# Modélisation 3D & Simulation

## Introduction

Hyewon Seo
Equipe ‘MLMS’, ICube

Partial credit to : G. C. Burdea, P. Coiffet, “Virtual Reality Technology”, Wiley-Interscience

# Course outline

## Course 1 :

- Introduction
- Trianguler mesh representation
- Building geometric models
- Kinematic modeling
  - Face modeling
  - Body modeling

## Course 2 :

- Data-driven modeling
  - Face modeling
  - Body modeling
  - Correspondence finding

# Course outline

## Course 3 :

- Learning-based modeling
  - Face modeling
- Physics-based modeling
  - cloth modeling
- Collision detection

## Course 4 :

- Programming practice (Matlab + Meshlab)
  - Geometric deformation and/or
  - Hierarchical modeling

# About me

- **CNRS researcher since 2009**
  - Research director since 2020
  - Co-head of Machine Learning, Modeling & Simulation team
- **Assistant/Associate prof. CNU, South Korea (2004-2009)**
  - Director of CGAL (Computer Graphics & Application Lab, 2005-2009)
- **Adjunct prof. POSTECH (2015-)**
  - Contact person, AI graduate school (2019-)

- **HDR thesis,** "Data-driven methods and intuitive control for modeling shape, motion & deformation", Univ. Strasbourg.
- **PhD thesis,** Univ. Geneve, Suisse.
- **BSc, MSc,** Korea Advanced Institute of Science & Technology, South Korea.

# Geometric modeling?

## Geometric modeling

- important constituent of computer vision as well as computer graphics research.
- concerned with the object representation, recognition, synthesis, and manipulation.

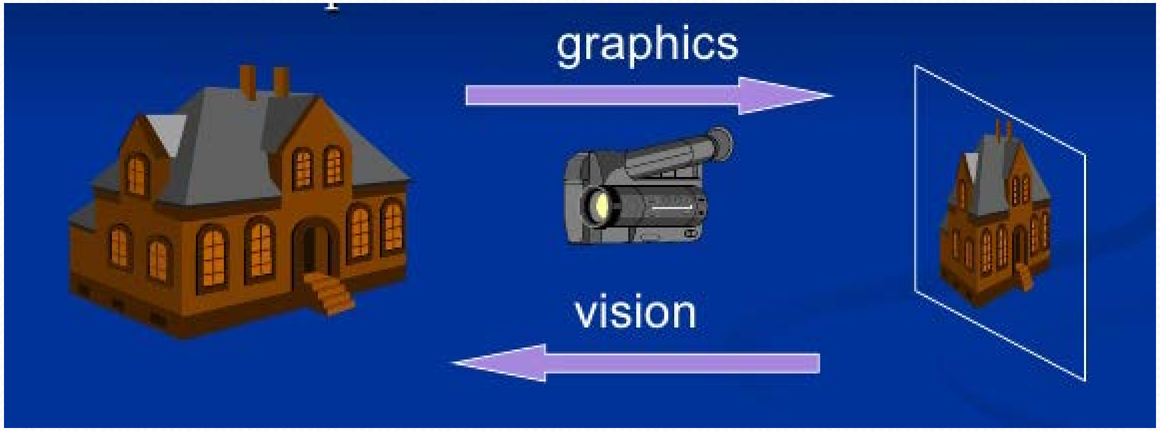
## Computer graphics

- a sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content.

## Computer vision

- is concerned with modeling and replicating human vision using computer software and hardware.

# CG vs CV:




- CV: Understanding the “content” of an image (usually based on a model of the depicted scene)
- CG: Creating an image from scratch using a geometric model.

# Modélisation 3D & Simulation

## Geometric modeling:
- Polygonal mesh representation
- Model acquisition

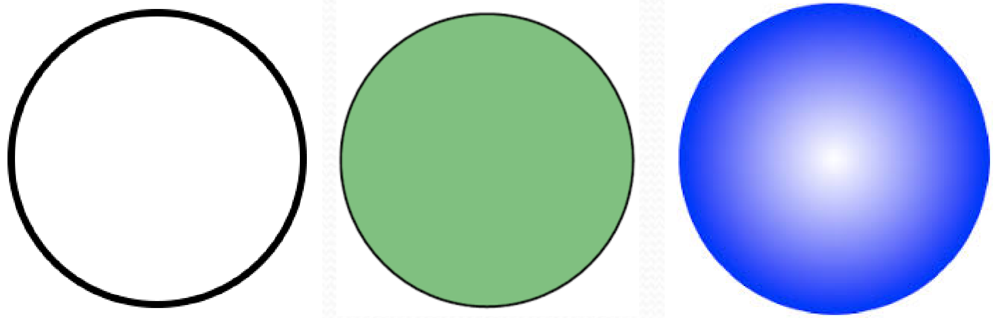
Hyewon Seo
Equipe 'MLMS', ICube

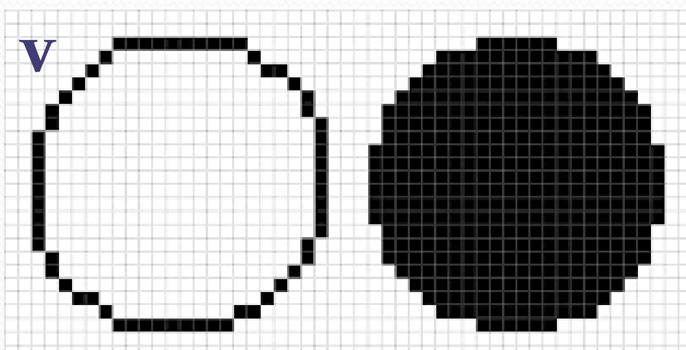
**Partial credit to : G. C. Burdea, P. Coiffet, "Virtual Reality Technology", Wiley-Interscience**

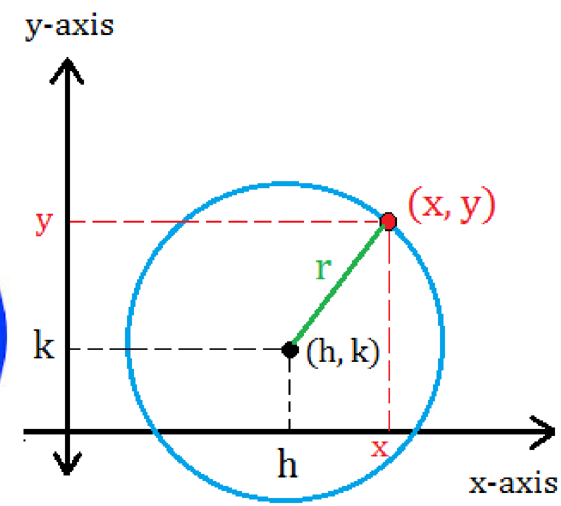
# How to describe a 2D object?

Continuous (functional) vs discreet representation

## Surface (boundary) vs volume representation



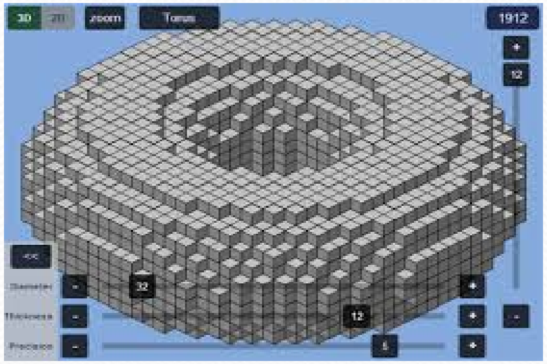


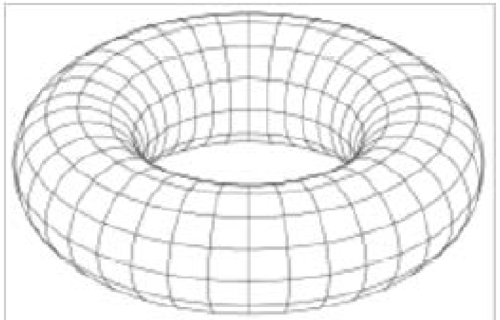

$$(x - h)^2 + (y - k)^2 = r^2$$

# Shape representation 3D

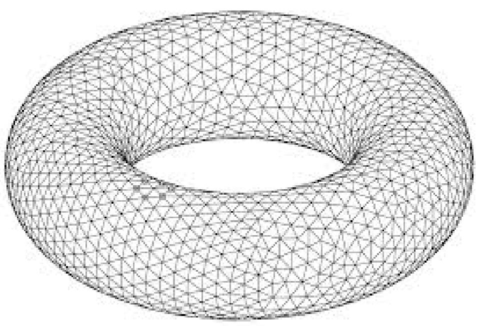
## Surface (boundary) vs volume representation




## Continuous (functional) vs discreet representation



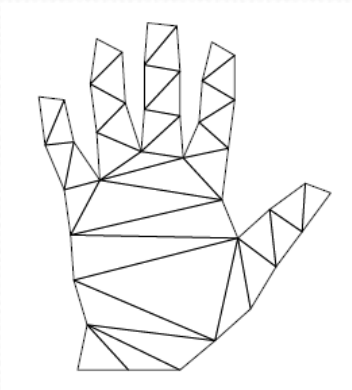
Implicit surface torus (R=40, a=15).

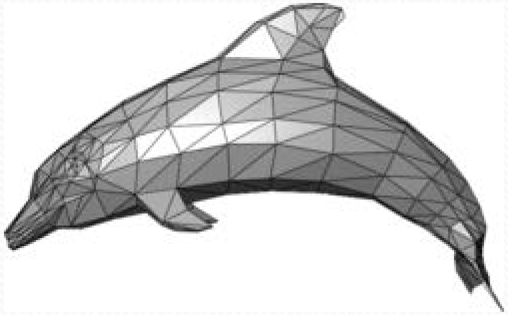


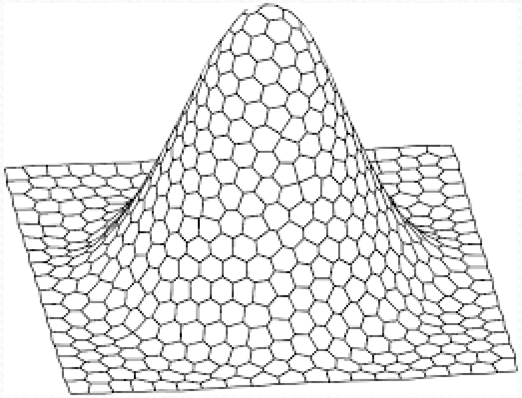
$$(x^2 + y^2 + z^2 + R^2 - a^2)^2 - 4R^2(x^2 + y^2) = 0$$

## Definition of a polygonal mesh

- **"In 3D modeling, a polygon mesh is a collection of vertices, edges and faces that defines the shape boundary of an object."**
  - The faces usually consist of triangles (triangle mesh), quadrilaterals (quads), or other simple convex polygons (n-gons).

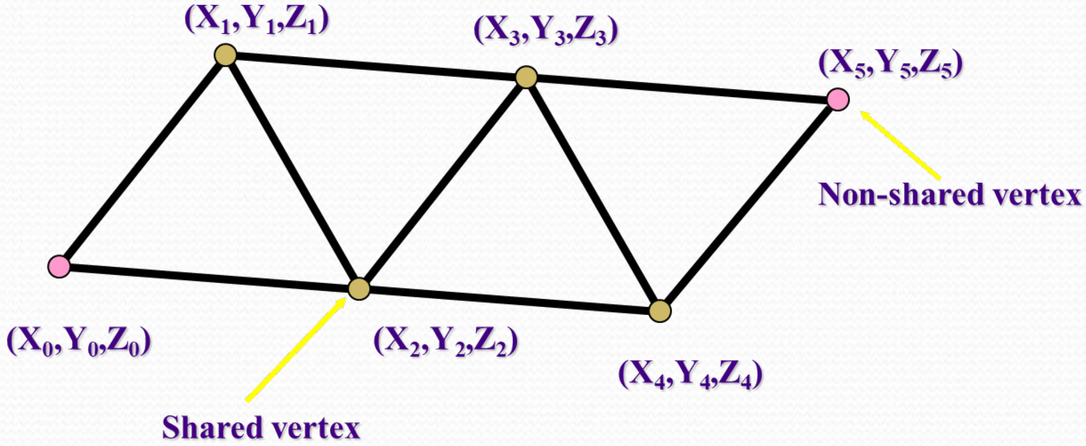




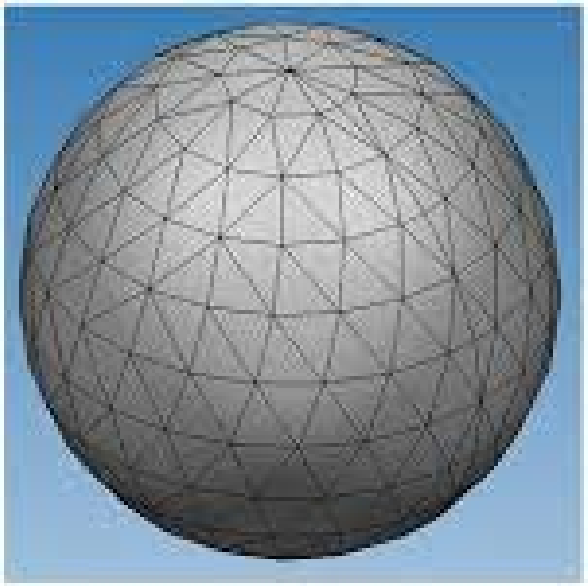


# Why mesh representation?

- **It simplifies rendering…**
  - In most cases, it's the **surface** of the object that **contributes to the image**
- **'Triangle' meshes are preferred since they are memory and computationally efficient**
  - All 3 vertices in a triangle lie on a same plane
  - Hardwares are optimized to render triangle meshes



# Polygonal mesh: an example



Sphere in polygonal representation (418 triangles)

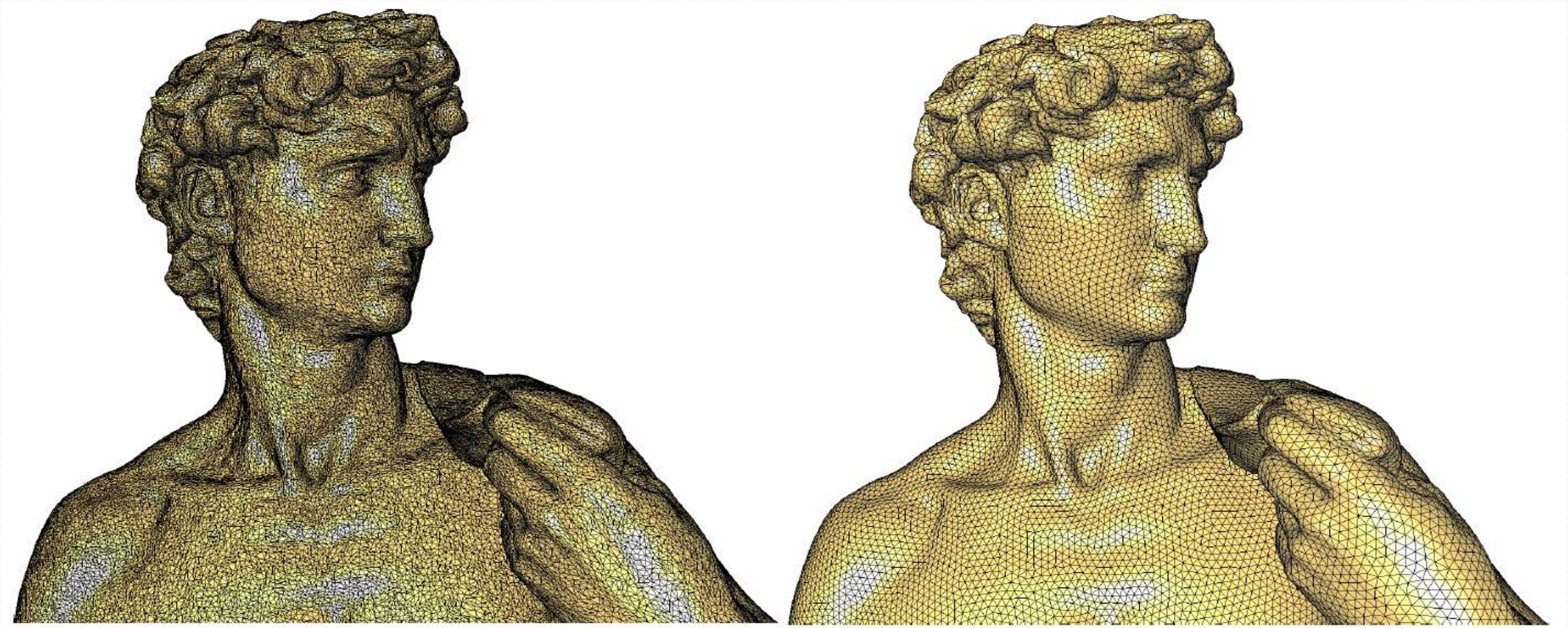
Sphere in polygonal representation (105 triangles)

**Detailed/fine vs coarse mesh**

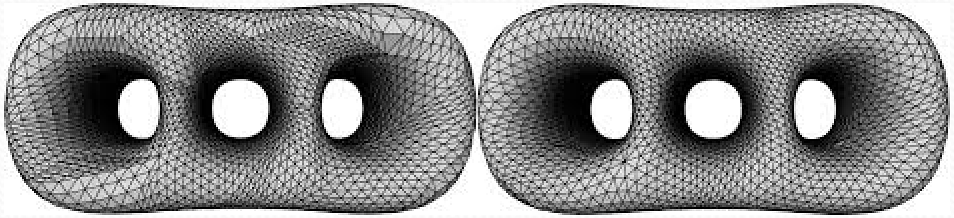
**Simplification/decimation**

# Remeshing

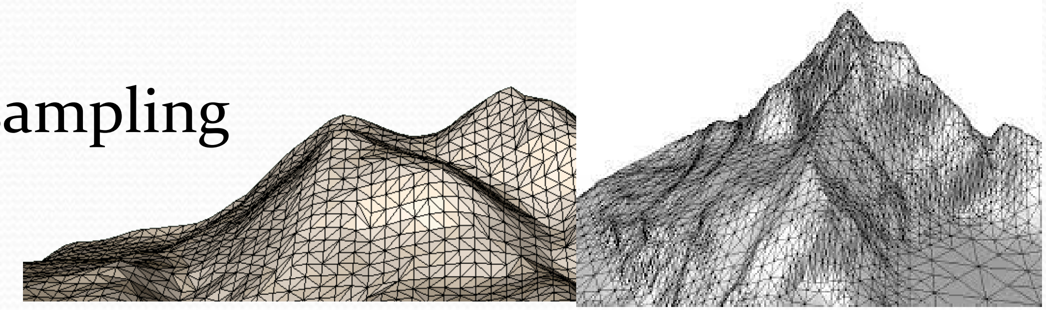
Given a 3D triangular mesh, find a "better" discrete representation of the underlying surface

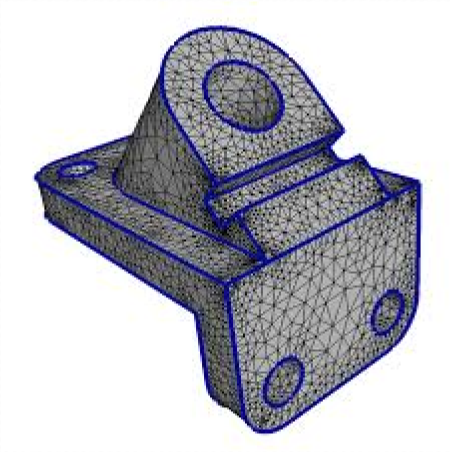


## What is a good mesh?



- Equal edge lengths
- Equilateral triangles
- Uniform vs. adaptive sampling





- Feature preservation

## Acquisition of the data

1. Programmed using OpenGL or other toolkit editor
   - tedious and requires skill;
2. Obtained from existing CAD files

**Our approach during the programming practice!**

3. Created using a 3-D digitizer (stylus), or a 3-D scanner
4. Purchased from online databases (i.e. Viewpoint database)

- Files have **vertex location** and **connectivity** information, but are mostly static

# **Vertex location** and **connectivity** information?

## Face-Vertex Meshes

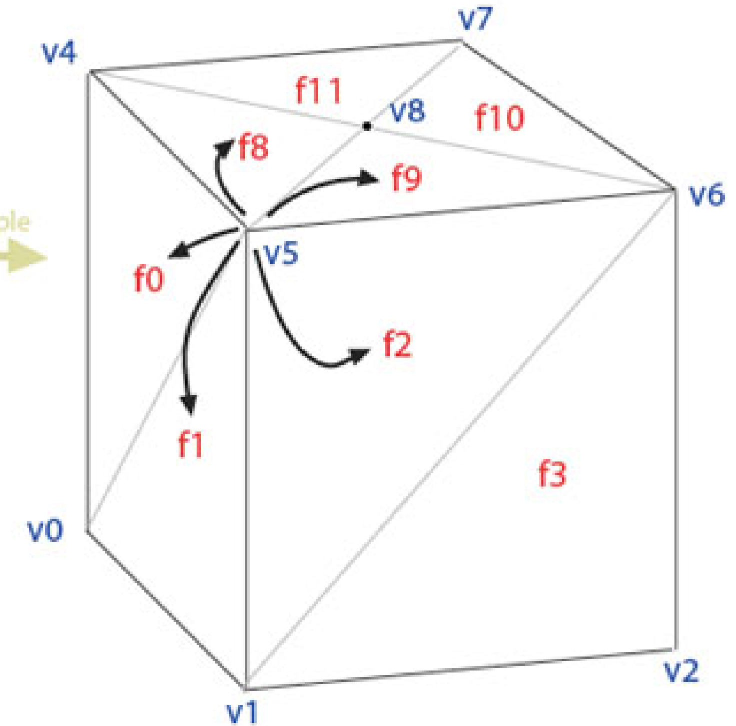
Face List

| | |
|---|---|
| f0 | v0 v4 v5 |
| f1 | v0 v5 v1 |
| f2 | v1 v5 v6 |
| f3 | v1 v6 v2 |
| f4 | v2 v6 v7 |
| f5 | v2 v7 v3 |
| f6 | v3 v7 v4 |
| f7 | v3 v4 v0 |
| f8 | v8 v5 v4 |
| f9 | v8 v6 v5 |
| f10 | v8 v7 v6 |
| f11 | v8 v4 v7 |
| f12 | v9 v5 v4 |
| f13 | v9 v6 v5 |
| f14 | v9 v7 v6 |
| f15 | v9 v4 v7 |

Vertex List

| | | |
|---|---|---|
| v0 | 0,0,0 | f0 f1 f12 f15 f7 |
| v1 | 1,0,0 | f2 f3 f13 f12 f1 |
| v2 | 1,1,0 | f4 f5 f14 f13 f3 |
| v3 | 0,1,0 | f6 f7 f15 f14 f5 |
| v4 | 0,0,1 | f6 f7 f0 f8 f11 |
| v5 | 1,0,1 | f0 f1 f2 f9 f8 |
| v6 | 1,1,1 | f2 f3 f4 f10 f9 |
| v7 | 0,1,1 | f4 f5 f6 f11 f10 |
| v8 | .5,.5,0 | f8 f9 f10 f11 |
| v9 | .5,.5,1 | f12 13 14 15 |

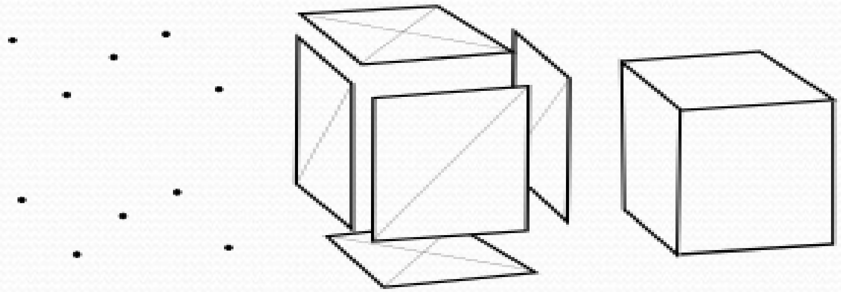
example



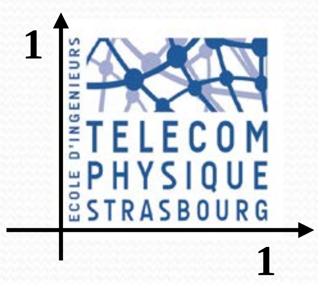

[wikipedia: Polygon mesh]

# 1. Programming

- Vertex list
- Face list



- Texture coordinate
- Vertex mapping to texture

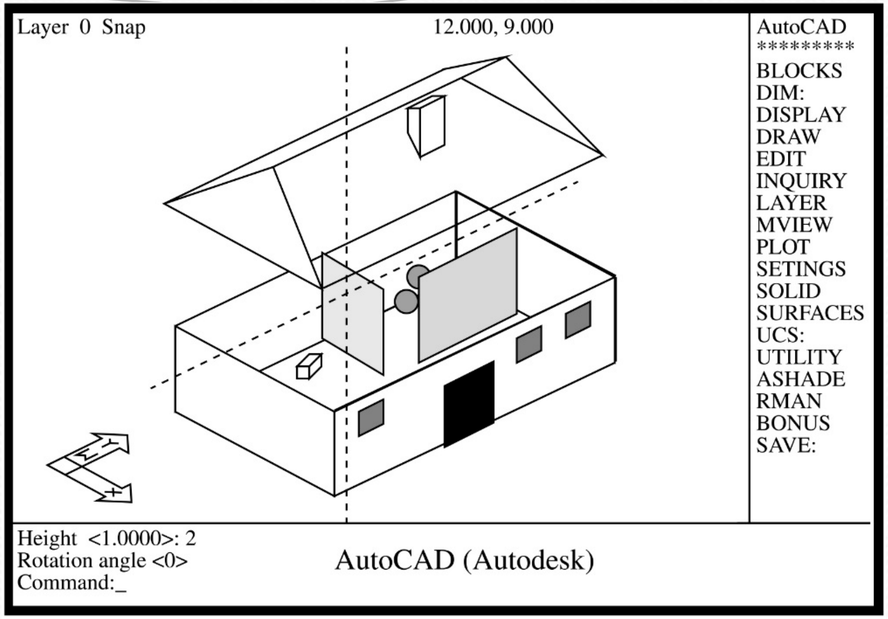



```
glBegin(GL_QUADS);
// Front Face
glTexCoord2f(0.0f, 0.0f); glVertex3f(-1.0f, -1.0f,  1.0f);
glTexCoord2f(1.0f, 0.0f); glVertex3f( 1.0f, -1.0f,  1.0f);
glTexCoord2f(1.0f, 1.0f); glVertex3f( 1.0f,  1.0f,  1.0f);
glTexCoord2f(0.0f, 1.0f); glVertex3f(-1.0f,  1.0f,  1.0f);
// Back Face
glTexCoord2f(1.0f, 0.0f); glVertex3f(-1.0f, -1.0f, -1.0f);
glTexCoord2f(1.0f, 1.0f); glVertex3f(-1.0f,  1.0f, -1.0f);
glTexCoord2f(0.0f, 1.0f); glVertex3f( 1.0f,  1.0f, -1.0f);
glTexCoord2f(0.0f, 0.0f); glVertex3f( 1.0f, -1.0f, -1.0f);
// Top Face
glTexCoord2f(0.0f, 1.0f); glVertex3f(-1.0f,  1.0f, -1.0f);
glTexCoord2f(0.0f, 0.0f); glVertex3f(-1.0f,  1.0f,  1.0f);
glTexCoord2f(1.0f, 0.0f); glVertex3f( 1.0f,  1.0f,  1.0f);
glTexCoord2f(1.0f, 1.0f); glVertex3f( 1.0f,  1.0f, -1.0f);
// Bottom Face
glTexCoord2f(1.0f, 1.0f); glVertex3f(-1.0f, -1.0f, -1.0f);
glTexCoord2f(0.0f, 1.0f); glVertex3f( 1.0f, -1.0f, -1.0f);
glTexCoord2f(0.0f, 0.0f); glVertex3f( 1.0f, -1.0f,  1.0f);
glTexCoord2f(1.0f, 0.0f); glVertex3f(-1.0f, -1.0f,  1.0f);
// Right face
glTexCoord2f(1.0f, 0.0f); glVertex3f( 1.0f, -1.0f, -1.0f);
glTexCoord2f(1.0f, 1.0f); glVertex3f( 1.0f,  1.0f, -1.0f);
glTexCoord2f(0.0f, 1.0f); glVertex3f( 1.0f,  1.0f,  1.0f);
glTexCoord2f(0.0f, 0.0f); glVertex3f( 1.0f, -1.0f,  1.0f);
// Left Face
glTexCoord2f(0.0f, 0.0f); glVertex3f(-1.0f, -1.0f, -1.0f);
glTexCoord2f(1.0f, 0.0f); glVertex3f(-1.0f, -1.0f,  1.0f);
glTexCoord2f(1.0f, 1.0f); glVertex3f(-1.0f,  1.0f,  1.0f);
glTexCoord2f(0.0f, 1.0f); glVertex3f(-1.0f,  1.0f, -1.0f);
glEnd();
```

# 2. CAD-file based models

## CAD-file based models:

- Done using AutoCAD;
- Each semantic part in a separate file;
- Files need to be converted to formats compatible with other softwares;
- Advantage – use of preexisting models in manufacturing applications.




AutoCAD (Autodesk)

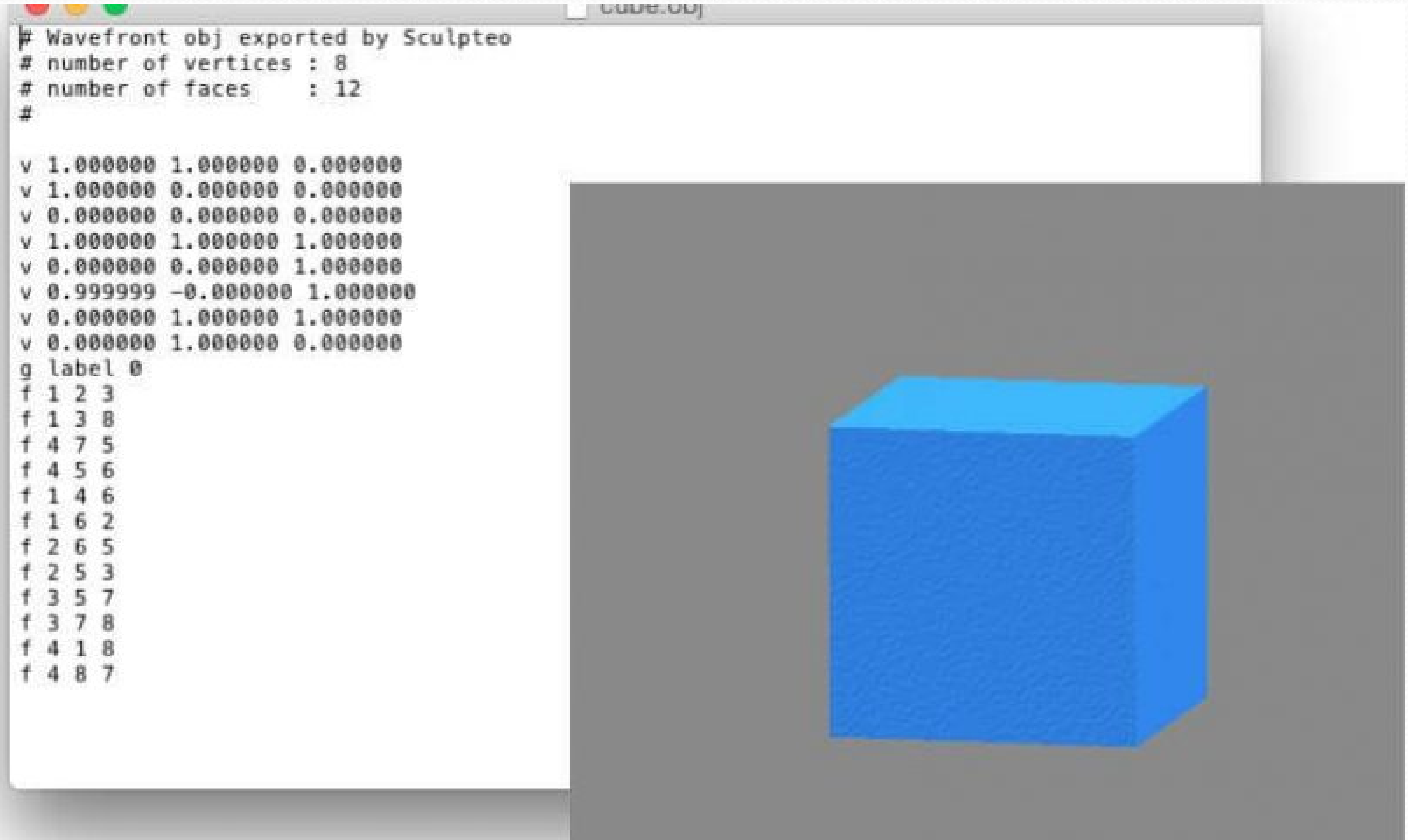
```
HOUSE.DXF
  LINE
   8
 0
   10
 2.934
   20
 6.5
   30
 1.060
   11
 4.500
   21
 6.500
   31
 -0.500
 210
 0.707
 220
 0.000

 230
 0.707
   0
 POLYLINE
   8
 0
   10
 3.292
   20
 4.139
   30
 0.707
 VERTEX
   8
 0
   10
 4.133
   20
 -1.828
   30
 2.567

 VERTEX
   8
 0
   10
 4.133
   20
 -0.828
   30
 2.828
   50
 0.000
   0
 VERTEX
   8
 0
   10
 4.1339
   20
 .
 .

 .
 .
 CIRCLE
   8
 0
   10
 5.500
   20
 2.000
   30
 -0.500
   40
 0.500
   0
 ARC
   8
 0
   10
 5.560
   20
 .
```

# CAD files can also be downloaded from the Internet



```
# Wavefront obj exported by Sculpteo
# number of vertices : 8
# number of faces    : 12
#

v 1.000000 1.000000 0.000000
v 1.000000 0.000000 0.000000
v 0.000000 0.000000 0.000000
v 1.000000 1.000000 1.000000
v 0.000000 0.000000 1.000000
v 0.999999 -0.000000 1.000000
v 0.000000 1.000000 1.000000
v 0.000000 1.000000 0.000000
g label 0
f 1 2 3
f 1 3 8
f 4 7 5
f 4 5 6
f 1 4 6
f 1 6 2
f 2 6 5
f 2 5 3
f 3 5 7
f 3 7 8
f 4 1 8
f 4 8 7
```

# 3. 3D Digitizers

**Venus de Milo created using the HyperSpace 3D digitizer**

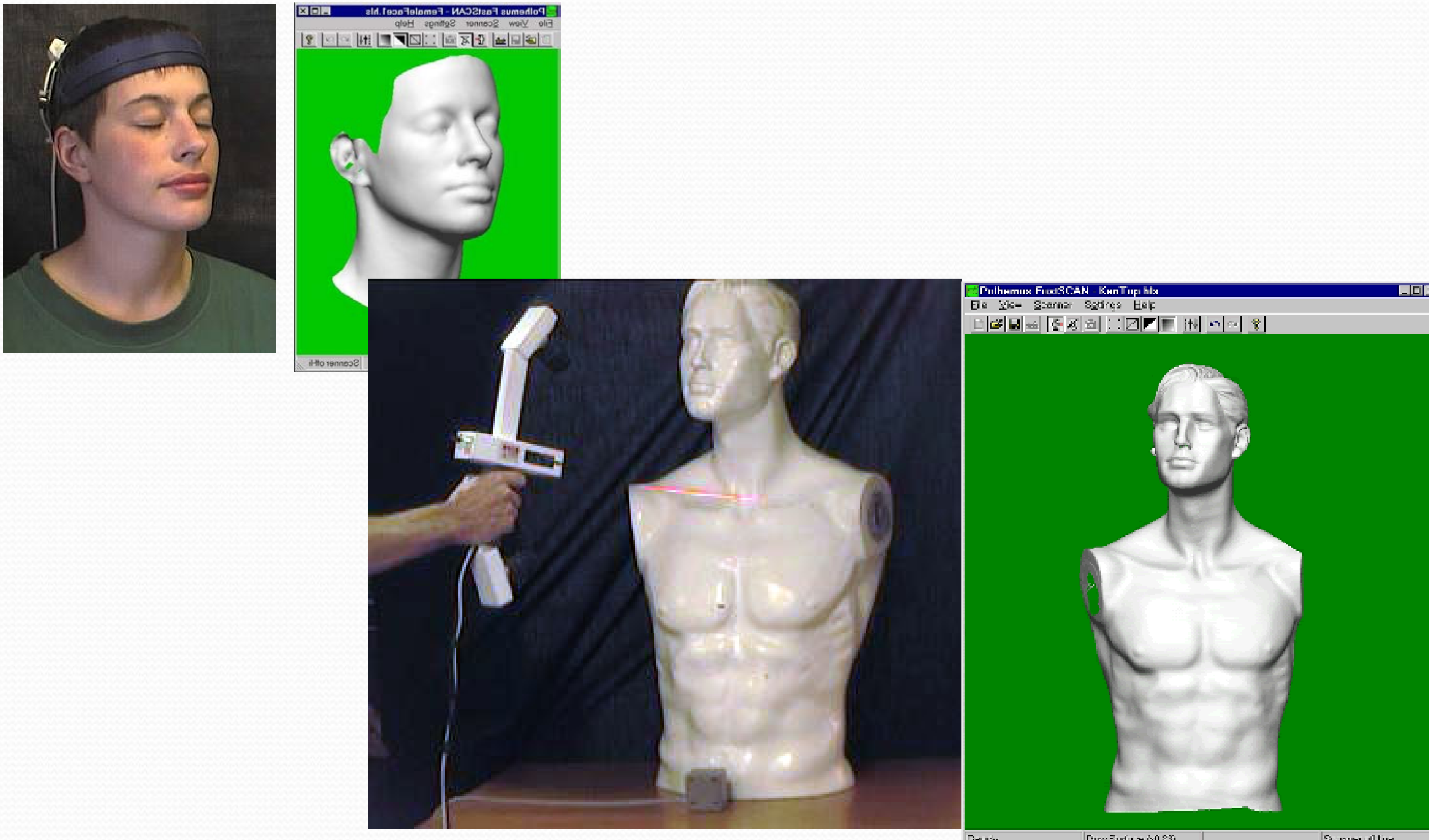
- Venus de Milo created using the HyperSpace 3D digitizer
- Highly precise but can damage the object

## Polhemus 3-D scanners

- Eliminate direct contact with object
- Uses two cameras, a laser, and magnetic trackers (if movable objects are scanned)
- Uses time-of-flight to measure the distance from the light source

- Scanning resolution : 0.5 mm at 200 mm range;
- Scanning speed : 50 lines/sec;
- Range : 75-680 mm scanner-object range.

# 3. 3D scanners



Polhemus FastScan 3D scanner (can scan objects up to 3 m long).

https://www.youtube.com/watch?v=SyzgBycPxyw&t=56s

**Polhemus scanner**

**DeltaSphere 3000 3D scanner**


www.3rdtech.com

| **Feature** | **Polhemus scanner** | **DeltaSphere scanner** |
|---|---|---|
| Range | 0.56 m | 14.6 m |
| Resolution | 0.5 mm @ 0.2 m | 0.25 mm |
| Control | manual | automatic |
| Speed | 50 lines/sec | 25,000 samples/sec |

# DeltaSphere 3000 image

www.3rdtech.com

## Nowadays: movements are tolerated while scaning

## Nowadays: dynamic scan data

- Video scanners: scan objects under movement
- Works especially well on human faces or bodies
- Representation of it is an open problem

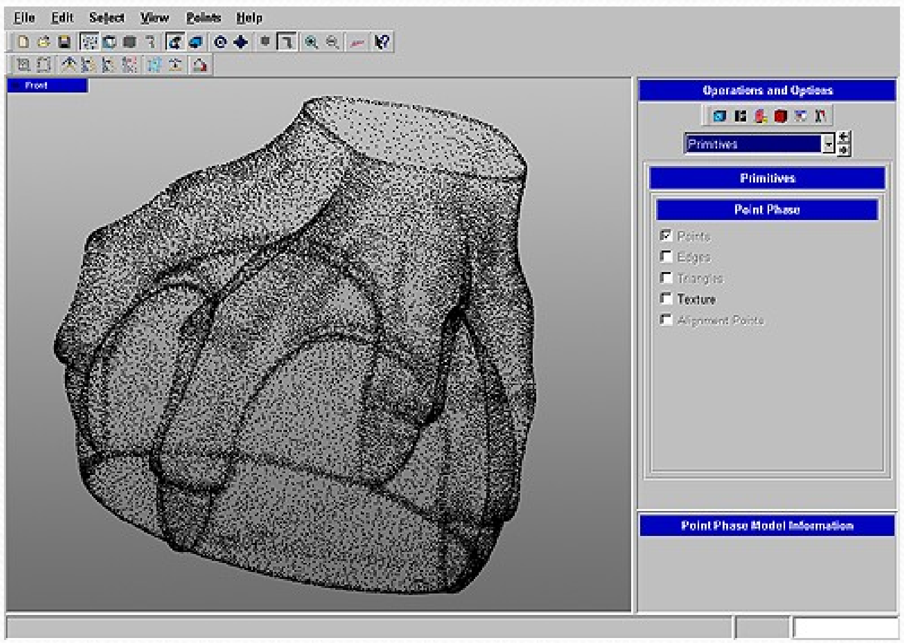
**https://www.youtube.com/watch?v=jh6msLkwqhE**

# 3. 3D scanners

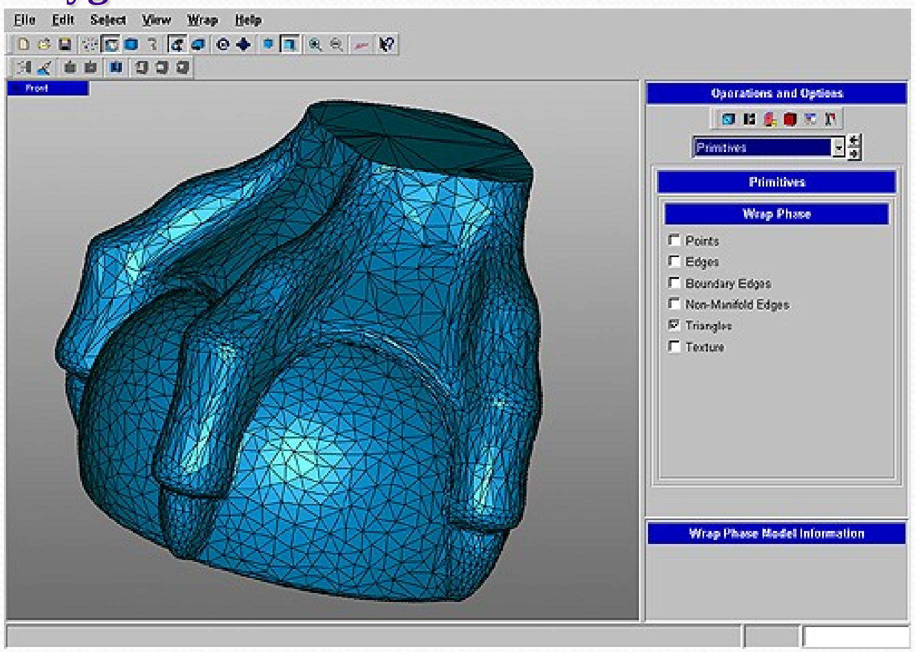
## Conversion of scanner data

- Scanners produce a dense "cloud" of vertices
- Using such packages as Meshlab, the point data is transformed into surface data (including editing and decimation)

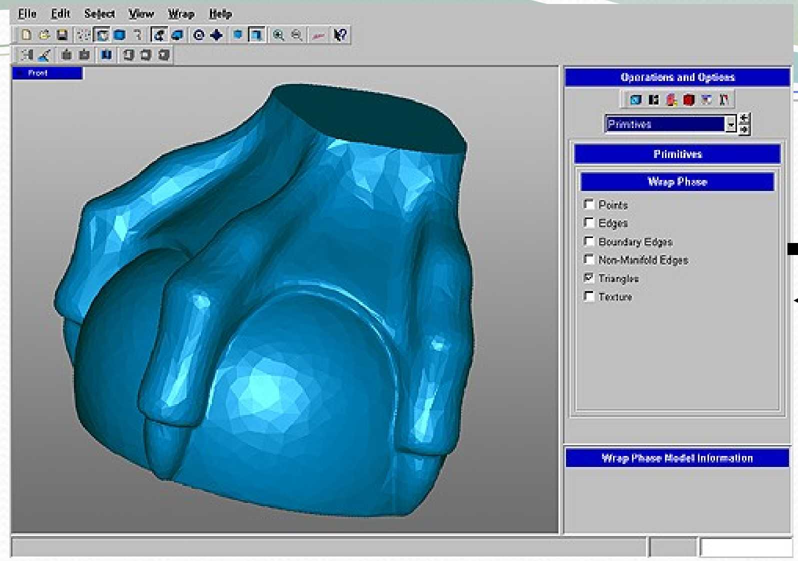
**Point cloud from scanner**



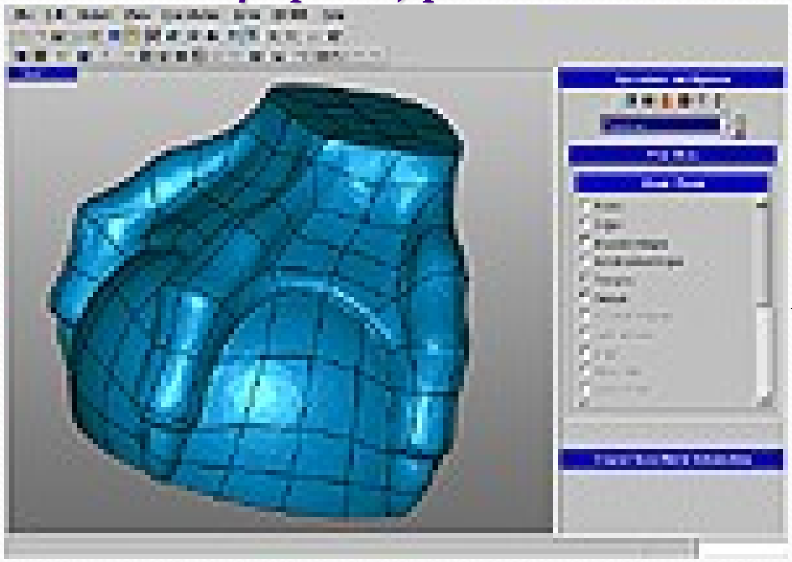

**Polygonal mesh after decimation**



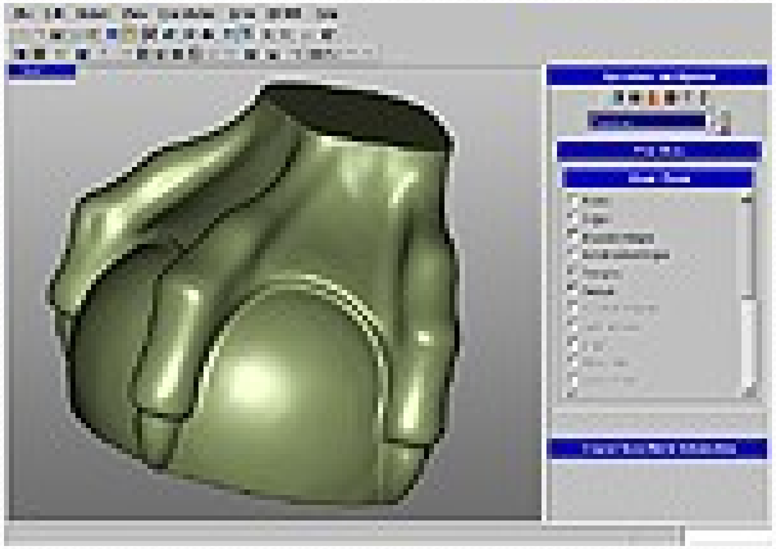
**Polygonal surface**




**NURBS (non-uniform rational β-splines) patches**



**NURBS surface**



# Purchase of 3D mesh data

- Can cost dozens to several thousand euros

Firefighter models from http://www.poserworld.com

Subaru 360 model from http://www.quality3dmodels.net

# Summary

*Table 5.1* Methods for modeling 3-D object geometry.

| METHOD | FEATURE | SOURCE |
|---|---|---|
| Toolkit<br>Editors | Tedious,<br>requires skill | OpenGL, Starbase,<br>PHIGS |
| CAD Programs | Interactive,<br>existing technology | AutoCAD (Autodesk)<br>3-D Studio, etc. |
| 3-D Digitizers | Interactive,<br>Allows<br>custom models | Autodesys Inc.<br>Mira Imaging,<br>Polhemus Inc.,etc. |
| 3-D Scanners | Fast multi-point acquisition<br>Large objects | Polhemus Inc.,<br>Cyra Technologies, etc. |
| Commercial<br>3-D databases | Vertice list,<br>connectivity,<br>static model, level of detail | Viewpoint Inc., etc. |