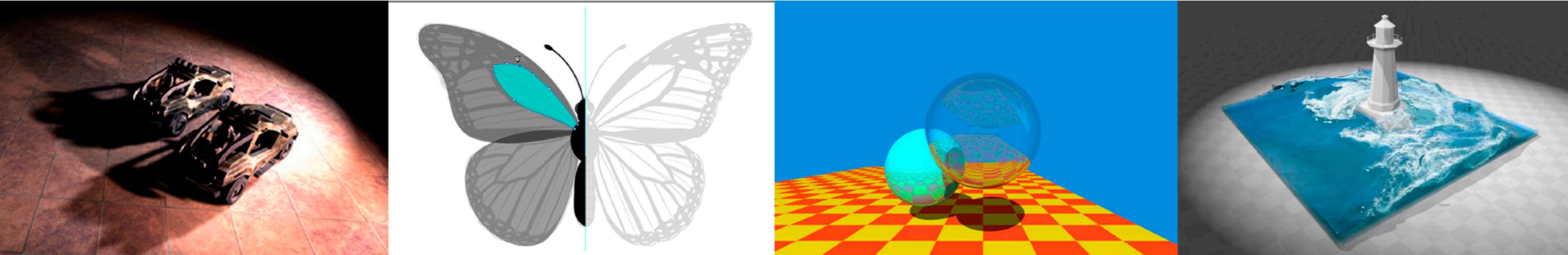


# Introduction to Computer Graphics

GAMES101, Lingqi Yan, UC Santa Barbara

## Lecture 10: Geometry 1 (Introduction)



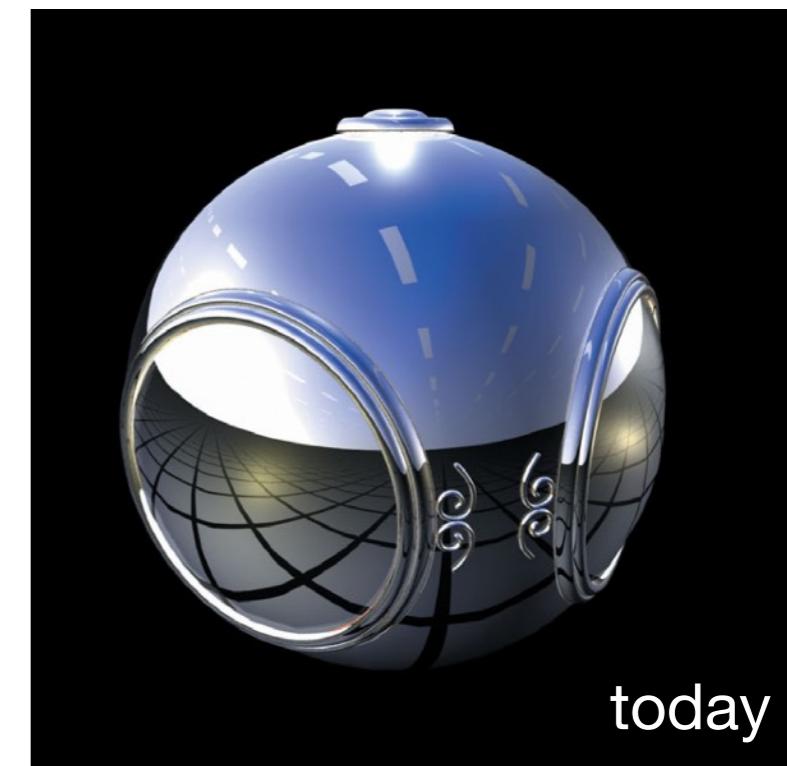
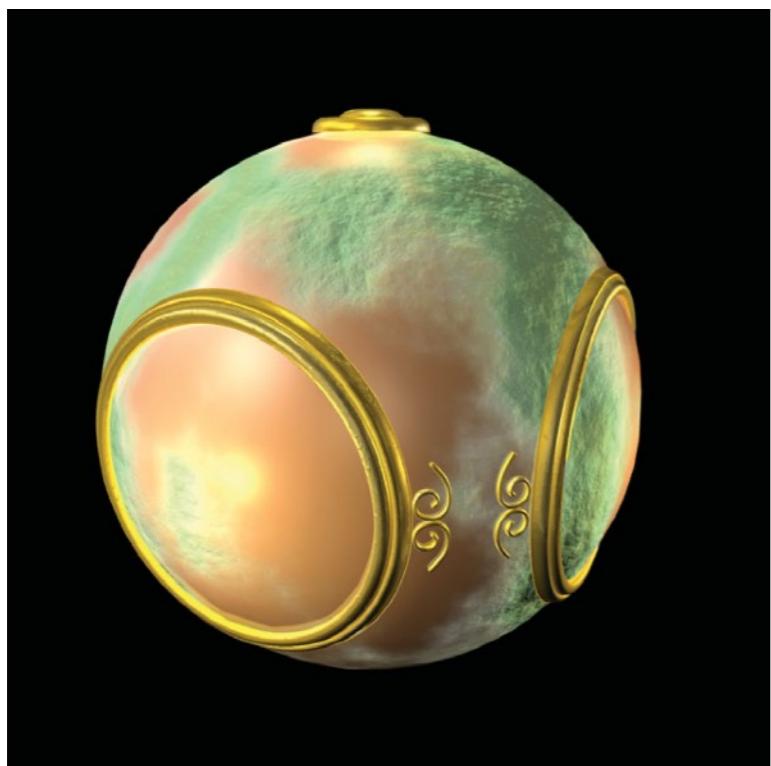
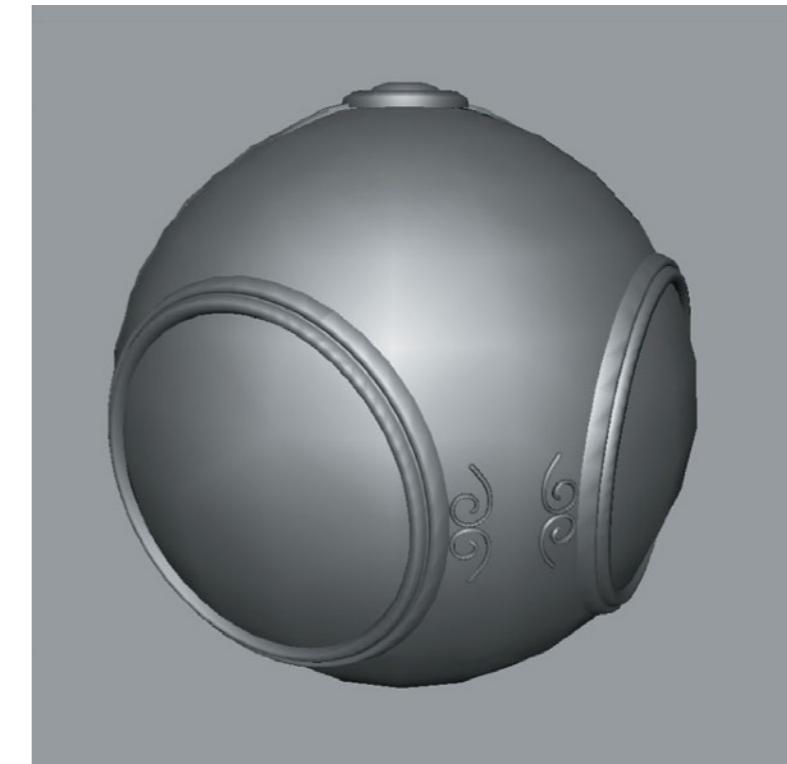
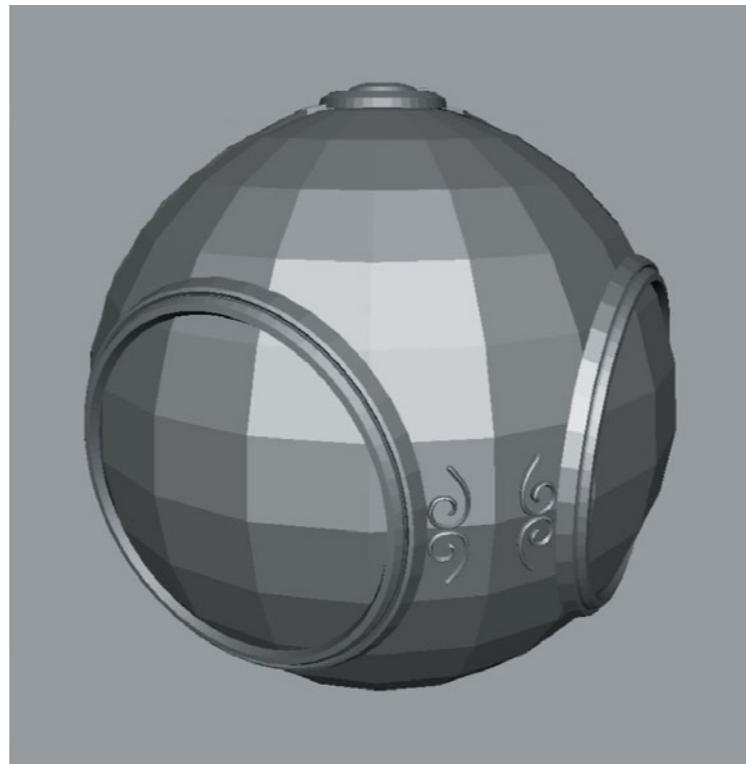
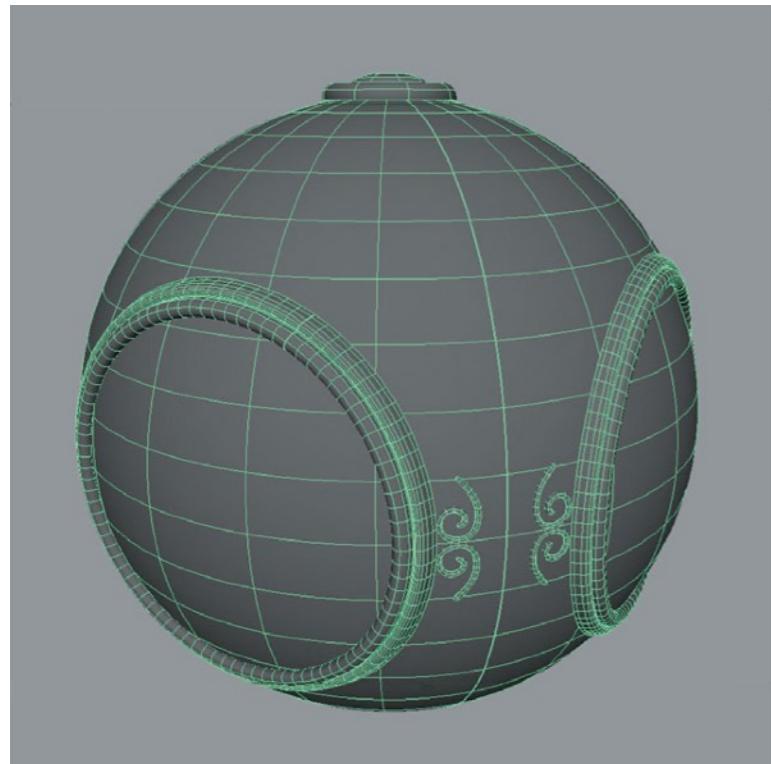
# Announcements

- Homework 3
  - The framework has been updated
  - Together with an “FAQ” section in the BBS
- New TAs
  - Peng Yu (禹鹏), BUAA, [y2505418927@gmail.com](mailto:y2505418927@gmail.com)
  - Wenxian Guo (郭文鲜), ZJU, [wxguojlu@hotmail.com](mailto:wxguojlu@hotmail.com)

# Last Lectures

- Shading 1 & 2
  - Blinn-Phong reflectance model
  - Shading models / frequencies
  - Graphics Pipeline
  - Texture mapping
- Shading 3
  - Barycentric coordinates
  - Texture antialiasing (MIPMAP)
  - Applications of textures

# Last Lectures



# Today

- Applications of textures
- Introduction to geometry (2nd part of this course!)
  - Examples of geometry
  - Various representations of geometry

# Applications of Textures

# Many, Many Uses for Texturing

In modern GPUs, texture = memory + range query (filtering)

- General method to bring data to fragment calculations

Many applications

- Environment lighting
- Store microgeometry
- Procedural textures
- Solid modeling
- Volume rendering
- ...

# Environment Map



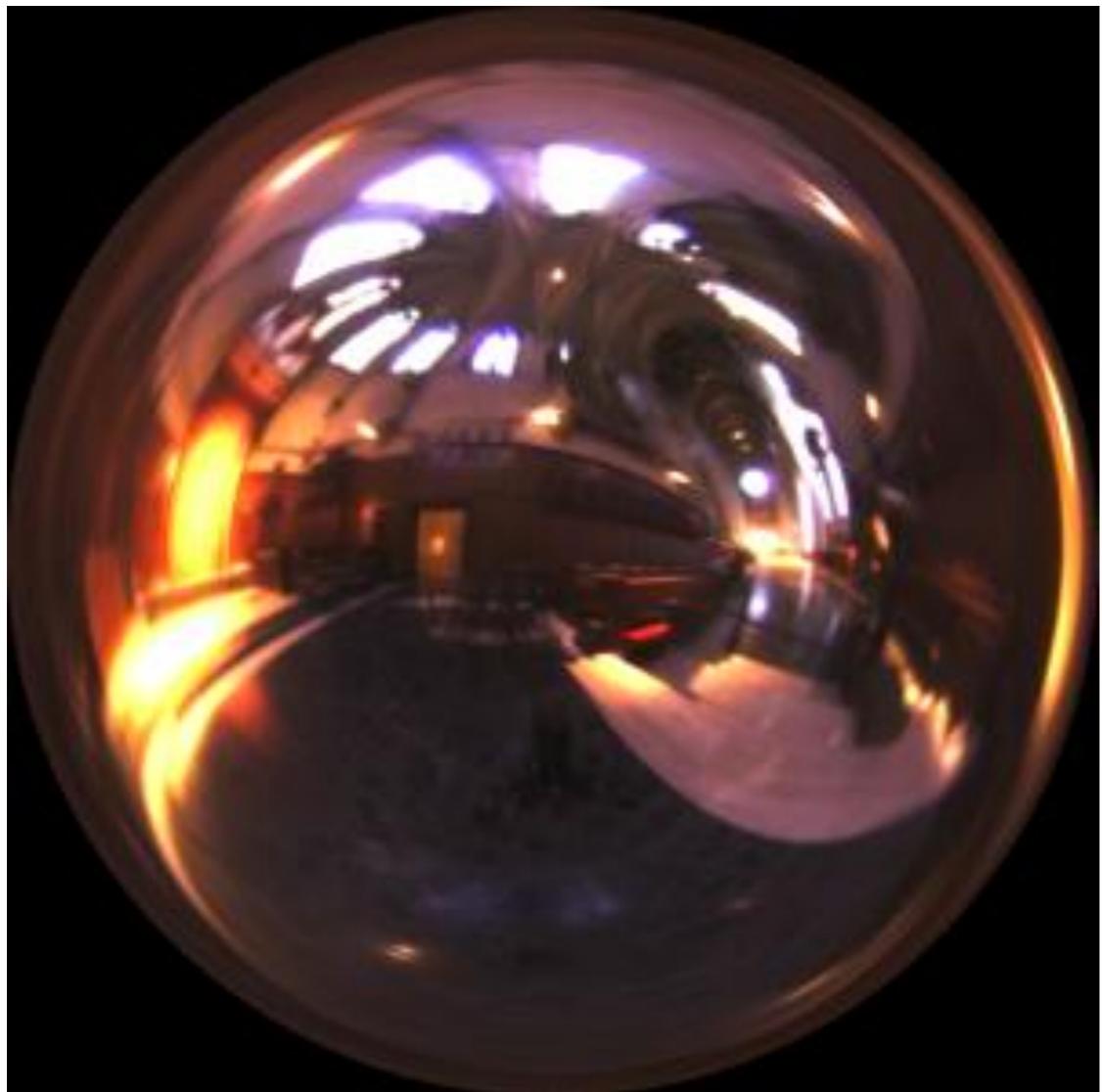
Light from the environment



Rendering with the environment

[Blinn & Newell 1976]

# Environmental Lighting

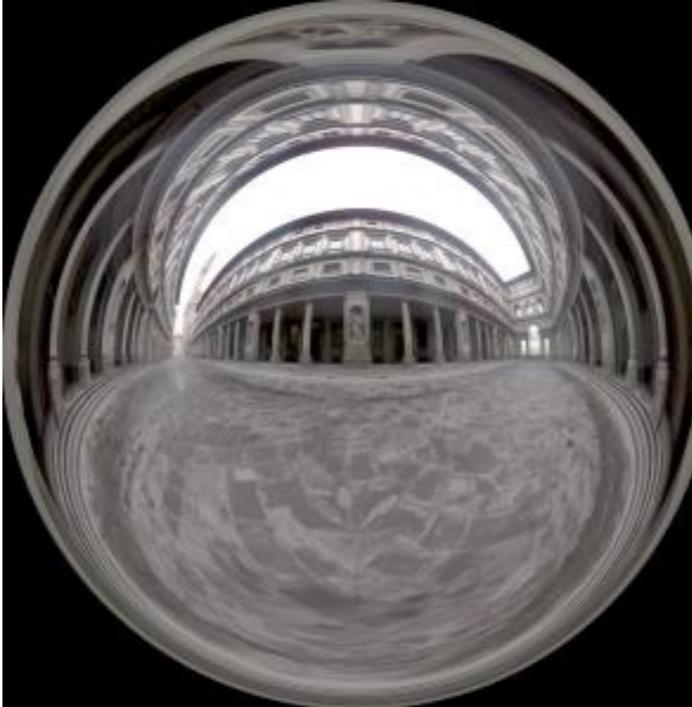


Environment map (left) used to render realistic lighting

# Spherical Environment Map



*Hand with Reflecting Sphere.* M. C. Escher, 1935. lithograph



Eucalyptus Grove Light Probe  
©1999 Paul Debevec  
<http://www.debevec.org/Probes>

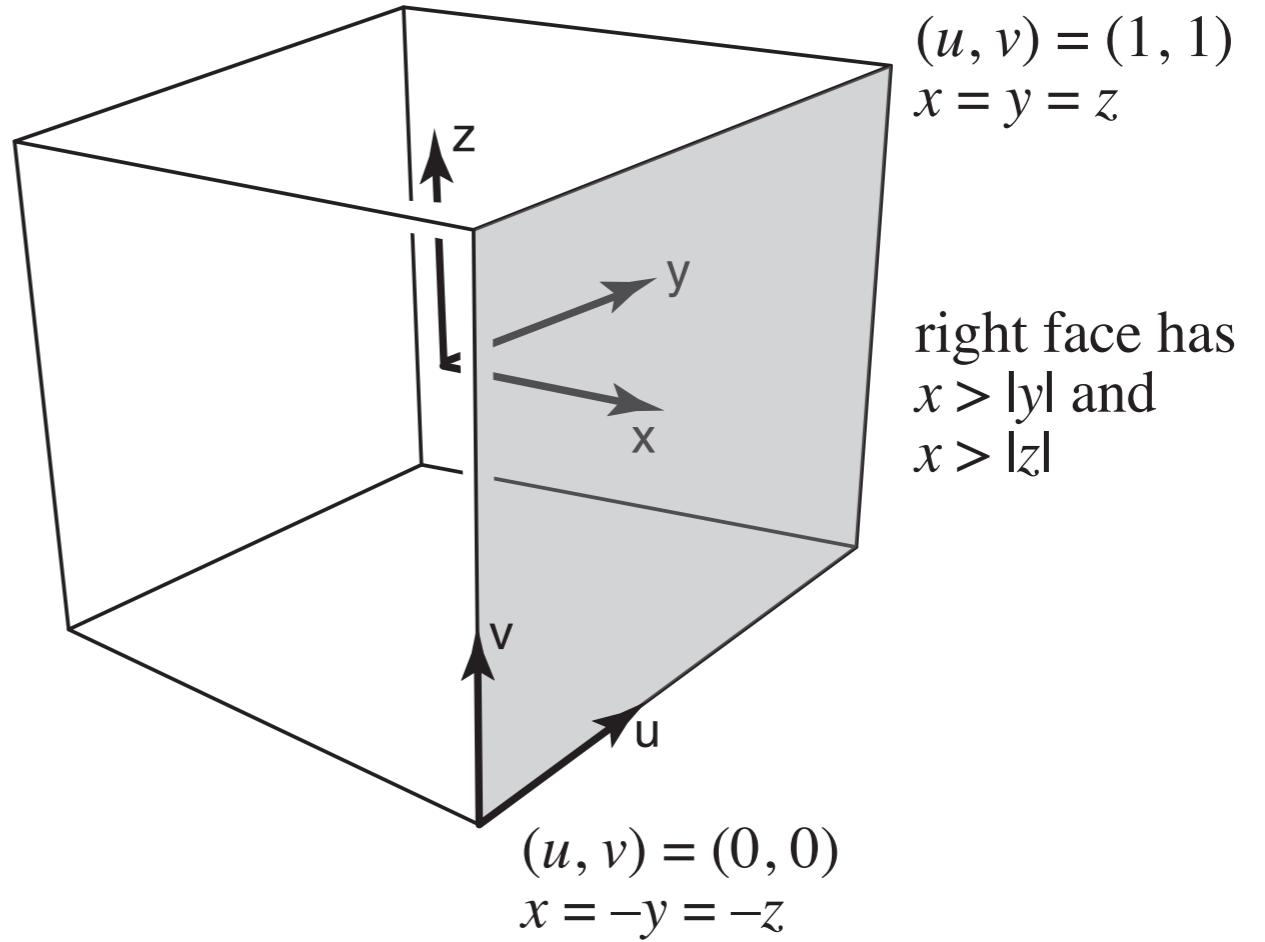
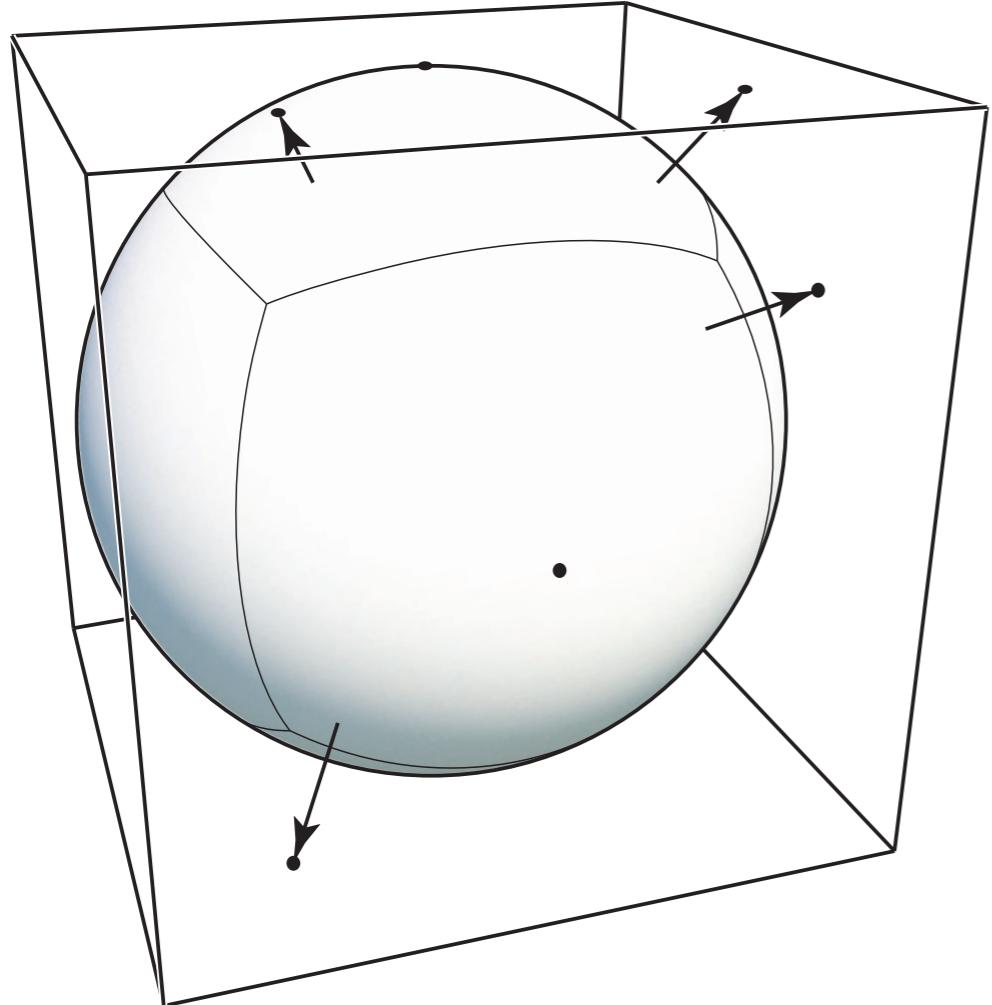
*Light Probes,* Paul Debevec

# Spherical Map — Problem



Prone to distortion (top and bottom parts)!

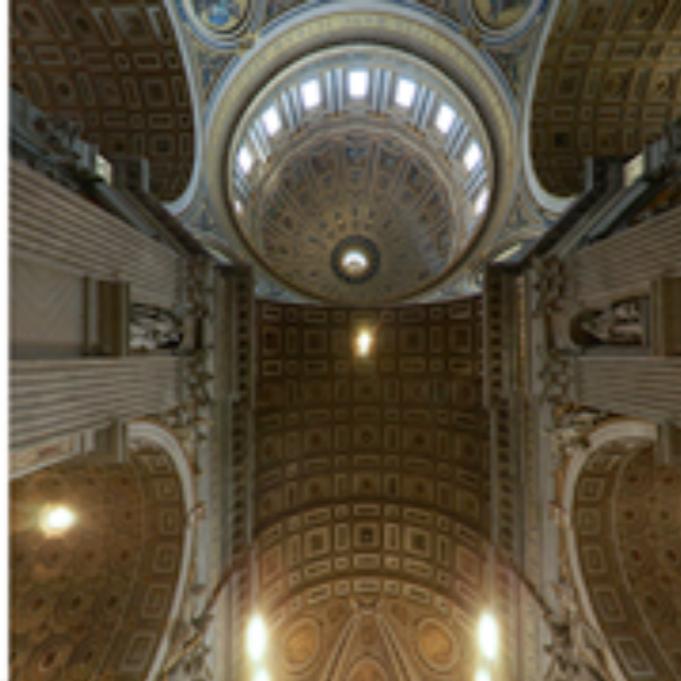
# Cube Map



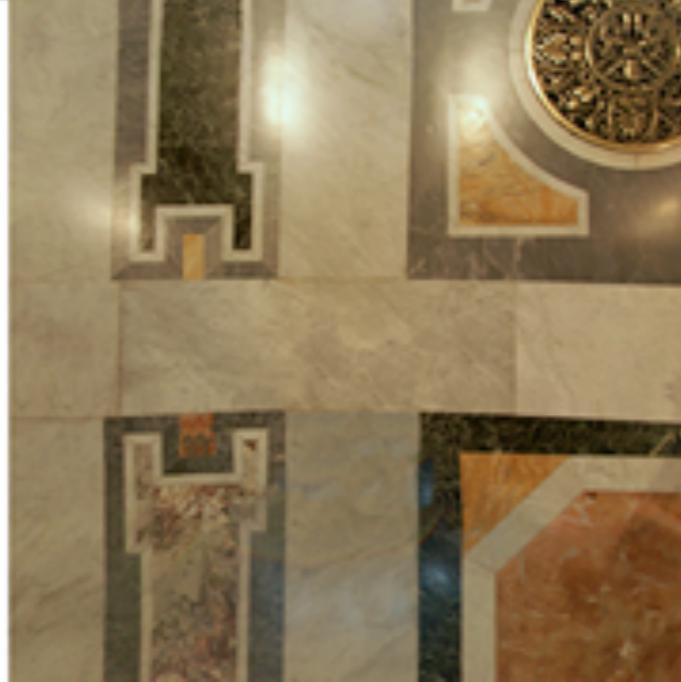
A vector maps to cube point along that direction.  
The cube is textured with 6 square texture maps.



[Emil Persson]



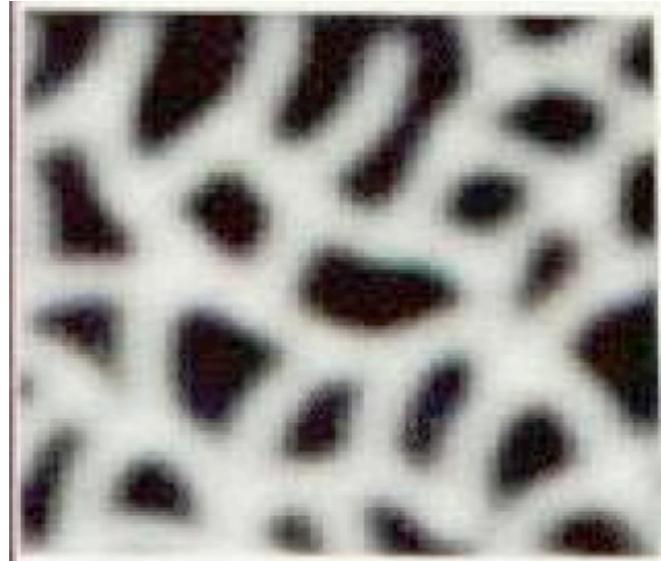
Much less distortion!



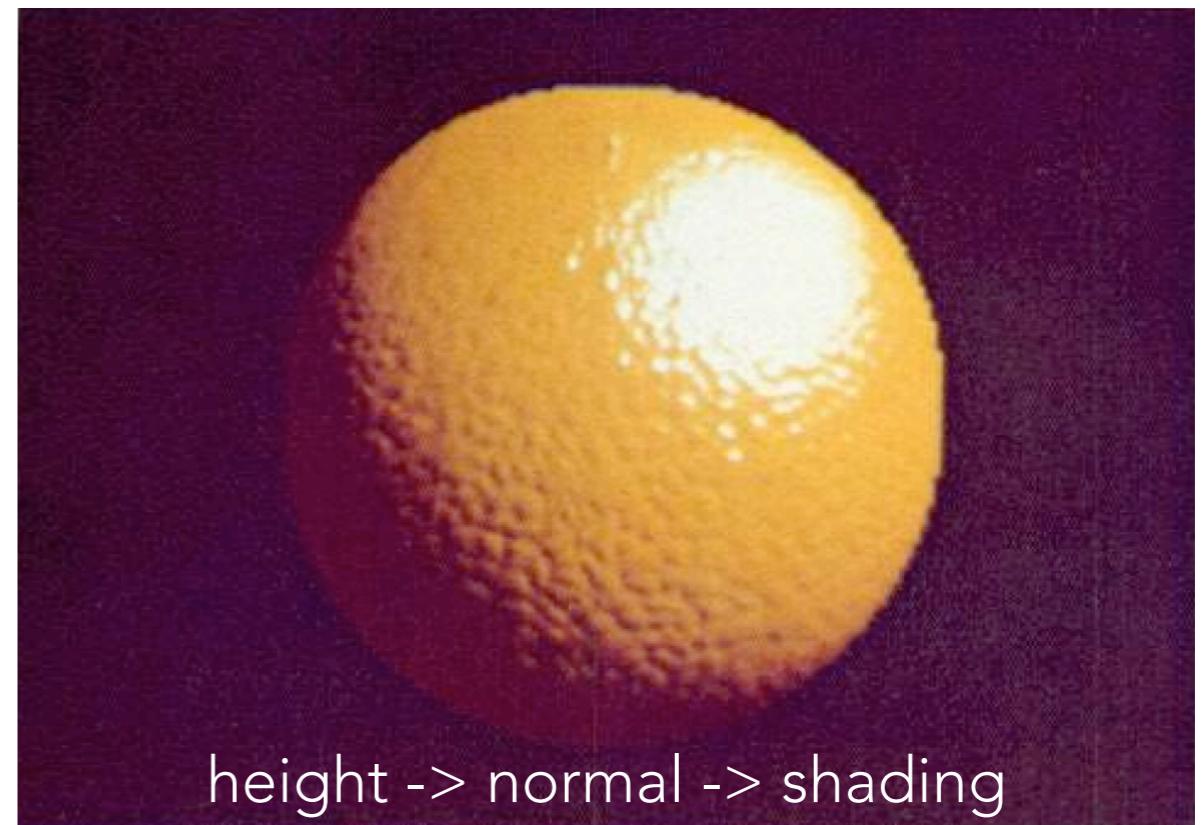
Need dir->face computation

# Textures can affect shading!

- Textures doesn't have to only represent colors
  - What if it stores the height / normal?
  - Bump / normal mapping
  - **Fake** the detailed geometry



Relative height to the underlying surface

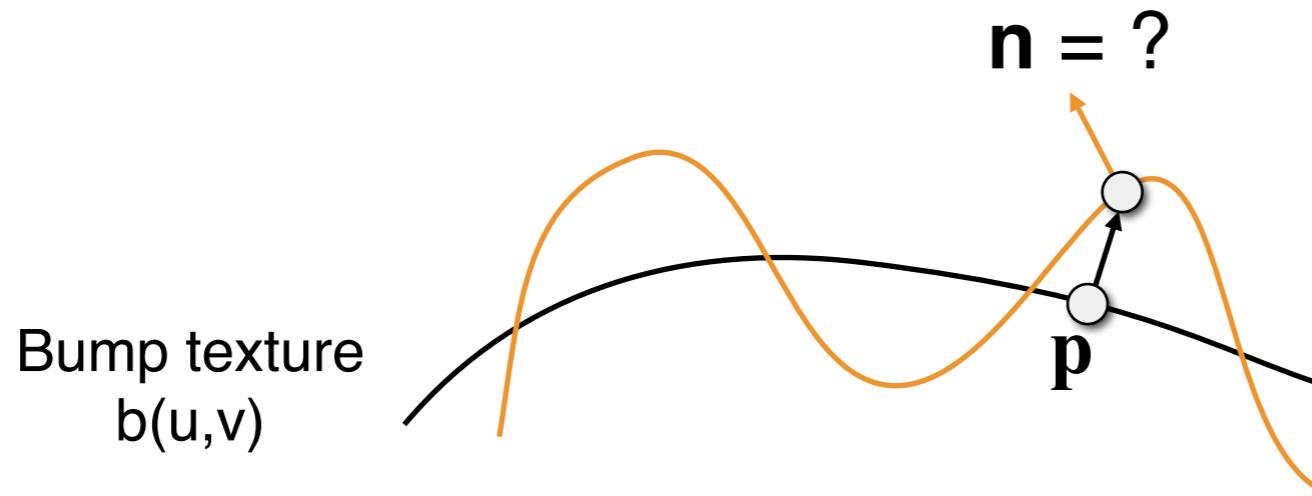


height -> normal -> shading

# Bump Mapping

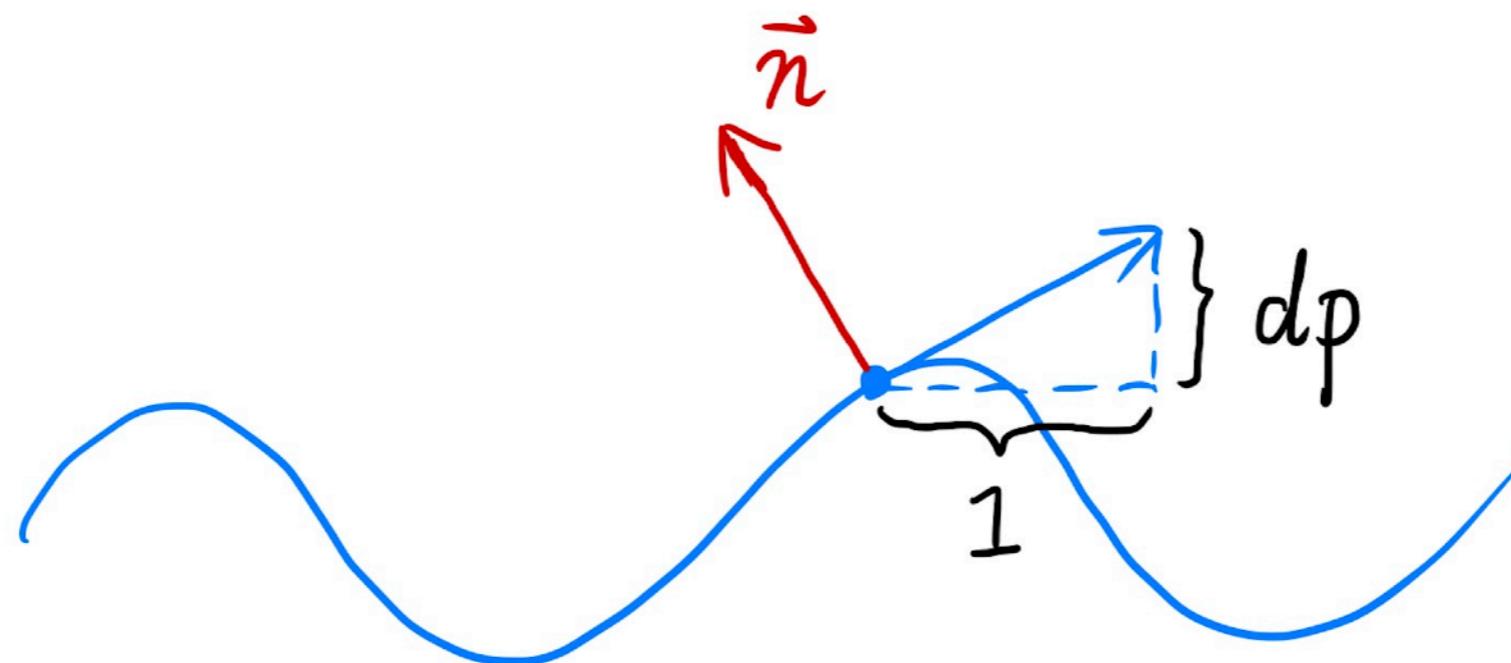
Adding surface detail without adding more triangles

- Perturb surface normal per pixel  
(for shading computations only)
- “Height shift” per texel defined by a texture
- How to modify normal vector?



# How to perturb the normal (in flatland)

- Original surface normal  $n(p) = (0, 1)$
- Derivative at  $p$  is  $dp = c * [h(p+1) - h(p)]$
- Perturbed normal is then  $n(p) = (-dp, 1).normalized()$

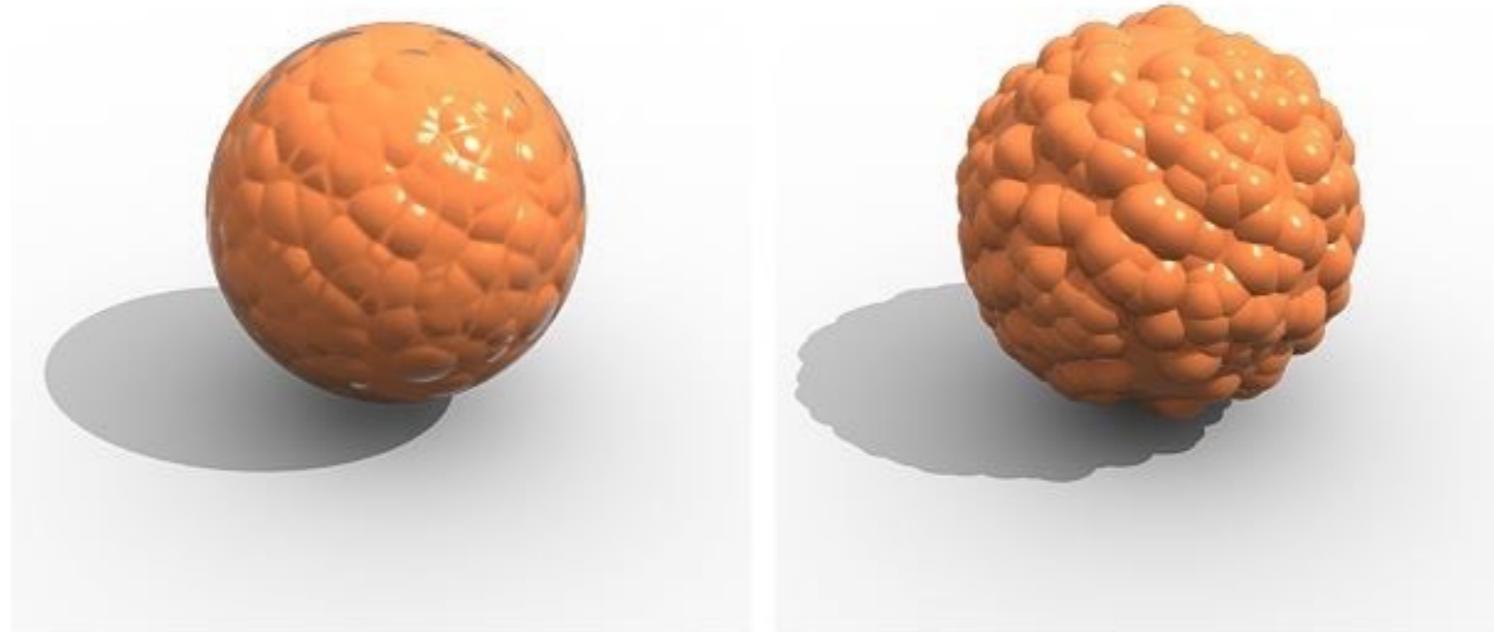


# How to perturb the normal (in 3D)

- Original surface normal  $n(p) = (0, 0, 1)$
- Derivatives at p are
  - $dp/du = c1 * [h(u+1) - h(u)]$
  - $dp/dv = c2 * [h(v+1) - h(v)]$
- Perturbed normal is  $n = (-dp/du, -dp/dv, 1).normalized()$
- Note that this is in **local coordinate!**  
More will be elaborated in FAQ of HW3

# Textures can affect shading!

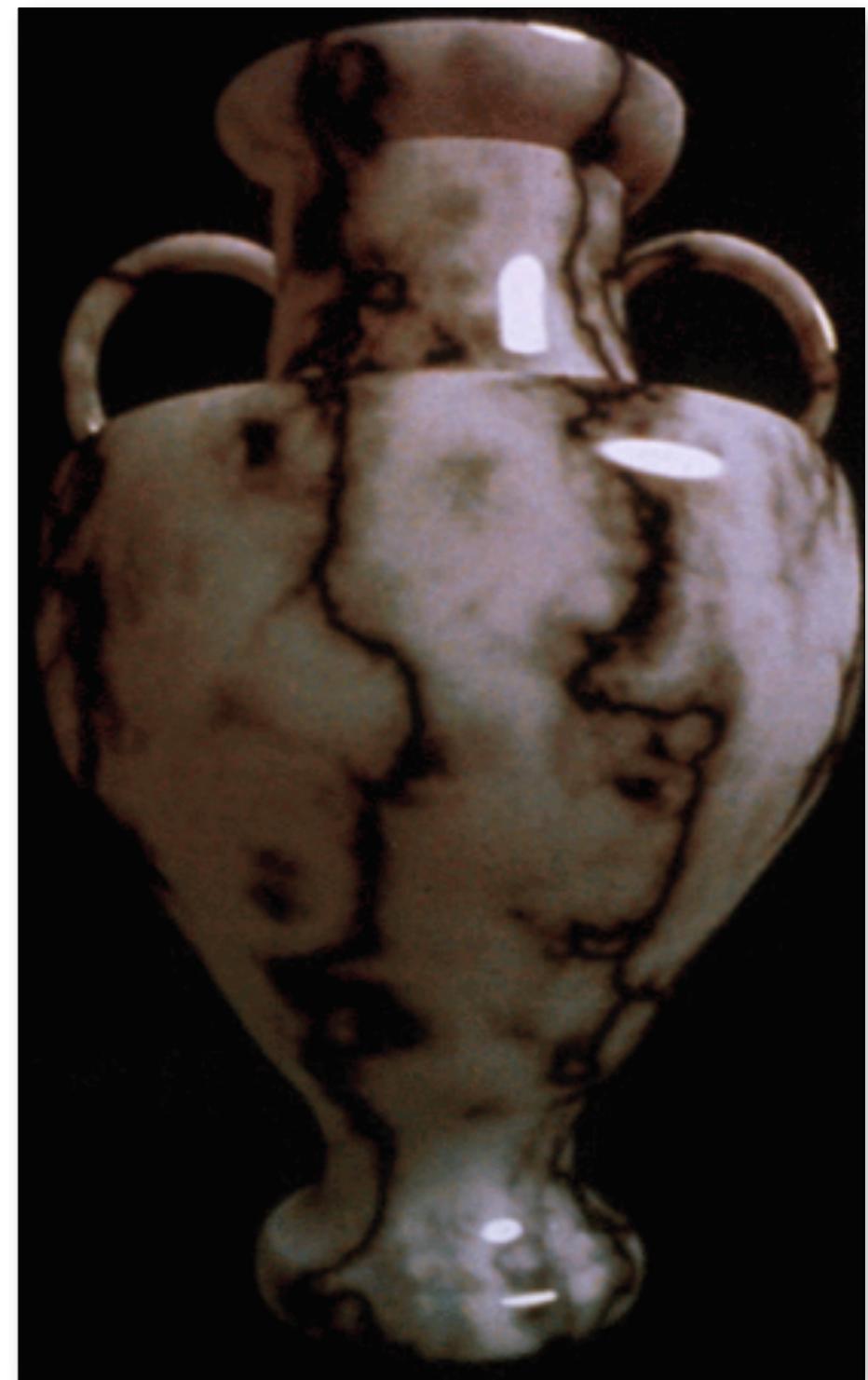
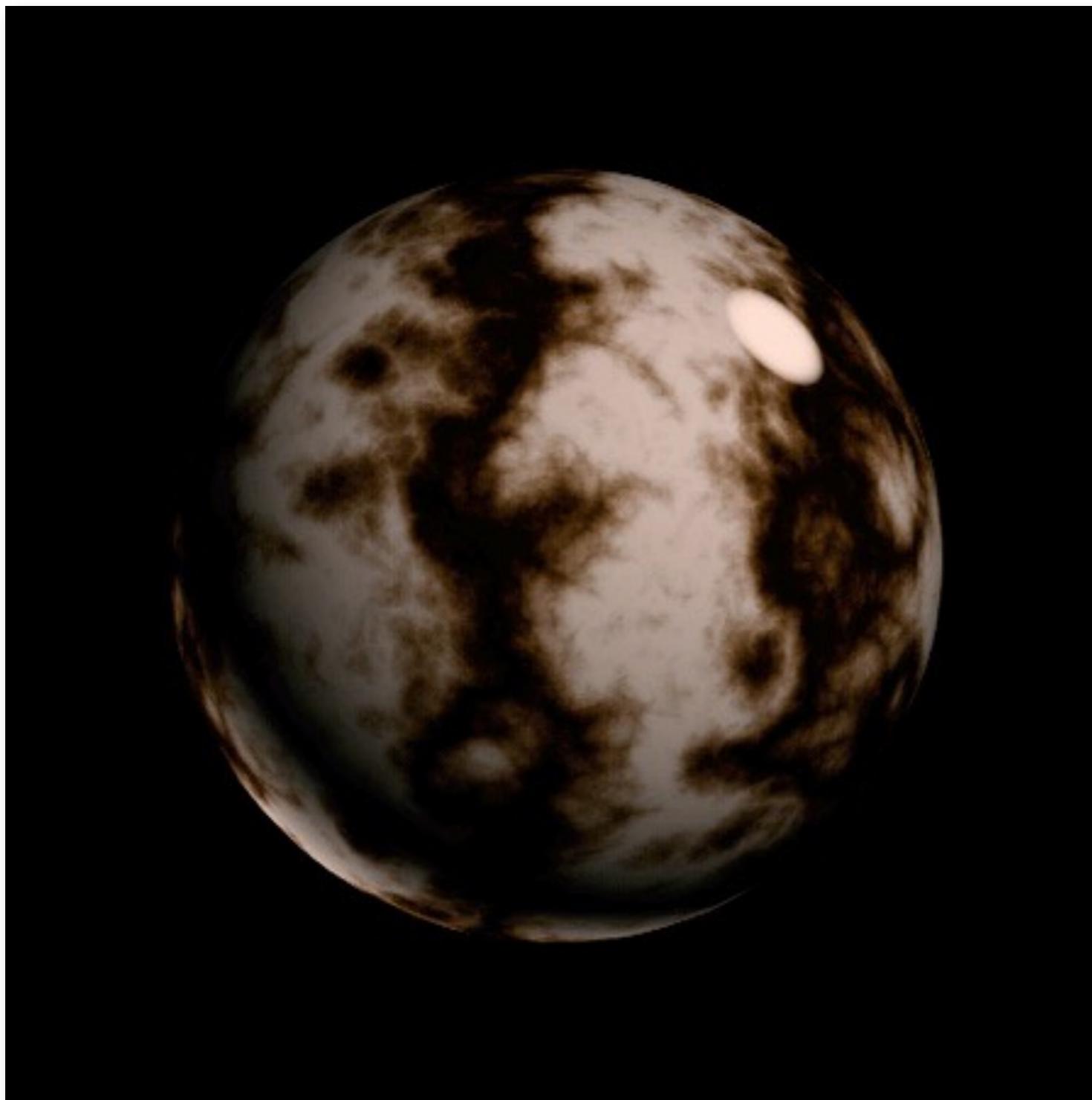
- **Displacement mapping** — a more advanced approach
  - Uses the same texture as in bumping mapping
  - Actually **moves the vertices**



Bump / **Normal** mapping

Displacement mapping

# 3D Procedural Noise + Solid Modeling



Perlin noise, Ken Perlin

# Provide Precomputed Shading



**Simple  
shading**



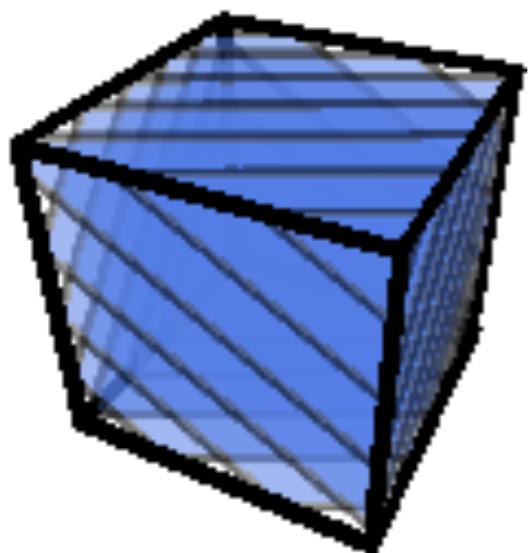
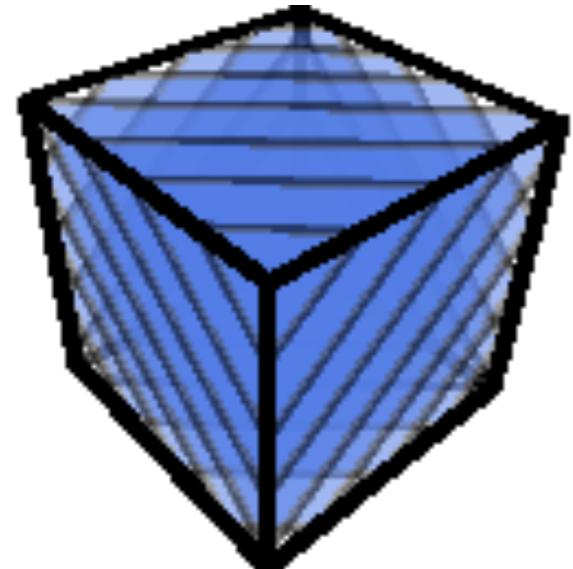
**Ambient occlusion  
texture map**



**With ambient  
occlusion**

Autodesk

# 3D Textures and Volume Rendering



Marc Levoy

# Today

- Shading 3
  - Applications of textures
- **Introduction to geometry**
  - Examples of geometry
  - Various representations of geometry

# Examples of Geometry

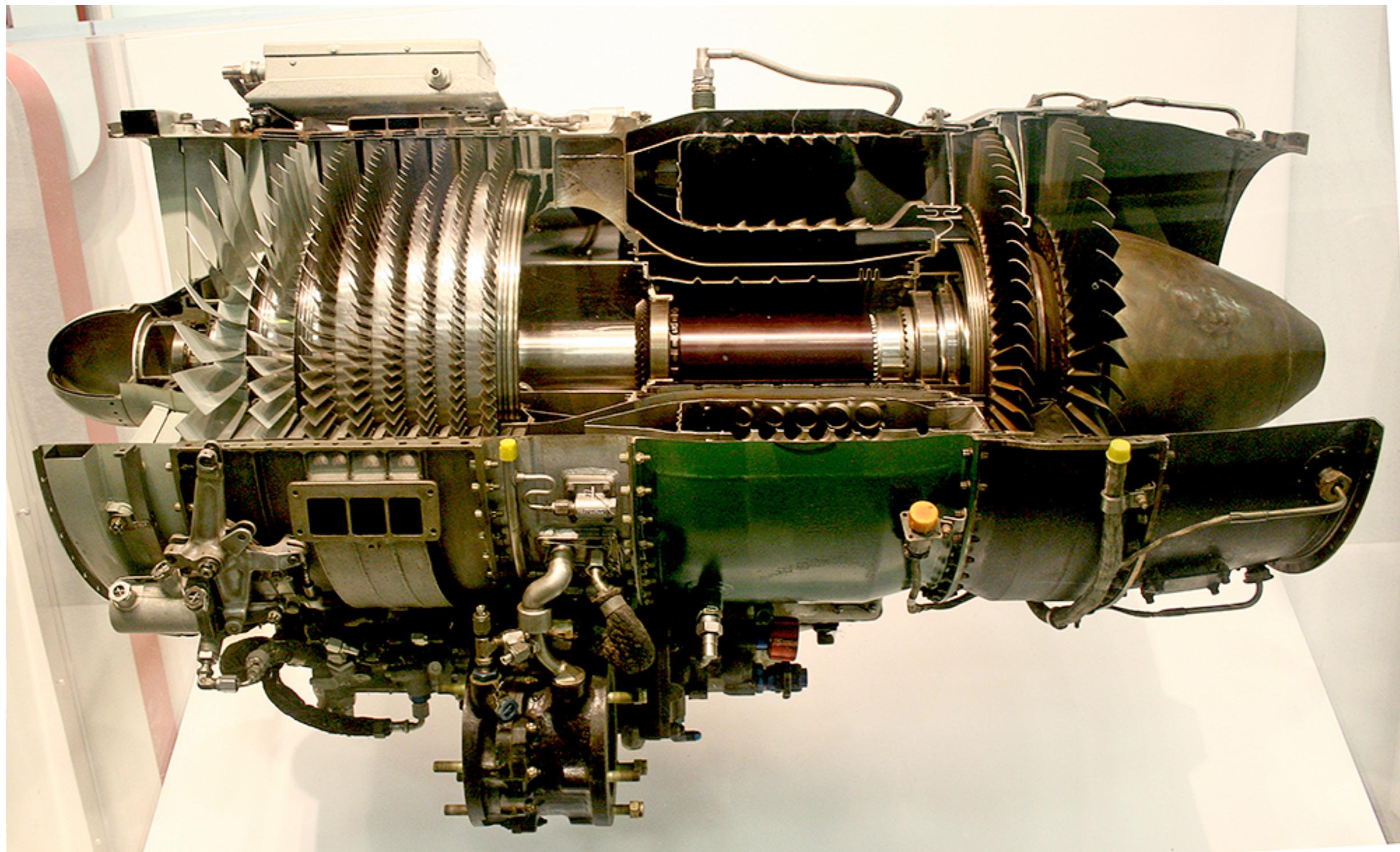


# Examples of Geometry



[NetCarShow.com](http://NetCarShow.com)

# Examples of Geometry



# Examples of Geometry



# Examples of Geometry



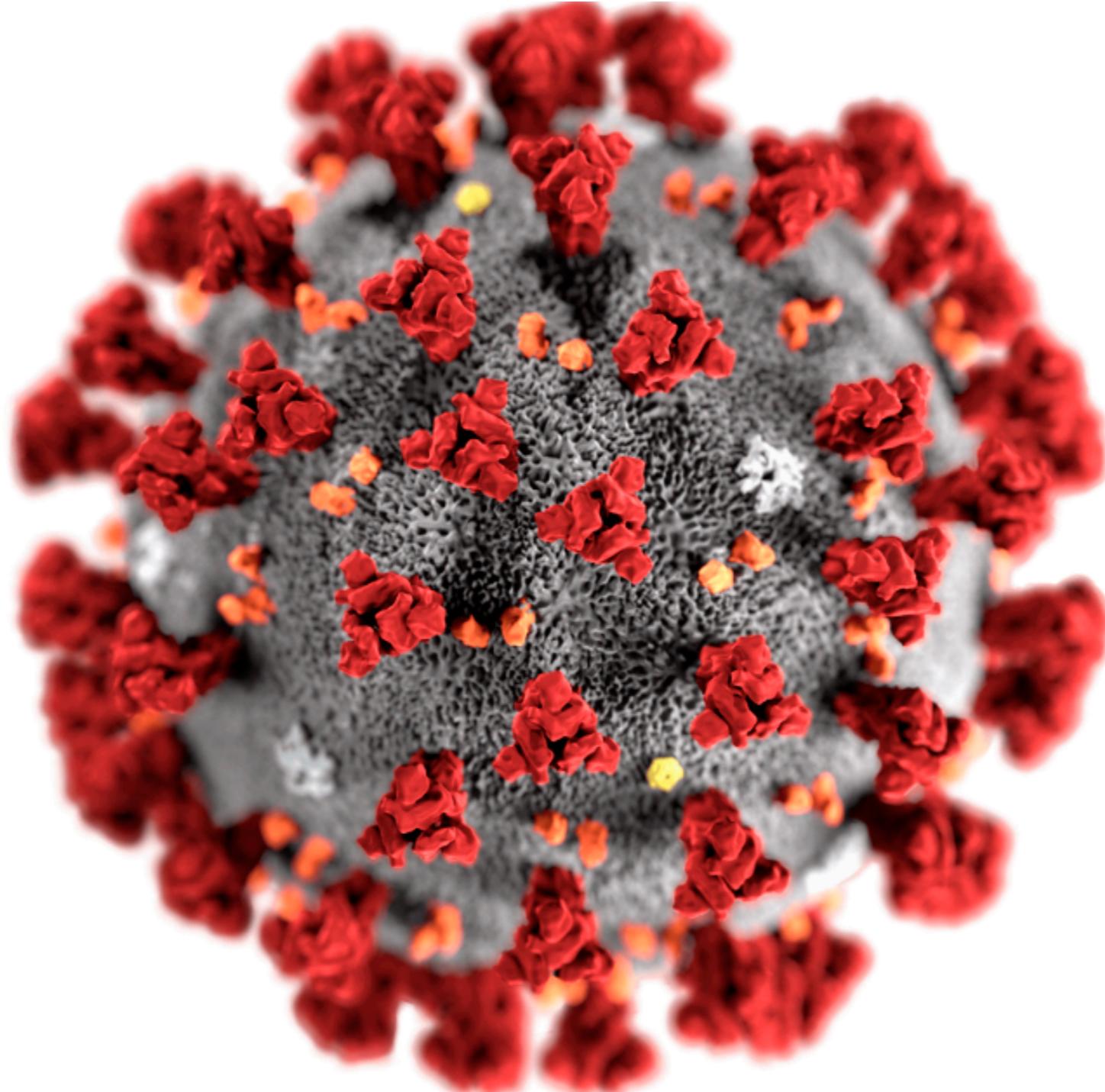
# Examples of Geometry



# Examples of Geometry



# Examples of Geometry



COVID-19

I will call it  
“Trump Virus”  
forever.

# Examples of Geometry



Adriana Franco, National Geographic

NATIONAL  
GEOGRAPHIC

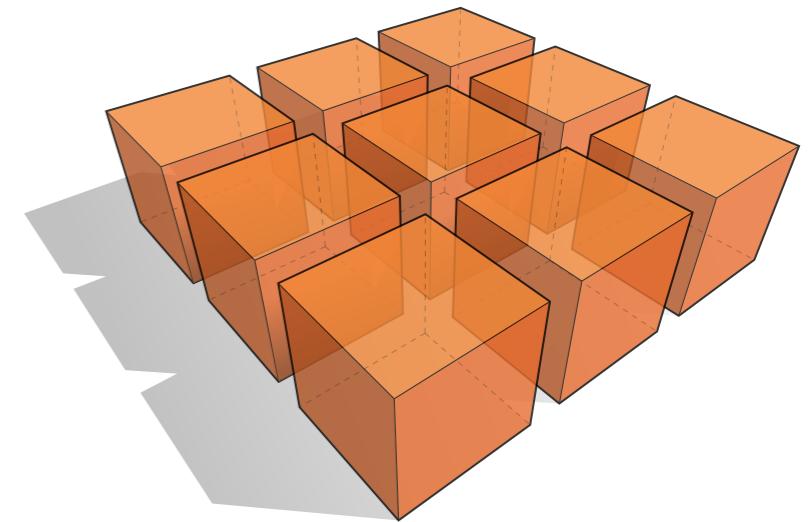
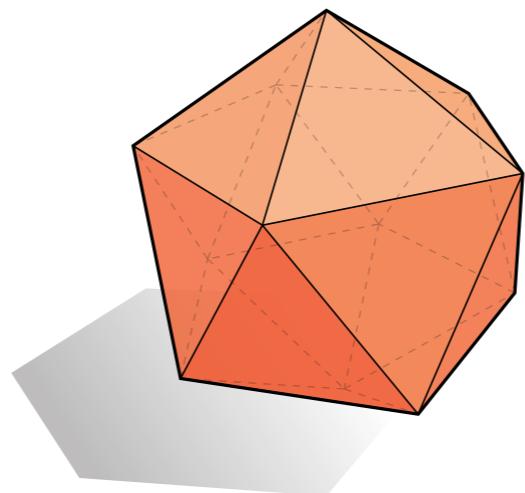
Photograph by Adriana Franco, Your Shot

© COPYRIGHT ADRIANA FRANCO. ALL RIGHTS RESERVED.

# Many Ways to Represent Geometry

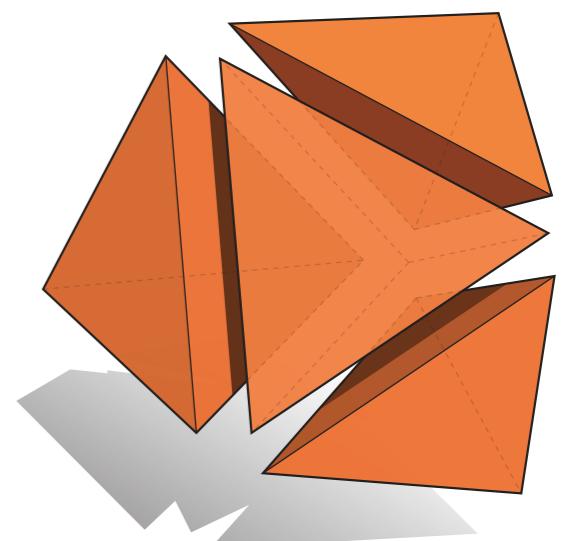
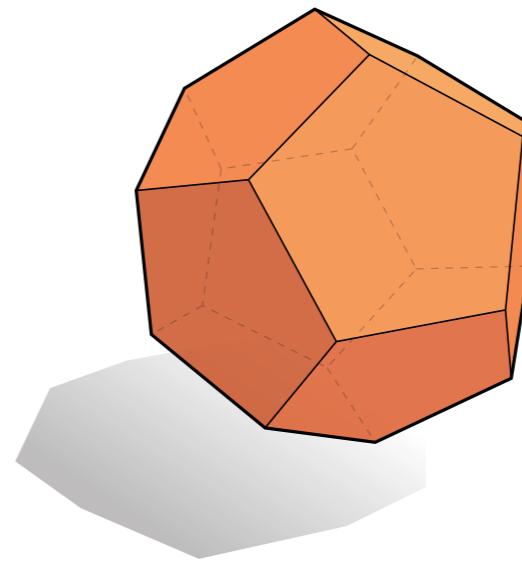
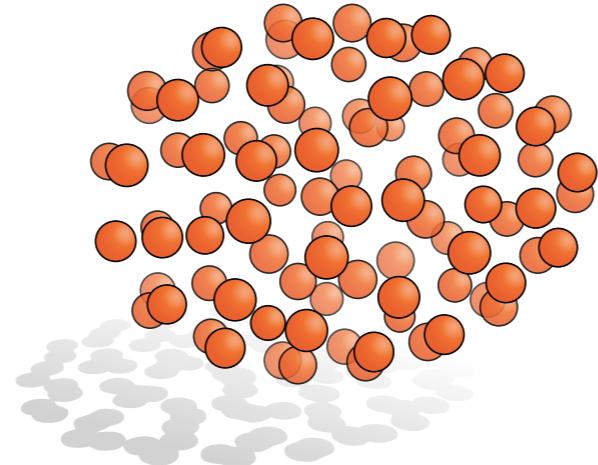
## Implicit

- algebraic surface
- level sets
- distance functions
- ...



## Explicit

- point cloud
- polygon mesh
- subdivision, NURBS
- ...



Each choice best suited to a different task/type of geometry

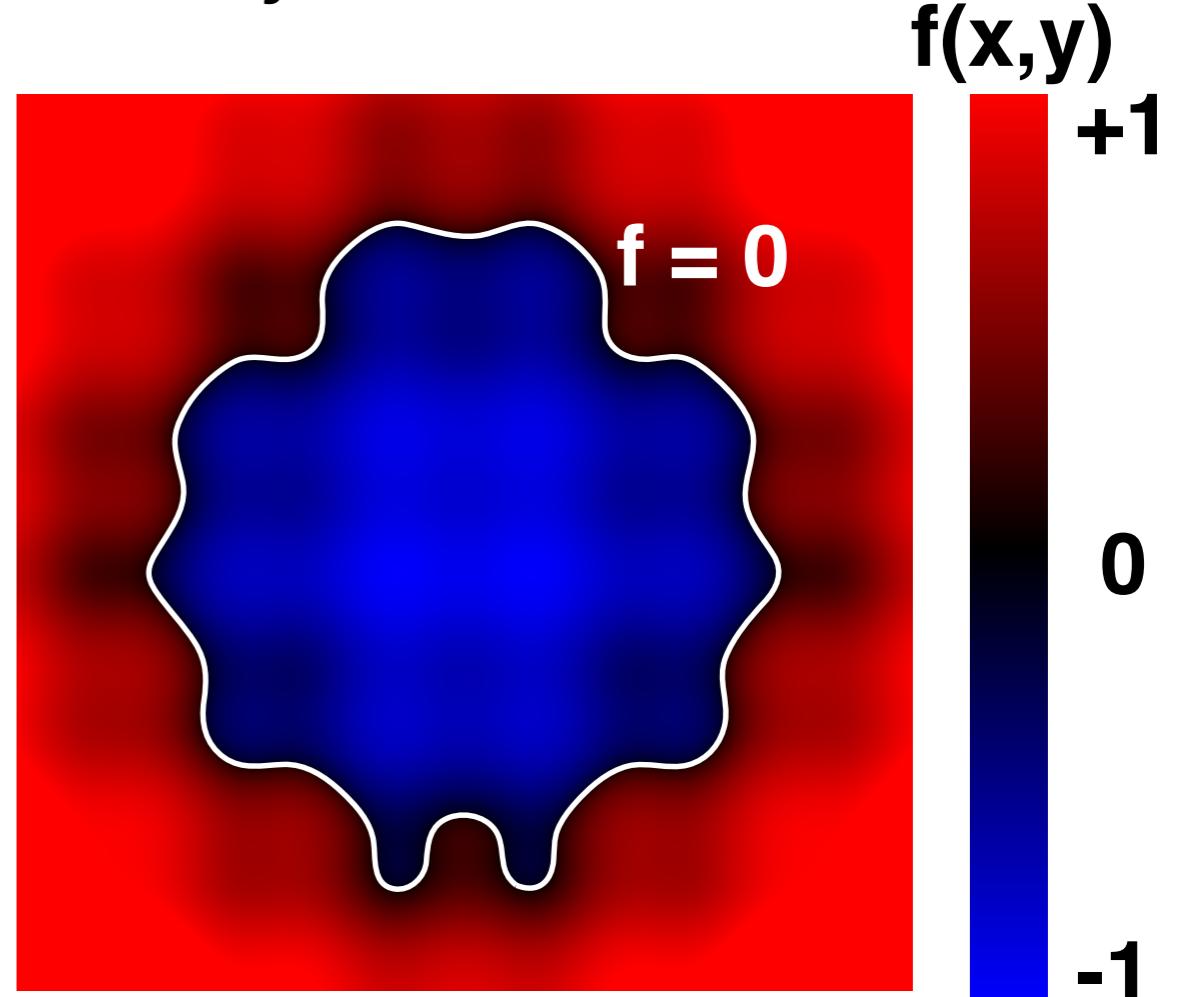
# “Implicit” Representations of Geometry

Based on classifying points

- Points satisfy some specified relationship

E.g. sphere: all points in 3D, where  $x^2+y^2+z^2 = 1$

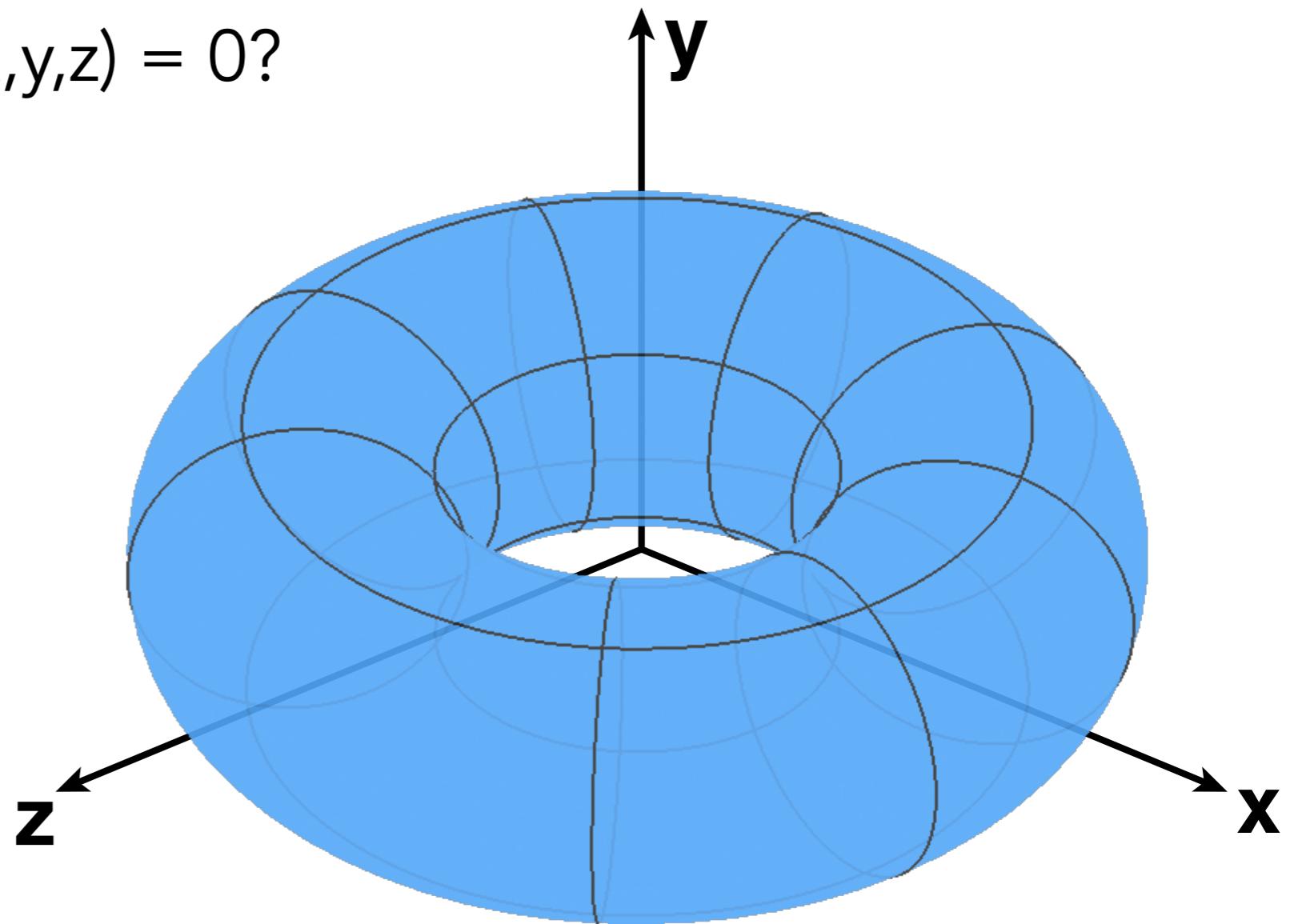
More generally,  $f(x,y,z) = 0$



# Implicit Surface – Sampling Can Be Hard

$$f(x, y, z) = (2 - \sqrt{x^2 + y^2})^2 + z^2 - 1$$

What points lie on  $f(x, y, z) = 0$ ?



Some tasks are hard with implicit representations

# Implicit Surface – Inside/Outside Tests Easy

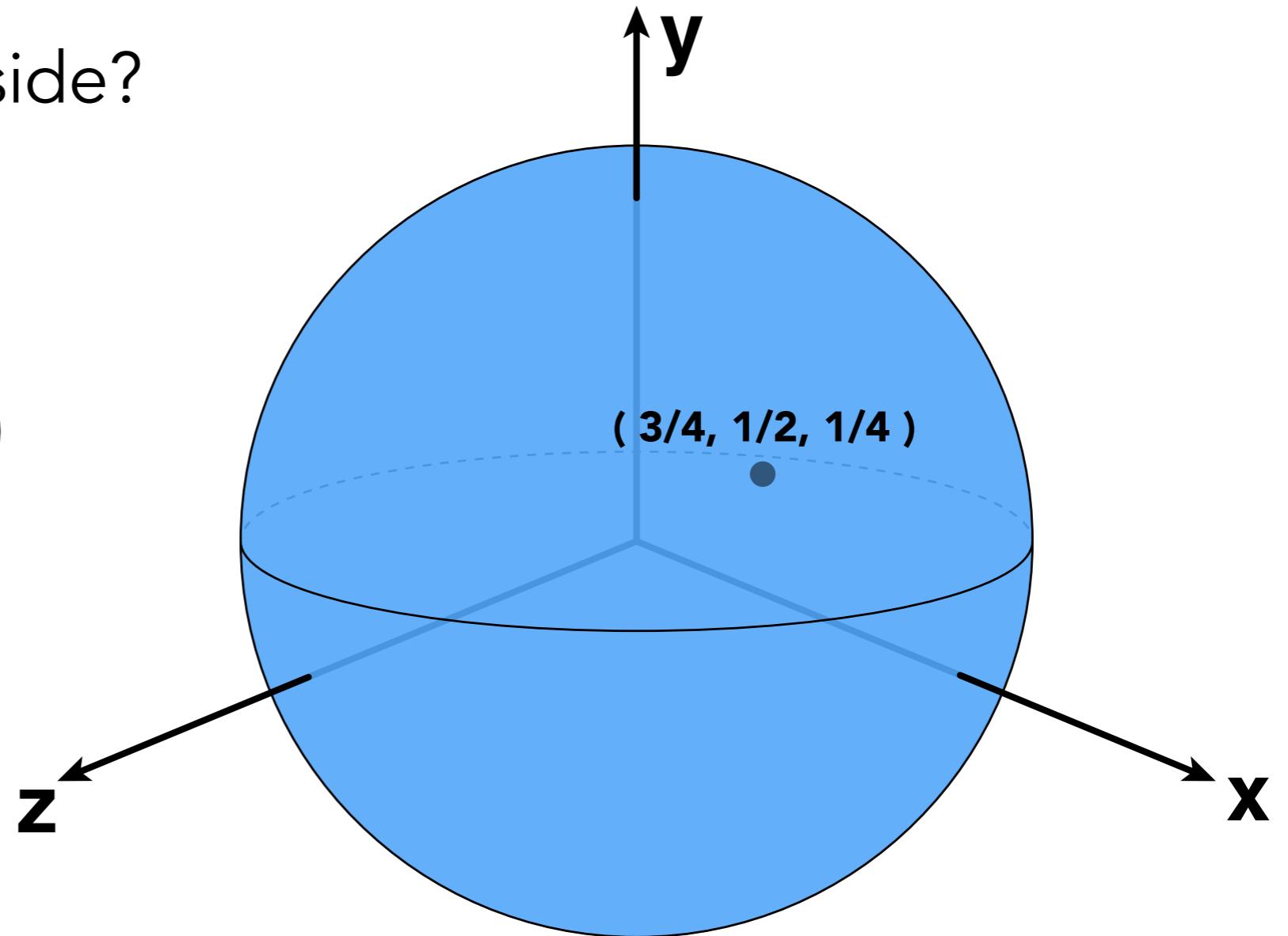
$$f(x, y, z) = x^2 + y^2 + z^2 - 1$$

Is  $(3/4, 1/2, 1/4)$  inside?

Just plug it in:

$$f(x, y, z) = -1/8 < 0$$

Yes, inside.



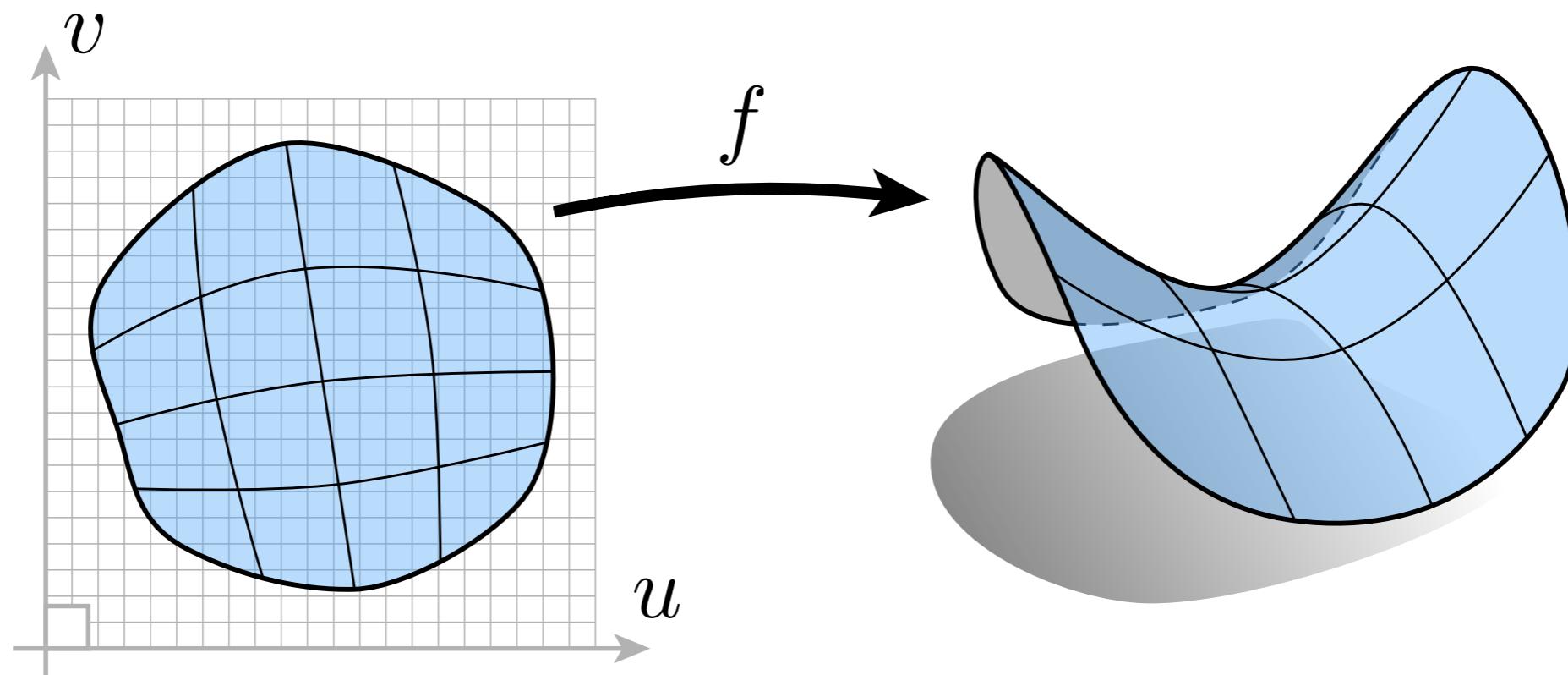
Implicit representations make some tasks easy

# “Explicit” Representations of Geometry

All points are **given directly or via parameter mapping**

Generally:

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^3; (u, v) \mapsto (x, y, z)$$

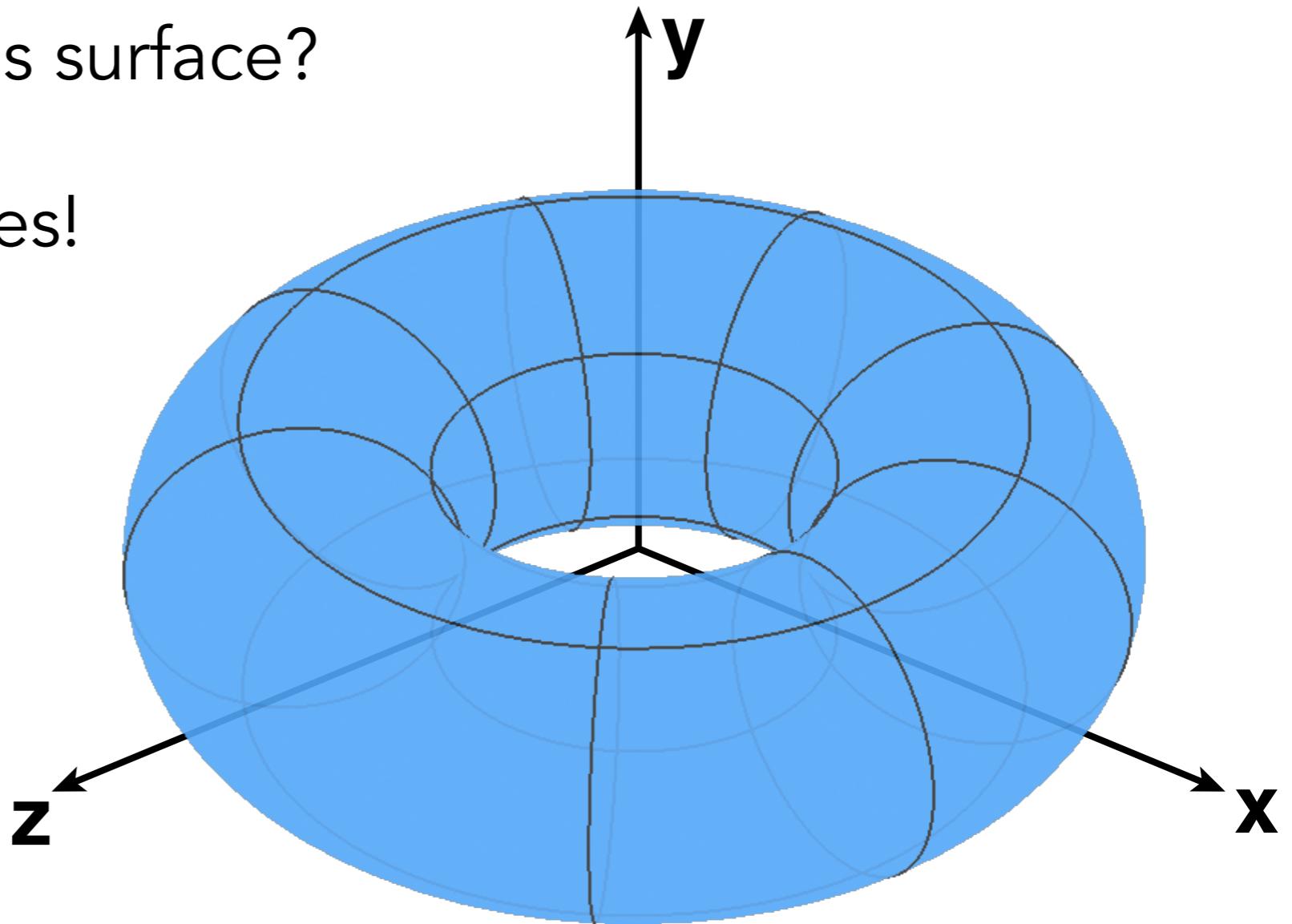


# Explicit Surface – Sampling Is Easy

$$f(u, v) = ((2 + \cos u) \cos v, (2 + \cos u) \sin v, \sin u)$$

What points lie on this surface?

Just plug in  $(u, v)$  values!

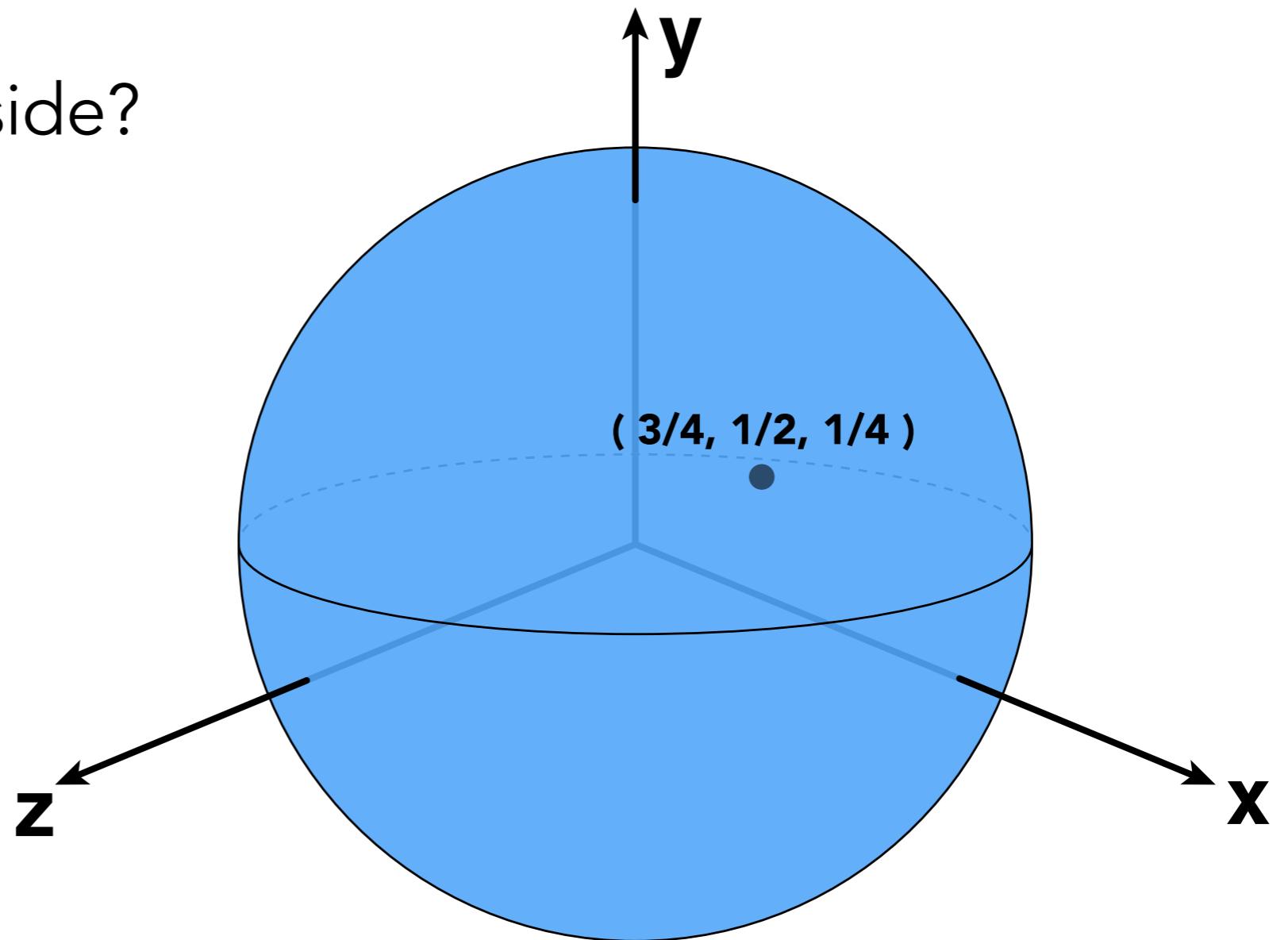


Explicit representations make some tasks easy

# Explicit Surface – Inside/Outside Test Hard

$$f(u, v) = (\cos u \sin v, \sin u \sin v, \cos v)$$

Is  $(3/4, 1/2, 1/4)$  inside?



Some tasks are hard with explicit representations

# No “Best” Representation – Geometry is Hard!

“I hate meshes.  
I cannot believe how hard this is.  
Geometry is hard.”

— David Baraff  
Senior Research Scientist  
Pixar Animation Studios

Best Representation  
Depends on the Task!

# More Implicit Representations in Computer Graphics

# Many Implicit Representations in Graphics

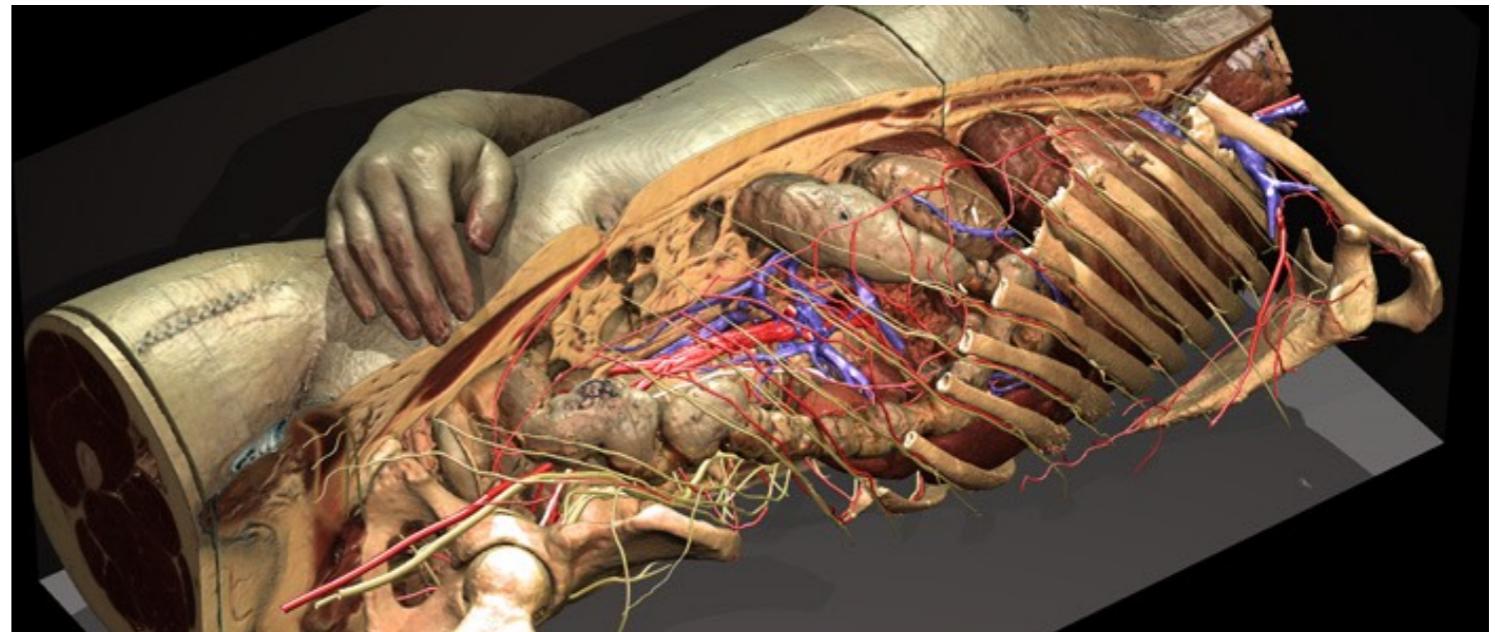
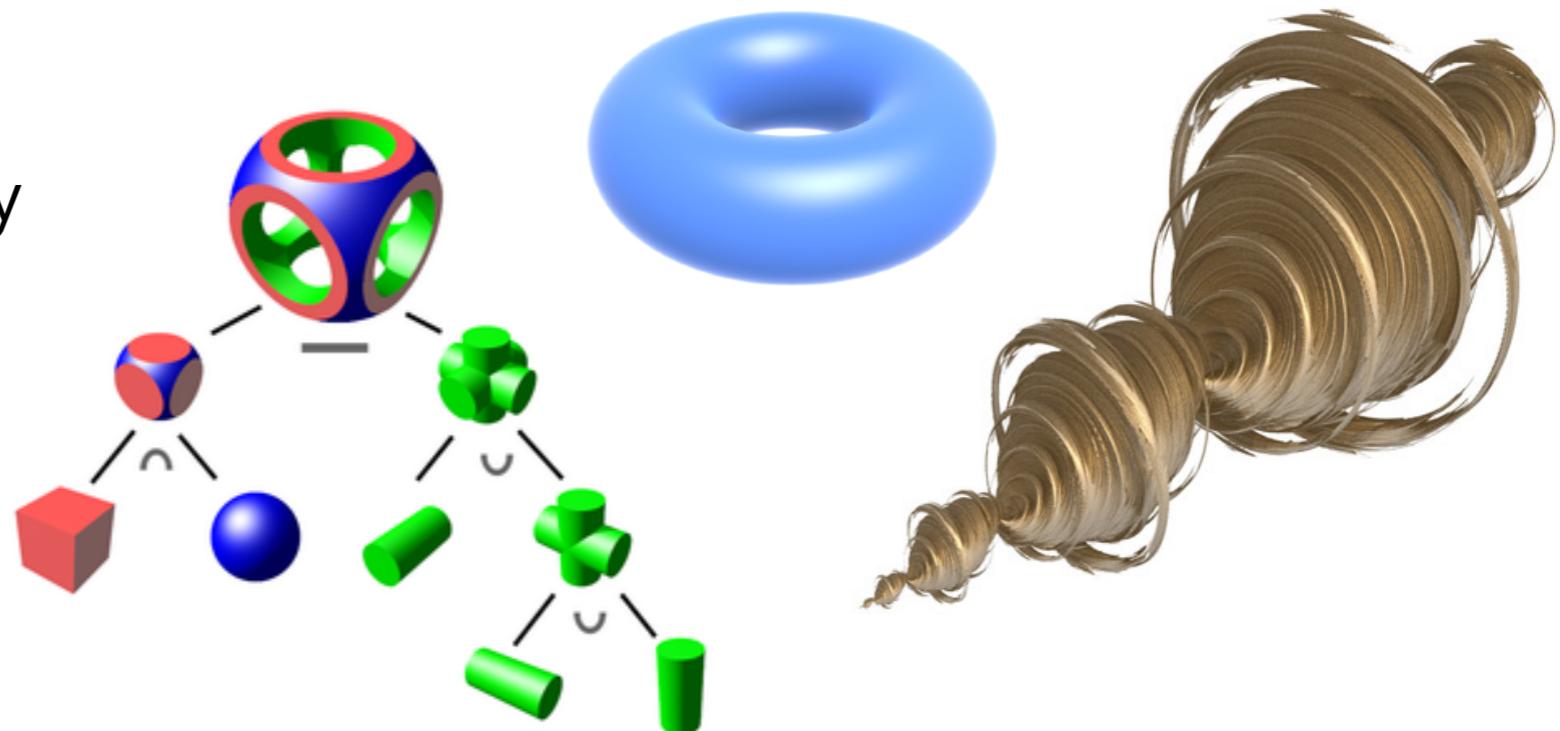
Algebraic surfaces

Constructive solid geometry

Level set methods

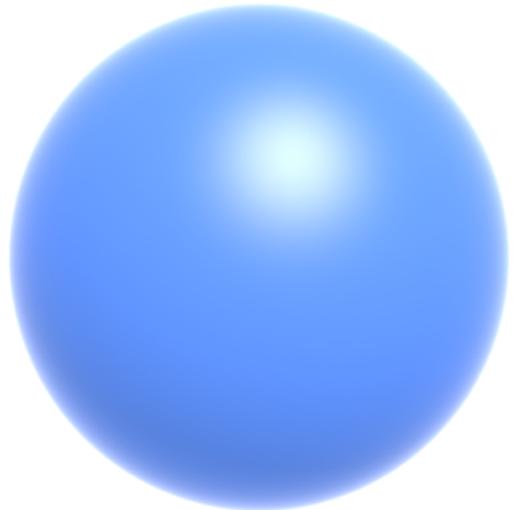
Fractals

...

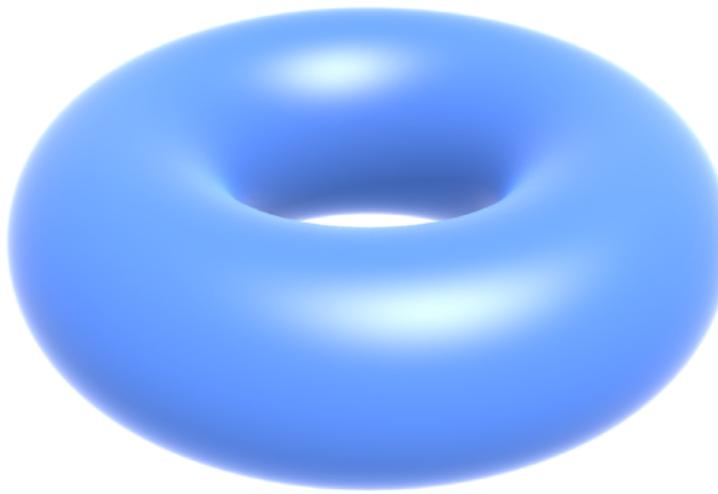


# Algebraic Surfaces (Implicit)

Surface is zero set of a polynomial in  $x, y, z$



$$x^2 + y^2 + z^2 = 1$$



$$(R - \sqrt{x^2 + y^2})^2 + z^2 = r^2$$

$$(x^2 + \frac{9y^2}{4} + z^2 - 1)^3 =$$

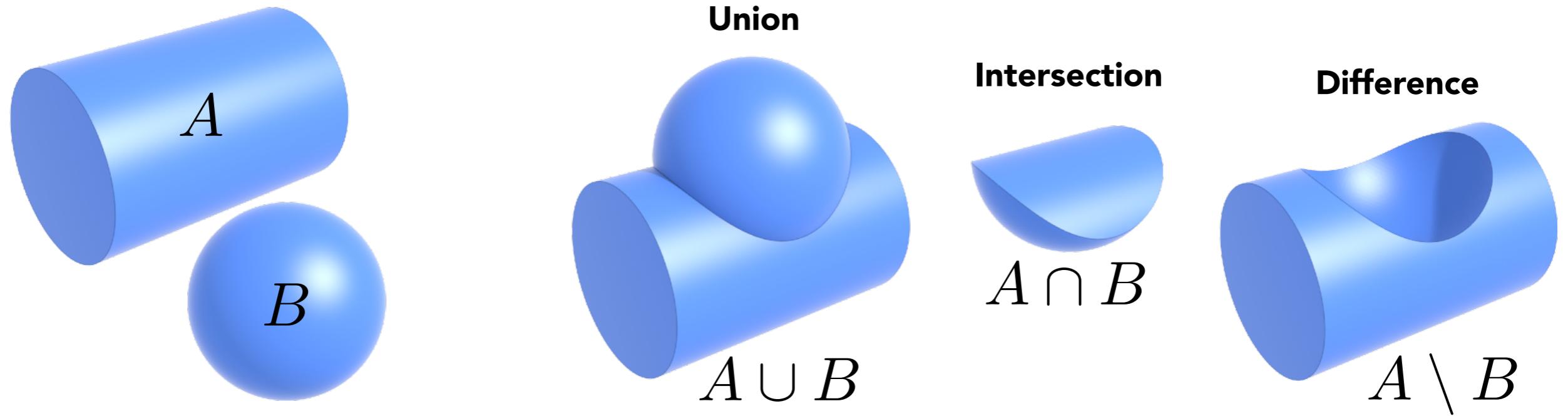
$$x^2 z^3 + \frac{9y^2 z^3}{80}$$



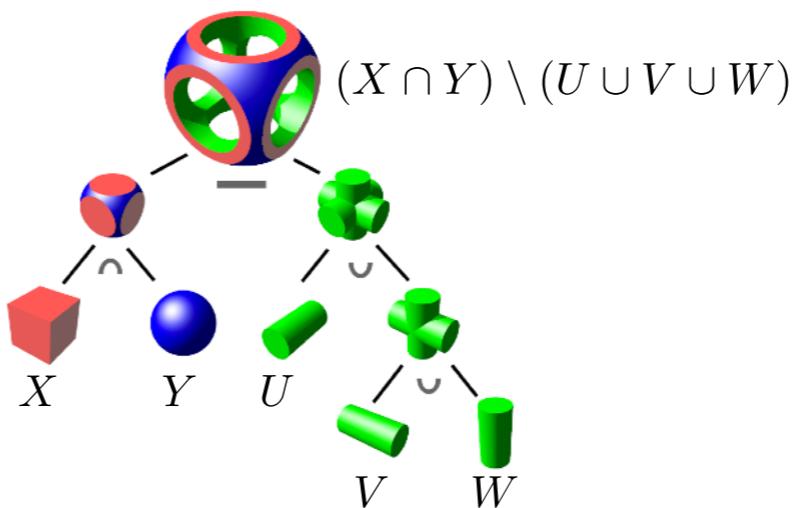
More complex shapes?

# Constructive Solid Geometry (Implicit)

# Combine implicit geometry via Boolean operations



# Boolean expressions:

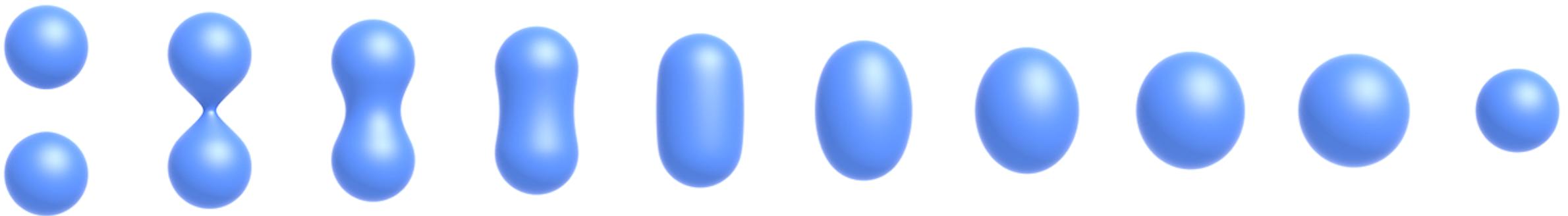


# Distance Functions (Implicit)

Instead of Booleans, gradually blend surfaces together using

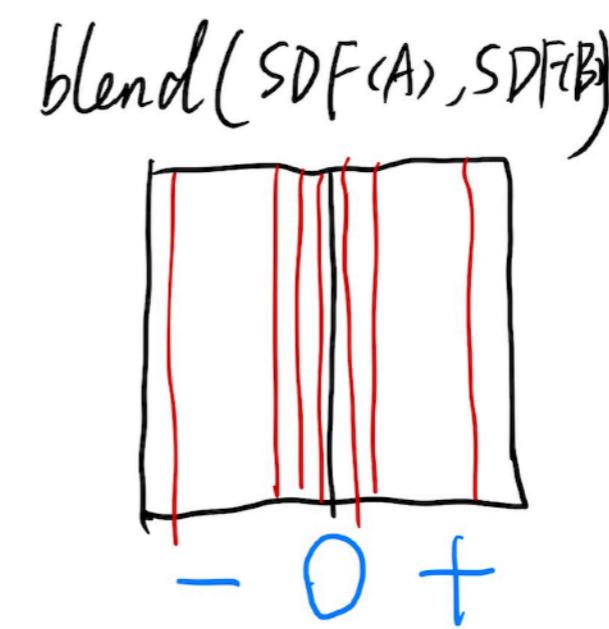
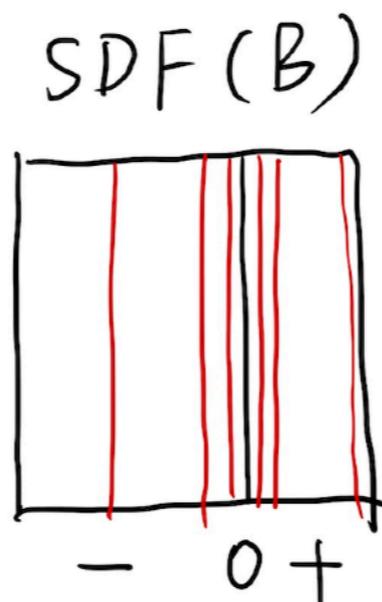
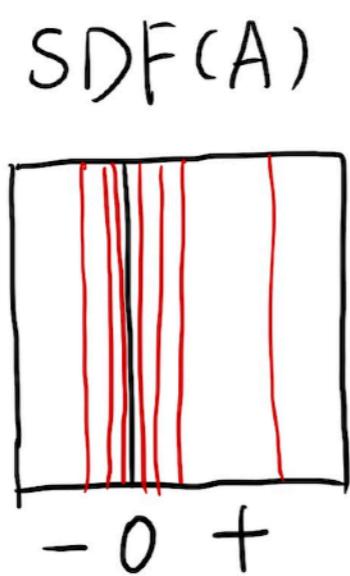
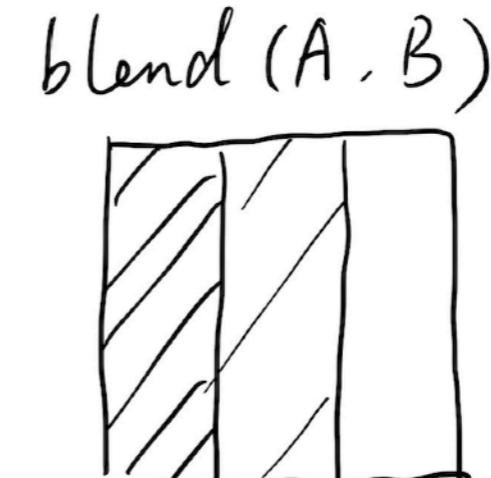
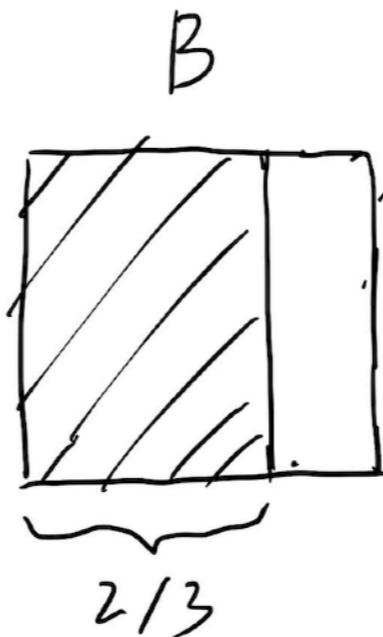
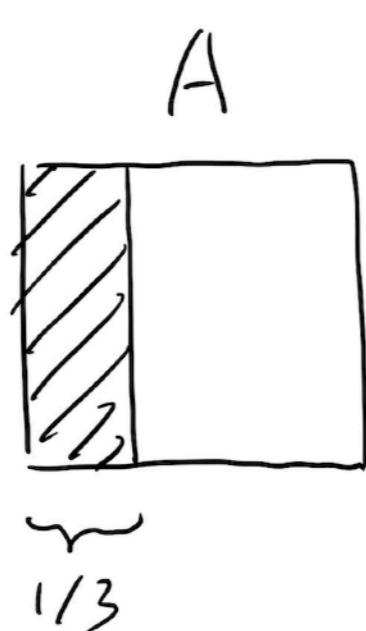
Distance functions:

giving minimum distance (could be **signed** distance)  
from anywhere to object



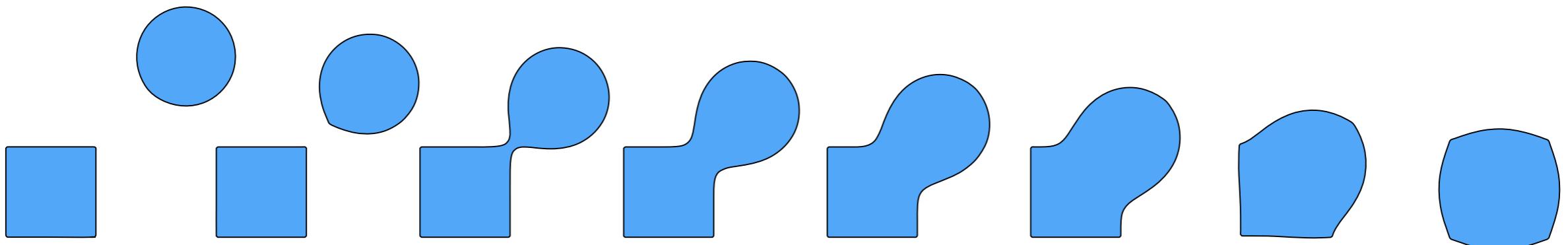
# Distance Functions (Implicit)

An Example: Blending (linear interp.) a moving boundary



# Blending Distance Functions (Implicit)

Can blend any two distance functions  $d_1, d_2$ :



# Scene of Pure Distance Functions



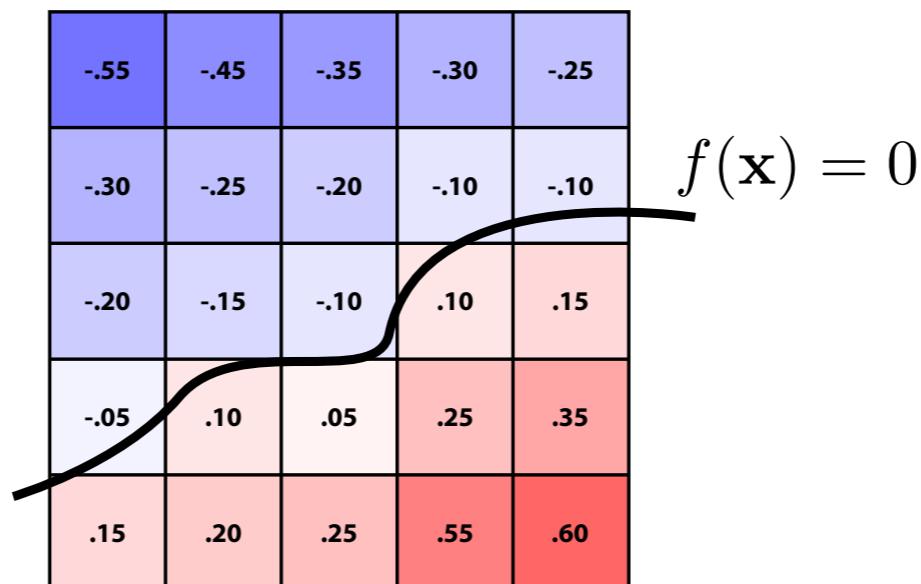
See <https://iquilezles.org/www/articles/raymarchingdf/raymarchingdf.htm>

# Level Set Methods (Also implicit)

(水平集)

Closed-form equations are hard to describe complex shapes

Alternative: store a grid of values approximating function

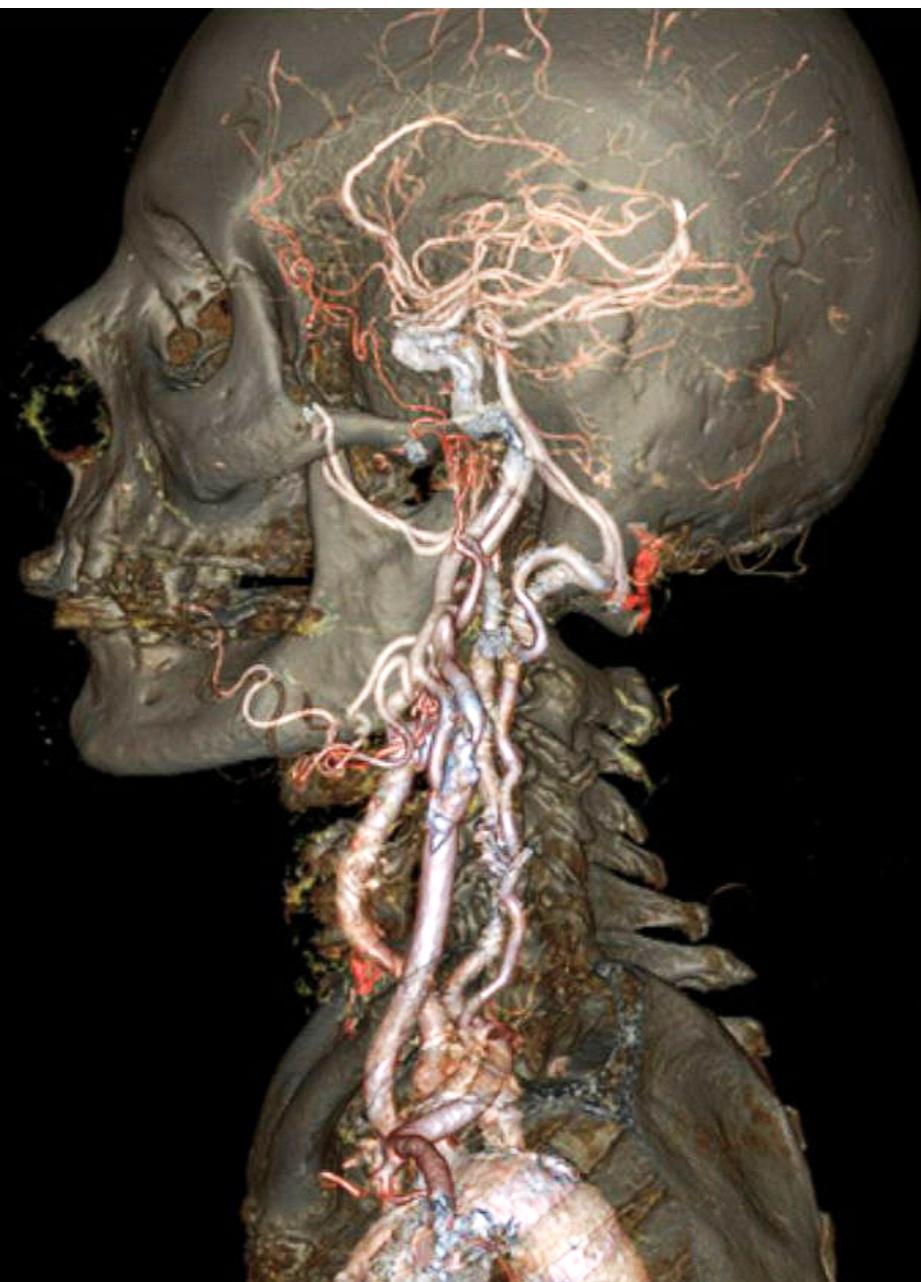


Surface is found where interpolated values equal zero

Provides much more explicit control over shape (like a texture)

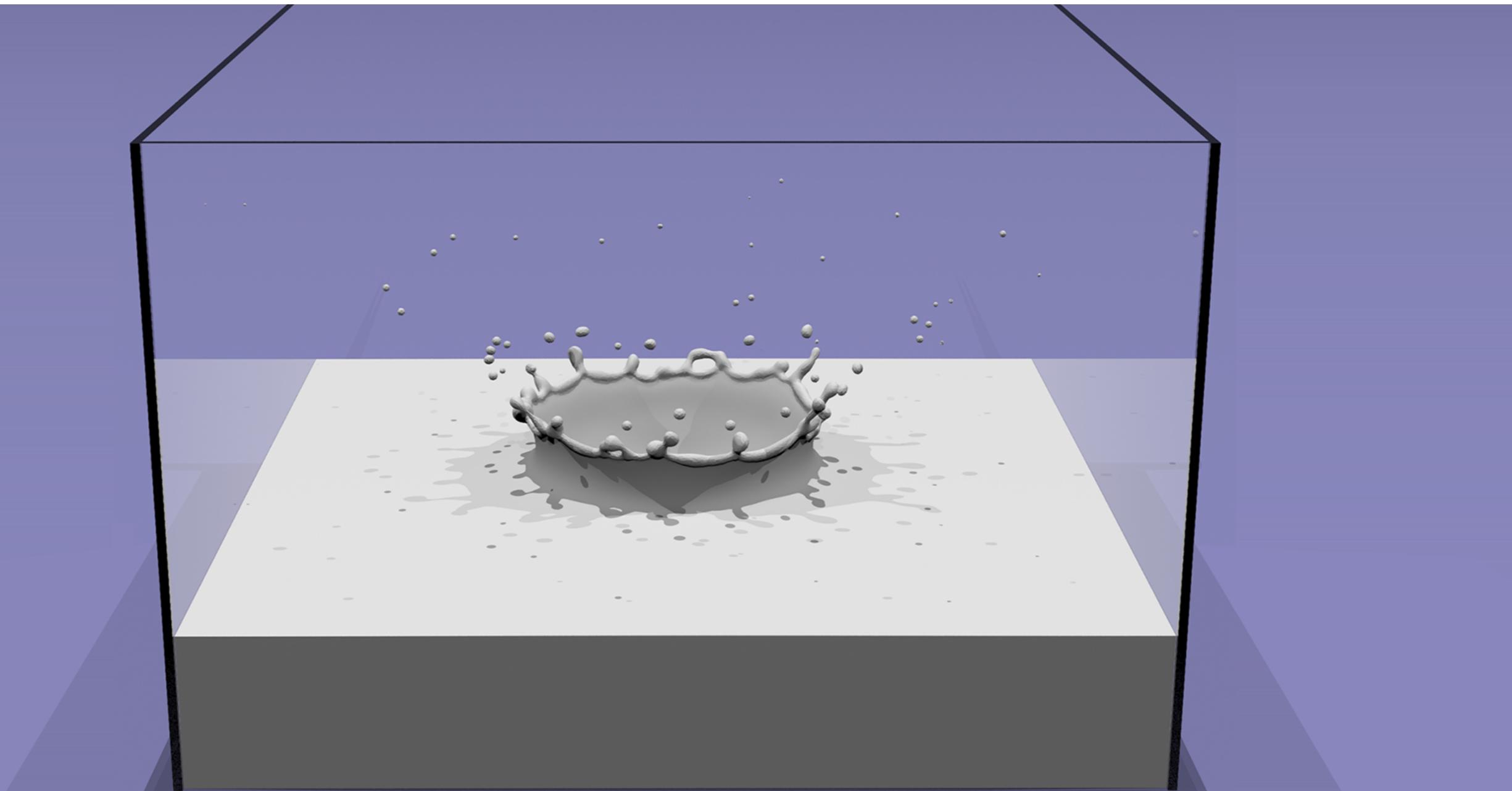
# Level Sets from Medical Data (CT, MRI, etc.)

Level sets encode, e.g., constant tissue density



# Level Sets in Physical Simulation

Level set encodes distance to air-liquid boundary



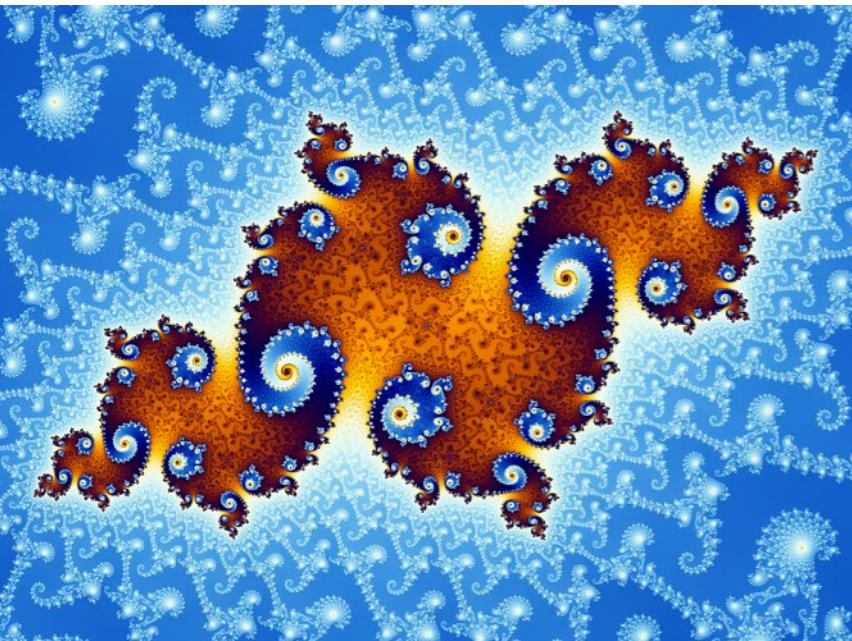
See <http://physbam.stanford.edu>

# Fractals (Implicit)

Exhibit self-similarity, detail at all scales

“Language” for describing natural phenomena

Hard to control shape!



# Implicit Representations - Pros & Cons

Pros:

- compact description (e.g., a function)
- certain queries easy (inside object, distance to surface)
- good for ray-to-surface intersection (more later)
- for simple shapes, exact description / no sampling error
- easy to handle changes in topology (e.g., fluid)

Cons:

- difficult to model complex shapes

# Thank you!

(And thank Prof. Ravi Ramamoorthi and Prof. Ren Ng for many of the slides!)