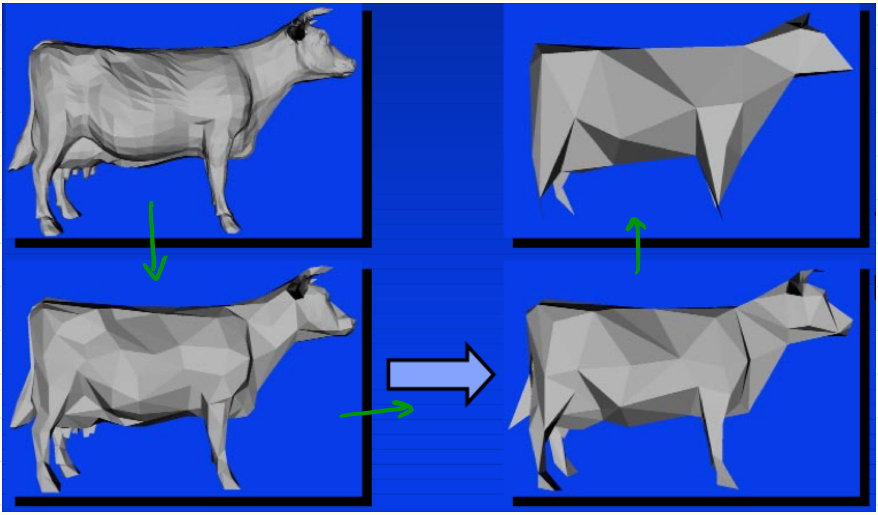
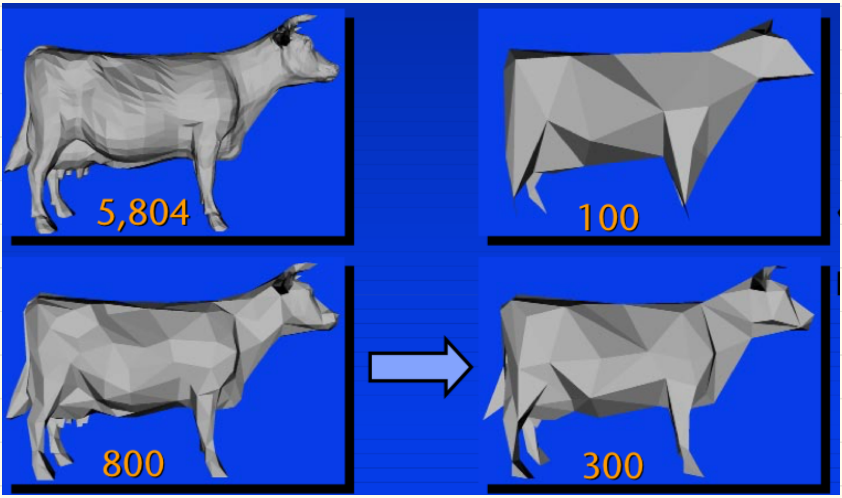
LECTURE 4: SIMPLIFICATION.





How to 'simplify' a mesh?

- design a basic simplication step
- use this step iteratively on the mesh.

COMPLEXITY OF A MESH

- faces
- edges
- vertices.

all related to each other <u>via</u>

EULER'S THEOREM

next year.

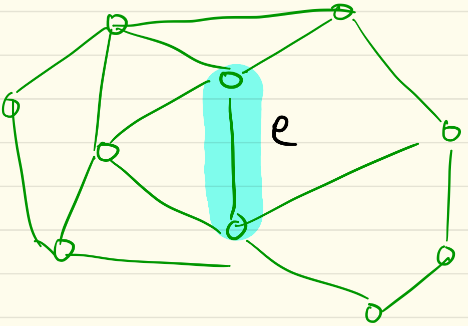
<u>1</u> Remove vertices

VERTEX DECIMATION

<u>2</u> Remove edges

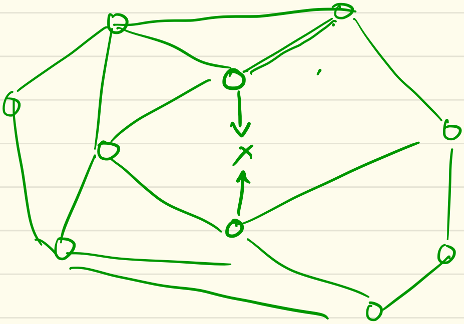
TODAY.

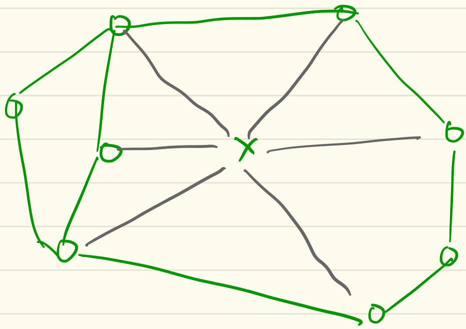
REMOVE EDGES



↓ ~~removing~~ e
collapse e

- remove e
- merge the two endpoints of e.

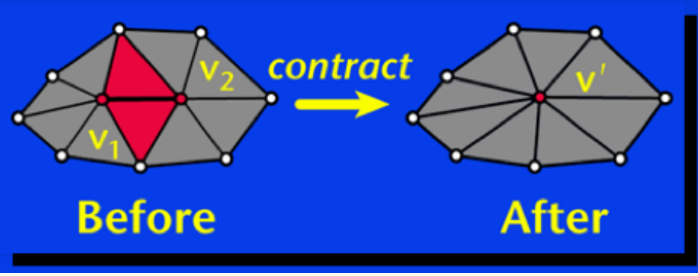




1. maintain co-planarity
2. maintain new faces
3. clean away duplicate edges.

Contract vertex pair $(v_1, v_2) \rightarrow v'$

- Move v_1 and v_2 to position v'
- Replace all occurrences of v_2 with v_1
- Remove v_2 and degenerate triangles
- Typically, we contract edges, as others have d



ALGORITHM (myMesh $\forall m$)

$m_0 = m$;
$i = 0$;
do {

all the complication is here !

① choose an edge of m_i say e_i .

② collapse e_i

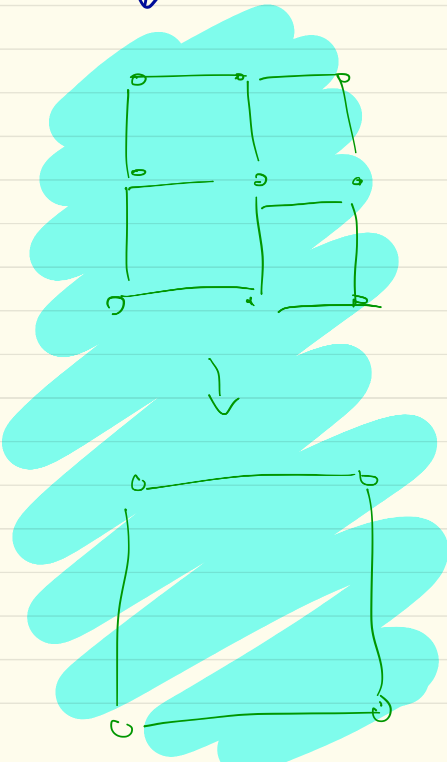
③ new mesh m_{i+1} ; i++ ;

} while (m_i is too complicated) ;

Q: which edge to collapse?

→ pick an edge that creates the smallest 'error'.

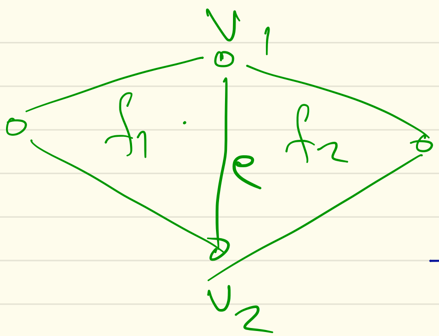
Smallest length edge



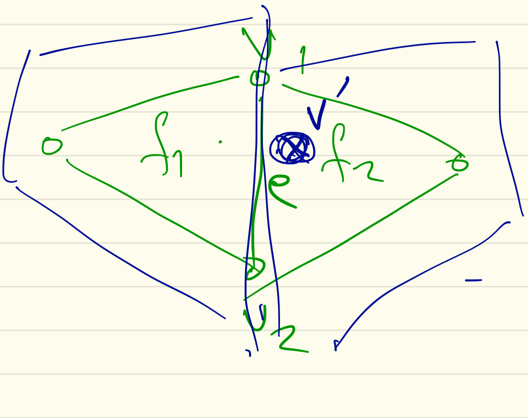
lots of clever ideas in the graphics literature.

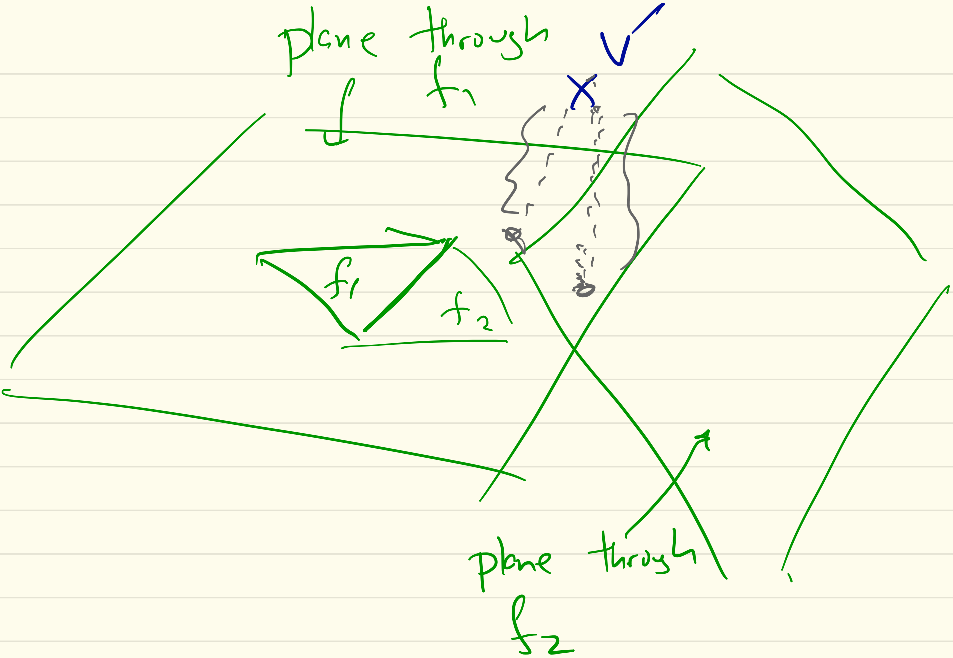
Q: which edge to collapse?

<u>FOR NOW:</u> pick an edge that minimizes the point to plane distence!



- f_1 & f_2 define 2 planes
- error: for e, find a new vertex position v' that minimizes the error of v' to the 2 planes passing through f_1 & f_2
-





Minimize distance of v' to the two planes f_1 & f_2.

$\rightarrow$ $\cancel{\underline{\mathrm{dist}(v', f_1) + \mathrm{dist}(v', f_2)}}$

$\downarrow$ error.

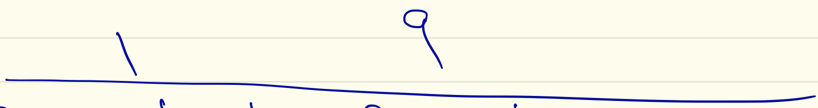
$\rightarrow$ $\mathrm{dist}(v', f_1)^2 + \mathrm{dist}(v', f_2)^2$

→ Why use dist^2 instead of dist?

1. avoid taking a square-roots for efficiency

2. mathematically easier to minimize
nicer inequalities ...

3. <u>example</u>:



- approximate 1 & 9 with a new number v'

- error of the approx. by sum of distance from v' to 1 and 9

$v' = 2 \longrightarrow (2-1) + (9-2) = 8$

$v' = 5 \longrightarrow (5-1) + (9-5) = 8$

$1 \qquad\qquad\qquad 9$

$$\left.\begin{array}{l} v' = 2 \rightarrow \text{error} = 8 \\ v' = 5 \rightarrow \text{error} = 8 \end{array}\right\} \text{sum of distences.}$$

$$\left.\begin{array}{l} v' = 2 \rightarrow (2-1)^2 + (9-2)^2 = 50 \\ v' = 5 \rightarrow (5-2)^2 + (9-5)^2 = 32 \end{array}\right\} \begin{array}{l}\text{sum of} \\ \text{squares.}\end{array}$$

Sum of squares tries more to make the ~~e~~distances more equal

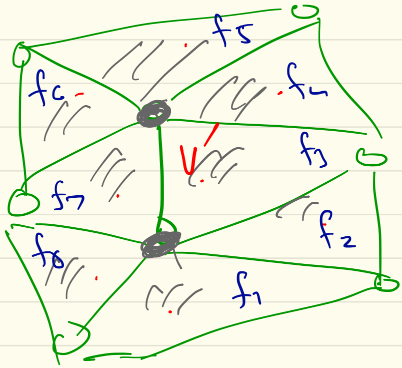
$\vdots$

$\text{dist}^3, \text{dist}^4 \ldots \underbrace{\text{dist}^\infty}_{\text{max.}}$

Back to business:

M

- find an edge of M that minimize the distance^2 to the surrounding plane.



find a point v' that minimize $\sum_{i=1}^{8} \left(d(v', f_i)\right)^2$.

M_1

$\downarrow$ contract e_1 minimum sum of square distance.

M_2

$\downarrow$ contract e_2

M_3

$\vdots$

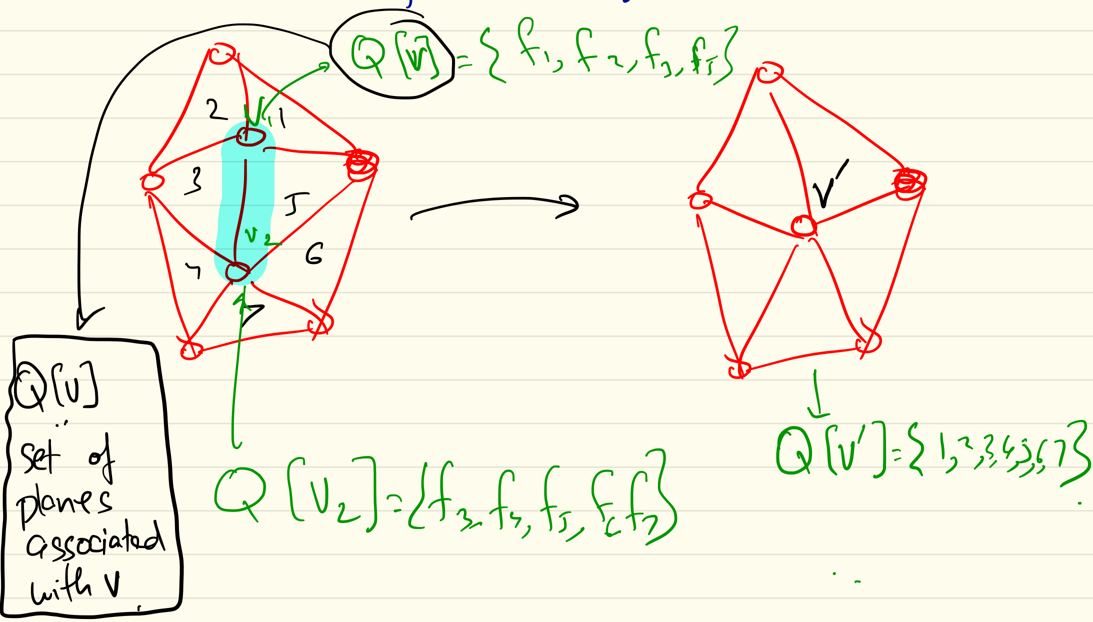
$\downarrow$ contract e_t

M_t

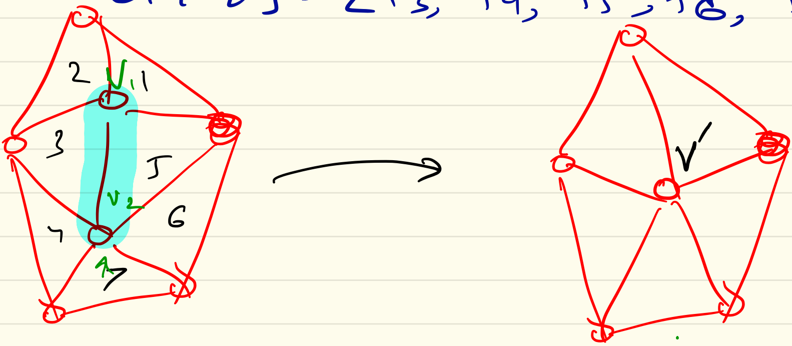
Idea: never contract the same vertex twice.

a restriction on choices of contractible edges. :-(

Idea: the set of planes from which we measure the distance² will come from original mesh



$$Q[v_1] = \{f_1, f_2, f_3, f_5\}$$
$$Q[v_2] = \{f_3, f_4, f_5, f_6, f_7\}$$



- For each vertex, define the set Q to be a list of planes associated with that vertex.
- To compute the error of contracting $v_1 - v_2$, minimize

$$\text{error}(v') = \sum_{\text{plane } h \in Q[v_1] \cup Q[v_2]} \left(d(v', h)\right)^2$$

- Set $Q[v'] = Q[v_1] \cup Q[v_2]$.

Algorithm Outline

Initialization

- Compute quadric **Q** for each vertex
- Select set of valid vertex pairs (edges + non-edges)
- Compute minimal cost candidate for each pair

Iteration

- Select lowest cost pair (v_1, v_2)
- Contract (v_1, v_2) — **Q** for new vertex is $\mathbf{Q}_1 + \mathbf{Q}_2$
- Update all pairs involving v_1 & v_2

DETAILS

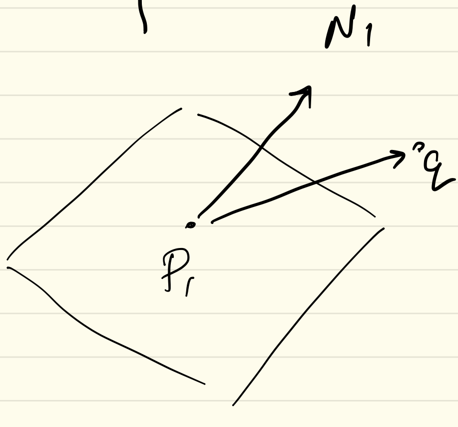
- point v', plane h_1.
- Compute $\text{distance}(h_1, q')^2$:

h_1: defined by $\rightarrow$ a point P_1
$\rightarrow$ a norm N_1.

if q lies on the plane h_1 $\quad (q - P_1) \cdot N_1 = 0$.

if q does not lie on h_1, then what is $\left| (q - P_1) \cdot N_1 \right|$?

distance of q to h_1 !



- point v', plane h_1.

- Compute $\text{distance}(h_p, q')^2$:

h_1: defined by → a point P_1
→ a norm N_1

$$\left|(q - P_1) \cdot N_1\right|$$ ← distance of q to h_1

$$= \left|q \cdot N_1 - P_1 \cdot N_1\right|$$

$$N_1 = (n'_x, n'_y, n'_z)$$

$$q = (x, y, z)$$

$$\left| n'_x \cdot x + n'_y \cdot y + n'_z \cdot z - \boxed{P_1 \cdot N_1} \right|$$

d

$$= \left| q \cdot N_1 - P_1 \cdot N_1 \right|$$

$$N_1 = (n'_x , n'_y , n'_z)$$

$$q = (x , y , z)$$

$$\left| n'_x \cdot x + n'_y \cdot y + n'_z \cdot z \underbrace{\boxed{- P_1 \cdot N_1}}_{d} \right|$$

set $H_1 = \begin{pmatrix} n'_x \\ n'_y \\ n'_z \\ d \end{pmatrix}$ 4×1

$q = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$ 4×1

then $\underset{1\times 4}{q^T} \underset{4\times 1}{H_1}$: distance of q to h_1.

distance^2 sum of v' to a set of plane Q.

$$\sum_{h_i \in Q} \operatorname{dist}(v', h_i)^2$$

$$= \sum_{h_i \in Q} \Big(\underset{1\times 4}{v'^T} \cdot \underset{4\times 1}{H_i} \Big)^2$$

$$= \sum_{h_i \in Q} (v'^T \cdot H_i)(v'^T H_i)$$

$$= \sum_{h_i \in Q} (v'^T \cdot H_i)(H_i^T \cdot v')$$

$$= \sum v'^T (H_i \cdot H_i^T)\, v'$$

$$= v'^T \Big(\sum_{h_i \in Q} H_i \cdot H_i^T \Big)\, v'.$$

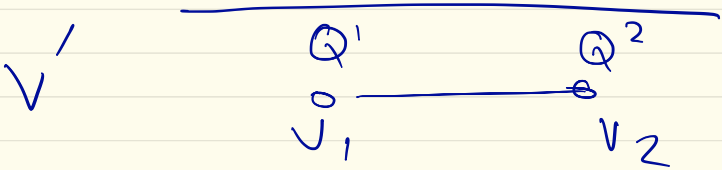
$$= \sum_{h_i \in Q} \left(v'^T \cdot H_i\right)\left(H_i^T \cdot v'\right)$$

$$= \sum v'^T \left(H_i \cdot H_i^T\right) v'$$

Define $\underbrace{Q_i}_{4\times 4}$ as the matrix $\underset{4\times 1}{H_i} \cdot \underset{1\times 4}{H_i^T}$.

$$= \boxed{v'^T \left(\sum_{h_i \in Q} Q_i\right) v'}$$

need only the sum of Q_i's!

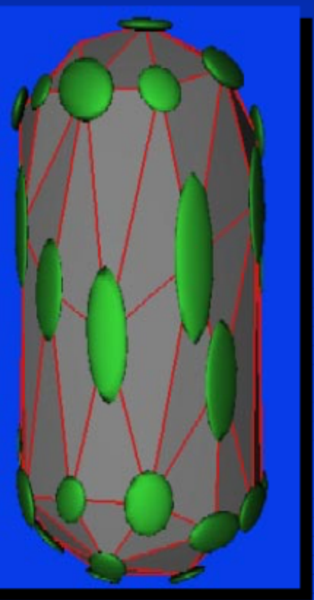


store only one matrix with each v.

$$v'^T \left(\sum_{h_i \in Q^1} Q_i^1\right) v' + v'^T \left(\sum_{h_i \in Q^2} Q_i^2\right) v'$$

$$= v'^T \left(\sum_{h \in Q^1 \cup Q^2} Q_i\right) v'$$

But What Are These Quadrics Really Doing?



Almost always ellipsoids

- When **Q** is positive definite

Characterize error at vertex

- Vertex at center of each ellipsoi
- Move it anywhere on ellipsoid with constant error

Capture local shape of surface

- Stretch in least curved direction

Finally: need to compute, given a Quadric error matrix Q, the vertex that minimizes the error to Q.

$$\underbrace{v'^T \cdot Q \cdot v}_{\text{minimized}} \qquad v = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

just a polynomial of degree two.

- differentiate in each of $\frac{\partial}{\partial x} \frac{\partial}{\partial y} \frac{\partial}{\partial z}$
- set to 0 and solve to find the best placement for v'.