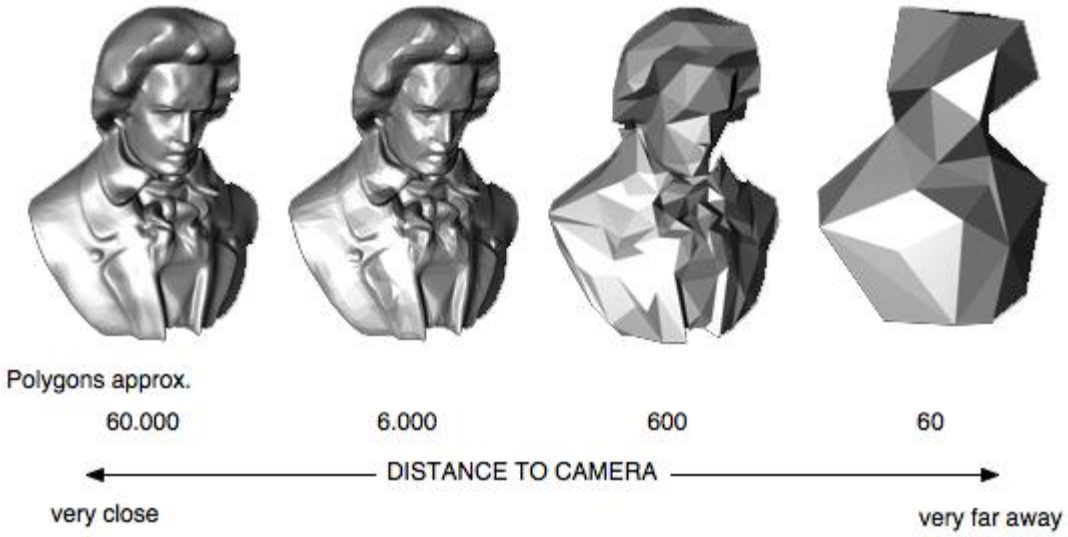
CG FINAL PROJECT

F74011255 陳易

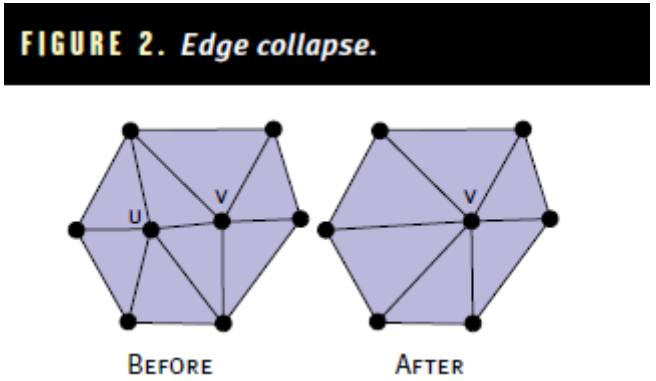
Level of Detail



Find Minimum Cost edge && Collapse

- 找出法向量最接近的兩個面，然後合併成一個

- 觀察：
  - 面與面之間會共用一邊
  - 越平滑得越不重要



How to remove?

- Remove any triangles that have both *u and v as vertices*
  - The collapse step
- Update

- Remove vertex *u*

How to pick u and v?

- formula

EQUATION 1. ***The edge cost formula.***

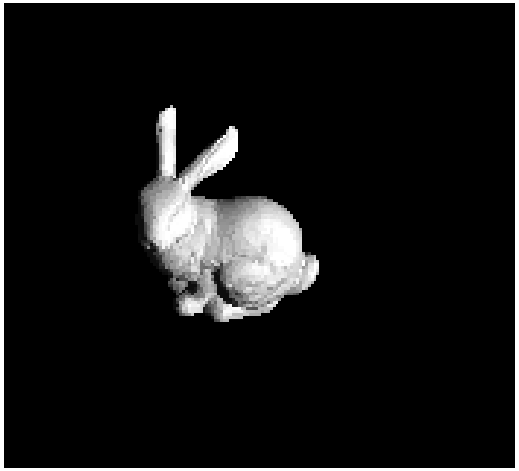
$$\text{cost}(u,v) = \|u - v\| \times \max_{f \in Tu} \left\{ \min_{n \in Tuv} \left\{ \left(1 - f.normal \bullet n.normal\right) \div 2 \right\} \right\}$$

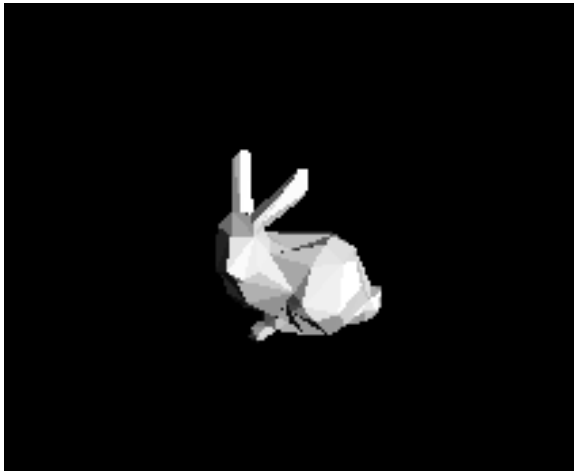
where Tu is the set of triangles that contain u and Tuv is the set of triangles that contain both u and v.

Structures

```
class Triangle
{
public:
    Triangle(Vertex* v0 , Vertex* v1 , Vertex* v2);
    ~Triangle();

    Vertex* vertex[3];
    glm::vec3 normal;

    void ComputeNormal();
    void ReplaceVertex(Vertex* vOld , Vertex* vNew);
    int HasVertex(Vertex* v);

};

class Vertex
{
public:
    Vertex(glm::vec3 position , unsigned int id);
    ~Vertex();

    glm::vec3 position;
    unsigned int id;
    vector<Vertex* > neighbors;//adjacent vertices
    vector<Triangle* > faces;//traveled faces
    float cost;//cached cost of collaspintg edge
    Vertex* collapse;

    void AddNeighbor(Vertex* v);
    void AddFace(Triangle* tri);
    void RemoveIfNoneNeighboor(Vertex* n);

    void RemoveFace(Triangle* triangle);
    void RemoveNeighbor(Vertex* v);
};
```

My Result





Demo Time

My reference

- http://dev.gameres.com/program/visual/3d/PolygonReduction.pdf