CMSC740
Advanced Computer Graphics

Matthias Zwicker
Fall 2025

3D geometry processing

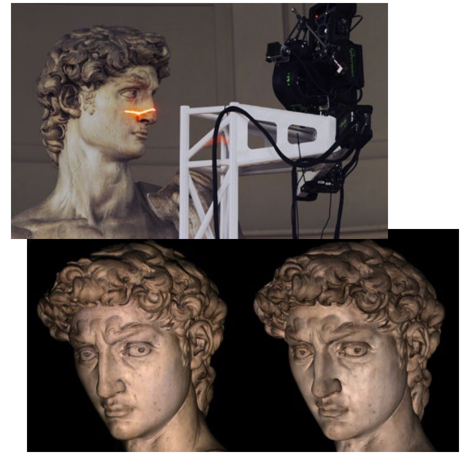
Objectives

- Theory and algorithms to capture, process, manipulate, and analyze 3D shapes

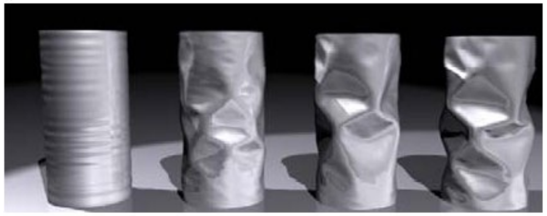
Applications

Medical

E-commerce

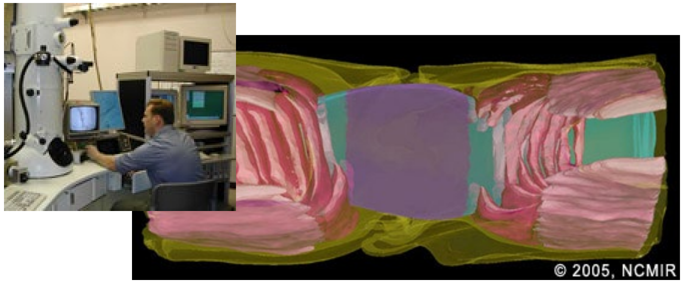


Cultural heritage



Simulation

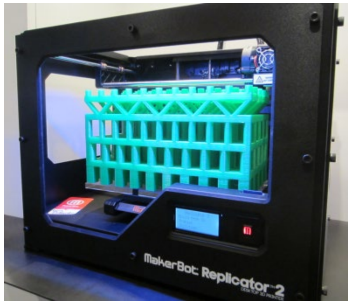
Games, movies



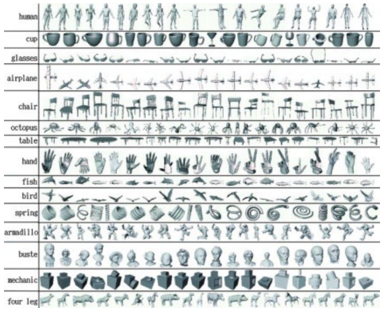
Scientific research, imaging

GIS

Architecture



3D printing, manufacturing



Shape databases

Processing pipeline

Processing pipeline

Acquisition → Processing → Application

Acquisition: raw data

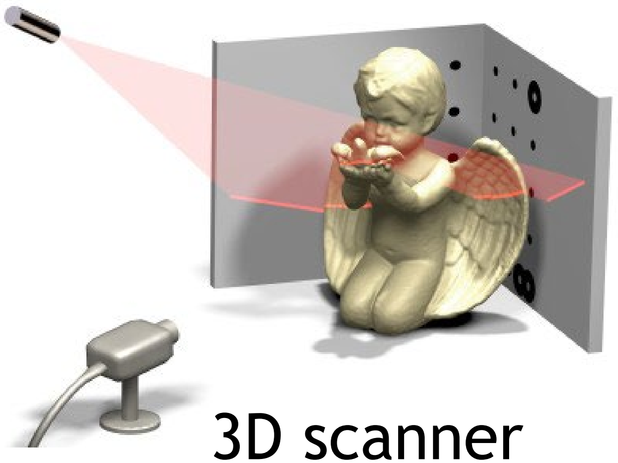
- 3D point clouds

Processing pipeline

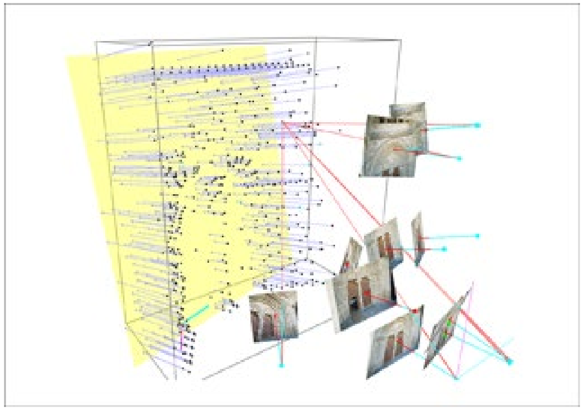
Acquisition → Processing → Application

Acquisition: devices

- 3D scanners (triangulation, time of flight)
- 3D reconstruction from images (computer vision: structure from motion, Multiview stereo)
- 3D imaging technologies (microscopy, MRI, CT)
- Interactive modeling



3D scanner



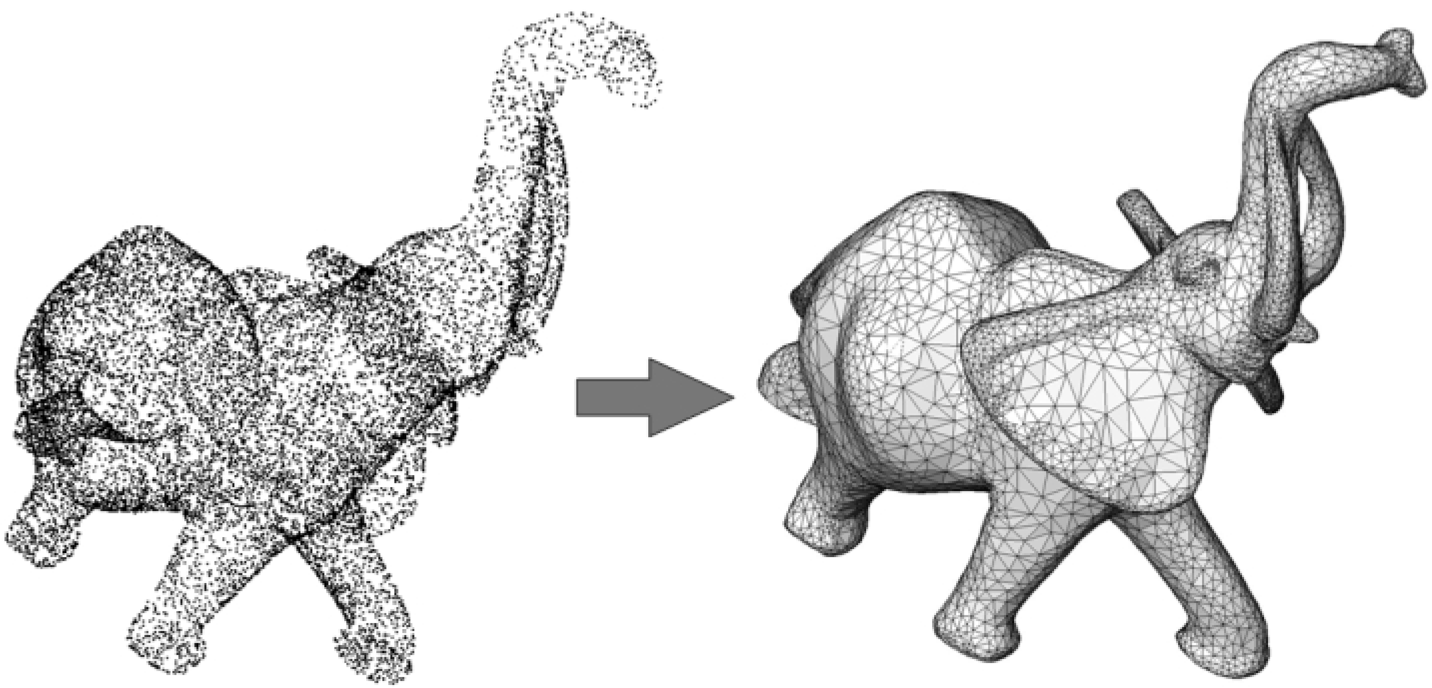
3D reconstruction from images

Processing pipeline

Acquisition → **Processing** → Application

Processing

- E.g., surface reconstruction from points



Processing pipeline

Acquisition → Processing → Application

Applications

- Visualization, rendering (games, movies, design)
- Manufacturing (3D printing)
- Analysis (classification, segmentation)
- Simulation (structural analysis, shape optimization, etc.)
- Databases, retrieval

Main topics

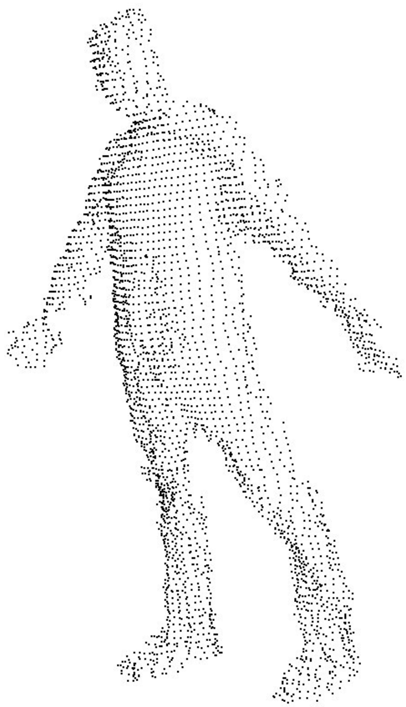
- Surface representations, mathematical background
- Automatic processing operations (reconstruction, segmentation, etc.)
- Generative modeling, editing and manipulation
- Deep learning techniques

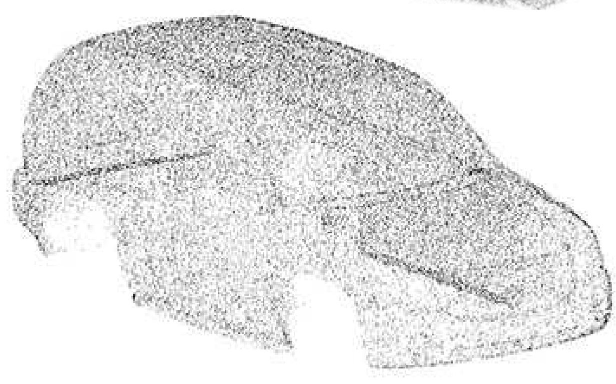
3D shape representations

- Point clouds
- Polygon soups
- Polygon meshes
- Parametric surfaces
- Boundary representations (b-reps)
- Implicit surfaces
- Medial axis

Point clouds

http://en.wikipedia.org/wiki/Point_cloud





Array of 3D points

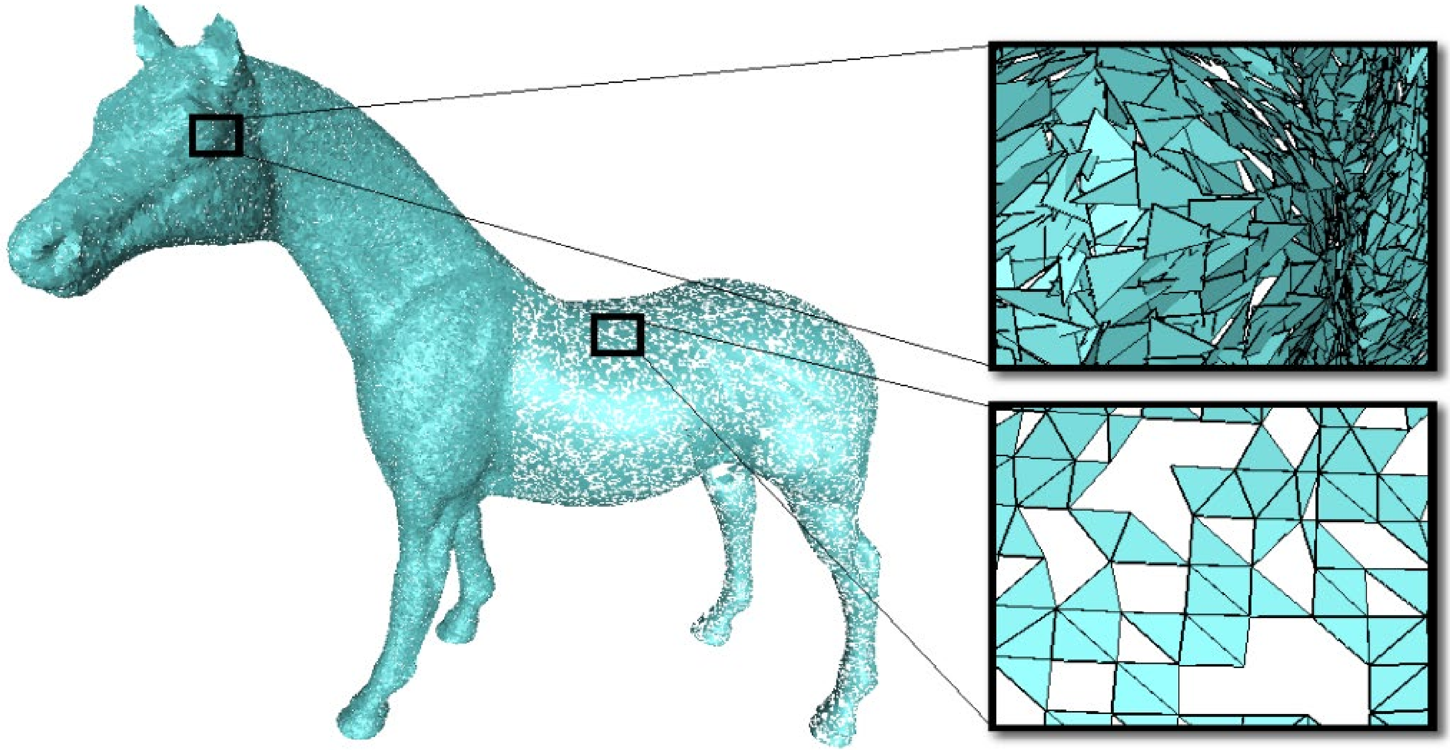
$$((x_0, y_0, z_0),$$
$$(x_1, y_1, z_1),$$
$$\ldots$$
$$(x_{n-1}, y_{n-1}, z_{n-1}))$$

Point clouds

- Often raw output of 3D scanners
- Points can be extended to primitives that locally approximate surface, e.g. ellipses
- Advantages
  - Simple data structure (list of coordinates, or primitives)
  - Render via rasterization (e.g. splatting, draw small ellipse per point)
- Disadvantages
  - Not continuous
  - No notion of topology
  - Need additional structure to apply mathematical operations (compute normals, etc.)

Polygon soup

- Array of separate polygons

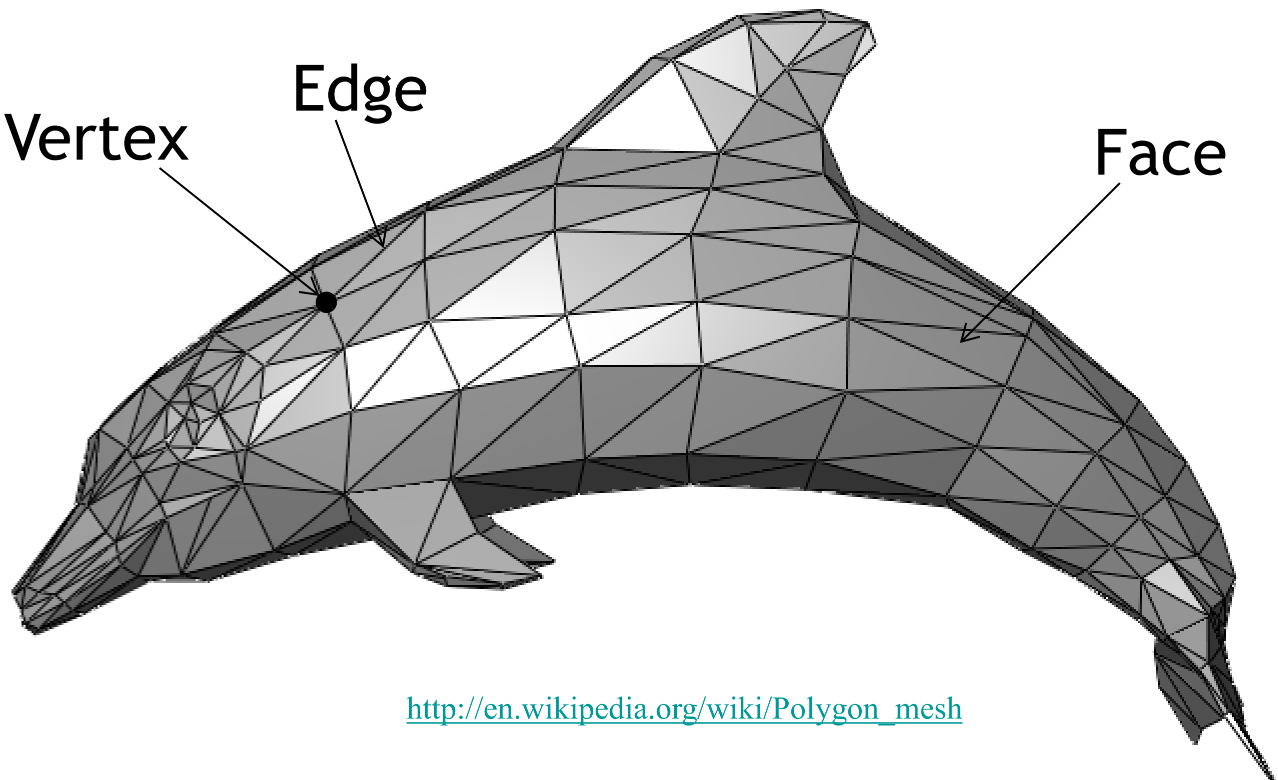


Polygon soup

- Often default output of 3D modeling programs
- Advantages
  - Simple data structure
  - Render via rendering pipeline, rasterization
- Disadvantages
  - Not continuous
  - No notion of topology
  - Not possible to define „watertight“ surfaces

Polygon meshes

- Polygons and how they are connected to a mesh
- Consist of vertices, edges, faces, and their adjacencies



http://en.wikipedia.org/wiki/Polygon_mesh

Polygon meshes

- Face-vertex tables
  - Other data structures possible (e.g., winged-edge data structure https://en.wikipedia.org/wiki/Winged_edge

Face List

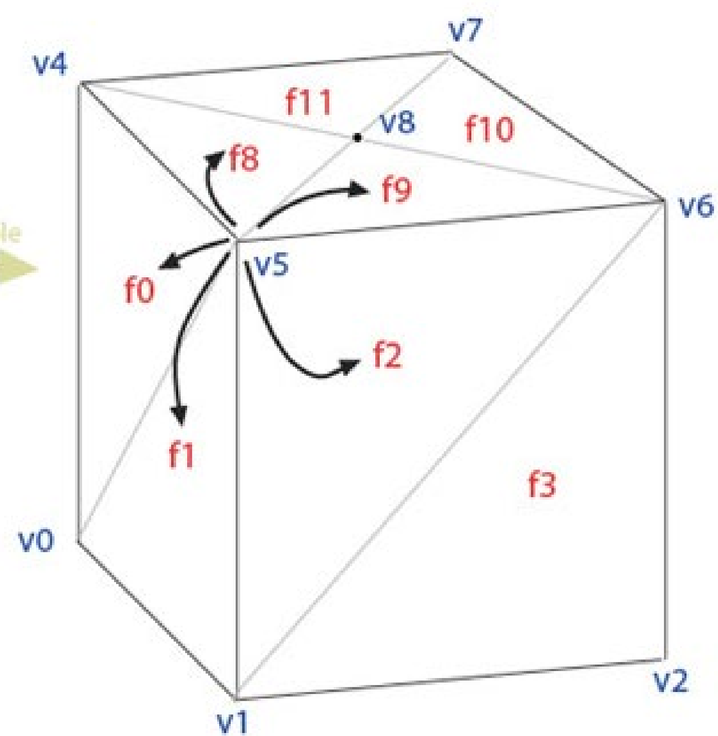
f0	v0 v4 v5
f1	v0 v5 v1
f2	v1 v5 v6
f3	v1 v6 v2
f4	v2 v6 v7
f5	v2 v7 v3
f6	v3 v7 v4
f7	v3 v4 v0
f8	v8 v5 v4
f9	v8 v6 v5
f10	v8 v7 v6
f11	v8 v4 v7
f12	v9 v5 v4
f13	v9 v6 v5
f14	v9 v7 v6
f15	v9 v4 v7

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

x,y,z Optional, often omitted

example

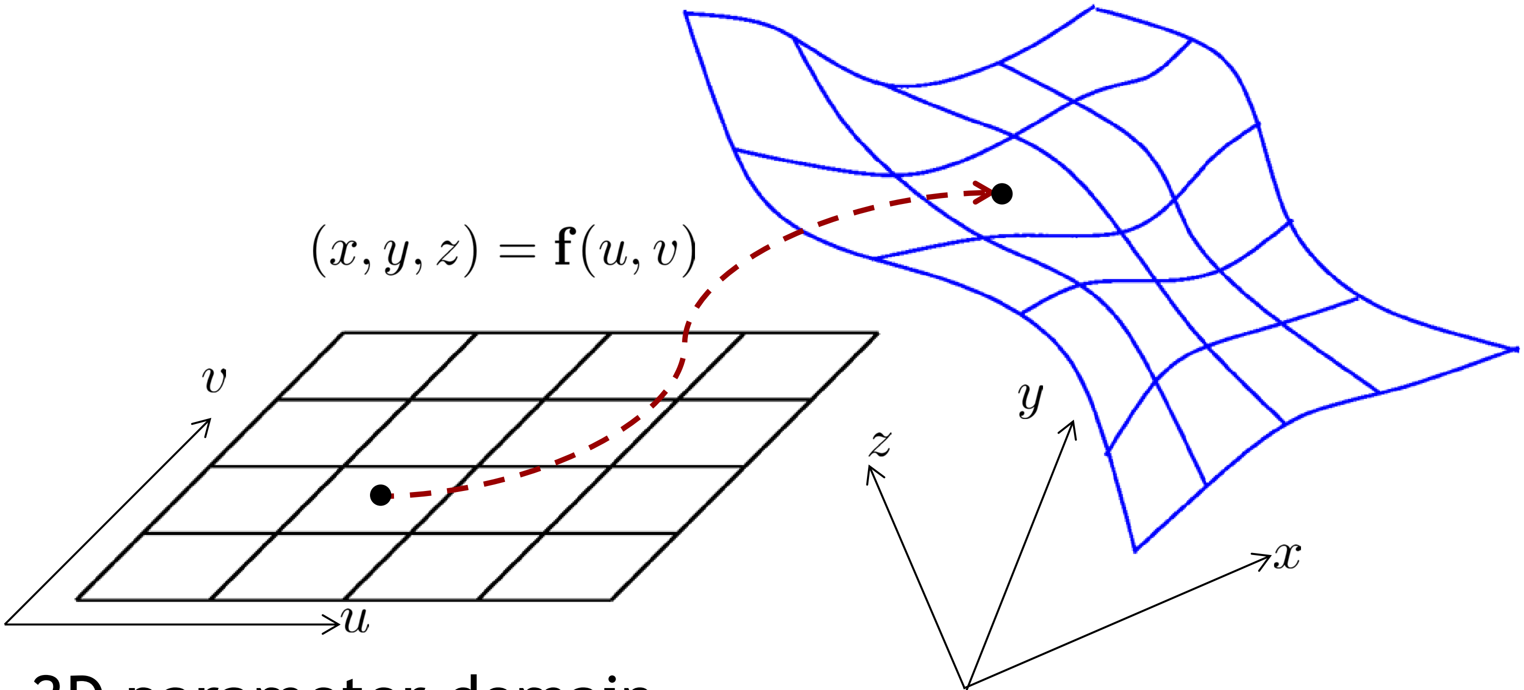


http://en.wikipedia.org/wiki/Polygon_mesh

Polygon meshes

- Main representation used in this class
- Advantages
  - Render via rasterization
  - Can make continuous and watertight surfaces (no cracks nor gaps)
  - Includes topology
  - Easy to make complex shapes
- Disadvantages
  - Not smooth
  - But: can obtain smoothness via subdivision
    http://en.wikipedia.org/wiki/Subdivision_surface

Parametric surfaces



2D parameter domain

$$\mathbf{f} : \mathrm{R}^2 \rightarrow \mathrm{R}^3$$

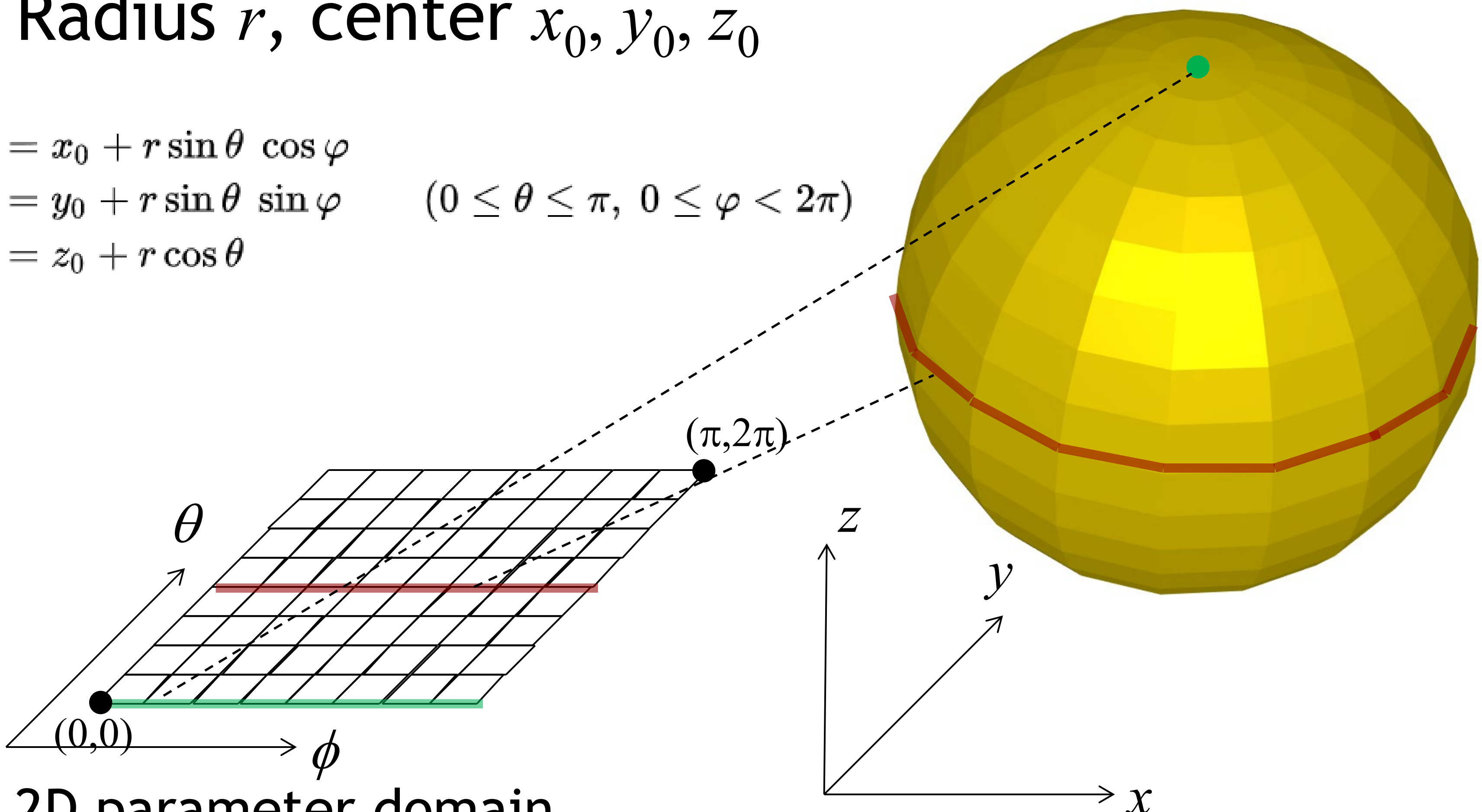
Alternative notation: three separate functions

x: $\mathbf{R}^2 \rightarrow \mathbf{R}^3$, y: $\mathbf{R}^2 \rightarrow \mathbf{R}^3$, z: $\mathbf{R}^2 \rightarrow \mathbf{R}^3$

Example: parametric sphere

- Spherical angles θ (elevation), ϕ (azimuth)
- Radius r, center x_0, y_0, z_0

$$
\begin{aligned}
x &= x_0 + r \sin\theta \ \cos\varphi \\
y &= y_0 + r \sin\theta \ \sin\varphi \qquad (0 \le \theta \le \pi,\ 0 \le \varphi < 2\pi) \\
z &= z_0 + r \cos\theta
\end{aligned}
$$



2D parameter domain

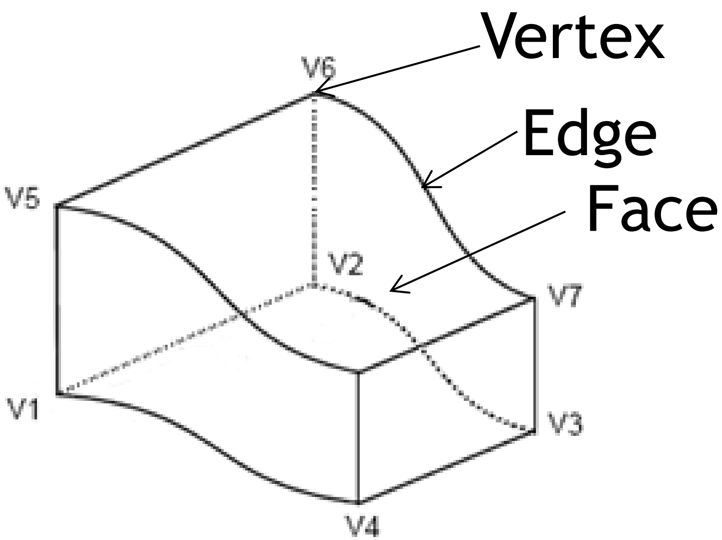
Parametric surfaces

- Most CAD programs built on parametric surfaces
  - NURBs (non-uniform rational b-splines) http://en.wikipedia.org/wiki/Non-uniform_rational_B-spline
- Advantages
  - With explicit definition of functions **f**, many mathematical operations (e.g., derivatives) and properties (e.g., continuity) well defined
  - Convenient for manual modeling
- Disadvantages
  - Given arbitrary shape, fitting parametric surface is non-trivial
  - Complex shapes require stitching together multiple parametric patches
  - Some editing operations such as cutting, merging complicated to implement

Boundary representations

http://en.wikipedia.org/wiki/Boundary_representation

- Consists of connected vertices, edges, faces
- Faces can be
  - Planar polygons, such as triangles
  - Curved parametric patches (NURBS)
- Edges can be
  - Straight line segments
  - Smooth curves



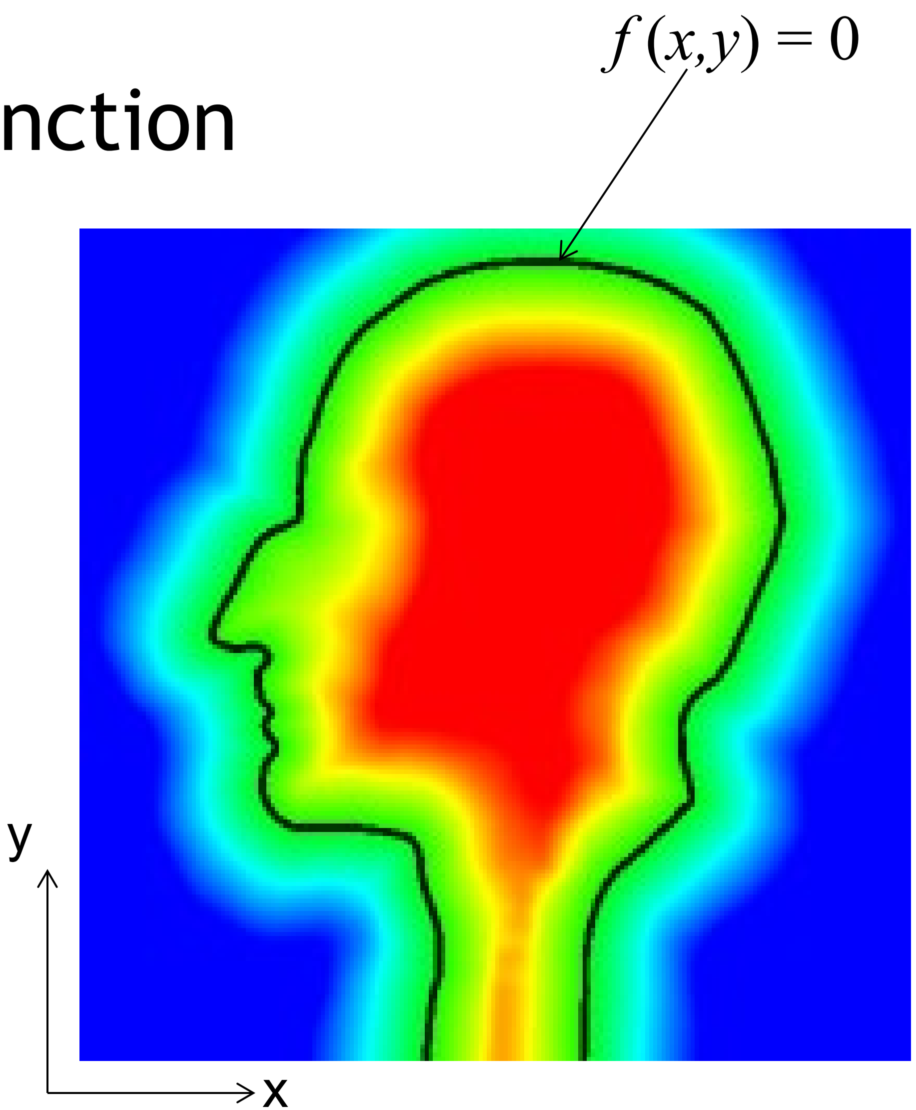
Implicit surfaces

- 2D example
- Curve defined by function

$f: \mathrm{R}^2 \to \mathrm{R}$

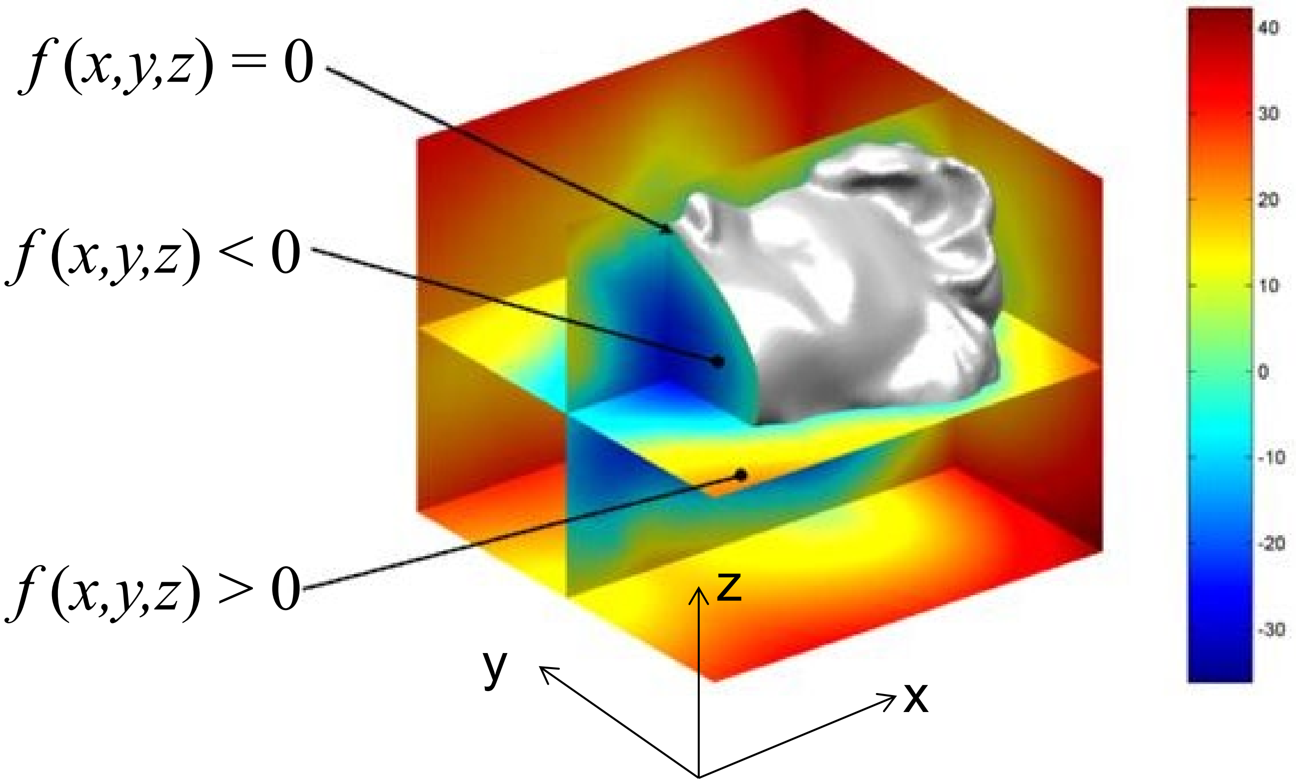
as set of points

$\{(x,y) : f(x,y) = 0\}$



Implicit surfaces

- 3D visualization, where $f: R^3 \rightarrow R$



Example: implicit sphere

- Radius r, center x_0, y_0, z_0
- Points on sphere given by set

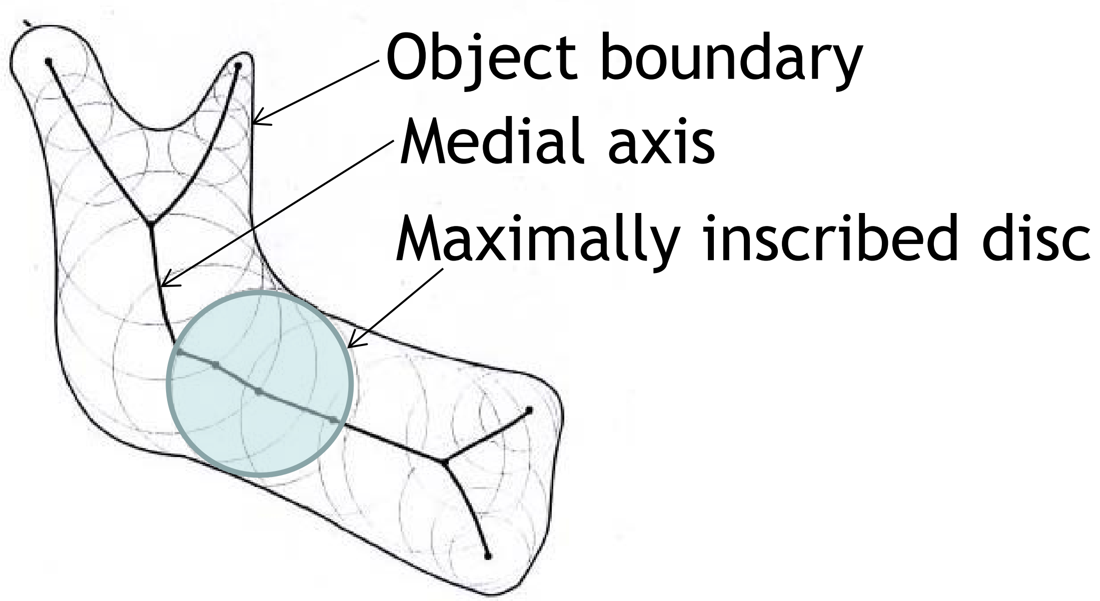
$$\{(x,y,z) : (x - x_0)^2 + (\mathrm{y} - y_0)^2 + (\mathrm{z} - z_0)^2 - r^2 = 0\}$$

Implicit surfaces

- Many alternatives to represent implicit function $\mathbf{f}$
  - Splines (grid of control points with polynomial interpolation)
  - Neural networks, etc.
- Advantages
  - With explicit definition of functions $\mathbf{f}$, many mathematical operations (e.g., derivatives) and properties (e.g., continuity) well defined
  - Editing operations such as cutting, merging easy to implement
- Disadvantages
  - For rendering, usually conversion to triangle mesh needed; otherwise using ray casting
    http://en.wikipedia.org/wiki/Ray_casting
  - Given arbitrary shape, fitting implicit surface non-trivial

Medial axis/surface

- Medial axis (2D shapes): set of all points having more than one closest point on the object's boundary
- Medial axis with associated radius function of the maximally inscribed discs is called medial axis transform (MAT)
- MAT can be used to reconstruct original shape



Medial axis/surface

- Medial surfaces of 3D objects can be complicated, non-trivial to compute
  http://lgg.epfl.ch/publications/2010/discrete_scale_axis_26_04_2010.pdf

Medial surface

Medial axis/surface

- Advantages
  - Useful for shape analysis
- Disadvantages
  - Non-trivial to compute
  - Need to convert to other representation via MAT for rendering

Summary

Representation	Topology/ watertight	Smoothness	Rendering	Main applications
Point cloud	No	No	Rasterization, ray casting	3D scanning, rendering, shape analysis
Polygon soup	No	No	Rasterization, ray casting	Rendering
Polygon mesh	Yes	Via subdivision	Rasterization, ray casting	Interactive modeling, rendering, simulation, manufacturing
Parametric	Yes	Yes	Ray casting, via conversion to meshes	Interactive modeling, CAD
Implicit	Yes	Yes	Ray casting, via conversion to meshes	Interactive modeling
Medial axis	Yes	Yes	Via conversion to meshes	Analysis

Shape representations and neural networks

- Which representations could benefit from using neural networks to define shapes? How?
- Which shape representations would be suitable to be used as inputs or output of neural networks? Pros/cons?

Summary

- Different representations have different advantages/disadvantages
  - No single, definite answer what is best
- Conversion between representations possible, but not always trivial
- Will see some examples later in course