Introduction to Computer Graphics: Geometric Processing and Modeling

Instructor:	Prof. Xin Li xinli@lsu.com xinli_xmu@163.com		
Lectures:	Mon:	10:10 – 11:50am	@海韵教学楼 207
Labs:	Wed:	2:30 – 5:25pm	@ 海韵实验楼 305

Course Synopsis

- What is this course about?
  - Concepts, algorithms, programming in 3D Graphics and Geometric Modeling
    - About representing, rendering, analyzing, and manipulating 3D shapes and scenes

- Topics:
  - Basic Computer Graphics pipeline
  - Basic OpenGL programming for 3D Graphics
  - Basic Geometric Computing for 3D Computer Graphics
  - Modeling (representing + analyzing) 3D Shapes and their applications

Workload

- Reading
  - No required textbook
  - Please follow the lecture and course slides closely
  - Learn OpenGL following the online tutorial
  - Do necessary research and online search to solve problems in your homework implementation
- Programming
  - 4 programming homework projects
  - You are expected to finish the homework by every Wednesday

Grading

- Homework (100%)
  - Homework 1 (20%)
  - Homework 2 (20%)
  - Homework 3 (30%)
  - Homework 4 (30%)

Prerequisites and Requirements

- Basic linear algebra, calculus
- C/C++ programming background
  - Some starter codes will be provided in C++ for homework projects

- Self-learning:
  - Objective Oriented Programming in C/C++ (to understand homework starter codes)
  - OpenGL Programming

Be comfortable in read/writing C++ codes such as the following:

Codes often written in Object-Oriented Programming style

```
class TriangleMesh {
public:
        struct Point { double x, y, z;};
        struct Face { int pt_ids[3]; };

         TriangleMesh(void){}
        ~TriangleMesh(void){}

        void loadMesh(char *filename);

        Point * vertexList;
        Face * faceList;
};
```

Be comfortable in read/writing C++ codes such as the following:

```
#include <vector>
class TriangleMesh{
public:
        struct Point { double x, y, z;};
        struct Face { int pt_ids[3]; };

         TriangleMesh (void){}
        ~ TriangleMesh (void){}

        void loadMesh(char *filename);

        std::vector<Point> vertexList;
        std::vector<Face> faceList;
};
```

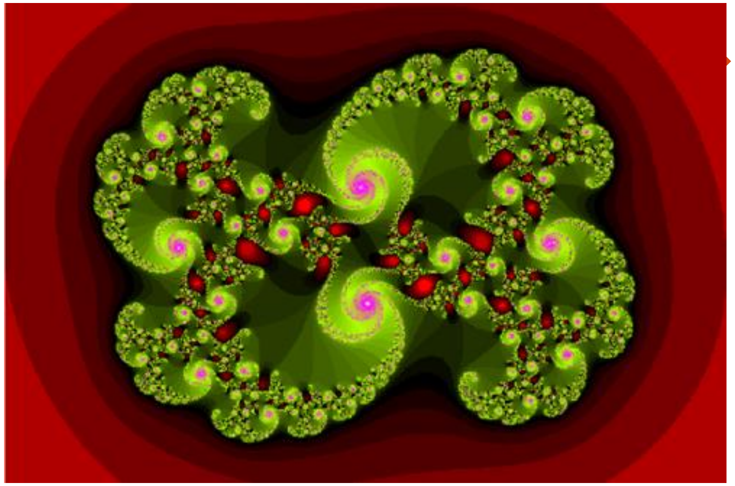
Sometimes you will see codes written using Standard Template Library (STL)

What is Computer Graphics?

The creation of, manipulation of, analysis of, and interaction with pictorial representations of objects and data using computers.

-- Dictionary of Computing

- A picture is worth a thousand words.



It looks like a swirl. There are smaller swirls at the edges. It has different shades of red at the outside, and is mostly green at the inside. The smaller swirls have purple highlights. The green has also different shades. Each small swirl is composed of even smaller ones. The swirls go clockwise. Inside the object, there are also red highlights. Those have different shades of red also. The green shades vary in a fan, while the purple ones are more uni-color. The green shades get darker towards the outside of the fan …

Why Computer Graphics?

CG techniques utilized in various scientific, engineering, and entertainment tasks:

- CAD/CAGD/CAM: to digitize the world + simulate the scientific phenomena and engineering manufacturing
- Visualization: CG has become the dominant form of computer output → help better display/analyse scientific phenomena/discoveries
- 3D Computer Vision and AI: teach computer to understand captured images/videos better
- And many more…

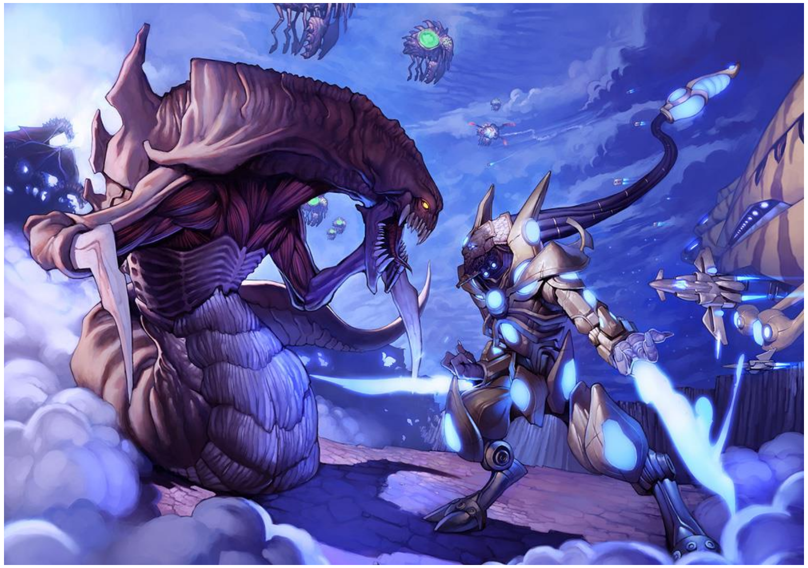
Movies

- CG has been changing Special Effects in the Movie Industry (Billions of dollars spent)
- Physically based animation makes them realistic

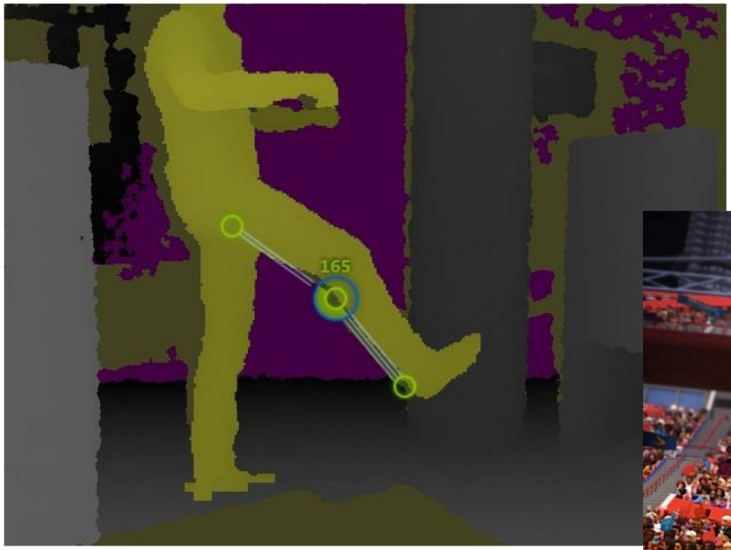
Video Games

- Important driving force
- Realistic rendering + interactive animations
- Try to avoid heavy computation and use various CG tricks

Quake IV



Somatosensory Video Games

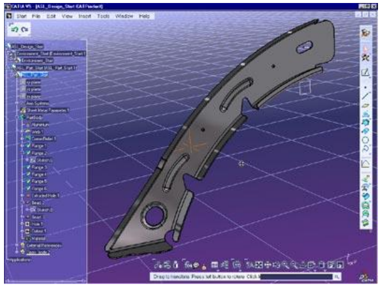


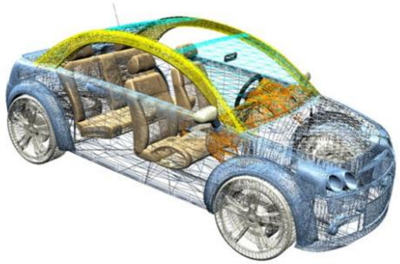
Microsoft XBox360 Kinect games

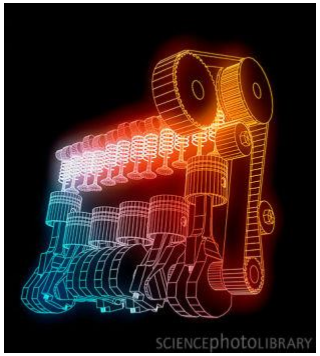
Computer Aided Design/Manufacturing

Significant impacts on the design process:

- Mechanical, electronic design
  (executed entirely on computer environments)
- Architectural & product design
  (being migrated to computer environments)

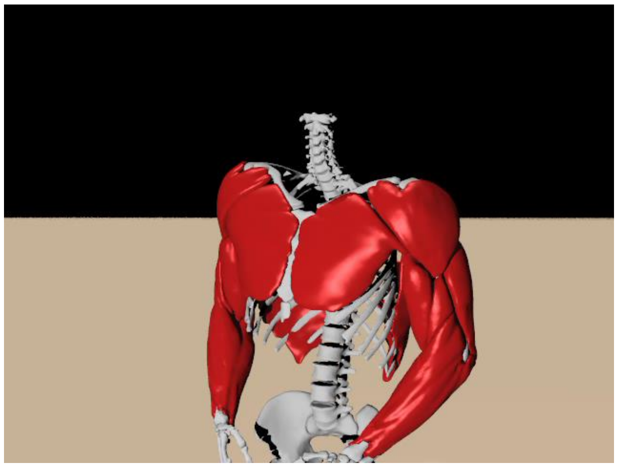


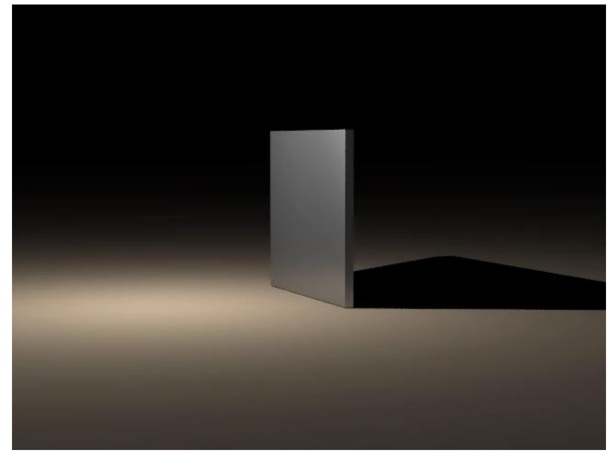


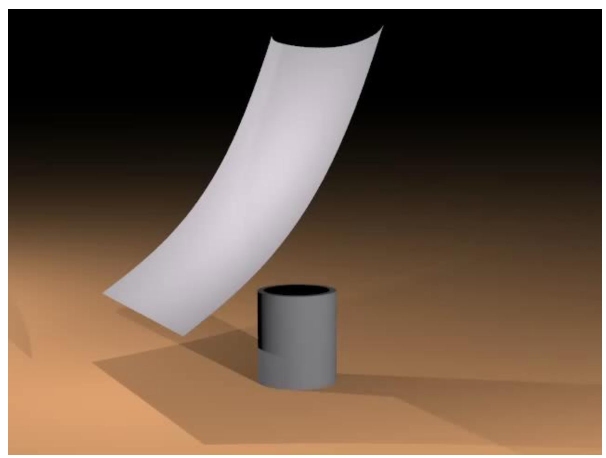


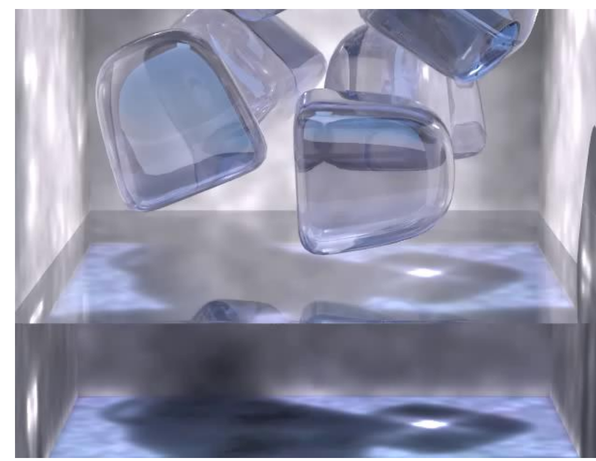
- ❑ Interactive design/visualization → assist modeling
- ❑ Simulating their behavior in the virtual environment

Scientific Simulation for Education



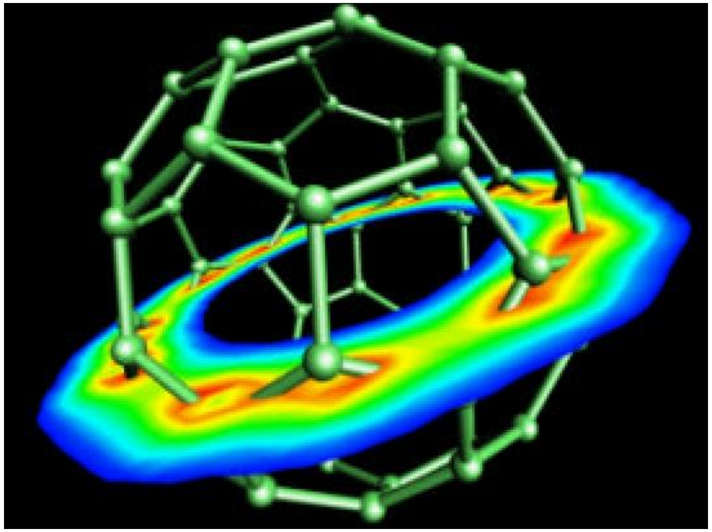




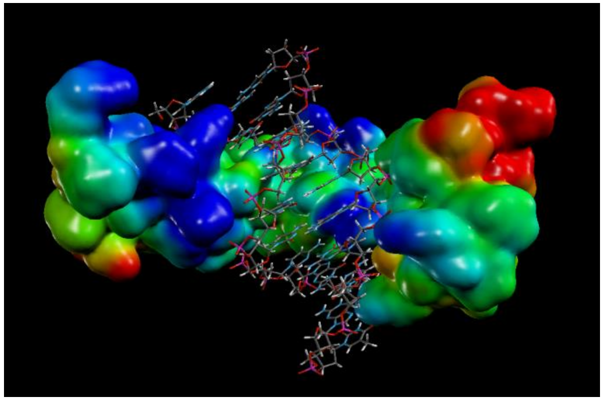


Scientific Visualization for Discovery

- Scientific data representation
- Picture vs. stream of numbers
- CG Techniques for visualization → contour plots, color en-coding, constant value surface rendering, custom shapes…



Display of a 2D slice through the total electron density of C-60; Created by Cary Sandvig of SGI

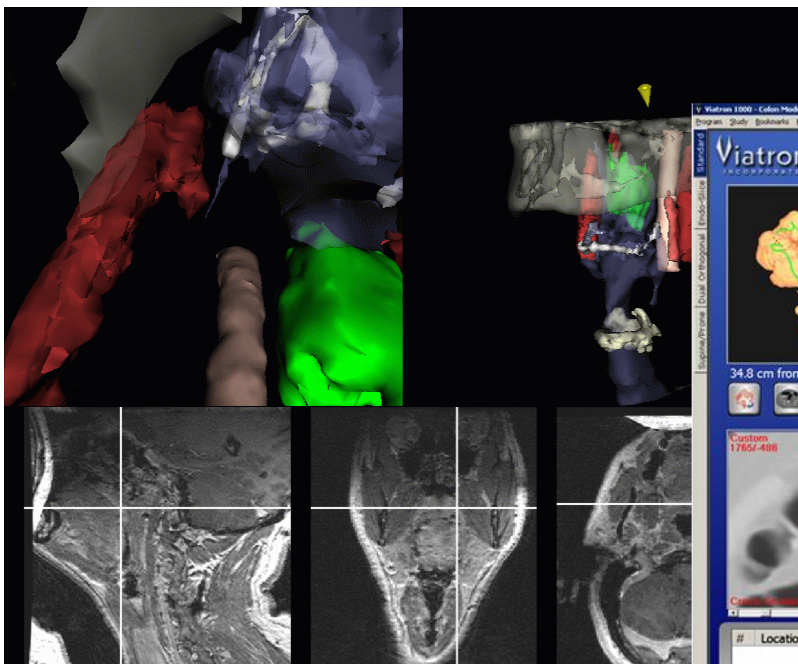


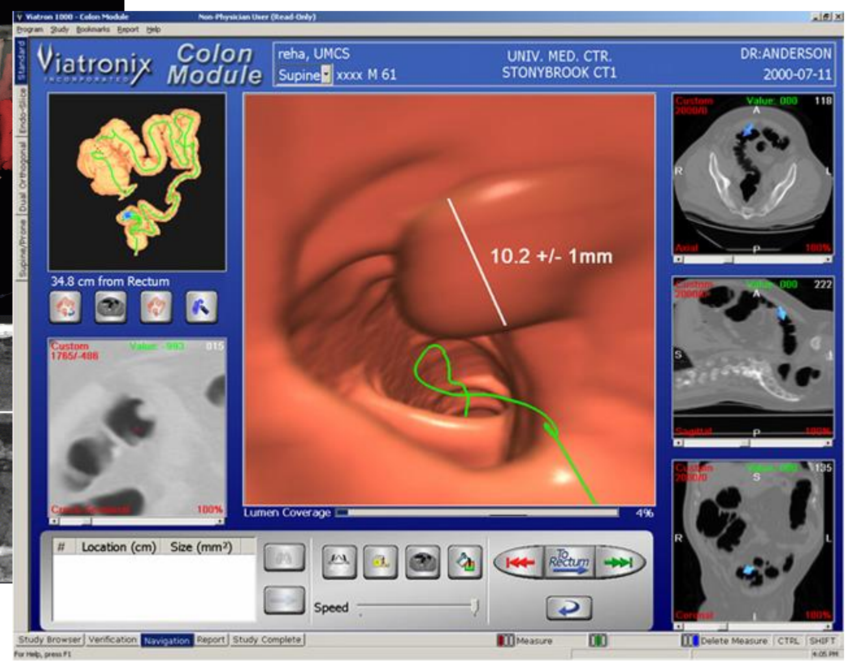
Predicted molecular structures

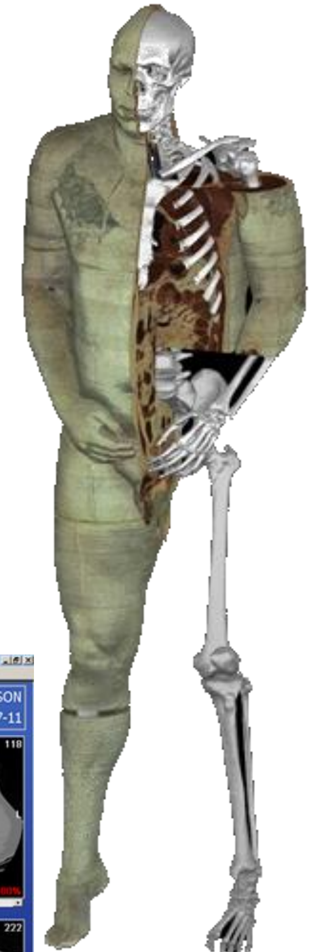
Social Media Visualization

Computer-assisted Medicine

- Aid in clinical analysis/diagnosis
- Virtual medical training and educations

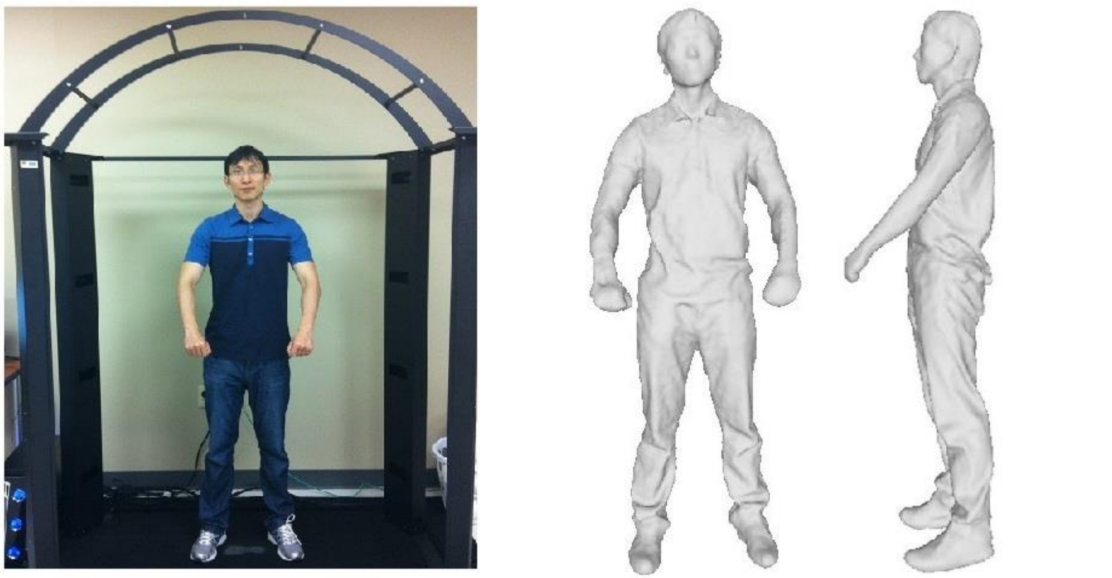






Computer-assisted Medicine

3D Body Scanning and Posture/Motion Analysis



Real-time Human Body Scan　　　Reconstructed Digital Model

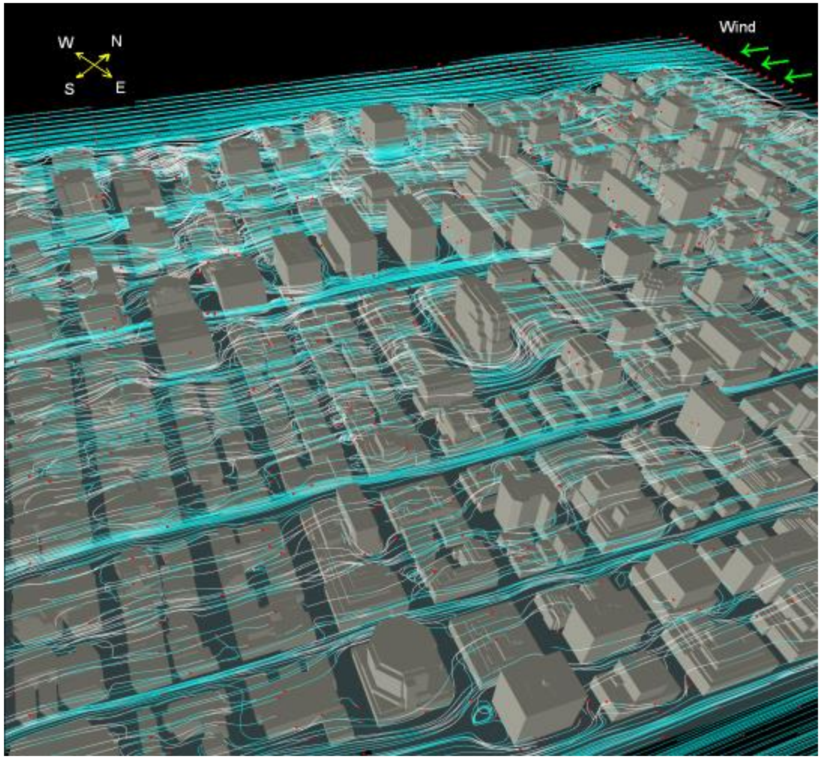
Scanning system developed by LSU GVC group and used in Baton Rouge Pennington Biomedical Research Center.

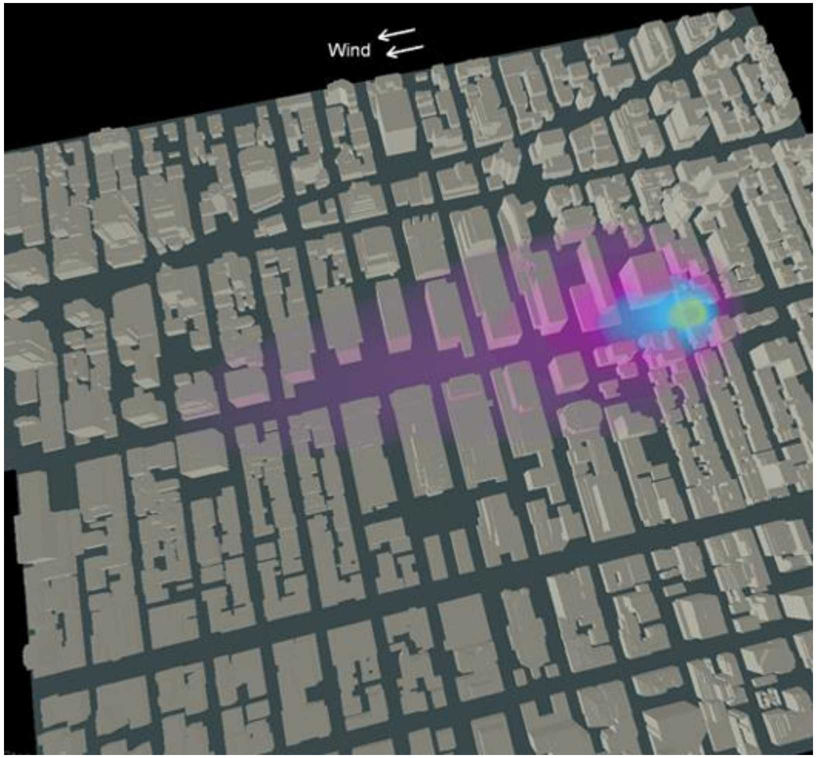
Digital body shape measurement for diabetes disease monitoring and analysis.

Navigation, Urban Security…

Google Earth

Navigation, Urban Security…

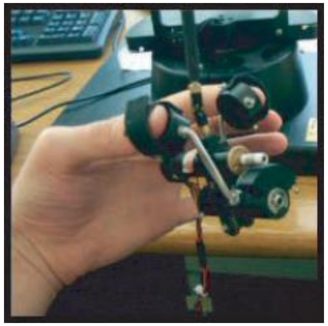




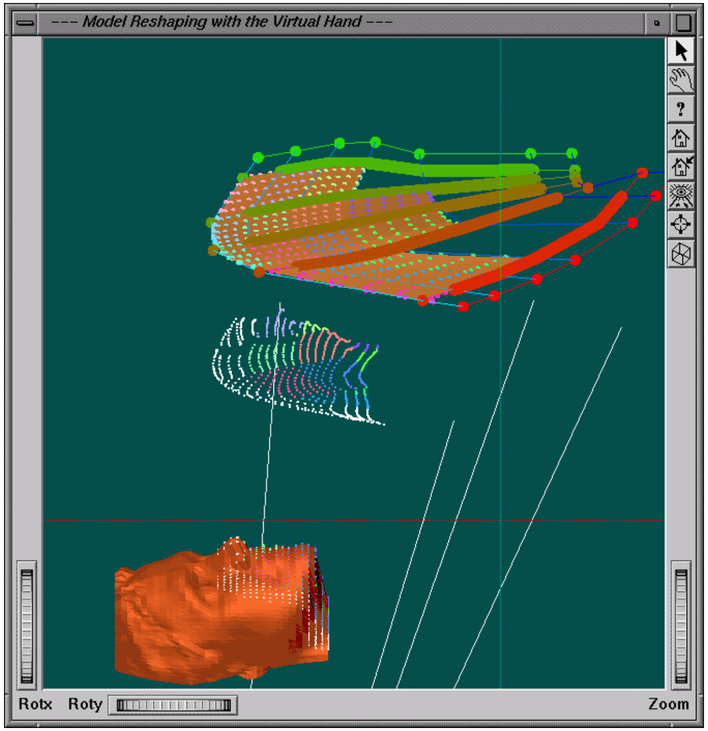
Simulate/visualize/predict the propagation of airborne contaminants in virtual Manhattan

Virtual/Mixed Reality

- Computer-aided Virtual Environment (CAVE)

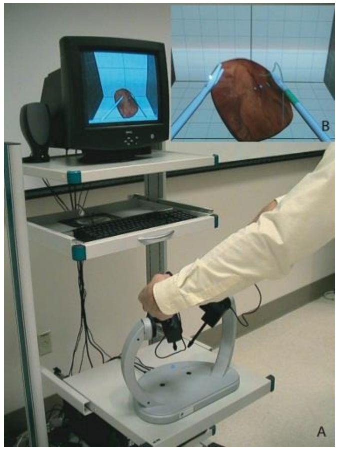




Virtual/Mixed Reality

- CAVE, Interactive modeling
- Virtual walkthroughs (training pilots, surgeons…)

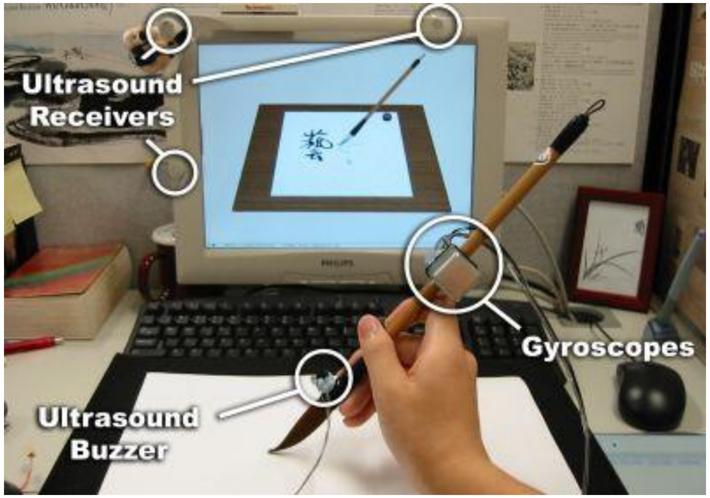


Textile/Cosmetics Industry

- Virtual fashion design
- Real-time cloth animation
- Web-based virtual try-on applications

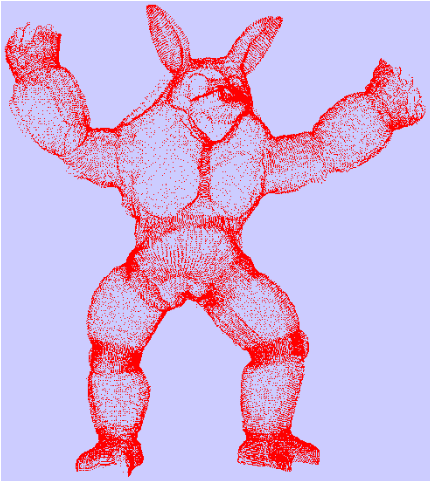
Computer Art

- Digital Painting, Digital Sculpting, Digital Calligraphy



And more …

3D Graphics Pipeline



3D Model Acquisition → Geometric Modeling and Processing → Animation, Rendering, Visualization

Reconstruction from Images, Range images, or Point clouds

Digital geom. processing

Surface and solid modeling

…

Ray tracing

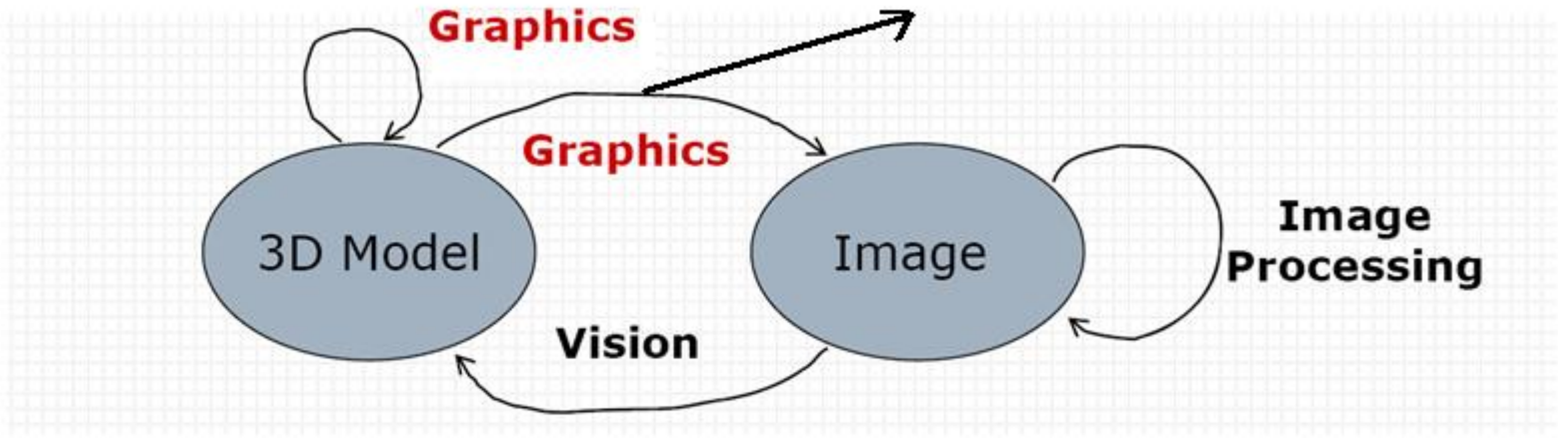
Texture synthesis

Physics-based simulation

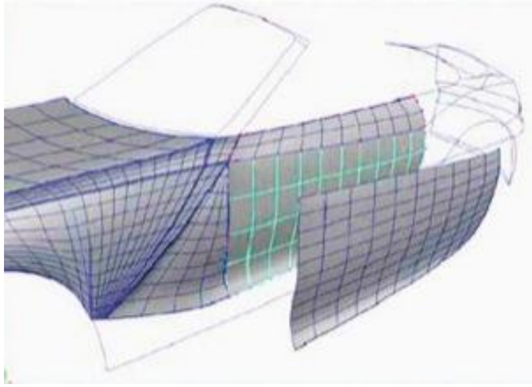
…

Graphics and Visual Computing

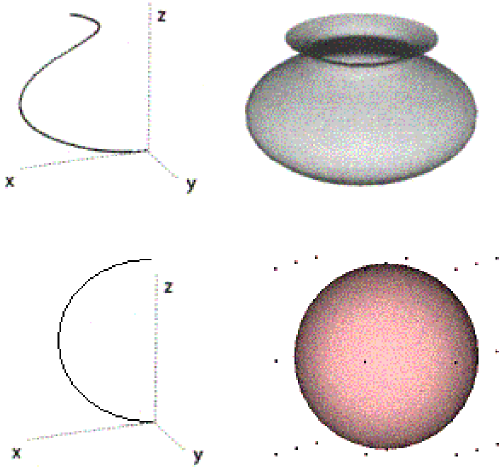
3D Display:
http://www.youtube.com/watch?v=G10bzatpuFc&feature=related

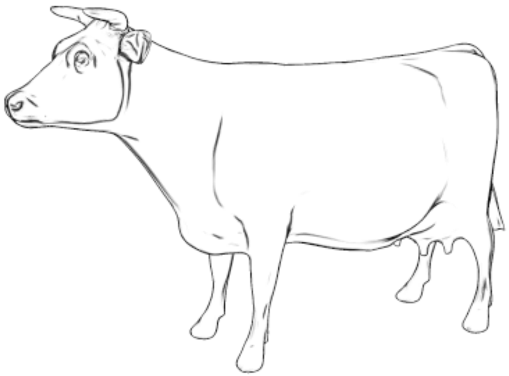


Acquiring 3D Digital Models



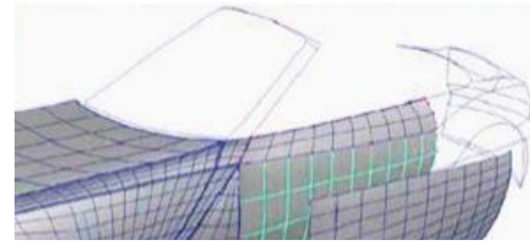
- Manual Design
  - interactive design and editing (AutoCad,3DMax)
  - sketch-based modeling
- Mathematical Description
- Scanning



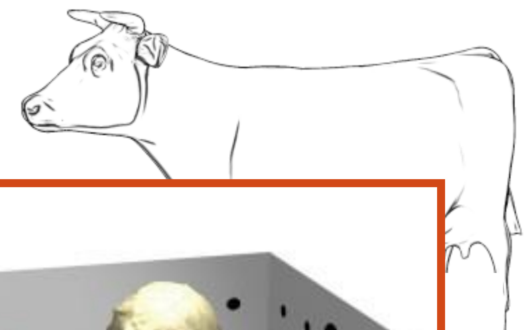


$$S(u,v) = \sum_{i=0}^{8}\sum_{j=0}^{m} R_{i,2;j,q}(u,v)P_{i,j}$$

Acquiring 3D Digital Models



- Manual Design
- Mathematical Description
- Scanning
  - By camera: image based reconstruction
  - By laser scanner: with depth



Representation of 3D Objects

A brief overview of common 3D representation techniques

- Polygonal Representation
- Space Subdivision Technique
- Spline Function
- Implicit Function Representation
- M-Representation (Medial Axis)
- Other Methods
  - Constructive Solid Geometry (CSG) Representation
  - GC (Generalized Cones, Generalized Cylinders)
  - Spatial Decomposition (Spherical harmonics, Zernike…)
  - Overlapping Spheres
  - …

Representation of 3D Objects

A brief overview of representation techniques we will go over this semester

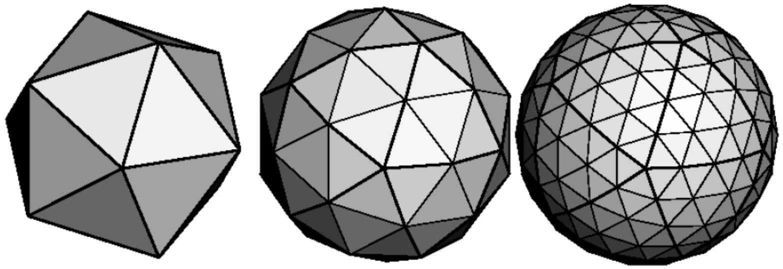
How do we choose a suitable representation?

- the nature of the object
- the particular geometric computation we need to apply
- the application

→ will explain later with details through each individual representation scheme and its suitable applications

Polygonal (Triangular) Mesh

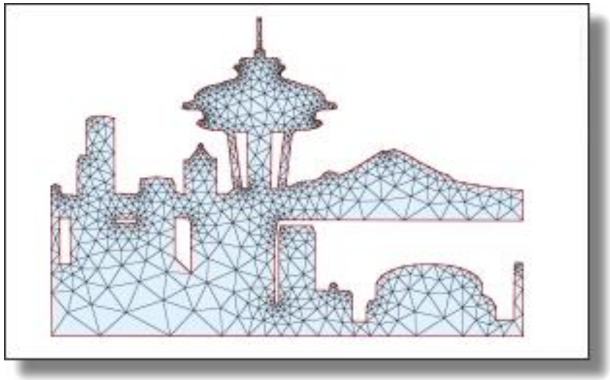
- Objects ← a mesh of planar triangular (polygonal) facets
  - can represent an object to different accuracy levels



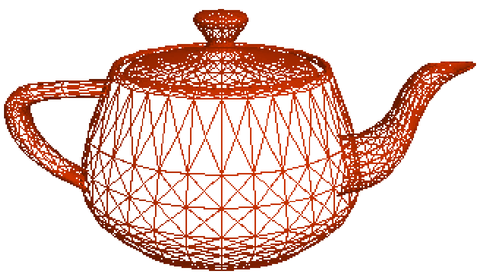
- Pro: A ubiquitous representation in Computer Graphics
  - Easy to generate and process
  - With effective algorithm for rendering (machine-oriented rep.)
  - Other rep. (CSG, splines, voxels…) → mesh for rendering
- Con: accuracy, often unstructural
  - Faceted rep. VS curved surfaces: different continuity condition
  - Constructing methods matter → mesh quality

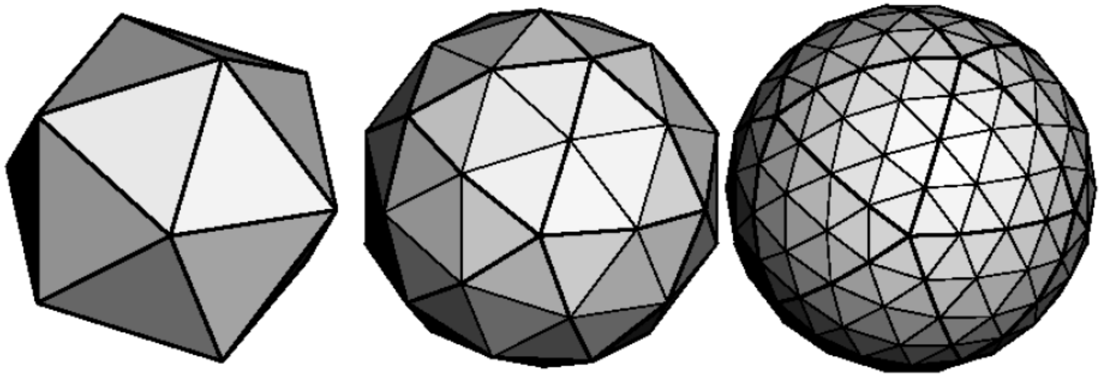
Triangular Mesh

- Surface shapes can be triangulated



Polygonal approximation of surfaces:



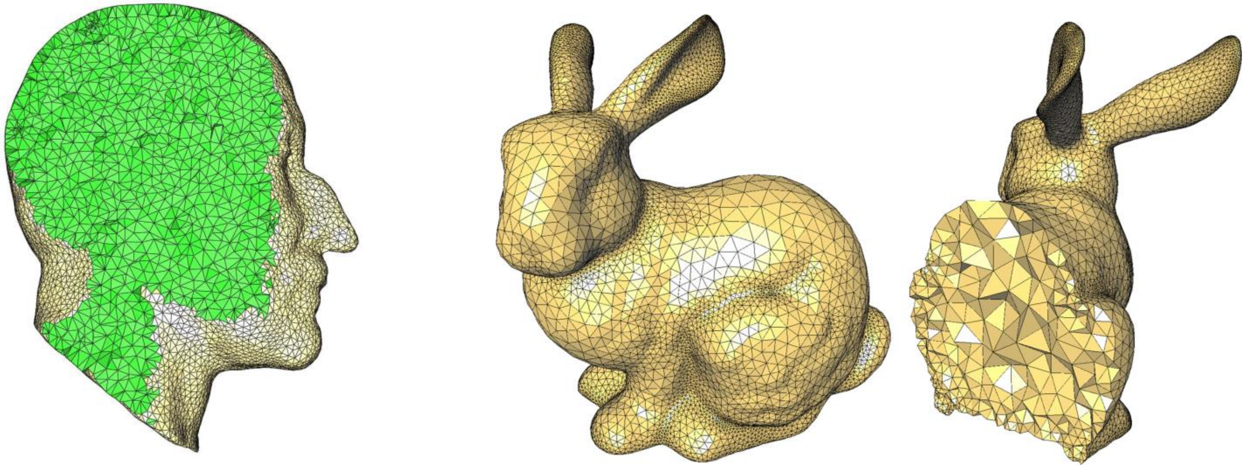


2D shape or 3D surface can be approximated with locally linear polygons. To improve accuracy (visual or numerical approximation quality), we only need to increase the number of edges

Tetrahedral Mesh

- Solid shapes can be tetrahedralized

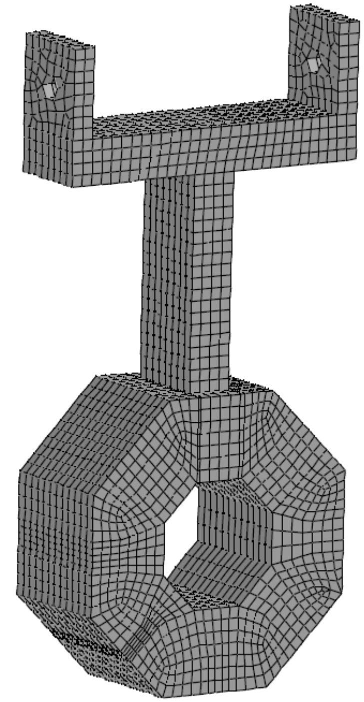
Polyhedra approximation of solid geometric data

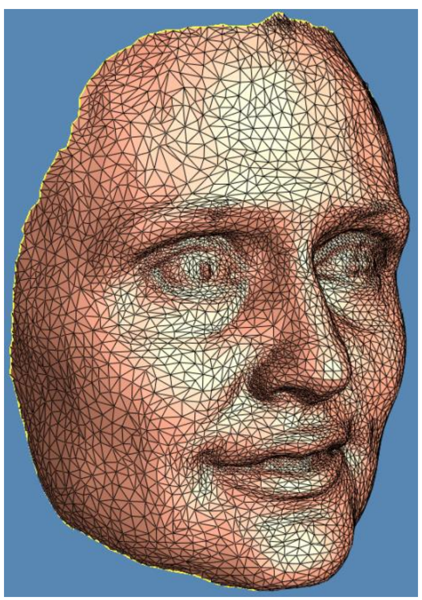


3D volumetric data (3-manifold) can be approximated with locally linear polyhedra. To improve accuracy (visual or numerical approximation quality), we only need to increase the number of edges

Polygonal Mesh

- Quad-Mesh
- Triangle Mesh
- A Mesh = {Vertex Positions,
  Connectivity,
  Additional Attributes}





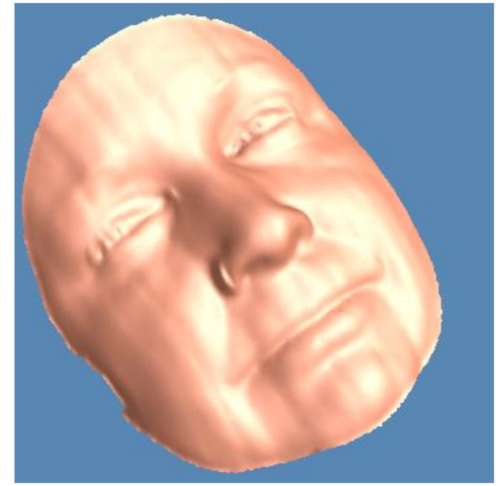
```
Vertex 1 0.6036570072 0.4613159895 0.07038059831
Vertex 2 0.6024590135 0.4750890136 0.07134509832
Vertex 3 0.6083189845 0.4888899922 0.07735790312
Vertex 4 0.611634016 0.5039420128 0.08098520339
Vertex 5 0.6236299872 0.5097290277 0.09412530065
Vertex 6 0.633580029 0.5194600224 0.1063940004
Vertex 7 0.6350849867 0.5272089839 0.1108580008
Vertex 8 0.6459569931 0.5308039784 0.1247610003
Vertex 9 0.6456980109 0.5446619987 0.1324290037
Vertex 10 0.6566579938 0.5420470238 0.1465270072
Vertex 11 0.6629710197 0.5443329811 0.1586650014
Vertex 12 0.671701014 0.541383028 0.1747259945
Vertex 13 0.6746420264 0.5451539755 0.1851660013
Vertex 14 0.6825680137 0.5424500108 0.206724003
Vertex 15 0.6884790063 0.5414119959 0.2314359993
Vertex 16 0.6935830116 0.5439419746 0.2590880096
Vertex 17 0.6981750131 0.5425440073 0.2817029953
Vertex 18 0.7026360035 0.5316519737 0.2960689962
Vertex 19 0.7058500051 0.5267260075 0.3085480034
Vertex 20 0.7095490098 0.5337790251 0.3253619969
Vertex 21 0.7104460001 0.5344949961 0.3296009898
Vertex 22 0.7158439755 0.5286110044 0.3463560045
Vertex 23 0.7237830162 0.5144050121 0.3689010143
Vertex 24 0.7282400131 0.5028949976 0.3827379942
```

```
Face 1 63 3 4
Face 2 64 63 4
Face 3 5 64 4
Face 4 65 5 6
Face 5 7 65 6
Face 6 8 65 7
Face 7 9 66 8
Face 8 10 66 9
Face 9 67 66 10
Face 10 11 67 10
Face 11 12 67 11
Face 12 14 75 13
Face 13 68 76 15
Face 14 16 68 15
Face 15 17 68 16
```

Polygonal Mesh

- Quad-Mesh
- Triangle Mesh
- A Mesh = {Vertex Positions,
  Connectivity,
  Additional Attributes}

Vertex Normal, Edge length, face area, any scalar/vector fields…



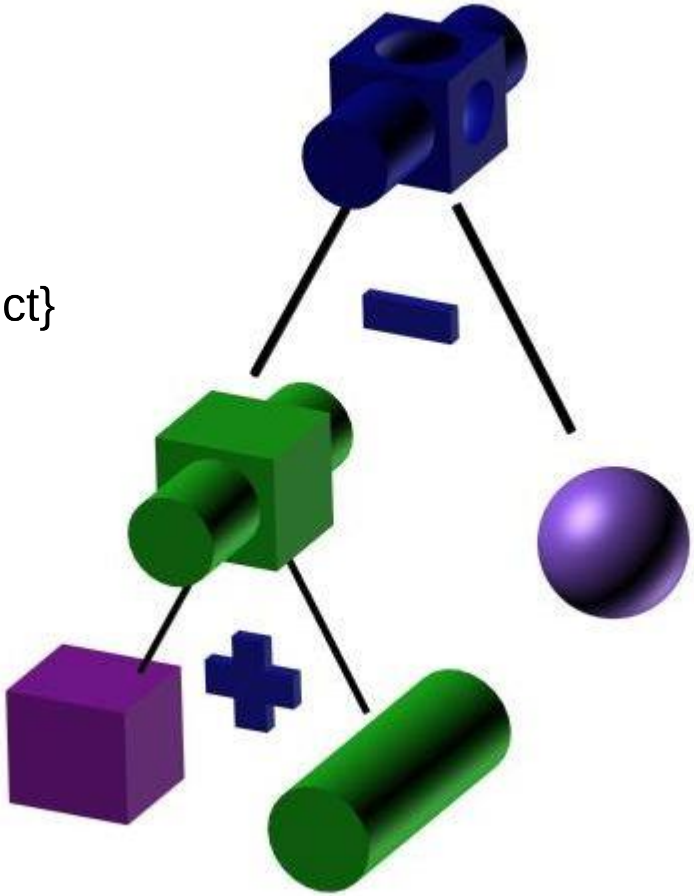
```
Vertex 1  123.472 75.6855 171.207 {rgb=(0.0207186 0.0137227 0.0205335) normal=(                   )}
Vertex 2  129.905 75.6904 169.427 {rgb=(0.0899862 0.0721164 0.0482489) normal=(                   4)}
Vertex 3  135.957 75.6998 168.927 {rgb=(0.117921 0.0953541 0.0583396) normal=(-                   )}
Vertex 4  138.285 75.7013 168.438 {rgb=(0.110971 0.0836528 0.0614068) normal=(-
Vertex 5  140.444 75.6976 166.931 {rgb=(0.102124 0.0731135 0.0495221) normal=(-                   }
Vertex 6  123.505 76.1939 169.629 {rgb=(0.0622525 0.0450163 0.0267677) normal=(                   )}
Vertex 7  125.371 76.192 169.316 {rgb=(0.172941 0.14031 0.111185) normal=(-0.07
Vertex 8  127.986 76.192 168.729 {rgb=(0.233185 0.19088 0.142915) normal=(-0.11
Vertex 9  131.737 76.2069 168.147 {rgb=(0.23693 0.191725 0.141712) normal=(-0.0507257 -0.894431 -0.44432)}
Vertex 10  136.328 76.1993 167.518 {rgb=(0.249965 0.209907 0.160202) normal=(-0.098178 -0.914369 -0.392798)}
Vertex 11  140.936 76.2291 165.272 {rgb=(0.243799 0.201224 0.151788) normal=(-0.233659 -0.915351 -0.327925)}
Vertex 12  142.15 76.1638 164.365 {rgb=(0.213539 0.175771 0.135716) normal=(-0.192717 -0.928922 -0.316173)}
Vertex 13  145.563 76.1924 162.923 {rgb=(0.234091 0.189093 0.142723) normal=(-0.0974924 -0.936706 -0.336269)}
Vertex 14  150.893 76.1359 162.13 {rgb=(0.233473 0.189348 0.145252) normal=(-0.0397114 -0.933055 -0.357534)}
Vertex 15  151.397 76.1899 162.135 {rgb=(0.170212 0.132446 0.0934432) normal=(-0.0345978 -0.9314 -0.36235)}
Vertex 16  152.895 76.2002 161.741 {rgb=(0.216202 0.174615 0.141327) normal=(-0.160623 -0.883519 -0.439993)}
```

CSG Representation

- Polygonal Mesh → machine-oriented representation
- CSG → user-oriented representation
  - store the “logic of the shape”
- A CSG modeling system

  = {building blocks, Boolean operations}

  {union, subtract, intersect}



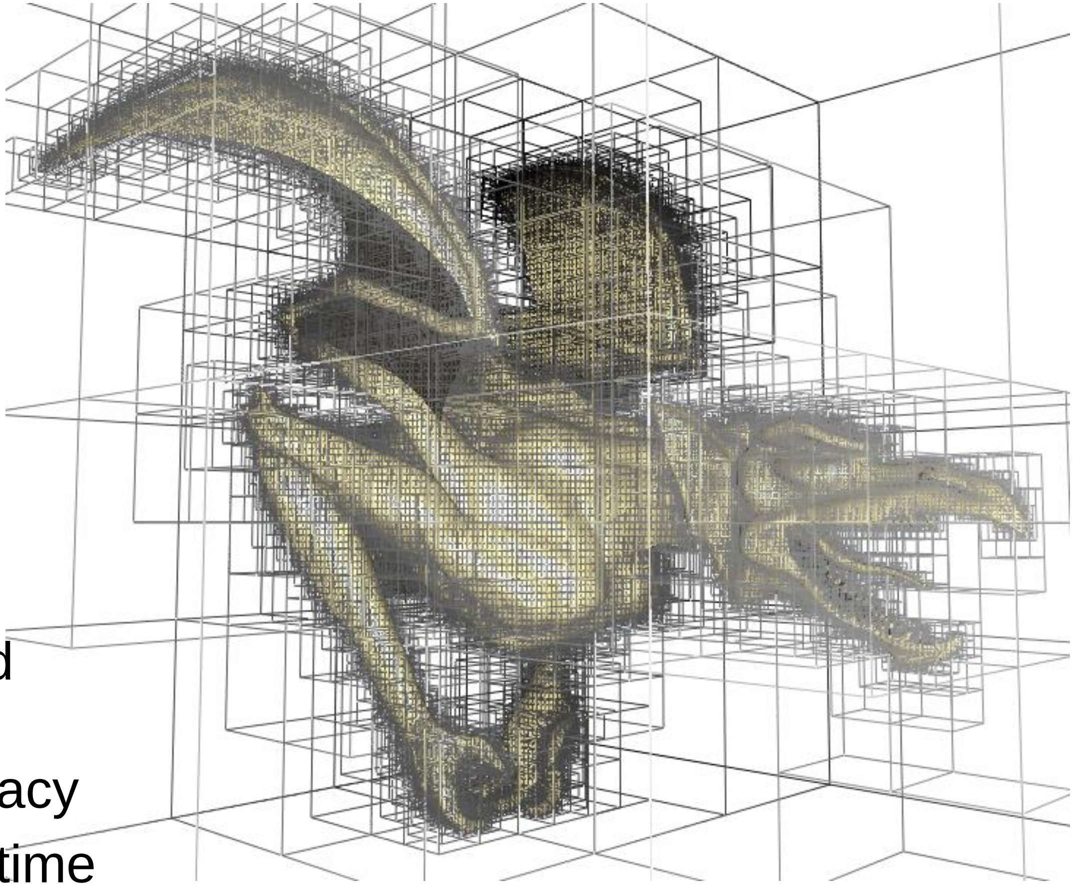
Widely used in 3DMax, Maya… as their modeling scheme:

❑Support user-intervention

❑Good for simple shapes

Space Subdivision Representation

- Not explicitly represents the geometric object
- But consider the space the object occupy



an octree rep.

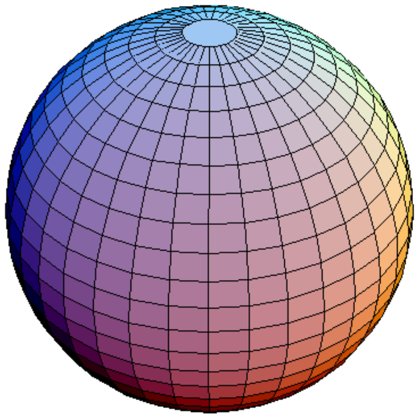
= a hierarchical tree built by sequential subdivision of occupied cells

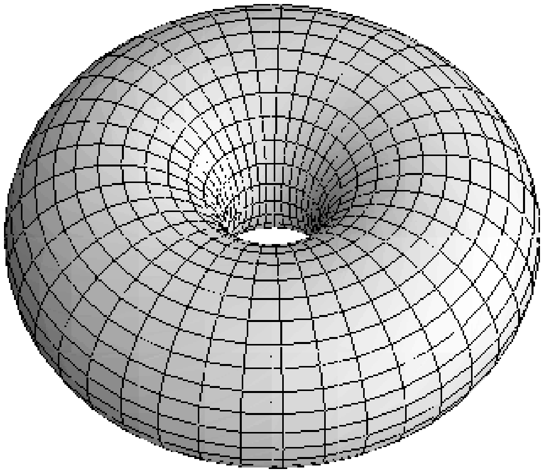
- Widely used for complicated scenes that need faster processing and lower accuracy

e.g. Collision detection in realtime simulation or animation

Implicit Representation

- Usually Compact
- Good for modeling shapes with closed-form expression
- Good for processing with topological changes
  - Simulation
  - Reconstruction (Hole-filling)
  - …

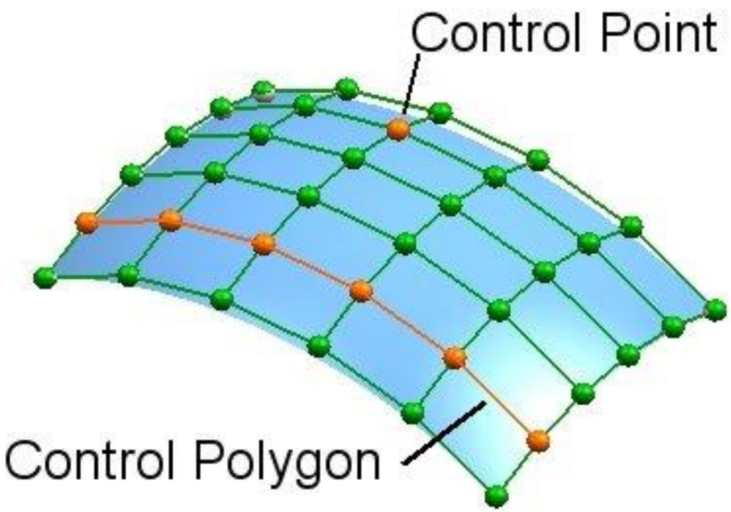


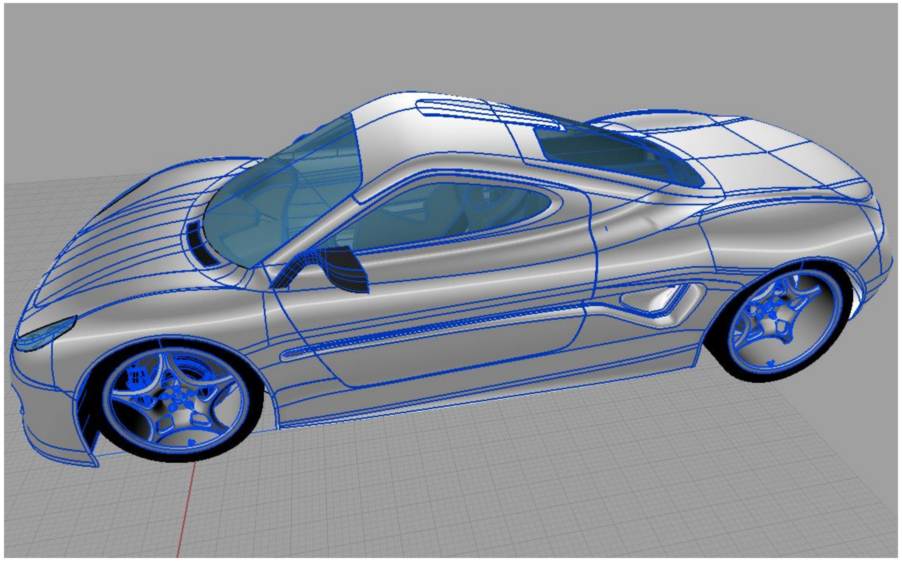


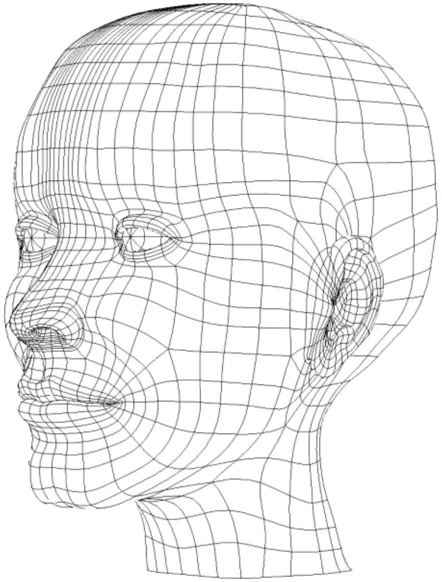
Spline

- Exact analytical rep.
- Support interactive shape editing
- Compact rep.

- Major modeling techniques in CAD



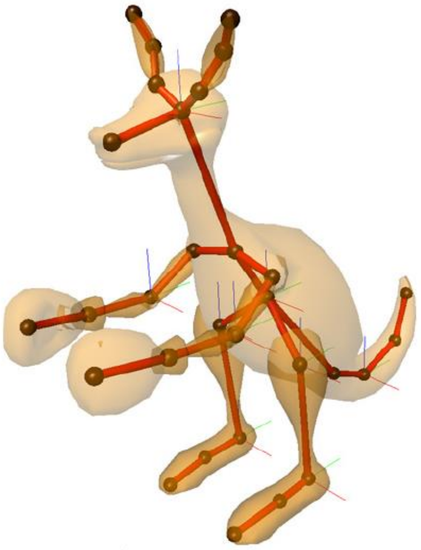




M-Representation

- Popular for animation, used in Maya, 3DMax
- Models
  - may still be represented as triangle meshes ,
  - but their movement driven by M-rep
- Consider vertices on the boundary surface (skin), their deformations are induced by the deformation of skeleton

  - A skeleton is a 1-D graph, each node represents a joint
  - The deformation of the skeleton → the transformation of joints



Resources

Reference books (optional) :

1. OpenGL Programming Guide (the Red Book)
   http://www.glprogramming.com/red/
2. 3D Computer Graphics
   by Alan Watt. Addison-Wesley.
3. Computer Graphics: Principles and Practice
   by James Foley, Andries van Dam, Steven Feiner, John Hughes. Addison-Wesley.

To do research in CG:

What math is important for Computer Graphics? (by Greg Turk)