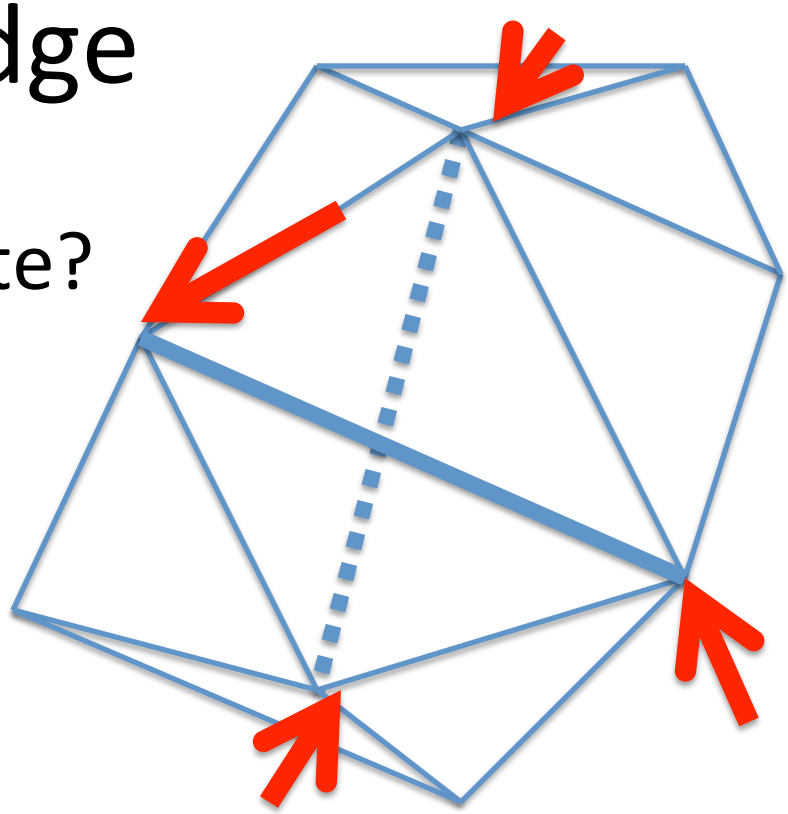
Flip an Edge

- What do we need to update?
  - Half-edges on the edge
    - vertex, next
  - Adjacent half-edges
    - next
  - Faces



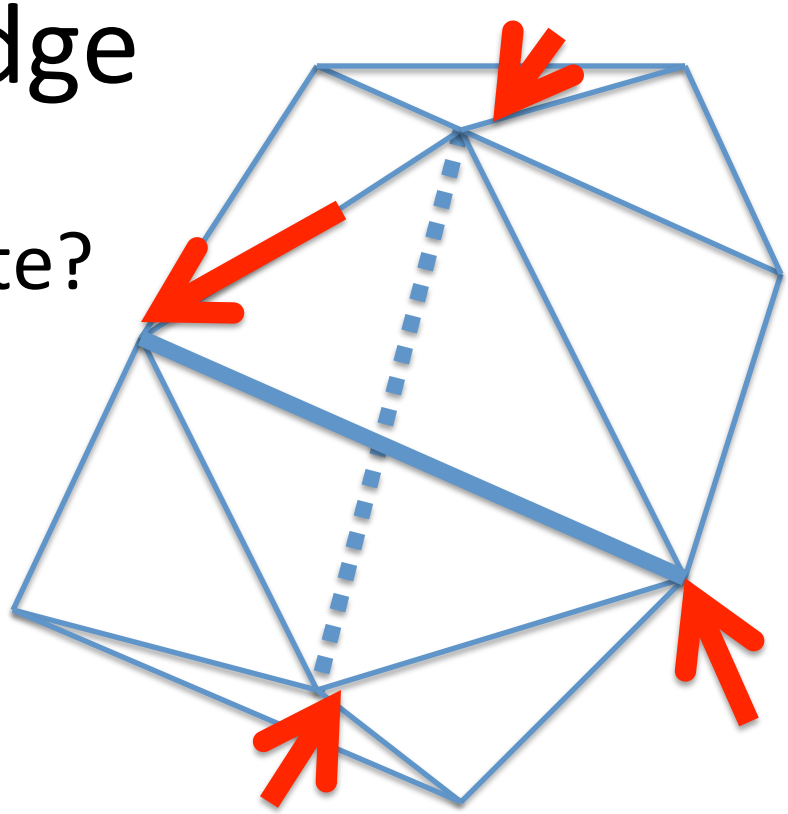
Flip an Edge

- What do we need to update?
  - Half-edges on the edge
    - vertex, next
  - Adjacent half-edges
    - next
  - Faces
  - Vertices

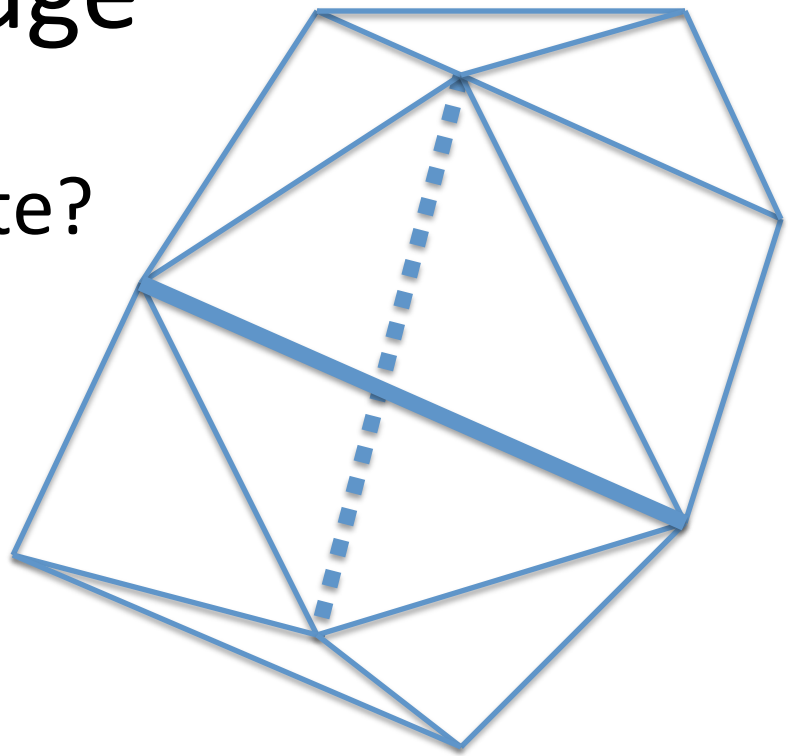


Flip an Edge

- What do we need to update?
  - Half-edges on the edge
    - vertex, next
  - Adjacent half-edges
    - next
  - Faces
  - Vertices
    - possibly ‘outgoing edge’



Flip an Edge



- What do we need to update?
  - Half-edges on the edge
    - vertex, next
  - Adjacent half-edges
    - next
  - Faces
  - Vertices
    - possibly ‘outgoing edge’
  - Problems? Can we always flip edges?