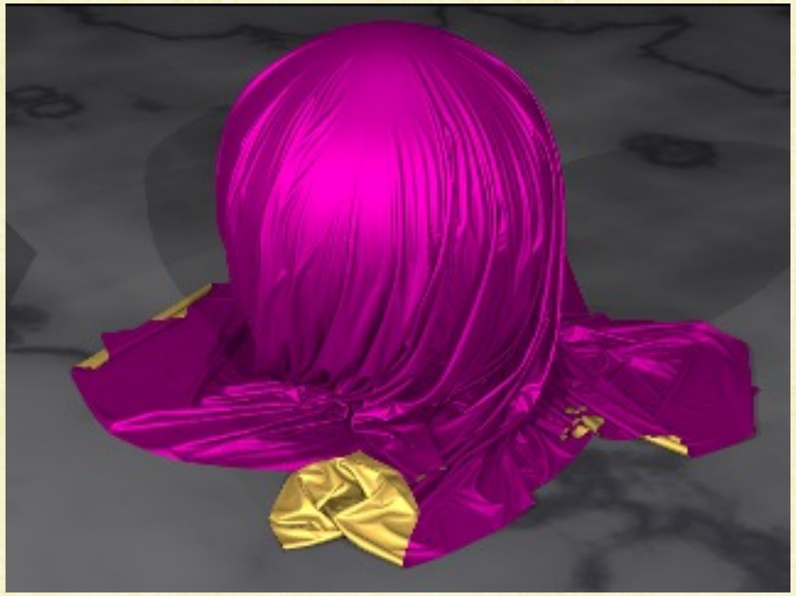
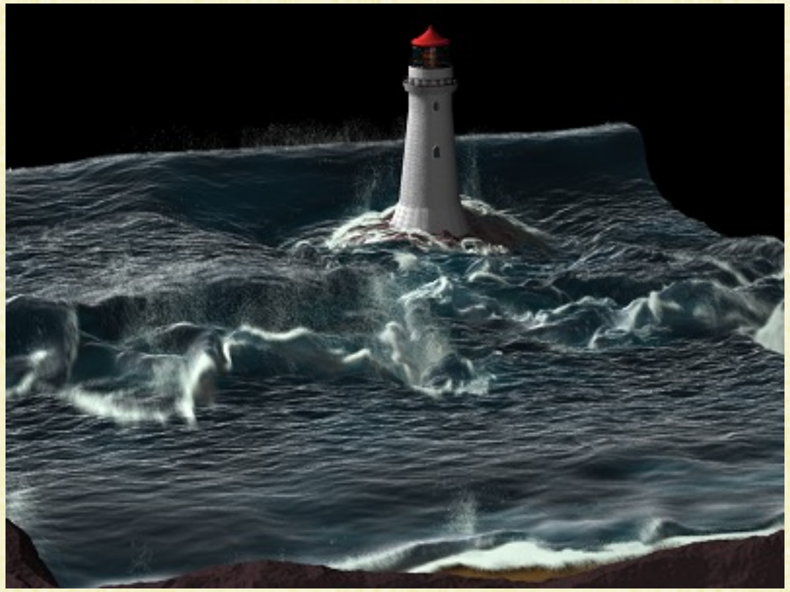
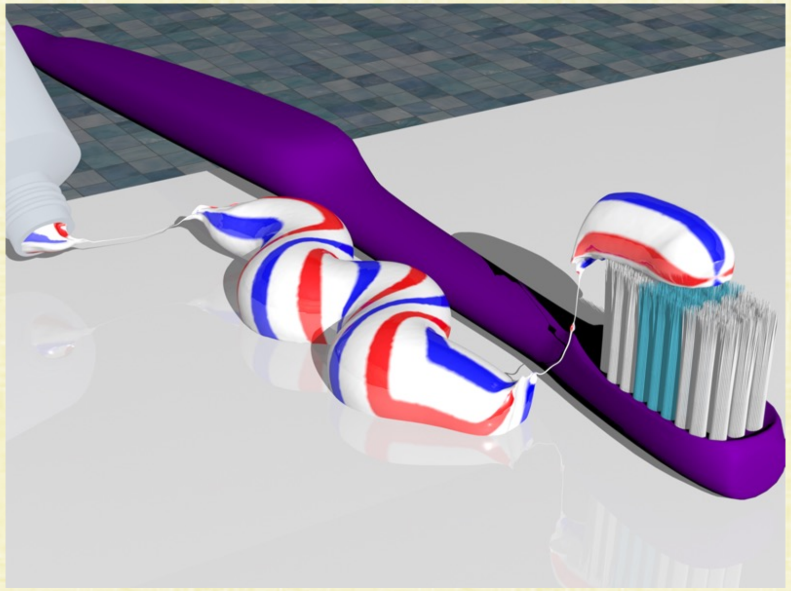
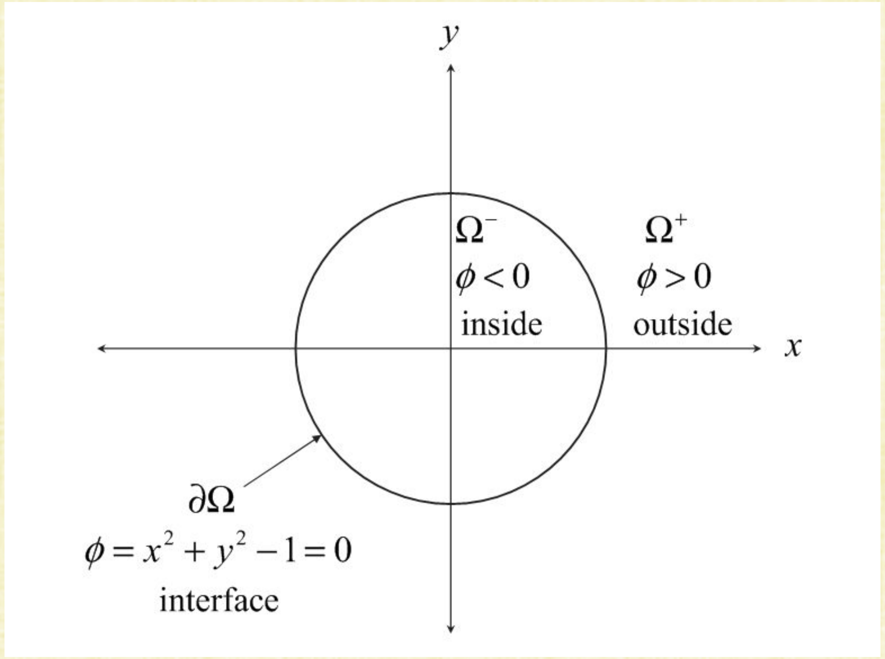
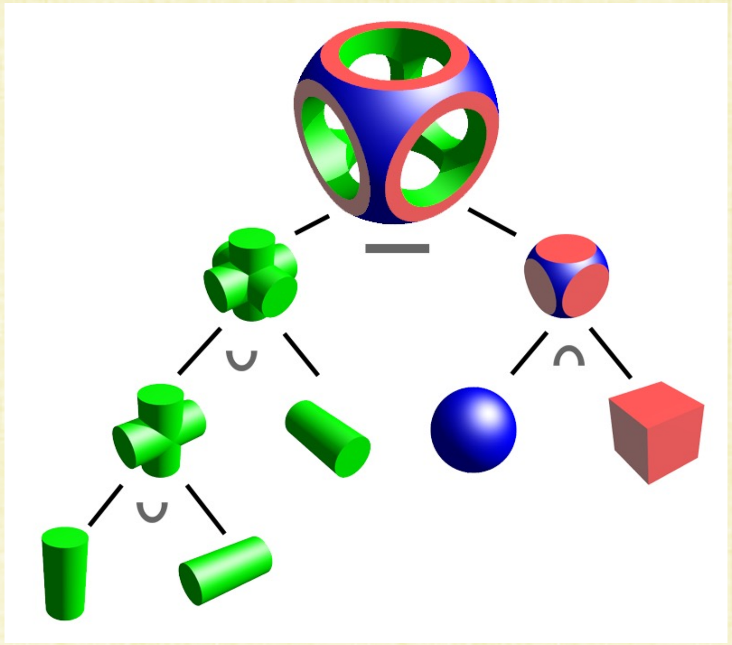
More Geometric Modeling

Implicit Surfaces

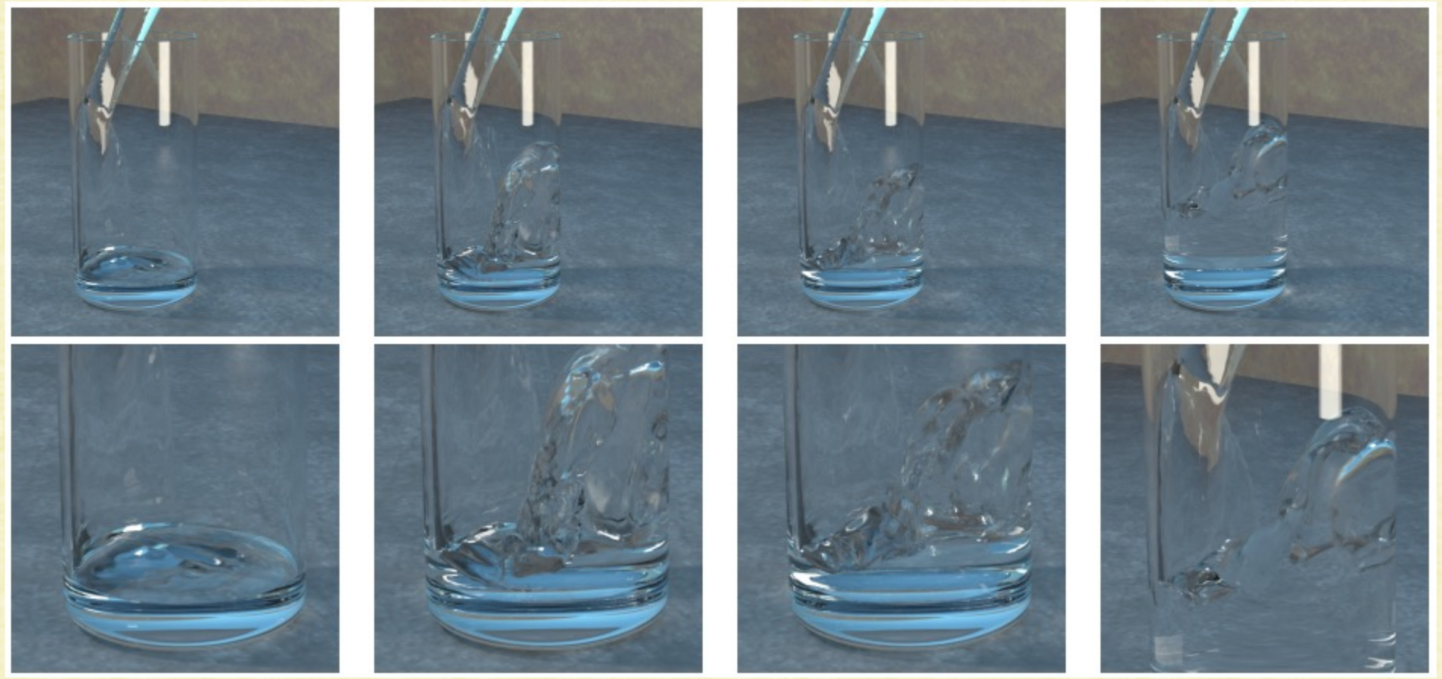
- Define a function $\phi(x)$ over $\forall x \in R^3$
- Interior Ω^- is defined by $\phi(x) < 0$, and exterior Ω^+ defined by $\phi(x) > 0$
- The surface $\partial\Omega$ is defined by $\phi(x) = 0$
    - We have already seen planes/spheres (and lines/rays/circles in 2D) defined by implicit surfaces
    - Easy to ray trace implicitly defined geometry
- Easy to check whether a point x is inside/outside: just evaluate $\phi(x)$
- Constructive Solid Geometry (CSG) operations: union, difference, intersection, etc.

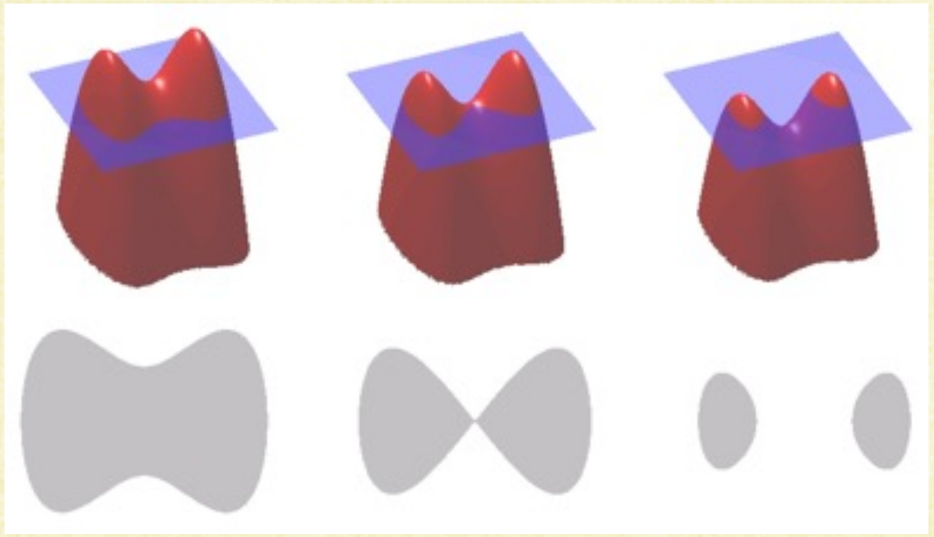


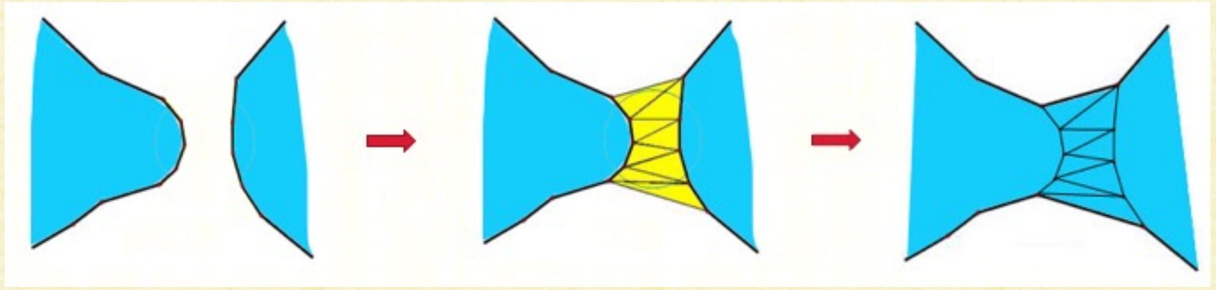


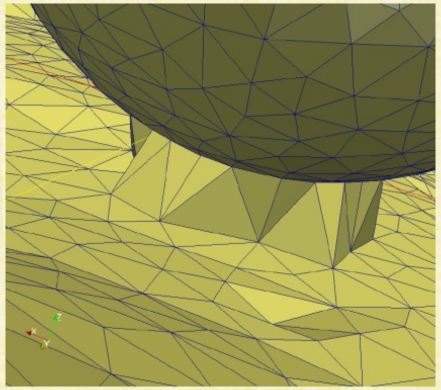
Topological Change

- Greatly superior to triangle meshes for topological change!



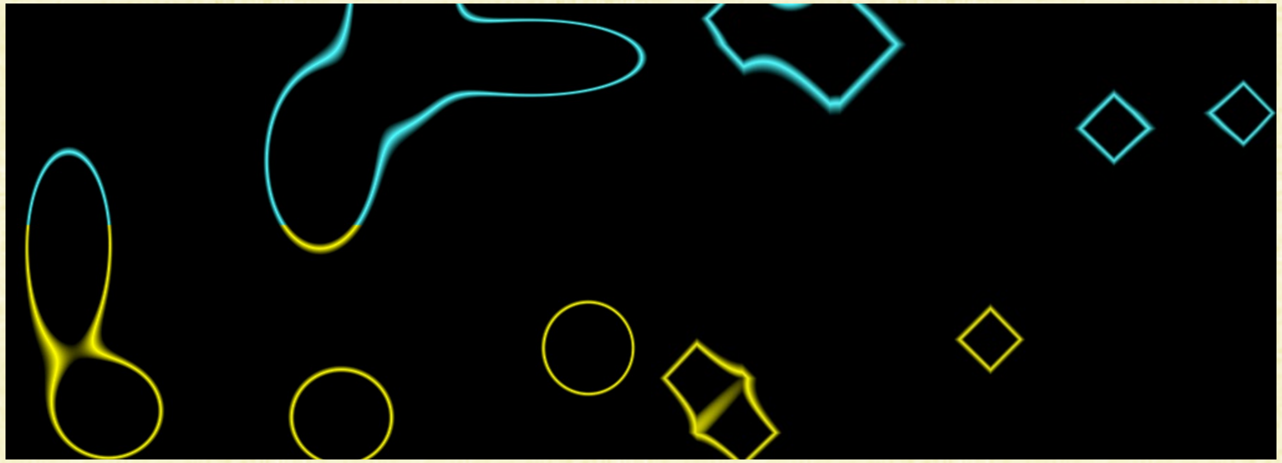


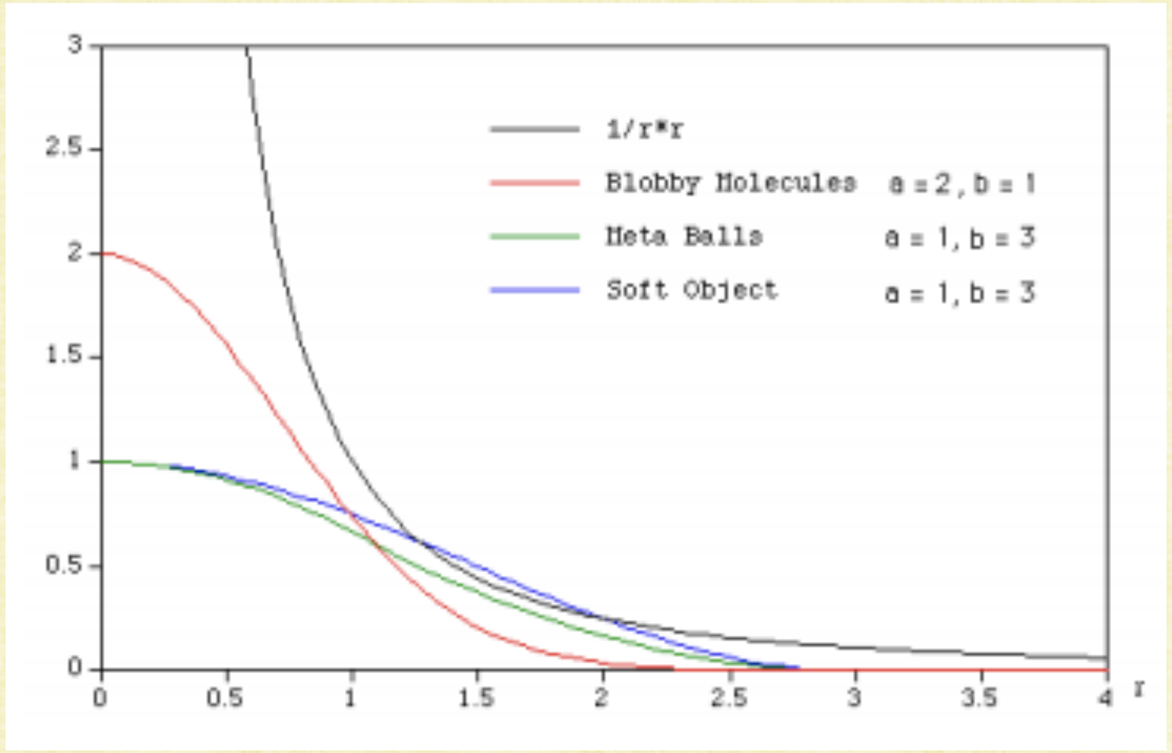




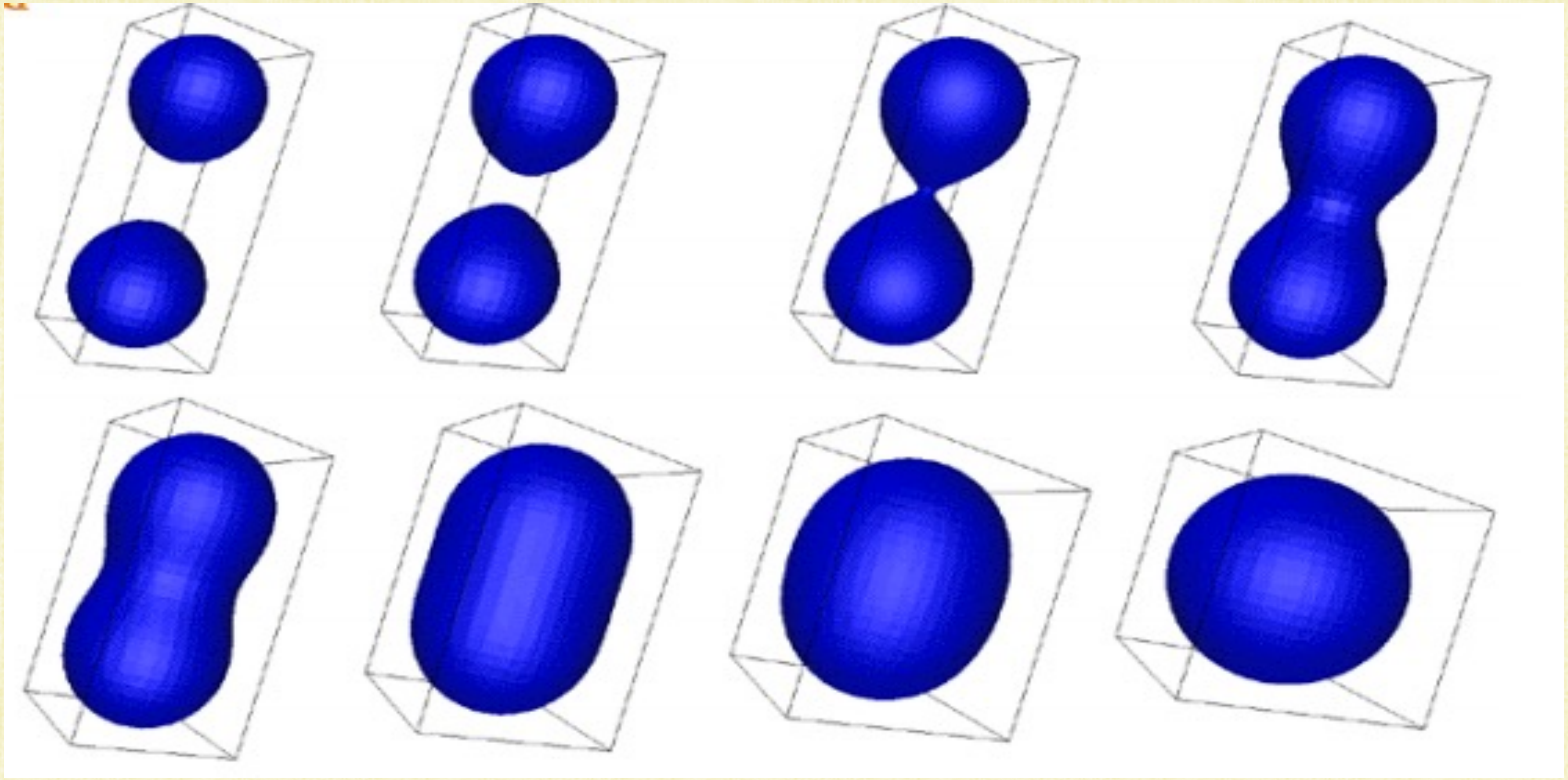
Blobbies

- Each blob is defined as a density function around a particle
- Blob kernels can be: 2D ellipses, 2D diamonds, 3D spheres, etc.
- For each pixel, the aggregate density is summed from all overlapping blobs
- Also known as:
  - Metaballs (in Japan), Soft objects (in Canada and New Zealand)
  - Slightly different density kernel functions

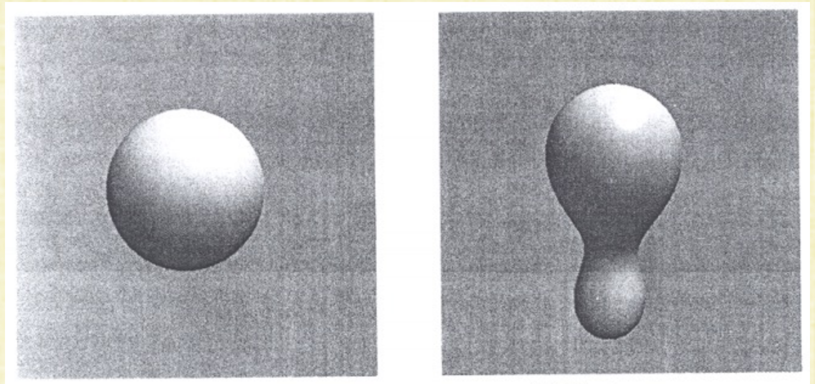


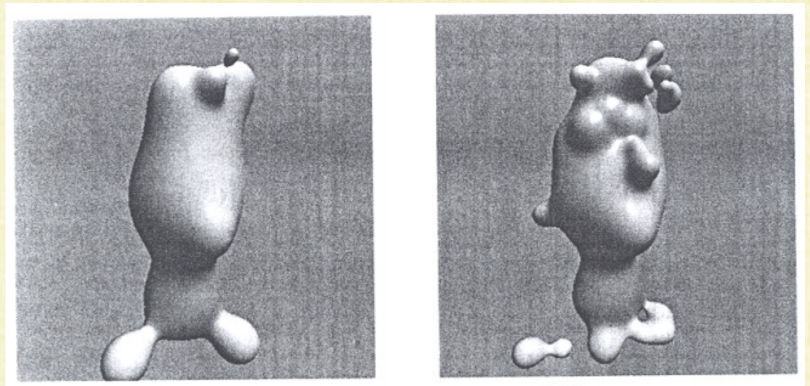


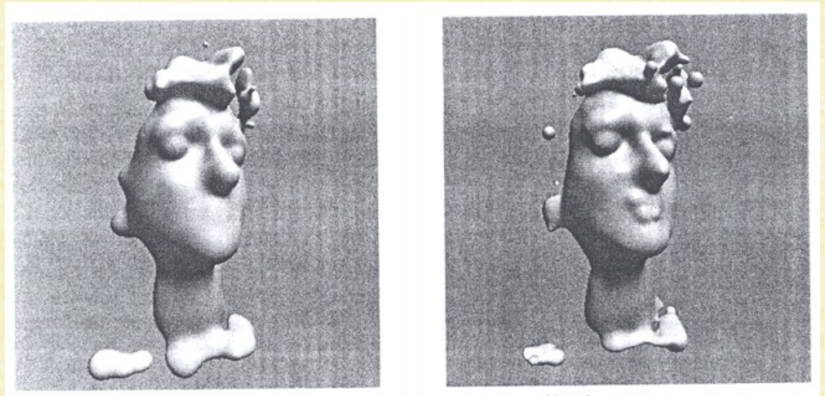
Topological Change

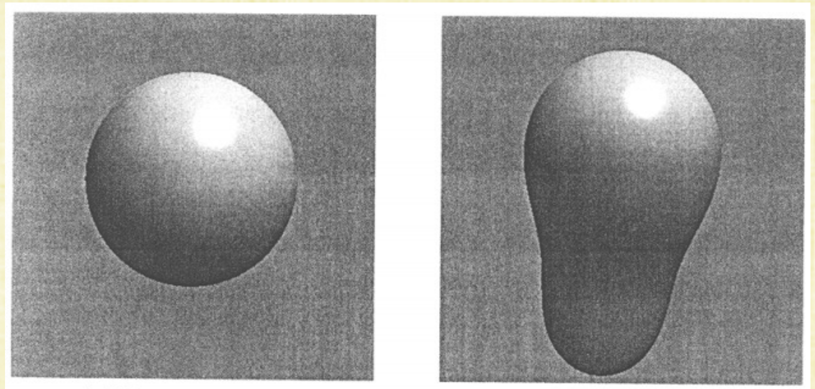


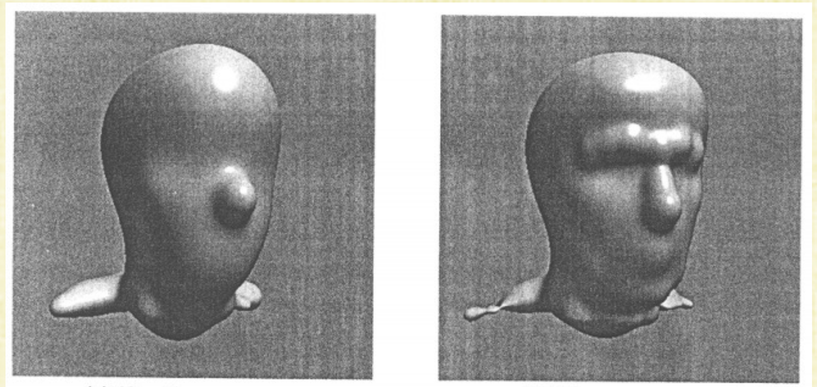
Blobby Modeling

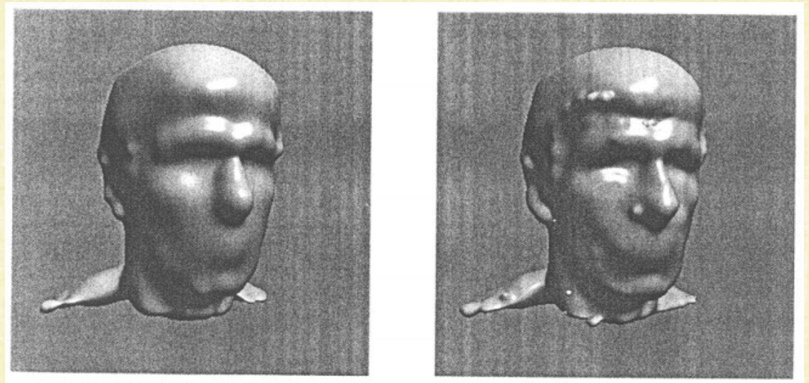

(a) $N = 1$ (b) $N = 2$


(c) $N = 10$ (d) $N = 35$


(e) $N = 70$ (f) $N = 243$

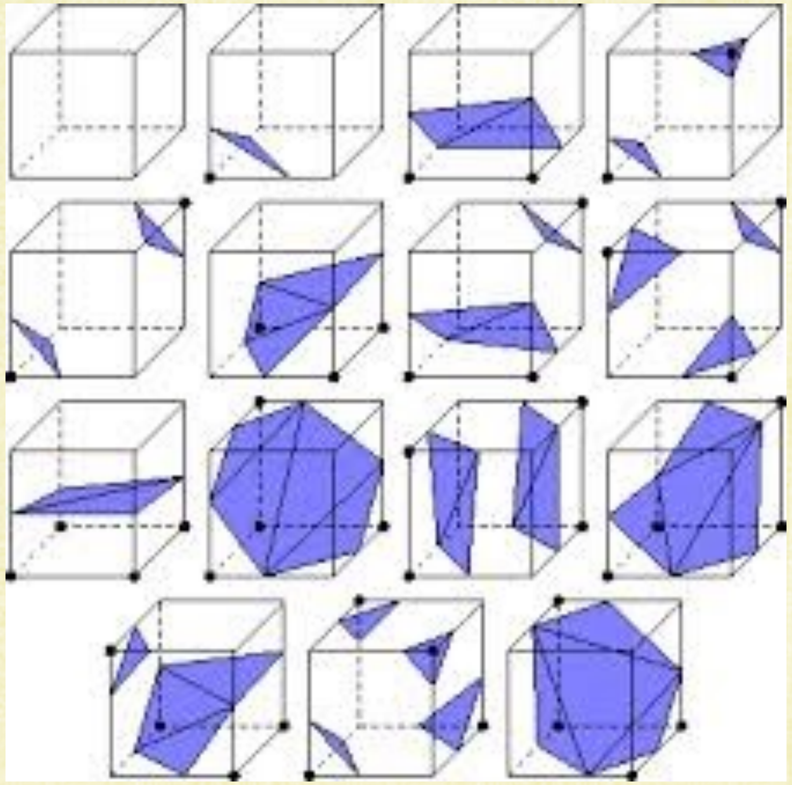

(a) $N = 1$ (b) $N = 2$

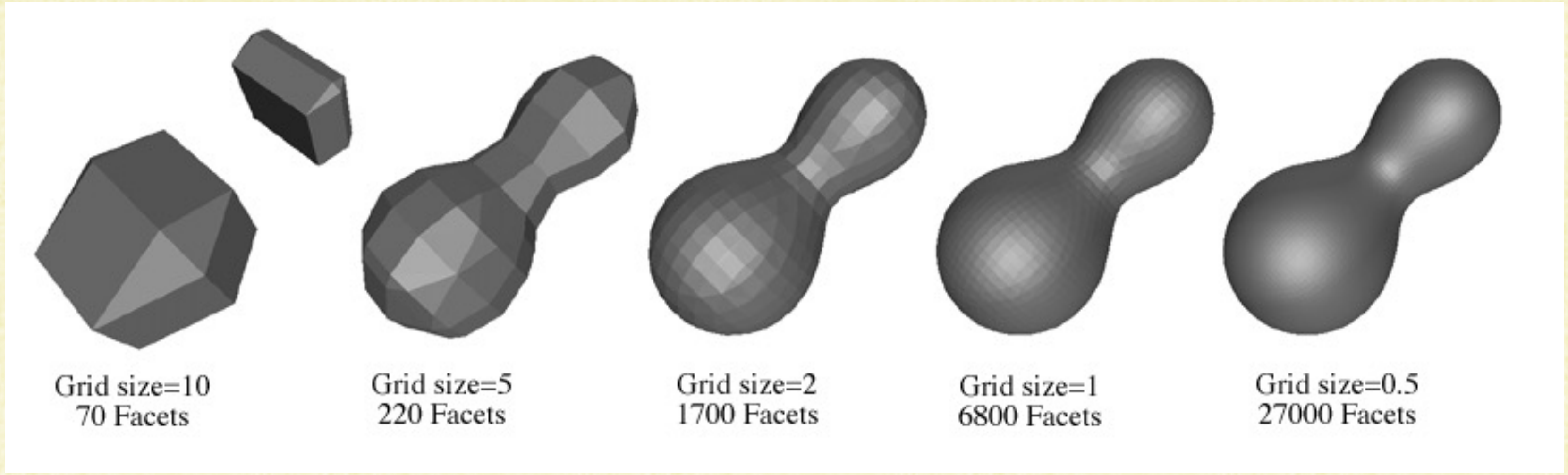

(c) $N = 20$ (d) $N = 60$


(e) $N = 120$ (f) $N = 451$

Marching Cubes (or Marching Tetrahedra)

- Turns an implicit surface into triangles
  - Define the implicit surface on a 3D grid
  - For each grid cell, use the topology of the volume to construct surface triangles

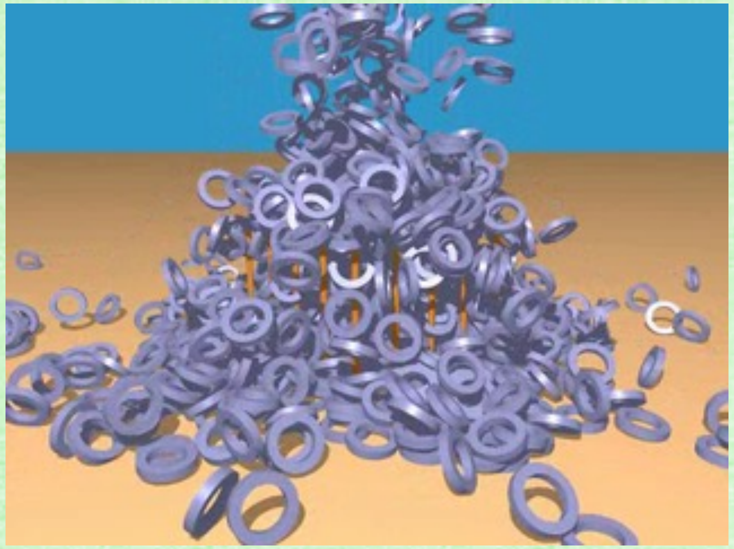
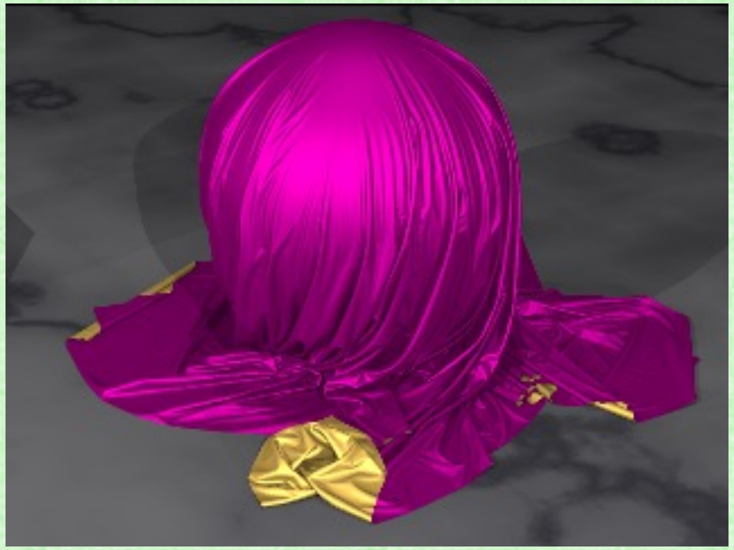
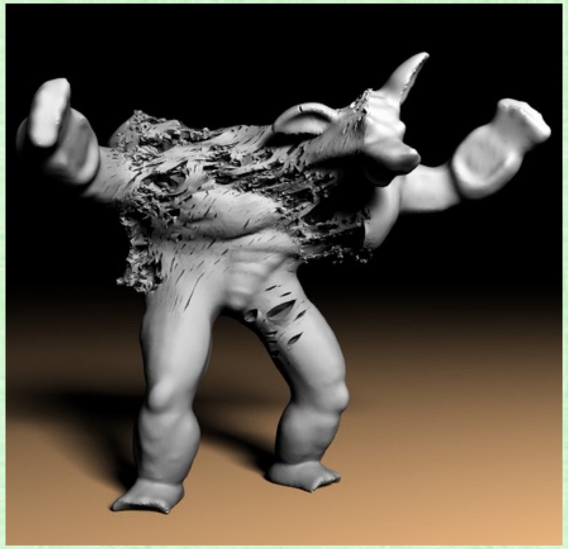
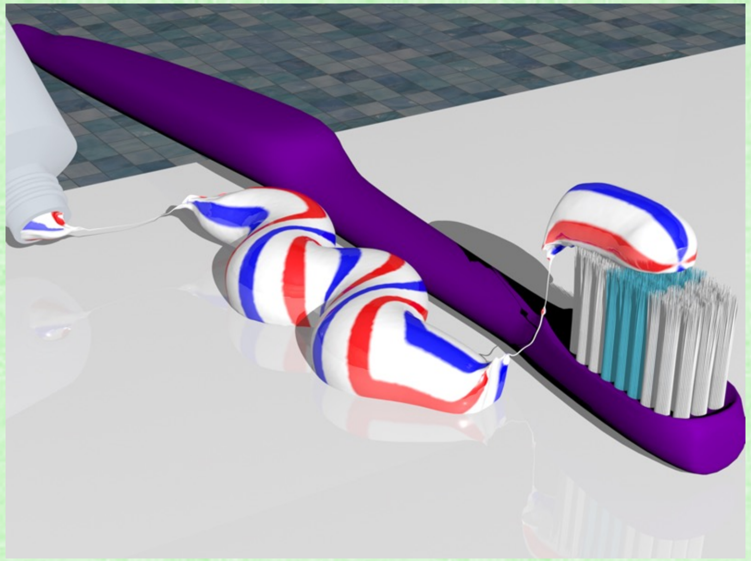
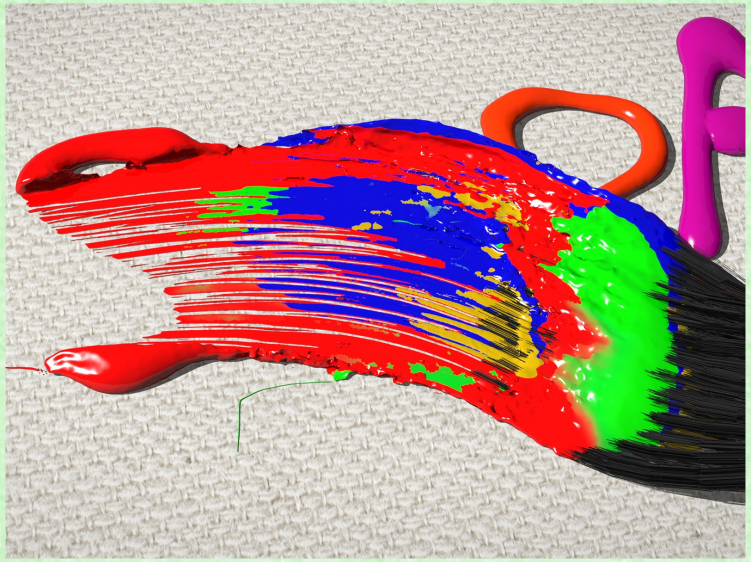
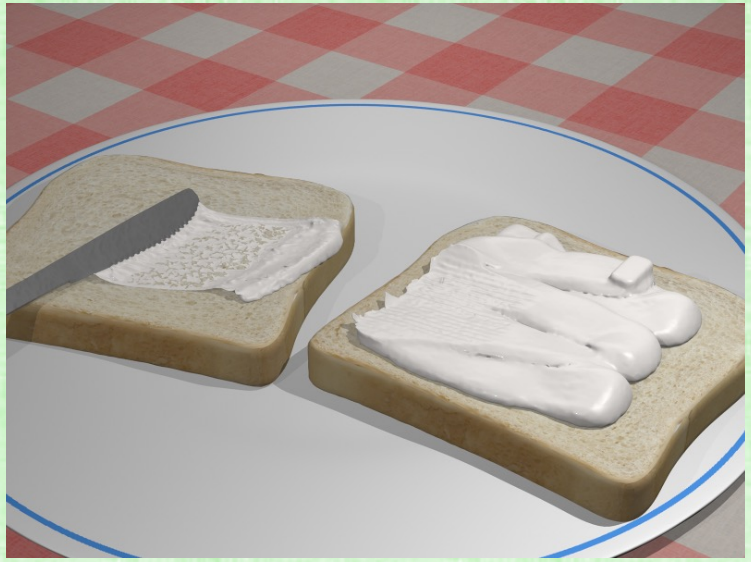




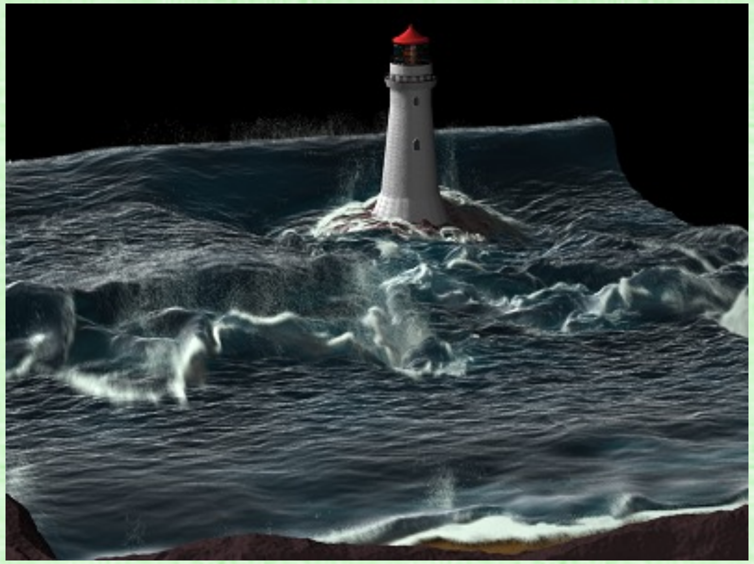
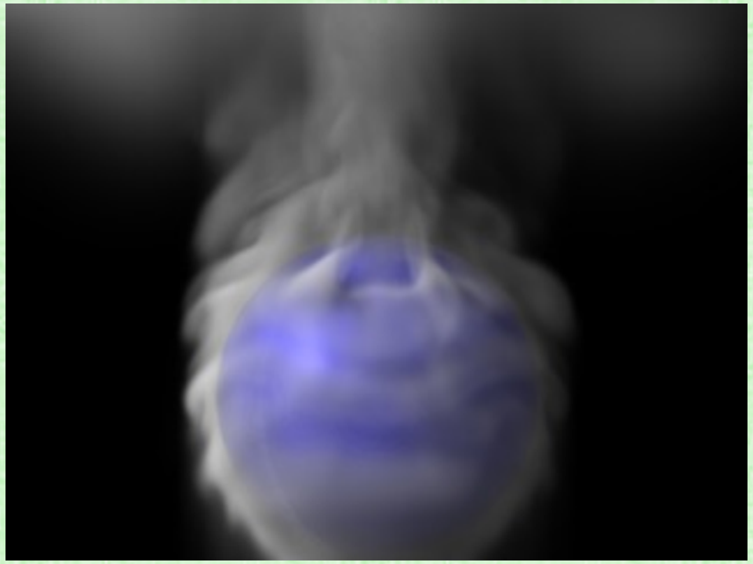
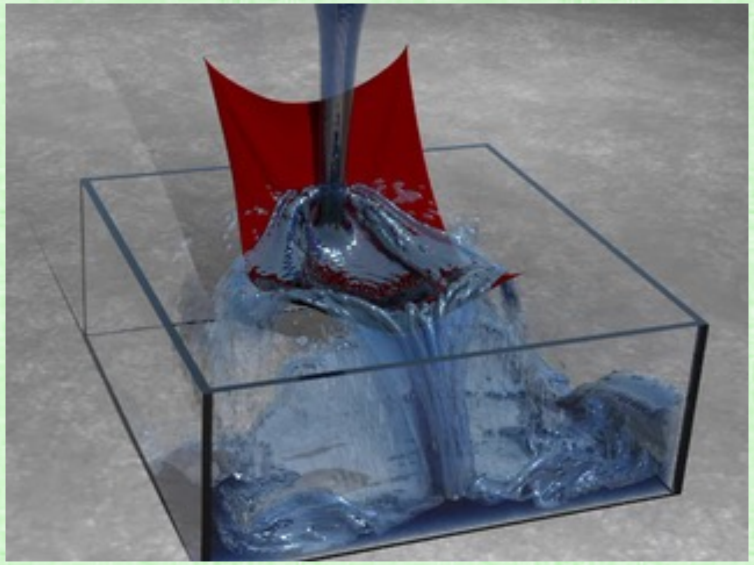
Netwon's Second Law (for Physics Simulations)

- Kinematics describe position $X(t)$ and velocity $V(t)$ as function of time t
  - $\frac{dX(t)}{dt} = V(t)$ or $X'(t) = V(t)$
- Dynamics describe responses to external stimuli
  - Newton's second law $F(t) = MA(t)$ is a dynamics equation
  - $V'(t) = A(t)$ implies $V'(t) = \frac{F(t)}{M}$ as well as $\frac{\mathrm{d}^2 X(t)}{dt^2} = X''(t) = \frac{F(t)}{M}$
- Combining kinematics and dynamics gives: $\begin{pmatrix} X'(t) \\ V'(t) \end{pmatrix} = \begin{pmatrix} V(t) \\ \frac{F(t, X(t), V(t))}{M} \end{pmatrix}$
  - Note: forces often depend on position/velocity
- Much of the physical world can be simulated with computational mechanics (FEM) and computational fluid dynamics (CFD), using Newton's second law
  - Create degrees of freedom, specify forces, and solve the resulting ordinary differential equations (ODEs)
  - Spatially interdependent forces lead to partial differential equations (PDEs)

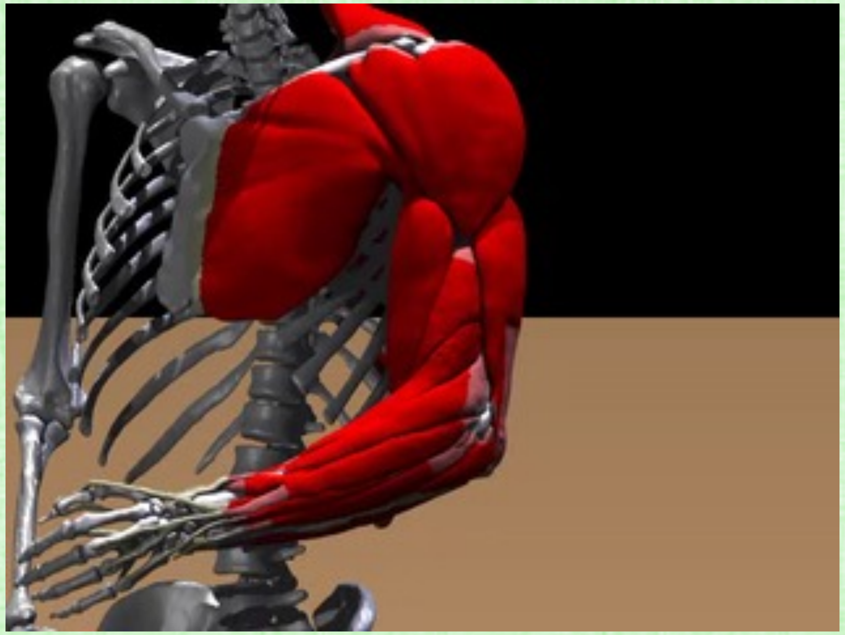
Computational Mechanics (FEM)

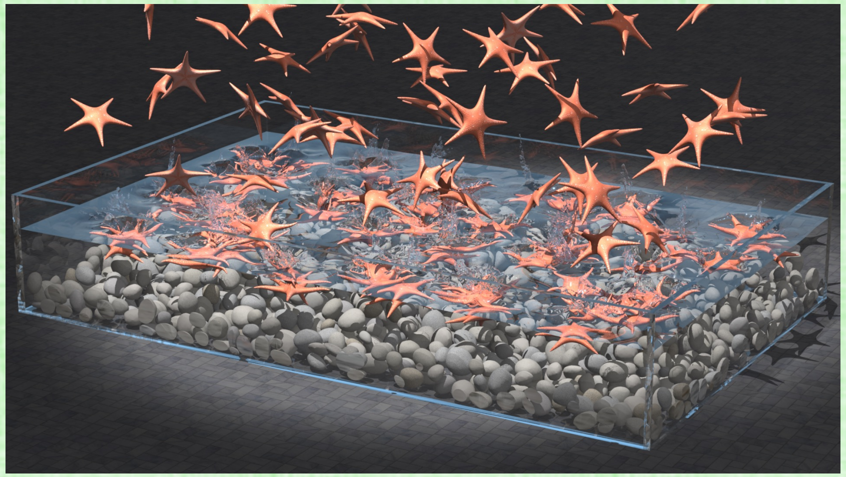







Computational Fluid Dynamics (CFD)

Computational Biomechanics

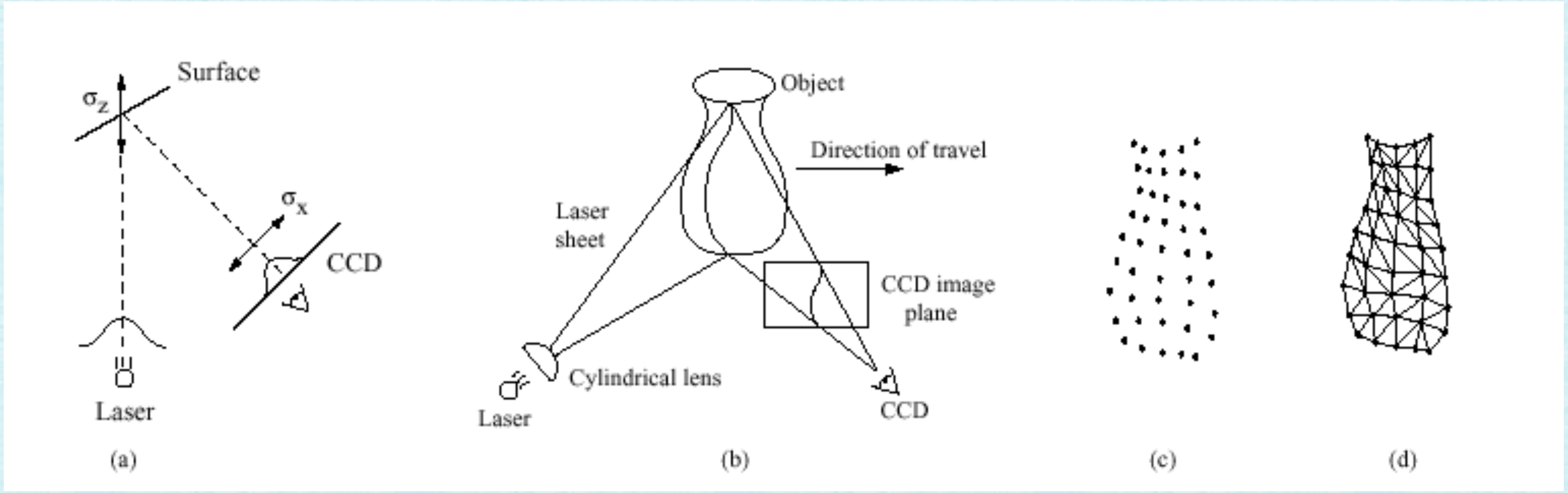


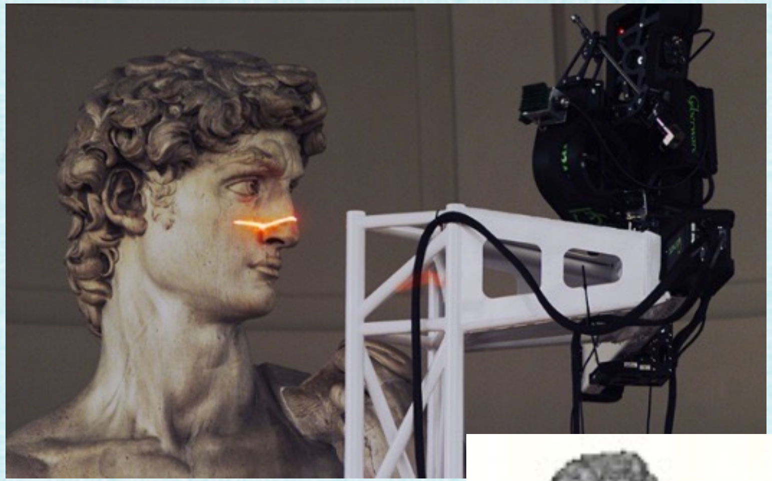


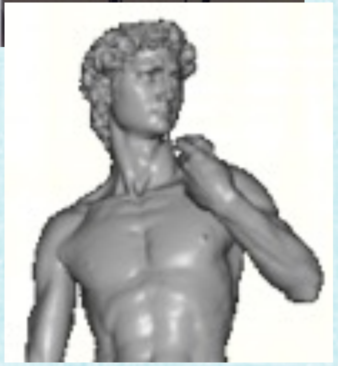
Computer Vision

Range Scanners

- Senses 3D positions on an object's surface, and returns a range image:
  - $m \times n$ grid of distances (m points per laser sheet, n laser sheets)
- Multiple range images are aligned with transformations
  - Transformations determined via Iterative Closest Point (ICP) and related/similar methods
- Aligned range images are combined using a zippering algorithm







Range Scanners

- Each sample point in the $m{\times}n$ range image is a potential vertex in a triangle mesh
- Special care is required to avoid joining together vertices separated by depth discontinuities

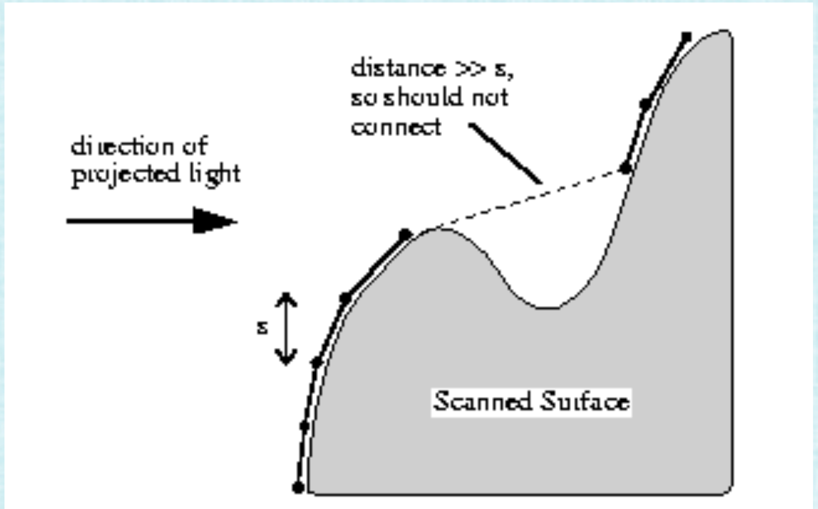
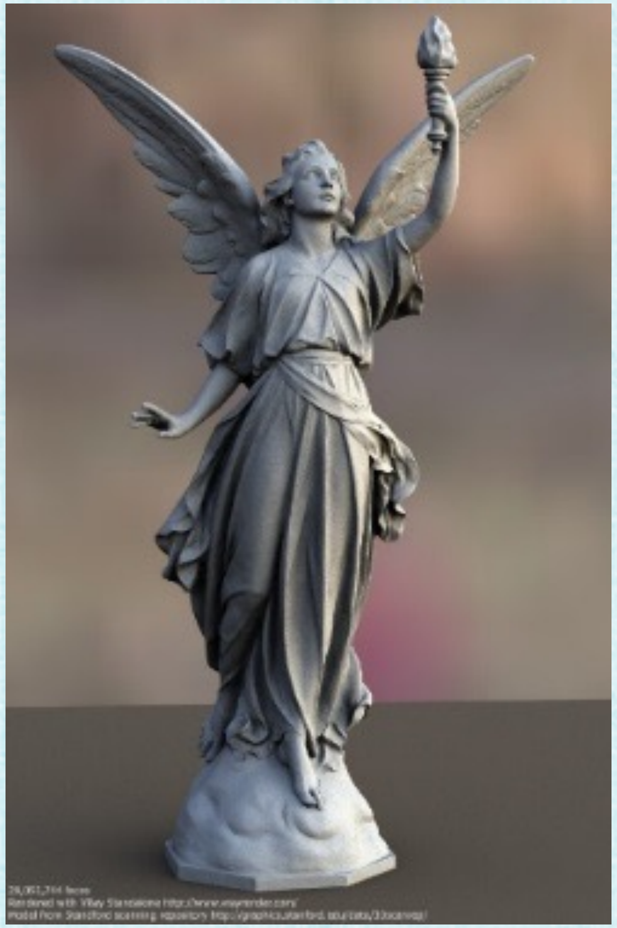
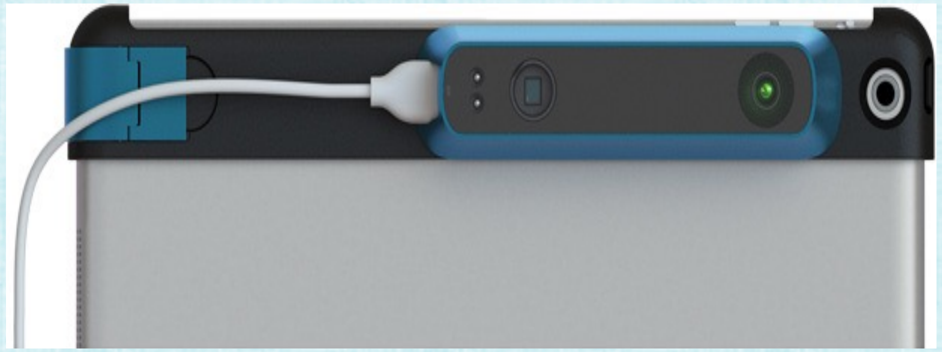
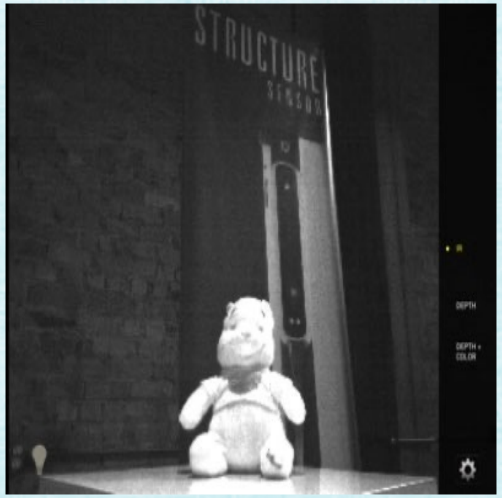


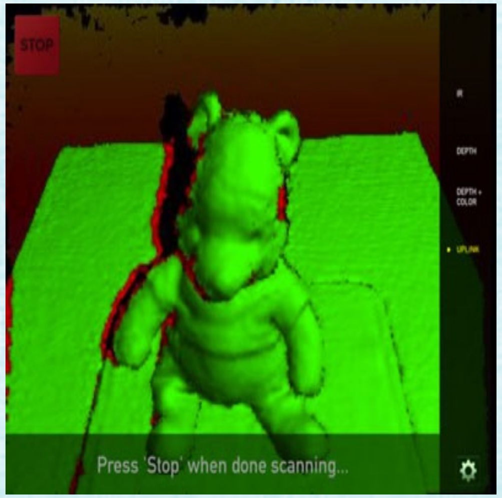
Figure 3: Building triangle mesh from range points.



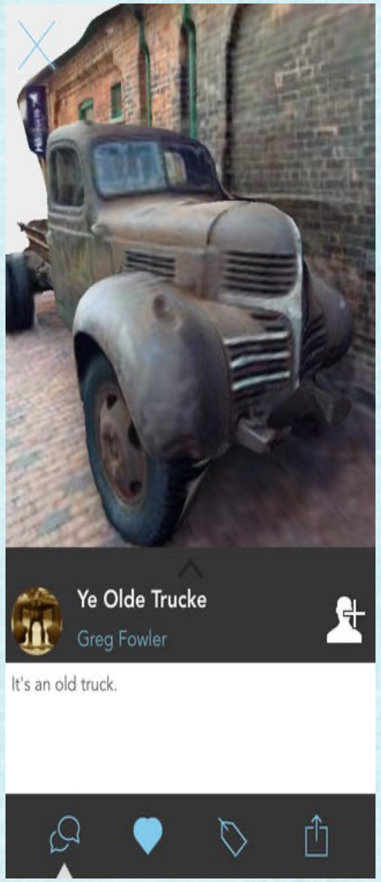
Scanning w/Mobile Devices







Structure Sensor for iPad



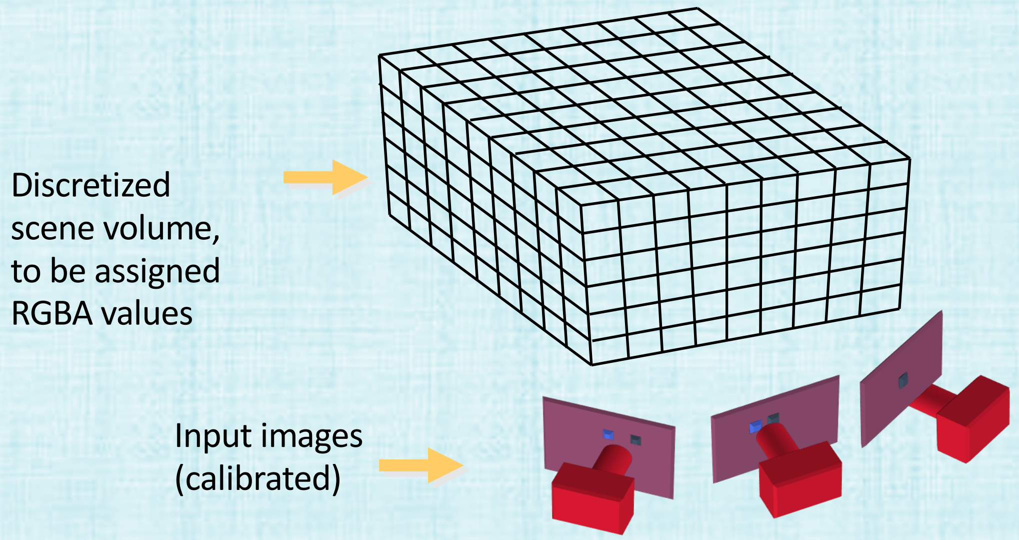
It's an old truck.

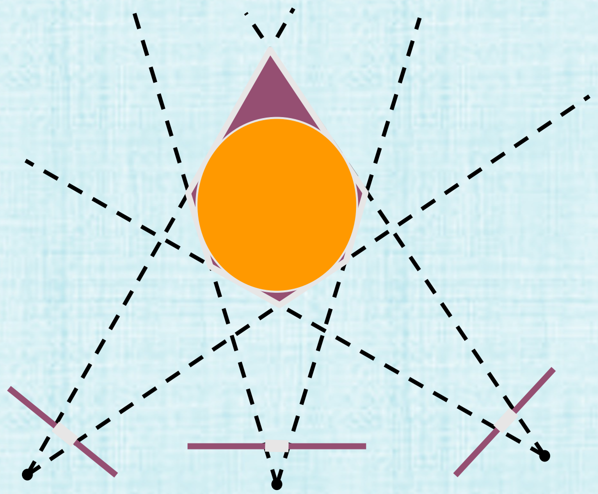
AUTODESK® 123D®

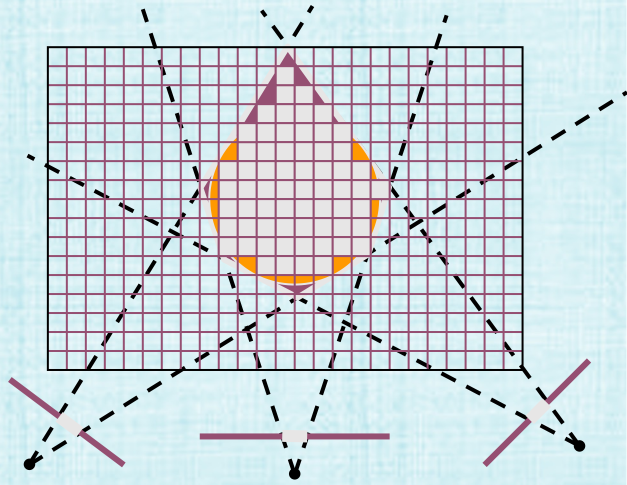
Autodesk 123D Catch

Voxel Carving

- Requires multiple images (from calibrated cameras) taken from different directions
- Construct a voxelized 3D model:
  - For each image, delete (carve away) voxels outside the object silhouette
- Colors can be projected onto the geometry


Discretized scene volume, to be assigned RGBA values
Input images (calibrated)

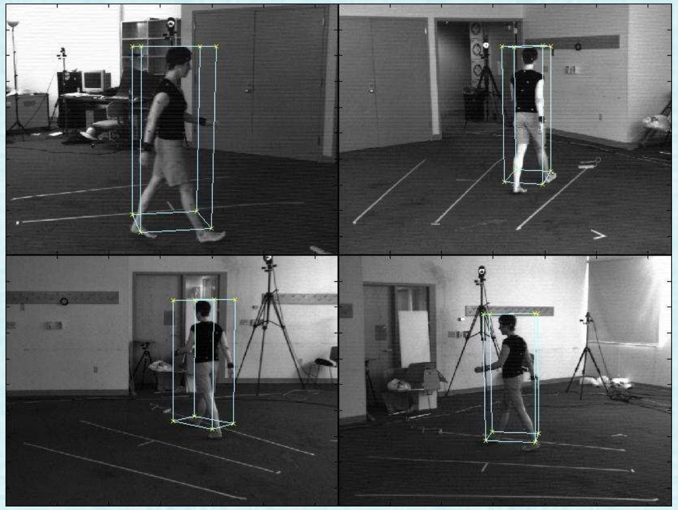




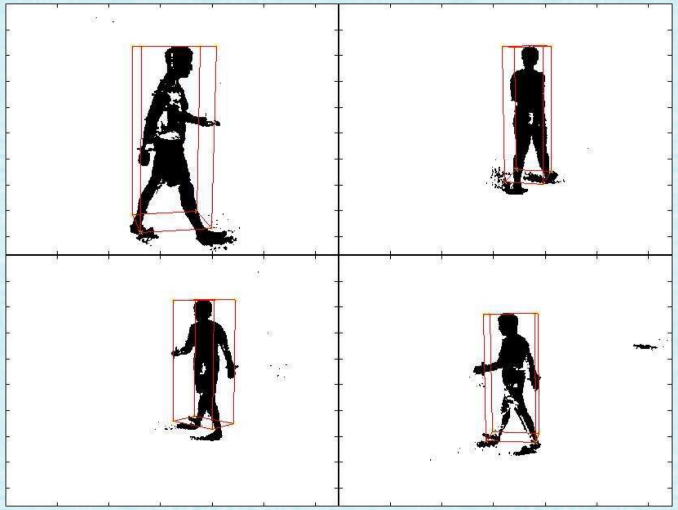
Color the voxel gray if silhouette is in every image

Voxel Carving

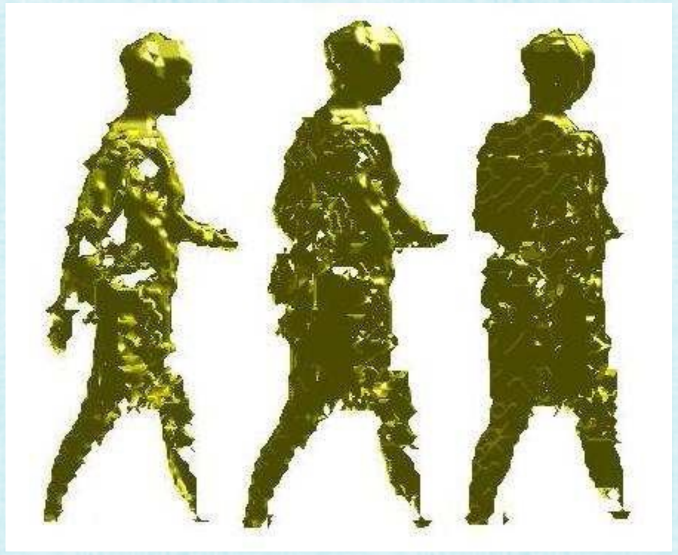
Original image



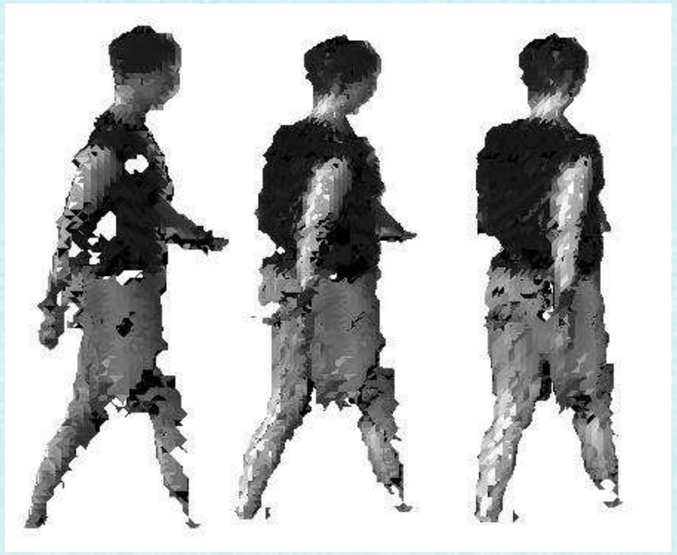
Extracted silhouettes



Carved out voxels

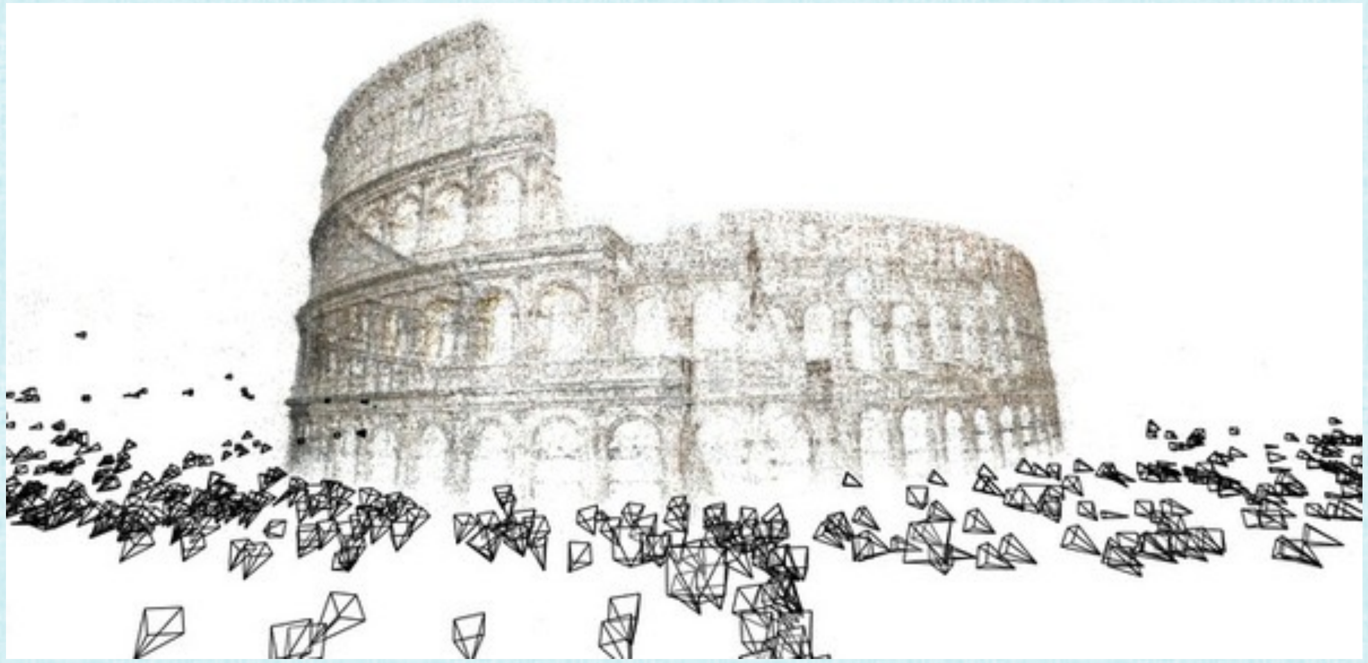


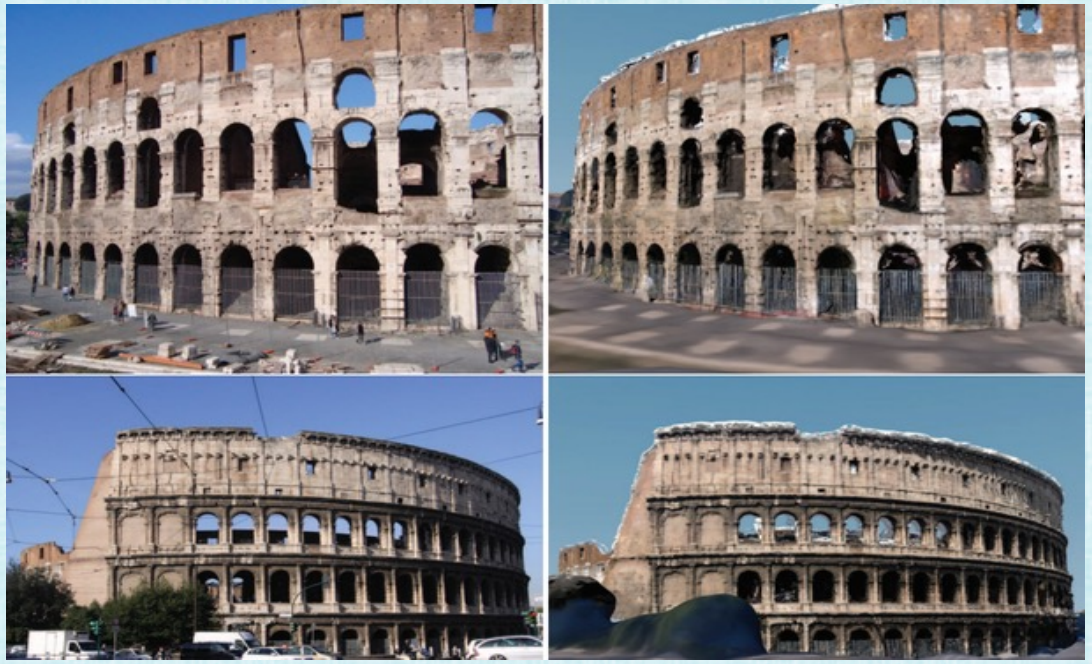
Back-projecting colors



Reconstruction from Large Photo Collections

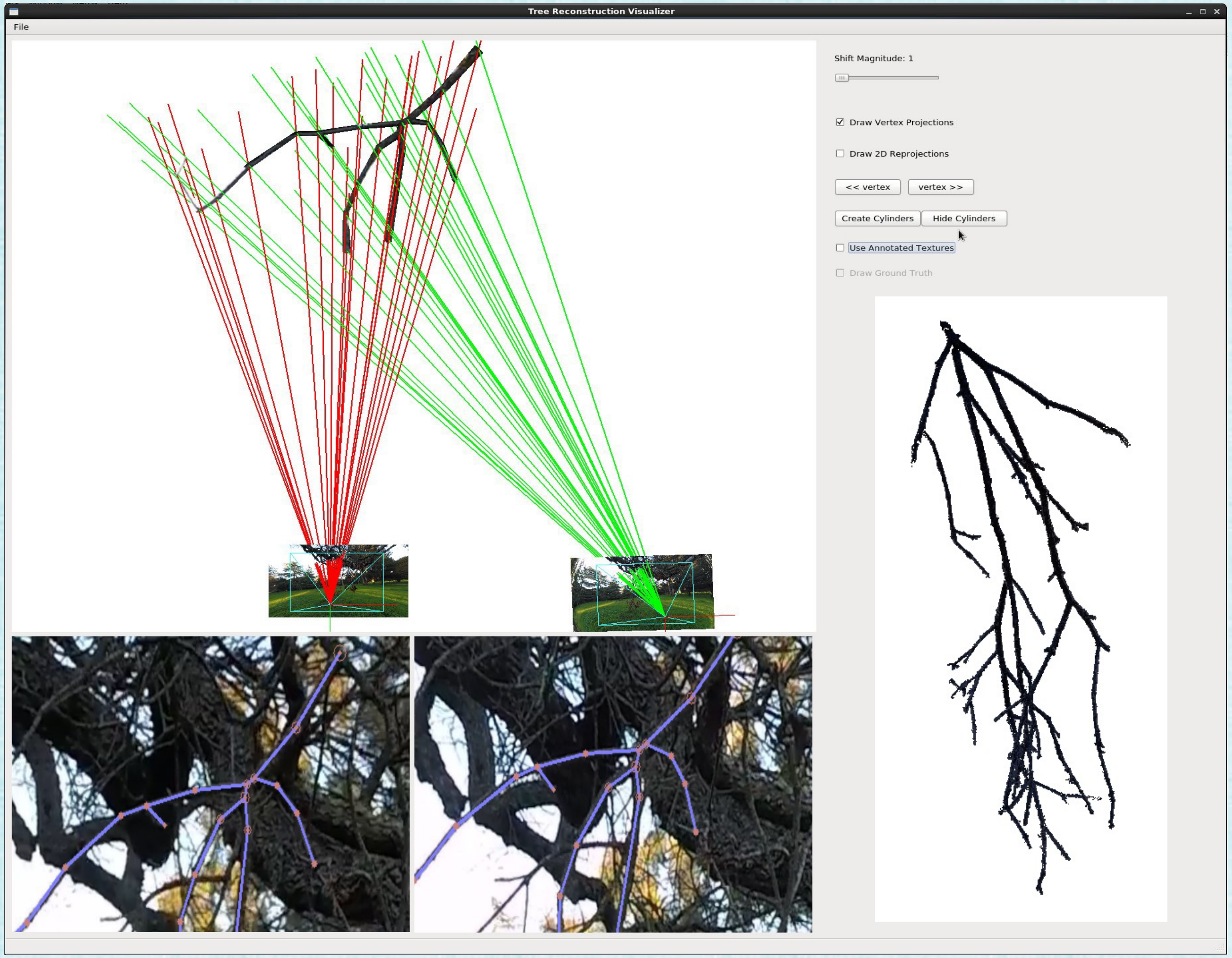
- Collect a large number of photos (e.g. from google images)
- Predict relative camera position/orientation for each image
- The position of a point that is visible in multiple images can be triangulated
- Obtain a sparse point cloud representation of the object
- Dense reconstruction algorithms can be used to improve the results





2D photos 3D geometry

Drones & Trees



Shift Magnitude: 1

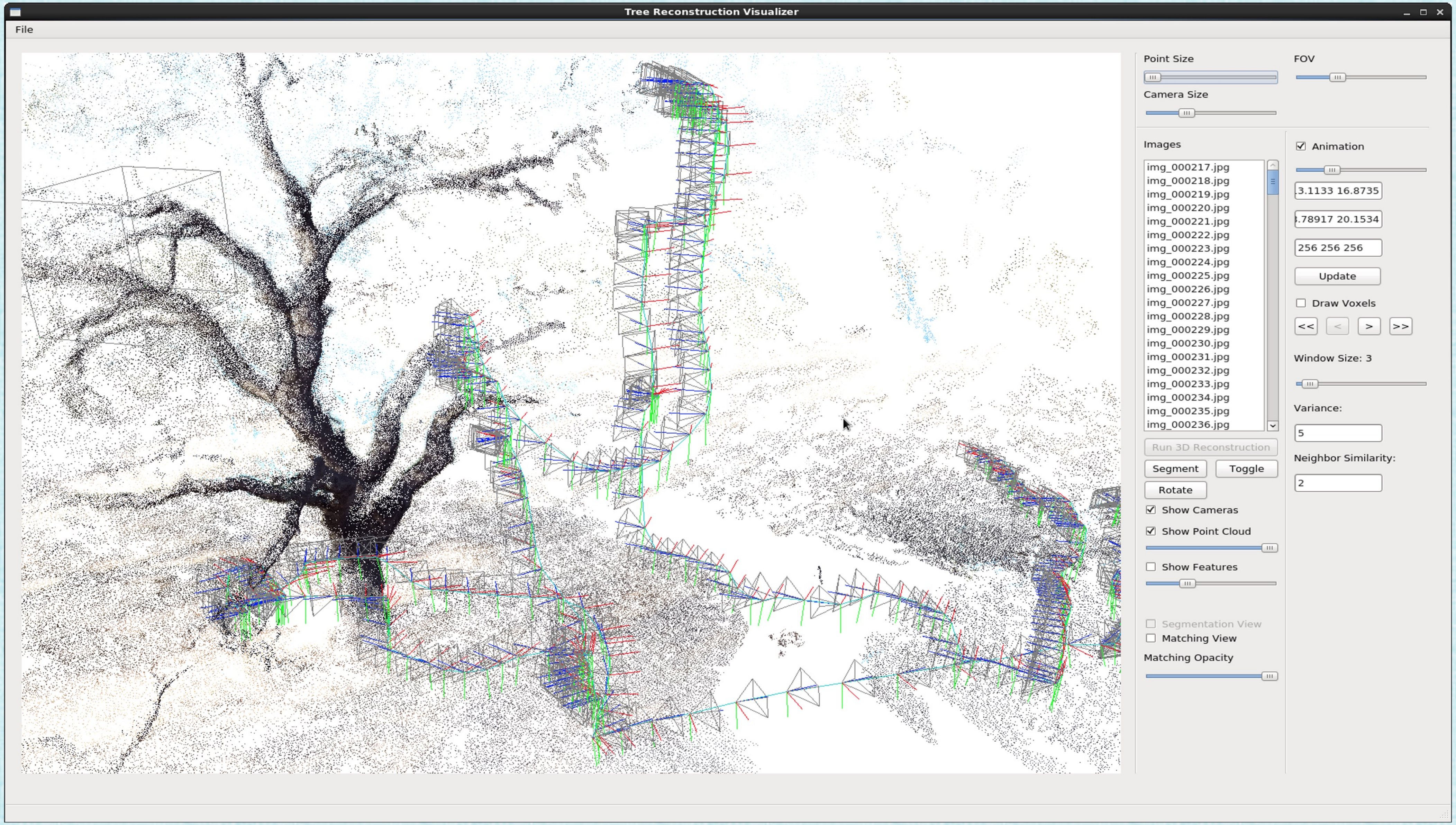
☑ Draw Vertex Projections

☐ Draw 2D Reprojections

☐ Use Annotated Textures

☐ Draw Ground Truth

Drones & Trees



Tree Reconstruction Visualizer

File

Point Size

FOV

Camera Size

Images

img_000217.jpg
img_000218.jpg
img_000219.jpg
img_000220.jpg
img_000221.jpg
img_000222.jpg
img_000223.jpg
img_000224.jpg
img_000225.jpg
img_000226.jpg
img_000227.jpg
img_000228.jpg
img_000229.jpg
img_000230.jpg
img_000231.jpg
img_000232.jpg
img_000233.jpg
img_000234.jpg
img_000235.jpg
img_000236.jpg

Run 3D Reconstruction

Segment

Toggle

Rotate

[✓] Show Cameras

[✓] Show Point Cloud

[ ] Show Features

[ ] Segmentation View

[ ] Matching View

Matching Opacity

[✓] Animation

3.1133 16.8735

.78917 20.1534

256 256 256

Update

[ ] Draw Voxels

<< < > >>

Window Size: 3

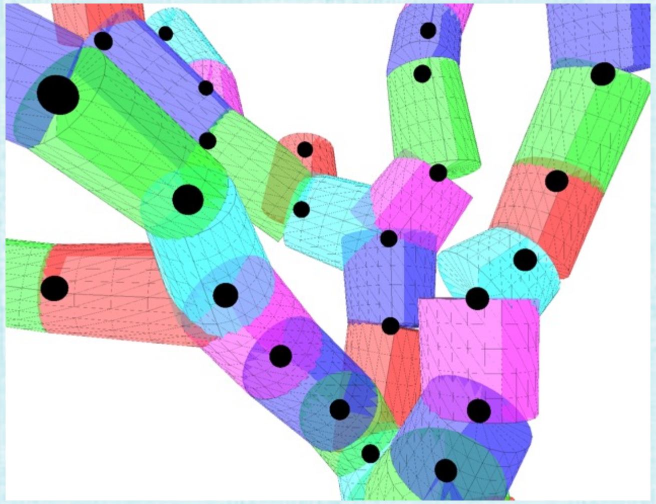
Variance:

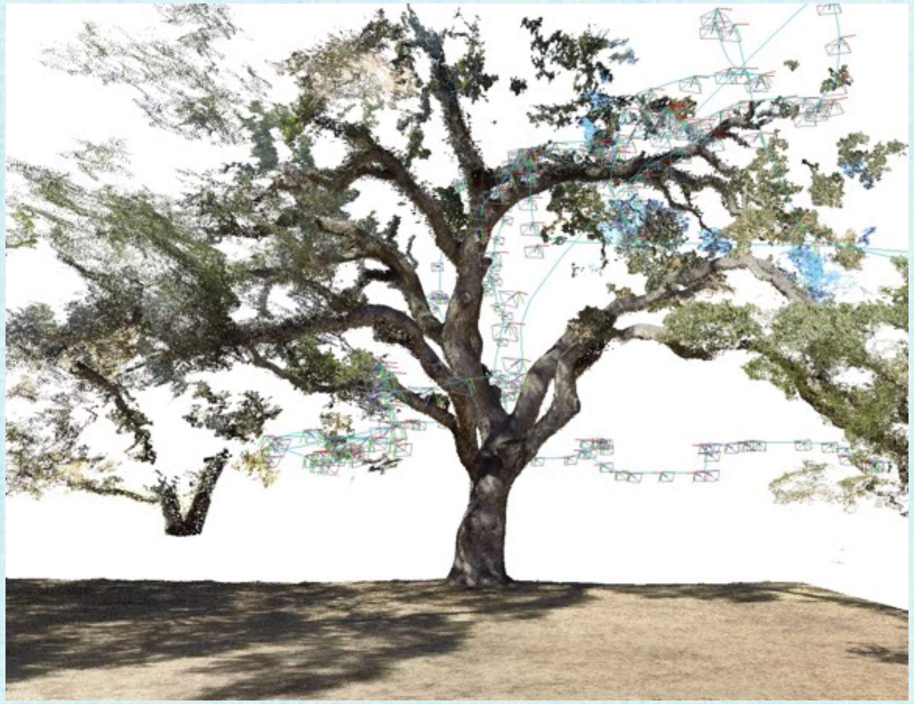
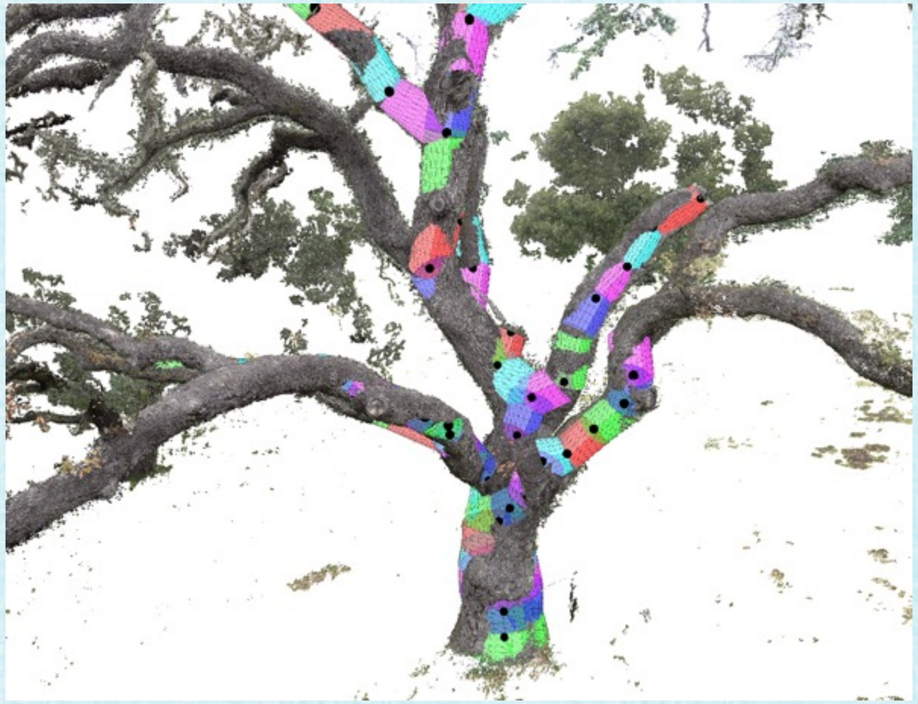
5

Neighbor Similarity:

2

Drones & Trees

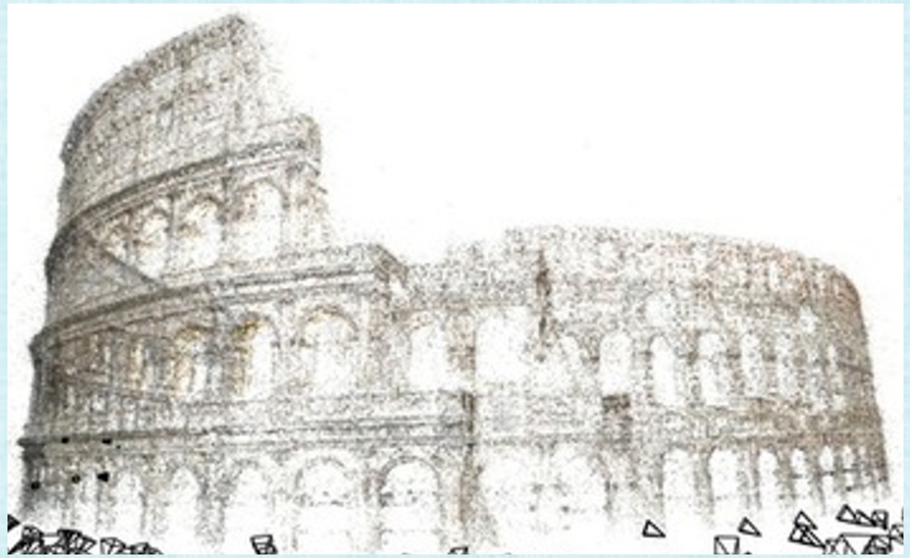


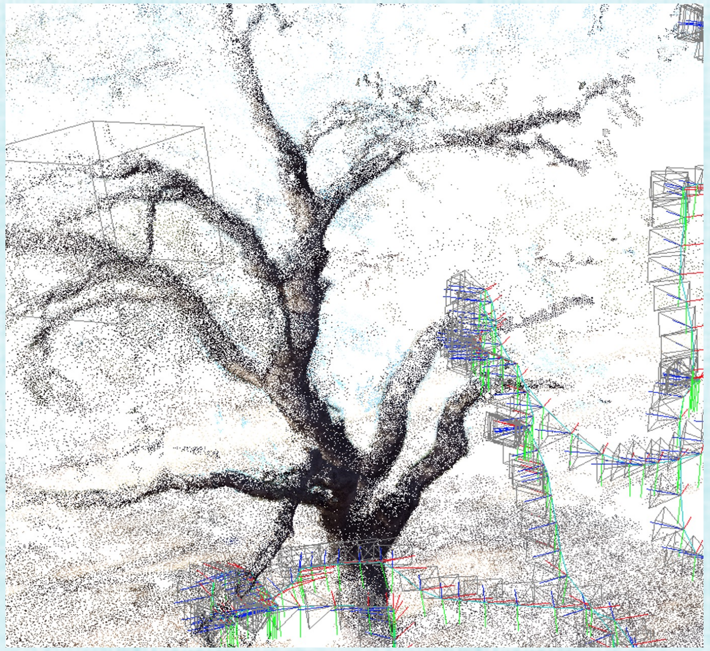



Drones & Trees

Noise & De-Noising

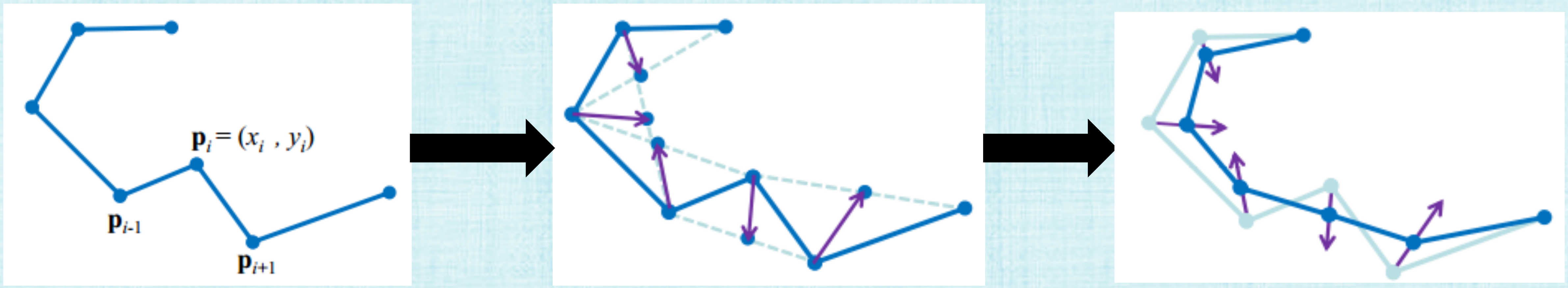
- Computer Vision algorithms use real-world sensors/data
- This results in noise corruption, which is the biggest drawback to such methods
- Denoising/smoothing algorithms are very important (for alleviating these issues)





Laplacian Smoothing

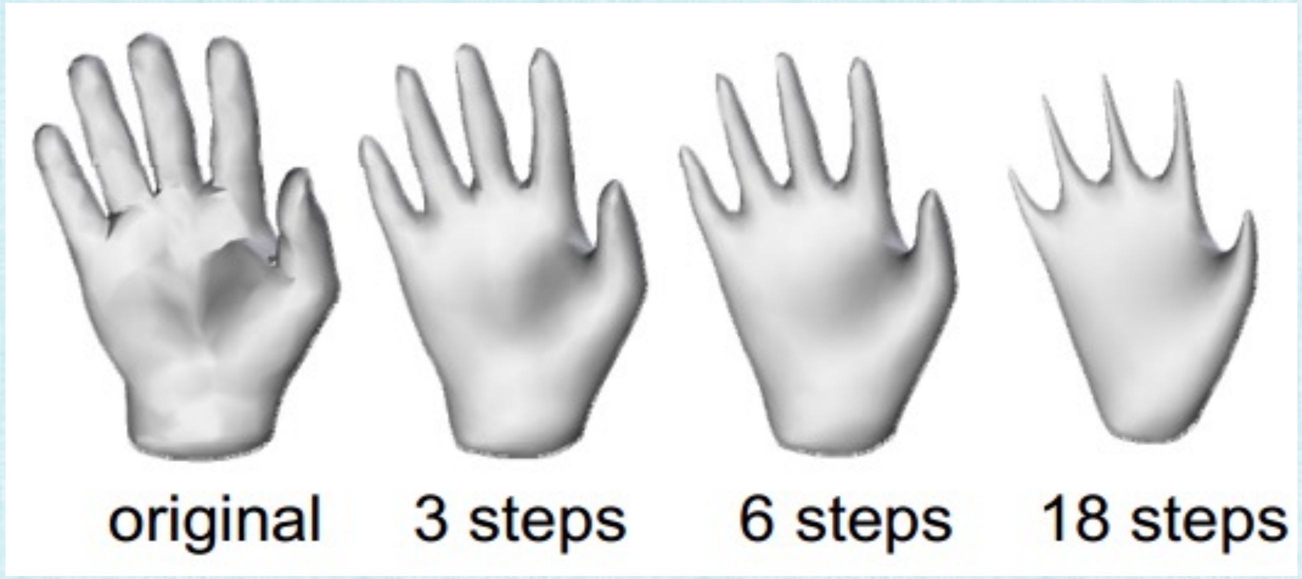
- Compute a Laplacian estimate using the one ring of vertices about a point
    - Similar to differential coordinates
- E.g., on a curve: $L(p_i) = \frac{1}{2}\left((p_{i+1} - p_i) + (p_{i-1} - p_i)\right)$
- Then, update $p_i^{new} = p_i + \lambda L(p_i)$ where $\lambda \in (0,1)$
- Repeat several iterations



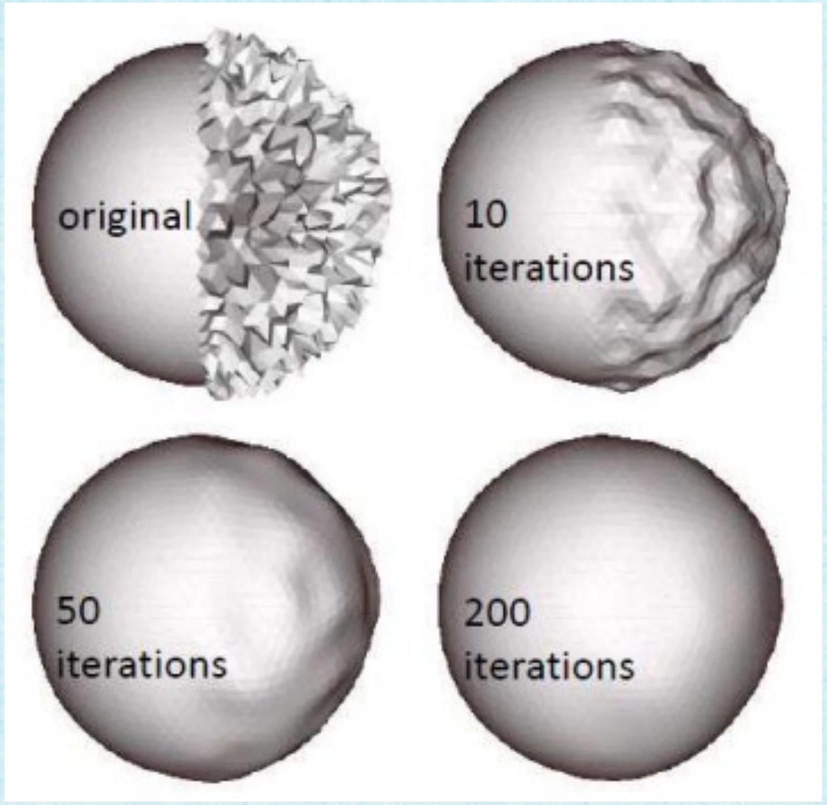
Taubin Smoothing

- Laplacian smoothing suffers from volume loss
- Taubin smoothing periodically performs an inflation step to add back volume:

$$p_i^{new} = p_i - \mu L(p_i) \quad \text{with} \quad \mu > 0$$



Laplacian smoothing (only)

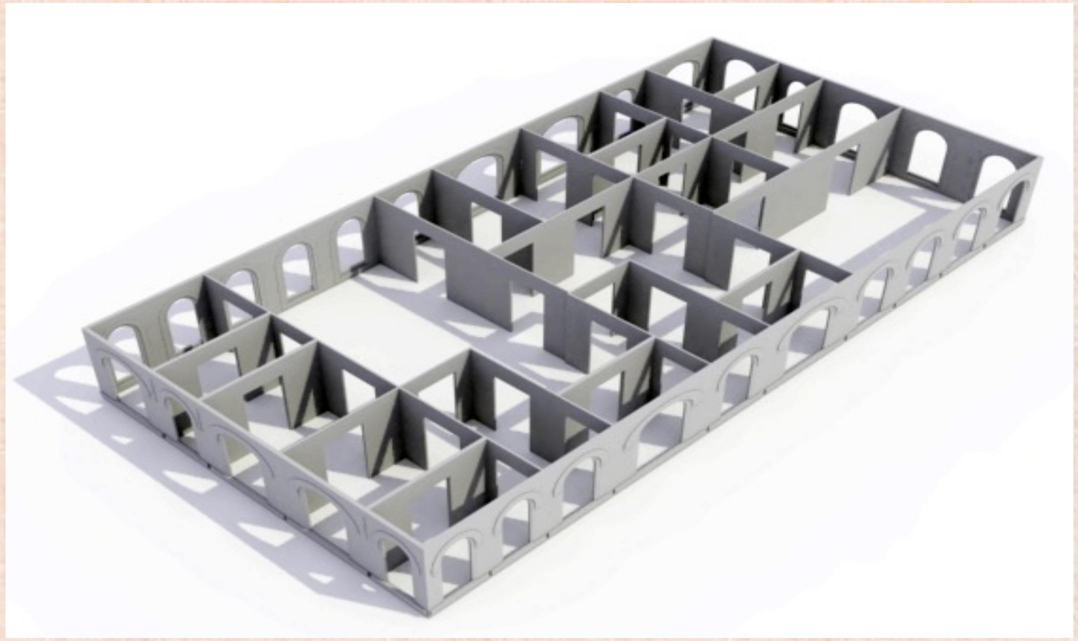


Taubin smoothing

Procedural Methods (for Geometry Construction)

- Generate geometry with an algorithm
  - Typically used for complex or tedious-to-create models
  - Perturb the algorithm to make variations of the geometry
- Start with a small set of data
- Use rules to describe high level properties of the desired geometry
- Add randomness, and use recursion

Neural Networks?
Generative AI?

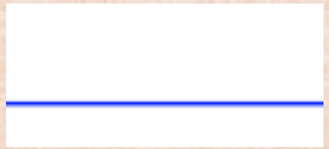


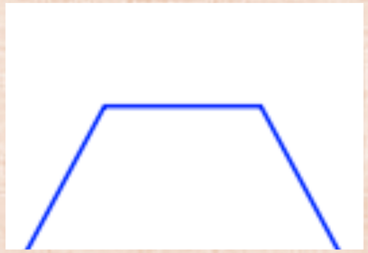
L-Systems

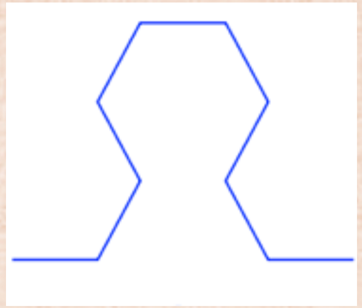
- Developed by a biologist (Lindenmayer) to study algae growth
- A recursive formal grammar:
  - An alphabet of symbols (terminal and non-terminal)
  - Production rules: non-terminal symbols recursively create new symbols (or sequences of symbols)
- Start with an initial string (axiom), and apply production rules
- A translator turns symbols into geometric structures

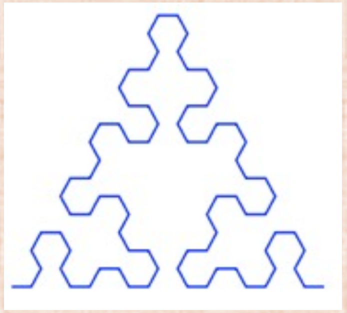
Natural Language Processing (NLP)?

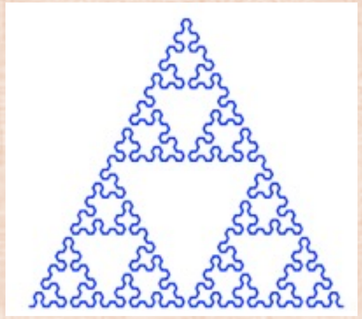
Nonterminals: A and B both mean “draw forward”
Terminals: +/- mean "turn” right/left (respectively) by 60 degrees
Production Rules: A → B + A + B and B → A – B – A

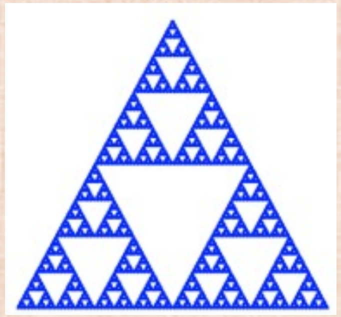

Intial Axiom: A


B+A+B


A-B-A + B+A+B + A-B-A

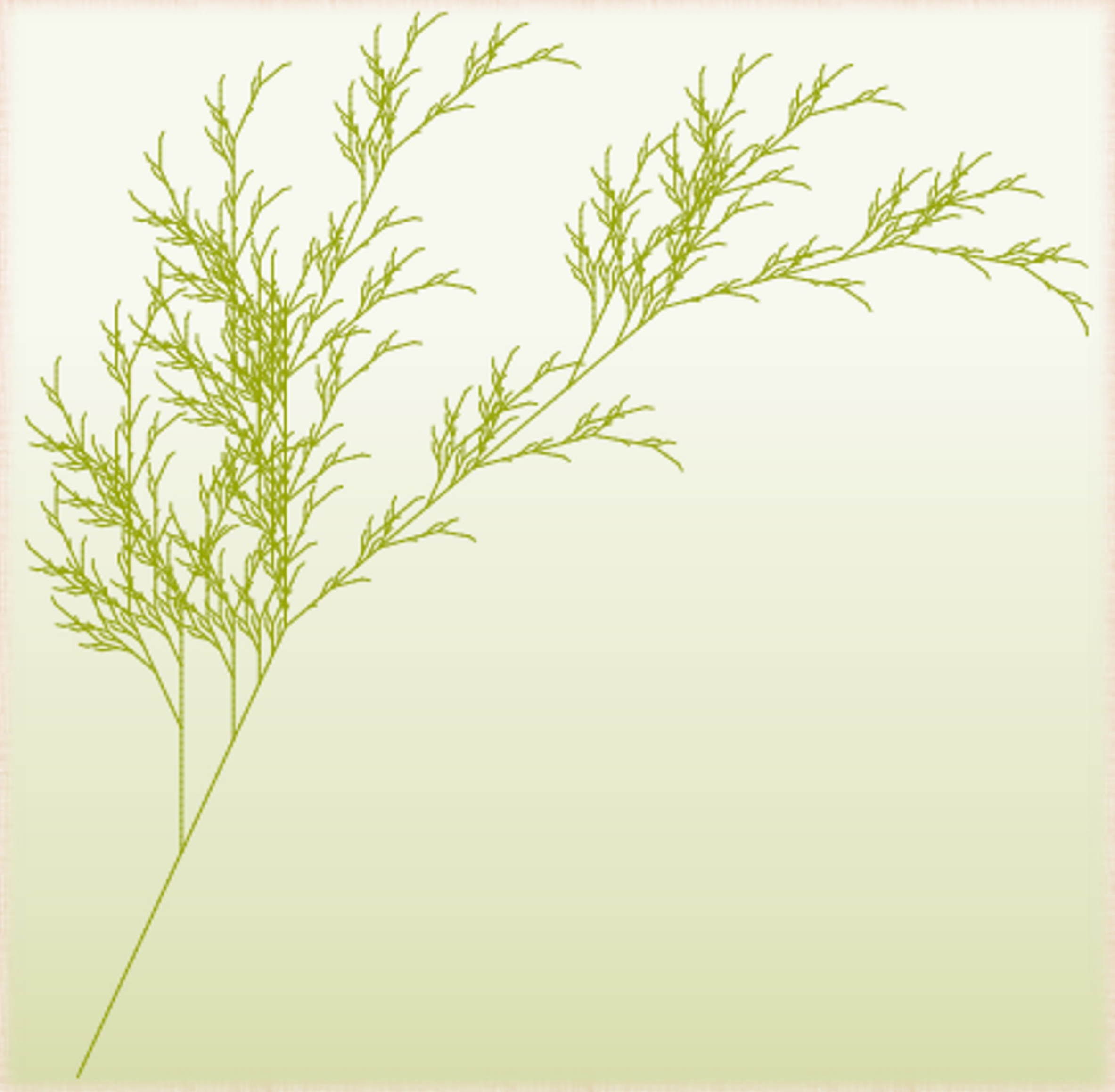

B+A+B-(A-B-A)-(B+A+B)
+A-B-A +B+A+B +A-B-A
+B+A+B-(A-B-A)-(B+A+B)


Etc.


Sierpinski Triangle

L-System + Stack = Branches

- Nonterminals: X is no action, F is draw forward
- Terminals: +/- means turn right/left by 25 degrees
  - **[means to store current state on the stack**
  - **] means to load the state from the stack**
- Initial Axiom: X
- Production Rules: X → F − [[X]+X] + F [+FX] − X, F → FF

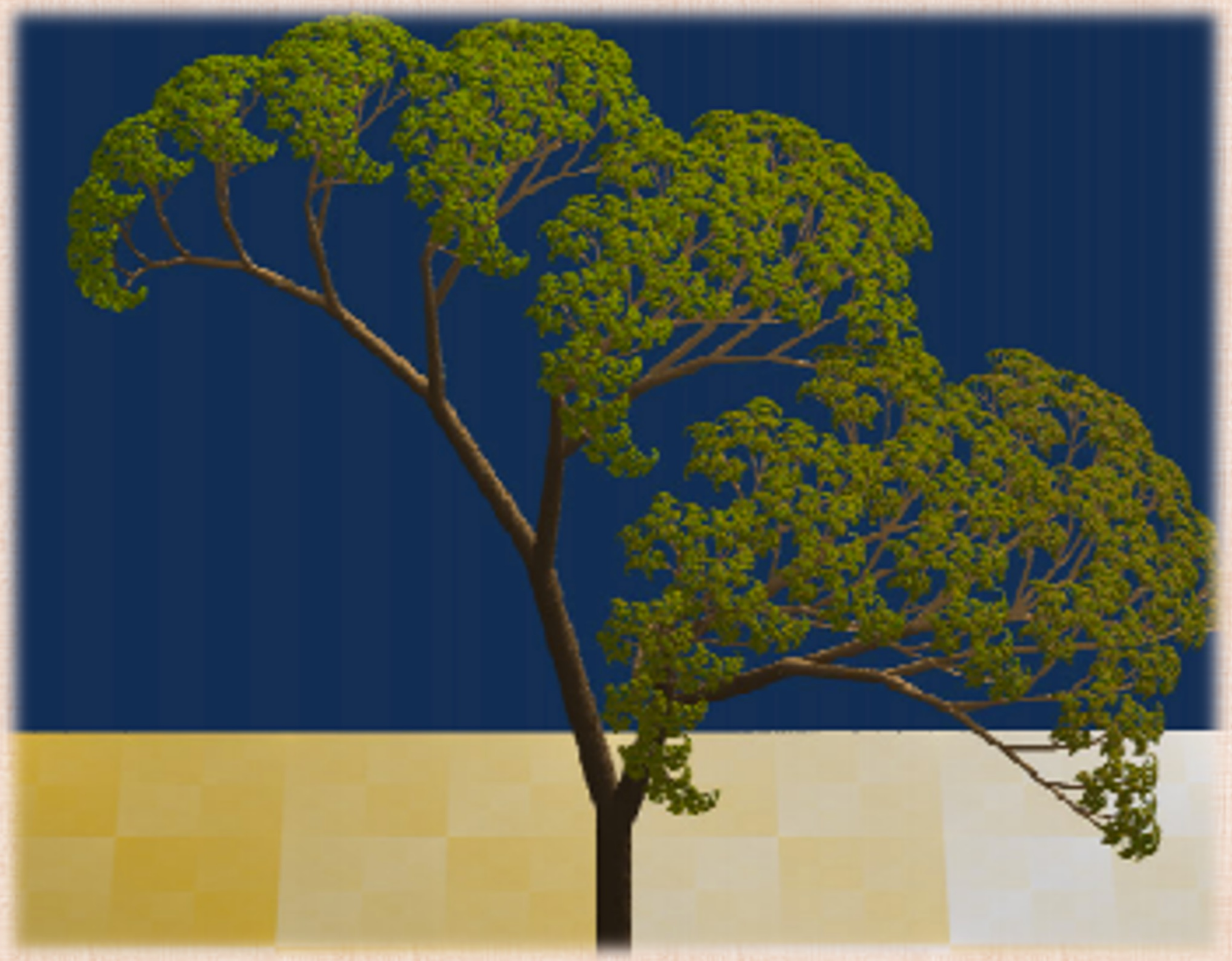


L-Systems

- Easily extended to 3D
- Model trunk/branches as cylinders
- As recursion proceeds:
  - shrink cylinder size
  - vary color (from brown to green)

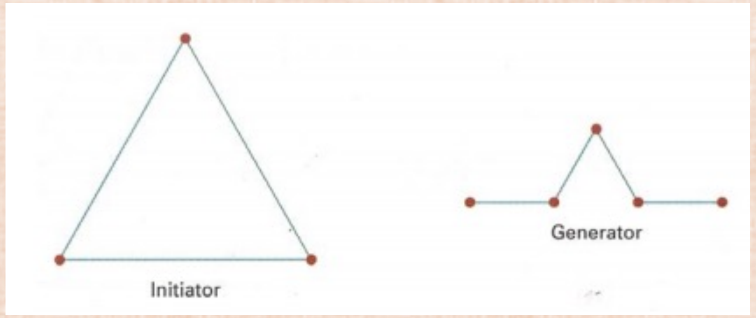
- Add more variety with a stochastic L-system
  - Multiple (randomly-chosen) rules for each symbol

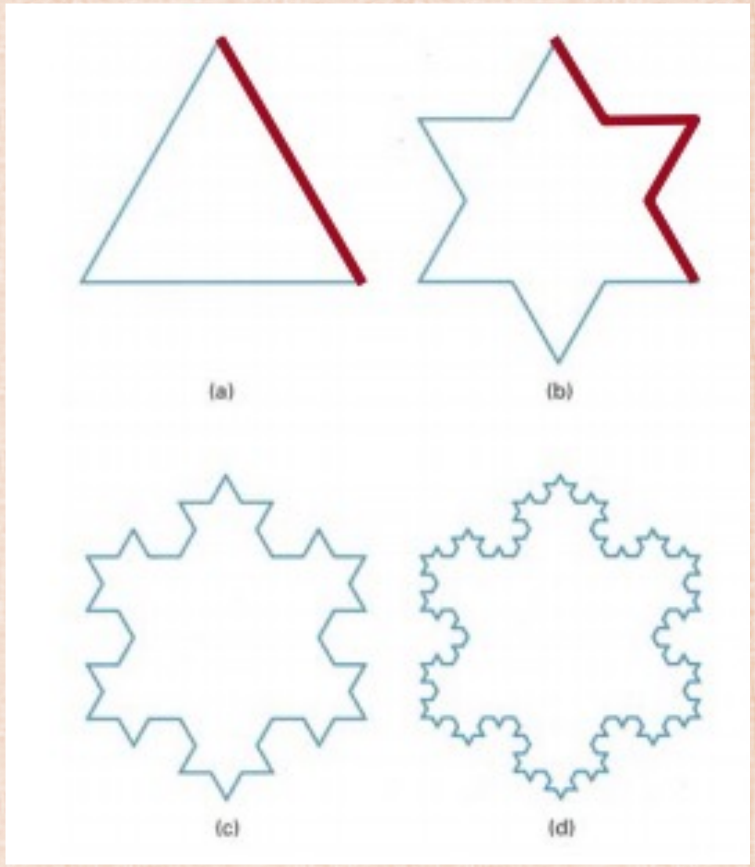
- Practice and experimentation is required in order to obtain good results
  - (just like for ML)

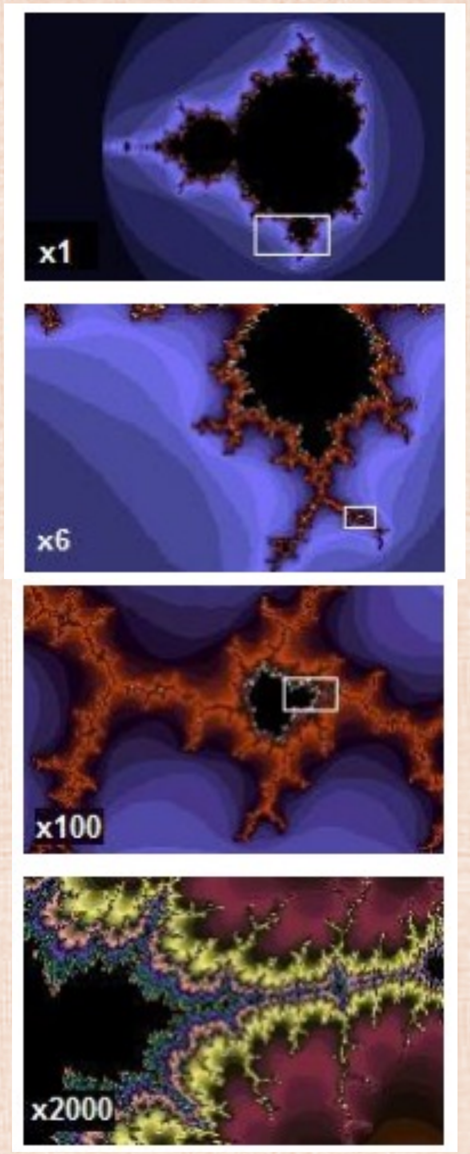


Fractals

- Initiator: start with a shape
- Generator: replace subparts with a (scaled) generator
- Repeat

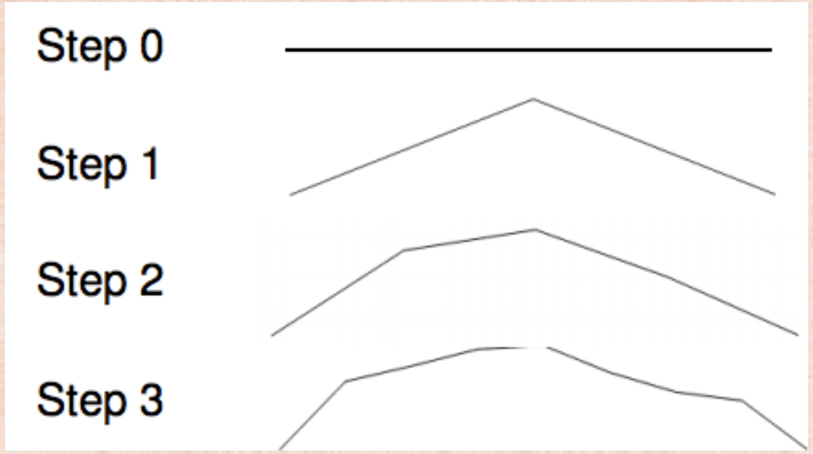






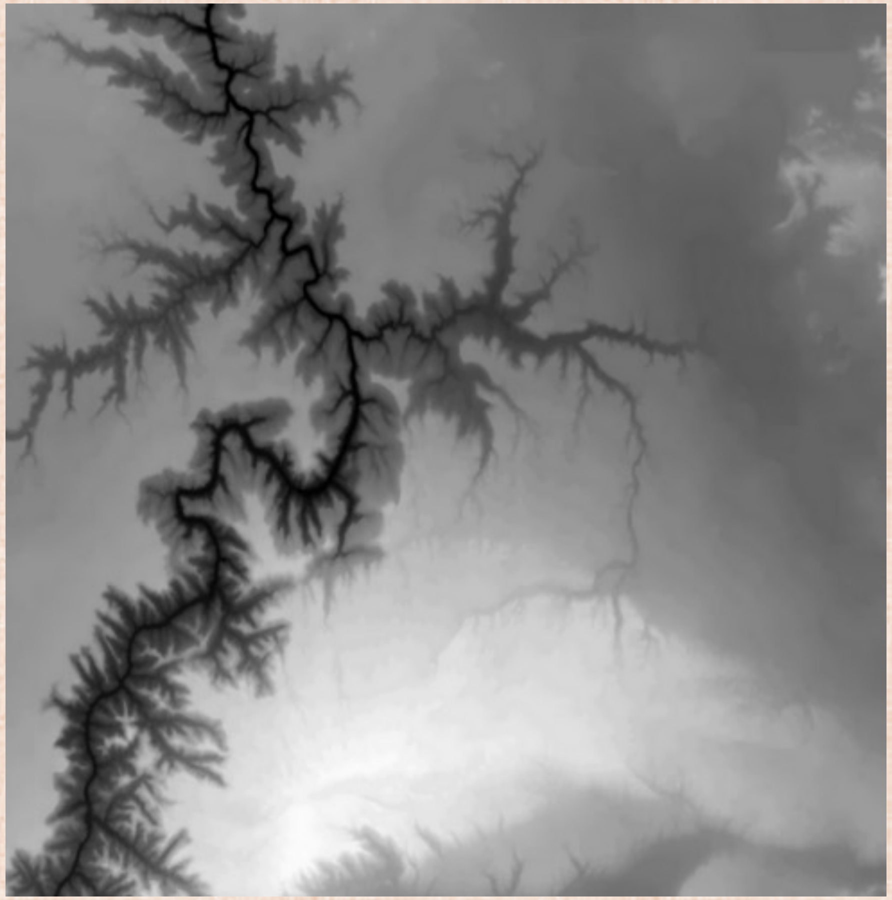
Fractals

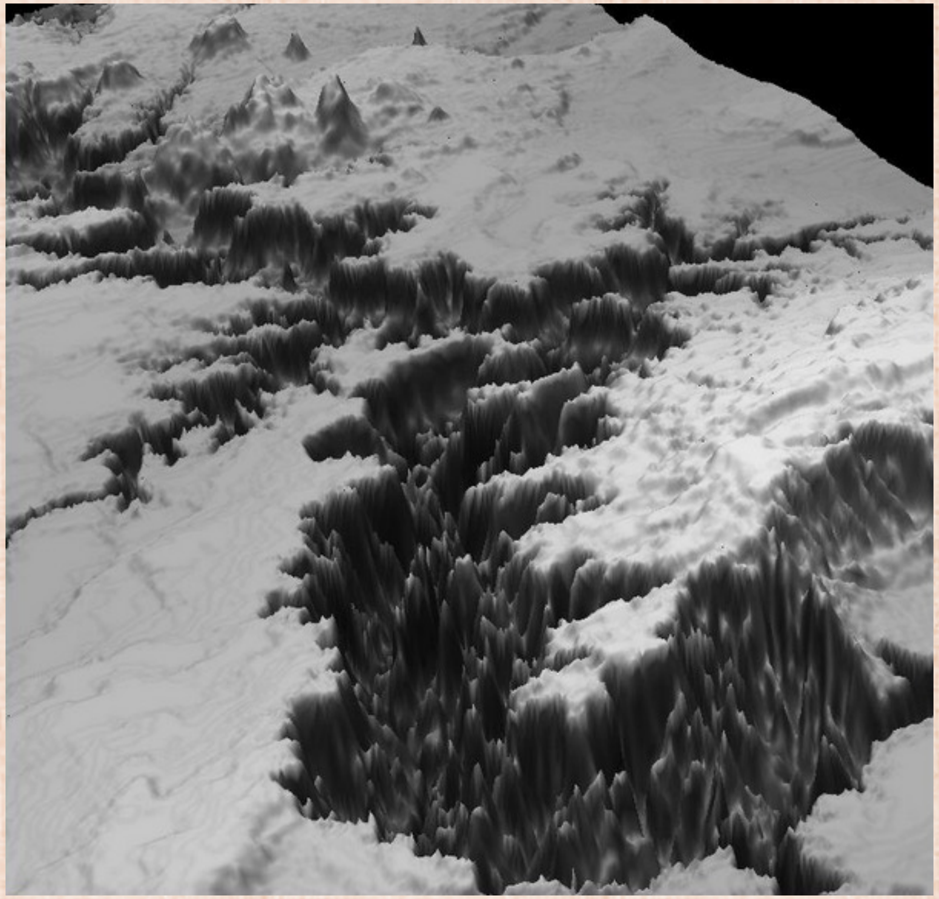
- Add randomness to the new vertex locations
- E.g., can create an irregular 2D silhouette (for far away mountains)



Height Fields

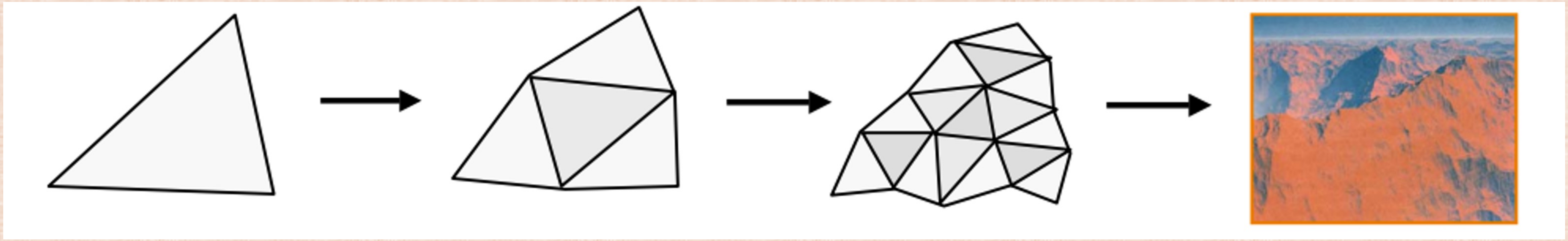
- Start with a 2D fractal (or any 2D grey-scale image)
- Place the image on top of a ground plane (subdivided into triangles)
- For each triangle vertex, displace its height based on the local pixel intensity





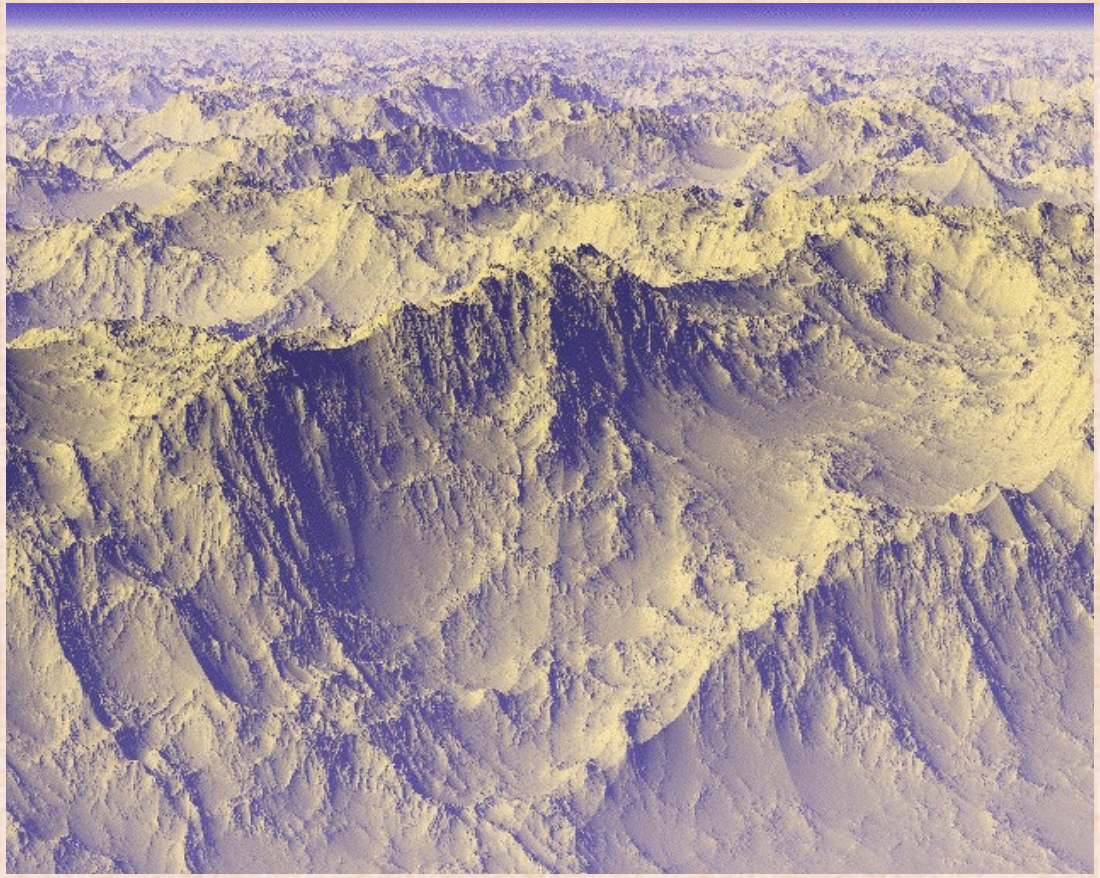
3D Landscapes

- Initiator: start with a shape
- Generator: replace random subparts with a self-similar (somewhat random) pattern

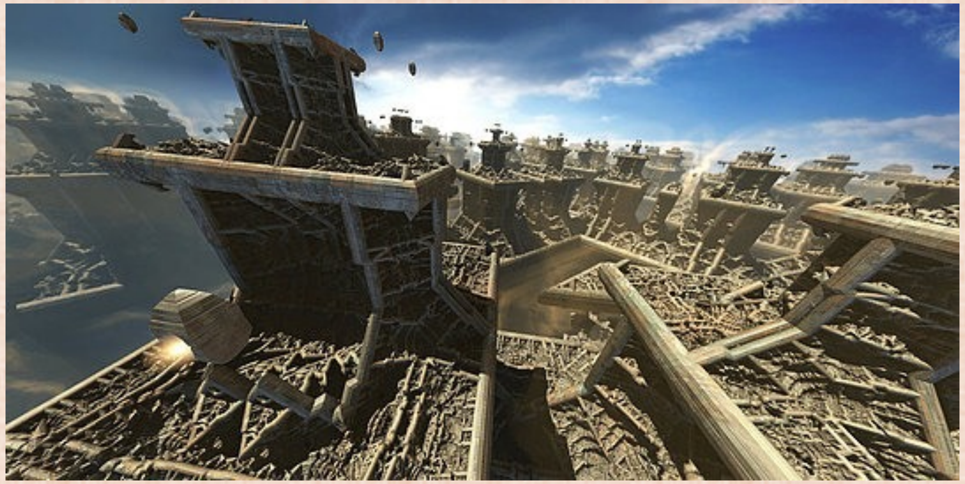
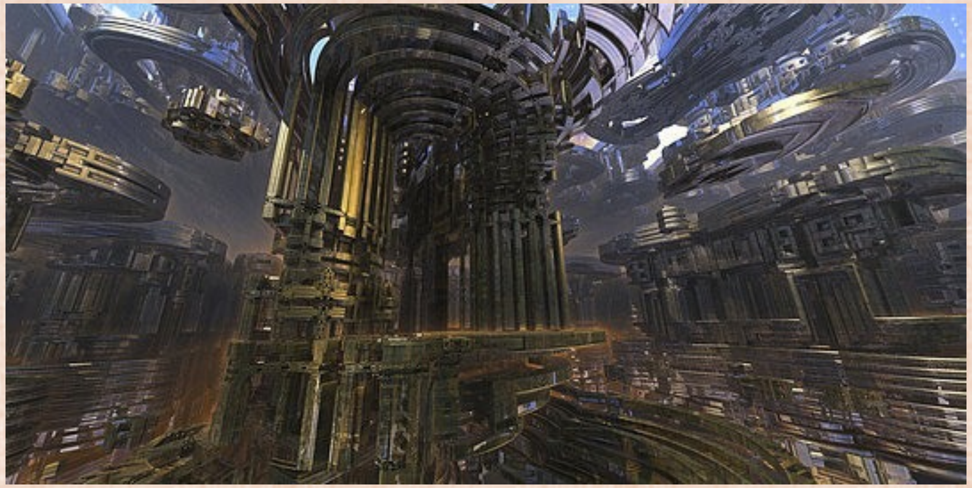
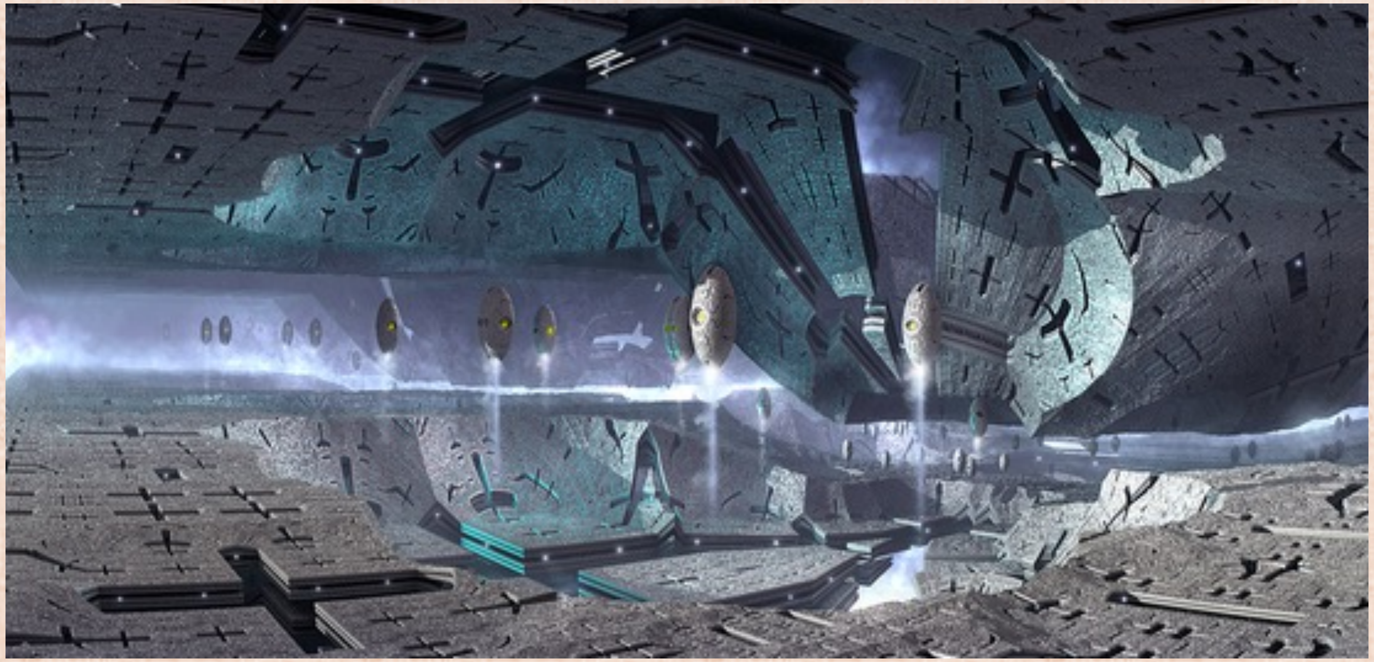


- Similar to subdivision, but with much more interesting rules for setting vertex positions

3D Landscapes

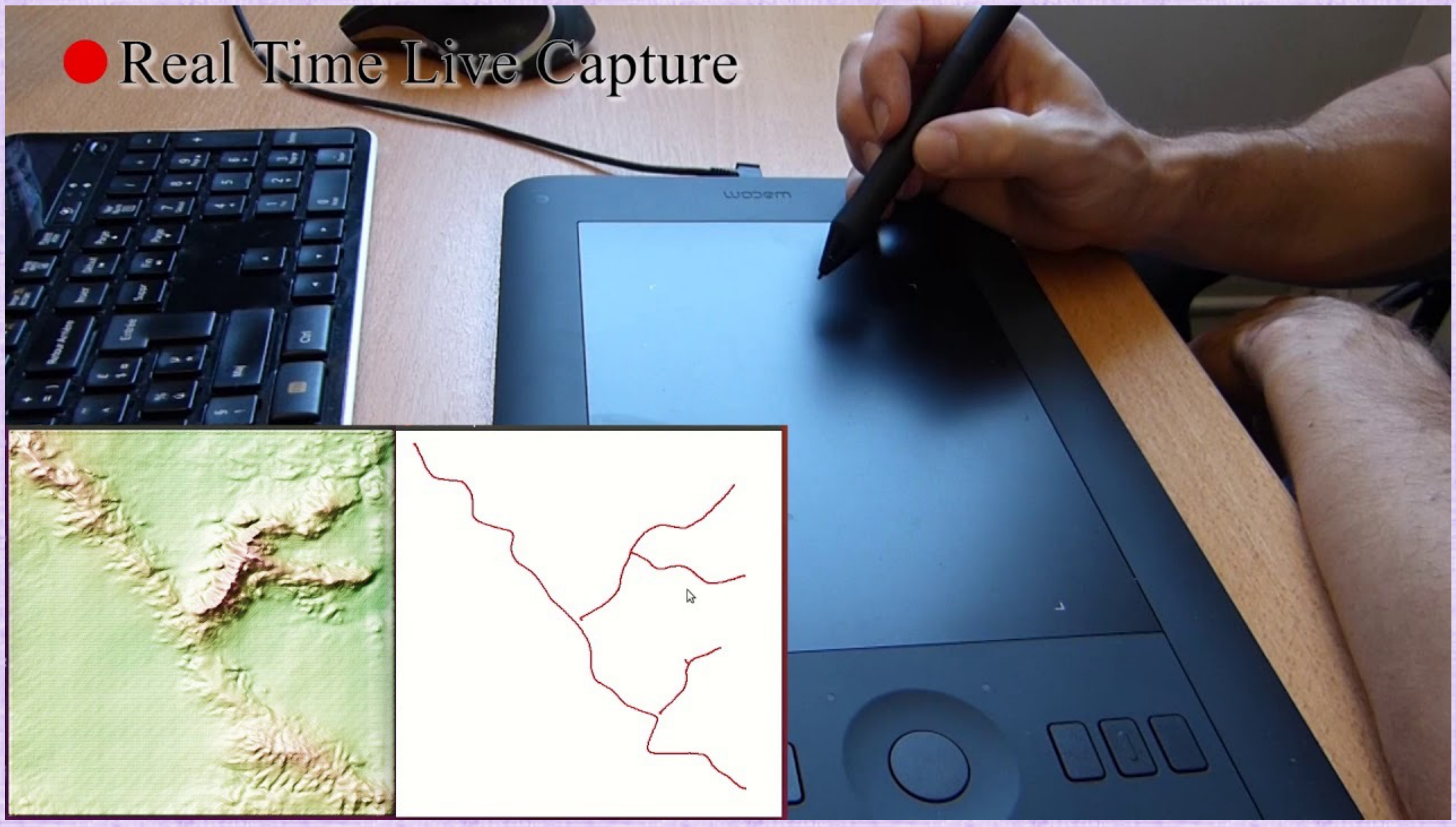


Fractal Worlds

Machine Learning

- Interactive Example-Based Terrain Authoring with Generative Adversarial Networks
  - Siggraph 2017



Generative AI

Rapidly Generate 3D Assets for Virtual Worlds with Generative AI

Jan 03, 2023

+21 Like Discuss (0)

By Gavriel State