

# L20: Mesh Processing

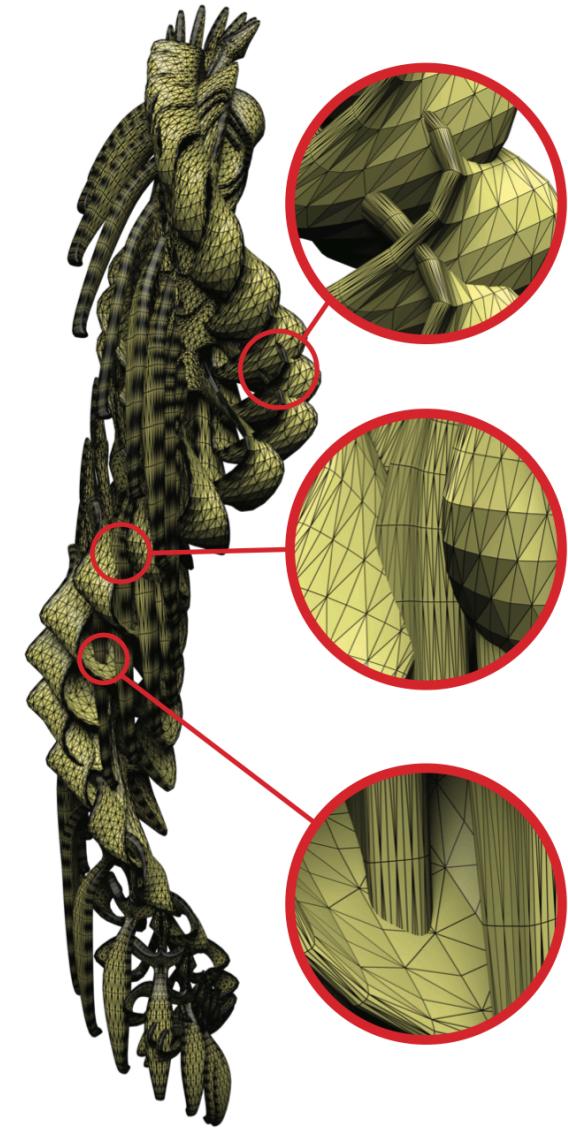
Hao Su

Ack: Minghua Liu helped to prepare slides

# A Peek into Various Mesh Processing Algorithms

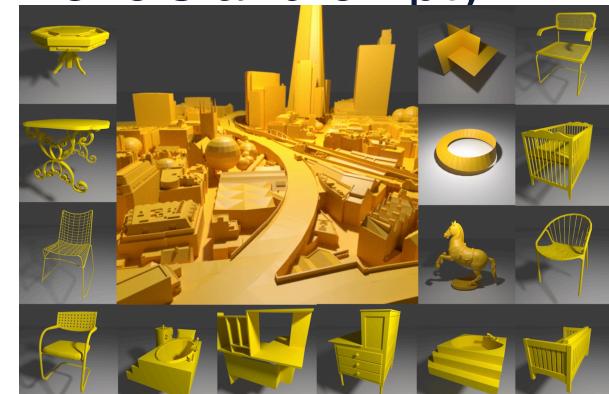
# Watertight Manifold Surface Generation

- Many downstream tasks require the input mesh to be a watertight manifold.
- However, meshes designed by artists or reconstructed by some algorithms are often non-manifold:
  - open boundaries
  - self-intersections
  - incorrect connectivity
  - ambiguous face orientation
  - double surfaces



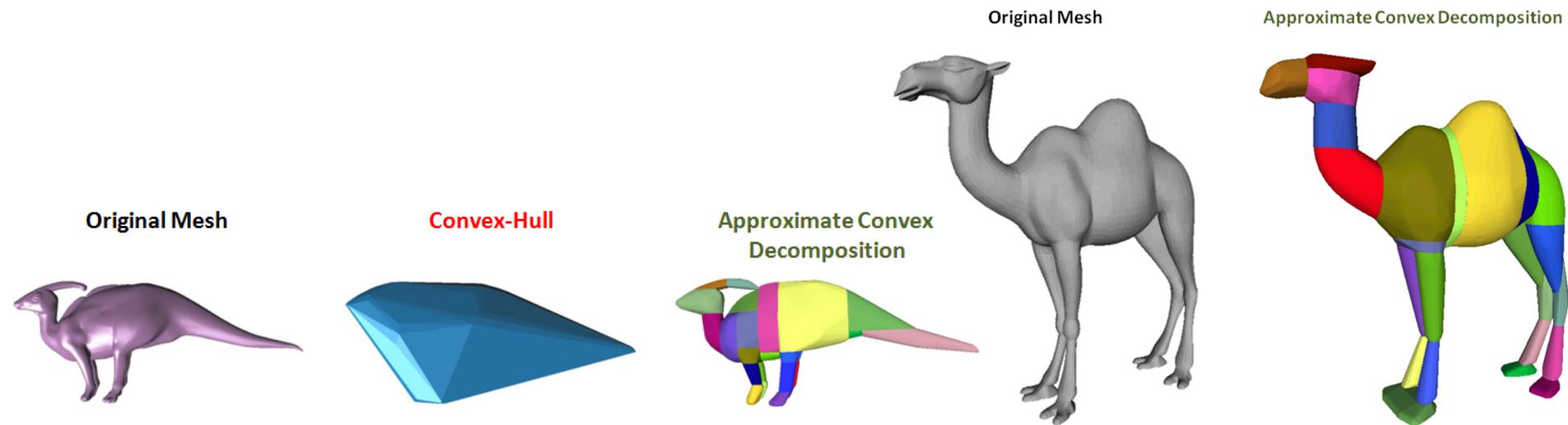
# Watertight Manifold Surface Generation

- Huang J, Su H, Guibas L. “**Robust watertight manifold surface generation method for ShapeNet models.**”
  - Small artifacts, all surfaces have volume (<https://github.com/hjwdzh/Manifold>)
- Huang J, Zhou Y, Guibas L. “**ManifoldPlus: A Robust and Scalable Watertight Manifold Surface Generation Method for Triangle Soups.**” (<https://github.com/hjwdzh/ManifoldPlus>)
  - Less artifacts, may contain zero-volume structures
- Basic steps:
  - Voxelize the surface
  - Extracts exterior faces between occupied voxels and empty voxels
  - Projection-based optimization



# Convex Decomposition

- In some applications (e.g., collision detection), convex shapes are an input requirement.
- Directly using convex-hull provides poor approximation for concave surfaces.
- Exact convex decomposition is NP-hard and may produce a high number of primitives.



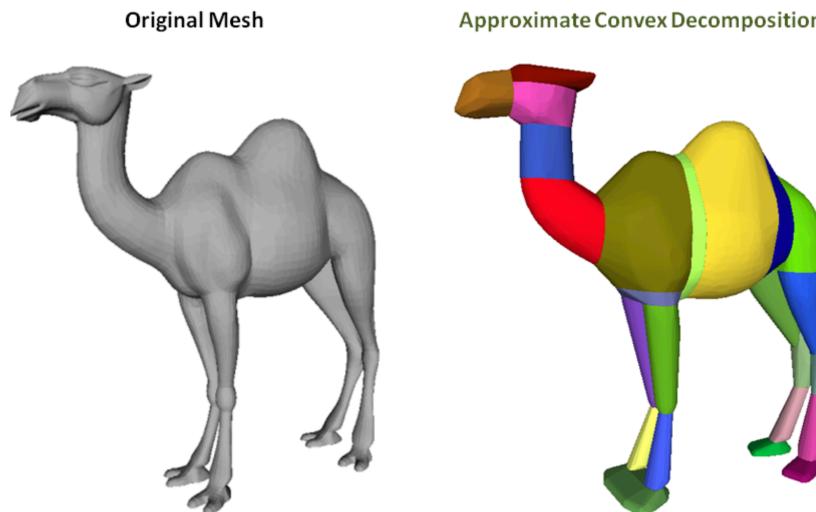
# Approximate Convex Decomposition

- Goal: partition the mesh into a minimal number of clusters while ensuring that each cluster has a concavity metric lower than a pre-defined threshold

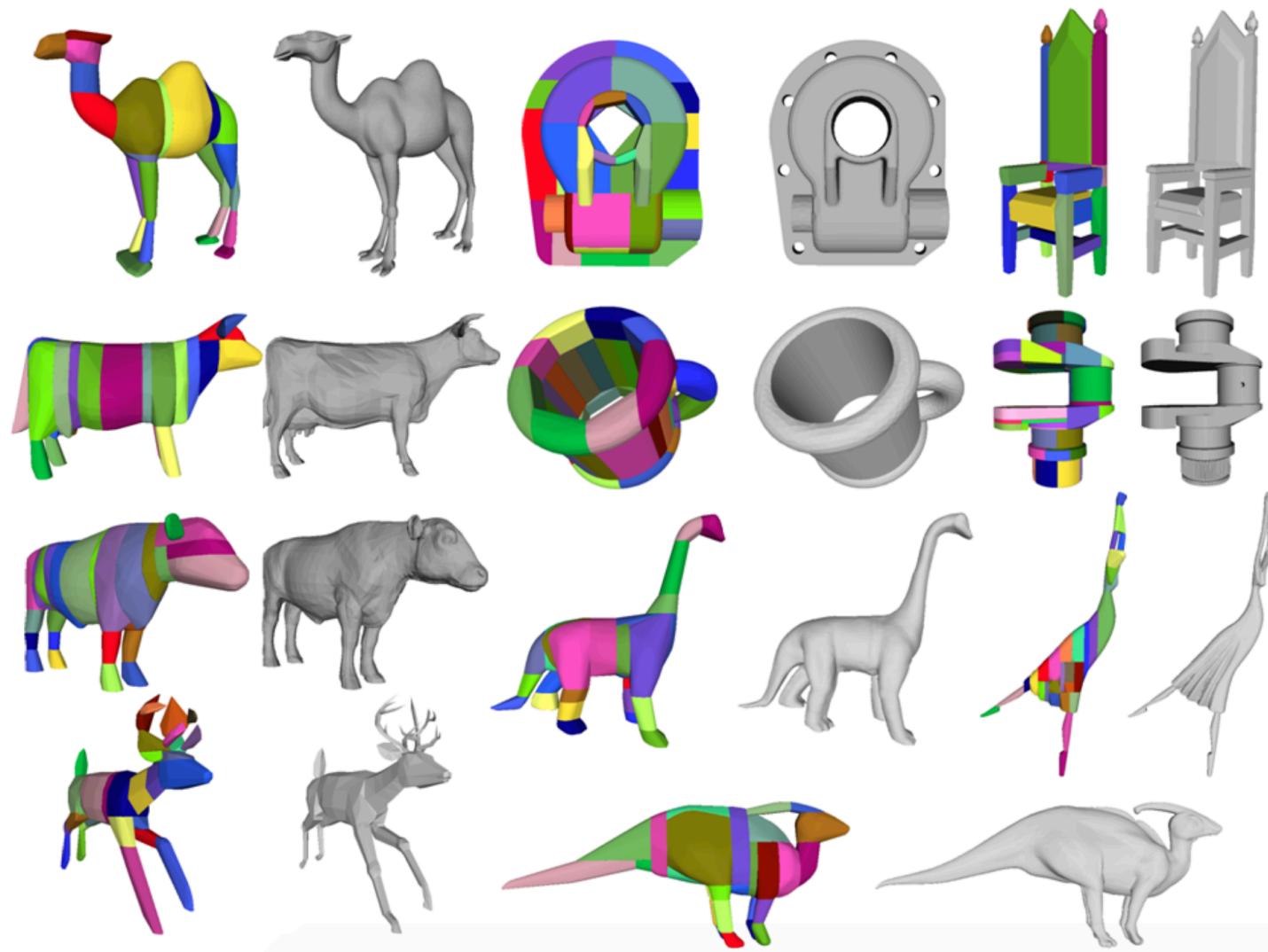
# V-HACD

- First, voxelize the shape
- Then, use a heuristic algorithm (e.g., cut by axis-aligned plane) to segment the shape into parts
- The segmentation optimizes a volume-based concavity measure:

$$\frac{V(ConvexHull(Sub_1)) + V(ConvexHull(Sub_2)) - V(original)}{V(original)}$$

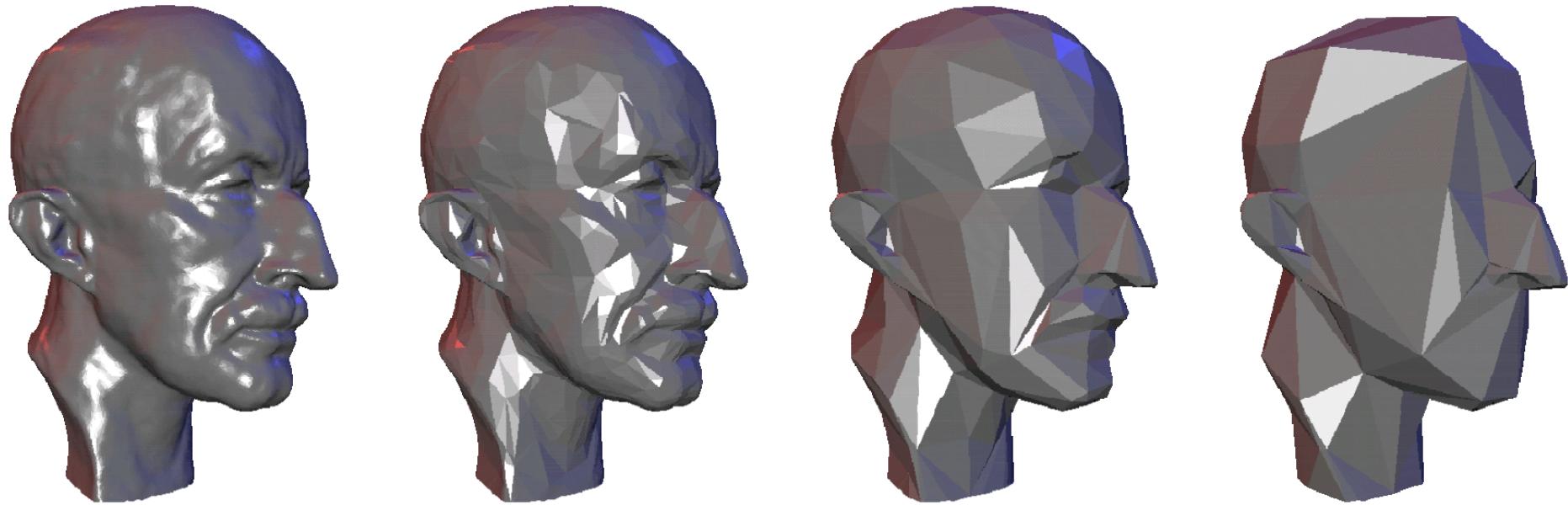


# V-HACD Results

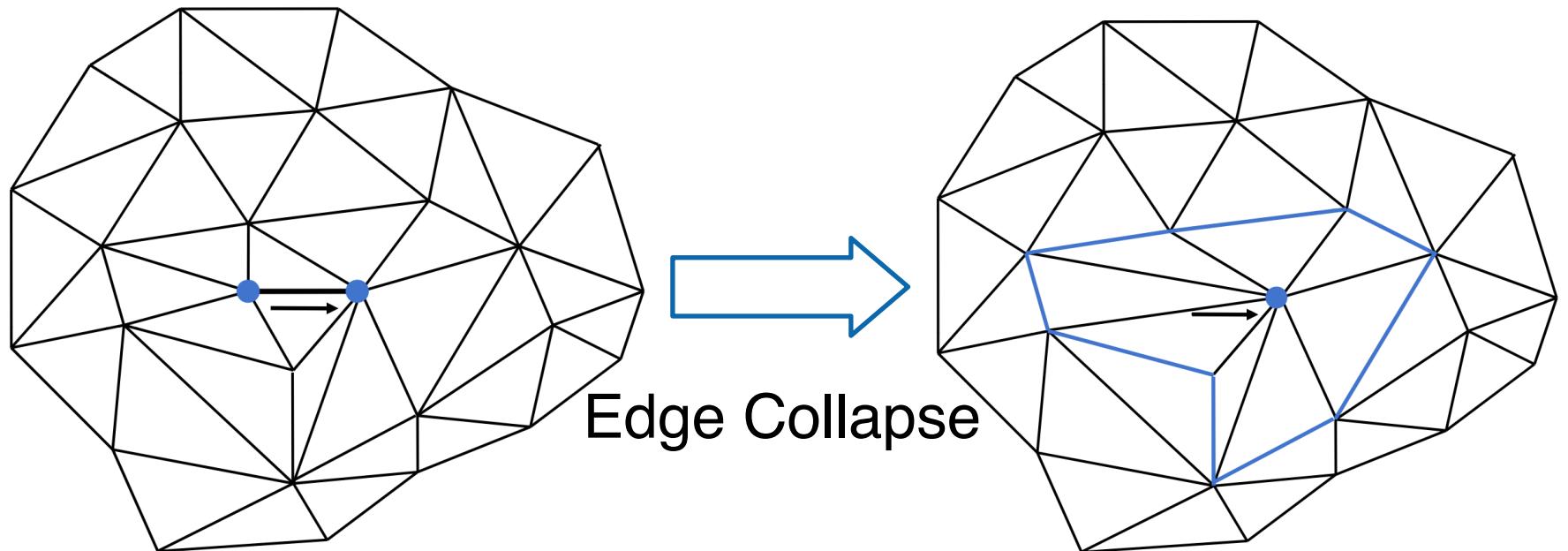


# Mesh Simplification (Downsampling)

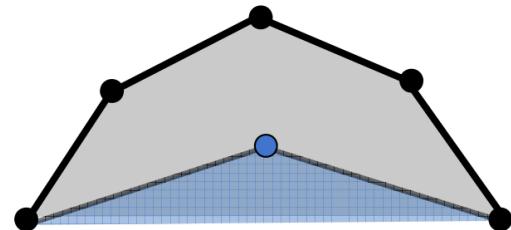
- Represent a mesh with fewer vertices
- Coarse mesh: memory efficient
- Fine-grained mesh: smooth, details



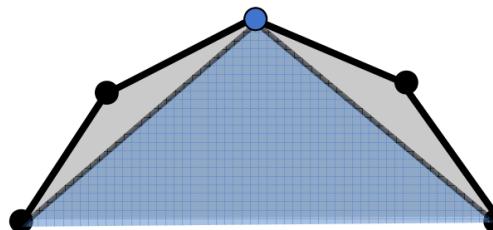
# Basic Procedure



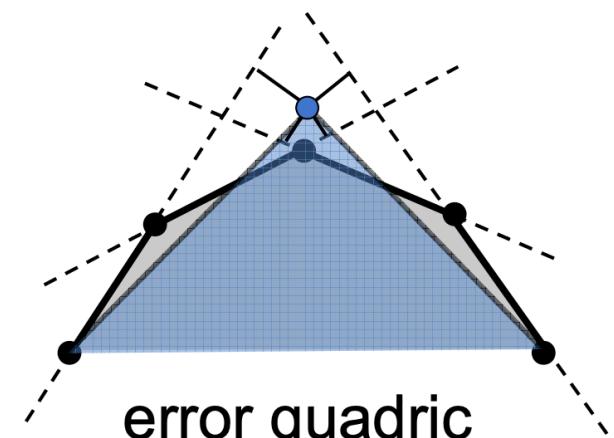
# Adjustment of New Node Position



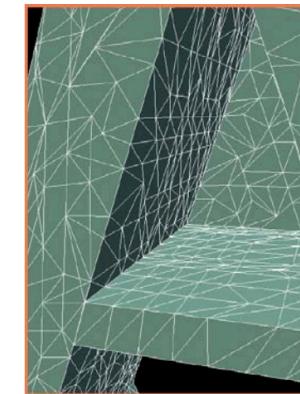
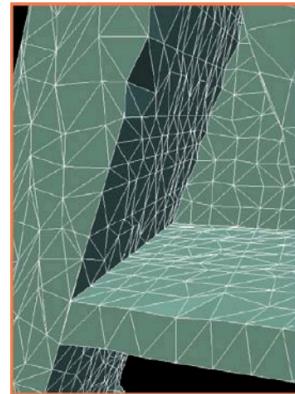
average



median



error quadric

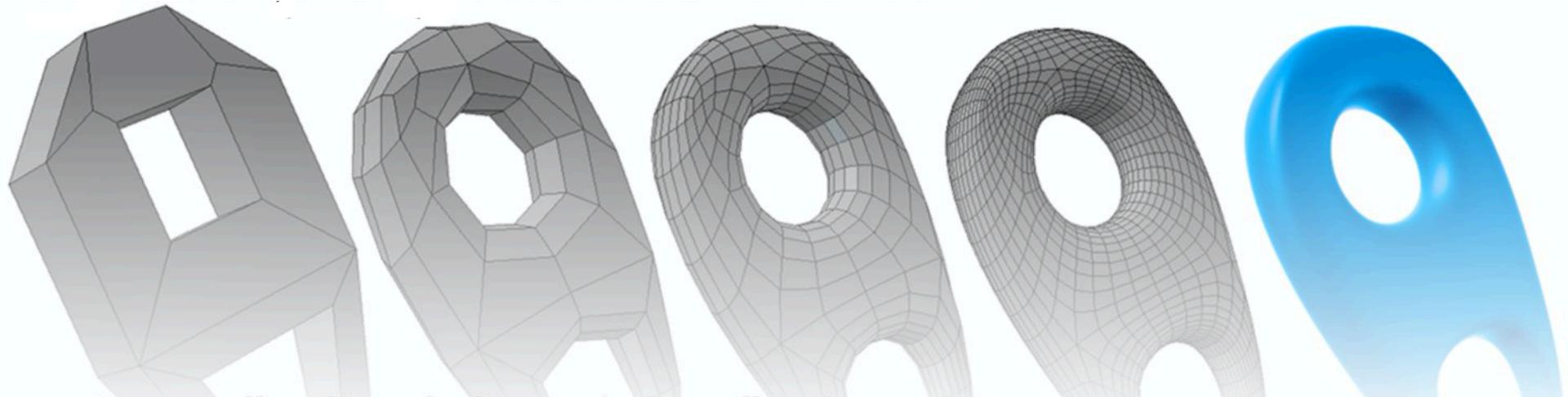
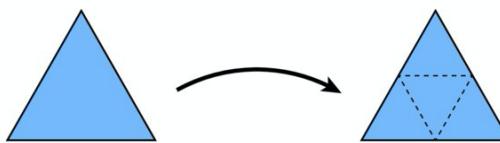


$$\min_p \sum_i \text{dist}(q_i, p)^2$$

$q_i$ : planes in the cell

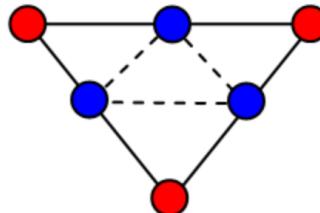
# Mesh Upsampling via Subdivision

- Recursive upsampling:
  - Divide mesh element (one triangle becomes four)
  - Calculate new positions of the mesh vertices

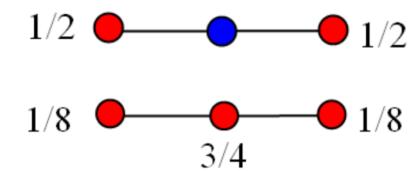
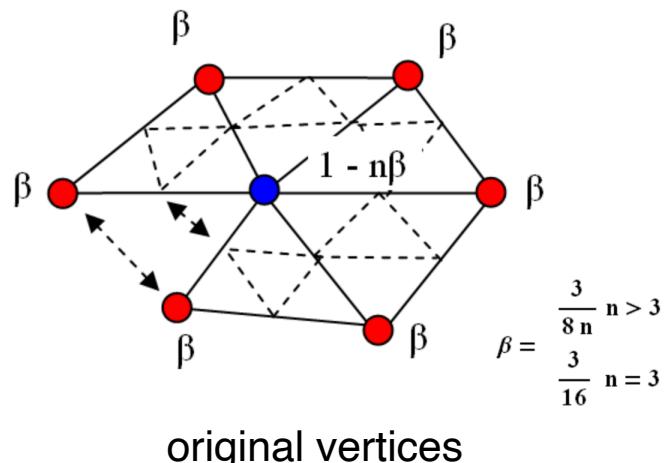
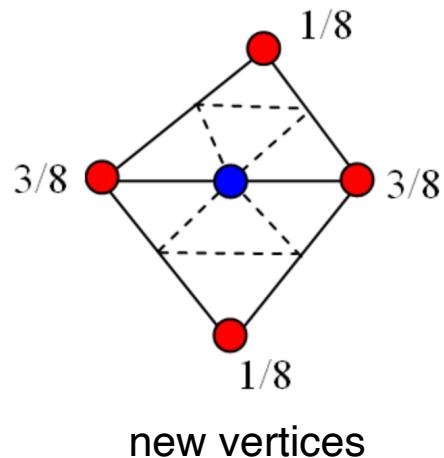


# Loop Subdivision

- Divide mesh element (one triangle becomes four)

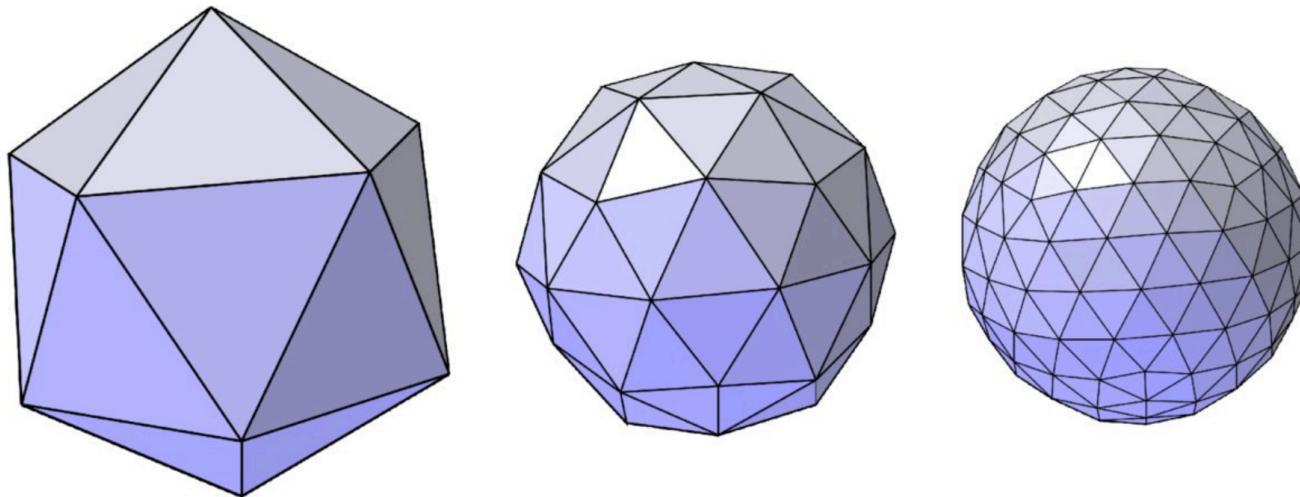


- Calculate new positions of the mesh vertices (linear combinations of original positions)



# Loop Subdivision

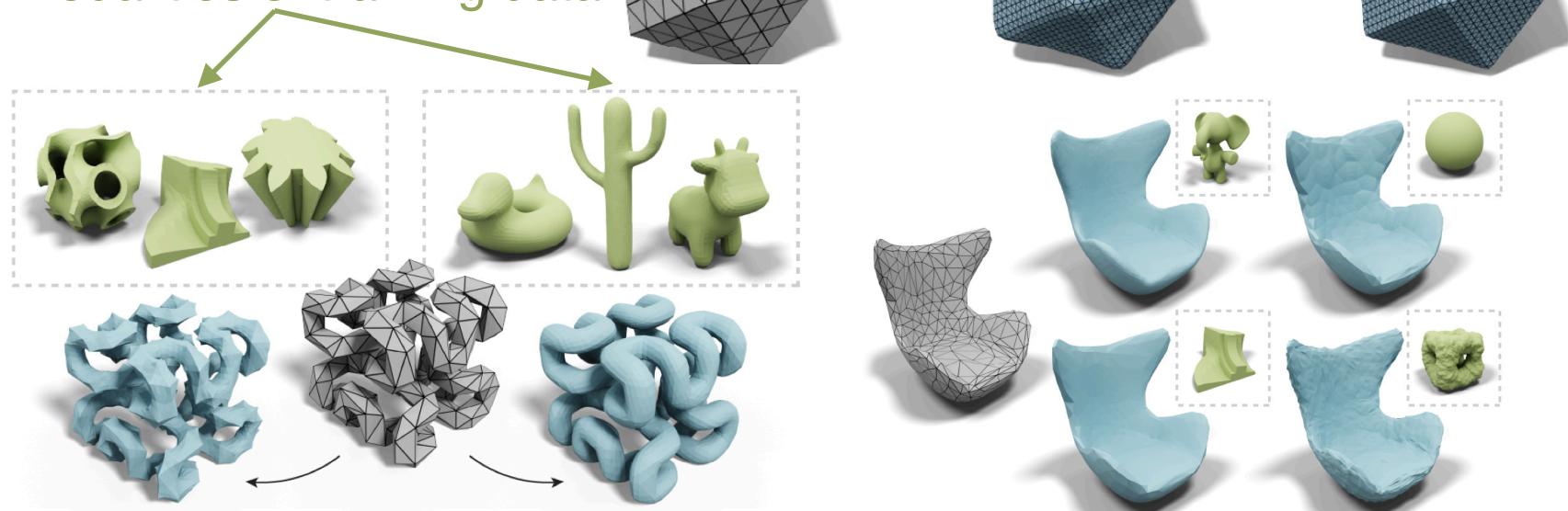
- Common subdivision rule
- “C2” smoothness of its limit shape
- Approximating, not interpolating



# Neural Subdivision

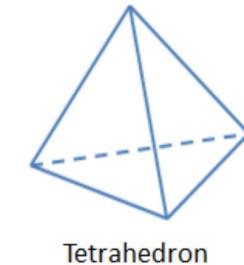
- Using network to predict vertex positions conditioned on the local geometry enables us to learn complex non-linear subdivision scheme (more than smoothing).
- Data-driven: training on different shapes can learn different styles

Two modalities of training data

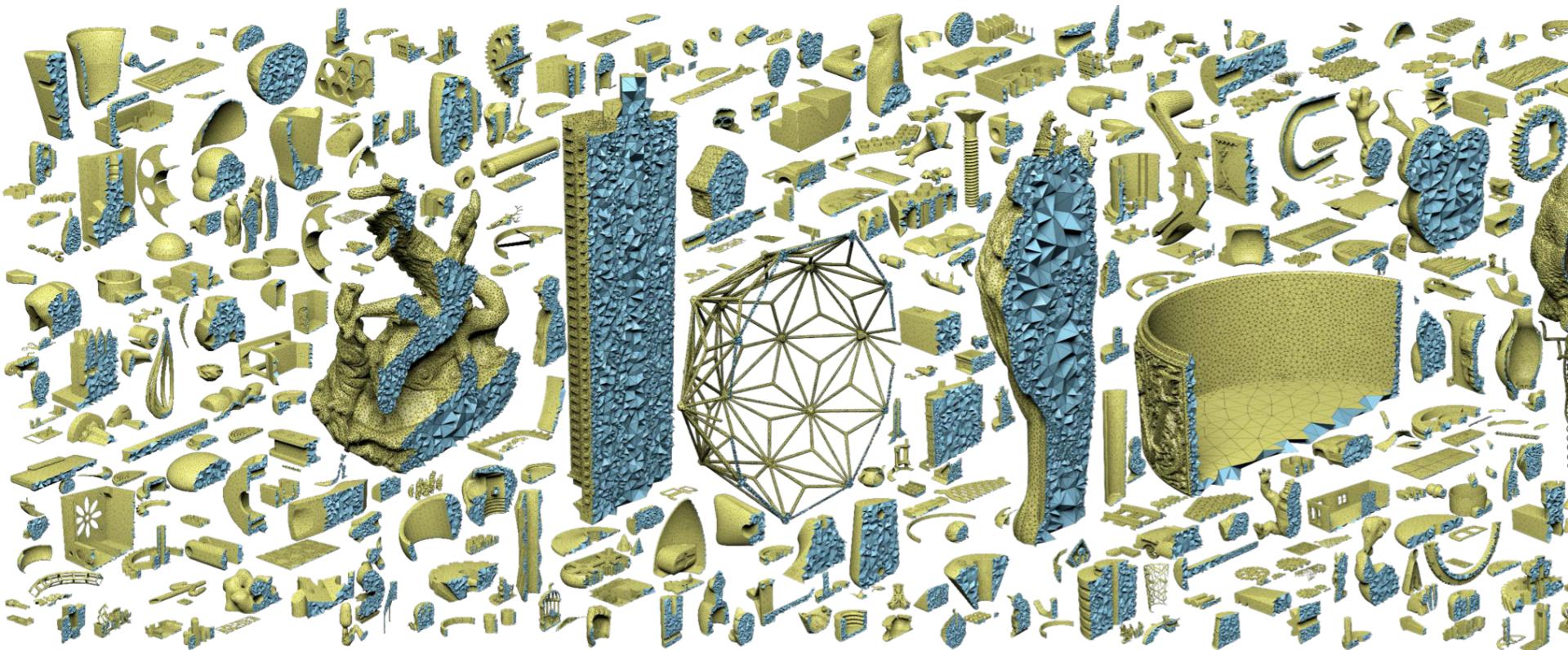


# Tetrahedral Meshing

- Triangle mesh -> tetrahedral mesh
- Surface (triangle) -> Volume (tetrahedron)
- Finite element method

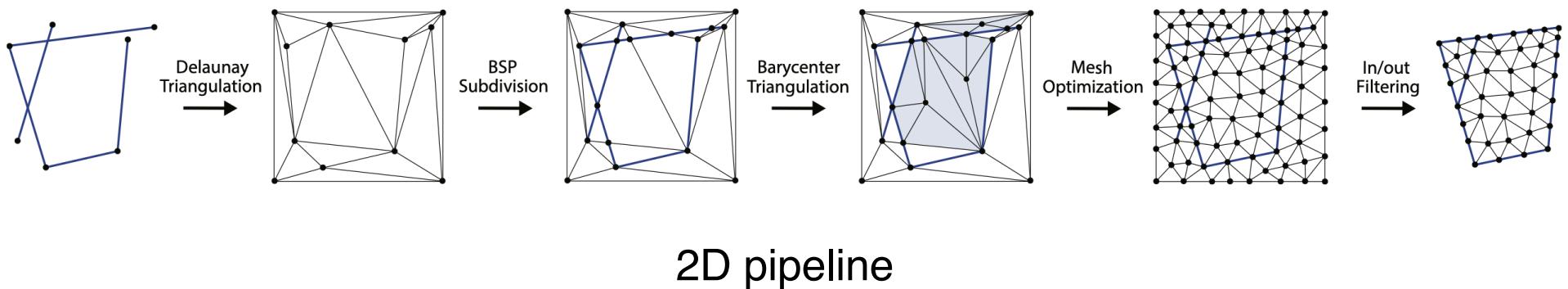


Tetrahedron



# Tetrahedral Meshing

- Hu Y, Zhou Q, Gao X, Jacobson A, Zorin D, Panozzo D. “**Tetrahedral meshing in the wild**”. Siggraph 2018
- Hu Y, Schneider T, Wang B, Zorin D, Panozzo D. “**Fast tetrahedral meshing in the wild.**” Siggraph 2020



*Intro to CSE291-D for Spring, 2021*

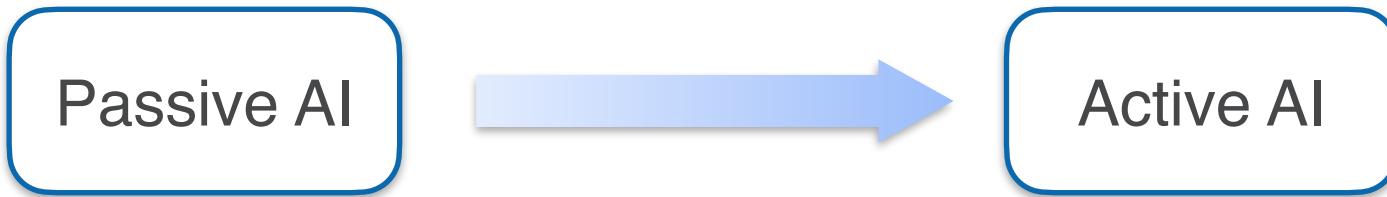
**Machine Learning for Robotics**

# Why Are We Interested In Robotics?

Passive AI

- We know how to fit data well (by “deep learning”)
  - e.g., computer vision, natural language processing

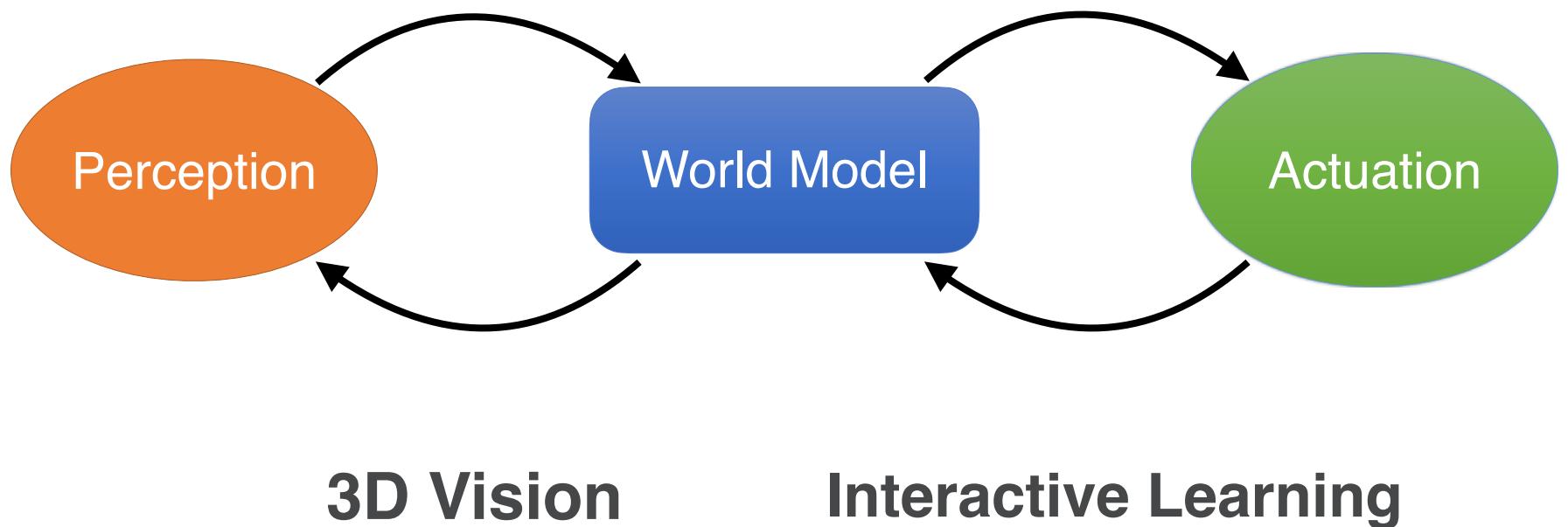
# Why Are We Interested In Robotics?



- We aspire that autonomous agents can perform tasks and “grow” through interaction experiences
  - Need the ability to **interact**

This quarter

Next quarter



# Syllabus

- Part I: Geometry and Physics of Robots
  - Robot Body
  - Robot Motion
  - Robot-Object Interaction
  - Physical Simulation
  - Classical Robotics Pipeline
- Part II: Reinforcement Learning
  - RL Concepts
  - RL as an Optimization Problem
  - Long-horizon RL
  - Generalizable RL

**Thank you for the efforts of the the quarter!**