EE 7000-1
3D Graphics and Visual Computing

Instructor: Xin (Shane) Li
Email: ece7000.lsu@gmail.com

Lectures: M W 3:40pm – 5:00pm
145 EE Building

Office Hours: M W 1:00pm – 3:30pm
313 Electrical Engineering Building
Louisiana State University

Website:
http://www.ece.lsu.edu/xinli/teaching/EE7000Fall2011.htm

Course Synopsis

- What is it about?
  - An advanced graduate CS/CE course
  - Concepts, algorithms, techniques, tools in 3D Geometric Computing for Computer Graphics and Vision
  - Research oriented: ~~exams~~, projects

- Topics:
  - Graphics pipeline overview
  - Basic OpenGL and Graphics programming
  - Analyzing and Processing 3D Shapes
  - Applications of Discussed Technologies in Graphics, Visions …

Workload

- A fun course:
  - You will learn algorithms and methodologies for manipulation 3D objects,
  - Do C/C++ programming
  - Develop your interactive Graphics User Interface (GUI)
- Need substantial reading and programming work
  - 3 homework projects
  - 1 final project + presentation
- Start early on your homework and projects
- Grading is generous

Grading

- Attendance (10%)
- Final Project (40%)
  - Demo + Presentation (25%)
  - Codes + Report (15%)
- Homework (50%)
  - Warm-up (8%)
  - Homework 1 (14%)
  - Homework 2 (14%)
  - Homework 3 (14%)
- A: >75; B: >55; C: >40

Prerequisites

- Understand basic linear algebra, calculus, data structure
- Be familiar with C++ programming (for most assignments, starter codes and solutions will be provided in C++)
- Self-learning:
  - We will cover OpenGL a little bit but you will learn them mostly by yourself
  - You will team-up to read and implement a paper for your final project

Questionnaire

(Part of the warm-up assignment: 4%)

1) List your background courses/knowledge related to graphics/visualization. Have you done any related projects?
2) Are you familiar with C/C++ and object-oriented programming? What projects have you done? How many lines of codes have you written?
3) What is your main goal for taking this course (e.g. to learn the knowledge, pursue a career in this area, or related to your current research …)

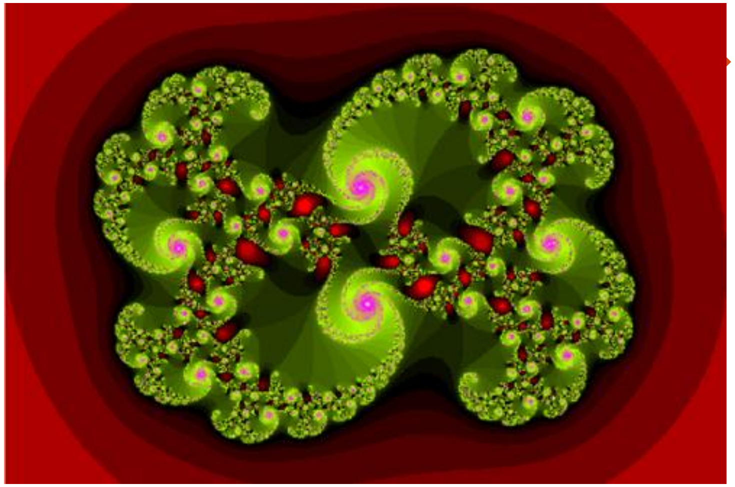
What is Computer Graphics?

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.

- Chinese Proverb



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?

1. For Visualization:

- Enable people to describe, share, and summarize their datasets (models)
- We are effective in processing images: about 50% of the brain neurons are associated with vision

2. But more:

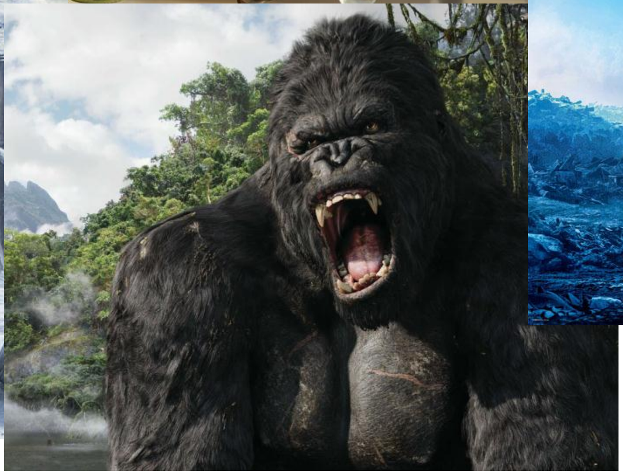
Recent innovation in 3D acquisition technology has enabled highly accurate digitization of complex 3D objects!

- Analysis, Processing, and Simulation: Numerous scientific disciplines, (e.g. medicine, neuroscience, mechanical engineering, and astrophysics) rely on the analysis and processing of such geometric data to understand intricate geometric structures and facilitate new scientific discoveries.
- Design and Manufacturing: We are experiencing a revolution in digital manufacturing technology. Novel materials and robotic production will soon allow the automated creation of complex physical artifacts from a digital design plan!

studied and widely used in many applications…

Movies

- CG has been changing Special Effects in Movie Industrial (Billions of dollars spent)
- Need to be realistic and physically natural (simulation of objects, motion, and natural phenomena…)


Video Games

- Important driving force
- Focus on **interactivity**
- Try to avoid computation and use various CG tricks (simplification, texture mapping…)

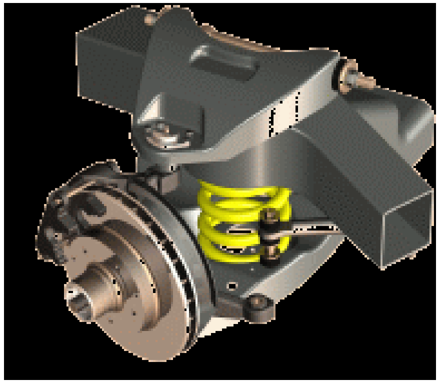
Age of Empire II

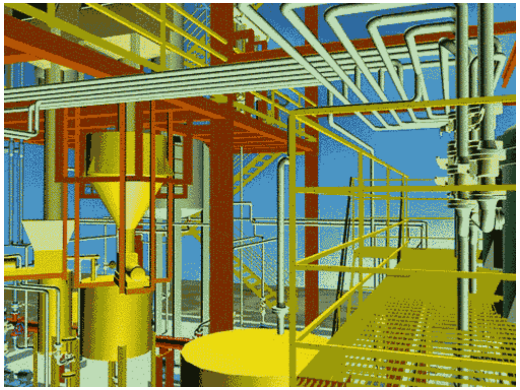
Quake IV

Computer Aided Design/Manufacturing

Significant impact on the design process:

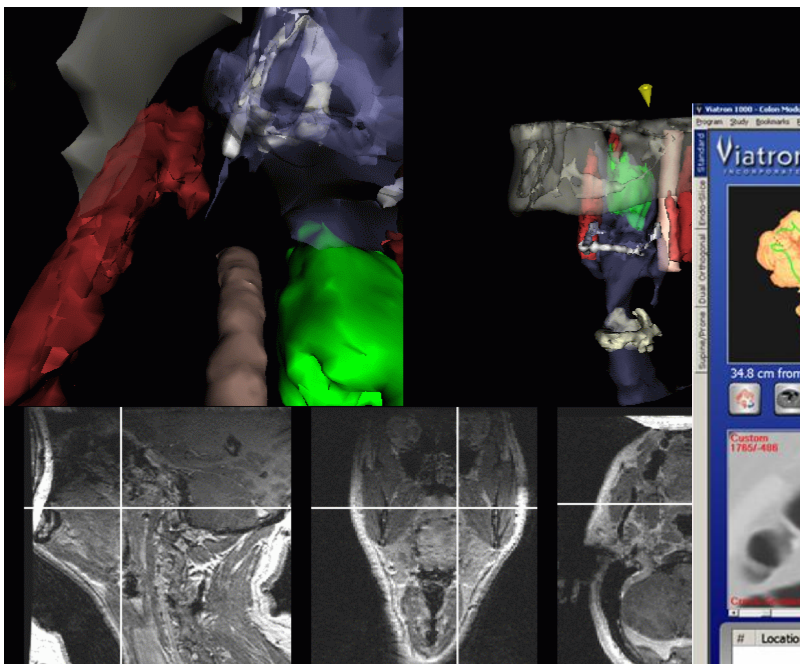
- Mechanical, electronic design (entirely on computer environments)
- Architectural and product design (migrate to computer environments)

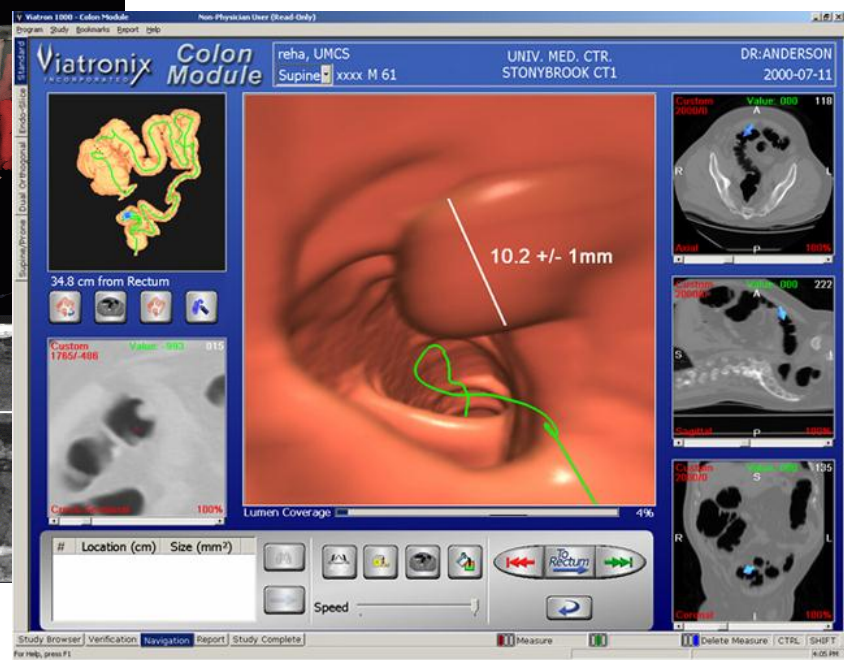




Medical Applications

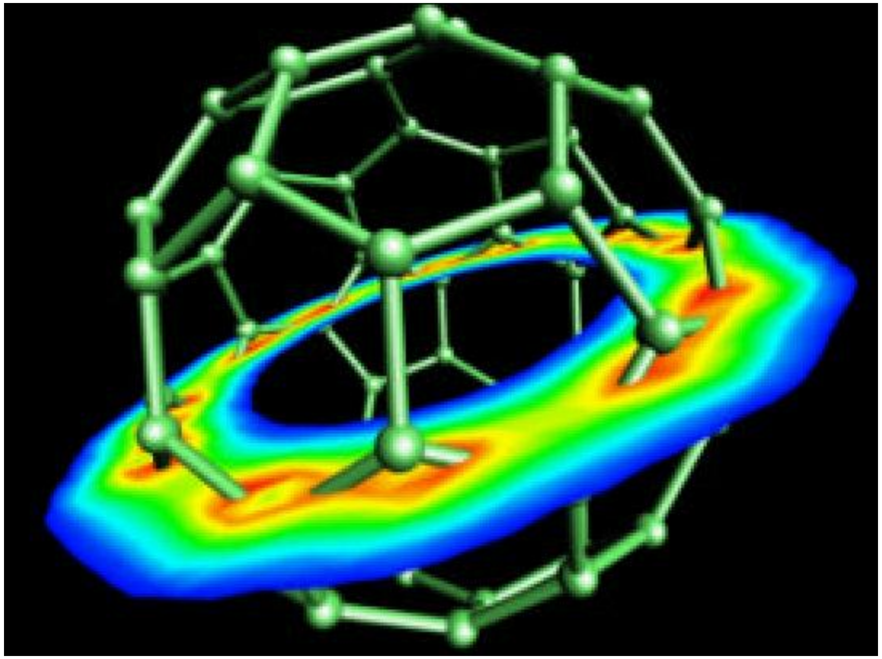
- Aid in clinical analysis/diagnosis
- Virtual medical training and educations





Scientific Visualization

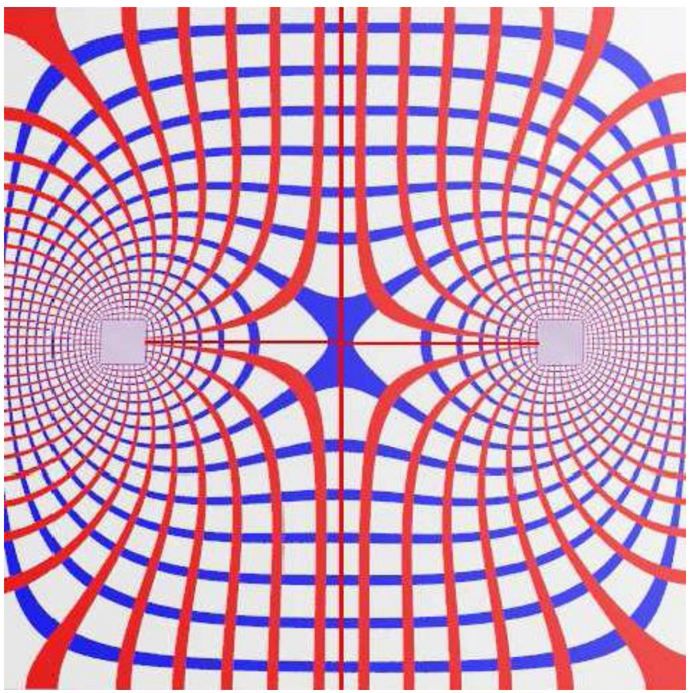
- Scientific data representation
- Picture vs. stream of numbers
- CG Techniques: contour plots, color coding, constant value surface rendering, custom shapes



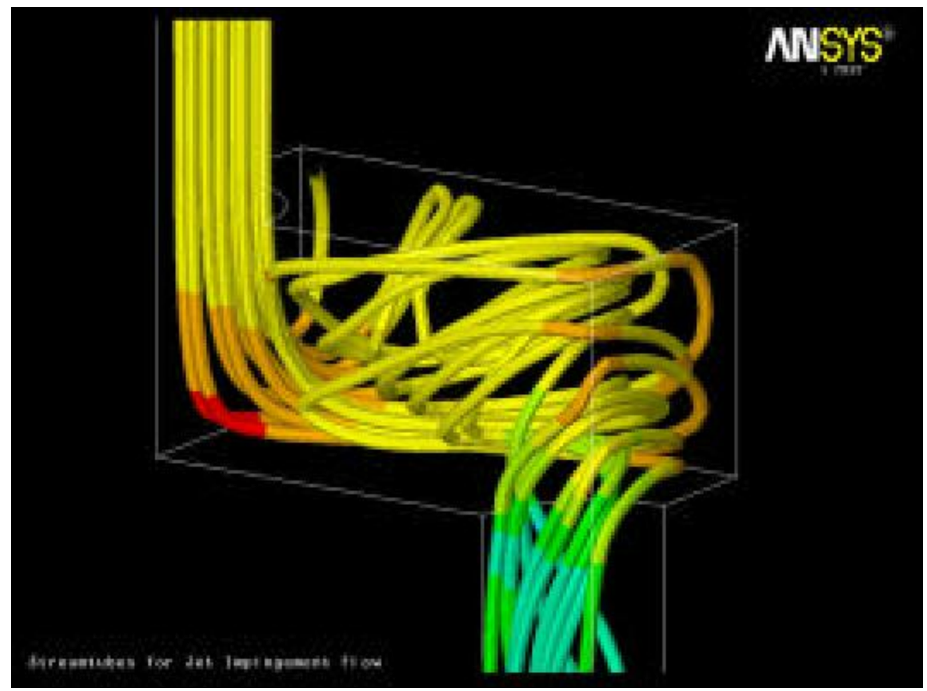
Display of a 2D slice through the total electron density of C-60; Created by Cary Sandvig of SGI

Scientific Simulation

Electromagnetic potential field



Computational Fluid Dynamics (CFD)



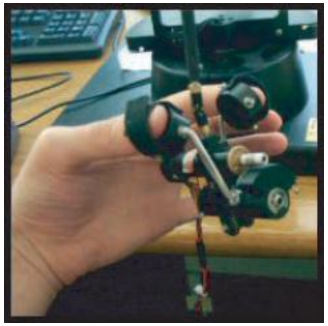
Courtesy of Mark Toscinski and Paul Tallon

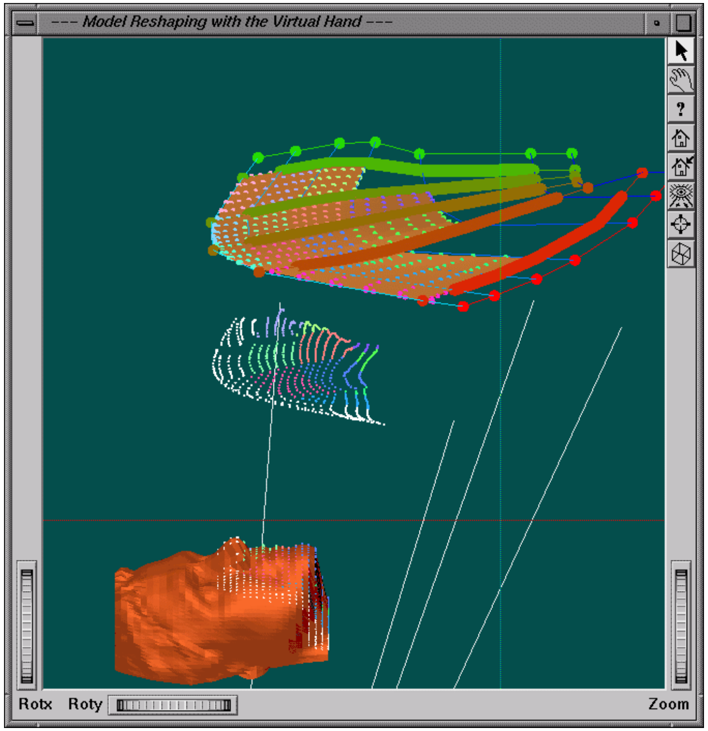
Navigations, Urban Security…

Google Earth

Virtual Reality

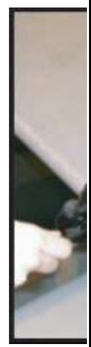
- CAVE, Interactive modeling

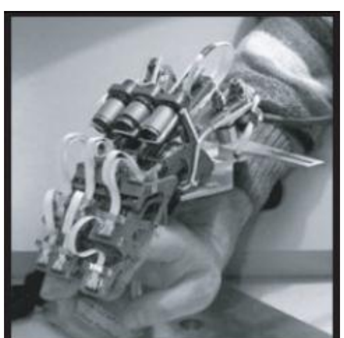


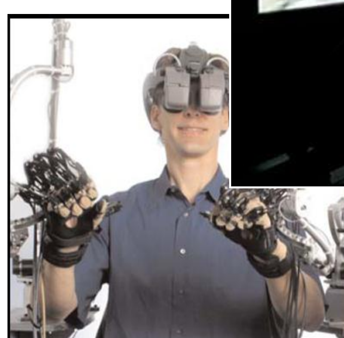




Virtual Reality

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





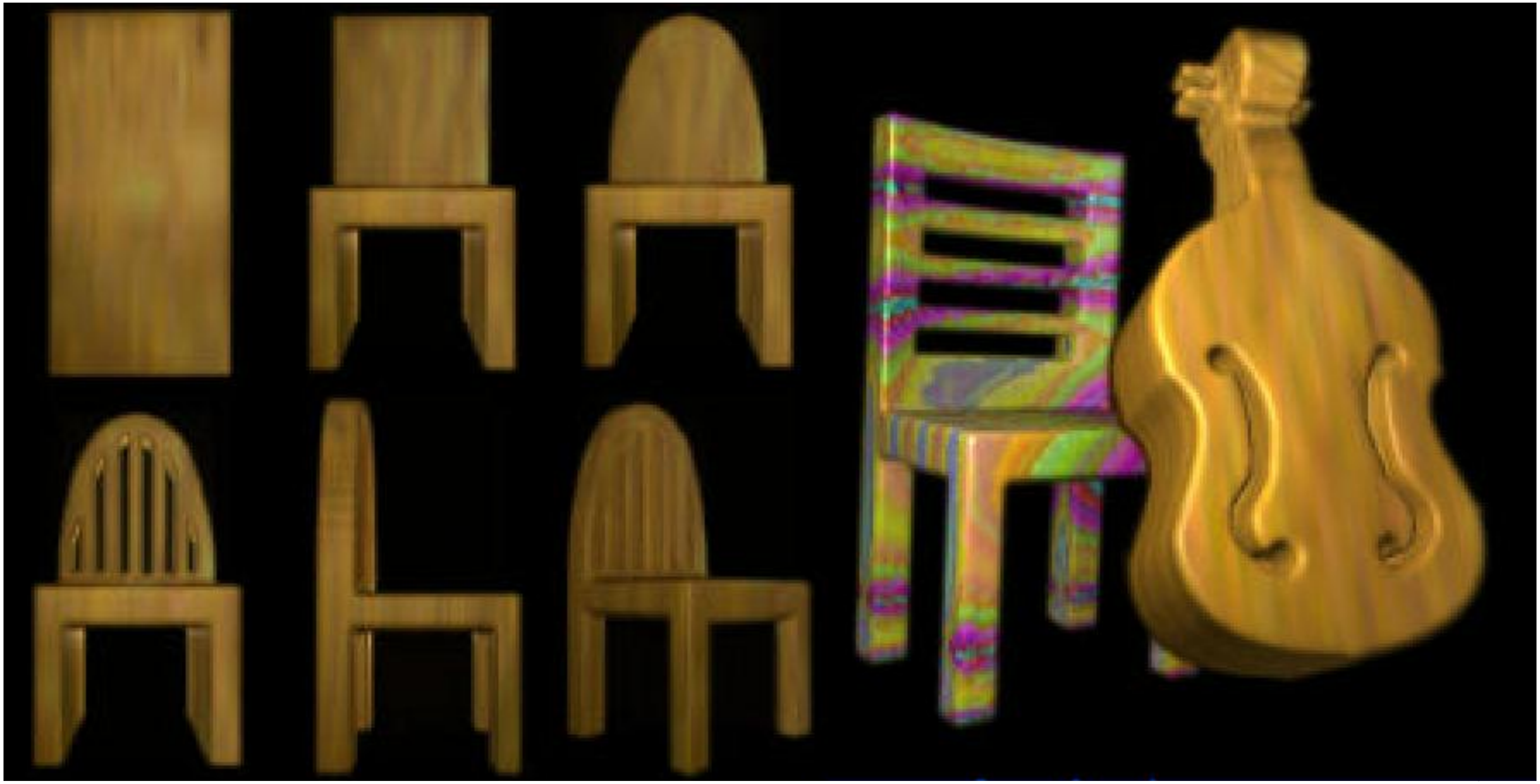


Textile/Cosmetics Industry

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

Computer Art

- Digital Sculpting

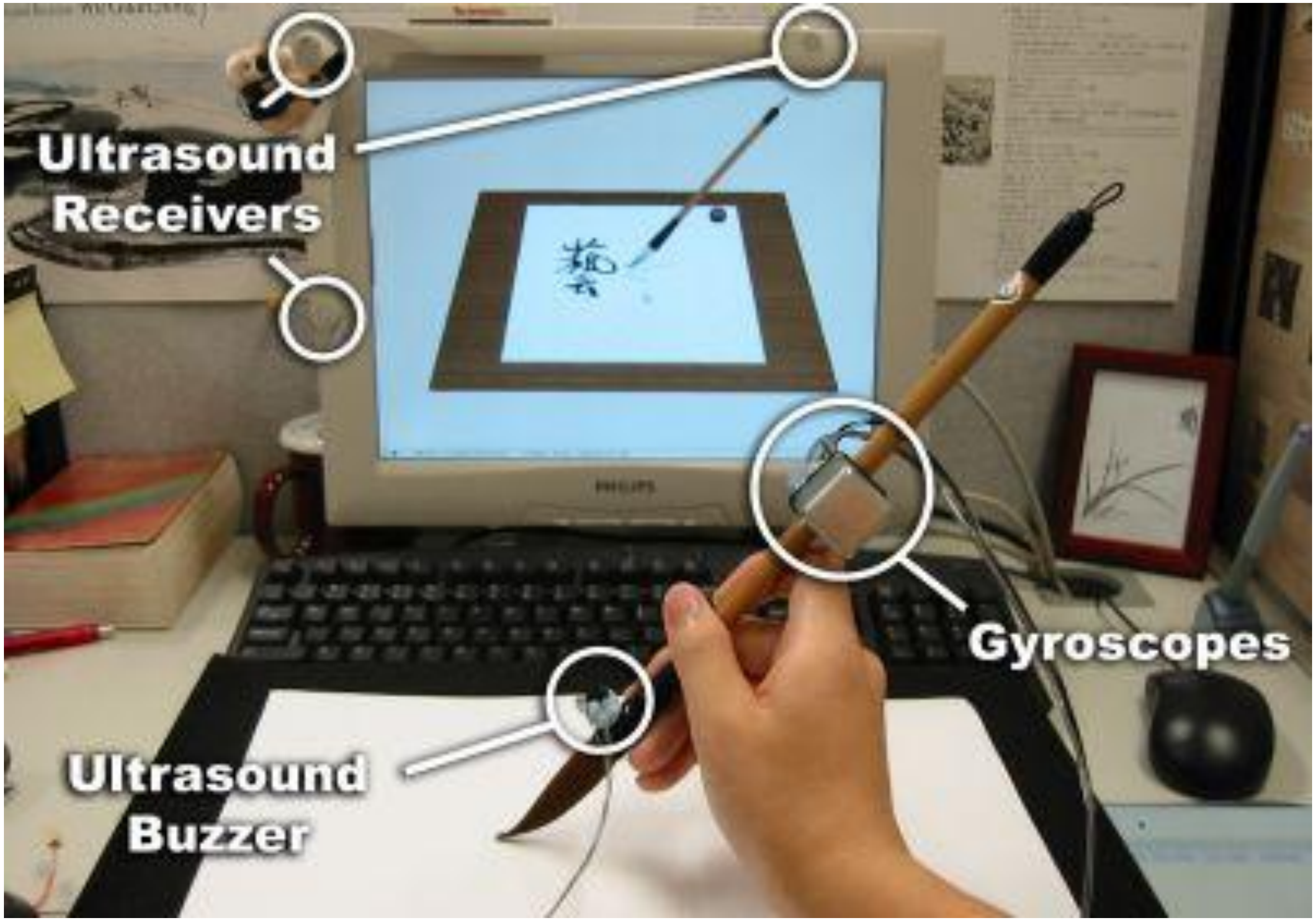


Computer Art

- Digital Sculpting, Digital Painting

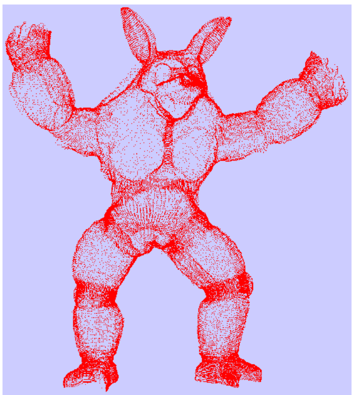
Computer Art

- Digital Sculpting, Digital Painting, Digital Calligraphy



And more applications…

3D Graphics Pipeline



3D Model Acquisition → Geometric Modeling and Processing → Animation & Rendering

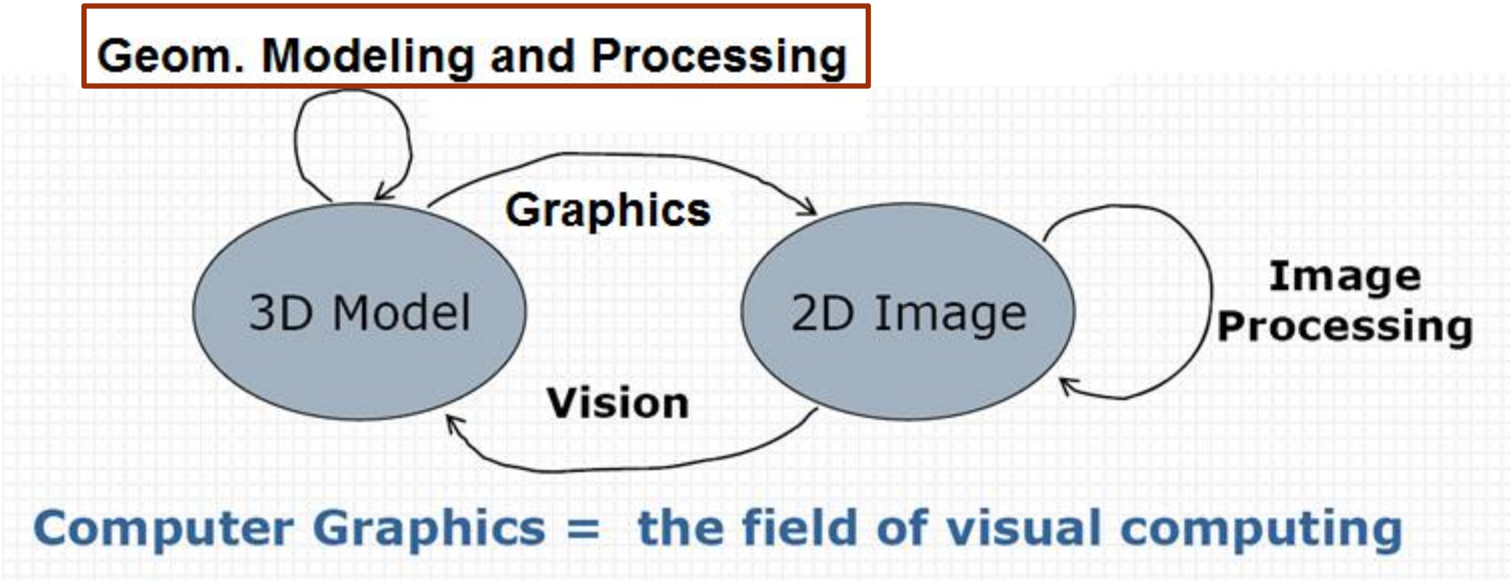
Image Reconstruction

Model Repair

Digital Geom. Processing,
Shape modeling
…

Ray tracing,
Texture synthesis,
Appearance modeling
…

Geometry Shapes vs Images



3D Digital Model Acquisition

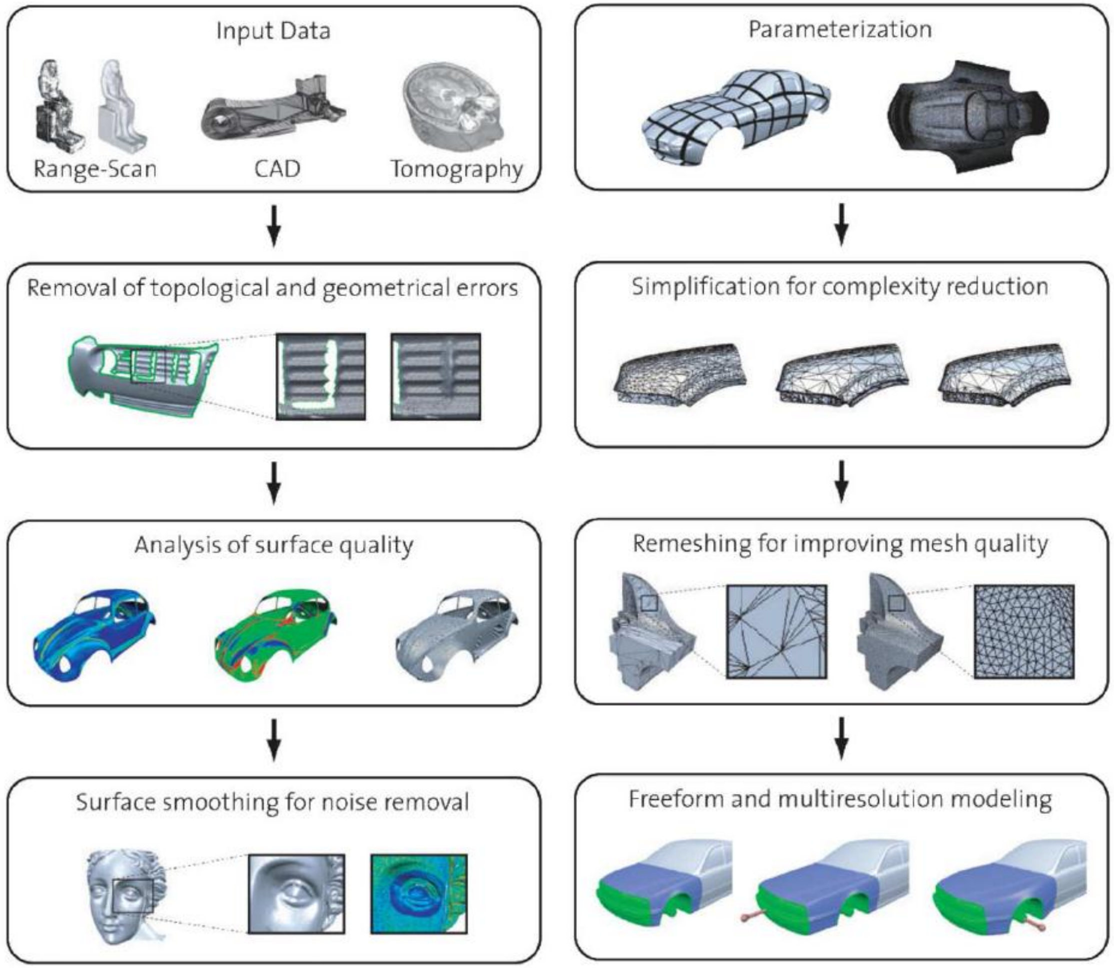
- Direct 3D sensing technologies
  - Computer tomography
  - Magnetic resonance imaging
  - 3D laser scanning
  - Ultrasound
  - Radar
  - Microscopy
- Manual Constructions
  - CSG/CAD software: http://www.youtube.com/watch?v=nMxCsmj-heI&feature=related
  - Sketch based modeling: http://www.youtube.com/watch?v=e2H35SlLmUA
- Vision-based Reconstruction from Image Sources
  - From images, videos…

Geometric Modeling and Processing

- How to represent geometric data?
  - The natural of the objects
  - The intended geometric tasks/applications
- Geometric algorithms on 3D models

Geometric Models	Geometric Algorithms
Objects (2D, 3D, …)	Actions

→ Find appropriate representation
← Develop efficient operations



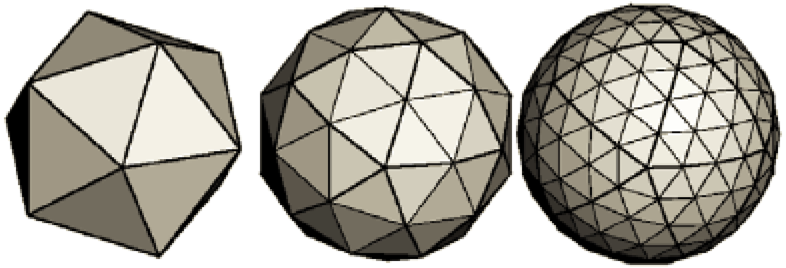
A typical geometric processing pipeline

Representation of 3D Objects

- Explicit (parametric) Representations
  - Polygonal (triangle, quadrilateral…) meshes
  - Splines, Subdivision Surfaces
  - Spatial decompositions
  - Medial axis representations …
- Implicit Representations
  - The zero set of a scalar-valued function $F: \mathbb{R}^3 \rightarrow \mathbb{R}$,
    i.e. a shape $S = \{\mathbf{x} \in \mathbb{R}^3 \mid F(\mathbf{x}) = 0\}$
- Strengths and Weaknesses of a representation:
  - Evaluation: important for rendering
    - e.g. computing the surface normal field
  - Query: important for shape modeling, analysis, simulation…
    - e.g. computing the distance from a point to a surface, checking inside/outside
  - Modification: important for simulation, deformation, animation
    - e.g. modifying the geometric shapes or topology

Polygonal Mesh

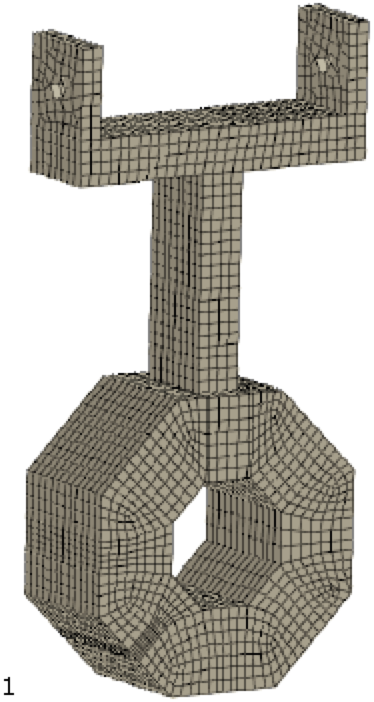
- Objects ← a net/mesh of planar polygonal facets
  - can represent an object to an accuracy that we choose

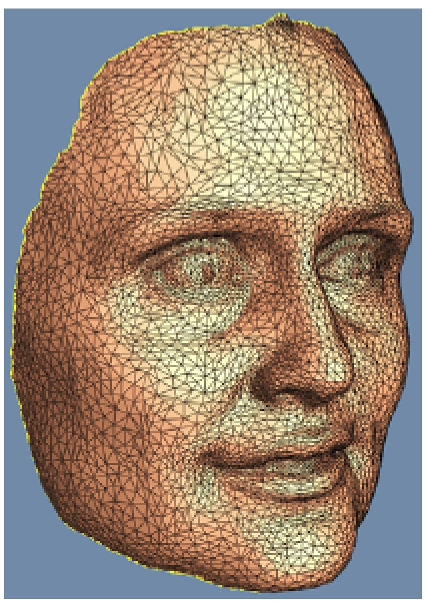


- Pro: A ubiquitous representation in Computer Graphics
  - Easy to generate and process
  - With effective algorithm for rendering (machine-oriented rep.)
  - Other rep. (CSG, splines, voxels…) → mesh before rendering)
- Con: accuracy
  - Faceted rep. VS curved surfaces : usually arbitrary
  - Constructing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

Polygonal Mesh

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





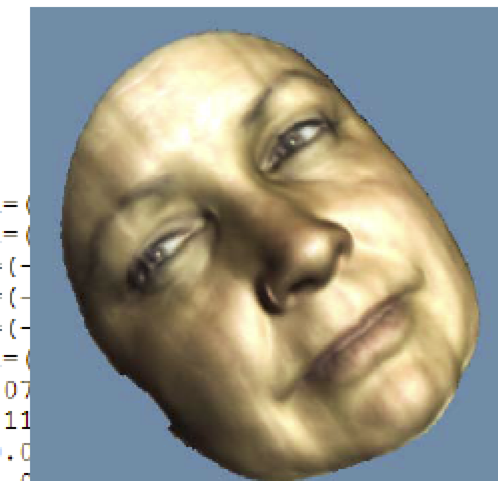
```
Vertex 1 0.6036570072 0.4613159895 0.07038059831
Vertex 2 0.6024590135 0.4750890136 0.07134509832
Vertex 3 0.6083189845 0.4888899922 0.07735790312
Vertex 4 0.611634016 0.5039420128 0.08098520339
Vertex 5 0.6236299872 0.5097290277 0.09412530065
Vertex 6 0.633580029 0.5194600224 0.1063940004
Vertex 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…



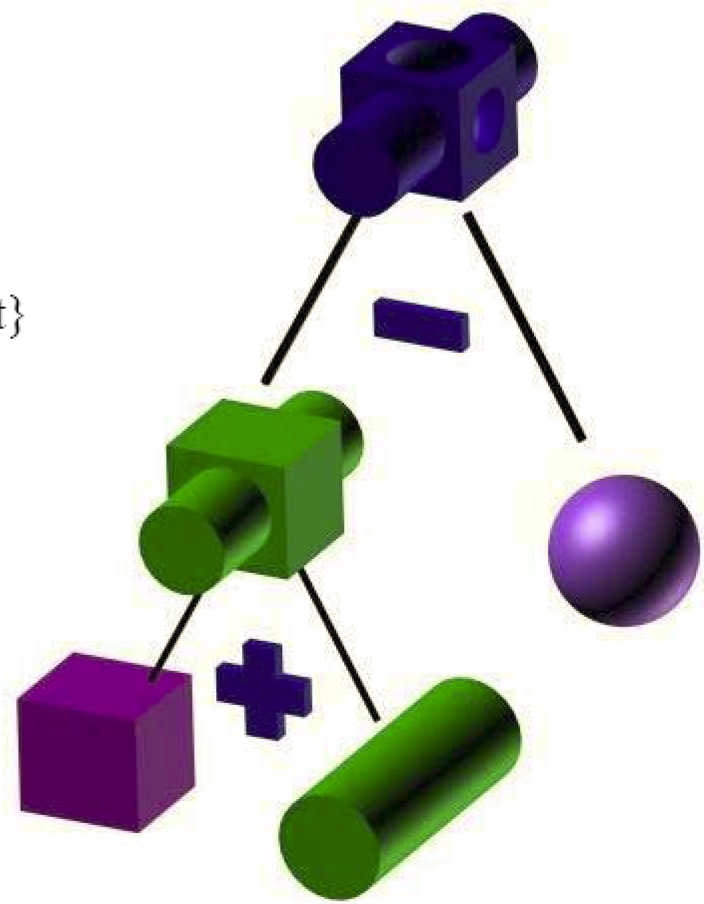
```
Vertex 1  123.472 75.6855 171.207 {rgb=(0.0207186 0.0137227 0.0205335) normal=(
Vertex 2  129.905 75.6904 169.427 {rgb=(0.0899862 0.0721164 0.0482489) normal=(
Vertex 3  135.957 75.6998 168.927 {rgb=(0.117921 0.0953541 0.0583396) normal=(-
Vertex 4  138.285 75.7013 168.438 {rgb=(0.110971 0.0836528 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=(-0.0
Vertex 10  136.328 76.1993 167.518 {rgb=(0.249965 0.209907 0.160202) normal=(-
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}

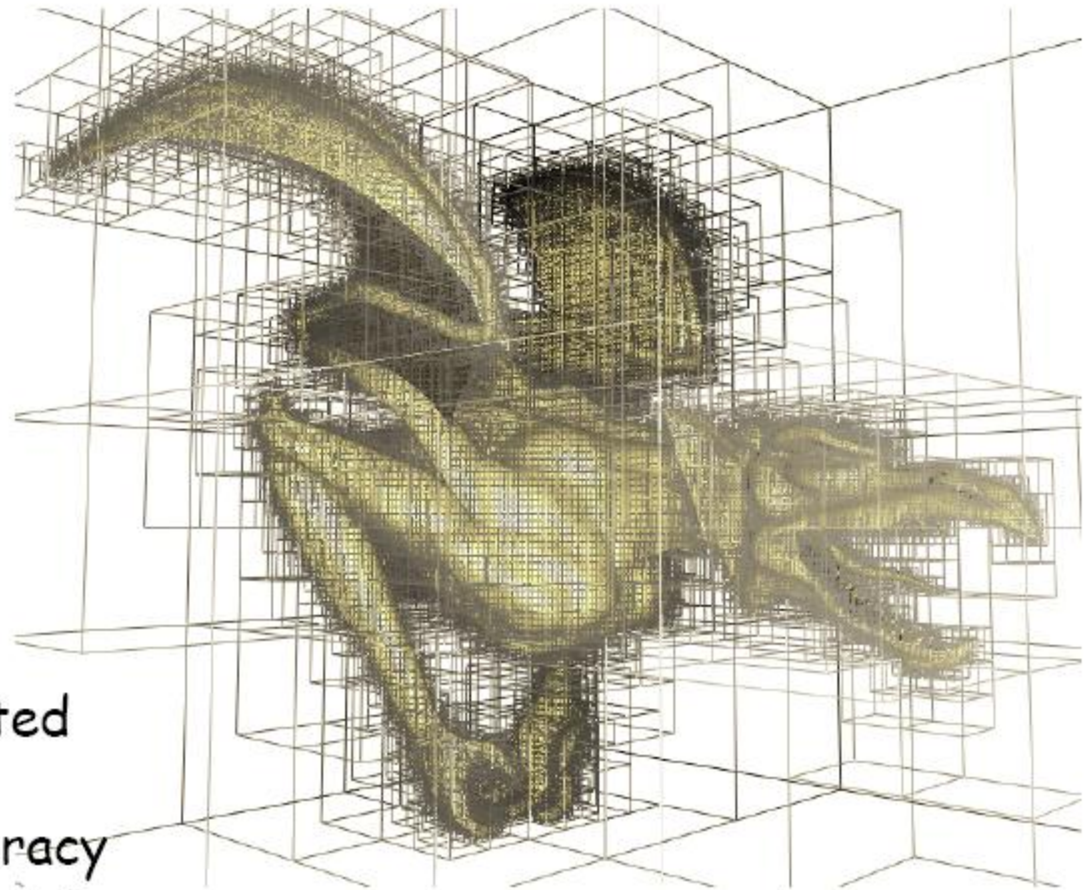
{union, subtract, intersect}

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.

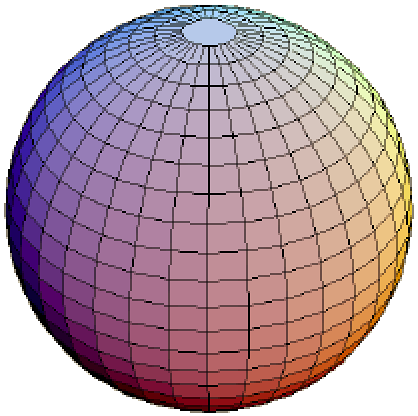
= a hierarchical tree built by sequential subdivision of occupied cells

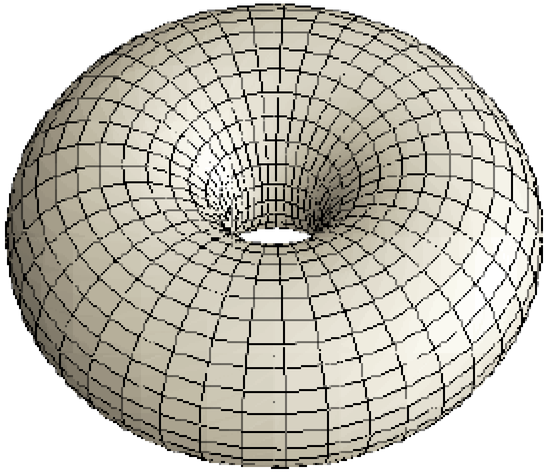
- Widely used for complicated scenes that need faster processing and lower accuracy

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)
  - ...

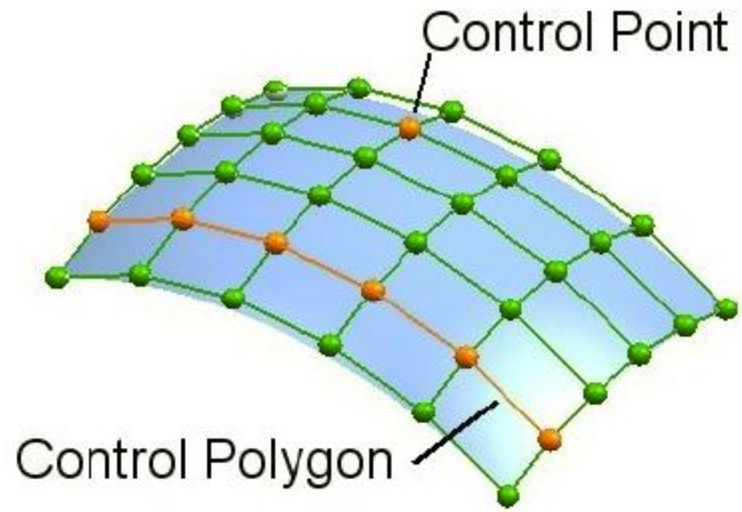


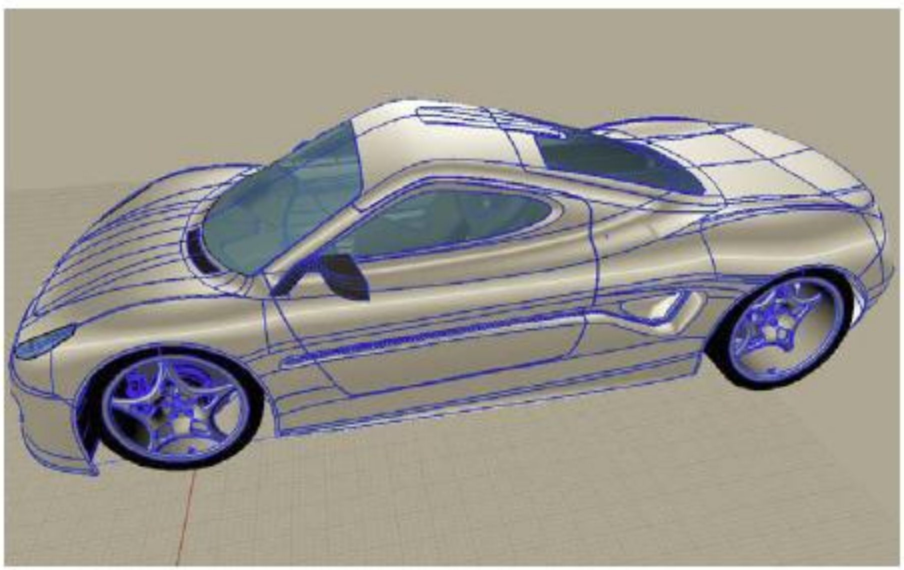


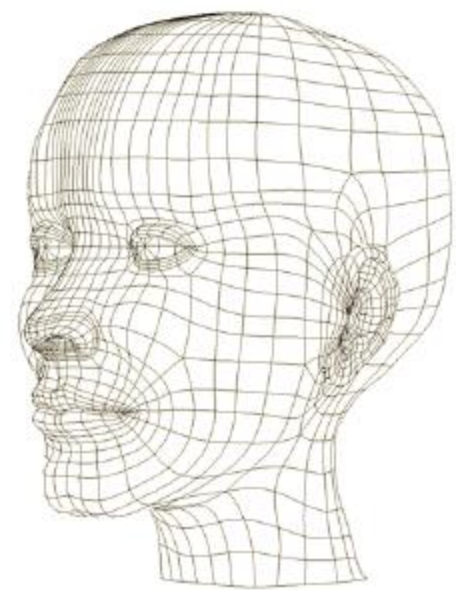
Spline

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

- Major modeling techniques in CAD







Resources

Textbooks and Reference books (not required) :

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

To do research in CG:

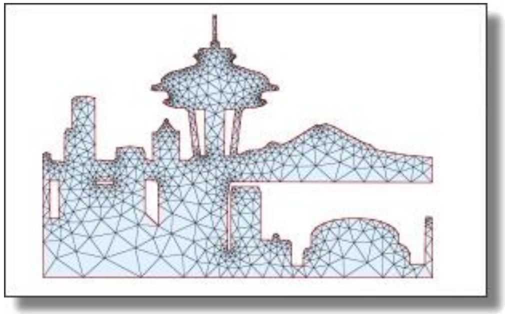
What math is important for Computer Graphics? (by Greg Turk)

Welcome to drop by my office for discussion, or check my webpage: www.ece.lsu.edu/xinli

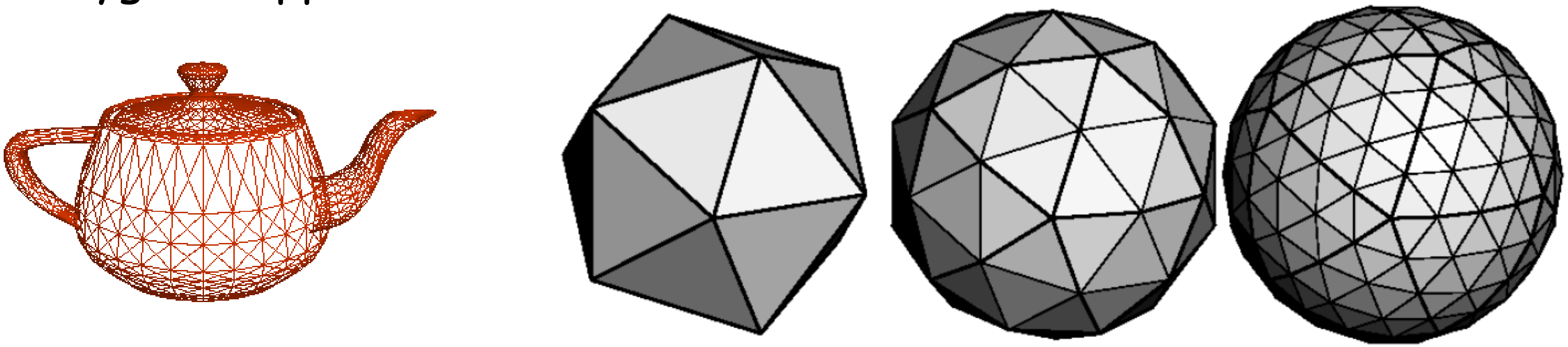
Questions?

Triangular Mesh

- Geometric shapes can be triangulated



Polygonal approximation of surfaces:

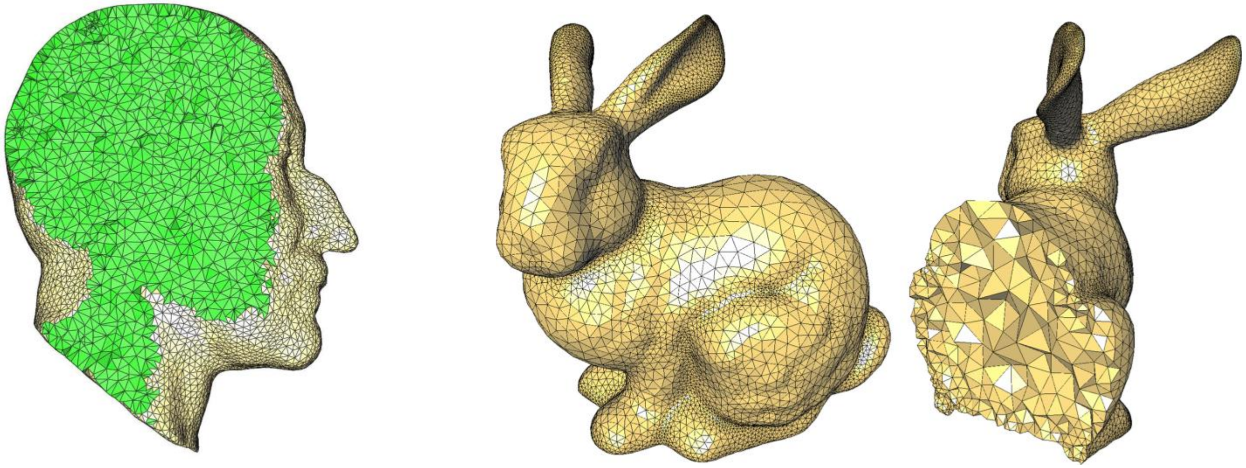


Any 2D shape or 3D surface (2-manifolds) can be approximated with locally linear polygons. To improve (visual or numerical approximation quality), we only need to increase the number of edges

Tetrahedral Mesh

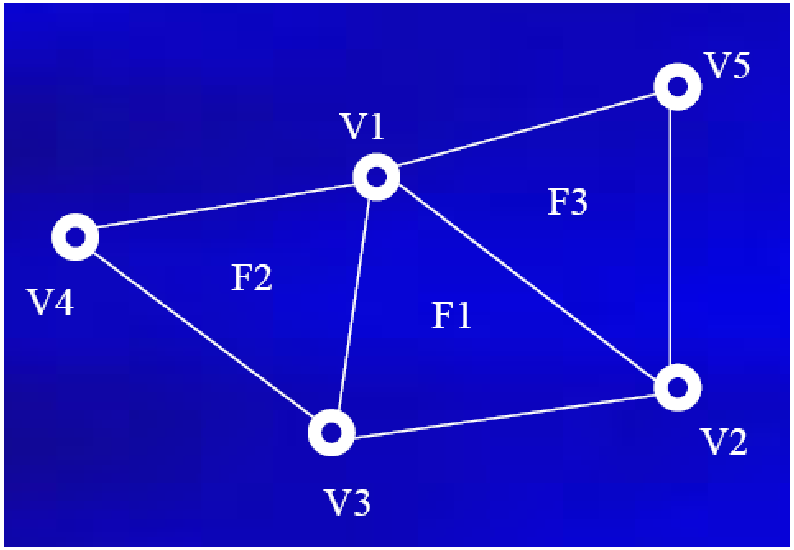
- Solid shapes can be tetrahedralized

Polyhedra approximation of solid geometric data



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

How to Represent Triangular Meshes?

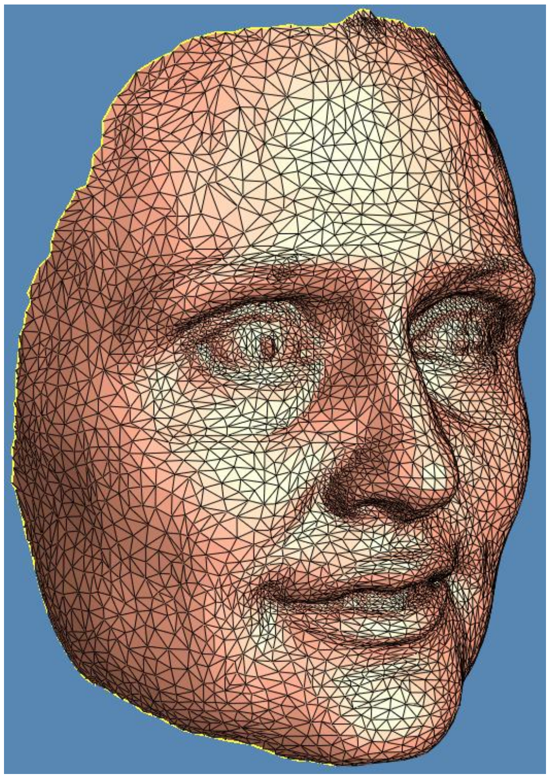


Vertex table	
V1	(x1,y1,z1)
V2	(x2,y2,z2)
V3	(x3,y3,z3)
V4	(x4,y4,z4)
V5	(x5,y5,z5)

Face table	
F1	V1,V3,V2
F2	V1,V4,V3
F3	V5,V1,V2

How to Represent Triangular Meshes?

Example: a female face mesh with 10k triangles



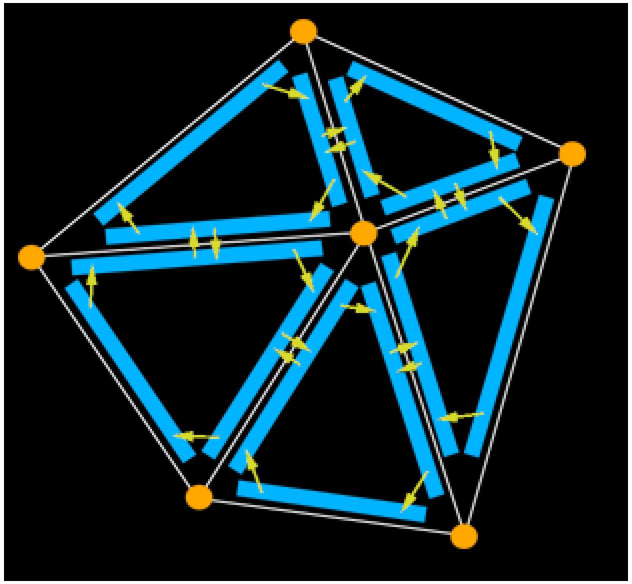
```
Vertex 1 0.6036570072 0.4613159895 0.07038059831
Vertex 2 0.6024590135 0.4750890136 0.07134509832
Vertex 3 0.6083189845 0.4888899922 0.07735790312
Vertex 4 0.611634016 0.5039420128 0.08098520339
Vertex 5 0.6236299872 0.5097290277 0.09412530065
Vertex 6 0.633580029 0.5194600224 0.1063940004
Vertex 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
```

How to Represent Triangular Meshes?

A widely-used data structure:

Half-Edge structure



- ❑Concepts and algorithm will be discussed soon
- ❑Full implementation will be provided
- ❑Get familiar with it (will be our starting point in future projects)
- ❑Your warm-up project is to compile it and run it

How to Render Triangle Meshes?

In the coming weeks:

How to use OpenGL to render triangular meshes: