Object Representations Computer Graphics

Taku Komura

How to create a virtual world?

- Need to compose scenes
- Need to define objects
  - Characters
  - Terrains
  - Objects (trees, furniture, buildings etc)

Problem

- World is continuous and infinite
- Computers represent things discretely and resources are finite
- Need a compact representation of objects
  - Can be loaded quickly
  - Can be rendered quickly

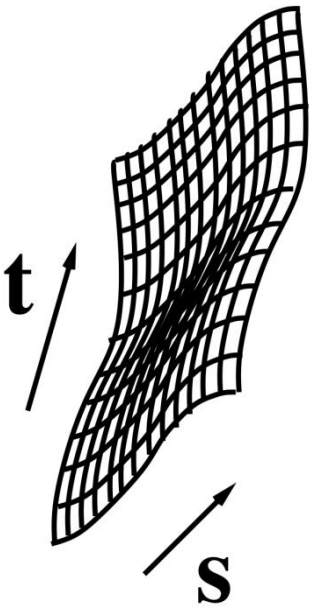
Overview for today

- Description of some representations
  - Parameteric surfaces
  - Metaballs
  - Mesh structures
    - Polygon format, triangle strip, triangle fans
    - Quad mesh
- How to produce mesh data

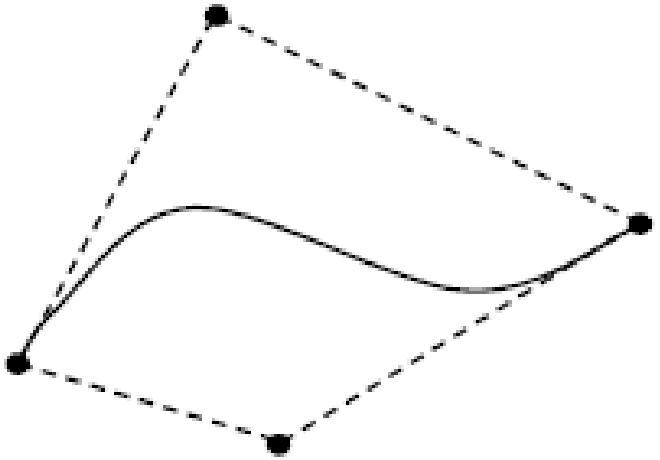
Parameteric surfaces

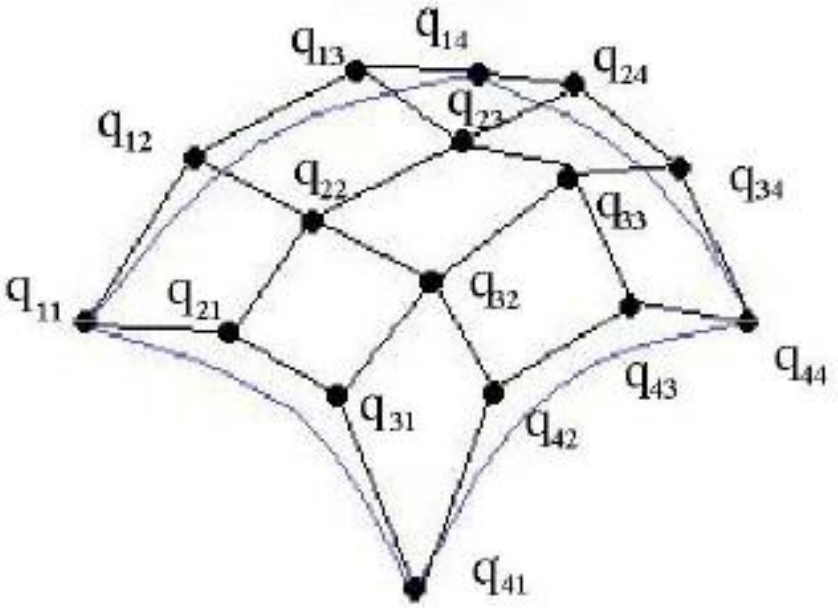
- Demand for continuous, smooth surfaces that are easy to control
  - Demand from the industry
- **Solution :** Using polynomial curves / surfaces to represent the shapes
- Quadratic, cubic curves and surfaces

$$x(t) = a_x \cdot t^3 + b_x \cdot t^2 + c_x \cdot t + d_x,$$
$$y(t) = a_y \cdot t^3 + b_y \cdot t^2 + c_y \cdot t + d_y,$$
$$z(t) = a_z \cdot t^3 + b_z \cdot t^2 + c_z \cdot t + d_z,$$



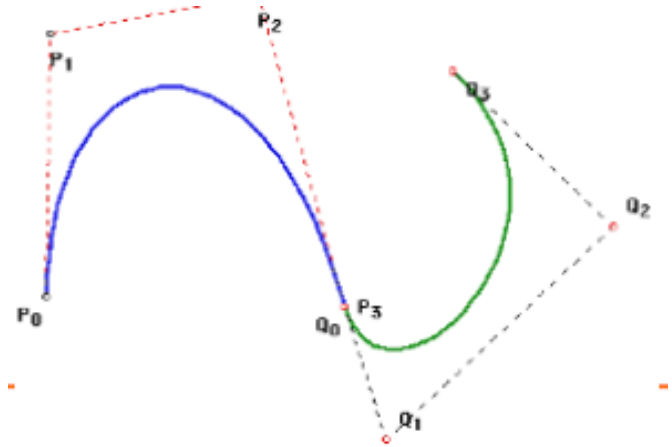
Bezier curve / surface





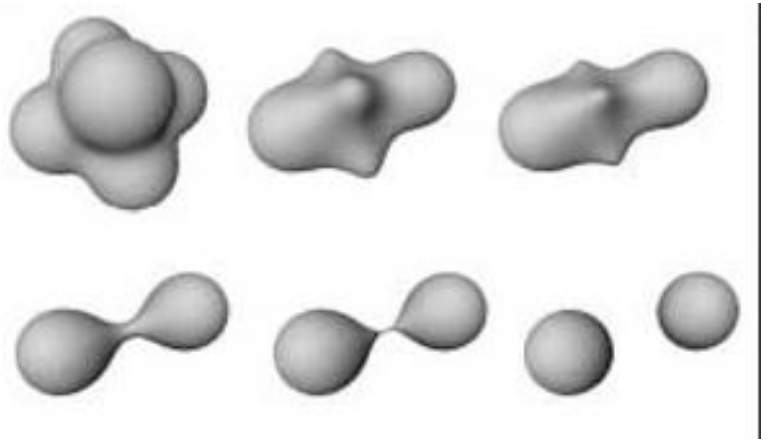
Increasing the controllability

- Quadratic, cubic surfaces can only produce smooth, low frequency surfaces
- Increasing the degree is not such a good idea
  - Quadric (4), quintic (5), sextic (6), septic (7)
  - Not easy to control
  - Require more computation
- Solution: Concatenate



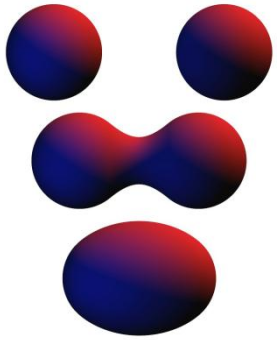
Problems with Parameteric Surfaces?

- Rendering blobby objects like liquid, clay might be difficult
  - Topology is dynamically changing



Metaballs, blobs

- Implicit surface
- Representing an object by the equipotential surface of electrons



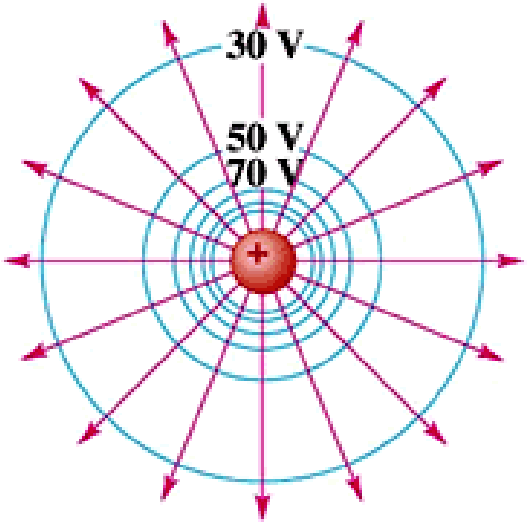
- Suitable for blobby objects that may stick and split
  - Can handle topological changes well

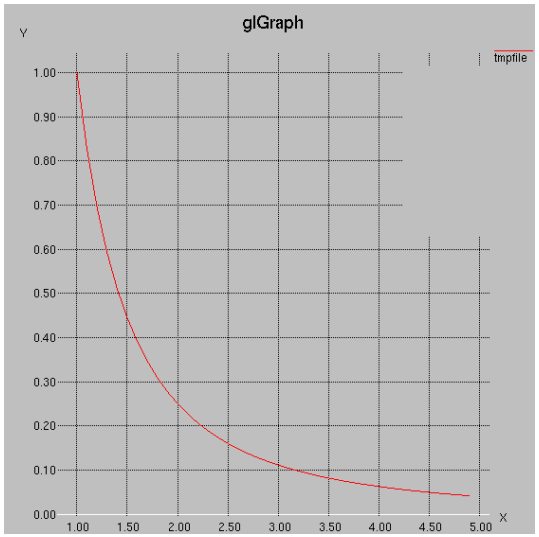
Equipotential Surface (isosurfacing)

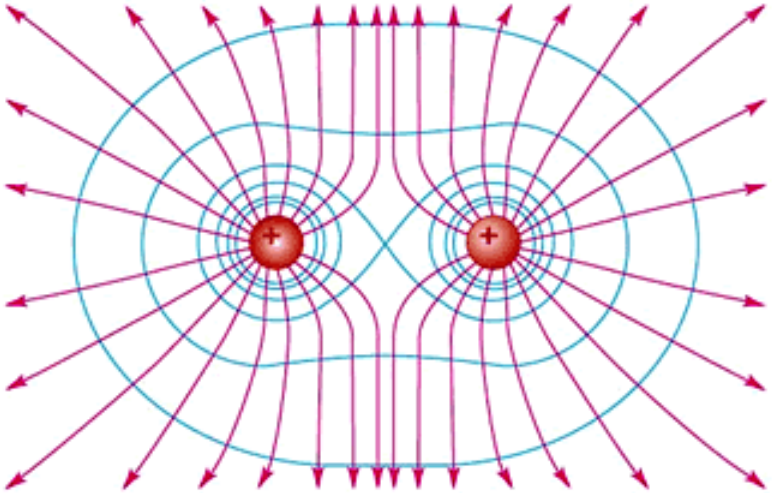
- Every electron produce a electric potential that falls off according to the distance
- The potential at arbitrary positions are computed by summation of all potentials

$$V = \sum_i^n \frac{a_i}{r^2}$$

- Specify the potential to produce the surface



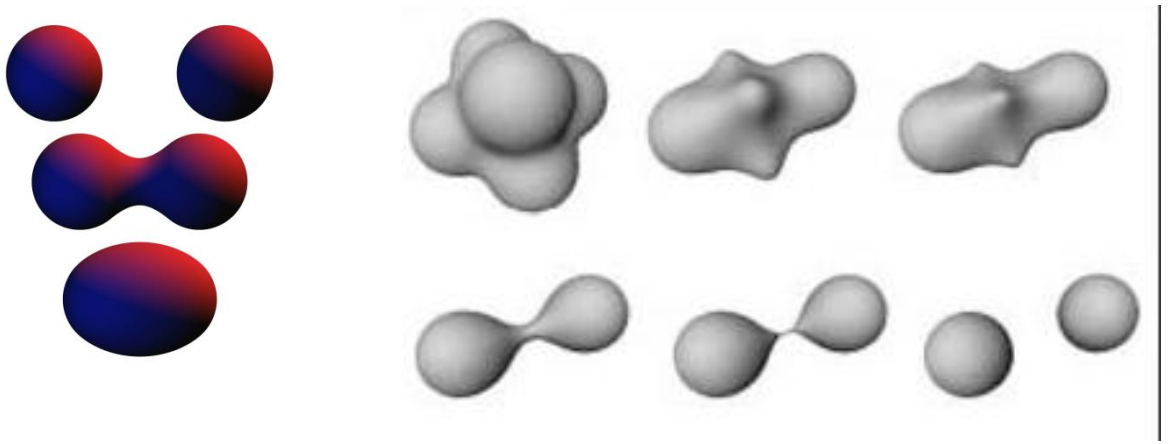




Generating objects by metaballs

- The location of the metaballs (electrons) and their charges are specified
- The threshold is specified

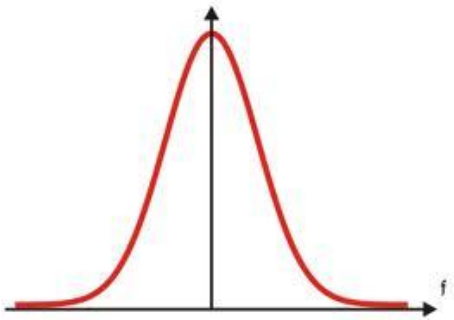
$$\sum_i^n \frac{a_i}{r^2} - \text{threshold} = 0$$



Potential function

- Can use other functions, i.e., Gaussian function

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$$



- Or finite support
  - Goes to zero at maximum radius
  - No need to evaluate the potential of those outside the support

Rendering metaballs

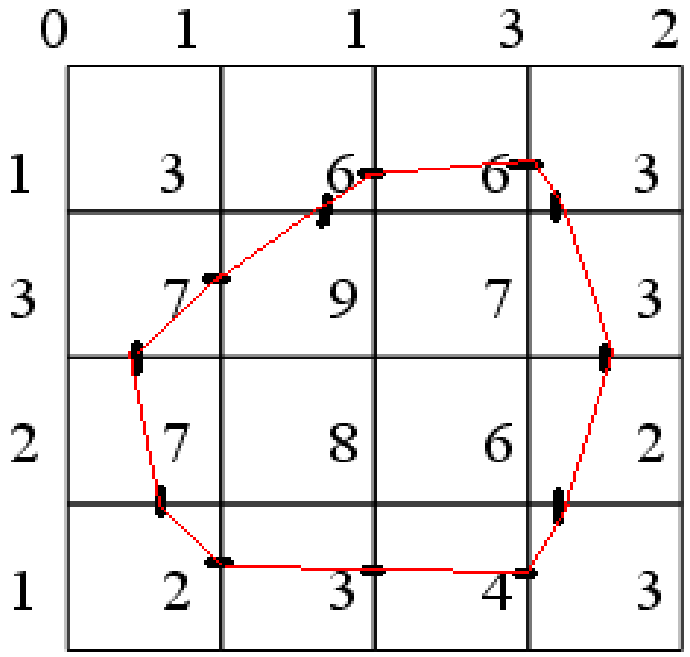
We cannot explicitly compute the isosurface

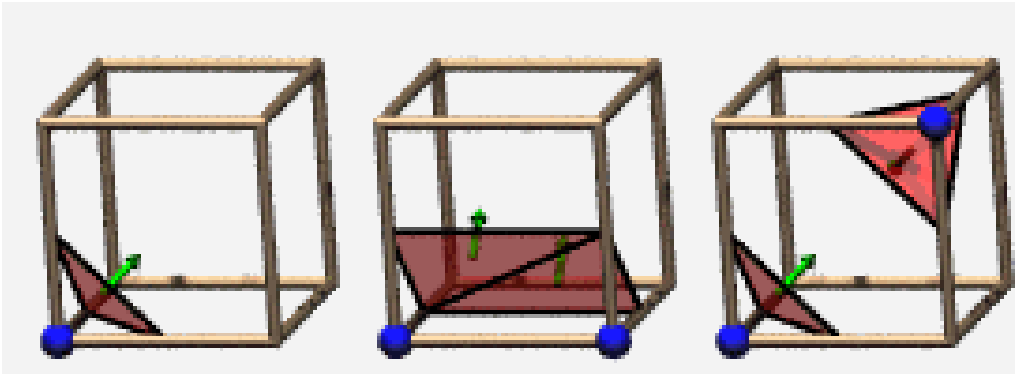
Two methods to render the isosurface

- Marching cubes
- Ray casting

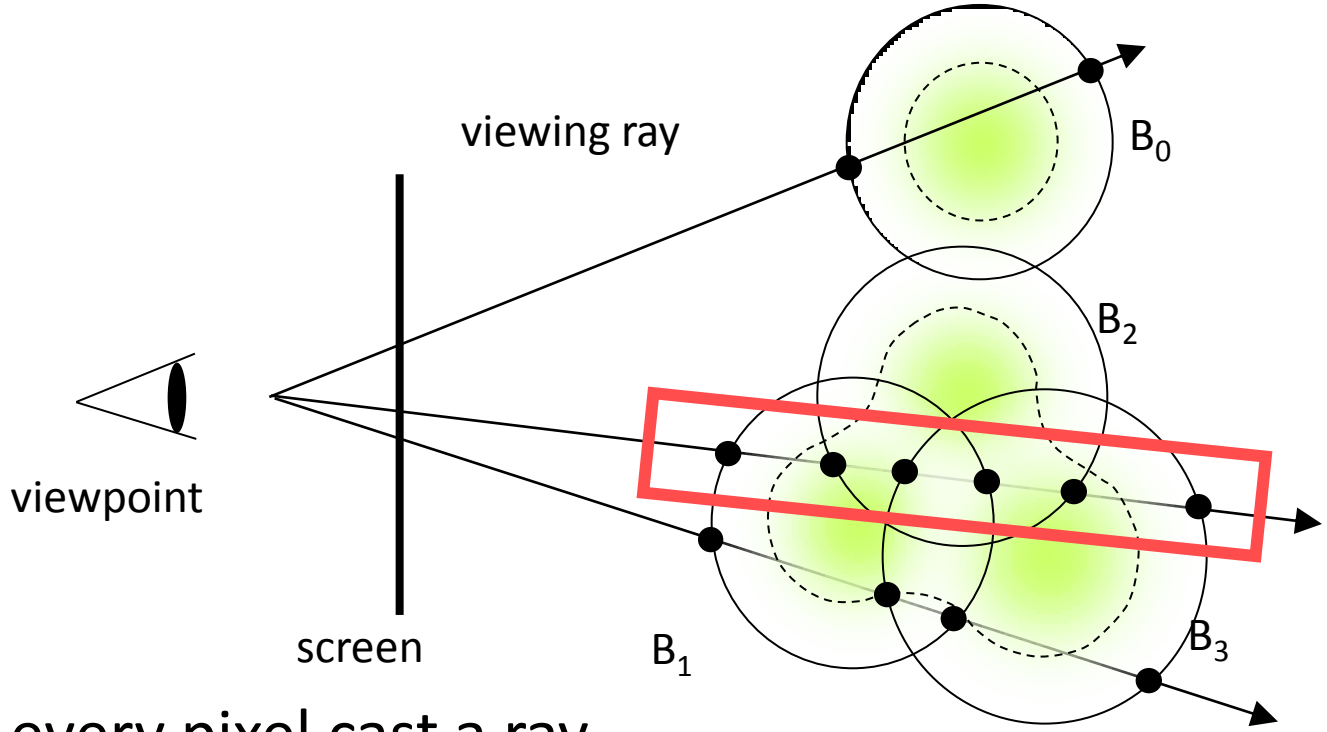
Marching cubes : Outline

- Prepare a grid
- Compute the potential at the grid points
- If the edges cross the threshold, produce the surface





Ray casting : Outline

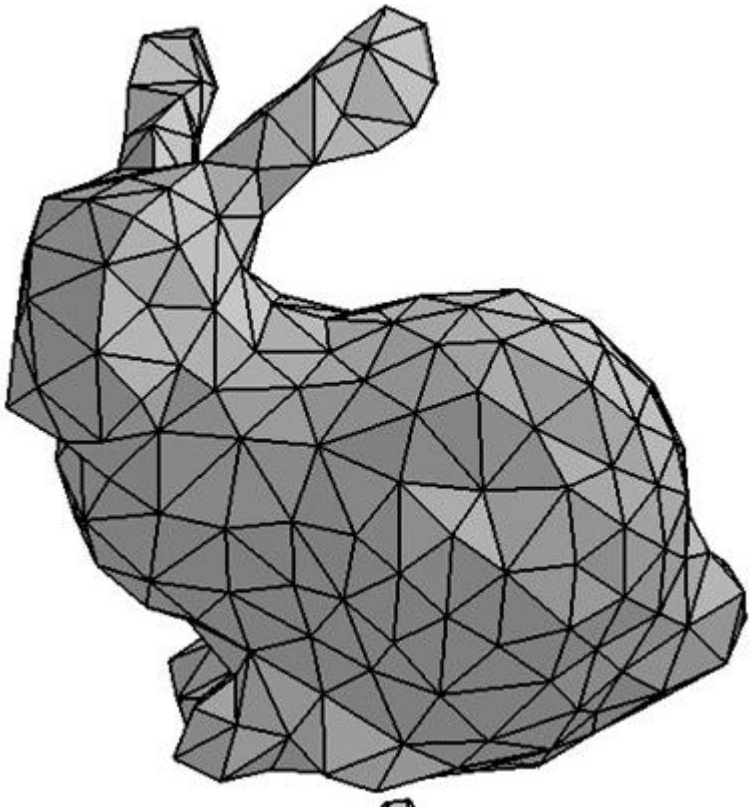


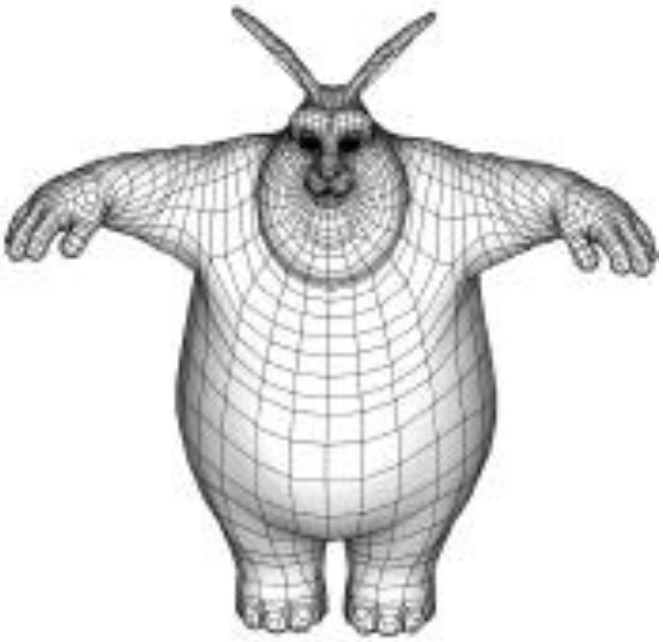
- For every pixel cast a ray
- Surround the metaballs by bounding spheres (use finite support)
- Check if the rays and the bounding spheres intersect
- If they do, do a finer test by, i.e., bisection method

Overview for today

- Description of some representations
  - Parameteric surfaces
  - Metaballs
  - Mesh structures
    - Polygon format, triangle strip, triangle fans
    - Quad mesh
- How to produce mesh data

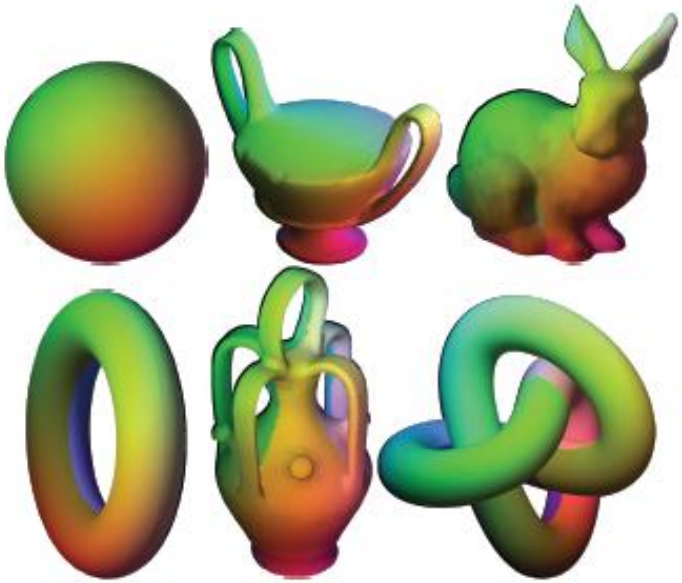
Mesh structure





Mesh structure

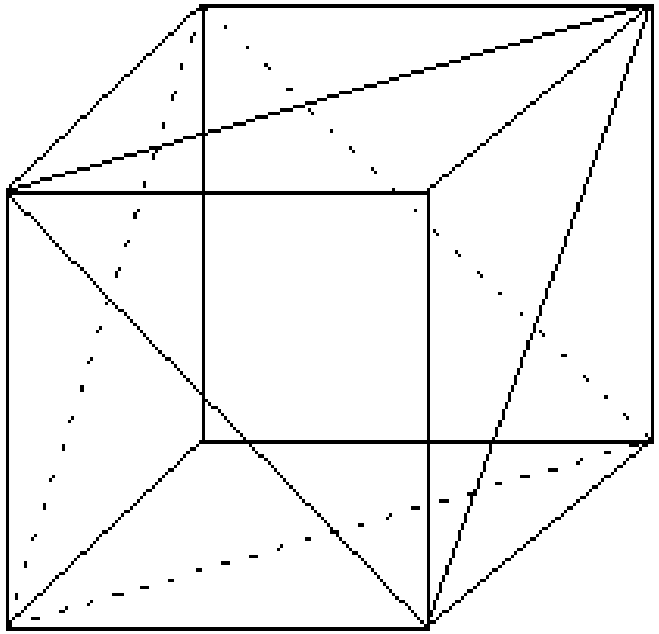
- Most general format
- A collection of polygons representing the shape
- Very low level but intuitive
- Can represent objects of any topology



Polygon format
(what is this shape?)

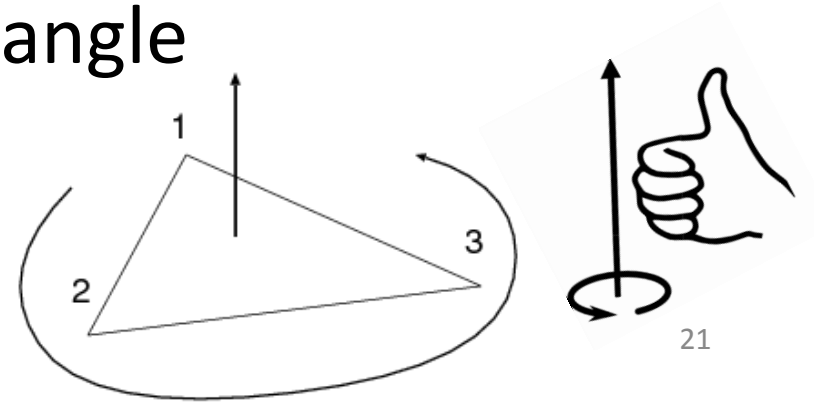
Vertex	Triangles	
0 1 0	1 2 3	1 5 2
0 0 0	1 3 4	5 6 2
1 0 0	1 4 8	2 6 7
1 1 0	1 8 5	2 7 3
0 1 -1	4 3 8	
0 0 -1	3 7 8	
1 0 -1	5 7 6	
1 1 -1	5 8 7	

Answer



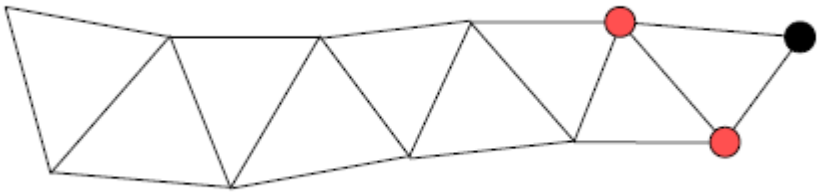
Polygon format

- Polygons of different number of vertices can be defined
- Most polygon file formats in this form
- For the polygons, the order of the vertices is important ： defines the normal direction
- Usually follows the right hand rule
- Requires 3 numbers per triangle



Tri-strip

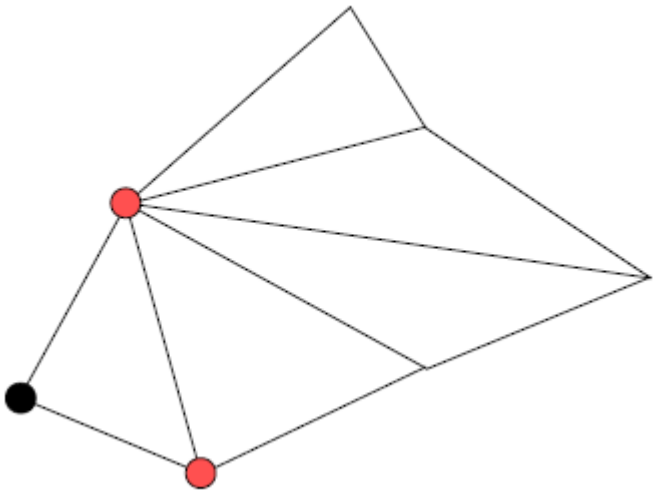
- Use triangles to represent a solid object as a mesh.
- Triangles frequently appear in strips :



- A new triangle is defined by 1 new vertex added to the strip.

Triangular fans

Triangles used in complex polygonal geometry.

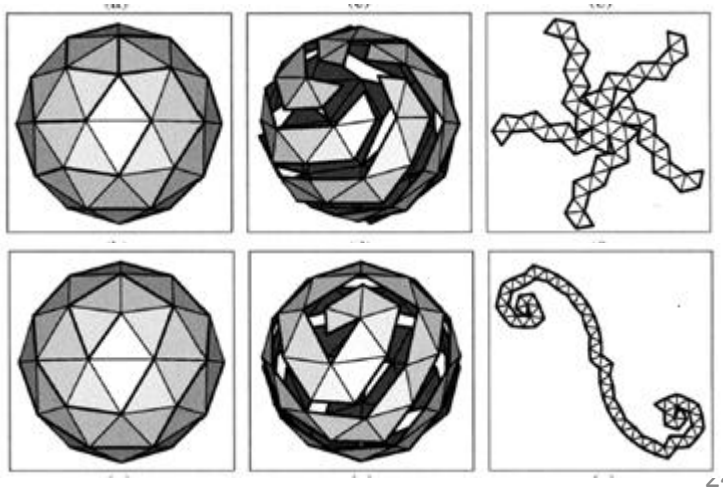


Triangular Fan

To add new triangle, only 1 vertex
needs to be added.
Red - existing vertices.
Black - new vertex

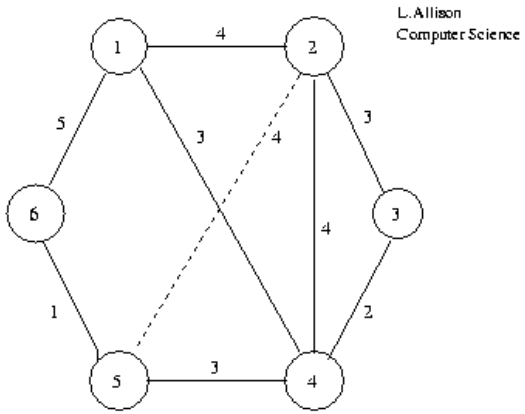
Topological surgery

- We can decompose meshes into a combination of triangle strips and fans to compress the data
- A few possible strategies:
  - Minimal spanning tree
  - Producing a spiral

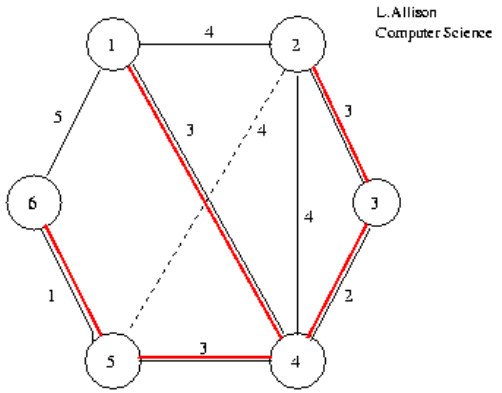


Decomposition by Minimal Spanning Tree

- Minimal spanning tree is a procedure to produce a tree that visits every node of a graph
- Every edge is given a cost
- It produces a tree with total minimal cost



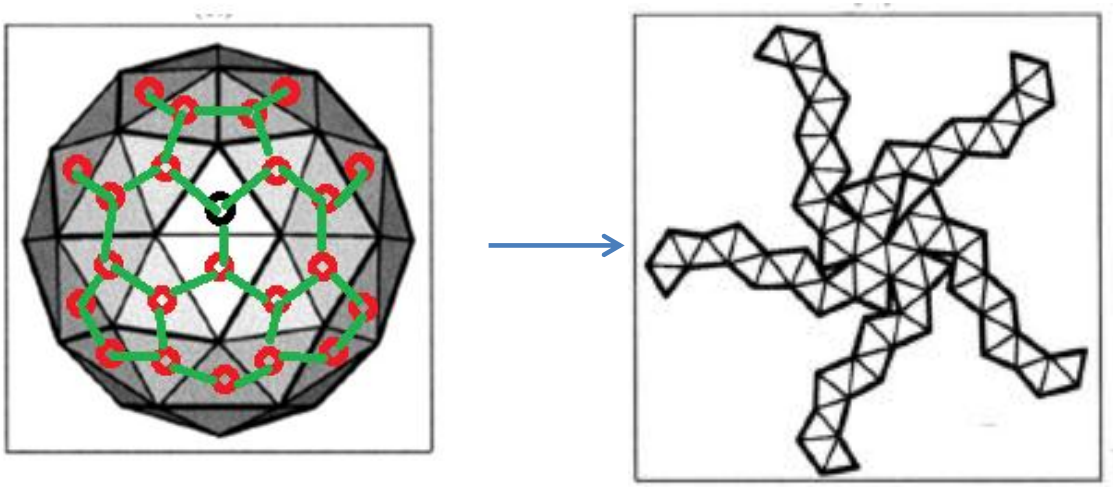
Weighted Undirected Graph



MST – red

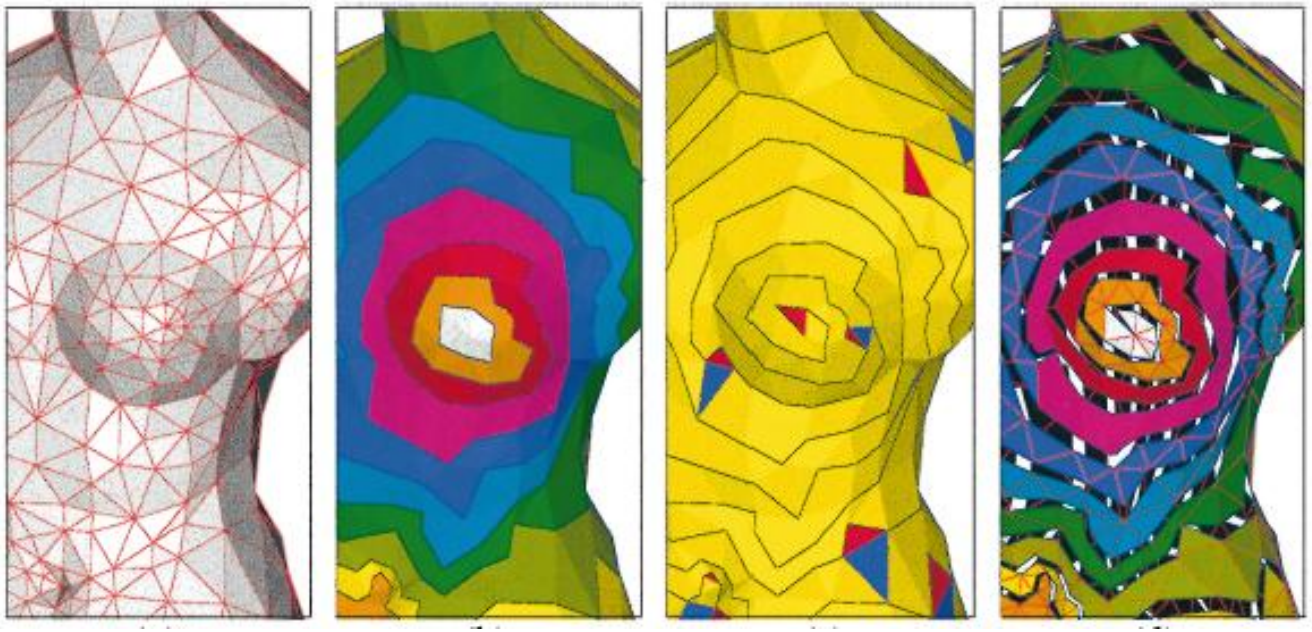
Decomposition by Minimal Spanning Tree

- Here, the triangles are the nodes of the graph
- The edges connecting the triangles are the edges of the graph
- For the cost, we can use the Euclidean distance between the triangle and the root triangle



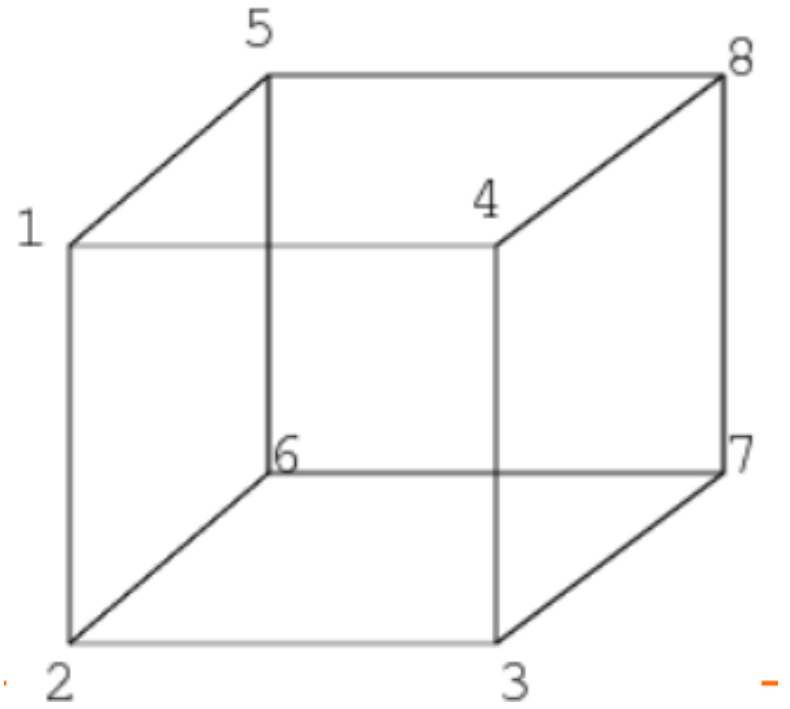
Decomposition into a spiral

- Produce layers of triangles
- Connect the layers and produce a single series of triangles



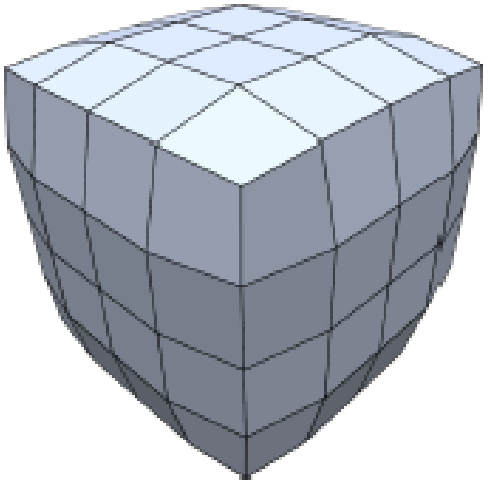
Can you surgery this one?

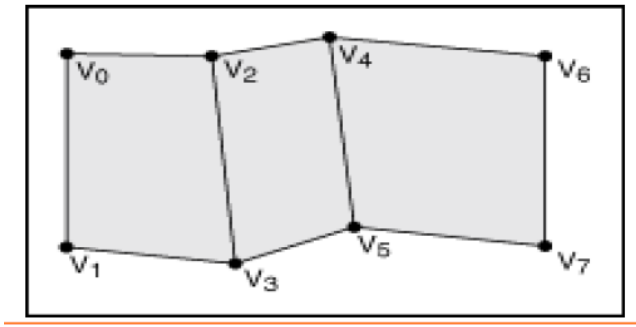
- What about a tree strip?
- What about a spiral strip?



Quadmesh

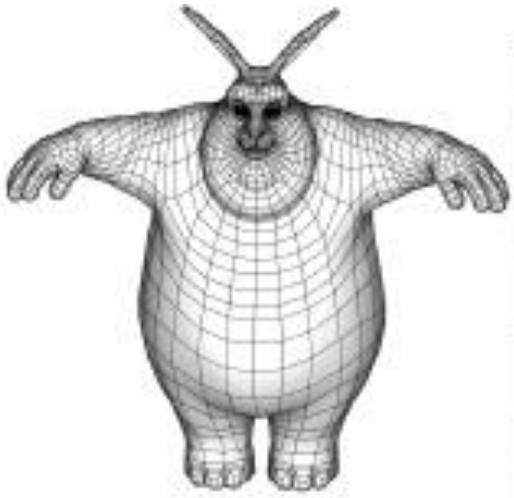
- Quadrilateral mesh (Quadmesh)
  - Polygons / strips made of quadrilaterals

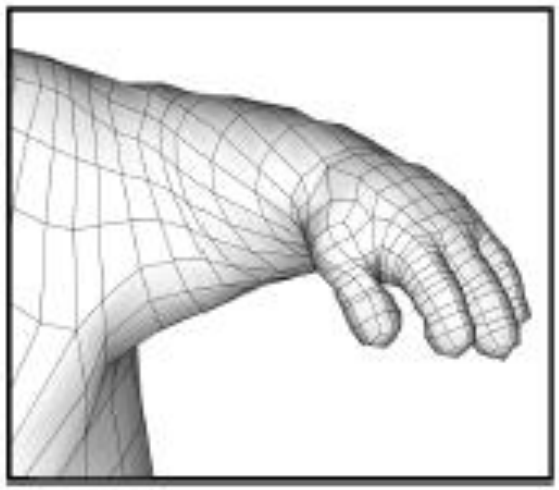


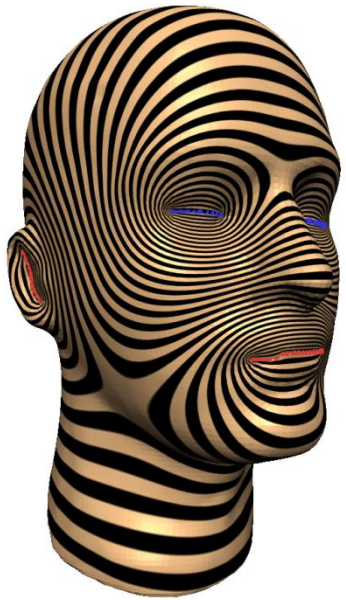


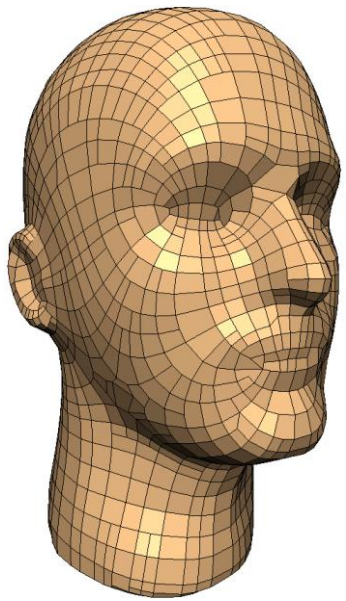
What is so good about Quadmesh?

- Easier to align the edges with the dominant local directions, such as principal curvature directions, or feature lines that define the object



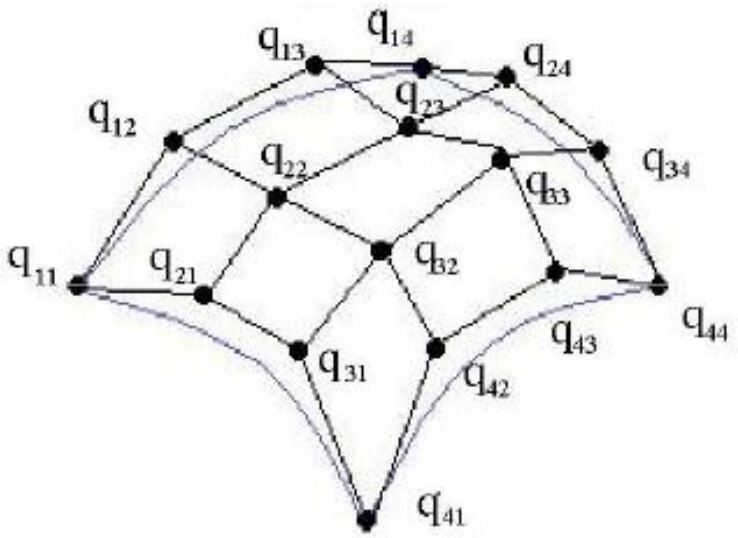






What is so good about Quadmesh

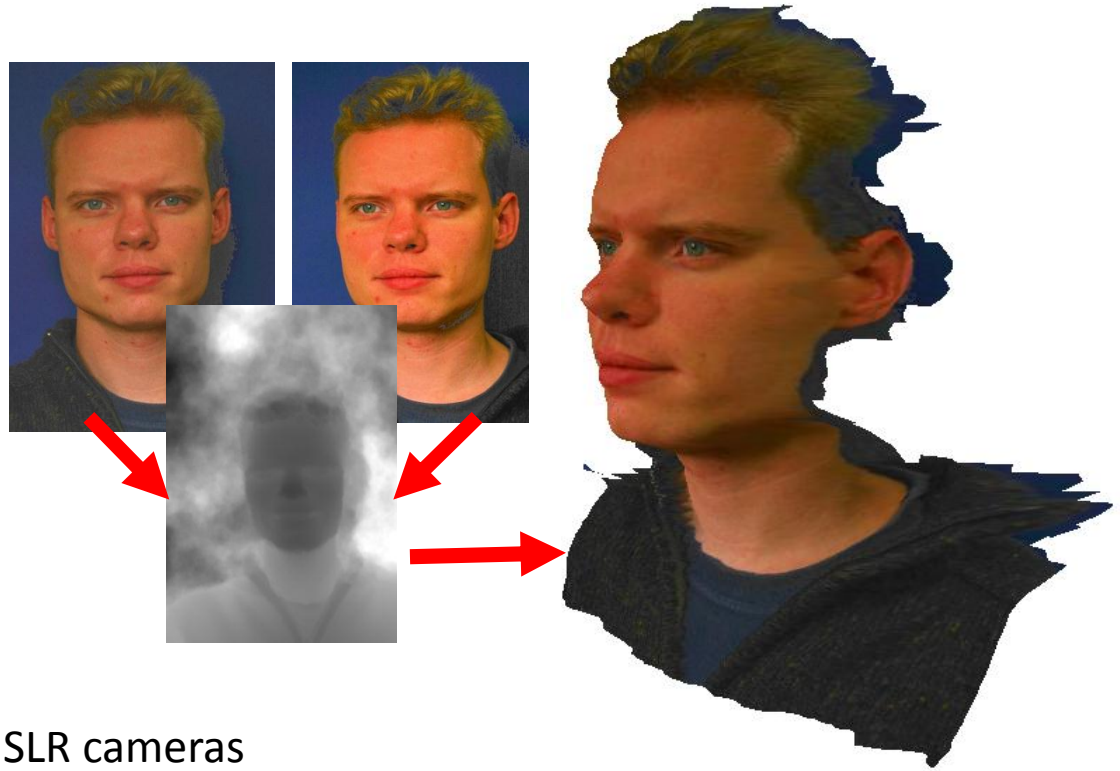
- Easier to fit with parametric surfaces
- Easier to produce a texture mapping
- Good for finite element simulation



Data production

- How do we produce polygon mesh data?
- Basically, 3 ways
  - Model by yourself
    - (using 3D modellers or write your own program)
  - Scan data
  - Procedural modelling

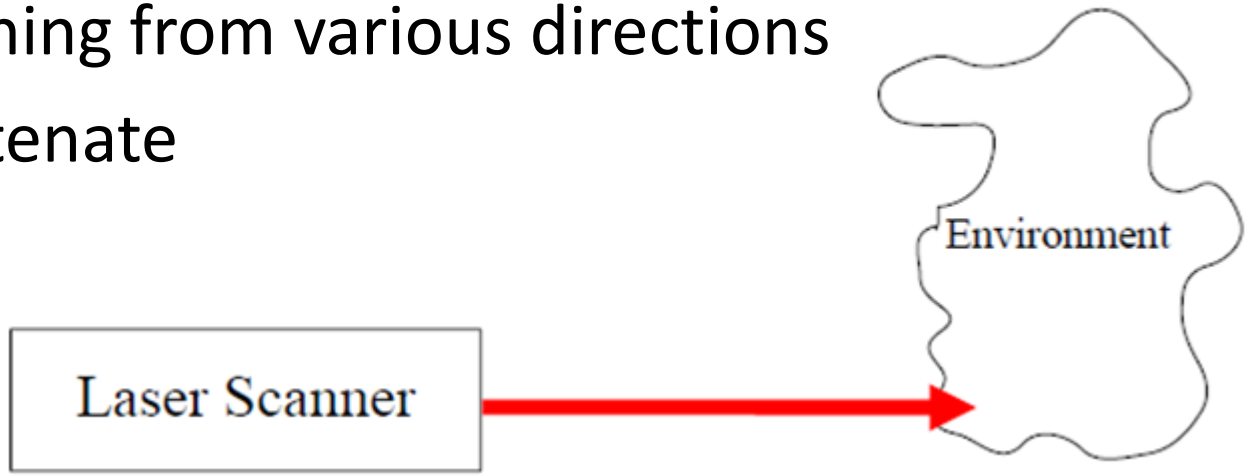
Scanning Data : Stereo Vision



- 2 x 6 mega-pixel digital SLR cameras
- Commercial 3D stereo software (http://www.di3d.com/)
  - *Results : 6 mega-pixel depth map / VRML 3D surface mesh model*

Scanning Data : Laser range scanner

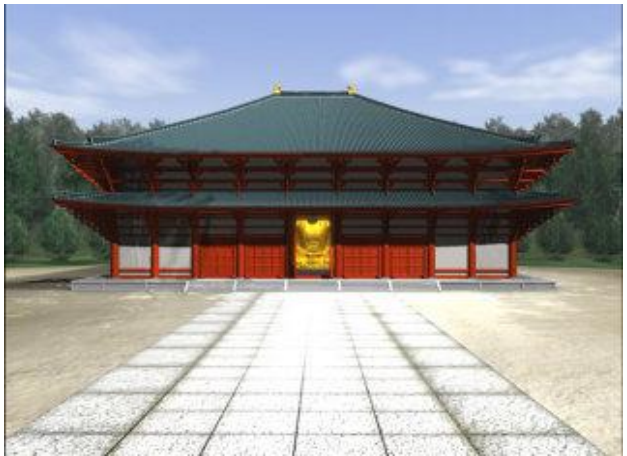
- Active depth sensing using laser beam signal
  - Evaluating the time required for the laser beam to bounce back
  - Accurate data and no ambiguity
  - 2 ½ - scanning from various directions and concatenate



Great Buddha Project in Japan

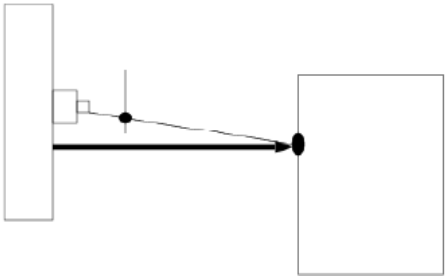
- Capturing took 3 weeks x 2 trips x 10 students/staff

http://www.youtube.com/watch?v=OoNr7DV0b-M&feature=channel_page



Microsoft KINECT

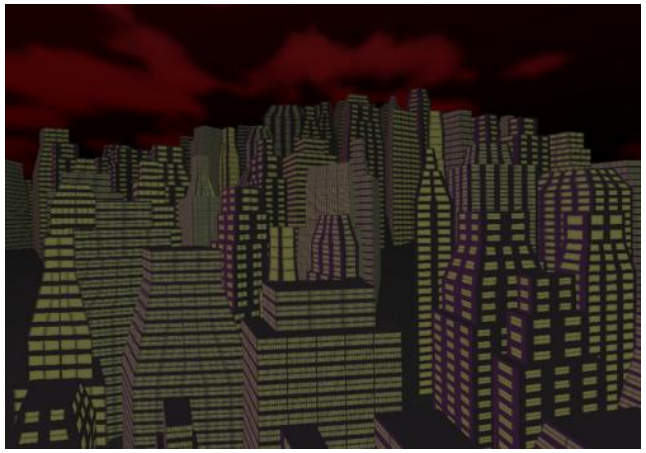
- Estimating the depth from projected structured infra-red light
- Practical ranging limit of 1.2–3.5 m (3.9–11 ft)

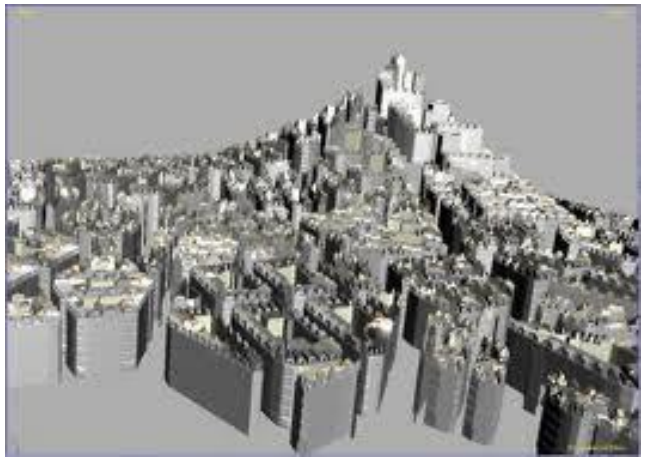


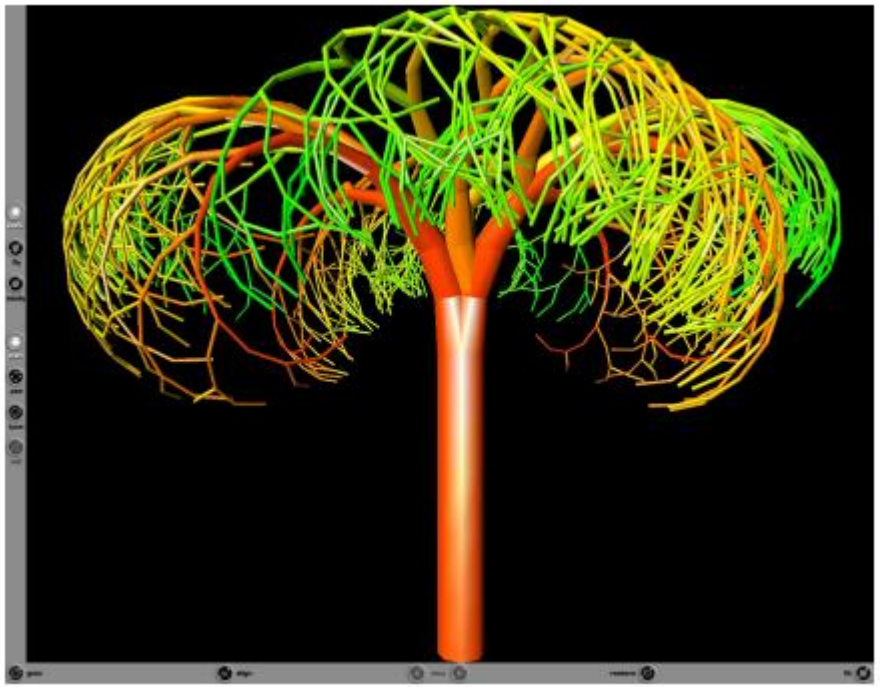
http://www.youtube.com/watch?v=dTKlNGSH9Po&feature=related

Procedural Modelling

- Producing new models based on rules







Readings

- **Geometric compression through topological surgery**, Taubin and Rossignac, ACM Transactions on Graphics 17(2) 1998
- State of the Art in Quad Meshing, Bommes et al. *Eurographics STARS, 2012*
- Blinn, J. F. (July 1982). "A Generalization of Algebraic Surface Drawing", ACM Transactions on Graphics, **1** (3): 235–256.
- Yoshihiro Kanamori, Zoltan Szego and Tomoyuki Nishita: "GPU-based Fast Ray Casting for a Large Number of Metaballs," Computer Graphics Forum 27-2, 2008