Course Introduction

Richard (Hao) Zhang

CMPT 464/764: Geometric Modeling in Computer Graphics

Lecture 1

CMPT 464/764

- **CG = synthesis** of all visual (especially 3D) content, with **four pillars**

modeling + rendering + processing/manipulation + animation

CMPT 464/764

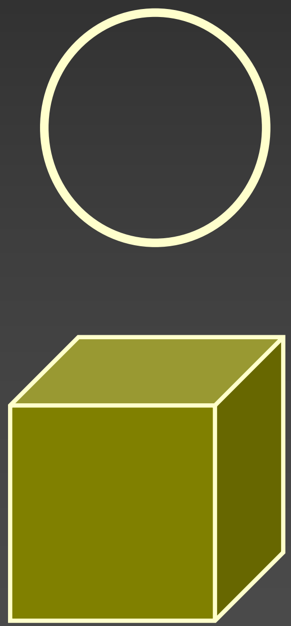
- **CG = synthesis** of all visual (especially 3D) content, with **four pillars**

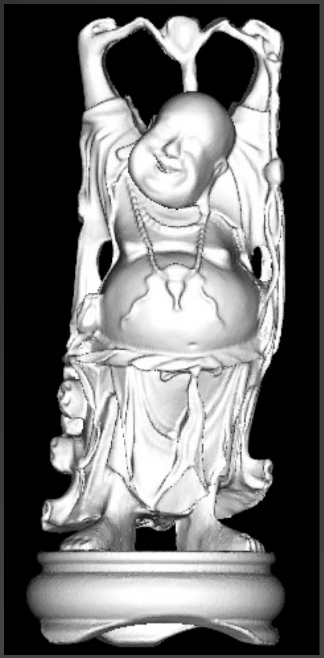
  modeling + rendering + processing/manipulation + animation

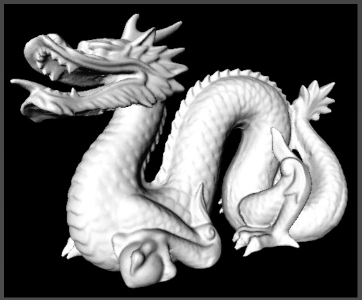
- Focuses on **modeling** (reconstruction + generation) and **understanding** of **3D shapes**
- Compared to CMPT 361, we now deal with much more complex shapes, rather than a single line, polygon, or curved patch

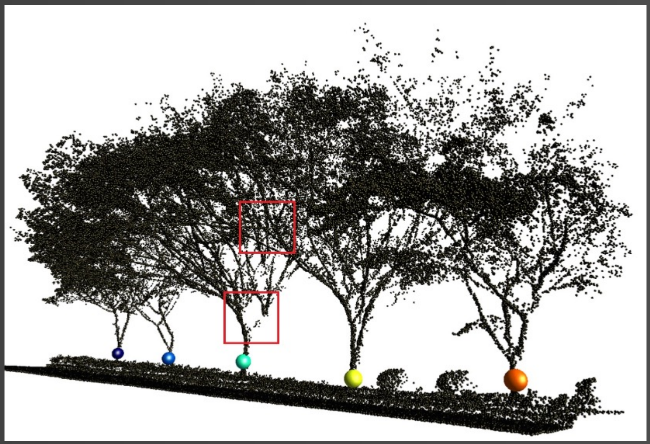
What is a shape?

- Give me a shape …

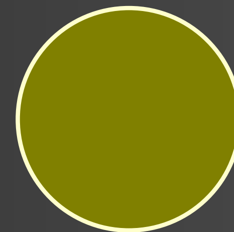


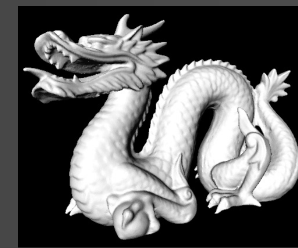






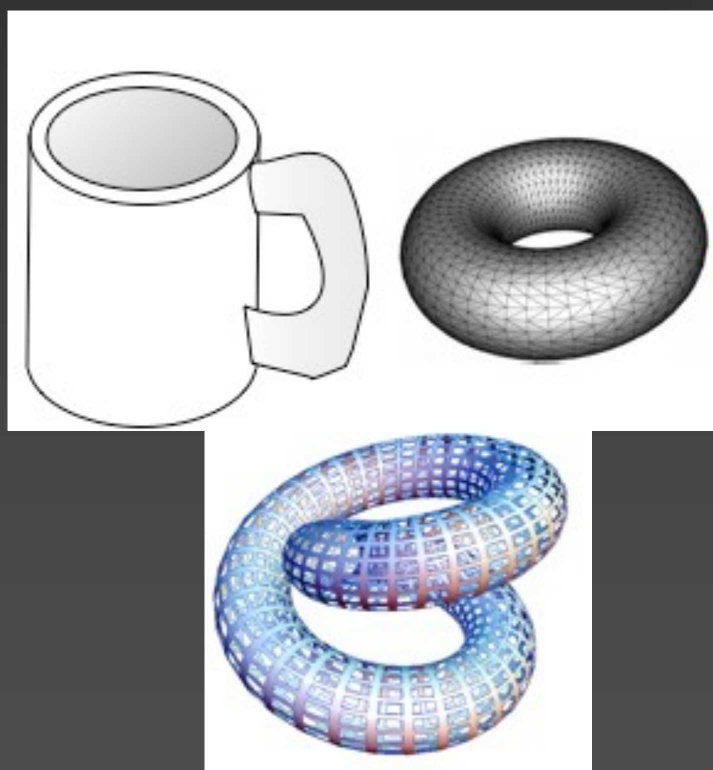
Sphere vs. the dragon

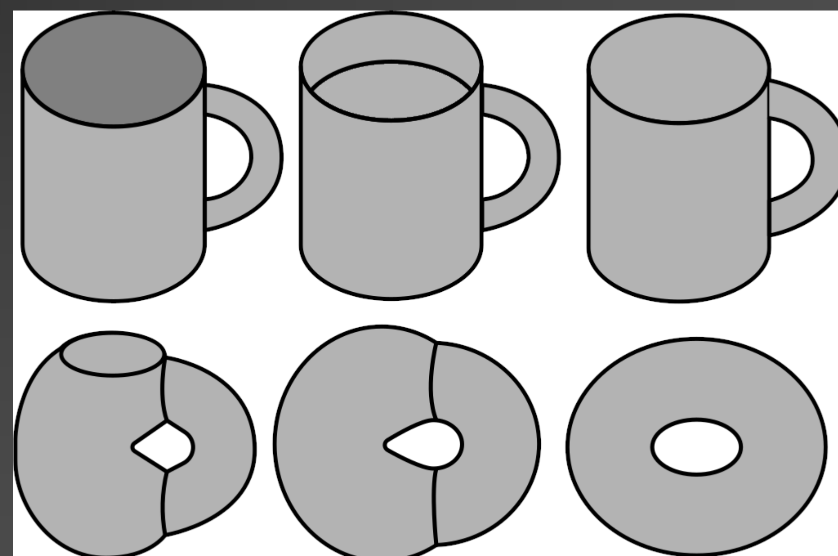




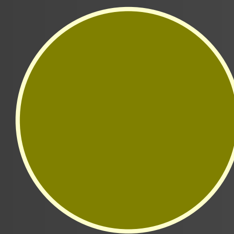
- Despite large geometric differences, could be the same **topologically**

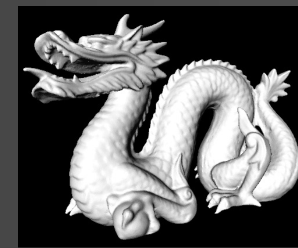
  Topological equivalence: clay sculpting without tearing or joining





Sphere vs. the dragon

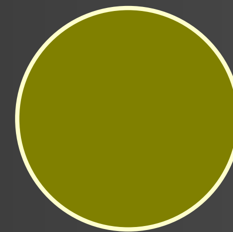




- Despite large geometric differences, could be the same **topologically**
- Sphere has a **close-form mathematical representation**
  - <u>Explicit</u> representation for hemisphere: $z = (r^2 - x^2 - y^2)^{1/2}$
  - <u>Implicit</u>: $x^2 + y^2 + z^2 = r^2$
  - <u>Parametric</u>: $(r \cos\phi \cos\theta, r \cos\phi \sin\theta, r \sin\phi)$

  Not so clear for the dragon

Sphere vs. the dragon



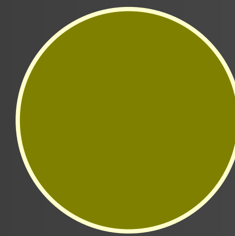
- Could be the same **topologically**
- Sphere has a **close-form mathematical representation**
  - Explicit representation for hemisphere: $z = (r^2 - x^2 - y^2)^{1/2}$
  -
  - Parametric: $(r \cos\phi \cos\theta, r \cos\phi \sin\theta, r \sin\phi)$

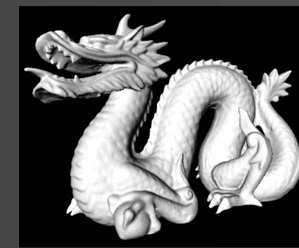
  Not so clear for the dragon

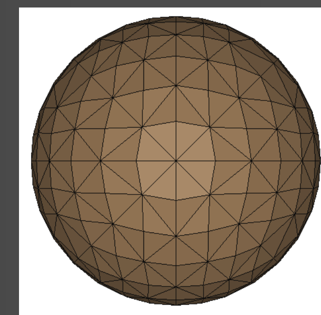
- Sphere representation is **compact**: radius & center (high-level)

  The dragon need many points **explicitly** specified, e.g., using a mesh with ~440K vertices and ~870K faces, or represented **implicitly** (later)

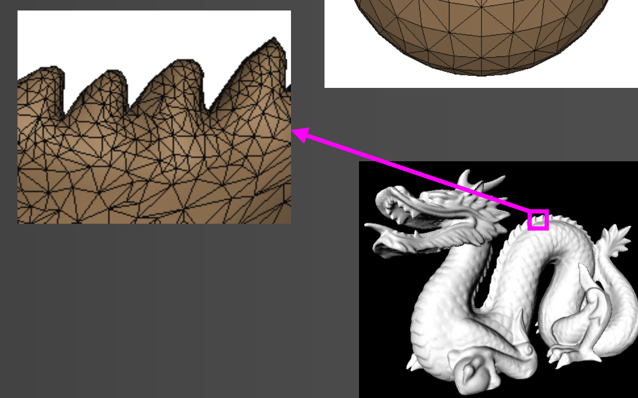
Sphere vs. the dragon



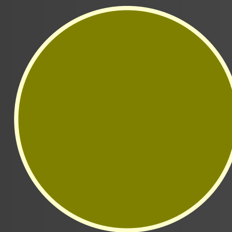


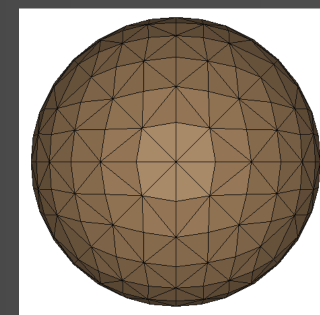


- How should the shapes be **rendered**? It depends on the (low-level) 3D representations
  - Parametric primitives → ray tracing
  - Meshes → polygon shading or ray tracing
  - Points (or Gaussians) → splatting
  - Voxels → volume rendering, e.g., through line integrals

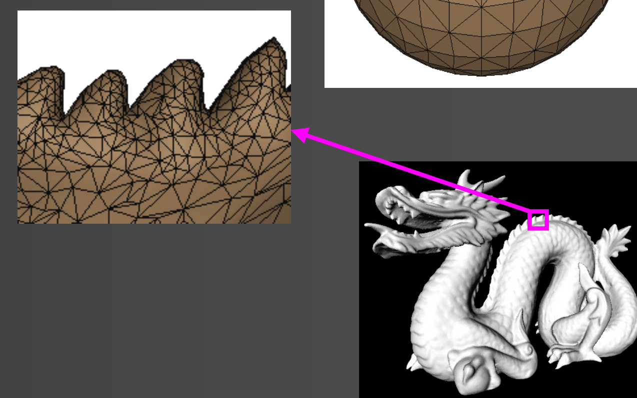


Sphere vs. the dragon

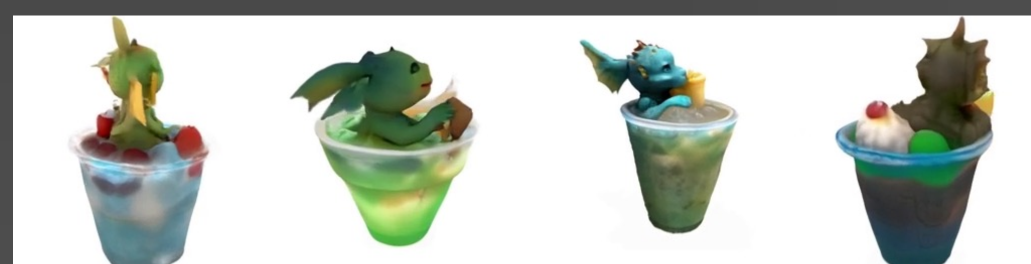




- How should the shapes be rendere[illegible] It depends on the level) 3D representations
  - Parametric primitives → ray tracing
  - [illegible] cing
  - Points (or Gaussians) → splatting
  - Voxels → volume rendering, e.g., through line integrals



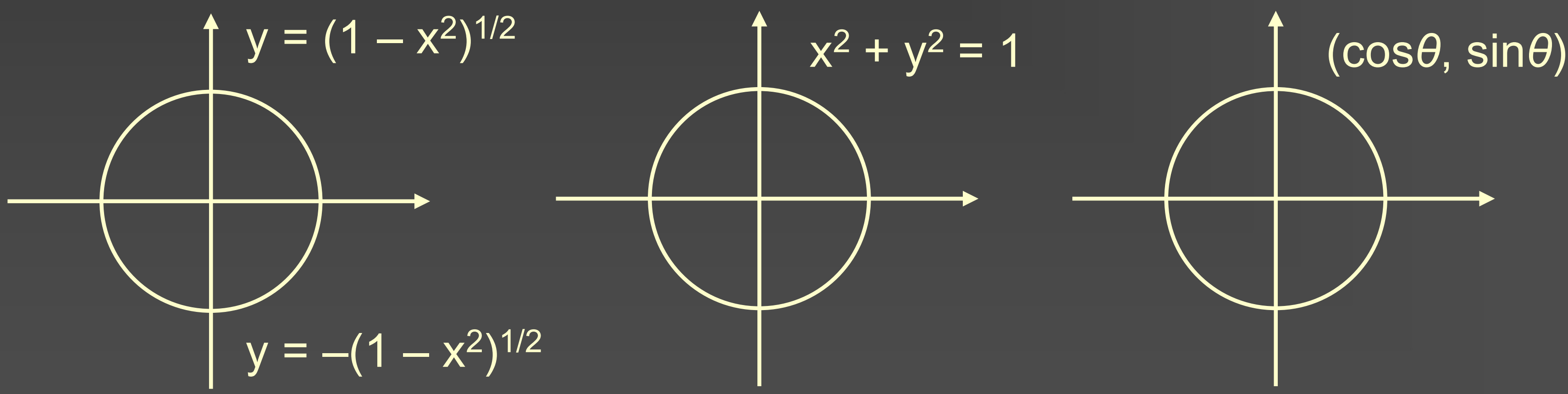
- How to generate? From text? Various modeling paradigms (next lecture)



"A baby dragon drinking boba"

How do we represent shapes?

- Classical way – **defined with closed-form functions**
  - **Explicit**: $y = f(x)$, $z = f(x, y)$, $w = f(x, y, z)$
  - **Implicit**: $f(x, y, z) = 0$ – implicit surface modeling
  - **Parametric**: $(x(t), y(t))$, $(x(u, v), y(u, v), z(u, v))$ – e.g., Bezier or B-spline

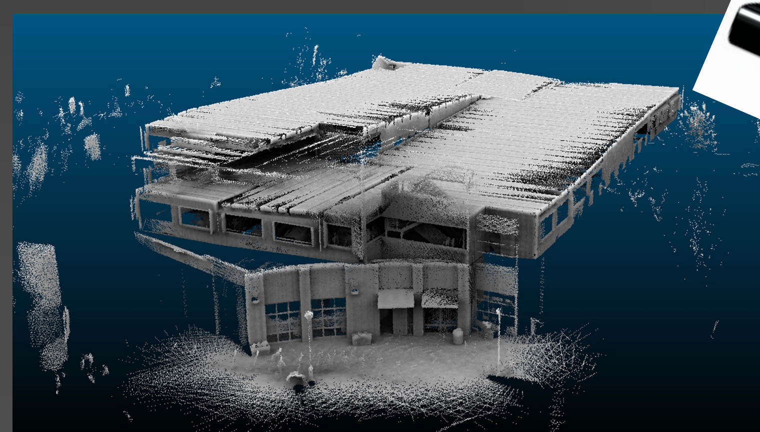


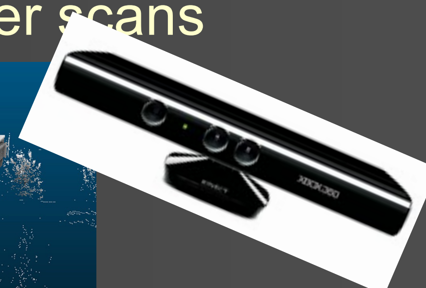
Model acquisition or learning

- What if a closed-form function is not readily available (e.g., for the dragon/Buddha)? — such model needs to be **acquired or learned**
- How? Text? — too coarse and too ambiguous

Model acquisition or learning

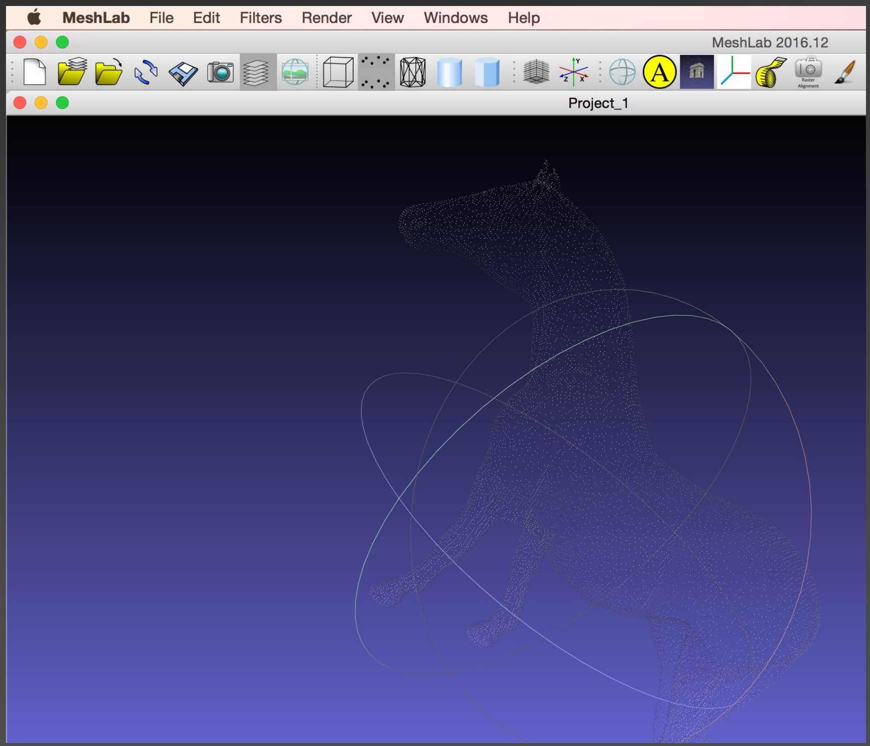
- What if a closed-form function is not readily available (e.g., for the dragon/Buddha)? — such model needs to be **acquired or learned**
- How? Text? — too coarse and too ambiguous
- Take a picture? — probably need more than one picture (**multi-view representation**) and 3D reasoning is still hard (lectures later)
- More reliable: acquire/learn from **point clouds** by laser scans

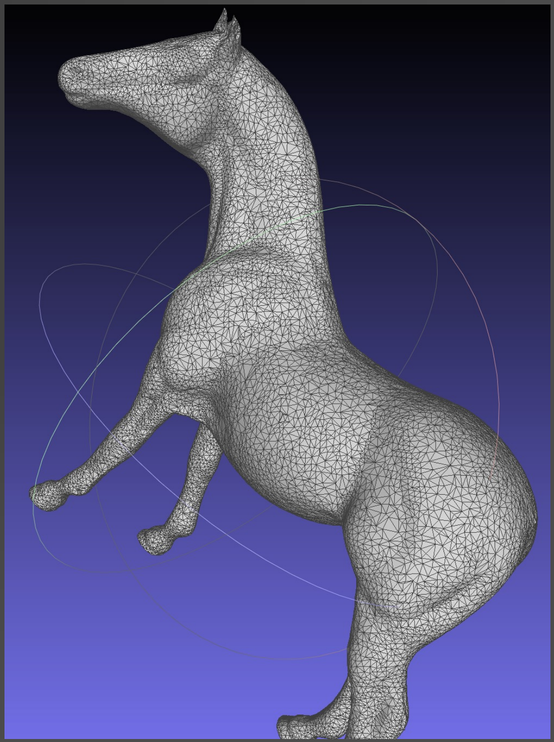




Example: points and meshes

- Points can be connected to form a **piecewise linear** approximation of original shape – a **polygonal mesh**



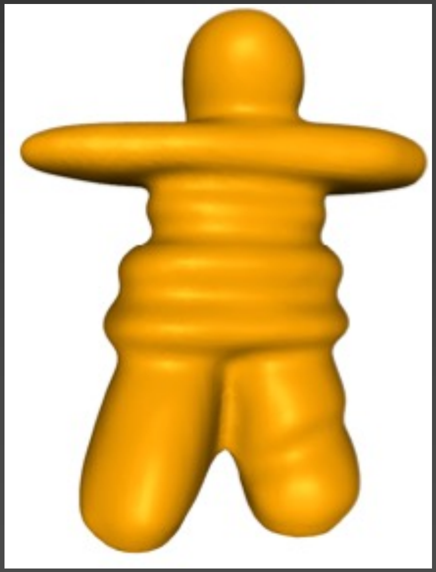


Key problems: reconstruction

- How to connect the points to obtain an accurate approximation?
  - **Surface reconstruction**

Harder with missing/incomplete data!

Laser scan



Poisson

MPU

Key problems: reconstruction

- Most popular: 3D from **multi-view and sensors**
  - – Reconstruction from **single- or multi-view images**
  - – Reconstructed results **“stored” in a neural network**

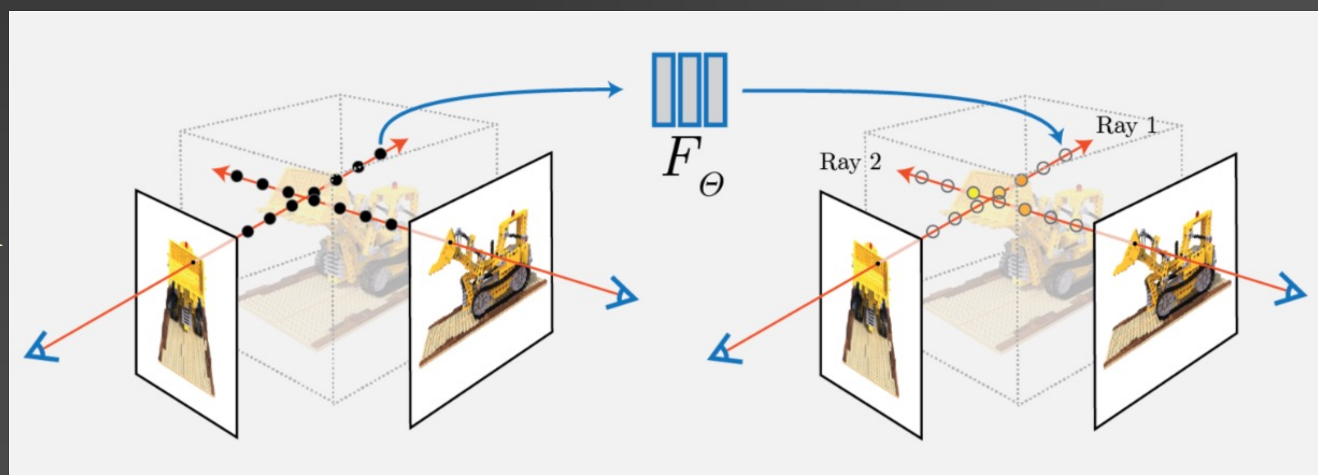
CVPR 2023 by the Numbers

Selecting a category below changes the paper list on the right.

SELECT ↓ Top 10 overall by number of authors

		AUTHORS	PAPERS
1	3D from multi-view and sensors	1,090	246
2	Image and video synthesis and generation	889	185
3	Humans: Face, body, pose, gesture, movement	813	166
4	Transfer, meta, low-shot, continual, or long-tail learning	688	153
5	Recognition: Categorization, detection, retrieval	673	139
6	Vision, language, and reasoning	631	118
7	Low-level vision	553	126
8	Segmentation, grouping and shape analysis	524	113
9	Deep learning architectures and techniques	485	92
10	Multi-modal learning	450	89
11	3D from single images	431	91
12	Medical and biological vision, cell microscopy	420	53
13	Video: Action and event understanding	373	83
14	Autonomous driving	359	69
15	Self-supervised or unsupervised representation learning	349	71
16	Datasets and evaluation	344	54
17	Scene analysis and understanding	276	54
18	Adversarial attack and defense	274	61
19	Efficient and scalable vision	252	48
20	Computational imaging	226	53
21	Video: Low-level analysis, motion, and tracking	215	46
22	Vision applications and systems	171	35
23	Vision + graphics	155	32
24	Robotics	141	23
25	Transparency, fairness, accountability, privacy, ethics in vision	129	30
26	Explainable computer vision	107	24
27	Embodied vision: Active agents, simulation	80	14
28	Document analysis and understanding	72	12
29	Machine learning (other than deep learning)	65	14
30	Physics-based vision and shape-from-X	55	12
31	Biometrics	51	11
32	Others	47	12
33	Optimization methods (other than deep learning)	46	12
34	Photogrammetry and remote sensing	38	8
35	Computer vision theory	33	5
36	Computer vision for social good	25	5

Single-view image



Neural Radiance Field (NeRF) for novel view synthesis
[Mildenhall et al. ECCV 2020]

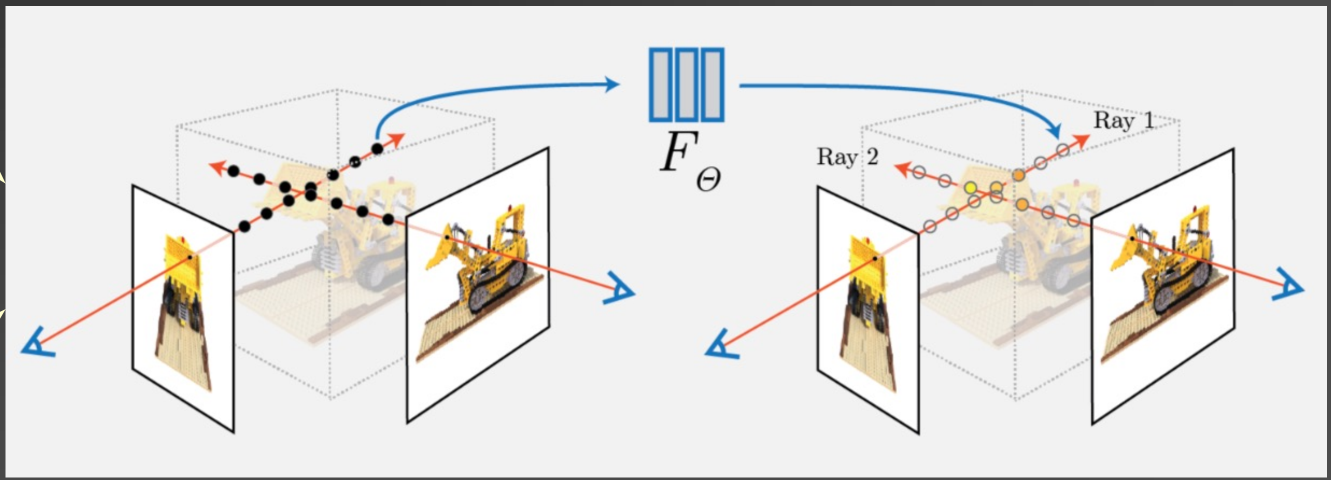
Key problems: 3D generative modeling

- Most popular: 3D Generative AI (GenAI)
  – Basically **hallucination** (“dreaming?”) with some level of **conditioning**

Text:
“A Lego front loader”

or

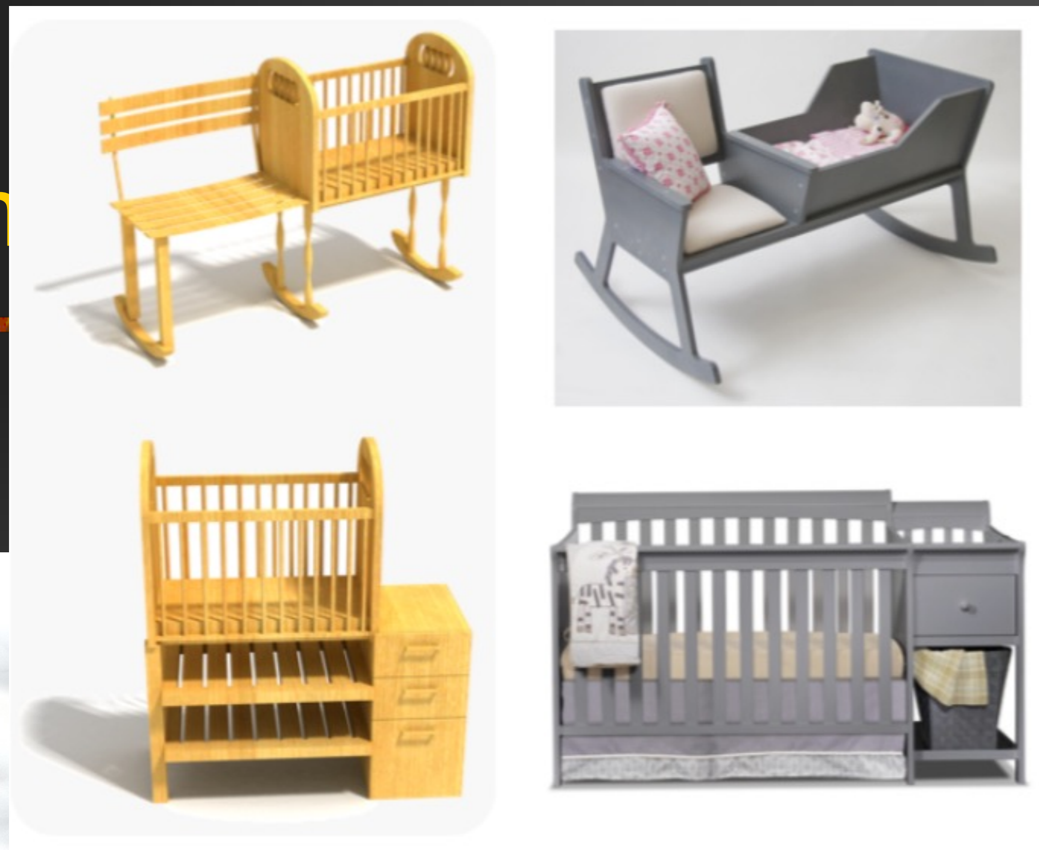
Single-view image



Neural Radiance Field (NeRF) for novel view synthesis
[Mildenhall et al. ECCV 2020]

Creative m

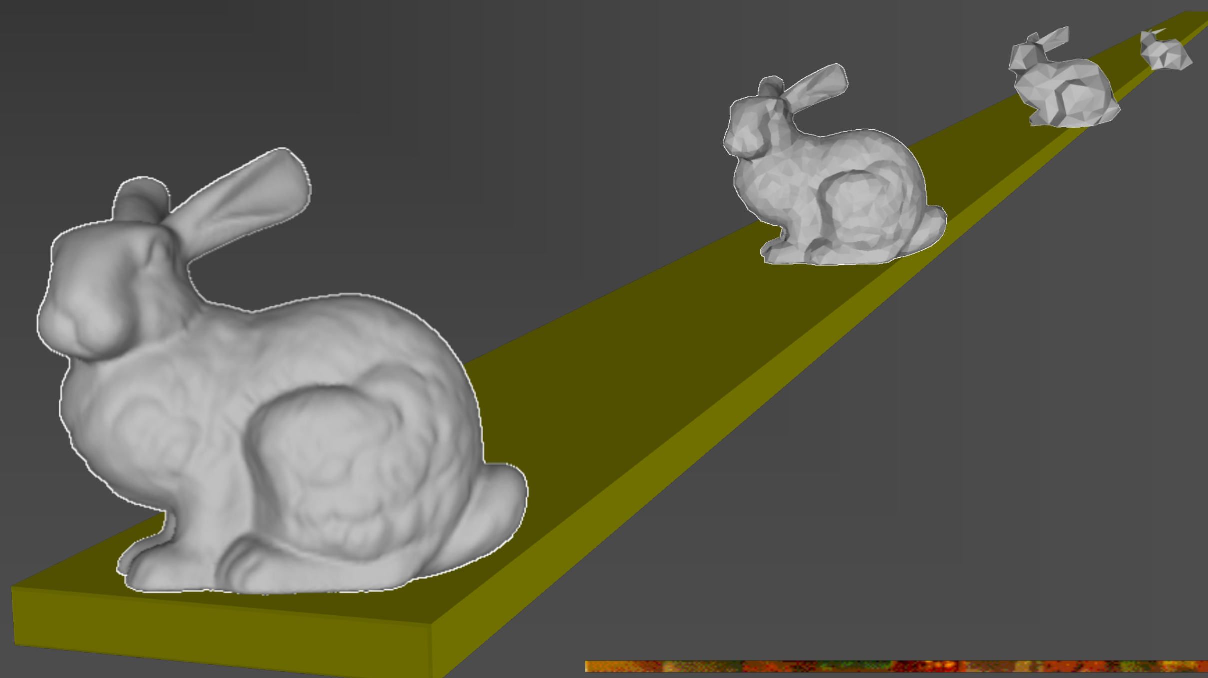
- How to **create**



FAME (Functionality-Aware Model Evolution) for example-based 3D shape creation [Guan et al. IEEE TVCG 2022]

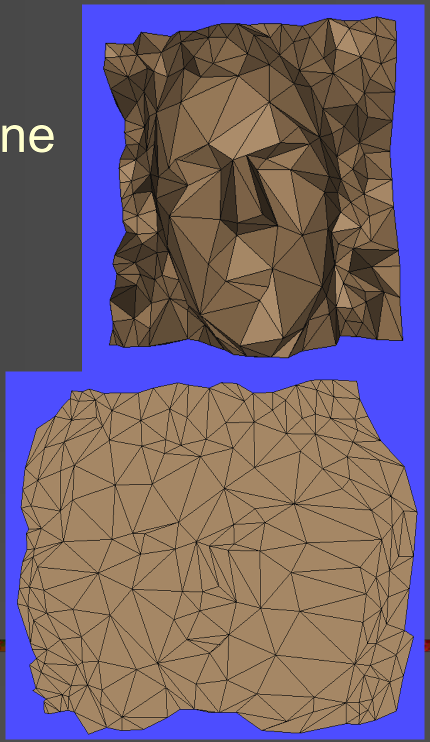
Other problems (not covered in this course)

- What if full details are not required, e.g., when object is far/moving?
  - – the problem of **mesh decimation** and **multiresolution (LOD) modeling**



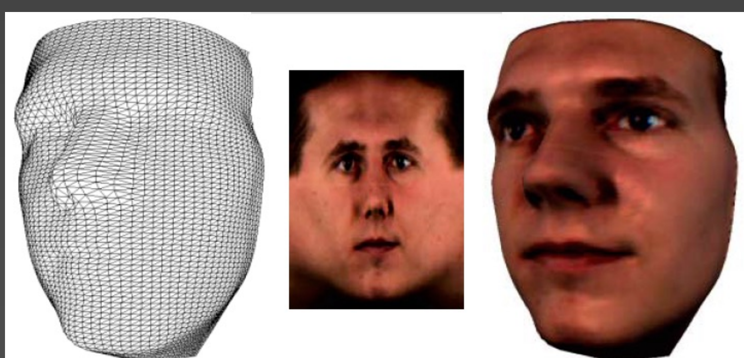
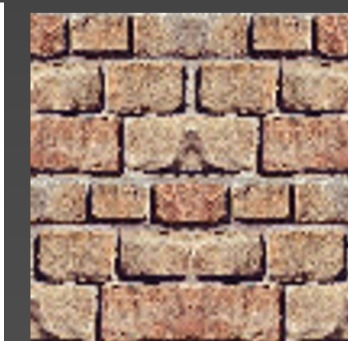
Other problems (not covered in this course)

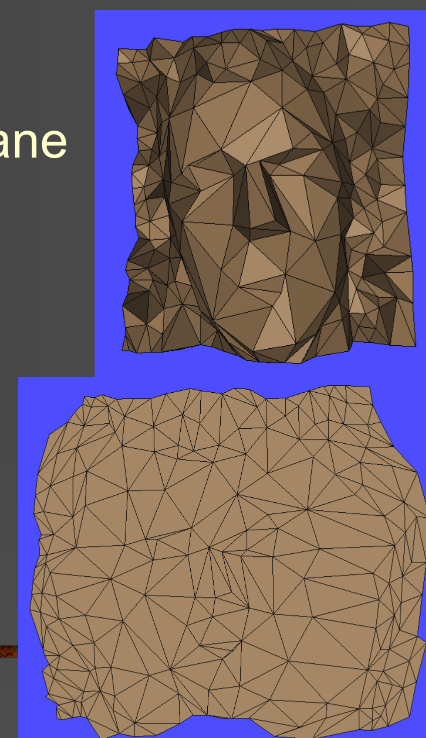
- Full model needed, but the file is too large to transfer
  - – the problem of **mesh compression**
- So much is known about functions (derivatives, etc.), but acquired mesh models are not functions yet
  - – make it a function: defined over regular domain, e.g., 2D plane



Other problems (not covered in this course)

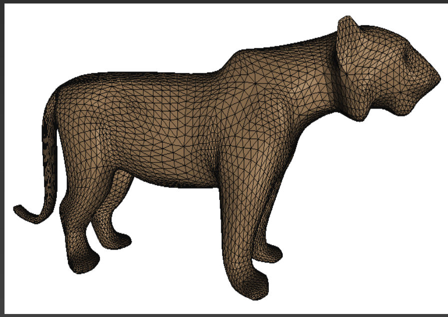
- Full model needed, but the file is too large to transfer
  – the problem of **mesh compression**
- So much is known about functions (derivatives, etc.), but acquired mesh models are not functions yet
  – make it a function: defined over regular domain, e.g., 2D plane
  – **mesh parameterization** a.k.a. **texture mapping**

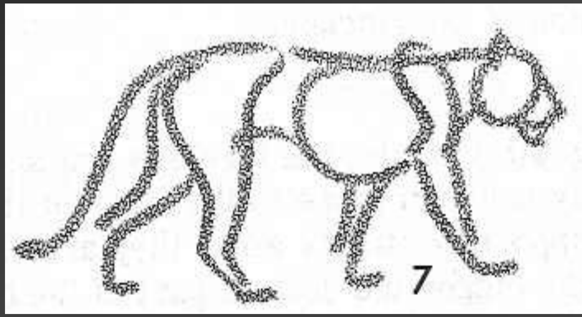



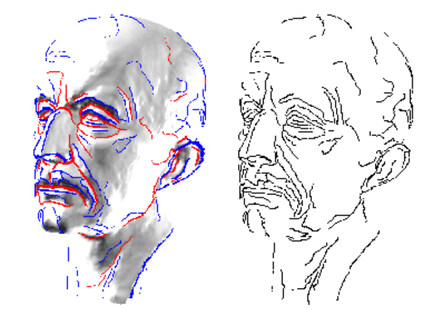


Analysis problems in CMPT 464/764

- No need for the tens of thousand of triangles! Only need few feature lines and a **high-level** description (e.g., an **abstraction**), e.g.,

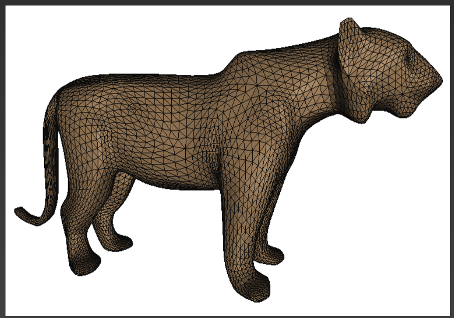
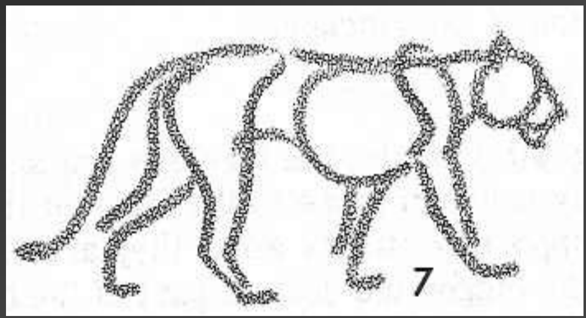


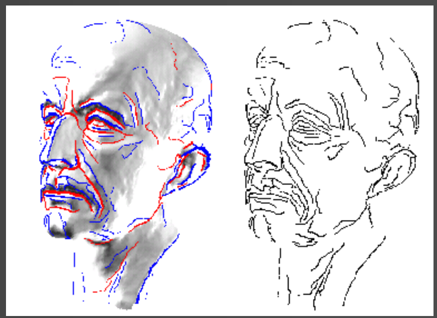




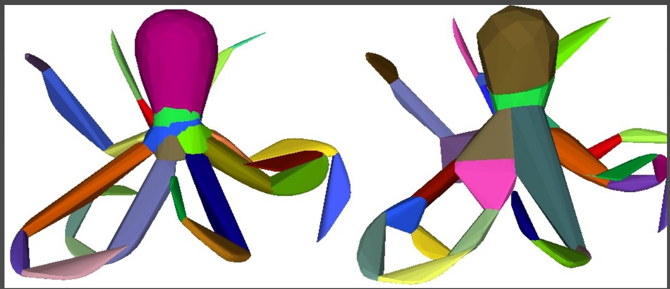
Analysis problems in CMPT 464/764

- No need for the tens of thousand of triangles! Only need few feature lines and a **high-level** description (e.g., an **abstraction**), e.g.,



- Or an organization of the **constituent parts** of shape
  – the problem of **mesh segmentation**, and in general, **3D shape analysis**




Low vs. high level processing

- Low-level geometry processing

The HIP thing: HIgh-level geometry Processing

✓ non-local analysis; structure-aware, e.g., symmetry

✓ not easy to formulate objectives mathematically

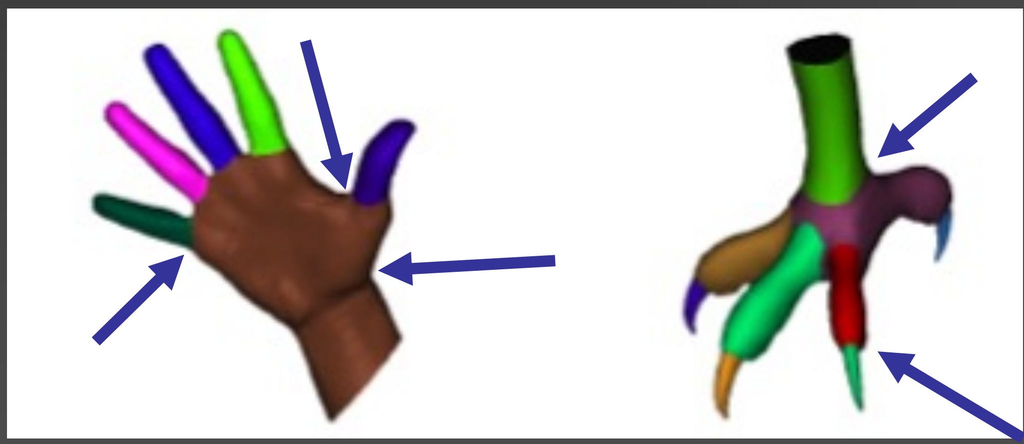
✓ utilization of prior knowledge

✓ moving from model-driven to data-driven and ML

Example: shape segmentation

- A local criteria from study of visual perception

Minima rule: cut boundary is at negative minima of curvature. Roughly speaking, over concavity.



Use of the minima rule

5 parts

16 parts

More meaningful … at a higher level

"An understanding of semantics"

Symmetry

5 parts

5 parts

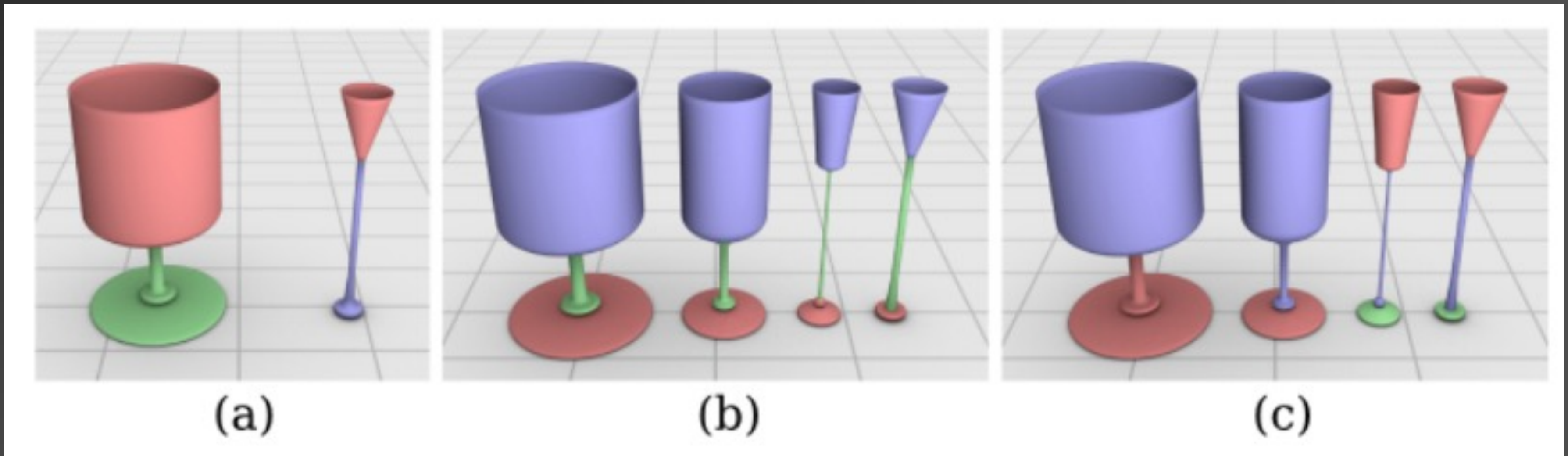
New challenges to segmentation and other analyses:

- ❑ How to do it really well to approach human ability?
- ❑ Knowledge-driven: supervised/unsupervised
- ❑ Other knowledge: e.g., utilization of a set = co-segmentation

Co-segmentation (higher-level)

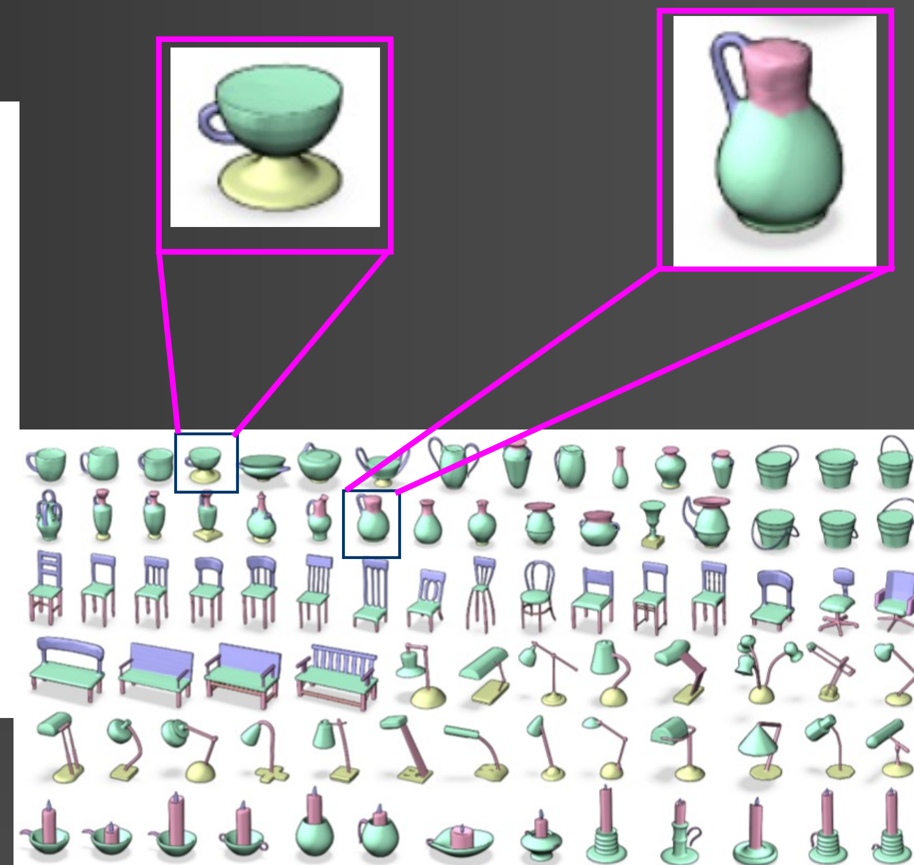
O. Sidi, O van Kaick, Y. Klienman, H. Zhang, D. Cohen-Or, "Unsupervised Co-Segmentation of a Set of Shapes via Descriptor-Space Spectral Clustering", *SIGGRAPH Asia 2011.*

Power of a set …



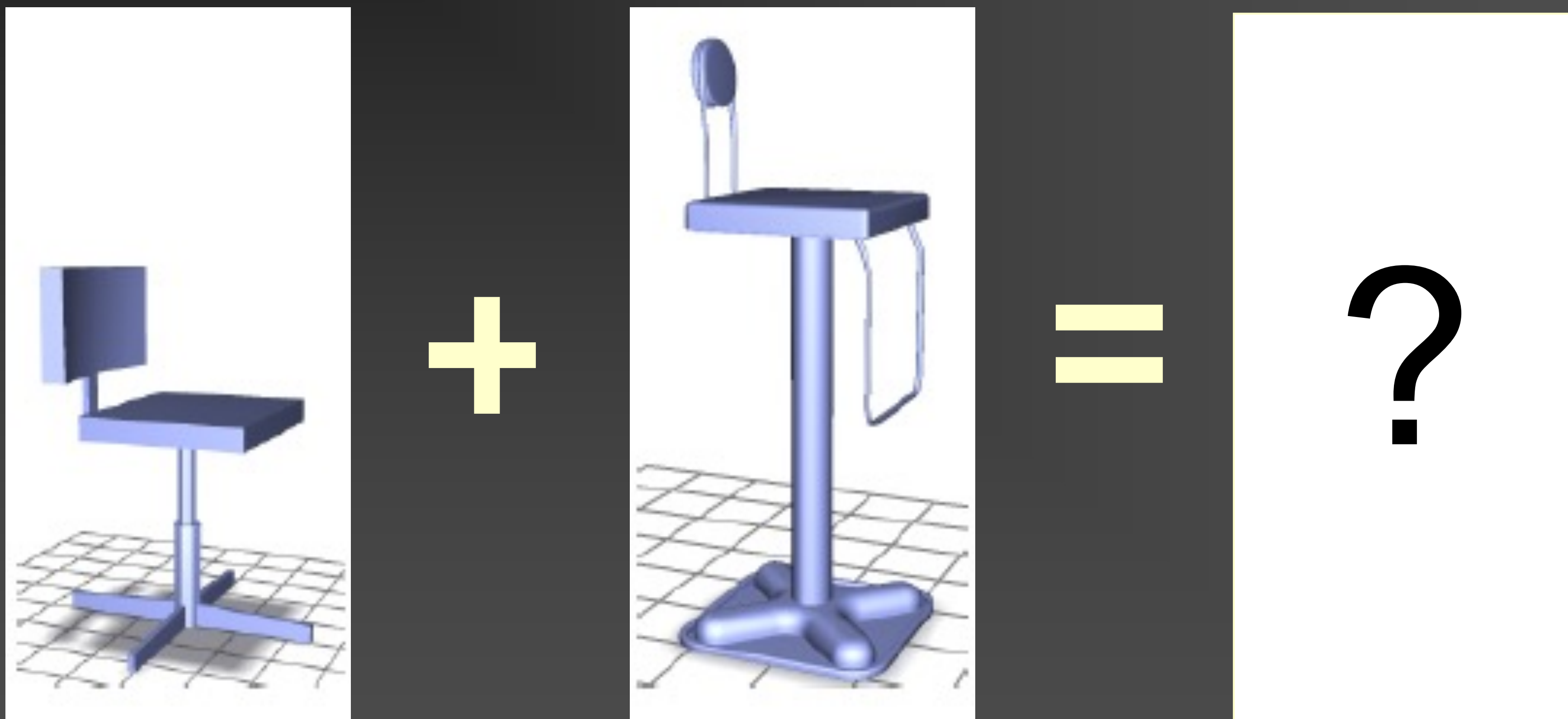
O. Sidi, O van Kaick, Y. Klienman, H. Zhang, D. Cohen-Or, “Unsupervised Co-Segmentation of a Set of Shapes via Descriptor-Space Spectral Clustering”, *SIGGRAPH Asia 2011.*

Shape correspondence (higher-level)



Utilization of prior knowledge (recognition) via training set or other learning mechanism

Recognize/understand before create



This course

- Modeling paradigms for 3D content creation
- Introduces various 3D representations and their neuralization
- Covers the **acquisition, analysis (understanding), (novel) creation, and fabrication** of **3D** (surface) shapes
- Quite a bit of machine learning: neural network basics, autoencoders, etc.
- Touches (lightly) upon topics from several mathematical fields, e.g.,
  - plane and stereo geometry; differential geometry, e.g., curvatures
  - combinatorial and computational geometry, e.g., Voronoi diagrams

What you can take away …

- Geometric modeling and processing basics
  - Basic concepts that have wide-ranging applications
  - Often beyond geometric modeling or computer graphics
  - Learned through lectures and reference readings
  - No advanced mathematics preparation assumed
- Becoming a semi-expert on a selected topic
  - Through completion of a programming assignment and course project

How can I do well?

- **You should feel inspired by the topics and future plans**
- Attend classes, be prepared, and be active in class
- Do not be shy about asking questions and work with you peers
- Good programming skills
- Good team work

Summary

- The new view of computer graphics
  - It is beyond image synthesis via rendering
  - It is about **creation and manipulation** of **all visual content**
  - Creation of **novel** 3D shapes is the new challenge
- We cover both classical topics on modeling, processing, and analysis of discrete geometric shapes as well as **emerging** topics
- Learn the classical techniques and be prepared for addressing challenges from new computer graphics

Exercises

- Play around with some existing mesh viewing software (Google around), in particular, **MeshLab**.
- Get familiar with some 3D geometry formats, e.g., OBJ, SMF, etc. (http://www.martinreddy.net/gfx/3d-hi.html)
- Try out some 3D modeling tools, e.g., Blender (https://www.blender.org/)
- Try out some attempts as automated 3D creation: e.g., IM-NET (https://github.com/czq142857/IM-NET)
- Check out latest buzz on 3D GenAI, from Nvidia/Google/Autodesk/Amazon