An Introduction to Geometric Modeling using Polygonal Meshes

Joaquim Madeira

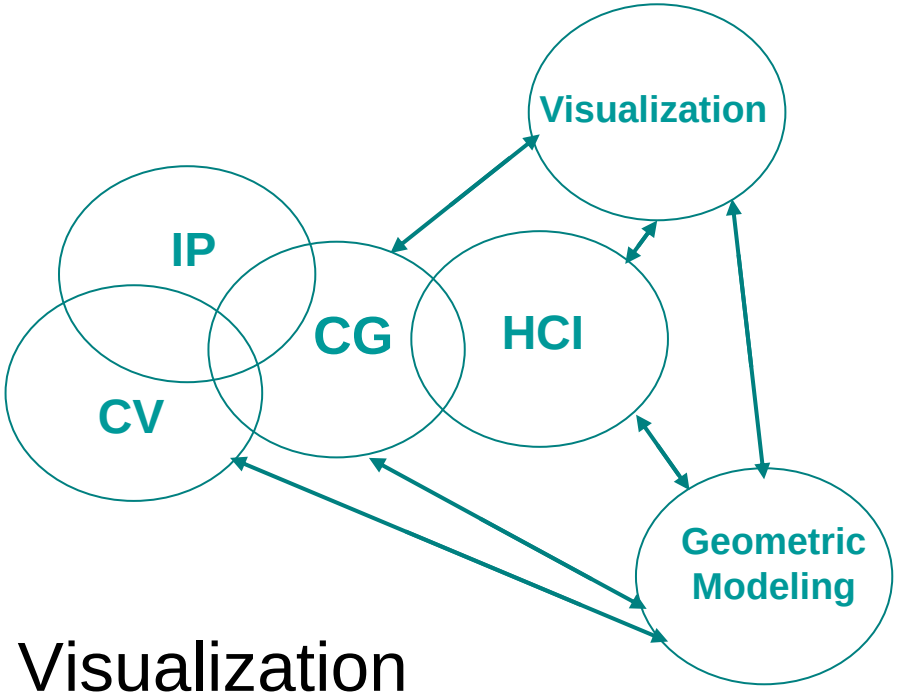
Version 0.3 – December 2020

Overview

- Motivation
- Polygonal meshes
- Exact vs. Approximate representation
- The processing pipeline
- Geometrical and topological information
- Valid vs. non-valid models
- The Euler formula
- Computational representation

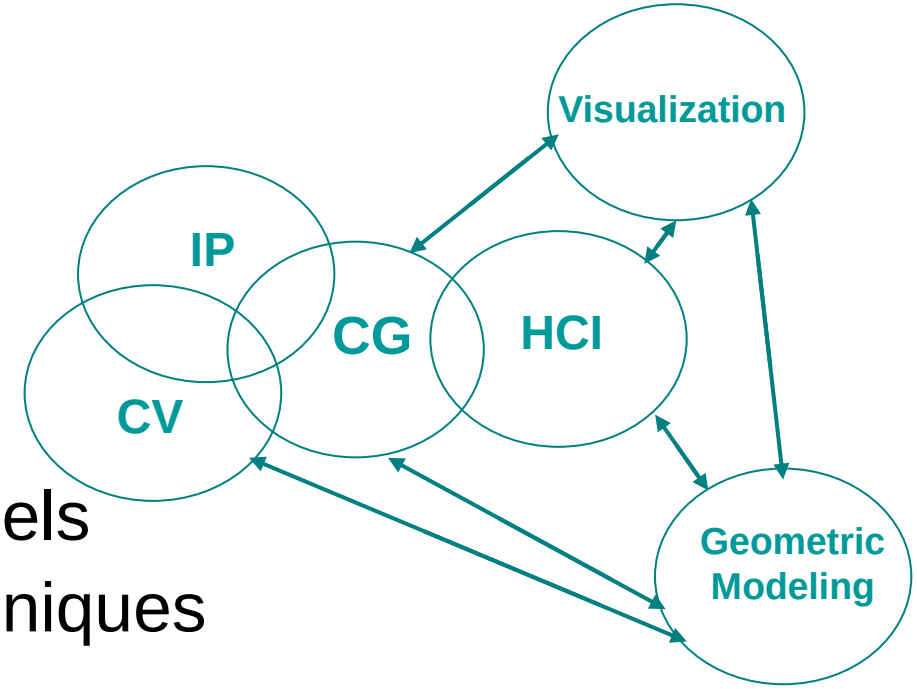
COMPUTER GRAPHICS & GEOMETRIC MODELING

CG is not alone…



- Core areas:
  - CG, IP, CV and HCI
- Satellite areas:
  - Geometric Modeling
  - Data and Information Visualization
- What is common?
  - CG, IP : image file formats, color models, …
  - CG, CV : 3D model representations, …
  - IP, CV : noise removal, filters, …

CG is not alone…



- Geometric Modeling
  - CV : 3D scanning
  - CG : 2D and 3D models
  - HCI : interaction techniques
- Visualization
  - HCI : interaction techniques
  - GeoM : 2D and 3D models
  - CG : rendering

Example – Medical Imaging

- Processing pipeline
  - Noise removal
  - Segmentation
  - Generating 2D / 3D models
  - Data visualization
  - User interaction
  - …

[www.mevislab.de]

CG Main Tasks

- Modeling
  - Construct individual models / objects
  - Assemble them into a 2D or 3D scene
- Animation
  - Static vs. dynamic scenes
  - Movement and / or deformation
- Rendering
  - Generate final images
  - Where is the observer?
  - How is he / she looking at the scene?

Geometric Modeling

- A geometric model describes the shape of an (real or virtual) object

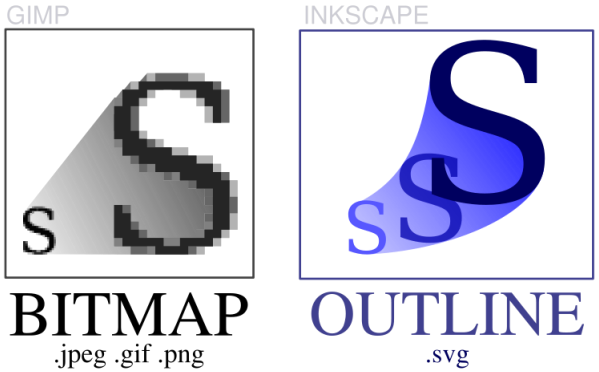
- How?
  - Different mathematical representations?
  - Data structures?
  - Possible operations?
  - Compactness ? Robustness ? Efficiency ?
  - Interpolation vs Approximation ?
  - …

Geometric Modeling

- What for?
  - Distinguish between inside, outside and border of a model
  - Compute properties
    - Centroid
    - Area / Volume
    - …
  - Detect interferences / collisions
  - Compute light reflections and / or transparencies
  - …

2D models – Applications

- Scalable Vector Graphics (SVG)
  - XML description of 2D graphics
  - Various primitives



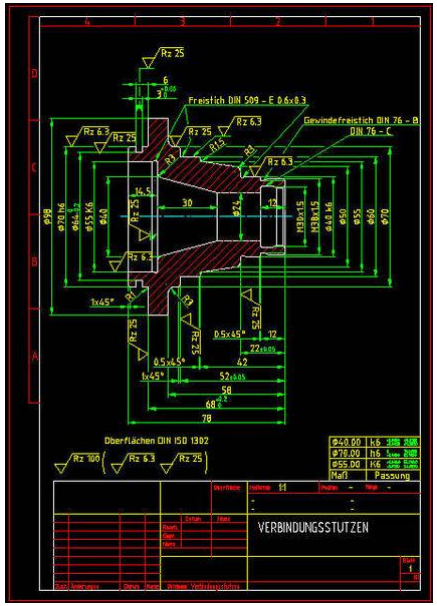
- Computer typography
  - Knuth's METAFONT uses Bézier curves

METAFONT

[Wikipedia]

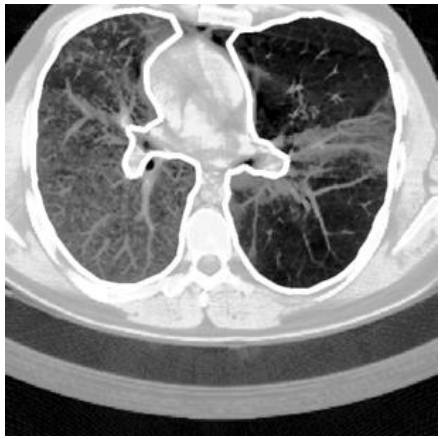
2D models – Applications

- Engineering drawings
  - E.g., AutoCAD



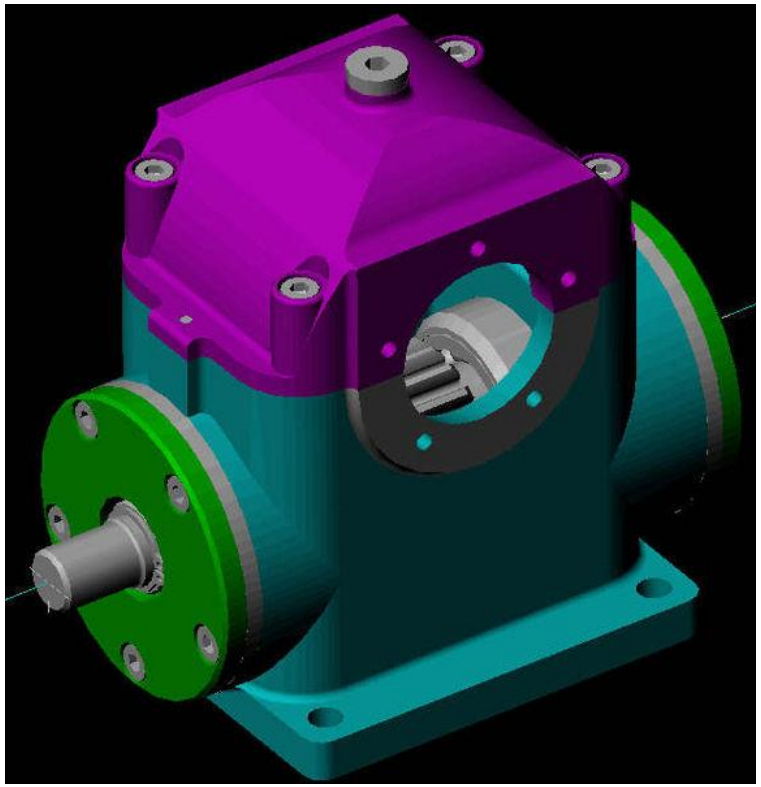
[Wikipedia]

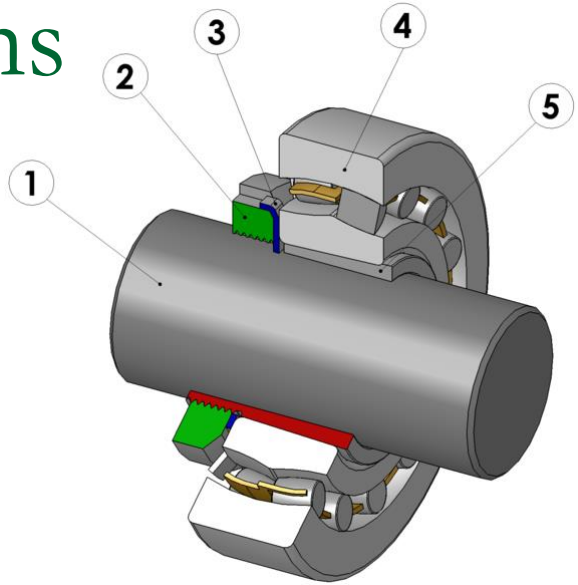
- Medical image processing
  - Representing contours
  - Interpolation vs approximation
  - Smoothness

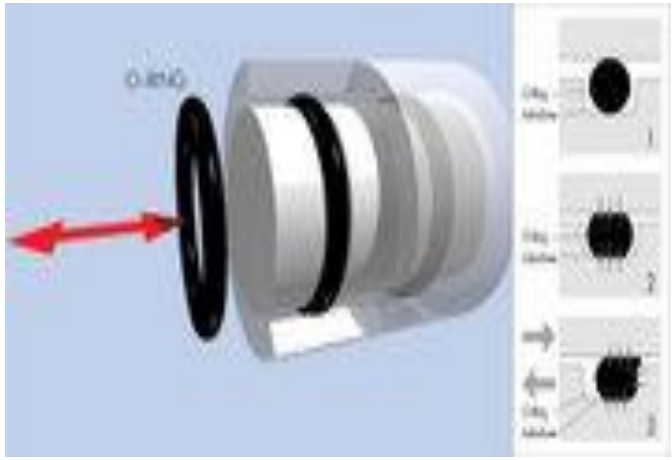


3D models – Applications

- CAD / CAM

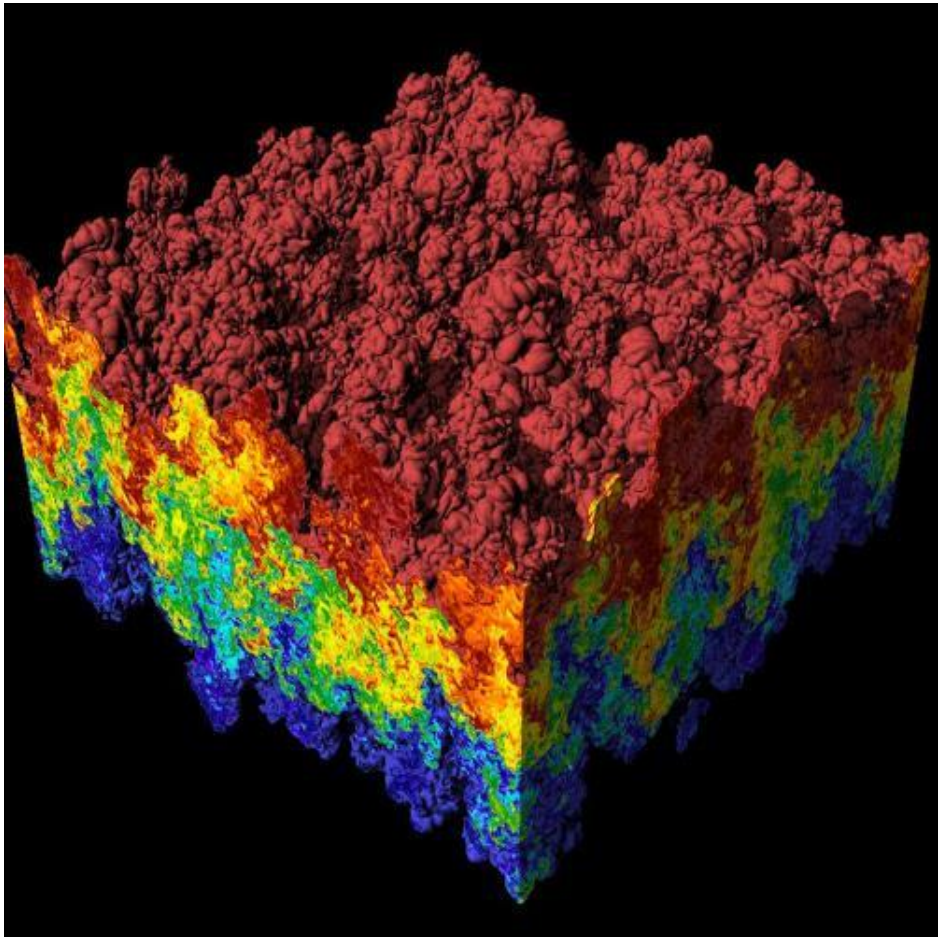






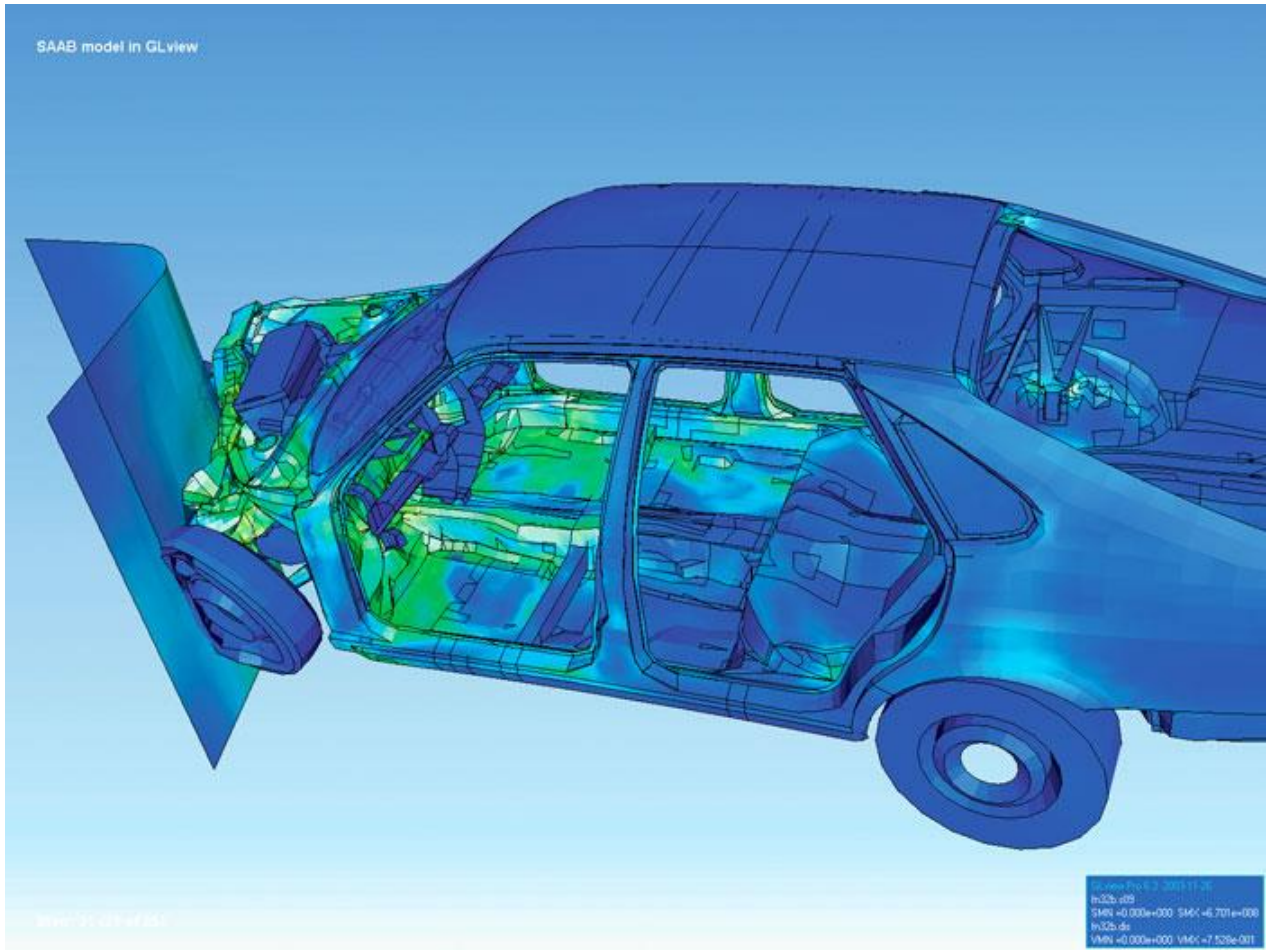
[Wikipedia]

Data Visualization



[Wikipedia]

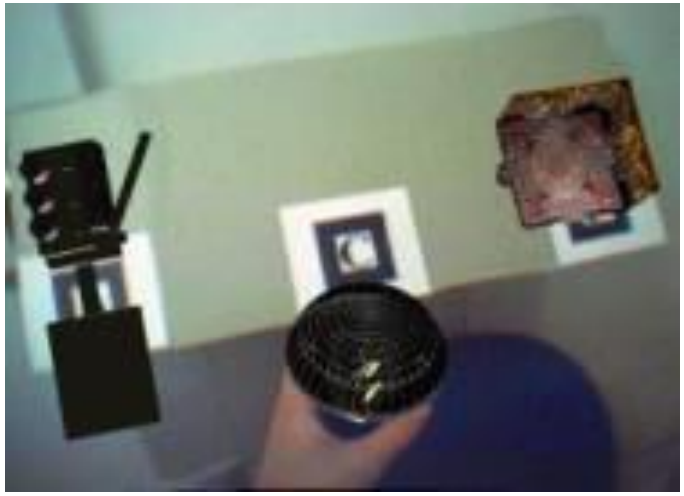
CAD – Simulation and Visualization



[Wikipedia]

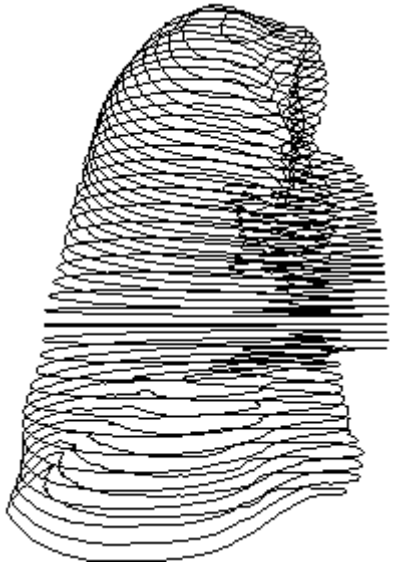
3D models – Applications

- Virtual / Augmented reality

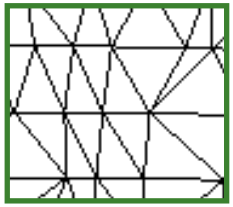


3D models – Applications

- Medical Data Processing



Mesh



Mesh detail

3D models – Applications

- Other application areas
  - Computer games
  - Geographical information systems (GIS)
  - Engineering analysis
  - 3D printing / Rapid prototyping
  - Medical solid modeling
  - …

3D models – Shape

- Define from scratch using VRML / X3D, OpenGL, VTK, …
  - Tedious; requires skill
- Obtain from CAD files or model databases
  - Convert to compatible formats
  - Use of existing models in manufacturing applications
- Create using a 3D digitizer or a 3D scanner
  - 3D digitizer : stylus
  - 3D scanner : tracker, cameras and laser

3D modeling tools

- Spatial Corp.'s ACIS; 3D modeling engine
  - http://www.spatial.com
- Siemens's Parasolid; 3D modeling engine
  - http://www.plm.automation.siemens.com

- Dassault Systemes's CATIA; CAD / CAM / CAE
  - http://www.3ds.com
- PTC's Pro/ENGINEER; 3D feature modeling
  - http://www.ptc.com
- SolidWorks; 3D feature modeling
  - http://www.solidworks.com

3D modeling tools

- Autodesk's 3ds max and Maya
  - http://www.autodesk.com
- Blender: Free open-source 3D content-creation suite
  - http://www.blender.org
- Rhino: Uninhibited free-form 3D modeling
  - http://www.rhino3d.com
- Trimble SketchUp: Intuitive 3D modeler
  - http://www.sketchup.com

- POV-Ray: Persistence of Vision Ray-Tracer
  - http://www.povray.org

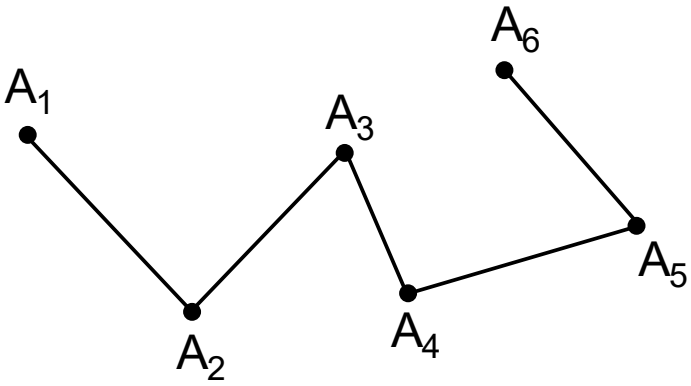
POLYGONAL MESHES

Geometric Modeling

- Main areas
  - Curve and surface modeling
    - Computer-Aided Geometric Design (CAGD)
  - Solid Modeling
  - Volume Modeling

- Simplest models
  - Curves : Polygonal lines
  - Surfaces : Polygonal meshes

Polylines



[Wikipedia]

- Questions :
  - Open or closed polylines ?
  - Exact representation ? When ?
  - Approximate representation ?

- A “good” approximation usually needs a larger number of points
  - Level of detail (LOD)

- Typical application
  - Representing contours in processed images, after locating dominant points

Polylines

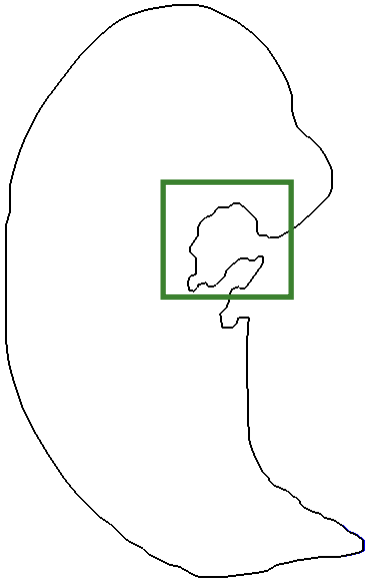
Contours segmented from CT images

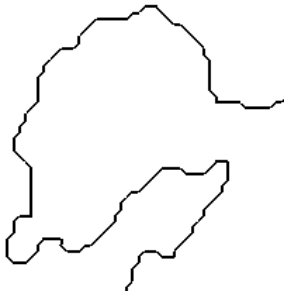
Contour description is point by point

↓

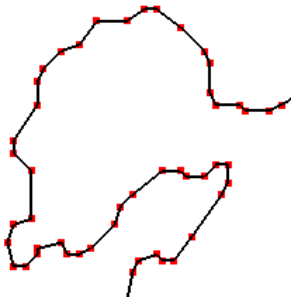
Too many points !

Solution: Dominant Point Detection





Contour detail



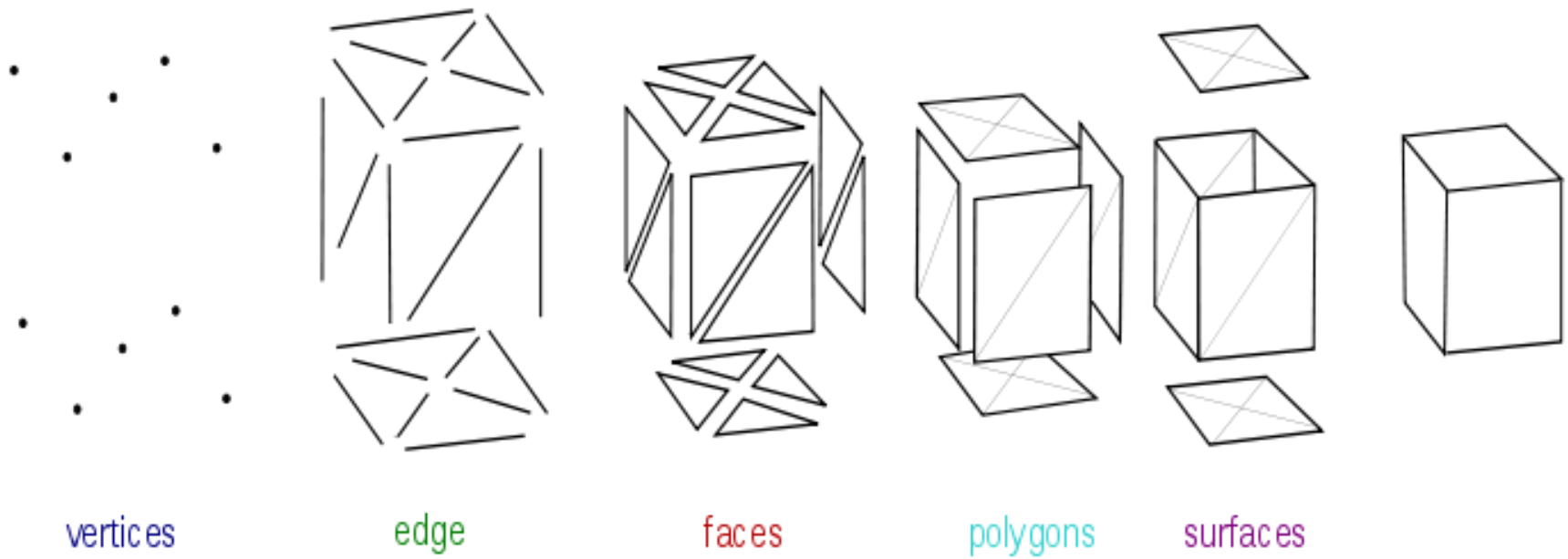
Dominant points

Polyline

Polygonal Meshes

- Surface is defined as a collection of neighboring faces (e.g., triangles)
  - Geometry + Topology (i.e., connectivity)
  - Vertices, edges, faces
- Euler formula for closed surfaces
  - $V + F - E = 2$
- Exact vs approximate representations
  - Polyhedral models
  - Curved surfaces
  - Terrain models
  - Complex surfaces / models

Polygonal Meshes

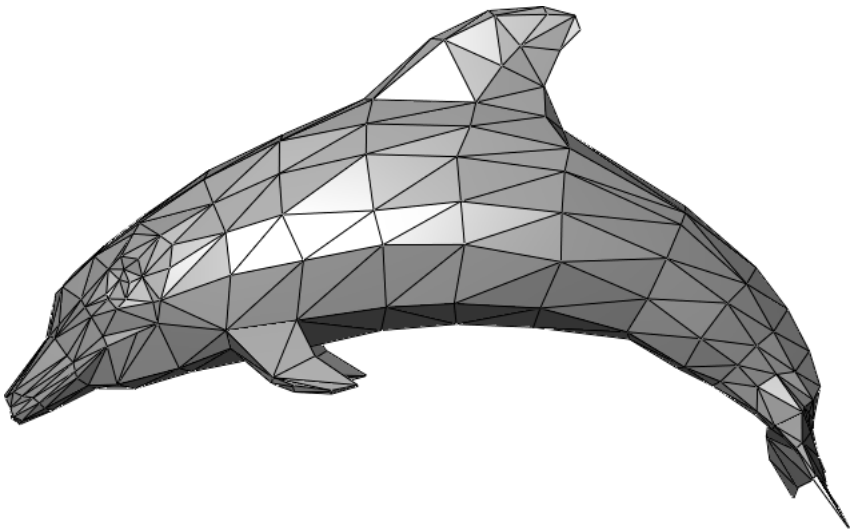


[Wikipedia]

Polygonal Meshes

- Collection of neighboring vertices, edges and polygons
  - Usually triangles !!
- Vertex
  - Shared by, at least, 2 edges
- Edge
  - Connects 2 vertices
  - Shared by 2 polygons, if the surface is closed
- Polygon
  - Sequence of, at least, 3 vertices

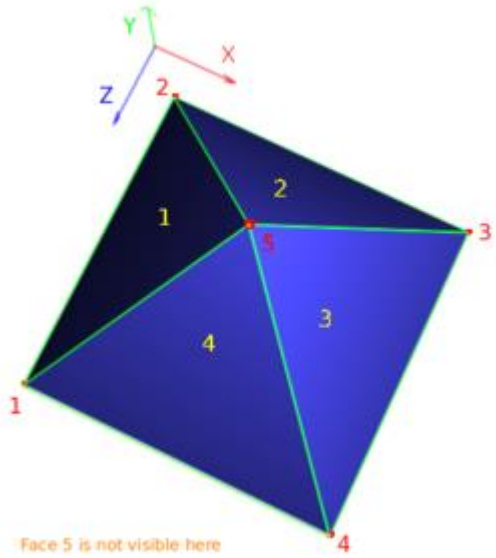
Polygonal Meshes



- Homogeneous ?
- Adaptive ?
- Easy to render

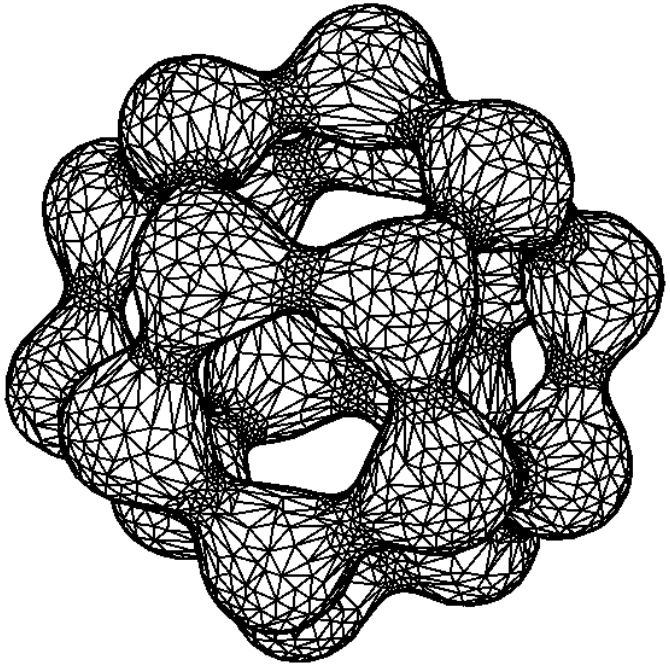
- Usually triangles !

- Pyramid
  - How many entities?
  - Check Euler formula!

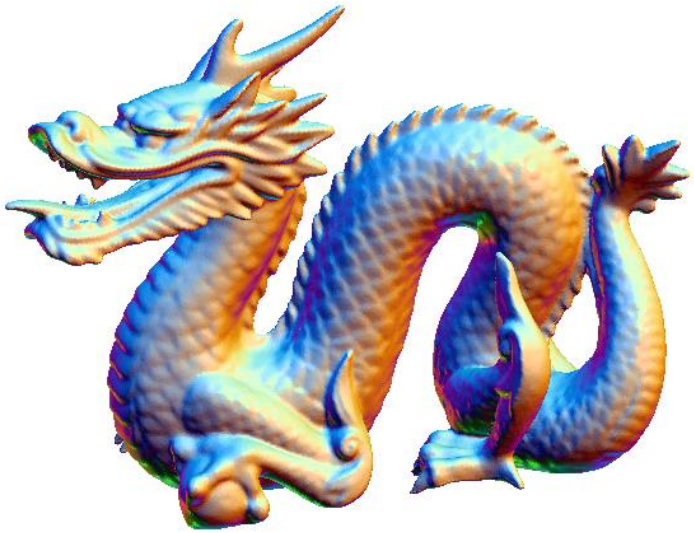


[Wikipedia]

Polygonal Meshes



Complex topology



Complex geometry

[Seidel and Belyaev, 2006]

Polygonal Meshes

- What / How to store?
  - Memory or file?
  - List of vertices – Topology ?
  - List of triangles – Neighbors?
  - Lists of vertices, edges and triangles – Efficiency?
  - Winged-edge or half-edge data structure

- Common operations
  - Smoothing
  - Decimation

- Toolboxes / Libraries
  - CGAL
  - OpenMesh

Polygonal Meshes

- The surface (i.e., the model) is defined as a set of adjacent faces (e.g., triangles)
- Which geometric information should be stored ?
  - Vertex coordinates
- Which topological information (i.e., connectivity) should be stored ?
  - How are edges and faces arranged ?
  - How to identify neighboring / incident /adjacent entities ?
  - Efficiency !
- Which additional properties should be stored ?
  - Normal vector to each face / vertex
  - Texture coordinates
- How to check the validity of a model ?
  - 2-manifolds
  - Euler Formulae

Some basic operations

- Find the vertices defining an edge
- Find the edges incident in a vertex
- Find all polygons sharing
  - A vertex
  - An edge
- Identify mesh errors. I.e., the lack of
  - A vertex / an edge / a face
- Rendering a mesh

Polygonal Meshes

- Supported by most applications
- Various file formats
- Triangle meshes are the most common !!
  - Planar faces
  - Algorithm simplicity
  - Numerical robustness
  - Efficient rendering

Polygonal Meshes

- Exact vs. approximate rep. – When ?
  - Polyhedral models
  - Curved surfaces
  - Terrain models
  - More complex models / surfaces

- A “good” approximation might require a large number of faces
  - Levels-of-Detail (LODs)

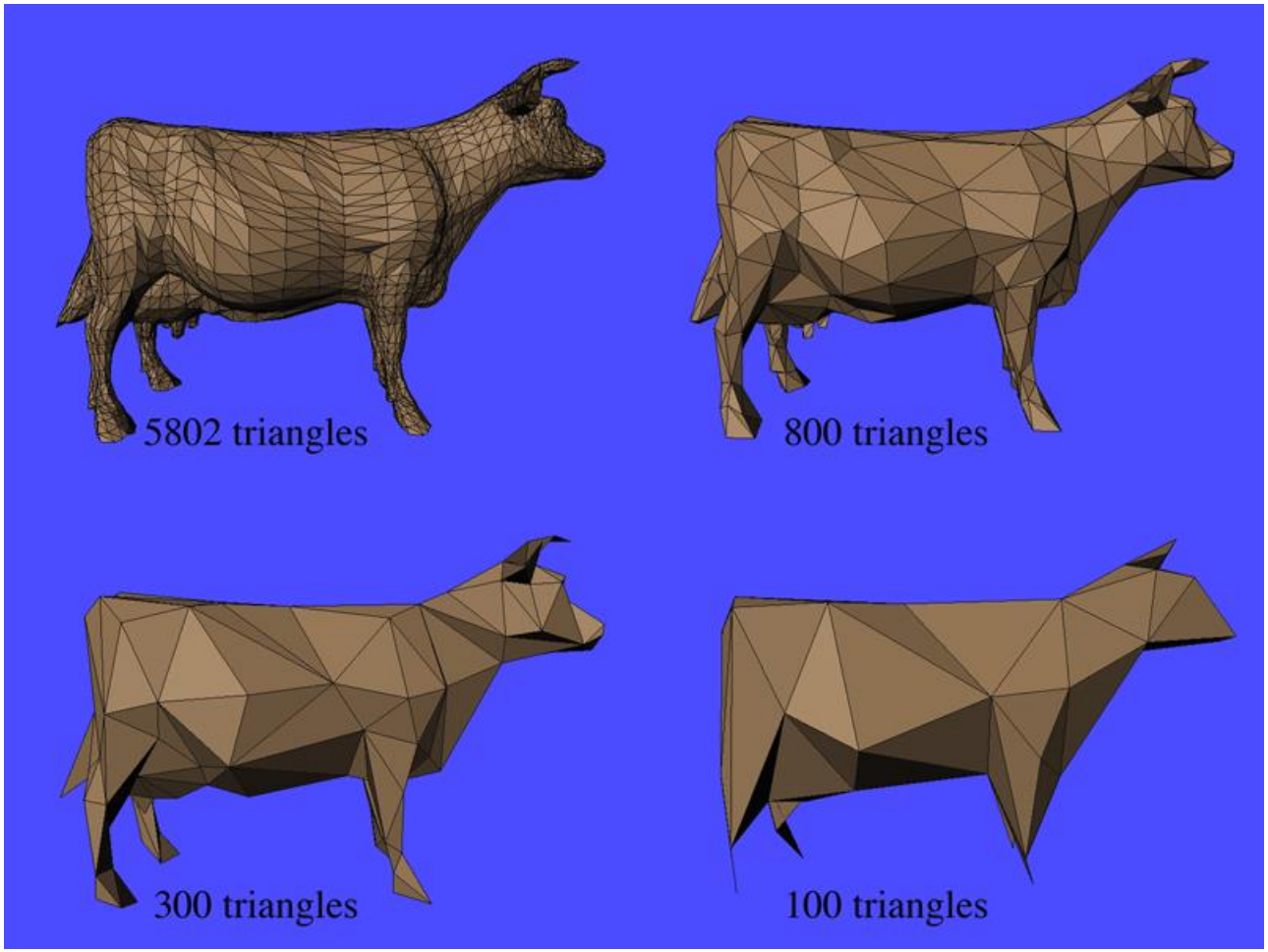
Polyhedral models

- The same polyhedral model might be represented by different polygonal meshes !!
  - Useful for shading / rendering
- Degrees of freedom
  - Number of mesh vertices
  - Distribution of mesh vertices
  - Arrangement of edges / polygons
- Example
  - Represent a cube using different polygonal meshes

Curved surfaces

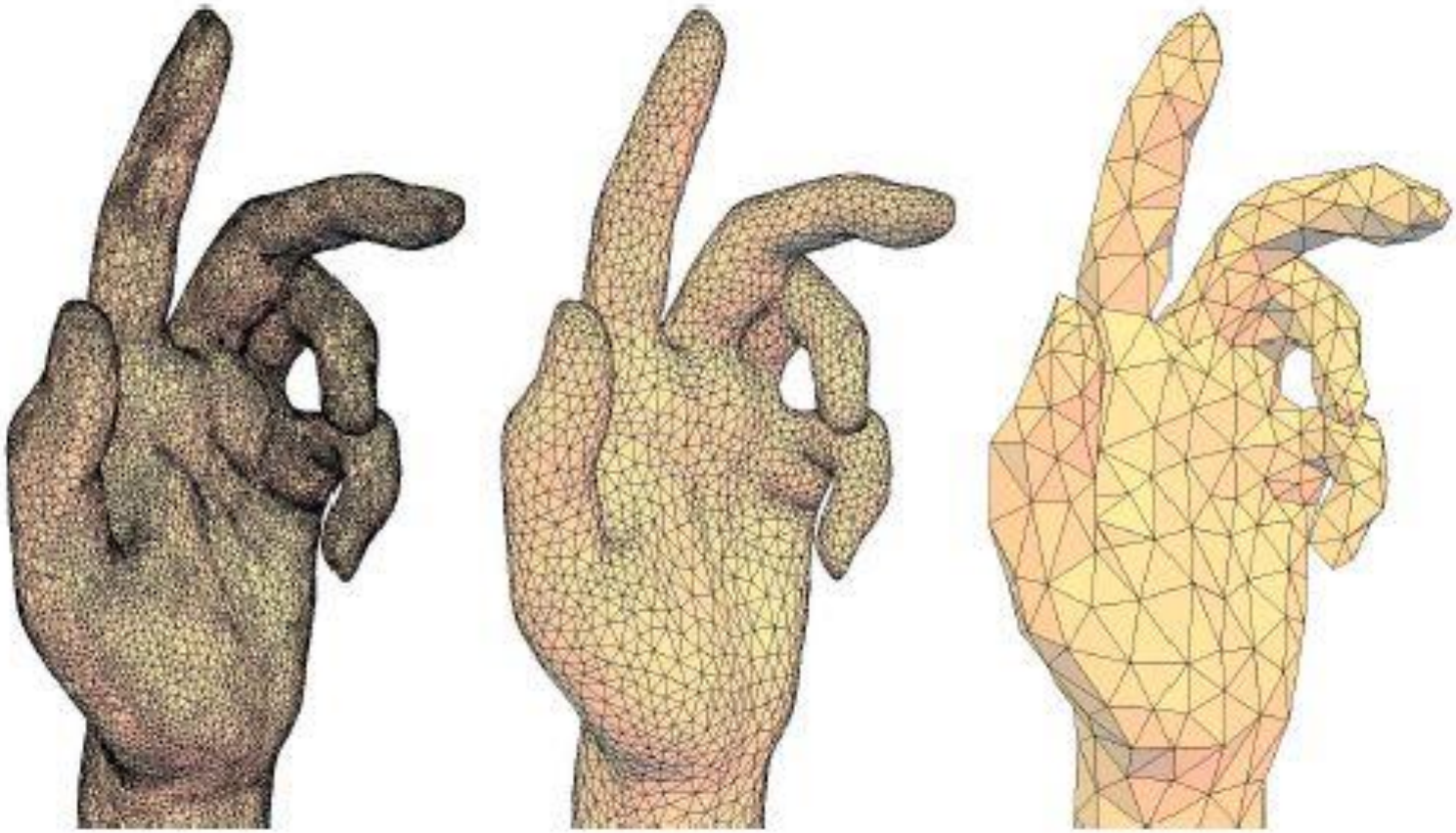
- Representing the shape of a curved surface is an approximation process
- There is no “unique” model!!
- Degrees of freedom
  - Number of mesh vertices
  - Distribution of mesh vertices
  - Arrangement of edges / polygons

How many triangles should be used?



[CMU, 2000]

How many vertices should be used?



(a) 25,000 vertices. (b) 5,000 vertices. (c) 500 vertices.

[Dyer et al.]

Criteria

- Smoothness
  - Differential geometry
  - Curvature ?
  - Triangle quality ?
  - …
- Complexity
  - Number of vertices / polygons
  - Memory space / File size
  - Computational cost of usual operations

Criteria – Restrictions

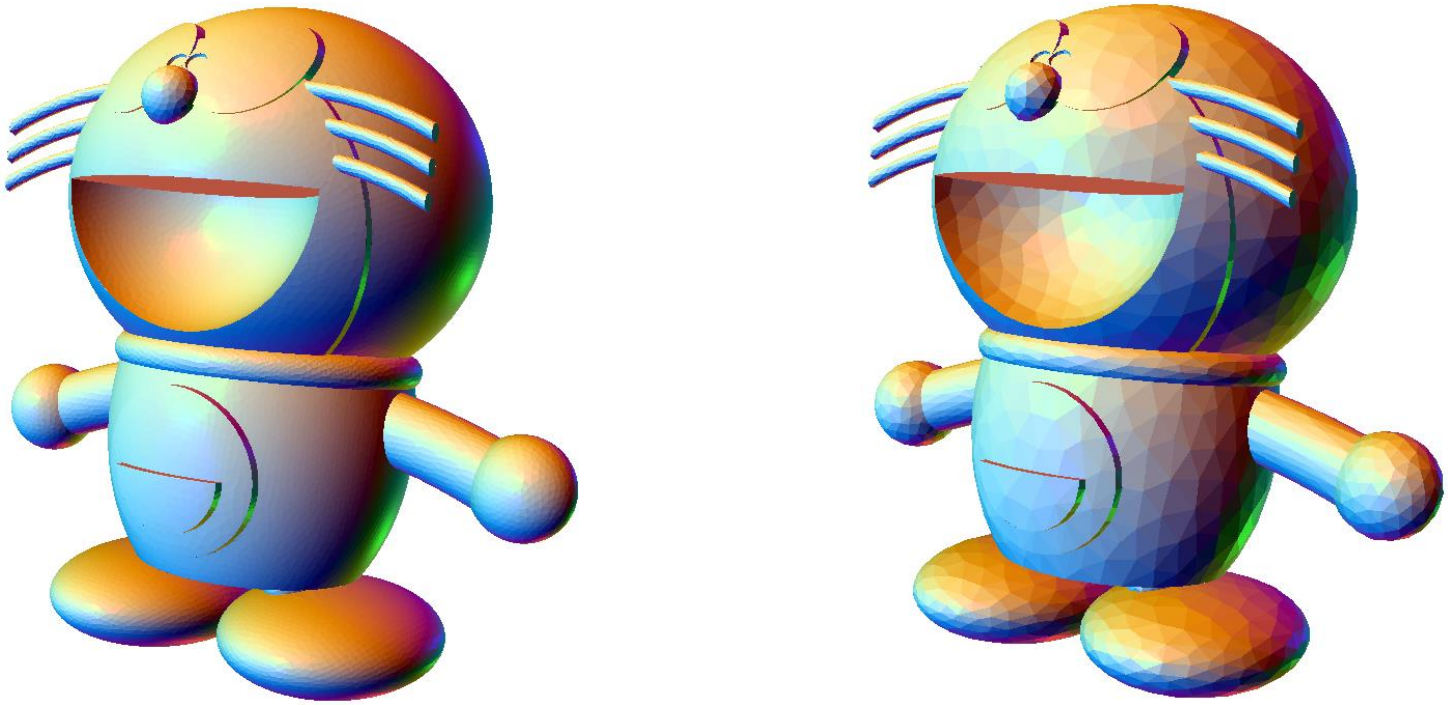
- Least admissible smoothness
  - Screen resolution ?
  - Perception ?
  - User studies
- Largest admissible complexity
  - Processing / rendering speed
  - Memory space / File size
- Balance ?

How to adjust?

- Refinement
  - Increase surface smoothness !!
  - How to compute new vertices and polygons ?
  - Where ?

- Decimation
  - Decrease the number of vertices / polygons !!
  - Which edges / polygons should be colapsed ?
  - Where ?

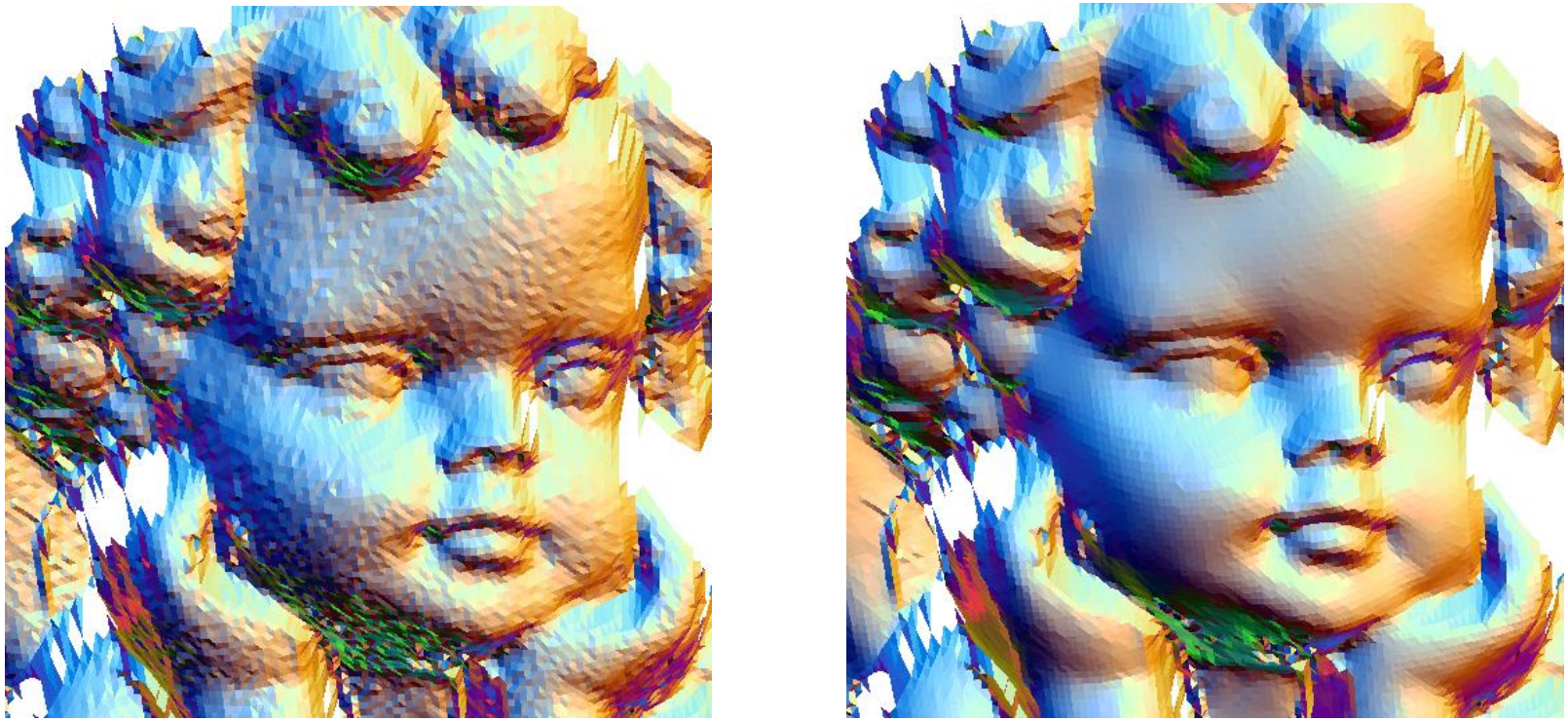
Mesh decimation



90% reduction

[Seidel and Belyaev, 2006]

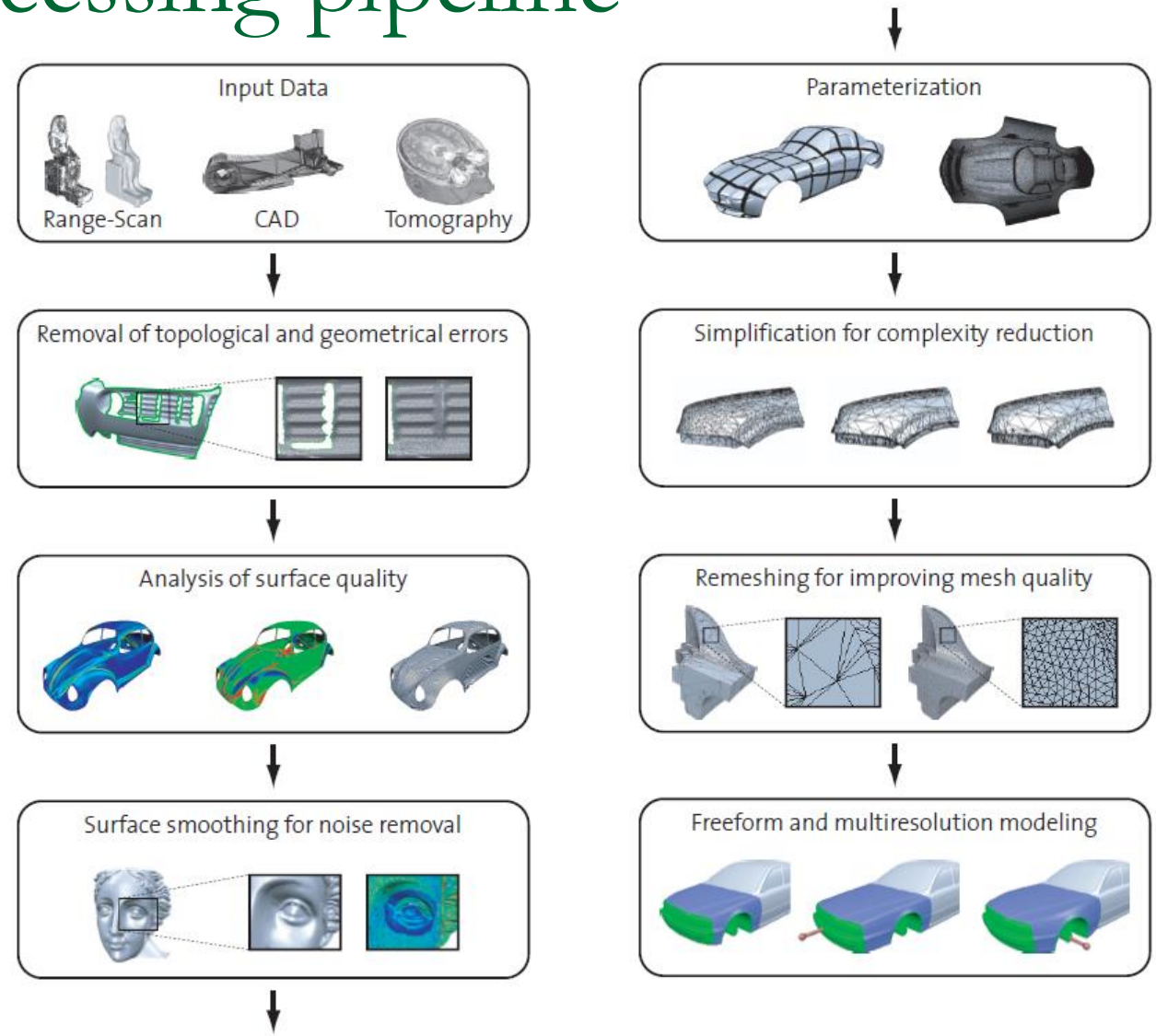
Mesh smoothing



[Seidel and Belyaev, 2006]

THE MESH PROCESSING PIPELINE

Processing pipeline



[Leif Kobbelt]

Mesh processing pipeline

- Acquiring a "Point Cloud"
  - 3D digitization
    - Laser scanning
    - …
  - Numerical simulation
  - Volumetric data
- Creating a geometric model
  - Triangulation

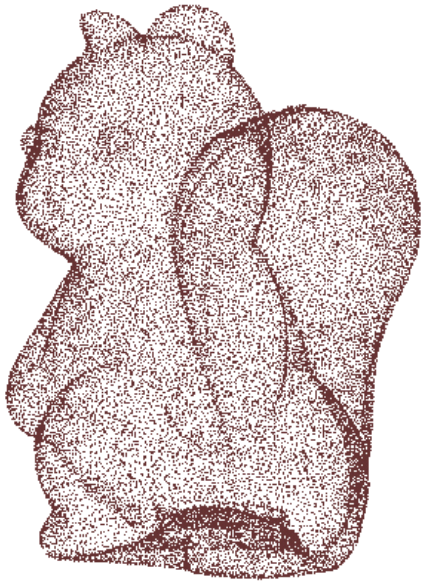
Mesh processing pipeline

- Checking the triangle mesh
  - E.g., “hole filling”
  - Manual editing ?
- Analyzing surface quality
  - Triangle quality
  - Curvatures
  - …
- Refinement ? / Decimation ?

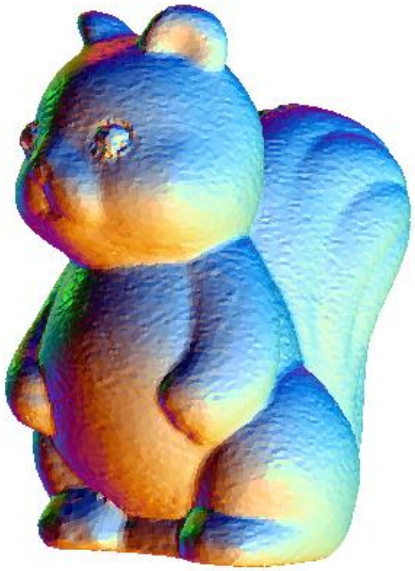
Digitizing

Real world object

Cloud of points



Digitized shape

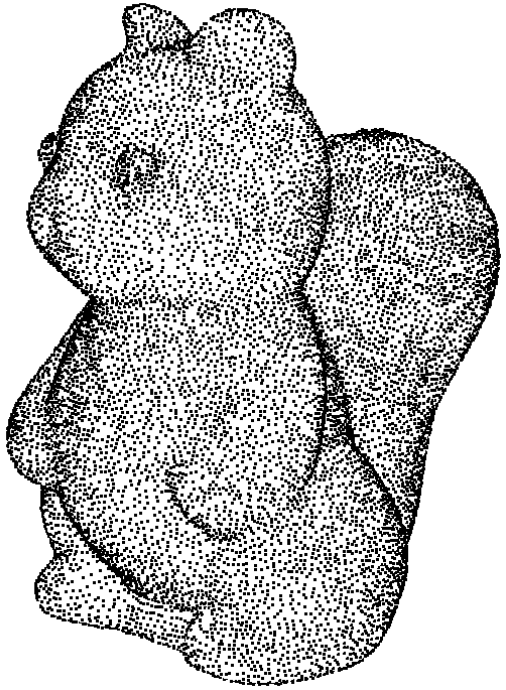


[Seidel and Belyaev, 2006]

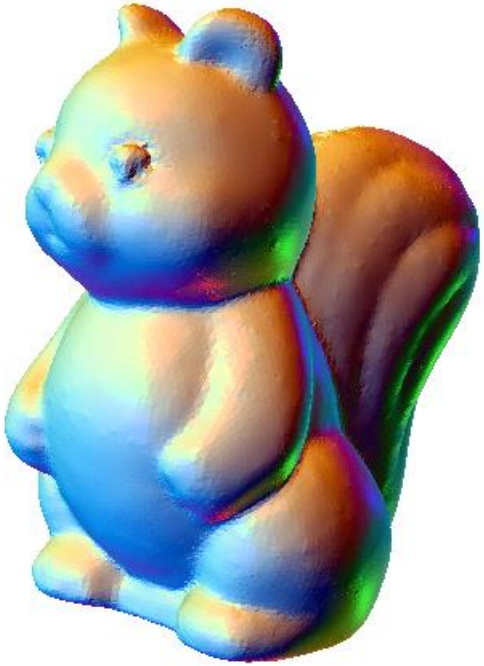
Digitizing

An Unorganized Set of Points

A Mesh or Implicit Surface



Reconstruction



[Seidel and Belyaev, 2006]

Application areas

- Mechanical Engineering
  - Shape Interrogation
    - Digitize a manufactured item and compare it to the initially designed 3D model
  - Reverse Engineering
    - Digitize an item to create its 3D model

- Civil Engineering
  - Surface Reconstruction
    - Create 3D models for large builduings / monuments
    - Analyze ageing, …

Application areas

- Medicine
  - Diagnosis
    - 3D models from CAT / MRI / … data
  - Simulation / Surgical training
    - Virtual patients
  - Surgical planning
    - Prosthetics

- E-Commerce
  - Send 3D models to clients
  - Virtual showrooms
    - Clothes / glasses / …
    - Cars

Application areas

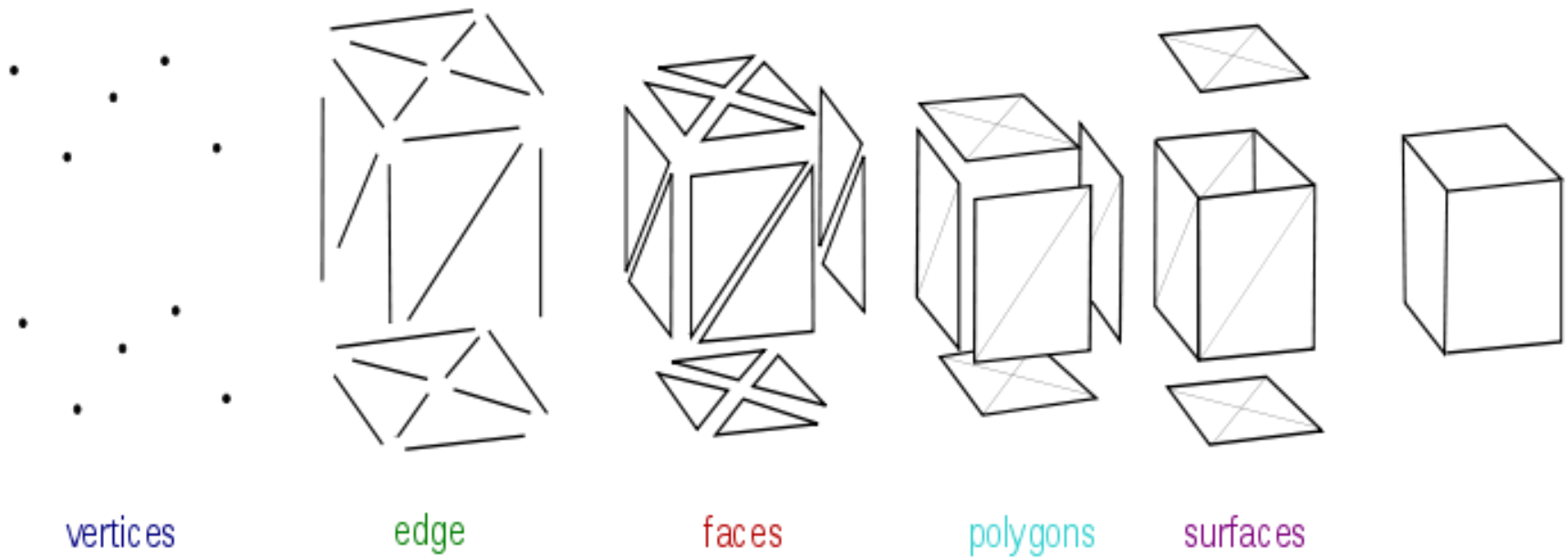
- Computer games
- Animation
- Movies
  - Special effects
  - Virtual actors

GEOMETRY + TOPOLOGY

Representing model surfaces

- Geometrical information
  - Vertex coordinates
- Topological or connectivity information
  - Abstract definition of vertices, edges and faces
  - Incidence and adjacency information
- Properties
  - Normal vectors (“Normal Maps”)
  - Texture coordinates

Topological information



[Wikipedia]

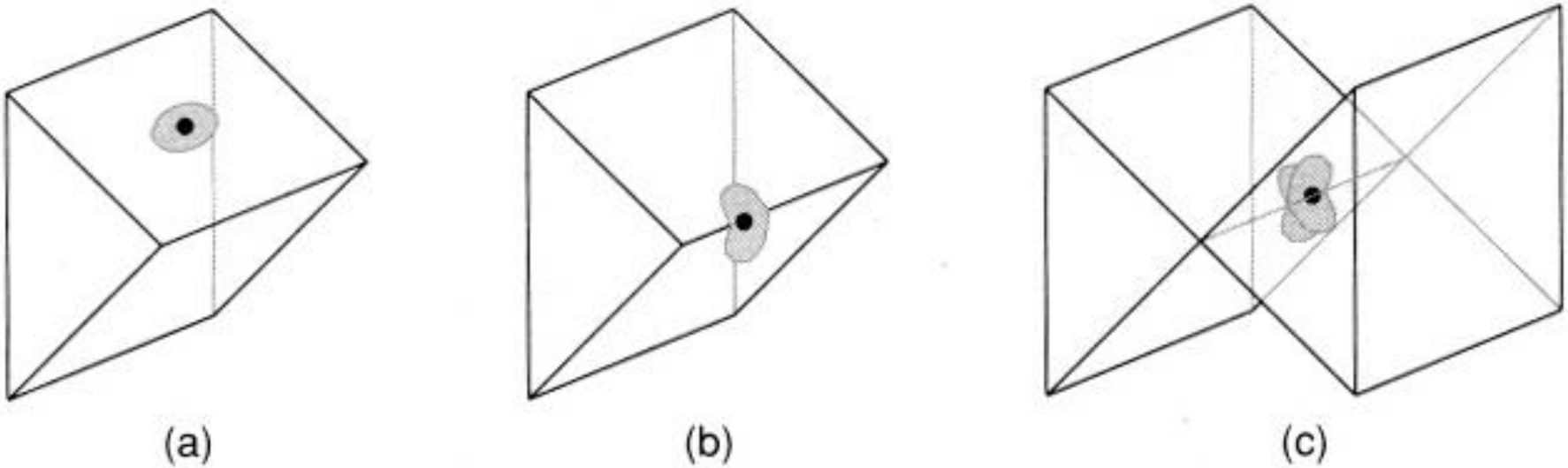
Topological information

- Vertex
  - Regular ?
  - Singular ?
- Edge
  - 2 vertices
  - Border edge : just 1 incident face
  - Regular edge : 2 incident faces
  - Singular edge : 3 or more incident faces
- Loop
  - Ordered edge sequence

Topological information

- Face
  - Limited by a set of disjoint edge sequences
  - Outer border
  - Possible inner borders (“holes”)
- Shell
  - Set of connected faces
- Examples ?

Valid vs. non-valid models



[Foley et al.]

Valid vs. non-valid models

- 2-Manifold Model
  - Any point has a “disk” neighborhood
  - No singular vertices !!
  - No singular edges!!
- Non-Manifold Model
  - Dangling Edges / Faces
  - Touching Faces
  - …
  - Non-valid / non-manufacturable models !!
- Examples ?

Euler Formula

- Allows checking the consistency of the topological information !!
- $V + F - E = 2$
- When to apply?
  - Model has a closed, orientable surface !
  - Each face is limited by a single edge loop !
  - No through-holes !
  - Nor cavities !
- Examples
  - Tetrahedron
  - Different mesh representations of a cube

Euler-Poincaré Formula

- Generalization !!
- $V + F - E - (L - F) - 2 (S - G) = 0$
- L – Number of loops
- S – Number of shells
- G – Genus : number of “handles”
- When to apply ?
  - Through-holes
  - Cavities
- Example ?

Consistency checking

- Check if
  - All polygons have closed borders
  - All edges are used at least once
  - Every vertex belongs at least to
    - 2 edges
    - 1 polygon
  - …

COMPUTATIONAL REPRESENTATION

Computational representation

- Memory or file ?
- Vertices list
  - Topological information ??
- Polygons list / Detached triangles
  - How to identify neighbors ??
- Vertices, edges and polygons lists
  - Efficiency ?
- Winged-edge or half-edge data structures

Detached / Isolated Polygons List

- Each polygon is represented by the ordered list of its vertices coordinates
  - CCW

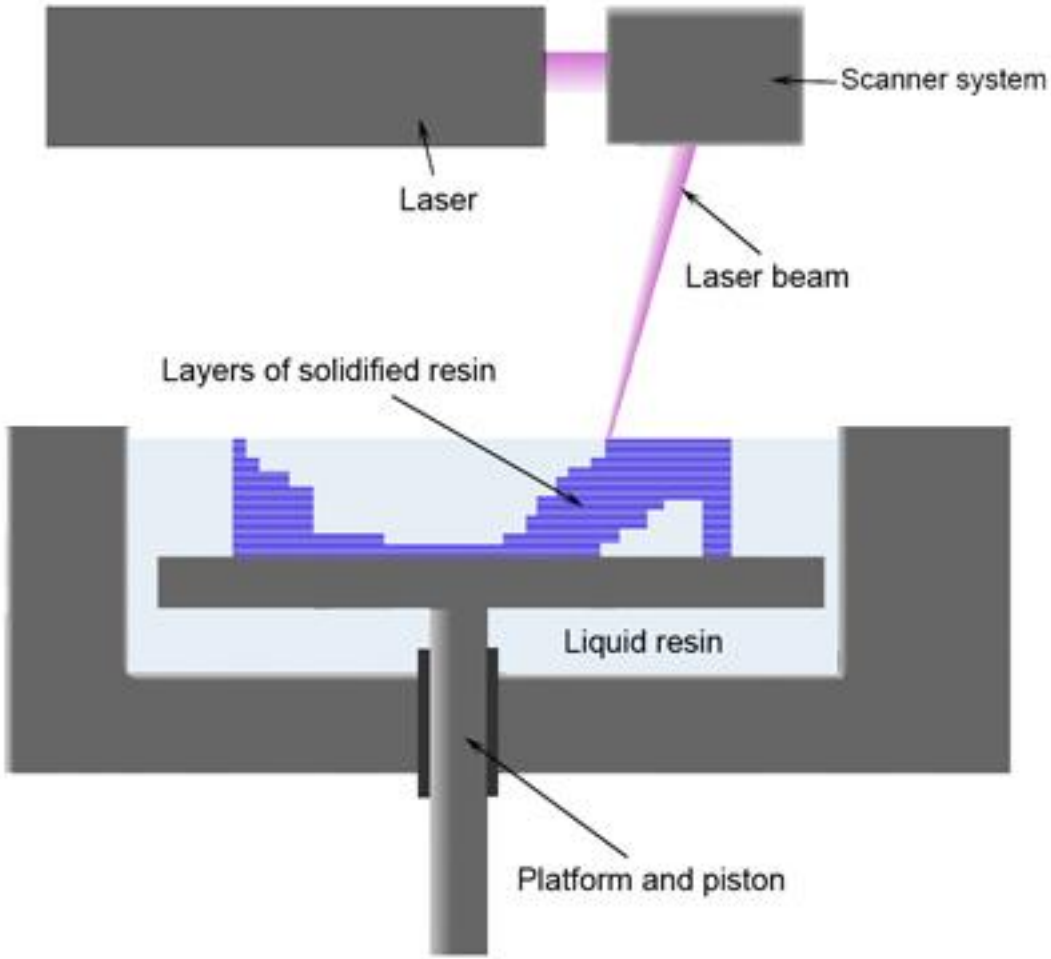
- Inefficient !!
  - Memory space : multiple vertex representation
  - Lack of information about shared vertices / edges
  - Cumbersome detection !!
  - Rendering : edges are drawn twice !!

- Example ?

STL File Format

- Stereolitography File Format
- Mesh defined by T detached triangles
  - Each triangle defined by 3 vertices
  - Unit normal vector for each triangle
- Total : 12T real values
- Usage
  - Rapid prototyping
  - CAM

Stereolithography – 3D Printing



[Wikipedia]

Vertices List

- Vertices list / array
  - Store just once the coordinates of each vertex !
  - Easy to edit / modify one vertex
- Each polygon is described by its vertices sequence
  - Pointer / index
  - Usage : storing in a file
- Inefficient !!
  - Hard to detect which polygons share a given edge !!
  - Rendering : edges are drawn twice !!
- Example ?

Indexed Face Set

- VRML or MCGL or …
- Array 3D vertex coordinates
  - One index for each vertex
- Convex n-sided polygons defined by n indices
- Example
  - [0,1,2,-1,2,1,3,4,-1]

OBJ File Format

- Vertices list

```
v 10 15 20
v 23 34 56
…
```

- Faces list

```
f 1 2 3
f 2 3 4
…
```

- Additional information
  - Normal vectors
  - Texture coordinates
  - …

Topological information

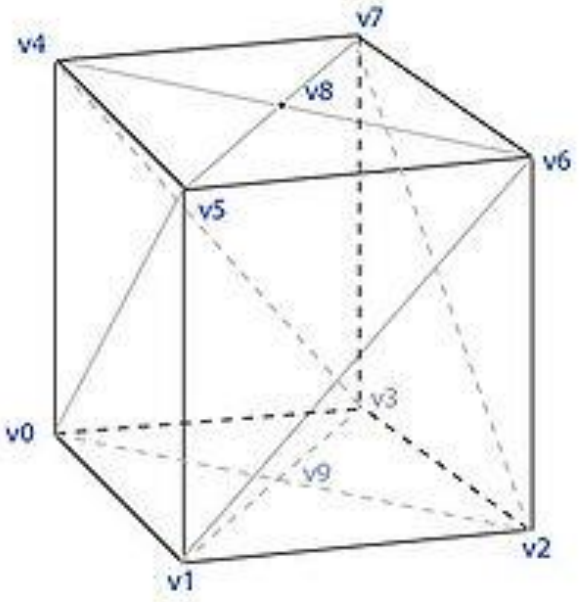
- How to store incidence and adjacency information ?
- How to answer basic queries fast ?
  - Which are the end vertices of a given edge ?
  - Which are the adjacent polygons of a given edge ?
  - Which are the incident edges in a given edge ?
  - Which are the incident edges in a given vertex?
  - Which are the neighboring vertices of a given vertex ?
  - …
- Efficiency
  - Time ?
  - Space ?

Adjacent Vertices List

Vertex-Vertex Meshes (VV)

Vertex List

v0	0,0,0	v1 v5 v4 v3 v9
v1	1,0,0	v2 v6 v5 v0 v9
v2	1,1,0	v3 v7 v6 v1 v9
v3	0,1,0	v2 v6 v7 v4 v9
v4	0,0,1	v5 v0 v3 v7 v8
v5	1,0,1	v6 v1 v0 v4 v8
v6	1,1,1	v7 v2 v1 v5 v8
v7	0,1,1	v4 v3 v2 v6 v8
v8	.5,.5,0	v5 v6 v7 v8
v9	.5,.5,1	v0 v1 v2 v3



[Wikipedia]

Vertices List + Faces List

Face-Vertex Meshes

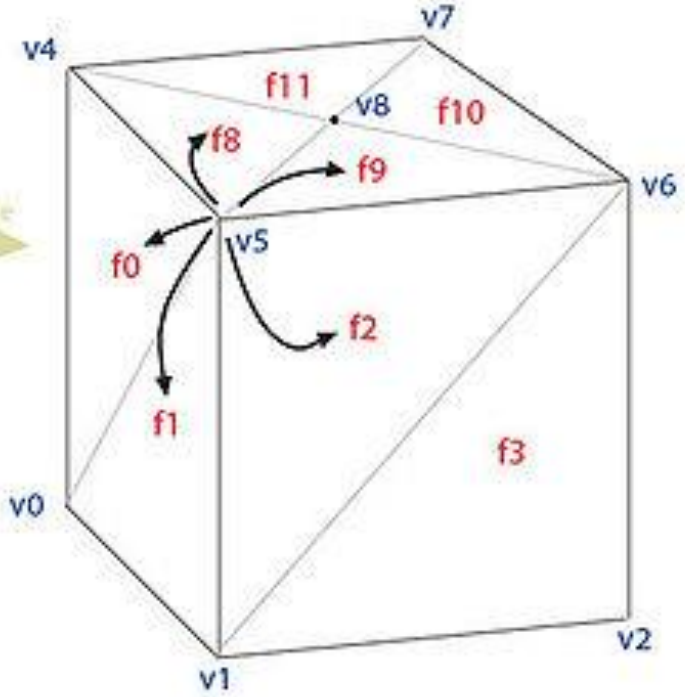
Face List

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]

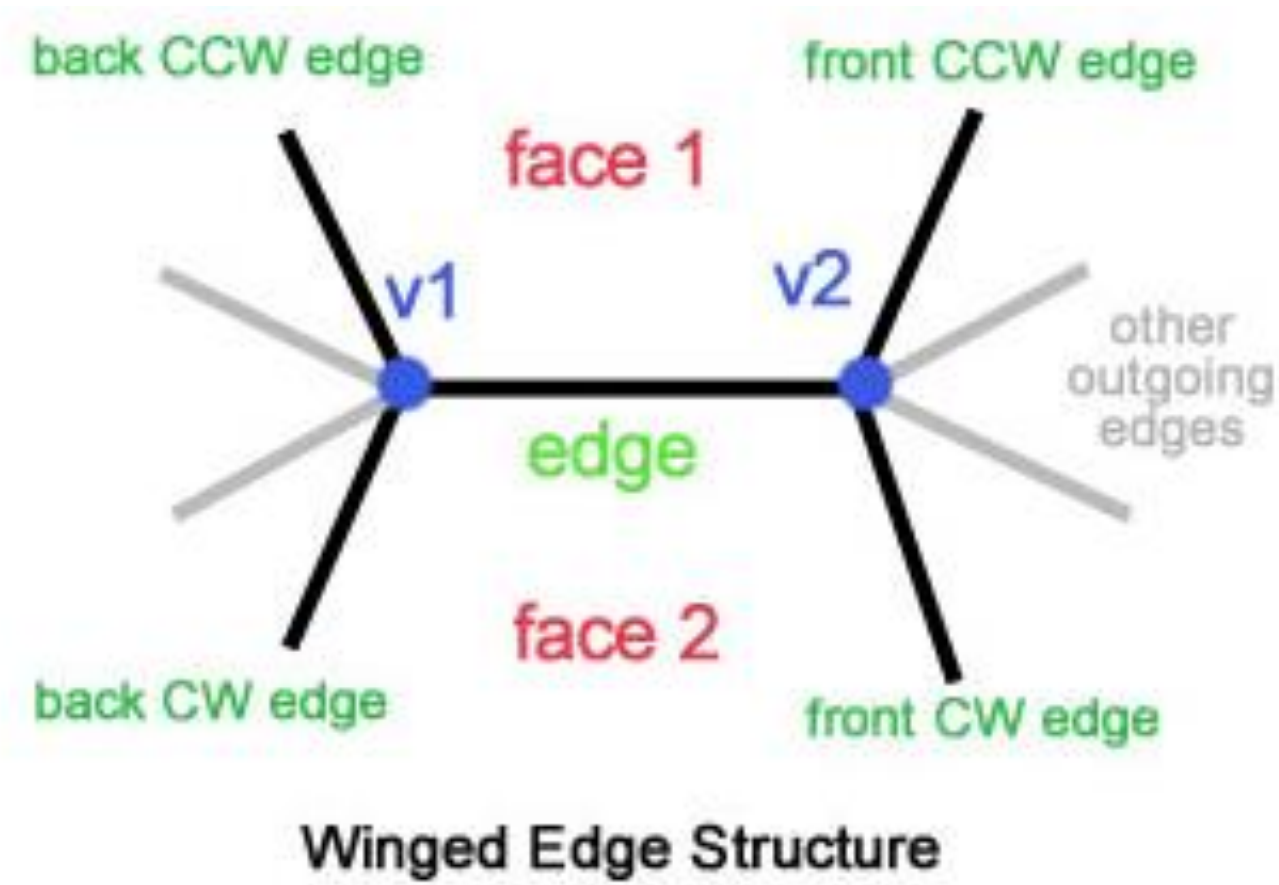
Vertices, Edges and Faces Lists

- Vertices list / array
- Edges list / array, which references
  - The respective end vertices
  - The respective polygons
- Polygons list / array, which references
  - The respective edges
- Rendering : draw edges, not polygons !!
- Issue
  - How to identify the edges incident in a given vertex ?
- Example ?

The “Winged-Edge” data structure

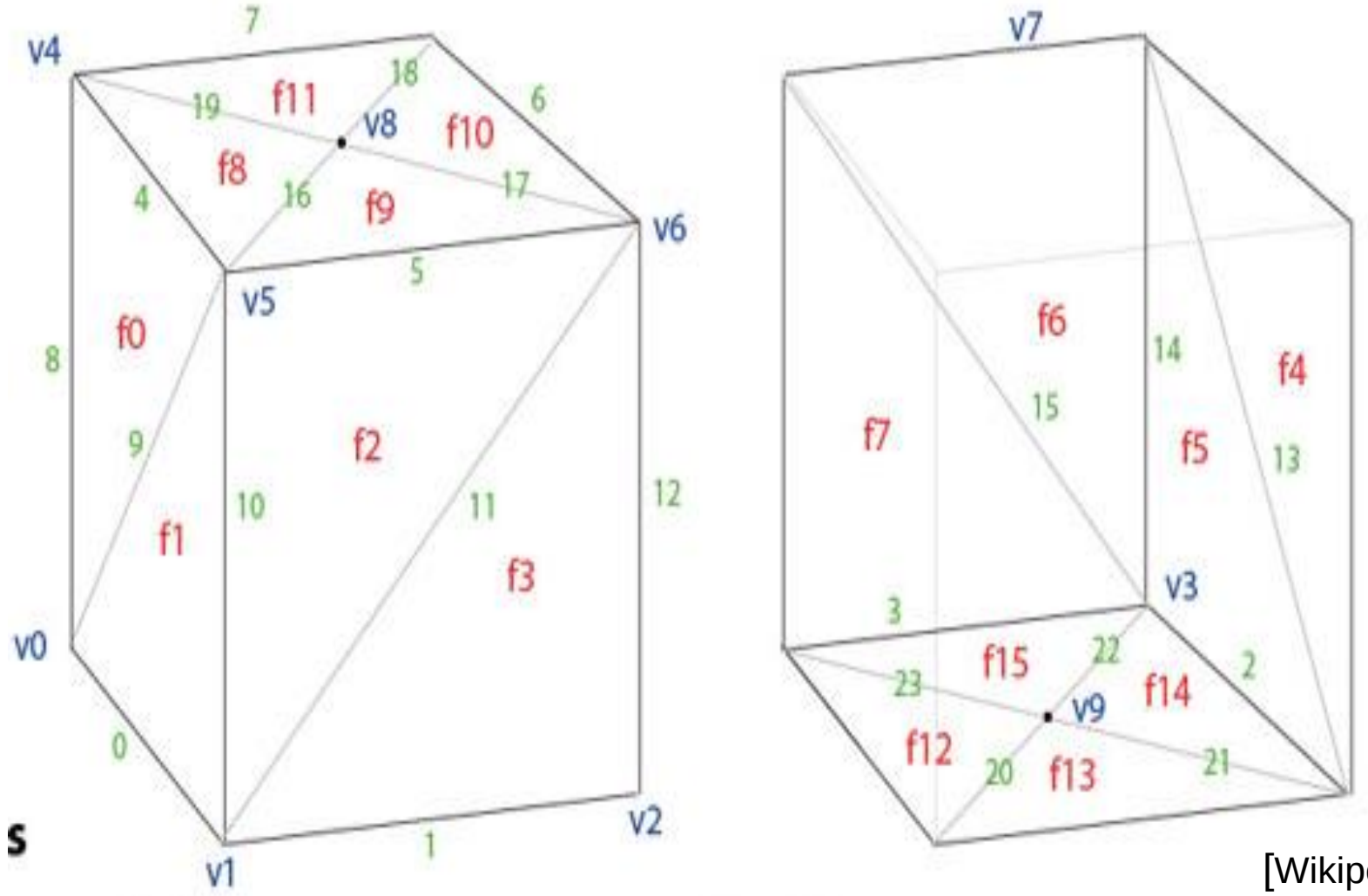
- Explicit representation of vertices, edges and faces
- Allows
  - Dynamic mesh modification
  - Efficient answer to some “queries”

The "Winged-Edge" data structure



[Wikipedia]

The "Winged-Edge" data structure



[Wikipedia]

The “Winged-Edge” data structure

Face List

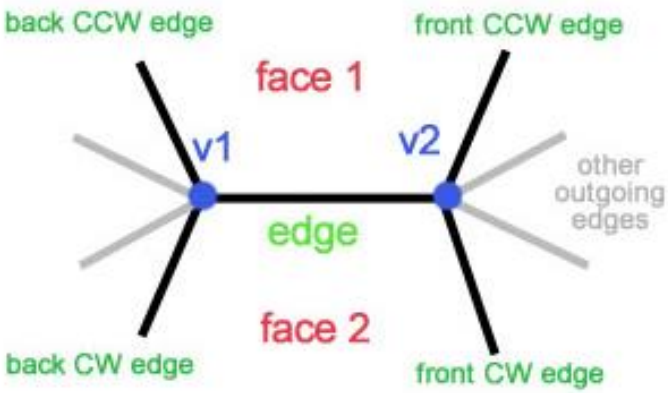
f0	4 8 9
f1	0 10 9
f2	5 10 11
f3	1 12 11
f4	6 12 13
f5	2 14 13
f6	7 14 15
f7	3 8 15
f8	4 16 19
f9	5 17 16
f10	6 18 17
f11	7 19 18
f12	0 23 20
f13	1 20 21
f14	2 21 22
f15	3 22 23

Edge List

e0	v0 v1	f1 f12	9 23 10 20
e1	v1 v2	f3 f13	11 20 12 21
e2	v2 v3	f5 f14	13 21 14 22
e3	v3 v0	f7 f15	15 22 8 23
e4	v4 v5	f0 f8	19 8 16 9
e5	v5 v6	f2 f9	16 10 17 11
e6	v6 v7	f4 f10	17 12 18 13
e7	v7 v4	f6 f11	18 14 19 15
e8	v0 v4	f7 f0	3 9 7 4
e9	v0 v5	f0 f1	8 0 4 10
e10	v1 v5	f1 f2	0 11 9 5
e11	v1 v6	f2 f3	10 1 5 12
e12	v2 v6	f3 f4	1 13 11 6
e13	v2 v7	f4 f5	12 2 6 14
e14	v3 v7	f5 f6	2 15 13 7
e15	v3 v4	f6 f7	14 3 7 15
e16	v5 v8	f8 f9	4 5 19 17
e17	v6 v8	f9 f10	5 6 16 18
e18	v7 v8	f10 f11	6 7 17 19
e19	v4 v8	f11 f8	7 4 18 16
e20	v1 v9	f12 f13	0 1 23 21
e21	v2 v9	f13 f14	1 2 20 22
e22	v3 v9	f14 f15	2 3 21 23
e23	v0 v9	f15 f12	3 0 22 20

Vertex List

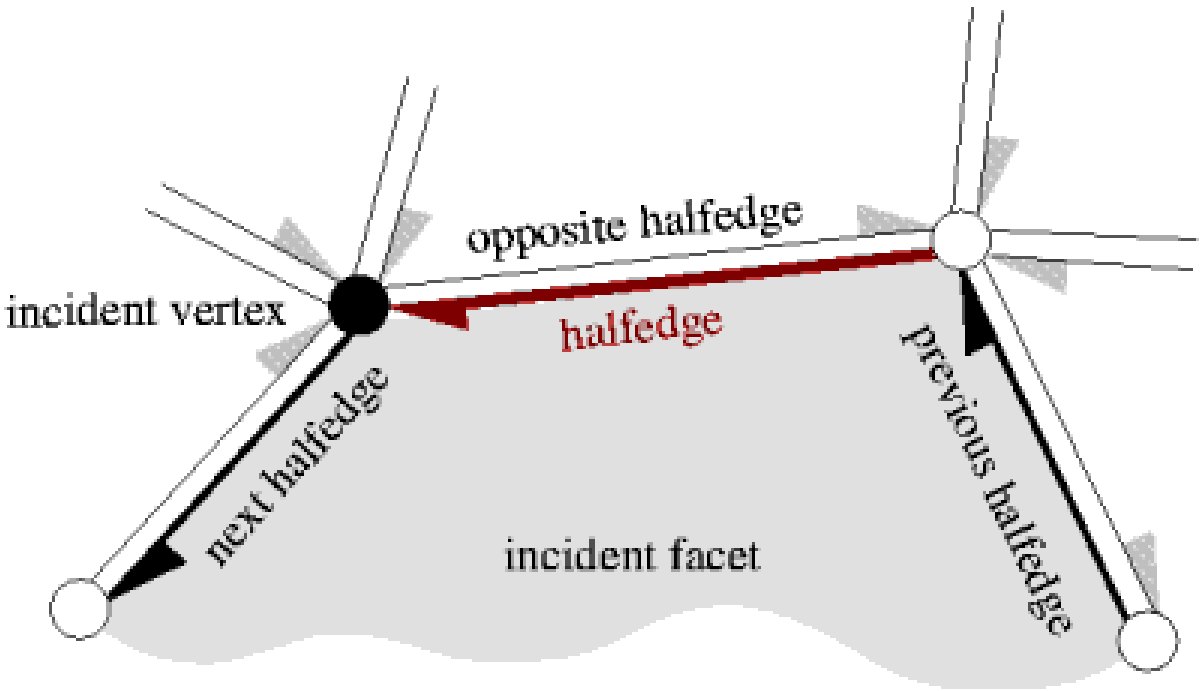
v0	0,0,0	8 9 0 23 3
v1	1,0,0	10 11 1 20 0
v2	1,1,0	12 13 2 21 1
v3	0,1,0	14 15 3 22 2
v4	0,0,1	8 15 7 19 4
v5	1,0,1	10 9 4 16 5
v6	1,1,1	12 11 5 17 6
v7	0,1,1	14 13 6 18 7
v8	.5,.5,0	16 17 18 19
v9	.5,.5,1	20 21 22 23



Winged Edge Structure

[Wikipedia]

The “Half-Edge” data structure



Mesh libraries / toolboxes

- OpenMesh
  - https://www.graphics.rwth-aachen.de/software/openmesh/
- OpenFlipper
  - https://www.graphics.rwth-aachen.de/software/openflipper/
- CGAL – Comp. Geometry Algorithms Library
  - https://www.cgal.org/
- MeshLab
  - https://www.meshlab.net/
- …