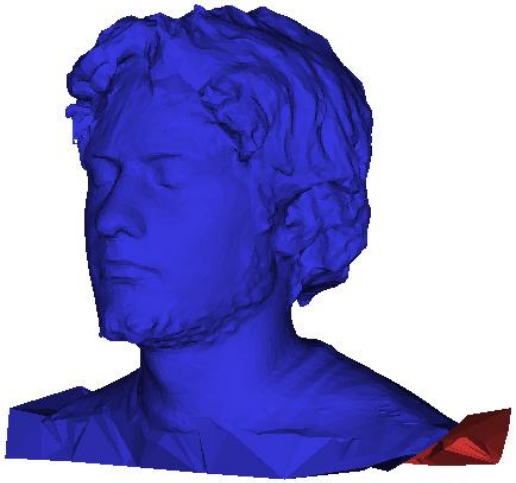
Assignment 5

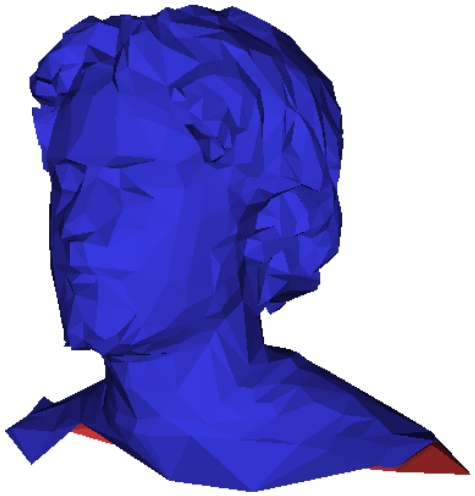
Qslim: Quadric Error Decimation Algorithm

QSlim

100‘000 vertices



10‘000 vertices



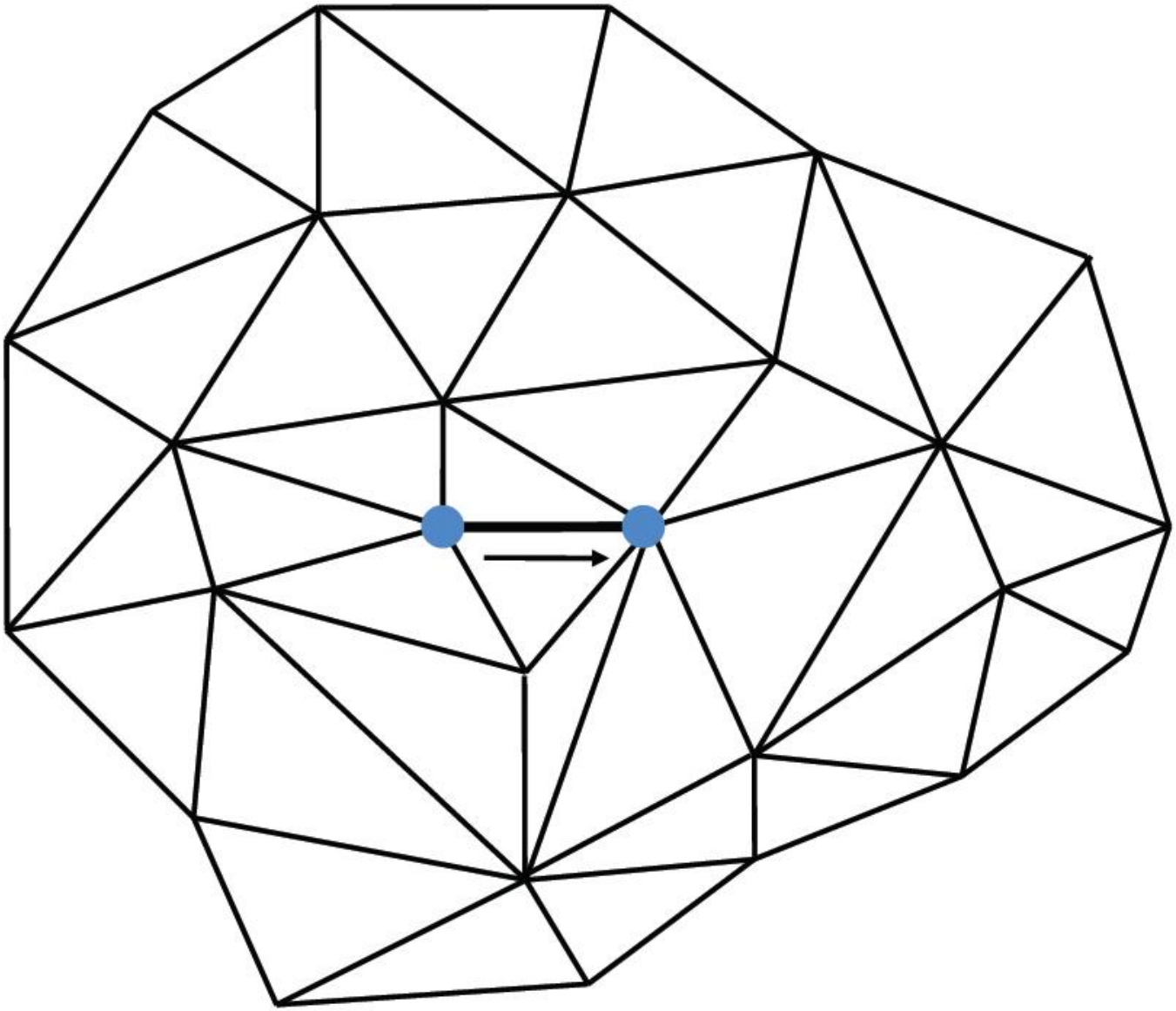
1‘000 vertices

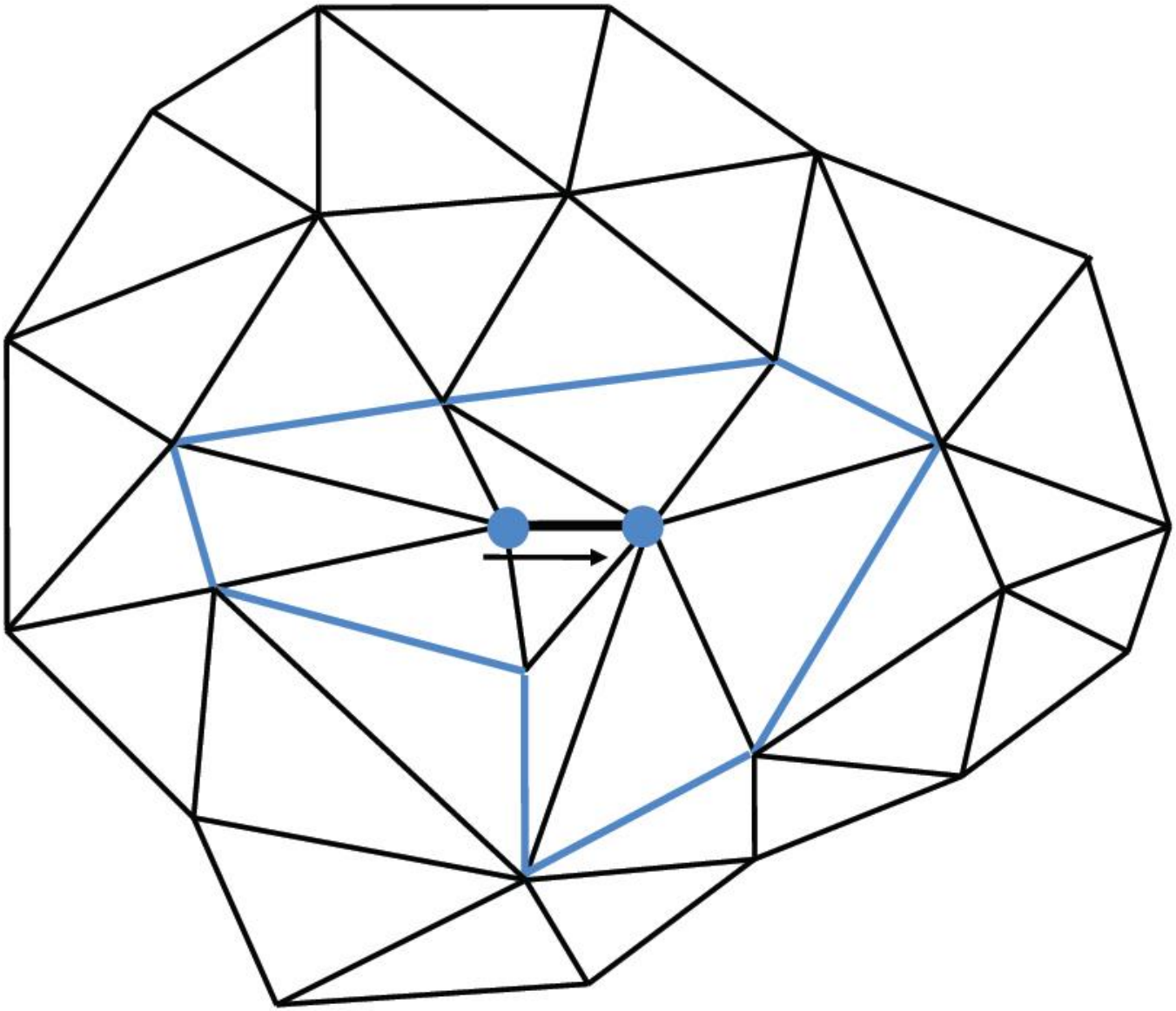
QSlim

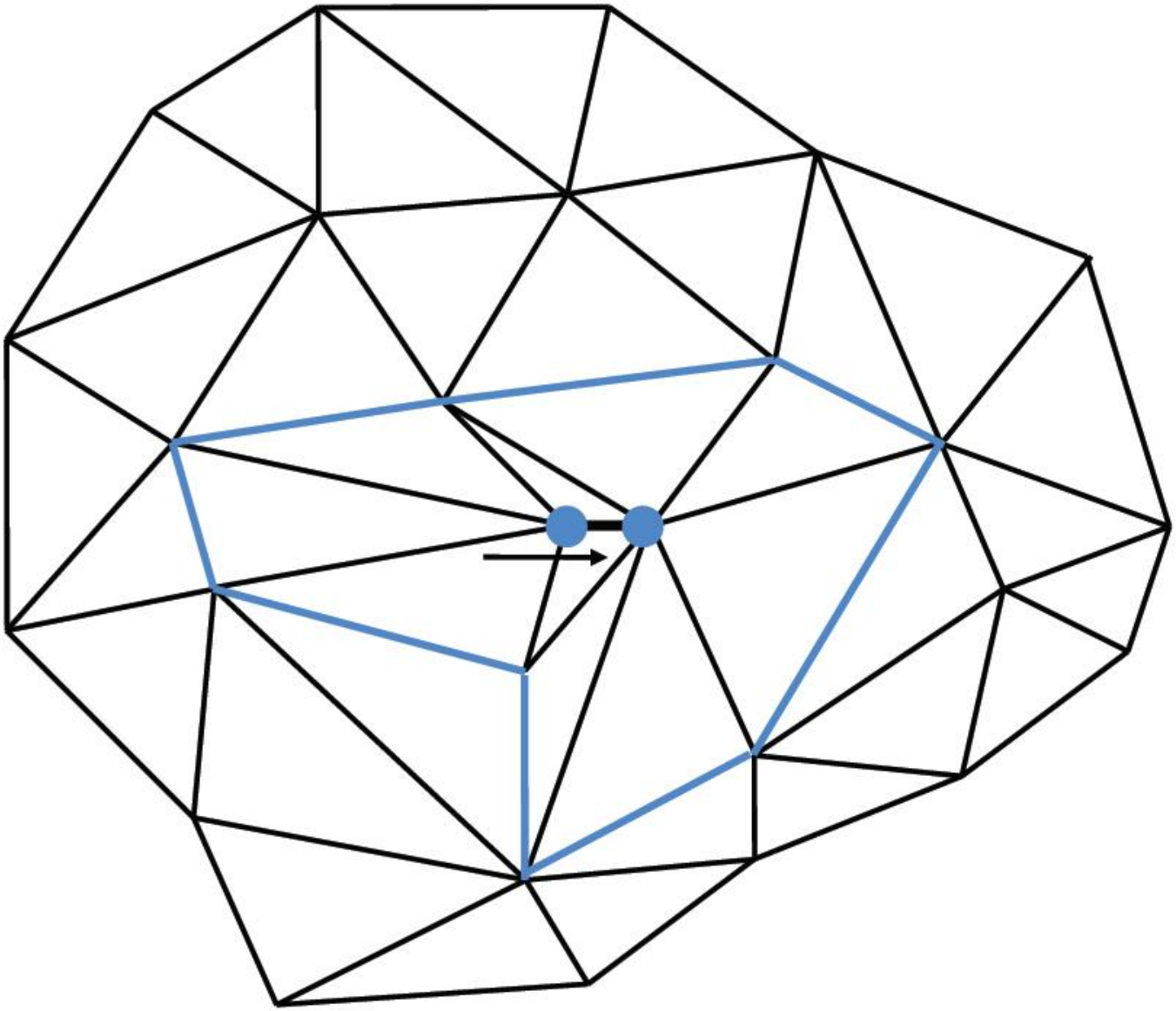
- Greedily Collapse Edges
- Heuristically Prioritize Edges

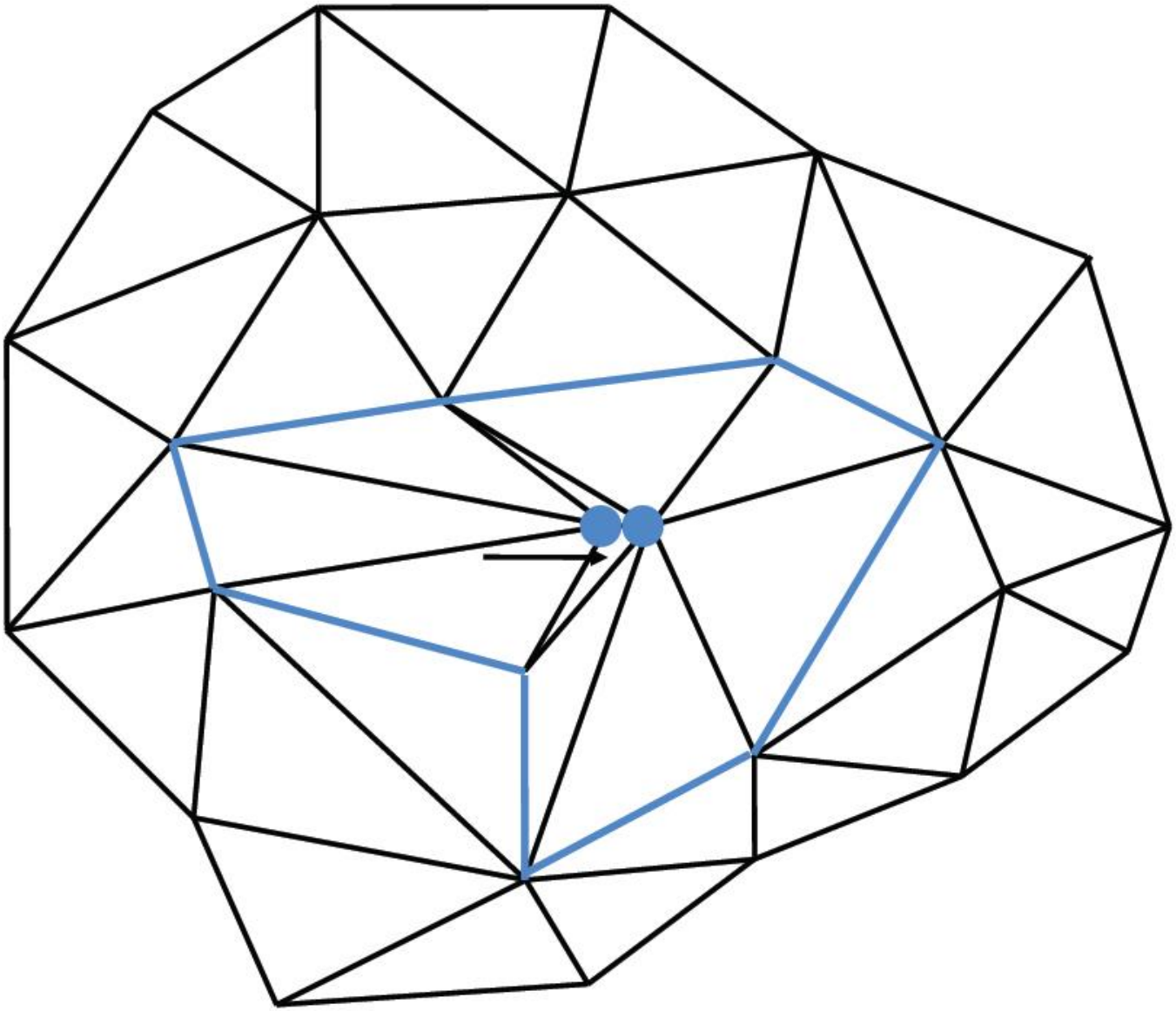
- No large matrices involved :-)

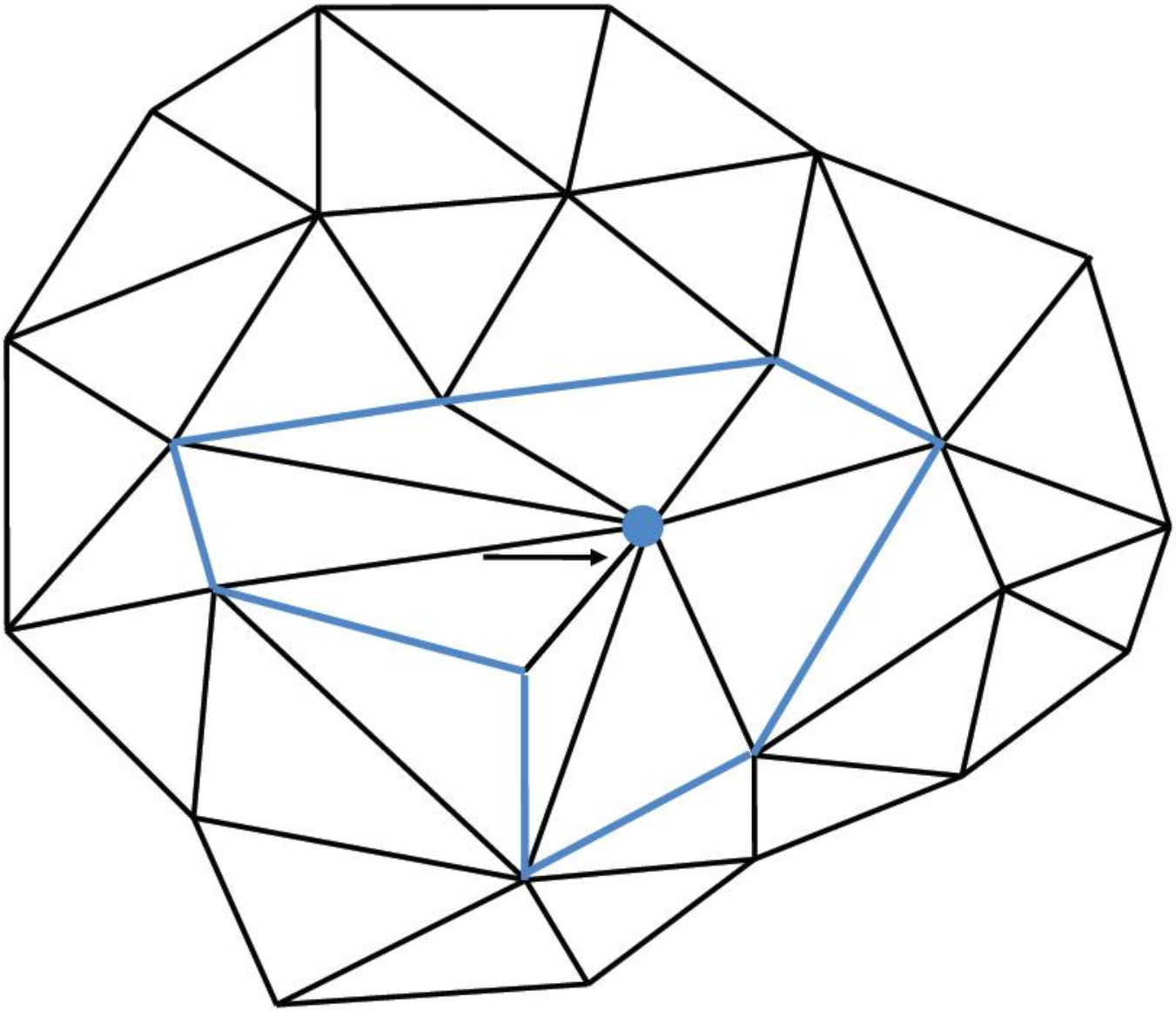
Edge Collapse



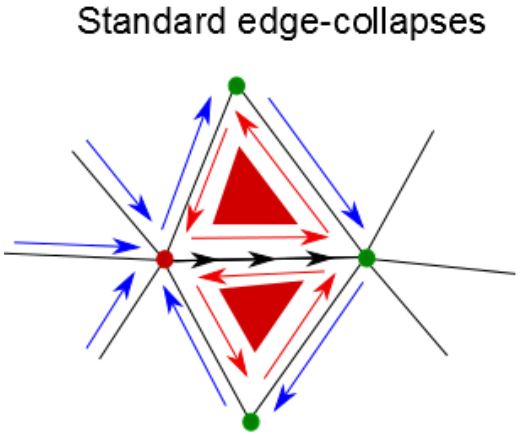




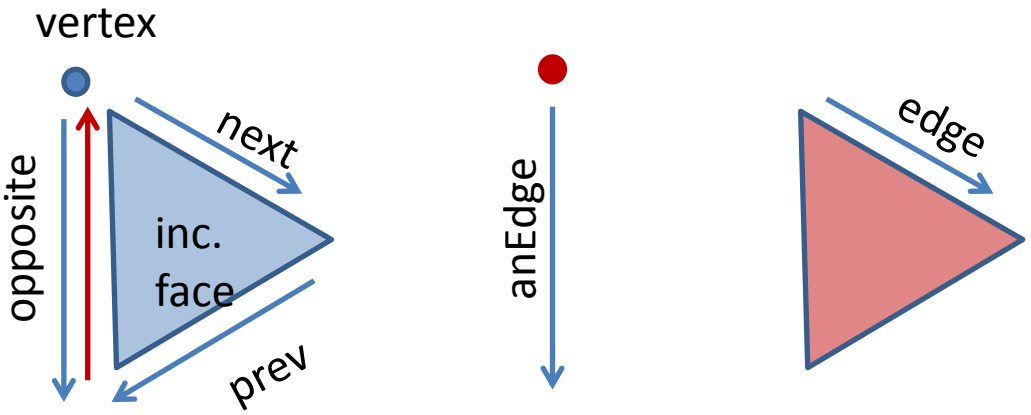




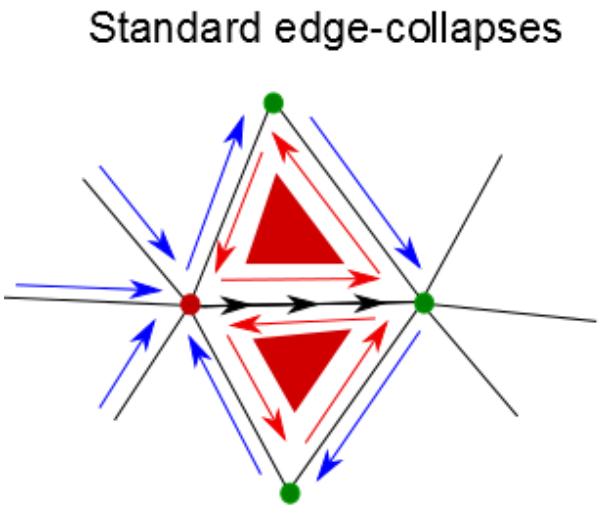
Edge Collapse



- Challenge 1: Relink everything to a valid half-edge structure.

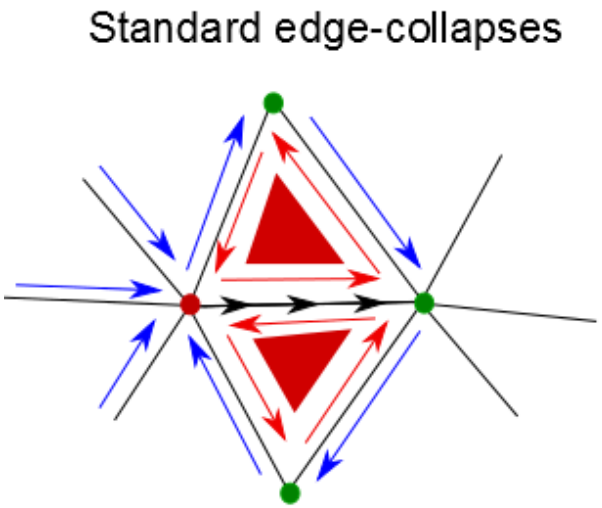


Edge Collapse



- Elements to remove
- Vertices that might reference obsolete edges
- Edges which have to be relinked

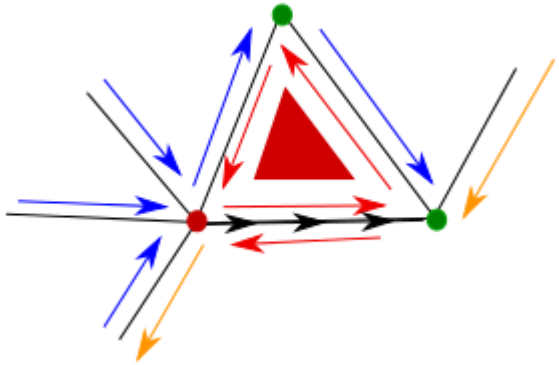
Edge Collapse



- Algorithm:
  1. Assign safe edges to the green vertices (method provided)
  2. Iterate around e.start() and assign the new end-vertex to the edges
  3. Relink e.next().opposite = e.prev.opposite, etc.
  4. Remove the obsolete elements/tag them as removed.

Edge Collapse (Boundaries)

- Relinking step is different!



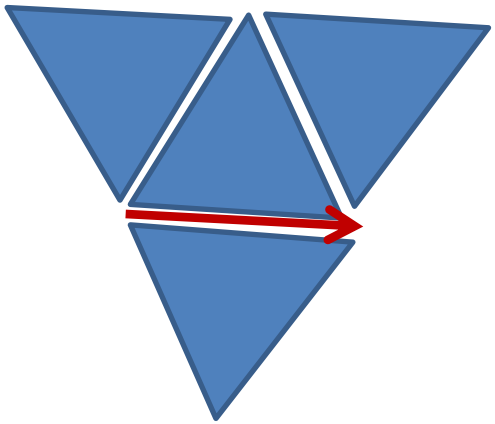
- In practice:
  - if(e.hasFace){...do standard case...} else{...do boundary case...}
  - if(e.getOpposite().hasFace){...do standard case...} else{...do boundary case...}

Edge Collapse

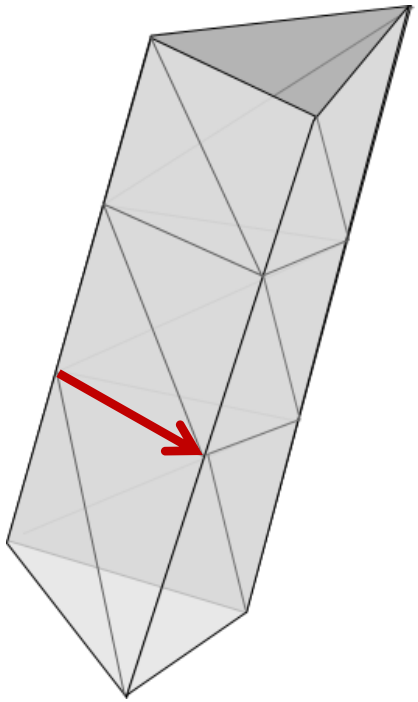
- Challenge 1: Relink everything ✓
- Challenge 2: Is edge collapsable, such that
  - mesh can still be represented as half-edge structure
  - Mesh has same topology (number of holes/components)

Illegal collapses

- Can happen on boundaries or inside the mesh



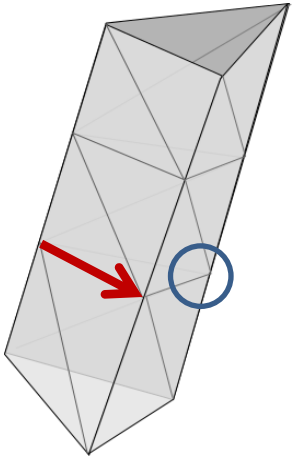
Edge will dangle!

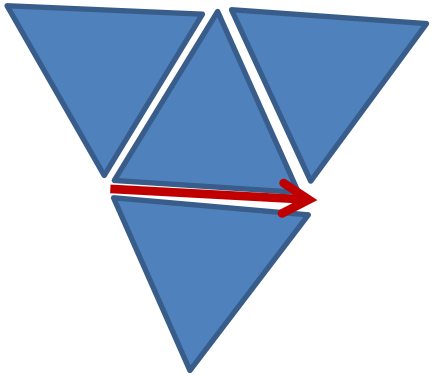


Remaining edge will
have 4 incident faces!

Illegal collapses

- Test:





- All common neighbor vertices e.start & e.end share a face with e.
- & Boundary specific special cases

Edge Collapse

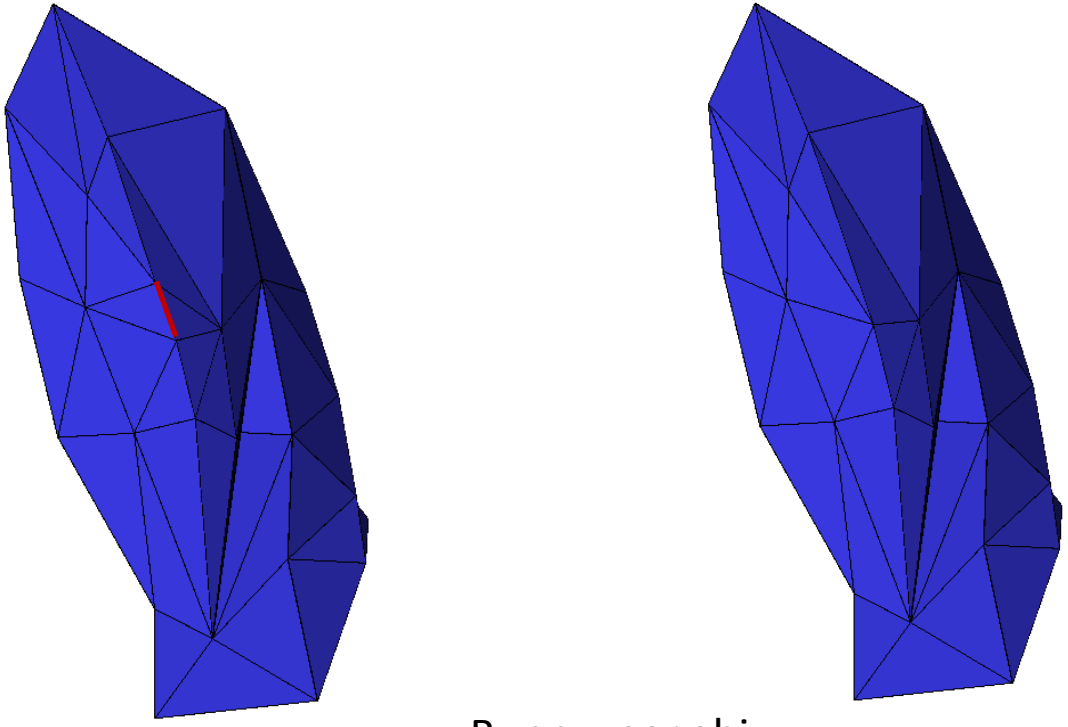
- Challenge 1: Relink everything ✓
- Challenge 2: Is edge collapsable?
  - Method provided ✓
- Challenge 3: Fold-overs?
  - Heuristic method provided.
  - Test if the normal after the collapse is too different from the original normal.
  - Can fail

Edge Collapse

- Challenge 1: Relink everything ✓
- Challenge 2: Is edge collapsable?
  - Method provided ✓
- Challenge 3: Fold-overs?
  - Heuristic method provided. ✓
  - Test if the normal after the collapse is too different from the original normal.
  - Can fail

QSlim

- Idea: iteratively collapse the edge with the least visual impact.



Bunny_ear.obj

Impact Heuristic

- Exercise Sheet.
  - One matrix per vertex, encoding a quadratic error form

$$cost(p) = p^T Q_v p$$

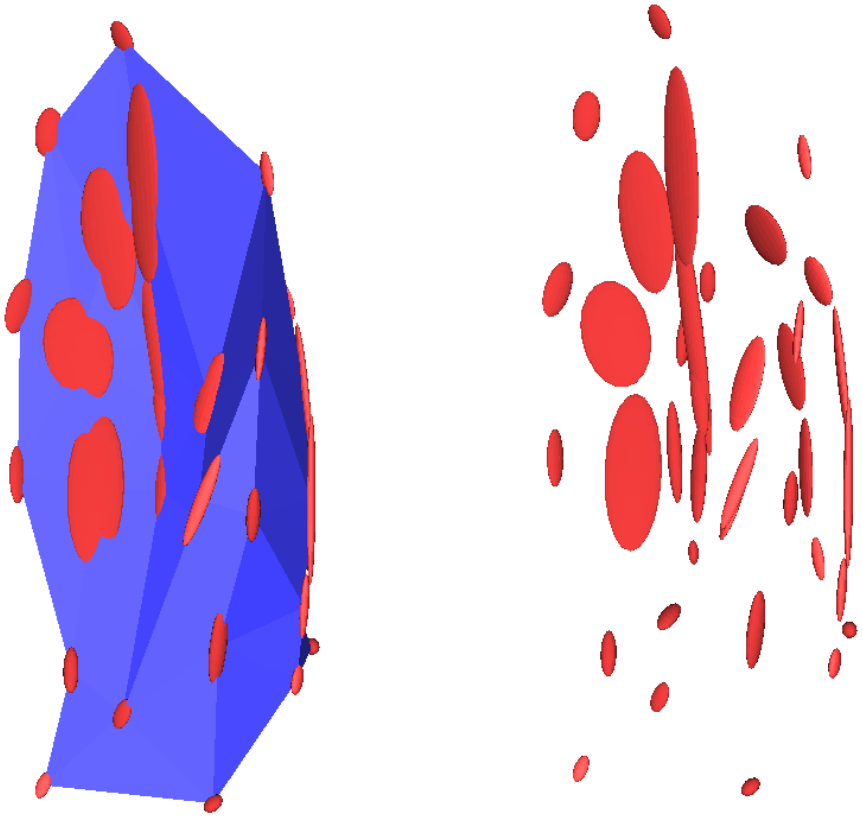
  - Cost of collapse (u->v), assigning p as a new position:

$$cost((u \rightarrow v), p) = p^T (Q_u + Q_v) p$$

  - Cost matrix of collapsed vertex:

$$(Q_u + Q_v)$$

Impact Heuristic



Qslim Algorithm

Setup Phase:

compute all error matrices, all edge collapse costs

Main Loop:

1. Select cheapest edge
2. Is Edge collapsable?
    1. No -> increment costs of the edge, select next edge
3. Collapse Edge
4. Compute Matrix for the new vertex
5. Update collapse costs of adjacent edges

http://www1.cs.columbia.edu/~cs4162/html05s/garland97.pdf

Efficient Prioritizing

- Use a Java PriorityQueue<>
  - Problem: does not allow efficient removal of elements or cost updates!

Efficient Priority updates

- Keep additional hashmap
  - Reference to undeleted collapse
- Add Flag: isDeleted

Additional
Hashmap<Edge, PotentialCollapse>

edge	Potential collapse
e	

Priority Queue

	Collapse 1	Collapse 2	...	Collapse j		
Flag: Deleted ?	no	yes	...	no		

Efficient Priority Updates

- Example: updates cost of collapse e:
  1. Retrieve current collapse of e

edge	Potential collapse
e	

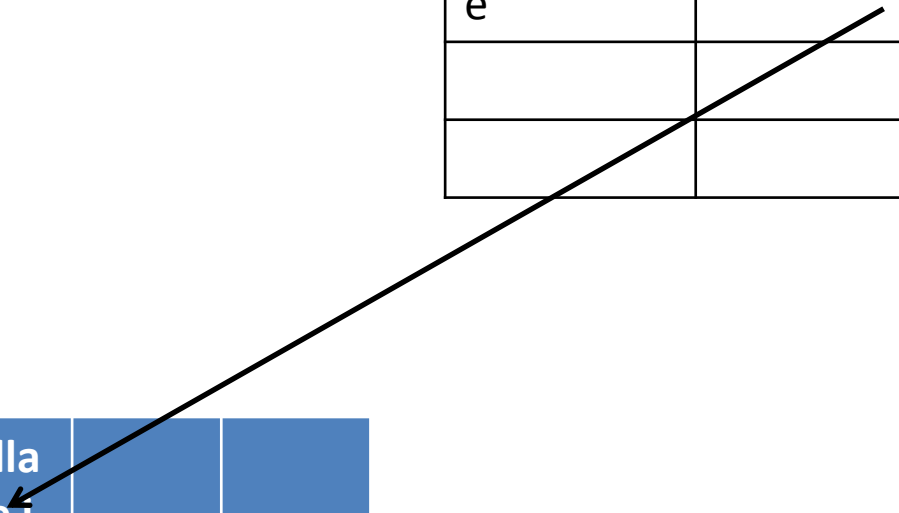
Priority Queue

	Collapse 1	Collapse 2	...	Collapse j		
Flag: Deleted ?	no	yes	...	no		

Efficient Priority Updates

- Example: updates cost of collapse e:
  1. Retrieve current collapse
  2. Mark as deleted

edge	Potential collapse
e	



Priority Queue

	Collapse 1	Collapse 2	...	Collapse j		
Flag: Deleted ?	no	yes	...	no		

Efficient Priority Updates

- Example: updates cost of collapse e:
  1. Retrieve current collapse
  2. Mark as deleted
  3. Add new collapse
  4. Update hashmap

edge	Potential collapse
e	

Priority Queue

	Collapse 1	Collapse 2	…	Collapse j		Collapse j
Flag: Deleted ?	no	yes	…	yes		no

Questions?

- Base code will be online at about 18:00.