# CS 468
# Data-driven Shape Analysis

# Introduction

*April 1, 2014*

# Instructors

**Vladimir (Vova) Kim**
vk2@stanford.edu

**Qixing (Peter) Huang**
huangqx@stanford.edu

# Course Info

## Administrative

- We meet: Tu, Th, 9:30-10:45, Gates B12
- Website: http://cs468.stanford.edu
- Office Hours:
  - Vova: Tuesdays, 11:00am-noon, Clark S297
  - Peter: Friday, 1:00pm-2:00pm, Clark S297

## Grading (tentative)

- Homework #1: Shape Descriptors - 20%
- Homework #2: Shape Matching / Correspondence - 20%
- Homework #3: Machine Learning for Shape Analysis - 20%
- Final Project: your choice, work in groups - 40%

# Course Prerequisites

Expected Background

- Computer Graphics
- Programming
- Math

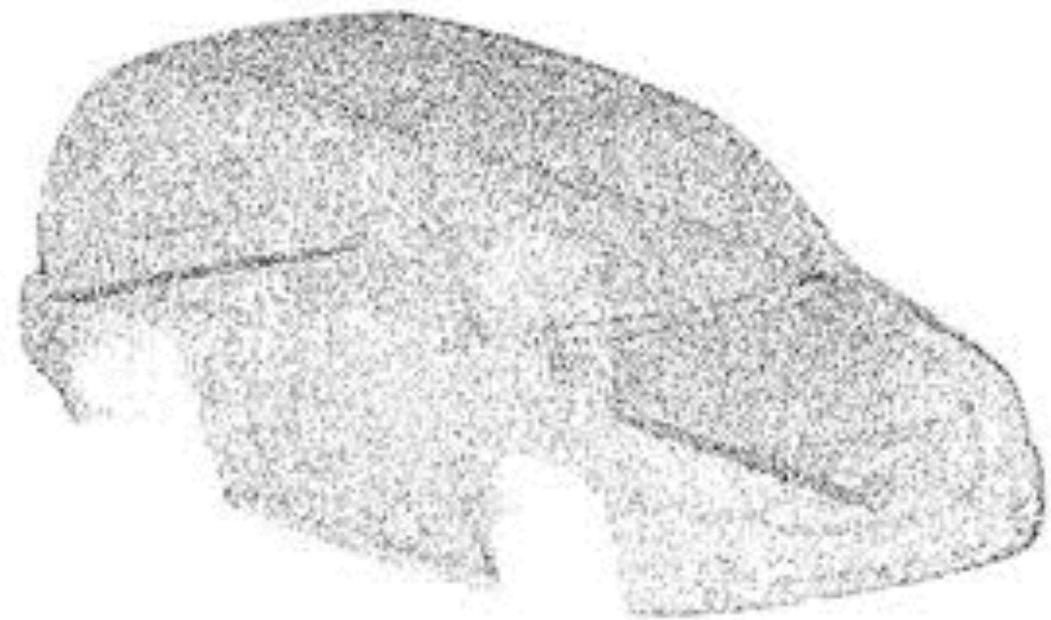
# Course Prerequisites

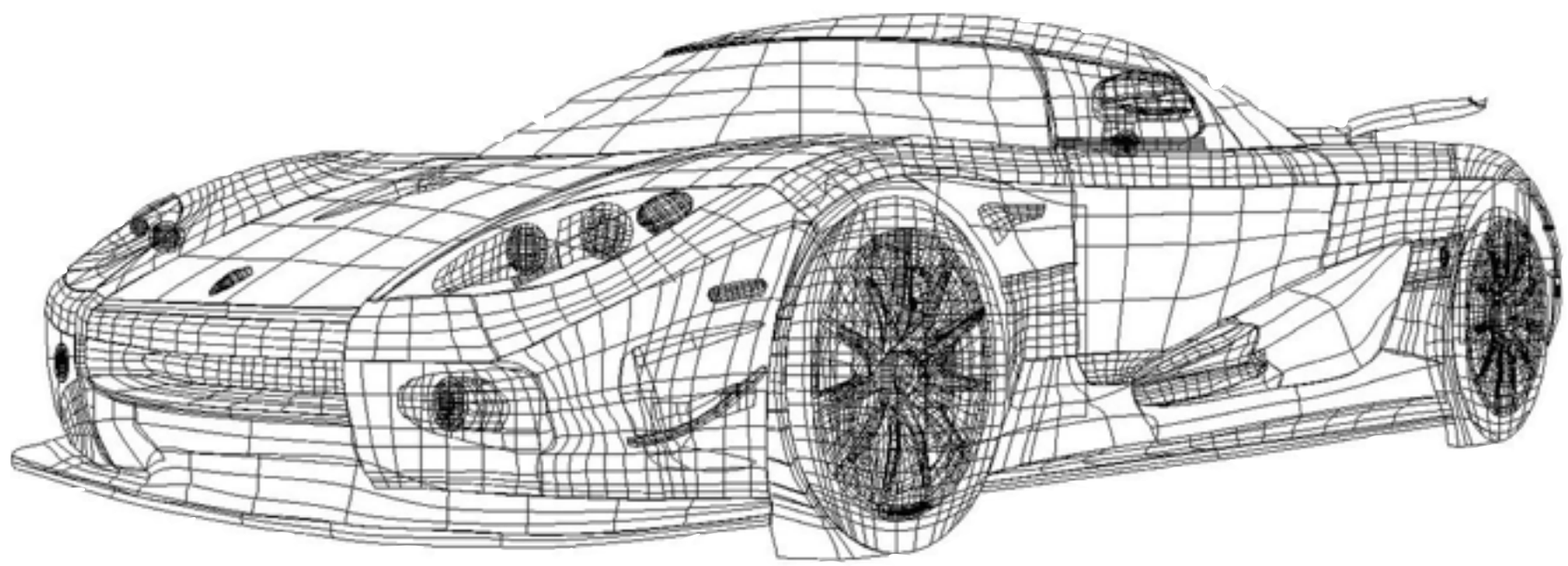
Expected Background

➡ **Computer Graphics**

- Point Clouds and Polygonal Meshes
- Kd-Trees
- Other basic data structures…

- Programming
- Math

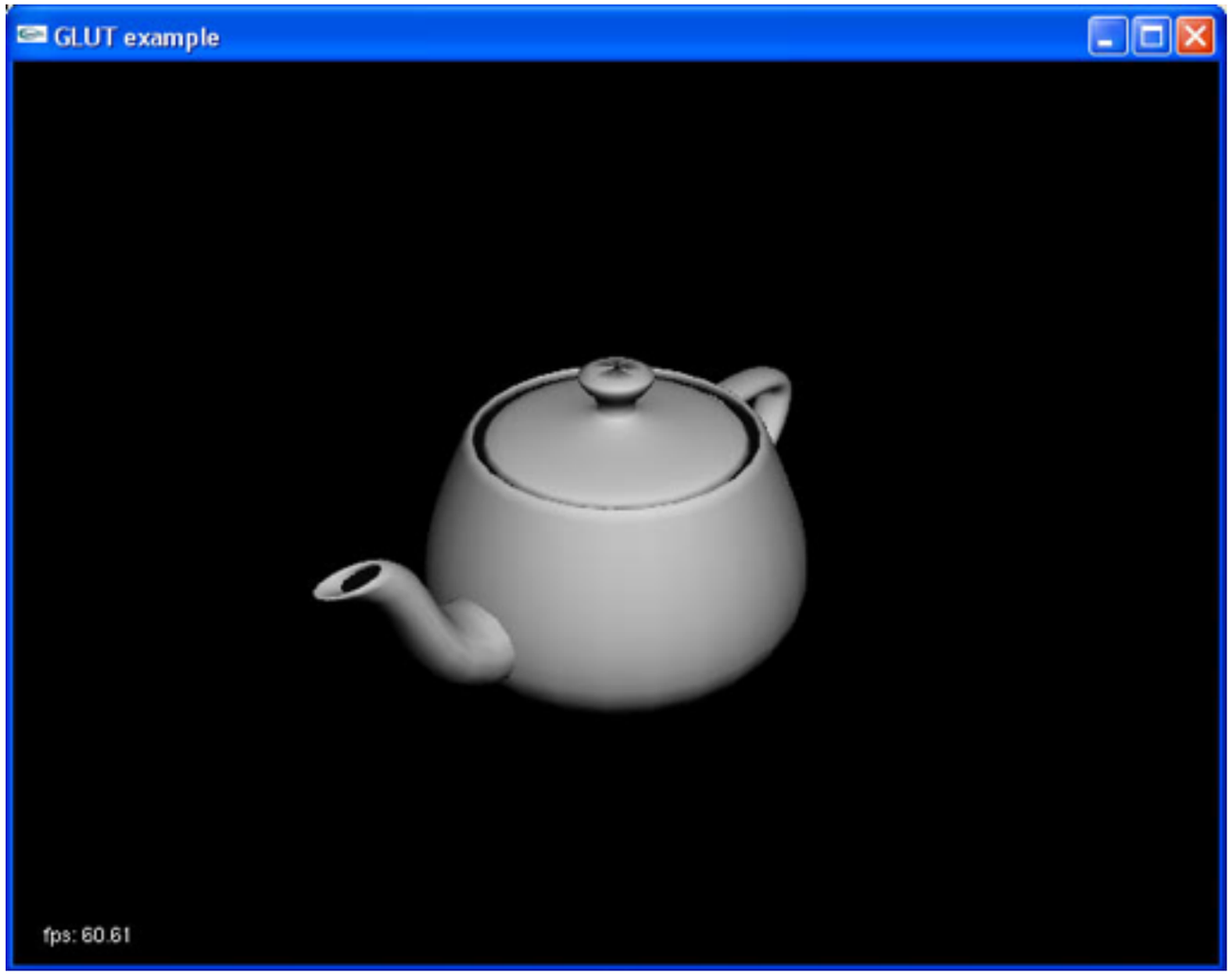




# Course Prerequisites

Expected Background

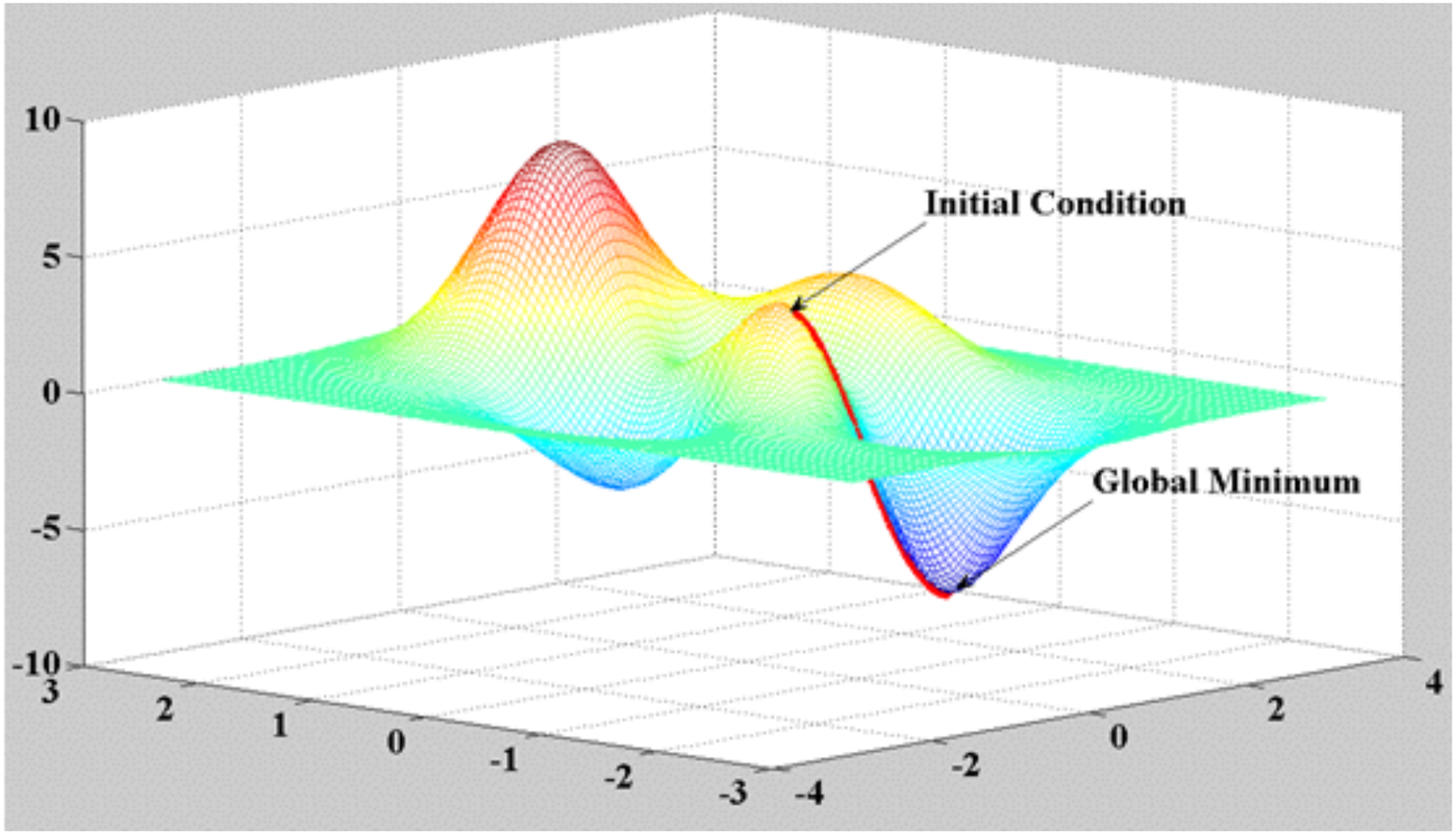
- Computer Graphics
- ➡ **Programming**
  - C++ / Matlab
  - OpenGL (a bit)
- Math



# Course Prerequisites

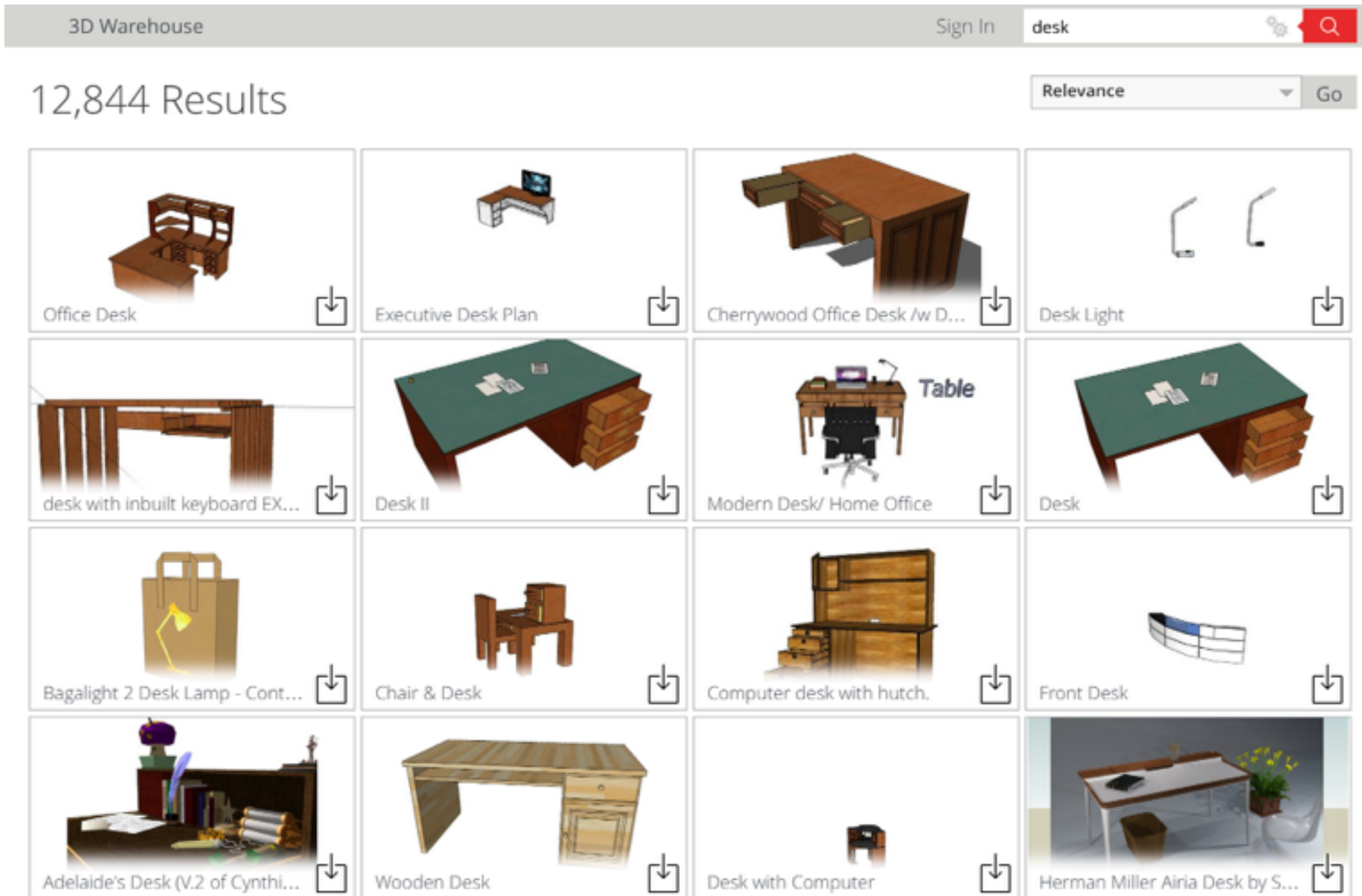
Expected Background

- Computer Graphics
- Programming
- ➡ **Math**
  - Linear Algebra
  - Optimization



# Motivation

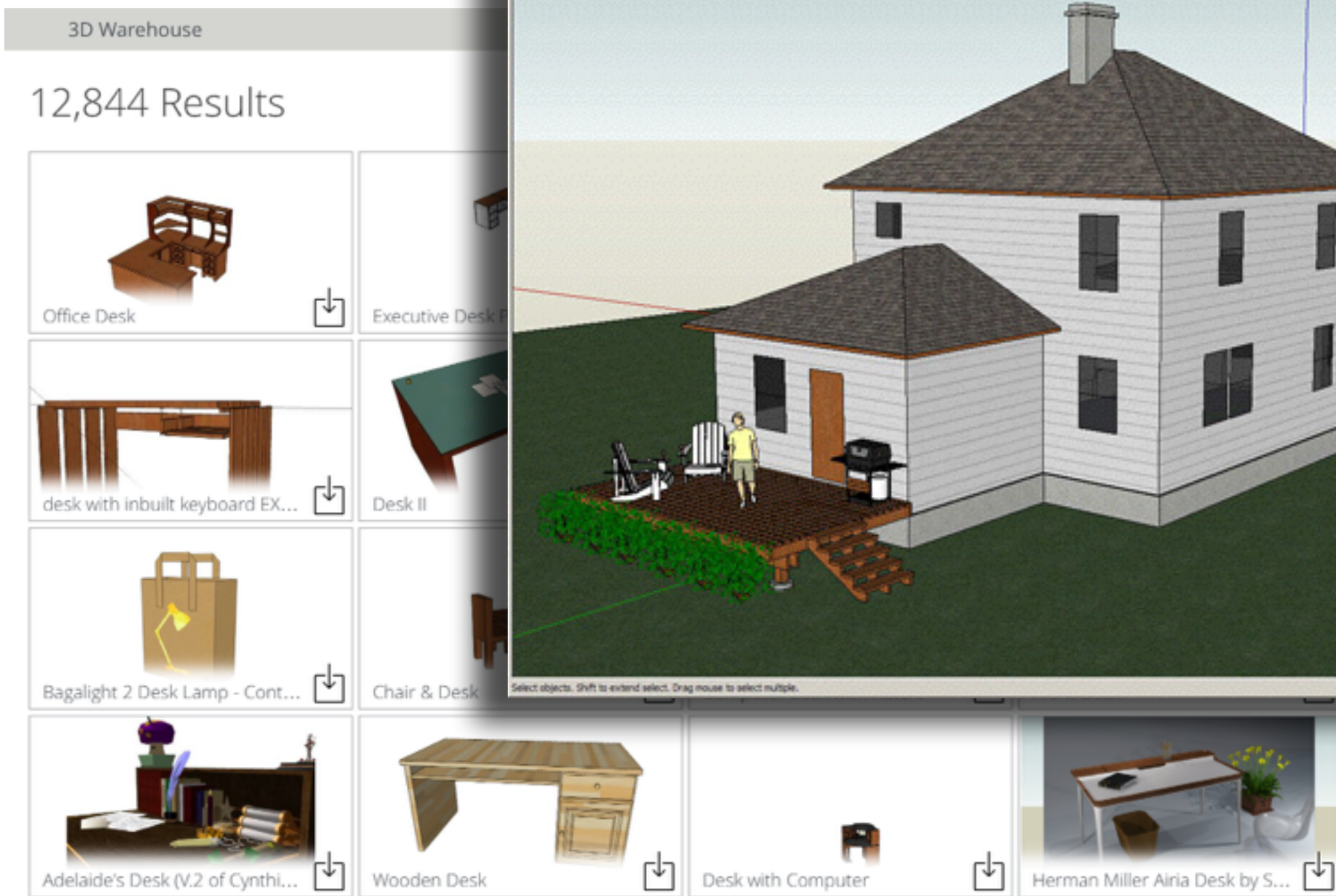
Large and growing repositories of 3D Models

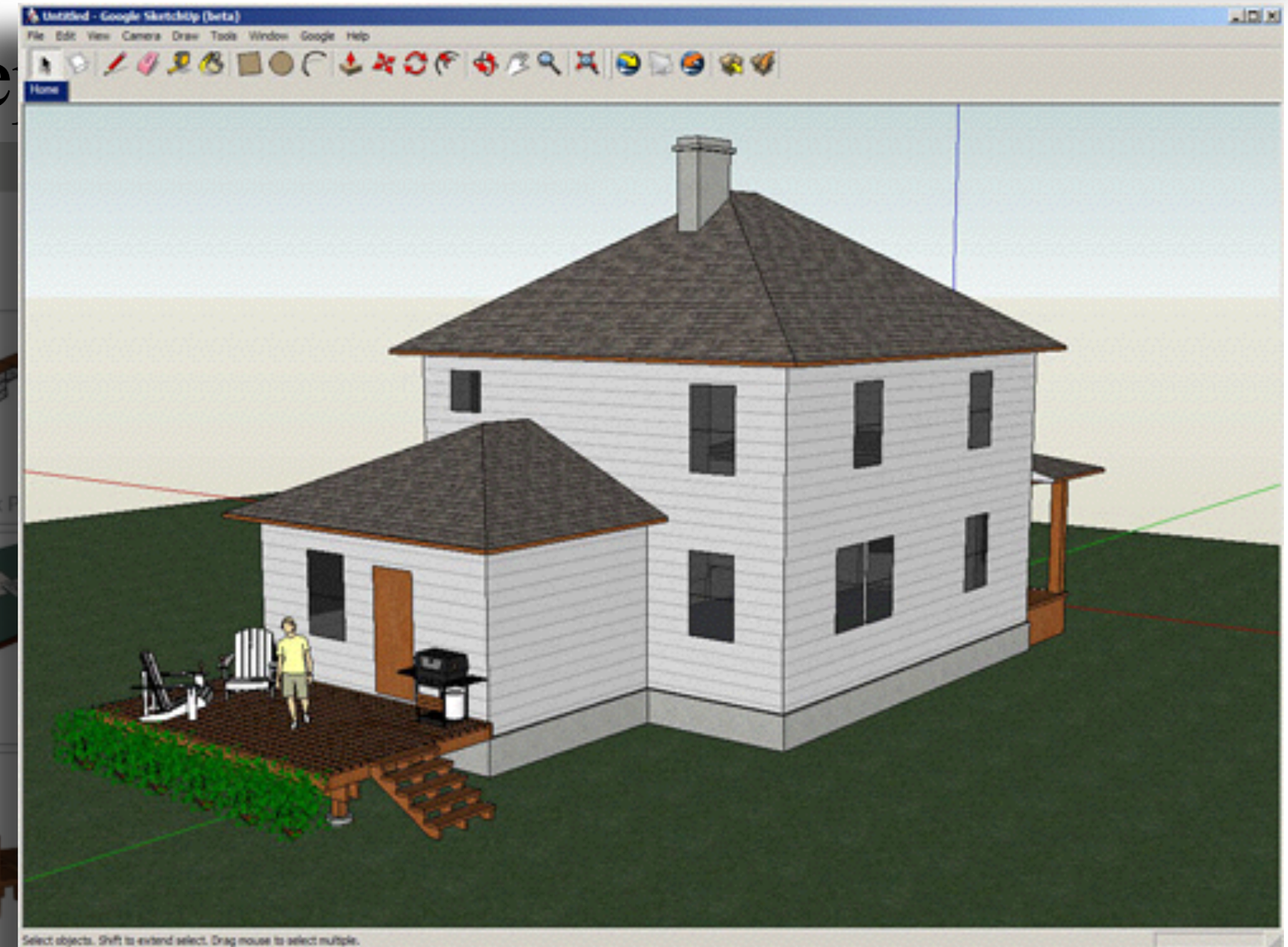



Trimble 3D Warehouse

# Motivation

Large and growing re

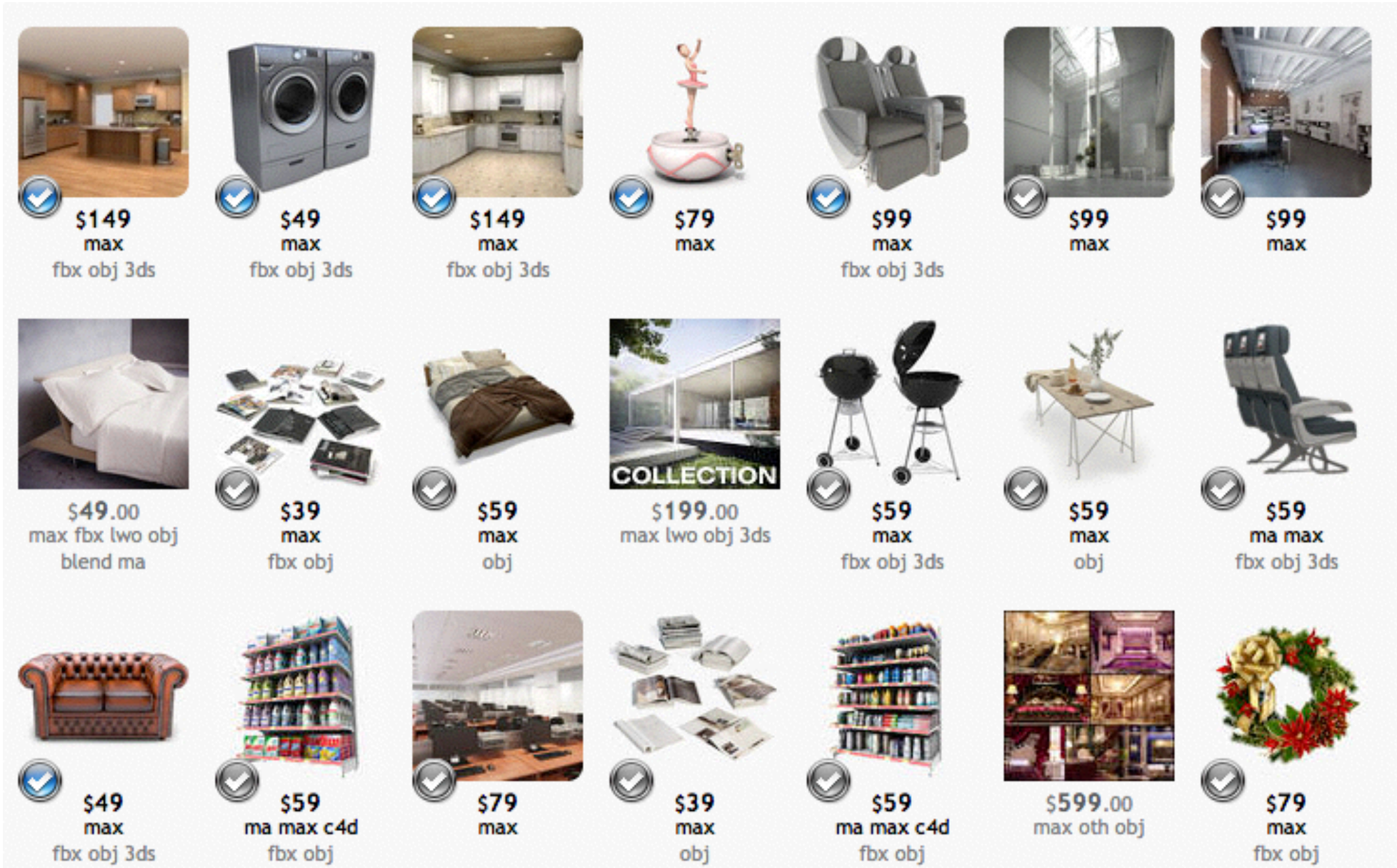



Trimble 3D Warehouse

# Motivation

Large and growing repositories of 3D Models

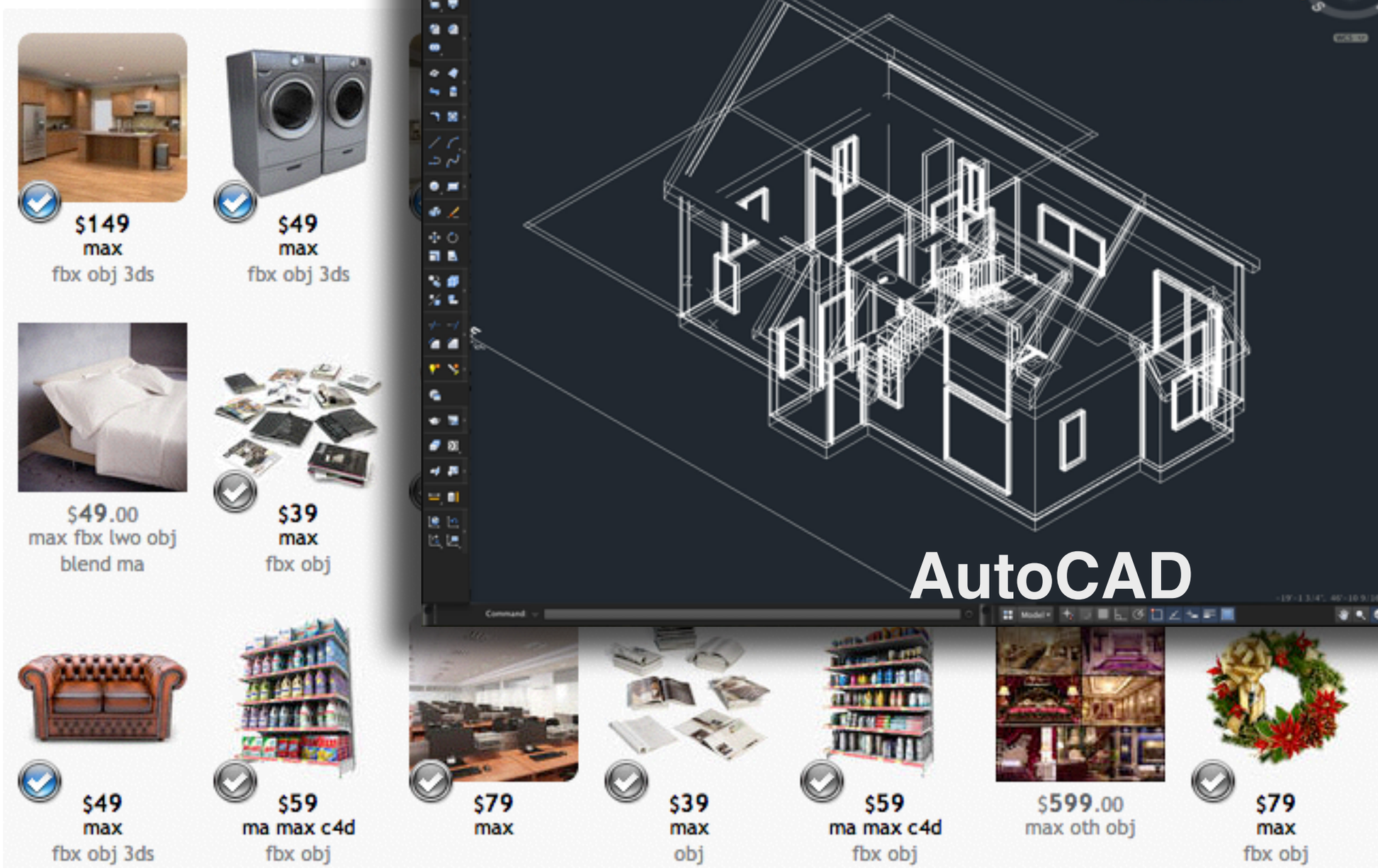



TURBOSQUID

Digimation

# Motivation

Large and growin





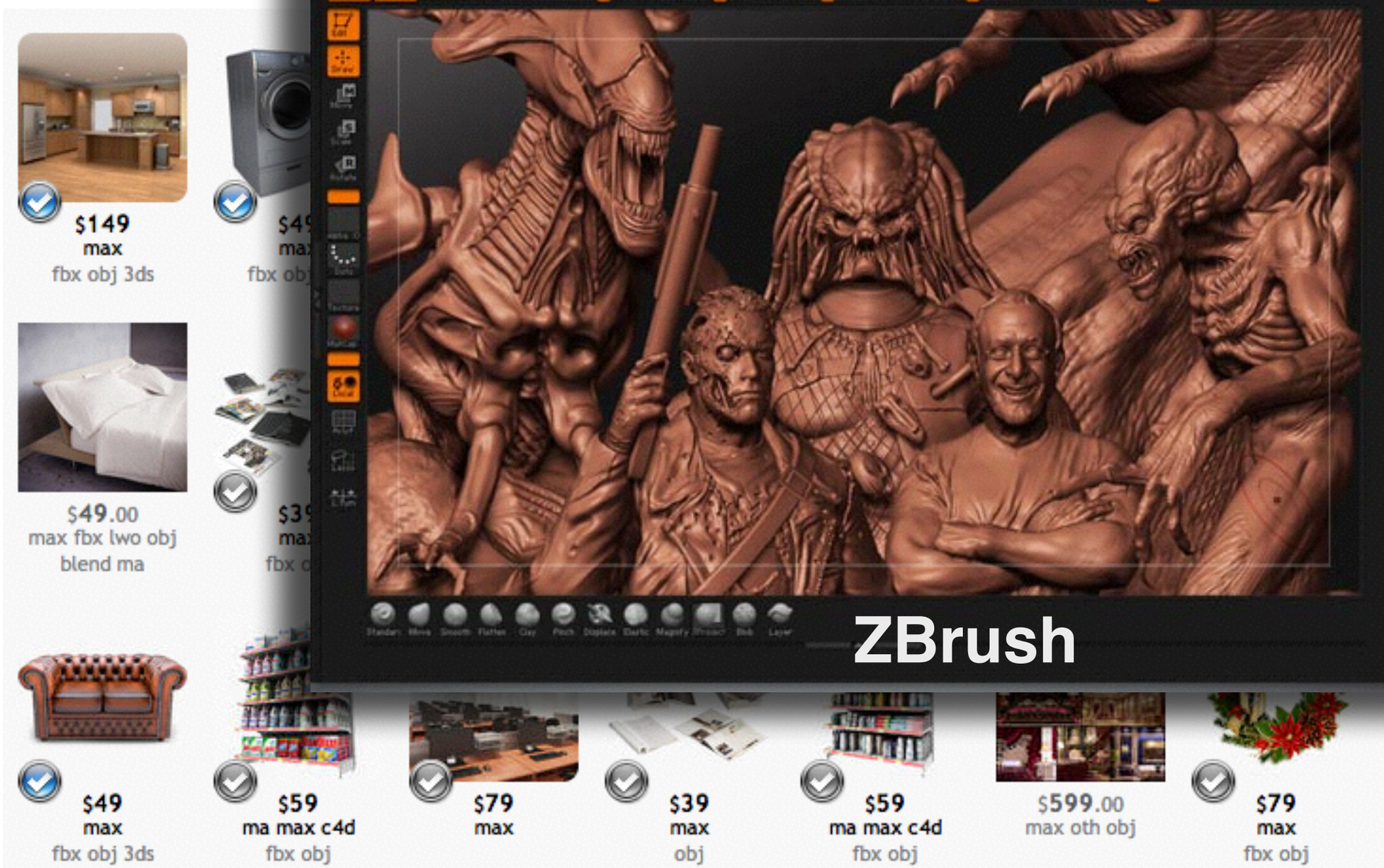


TURBOSQUID

Digimation

# Motivation

Large and gro



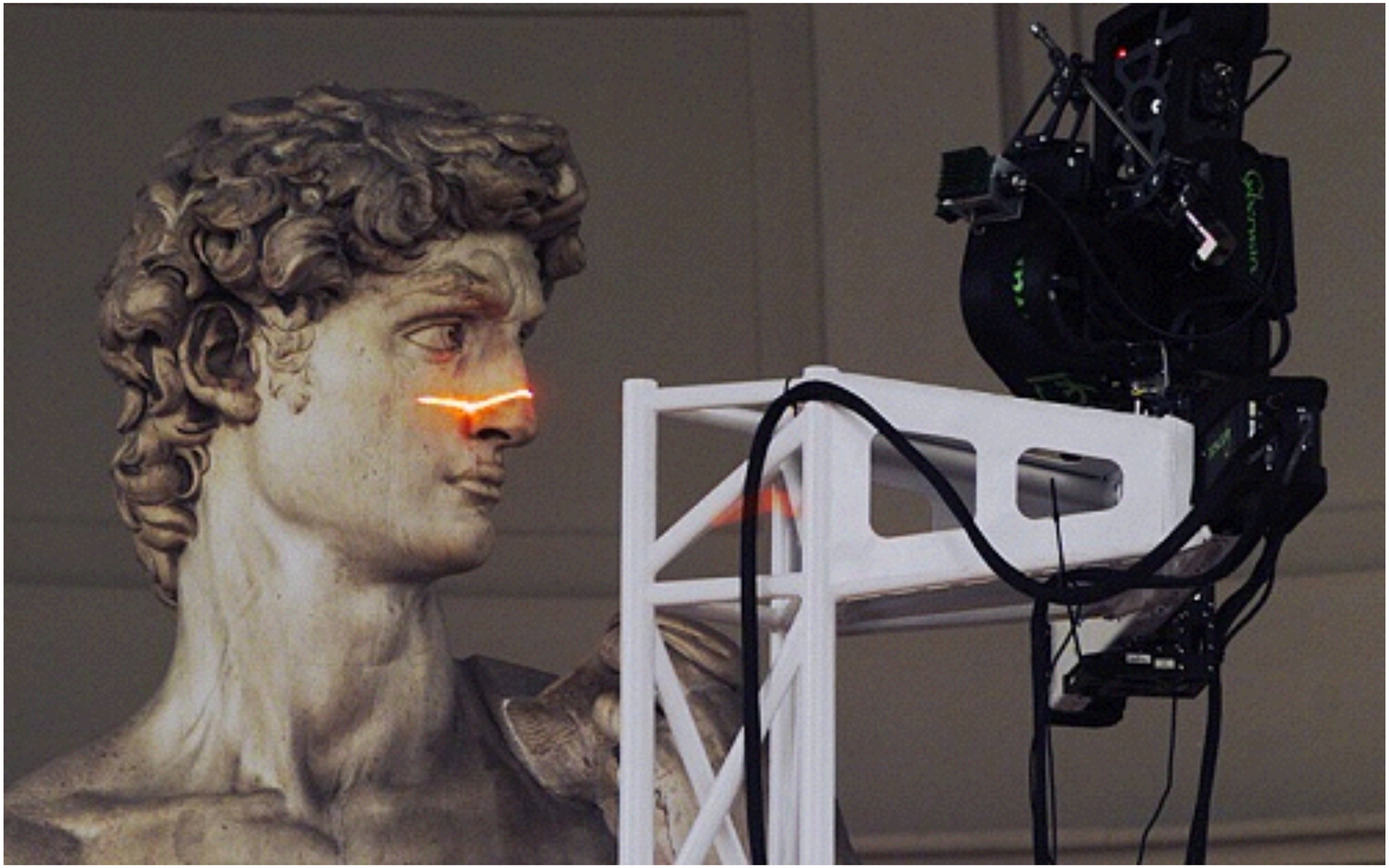

ZBrush

TURBOSQUID

Digimation

# Motivation

Large and growing repositories of 3D Models



**Laser Scans**

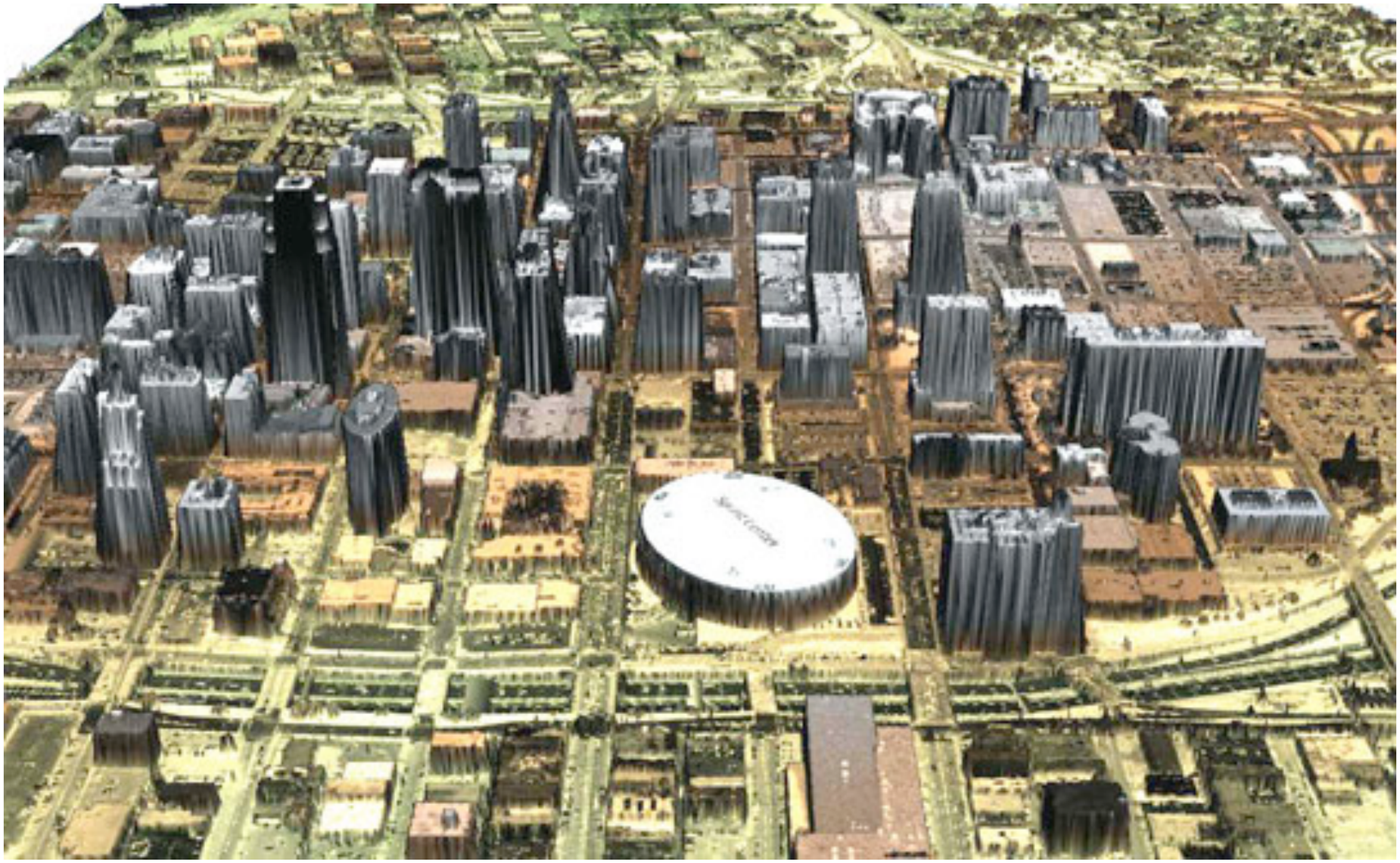
# Motivation

Large and growing repositories of 3D Models

**Laser Scans**

# Motivation

Large and growing repositories of 3D Models



**Streetview Scans**

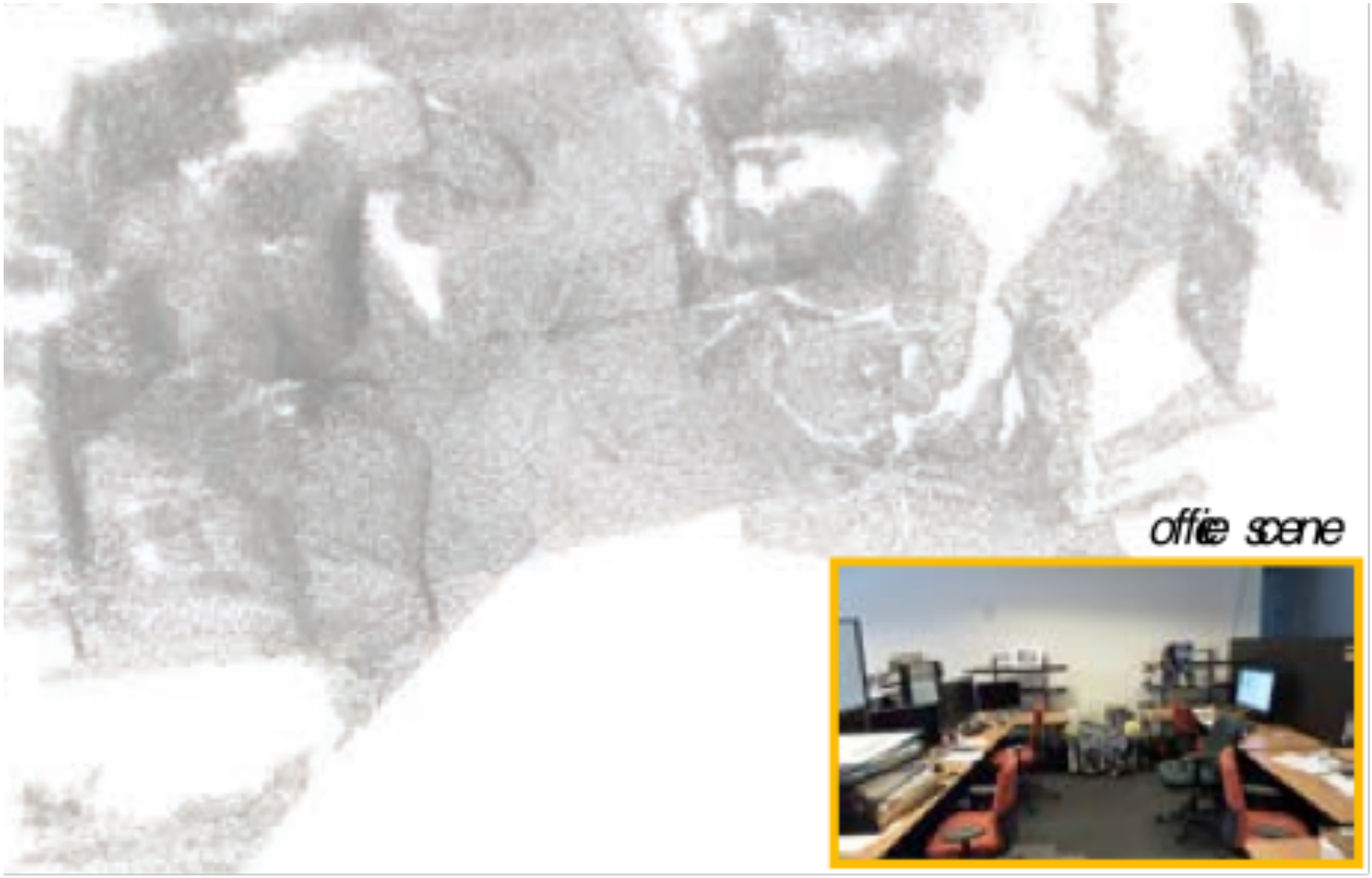
# Motivation

Large and growing repos

**Streetview Scans**

# Motivation

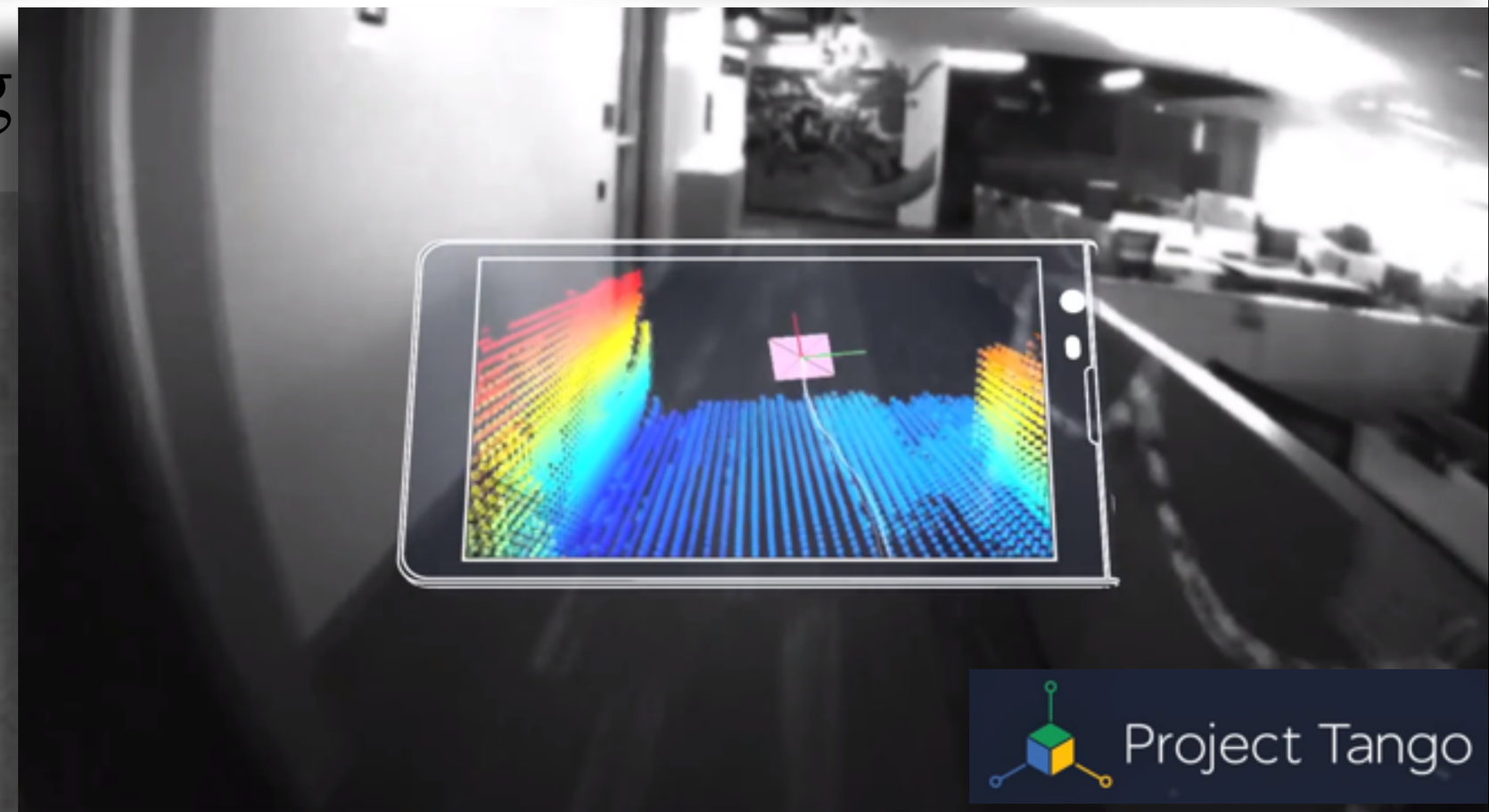
Large and growing repositories of 3D Models

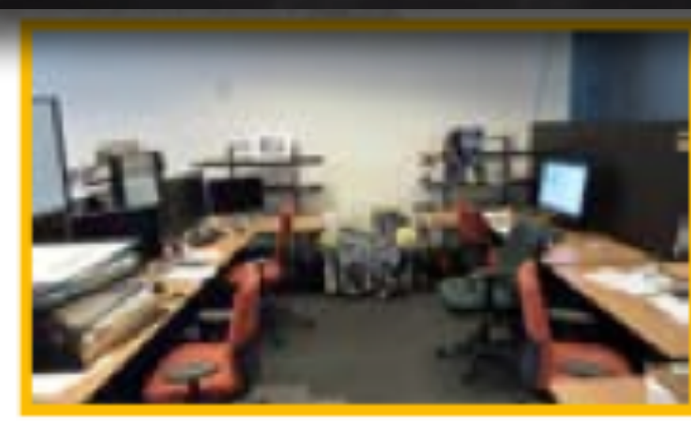



**RGB-D data**

# Motivation

Large and growing

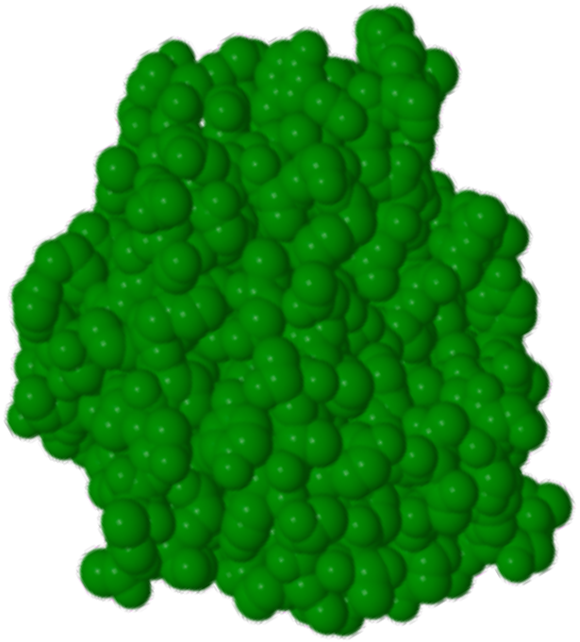


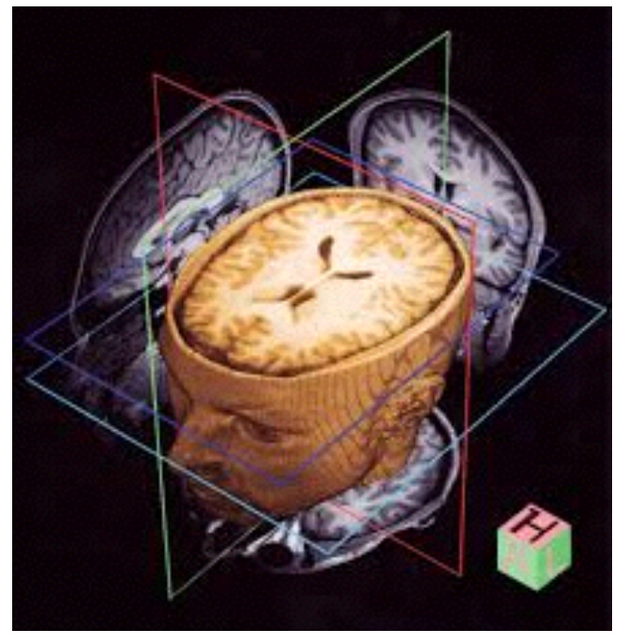
**RGB-D data**

# Motivation

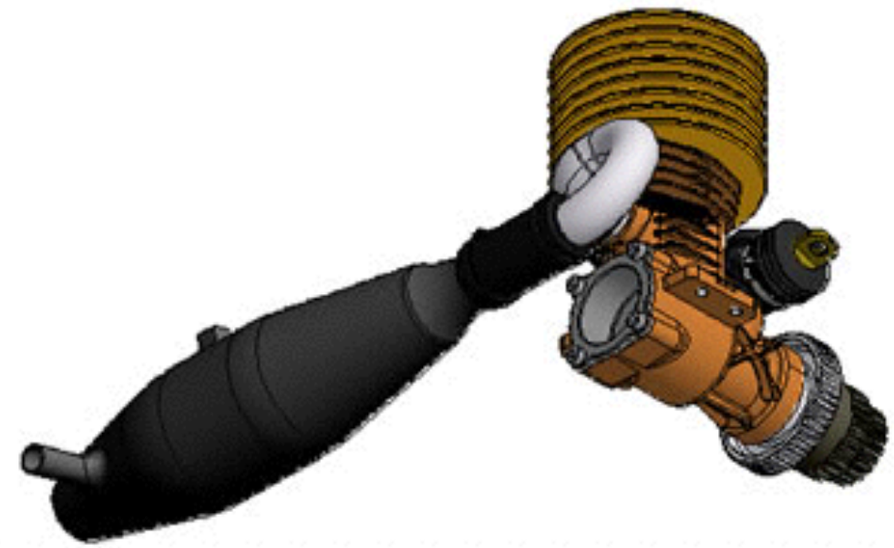
**<u>Lots</u>** of geometric data in different domains!



**Molecular Biology**

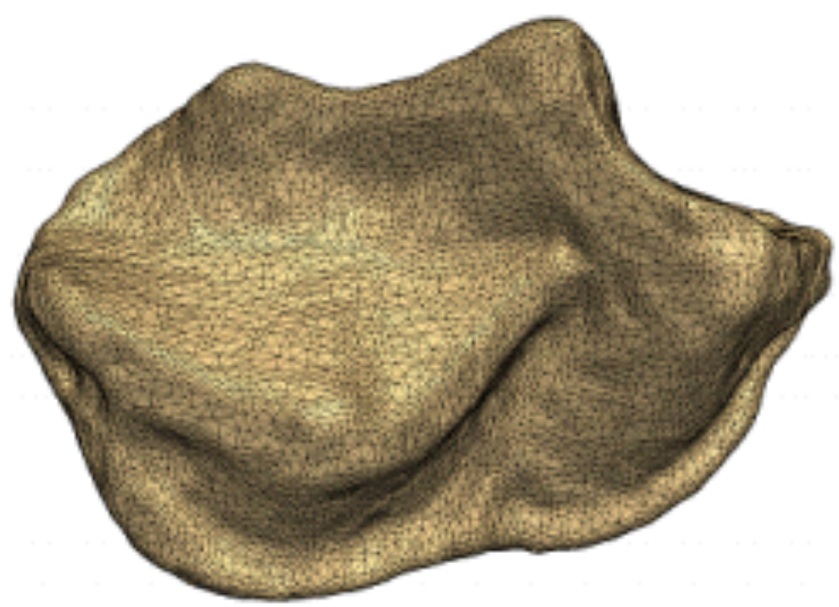


**Medicine**



**Engineering**

**Cultural Heritage**



**Paleontology**

**Computer Graphics**

# Introduction

We need algorithms to:

- Organize Geometric Data
- Understand Relationships
- Represent Variations
- Interpret New Shapes

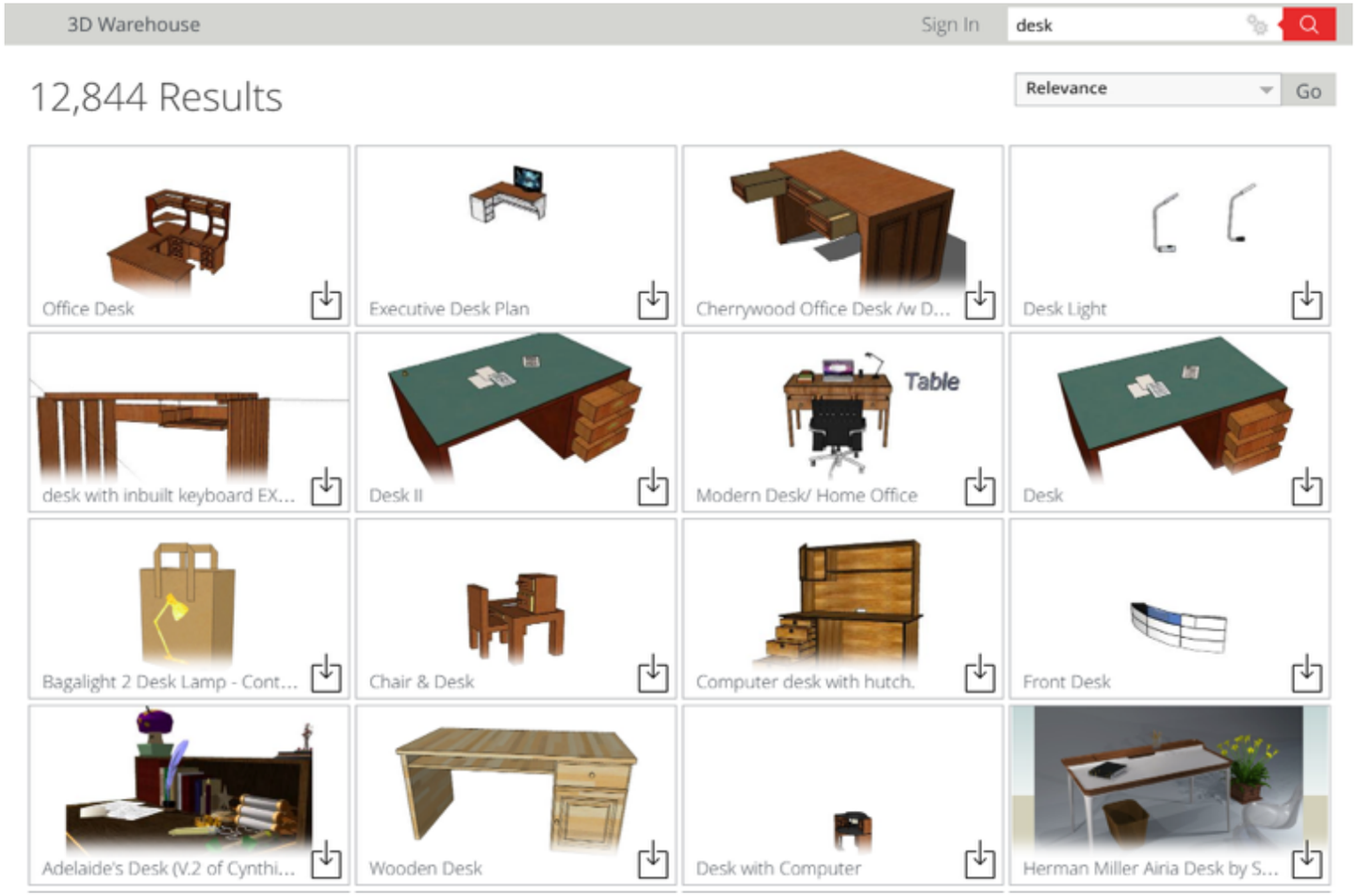
# Introduction

We need algorithms to:

➡ **Organize Geometric Data**

- Shape similarity
- Browse, retrieve, and explore

- Understand Relationships
- Represent Variations
- Interpret New Shapes

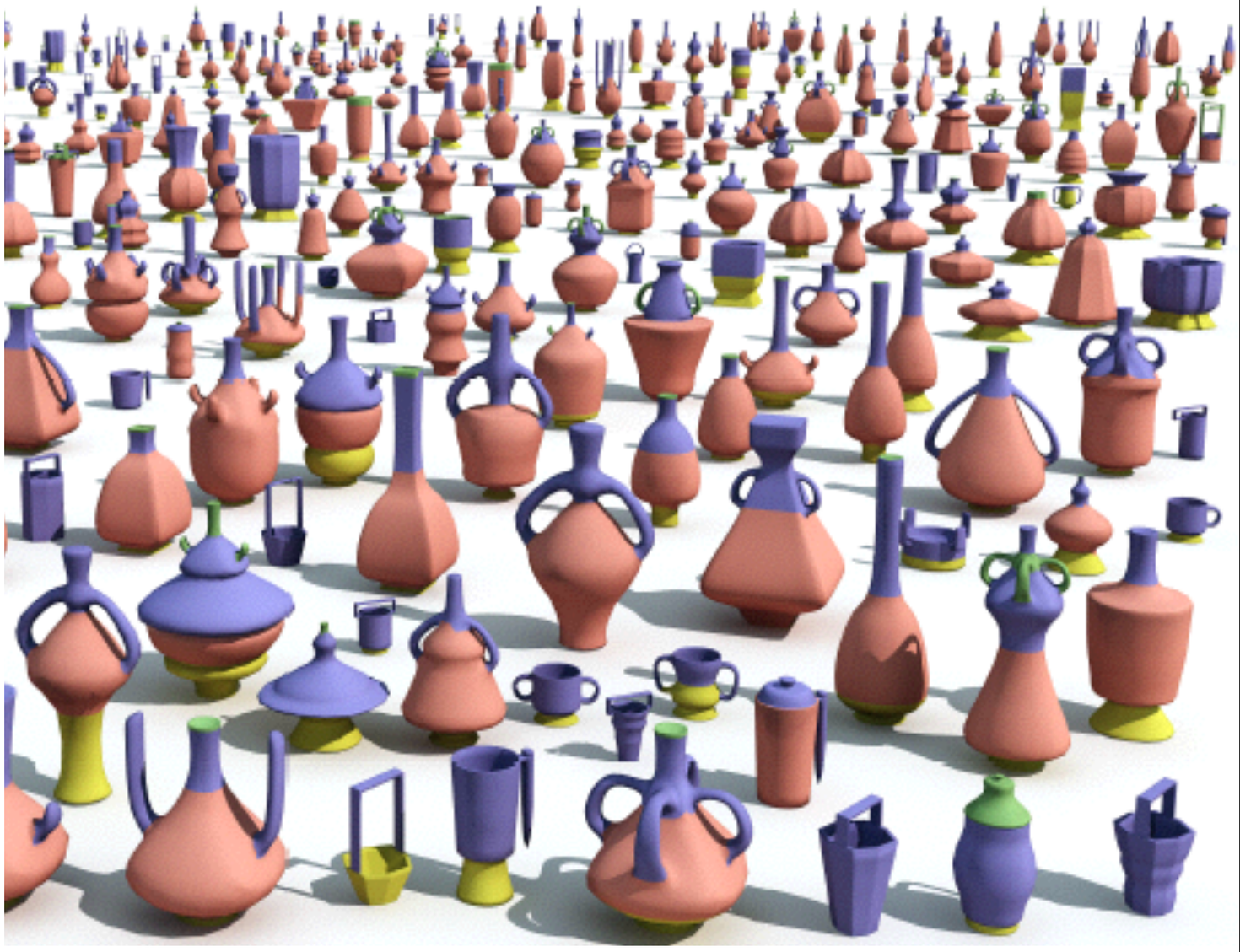



*Trimble 3D Warehouse*

# Introduction

We need algorithms to:

- Organize Geometric Data
- ➡ **Understand Relationships**
  - Identify similar regions
  - Find recurring patterns
- Represent Variations
- Interpret New Shapes

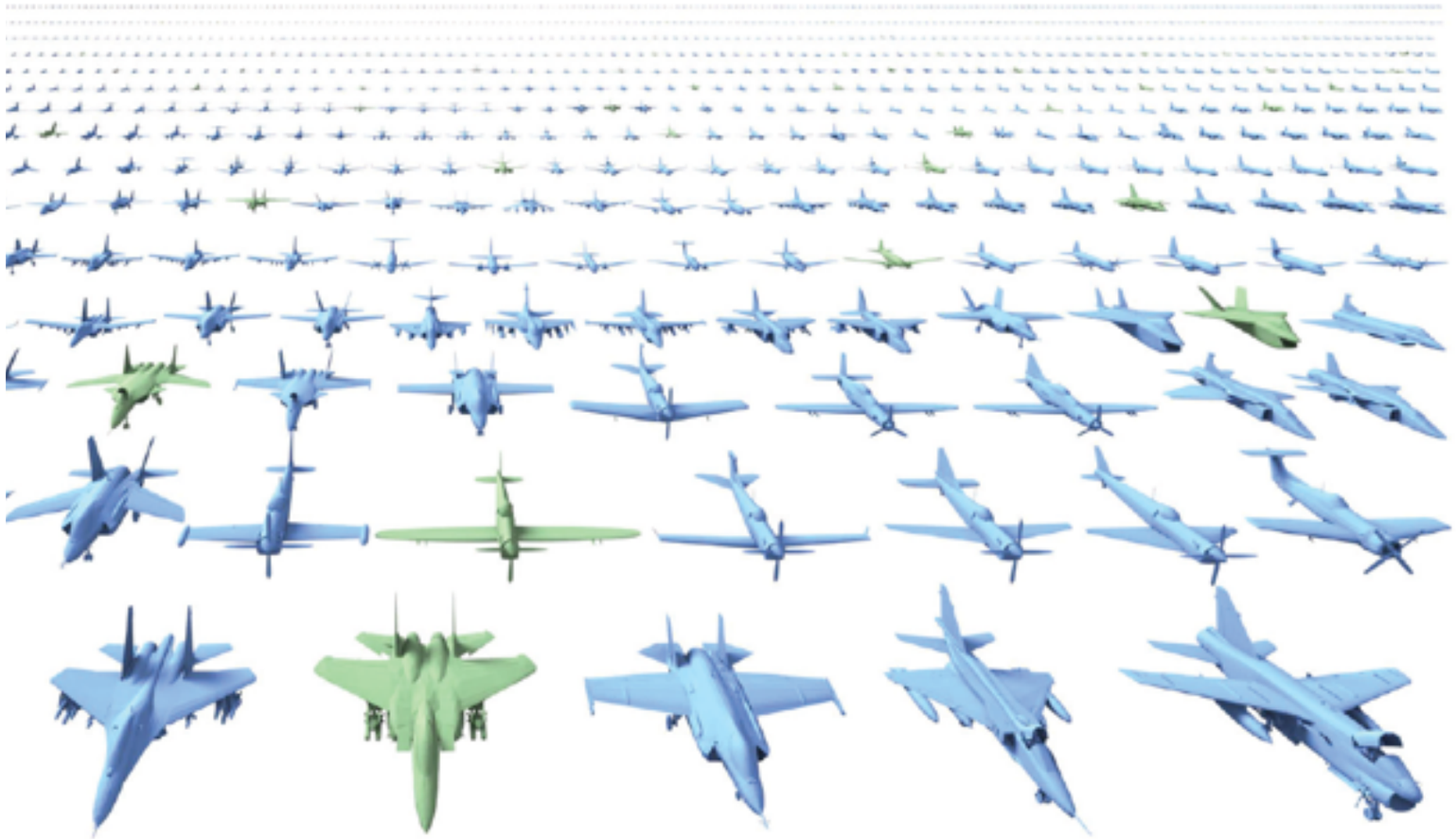


*Wang et al. SIGGRAPH Asia'12*

# Introduction

We need algorithms to:

- Organize Geometric Data
- Understand Relationships
- ➡ **Represent Variations**
  - Probabilistic models
  - Shape grammars
- Interpret New Shapes



*Kalogerakis et al. SIGGRAPH'12*

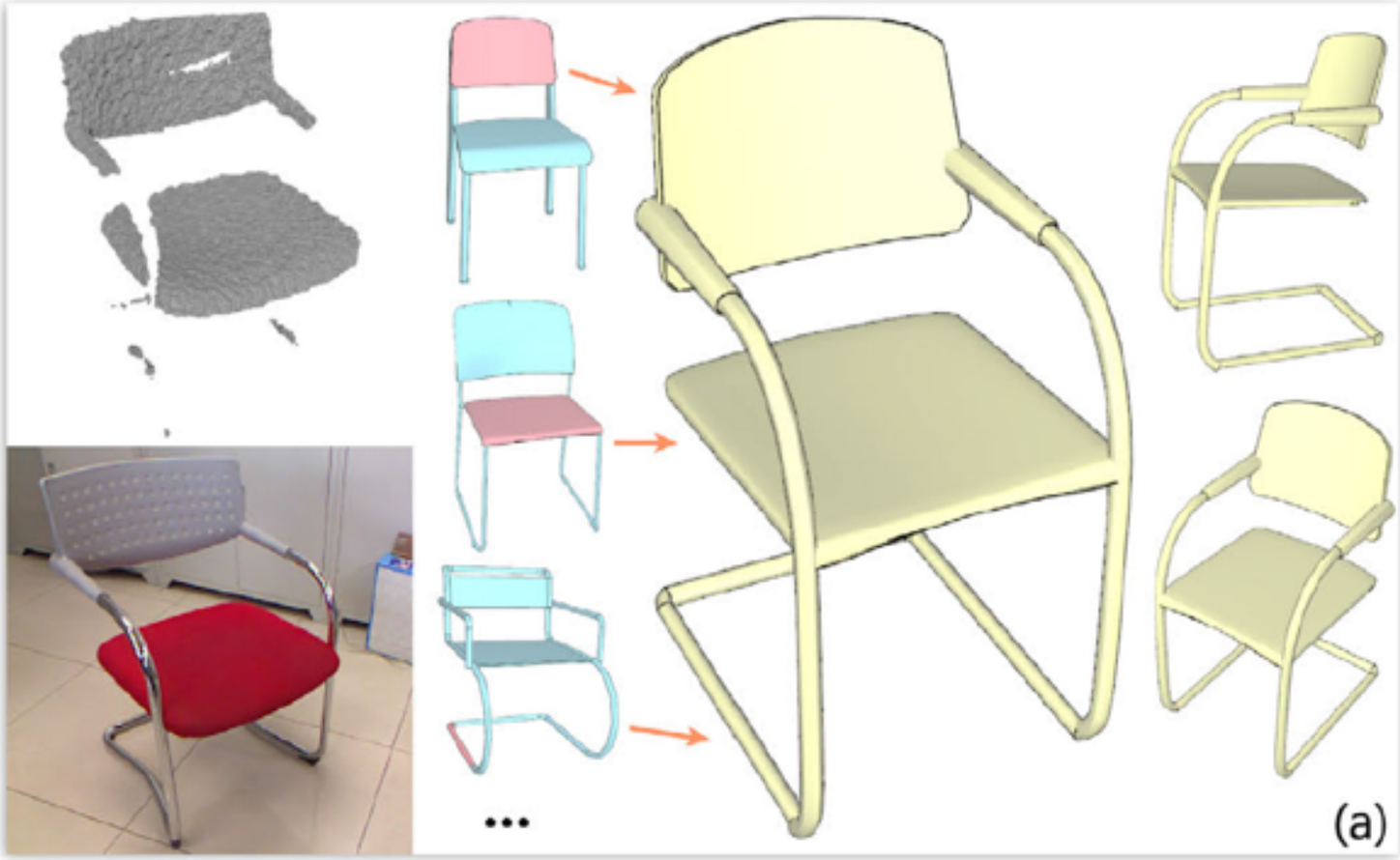
# Introduction

We need algorithms to:

- Organize Geometric Data
- Understand Relationships
- Represent Variations

➡ **Interpret New Shapes**

- Recognition
- Reconstruction




*Shen et al. SIGGRAPH Asia'12*

# Today's Lecture

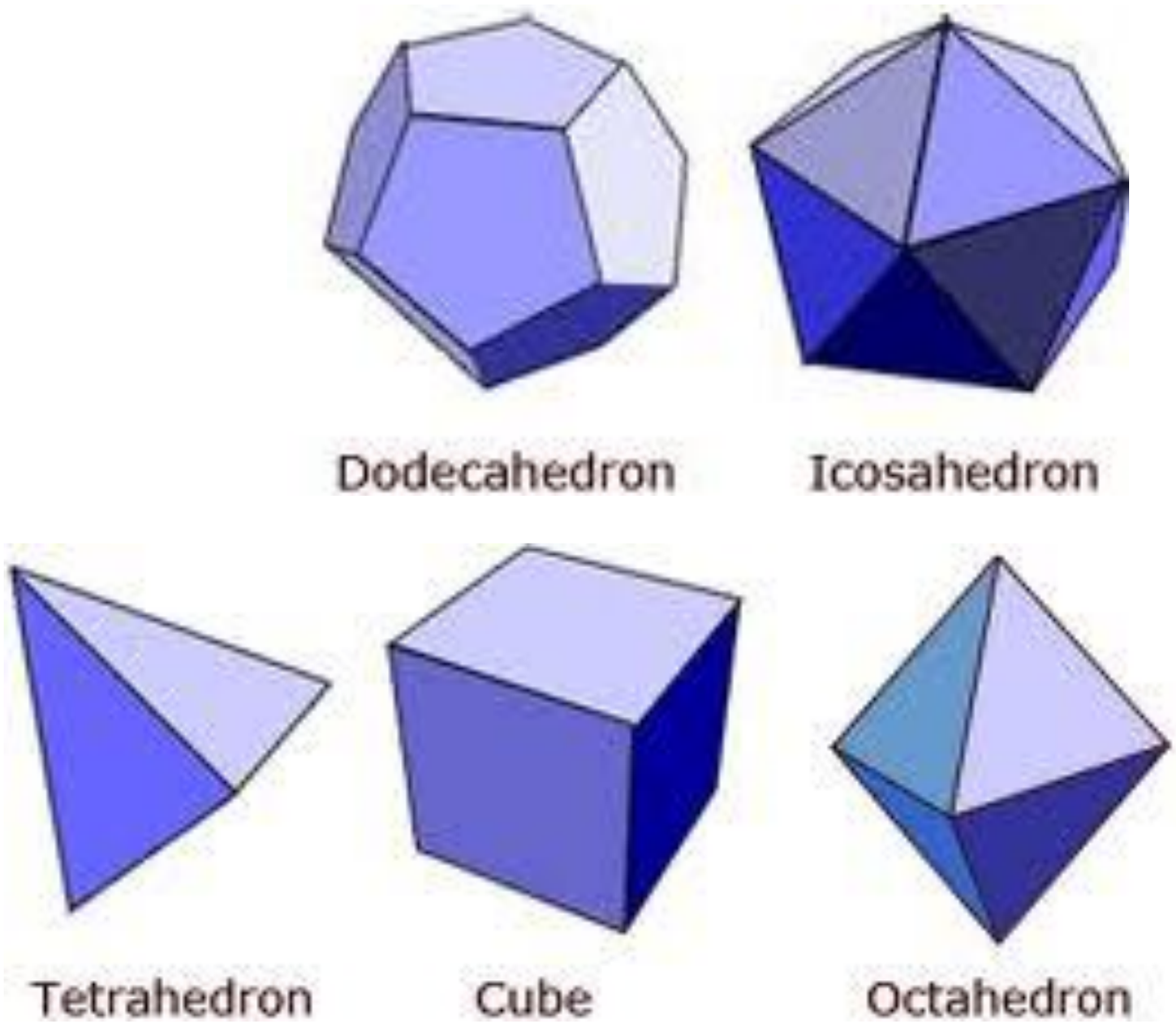
Rest of today

➡ **Basics: meshes & point clouds**

- Shape Descriptors

# Polygonal Mesh

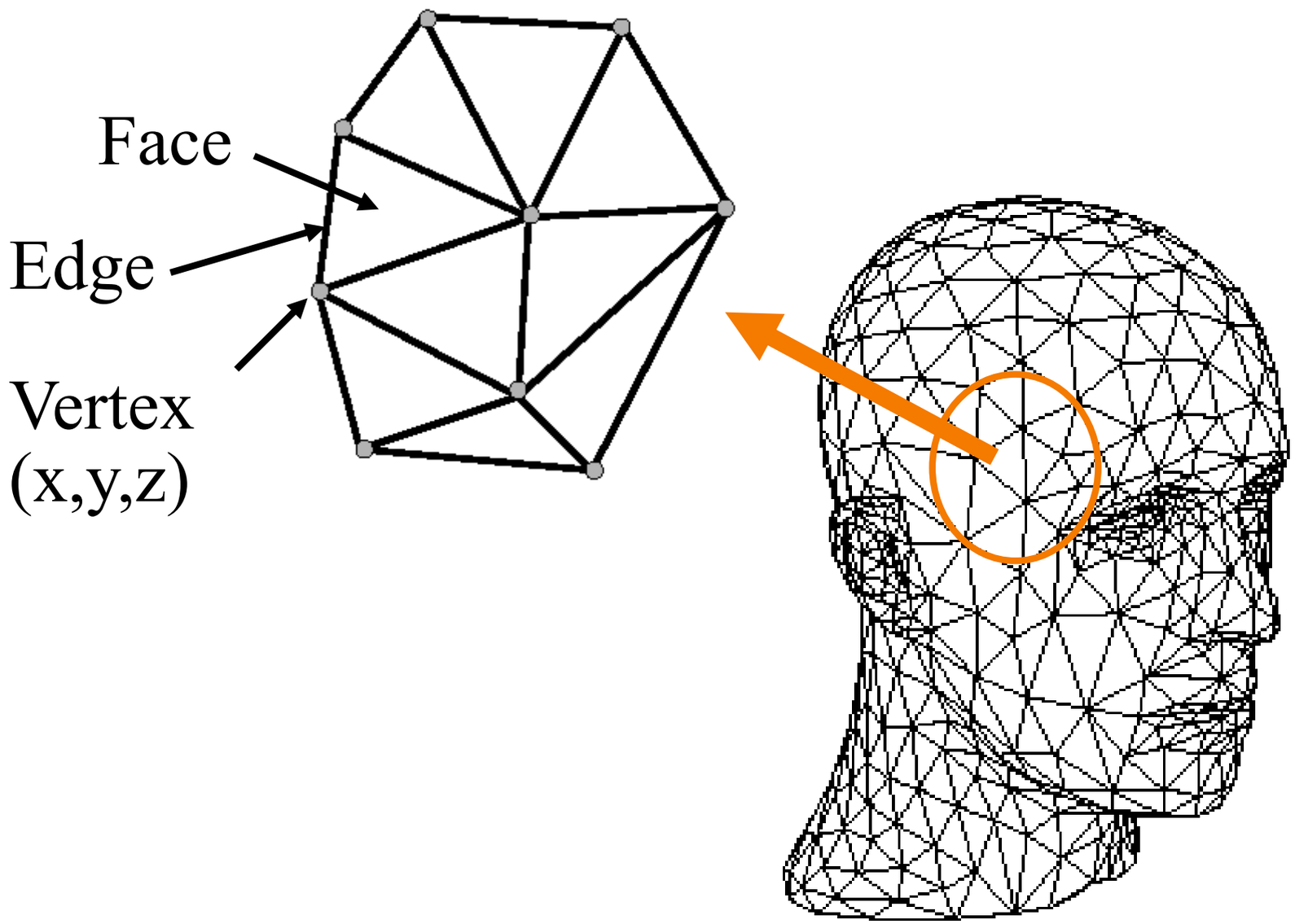
Set of polygons representing a 2D surface embedded in 3D




*Funkhouser*

# Polygonal Mesh

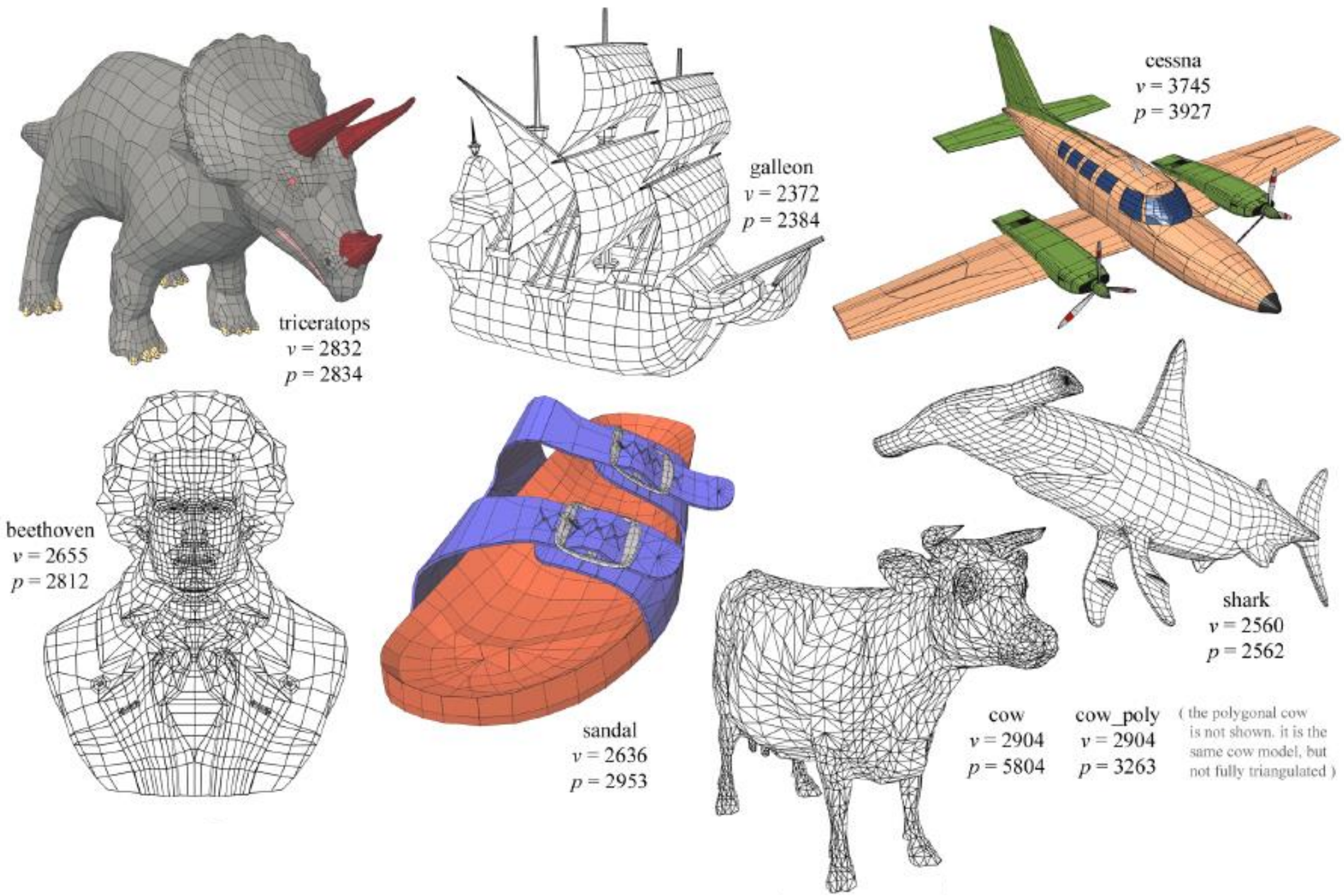
Set of polygons representing a 2D surface embedded in 3D




Zorin & Schroeder

# Polygonal Mesh

Set of polygons representing a 2D surface embedded in 3D

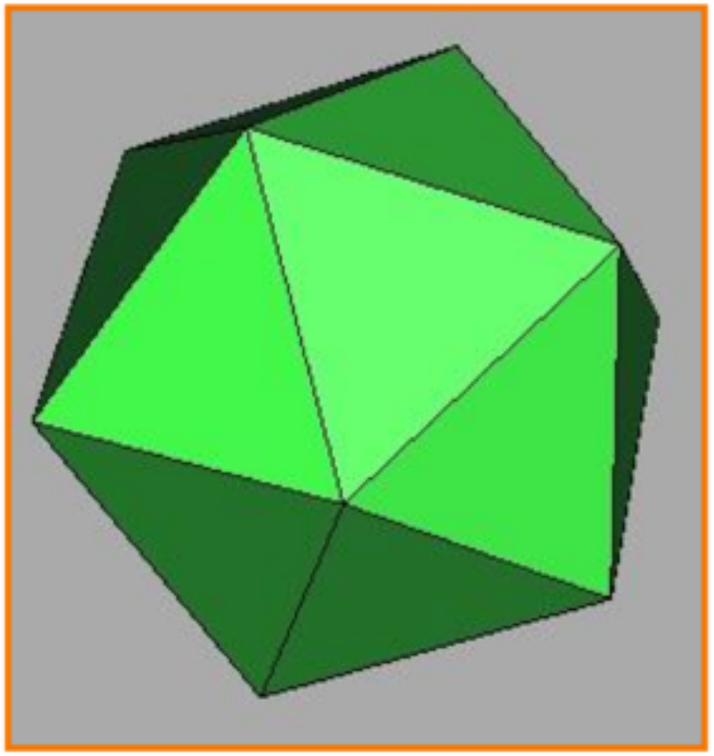



Isenberg

# Mesh Representation

Datastructures

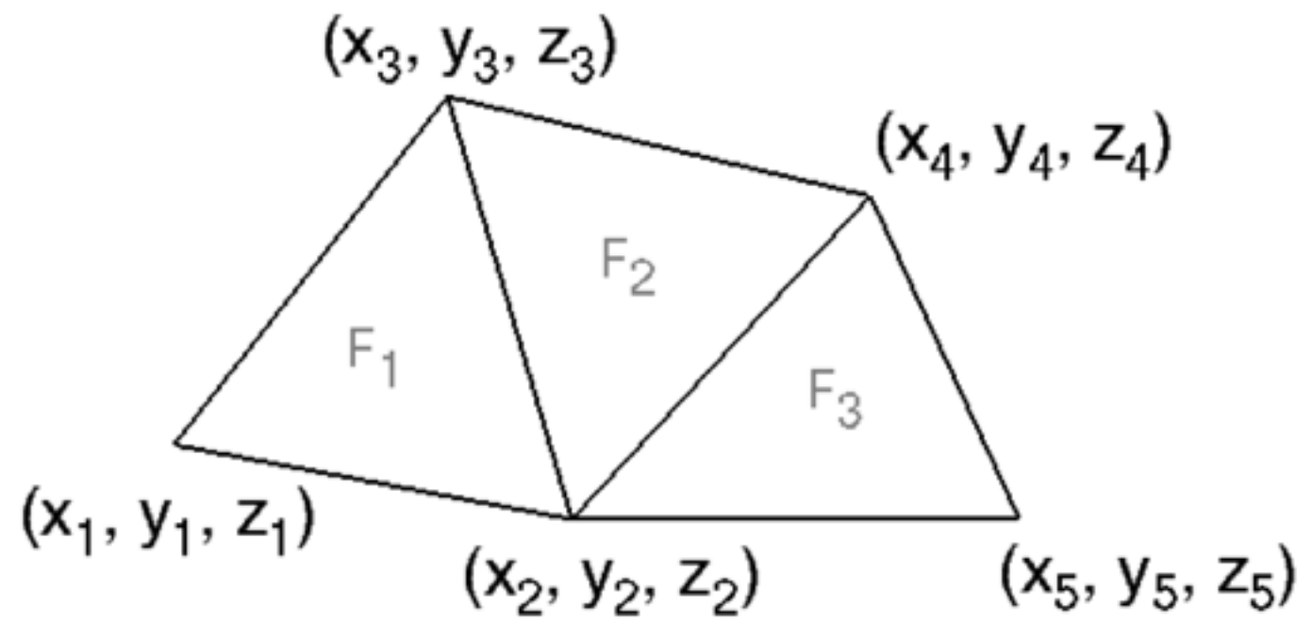
- List of faces
- Vertex and face tables
- Adjacency lists
- Partial adjacency lists

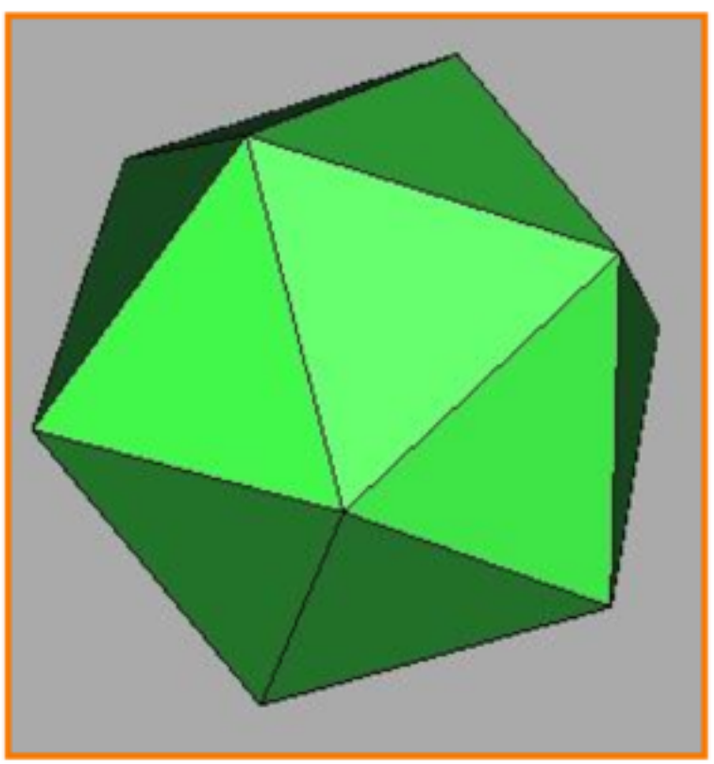


# Mesh Representation

Datastructures

- ➡ **List of faces**
- Vertex and face tables
- Adjacency lists
- Partial adjacency lists



| FACE TABLE | |
|---|---|
| $F_1$ | $(x_1, y_1, z_1)$ $(x_2, y_2, z_2)$ $(x_3, y_3, z_3)$ |
| $F_2$ | $(x_2, y_2, z_2)$ $(x_4, y_4, z_4)$ $(x_3, y_3, z_3)$ |
| $F_3$ | $(x_2, y_2, z_2)$ $(x_5, y_5, z_5)$ $(x_4, y_4, z_4)$ |

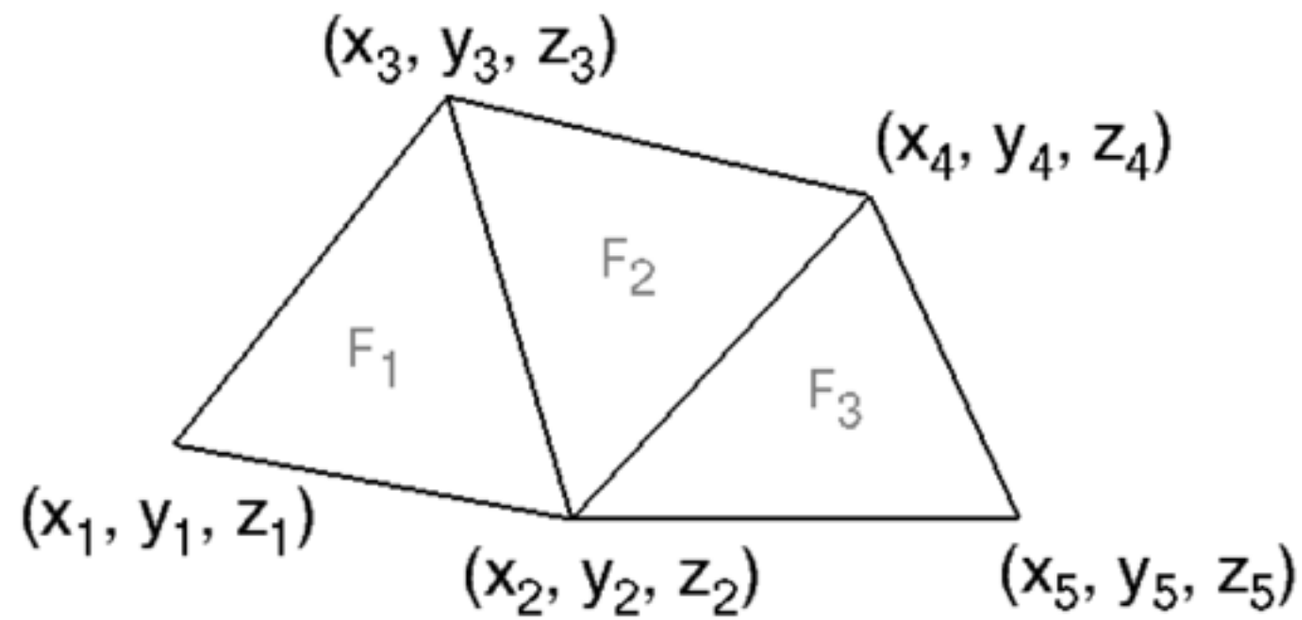
**No Adjacency Info**

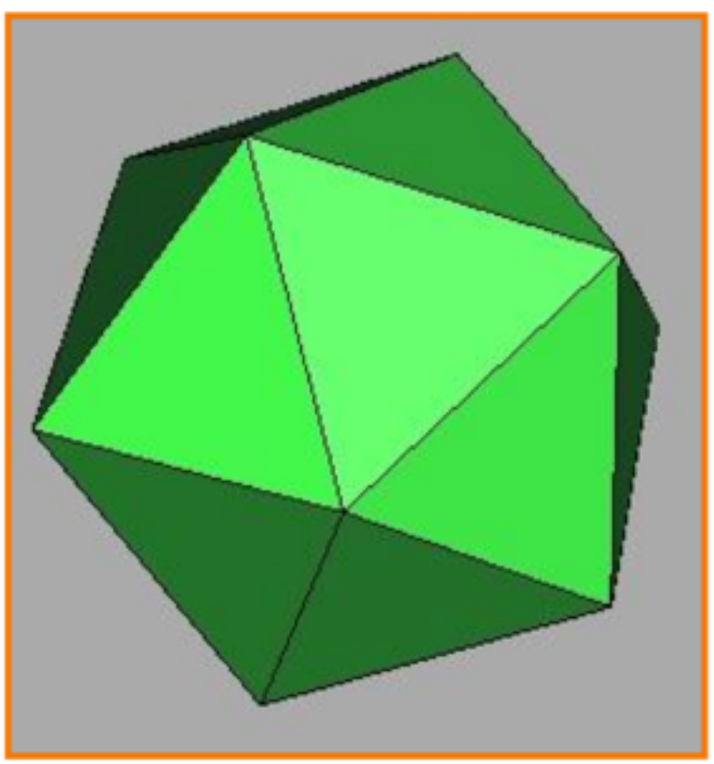
*Funkhouser*

# Mesh Representation

Datastructures

- List of faces
- ➡ **Vertex and face tables**
- Adjacency lists
- Partial adjacency lists



| VERTEX TABLE | | | |
|---|---|---|---|
| $V_1$ | $X_1$ | $Y_1$ | $Z_1$ |
| $V_2$ | $X_2$ | $Y_2$ | $Z_2$ |
| $V_3$ | $X_3$ | $Y_3$ | $Z_3$ |
| $V_4$ | $X_4$ | $Y_4$ | $Z_4$ |
| $V_5$ | $X_5$ | $Y_5$ | $Z_5$ |

| FACE TABLE | | | |
|---|---|---|---|
| $F_1$ | $V_1$ | $V_2$ | $V_3$ |
| $F_2$ | $V_2$ | $V_4$ | $V_3$ |
| $F_3$ | $V_2$ | $V_5$ | $V_4$ |

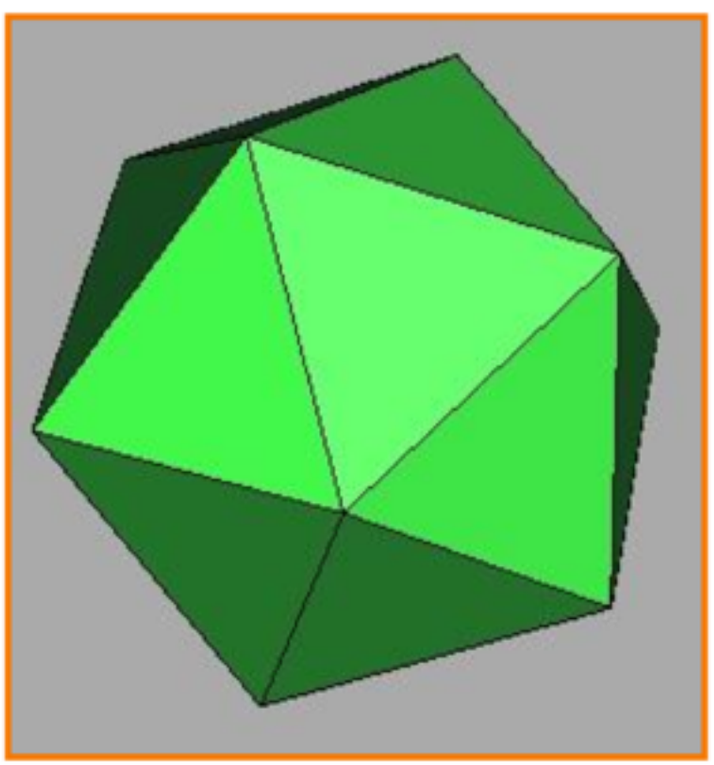
**Adjacency is expensive to compute**

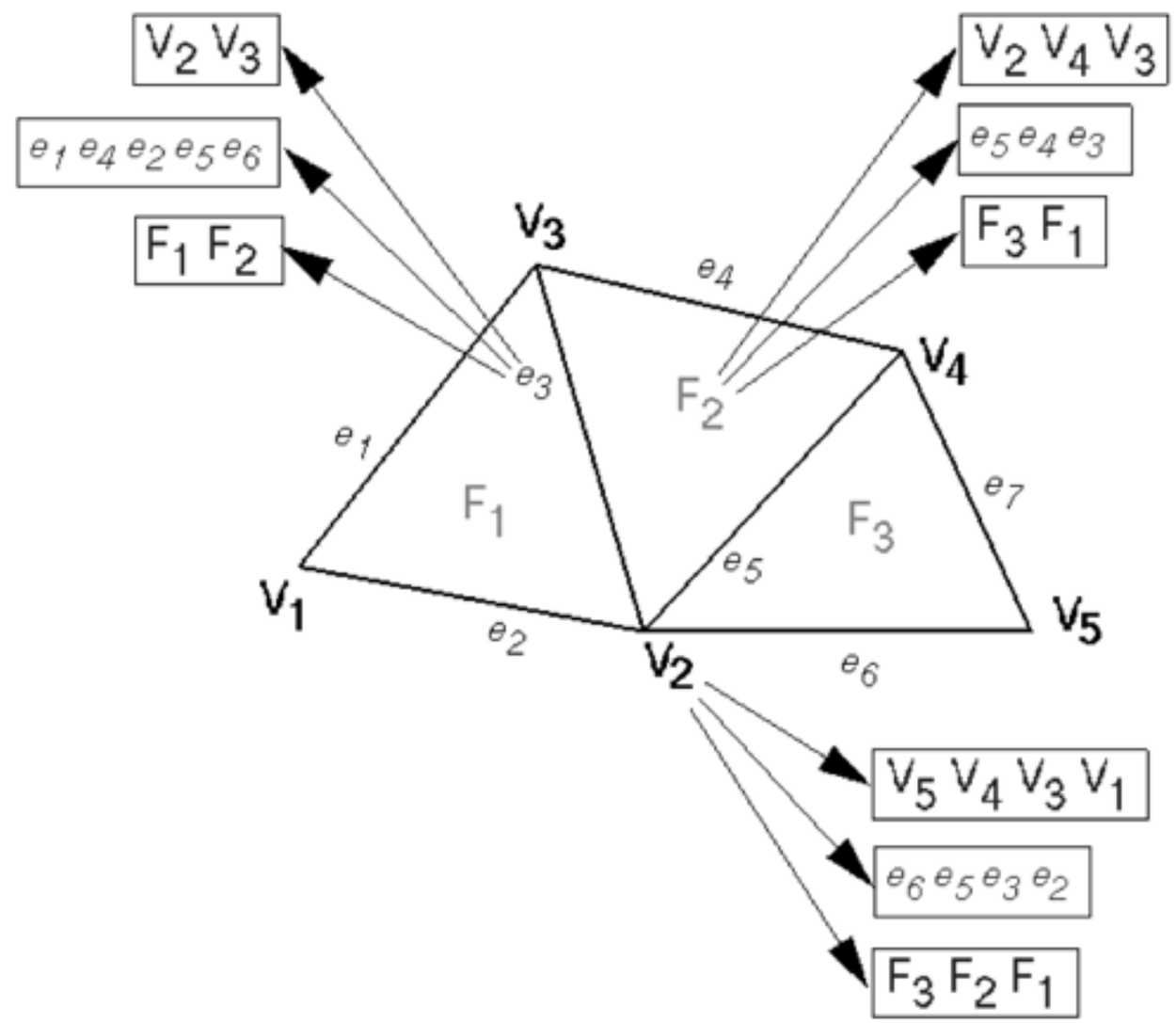
*Funkhouser*

# Mesh Representation

Datastructures

- List of faces
- Vertex and face tables
- ➡ **Adjacency lists**
- Partial adjacency lists





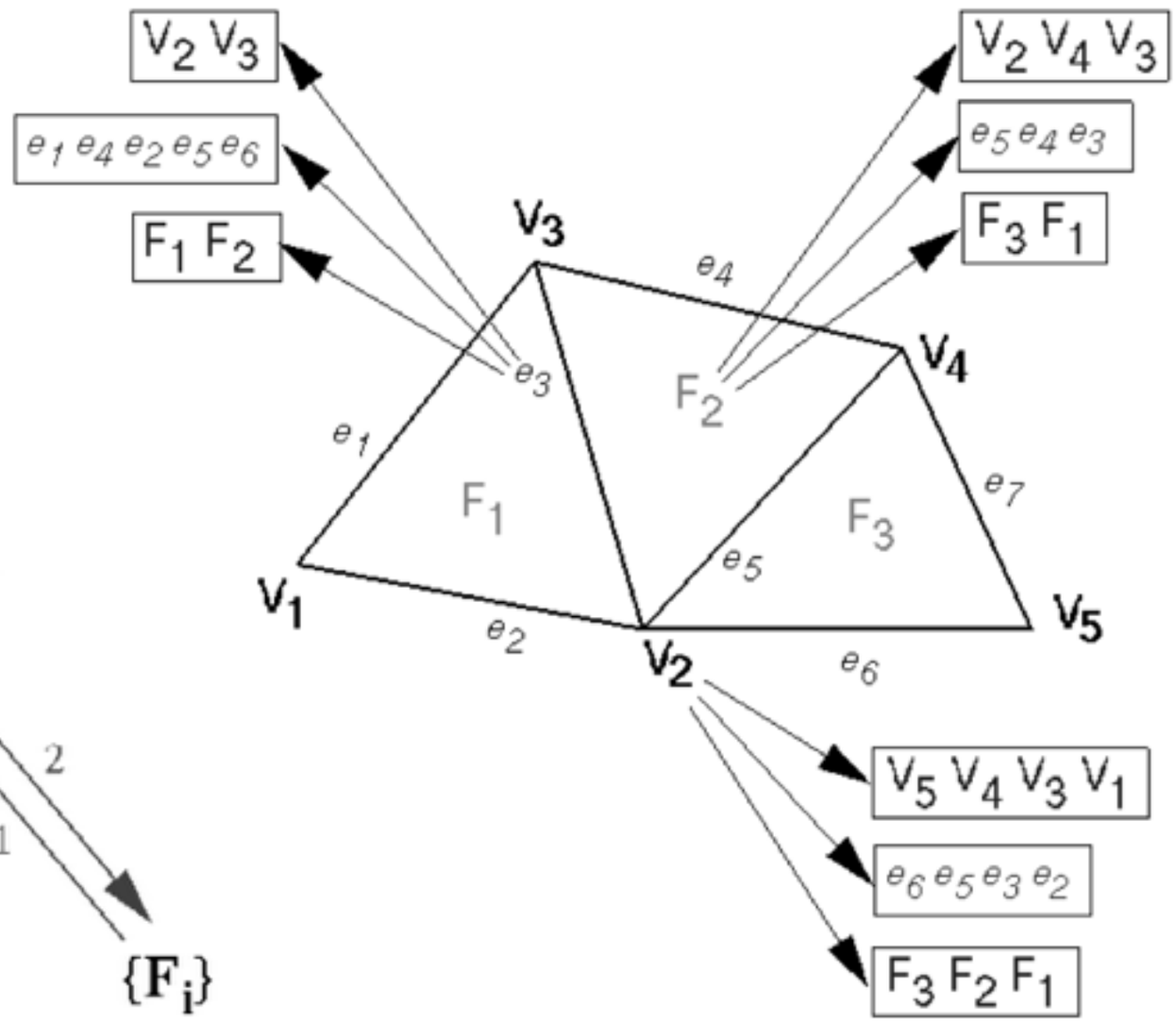

**Extra storage**

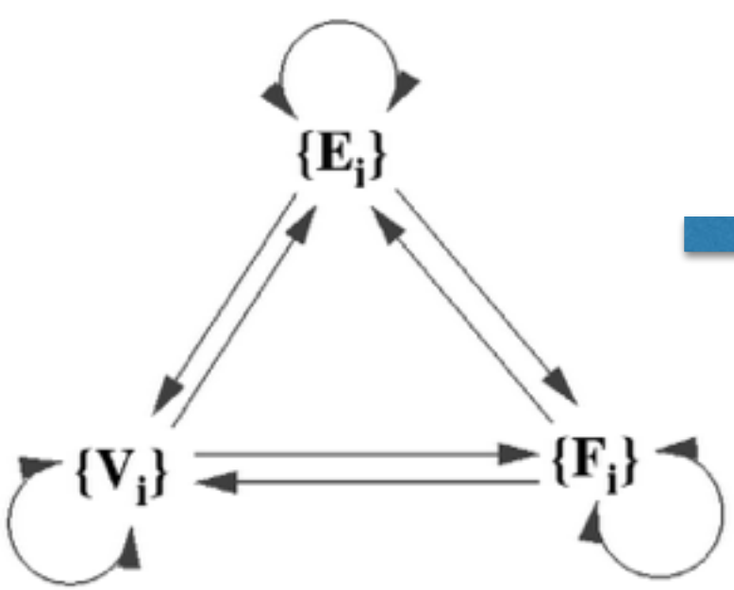
*Funkhouser*

# Mesh Representation

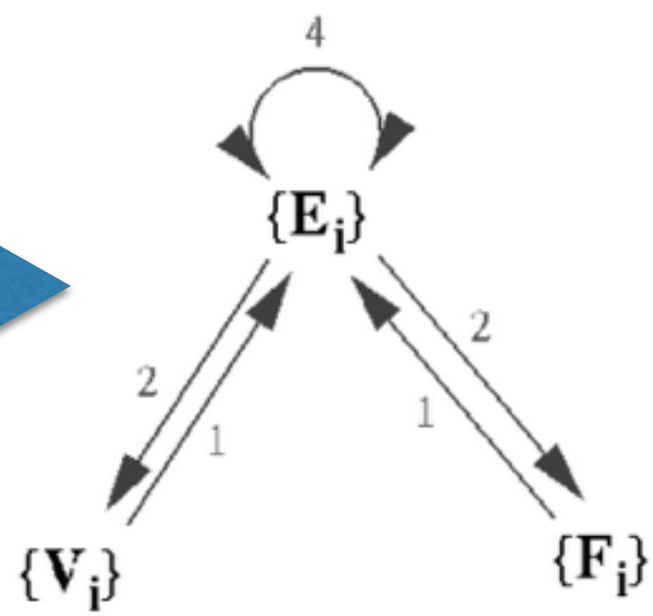
Datastructures

- List of faces
- Vertex and face tables
- Adjacency lists
- ➡ **Partial adjacency lists**




**All adjacency info**



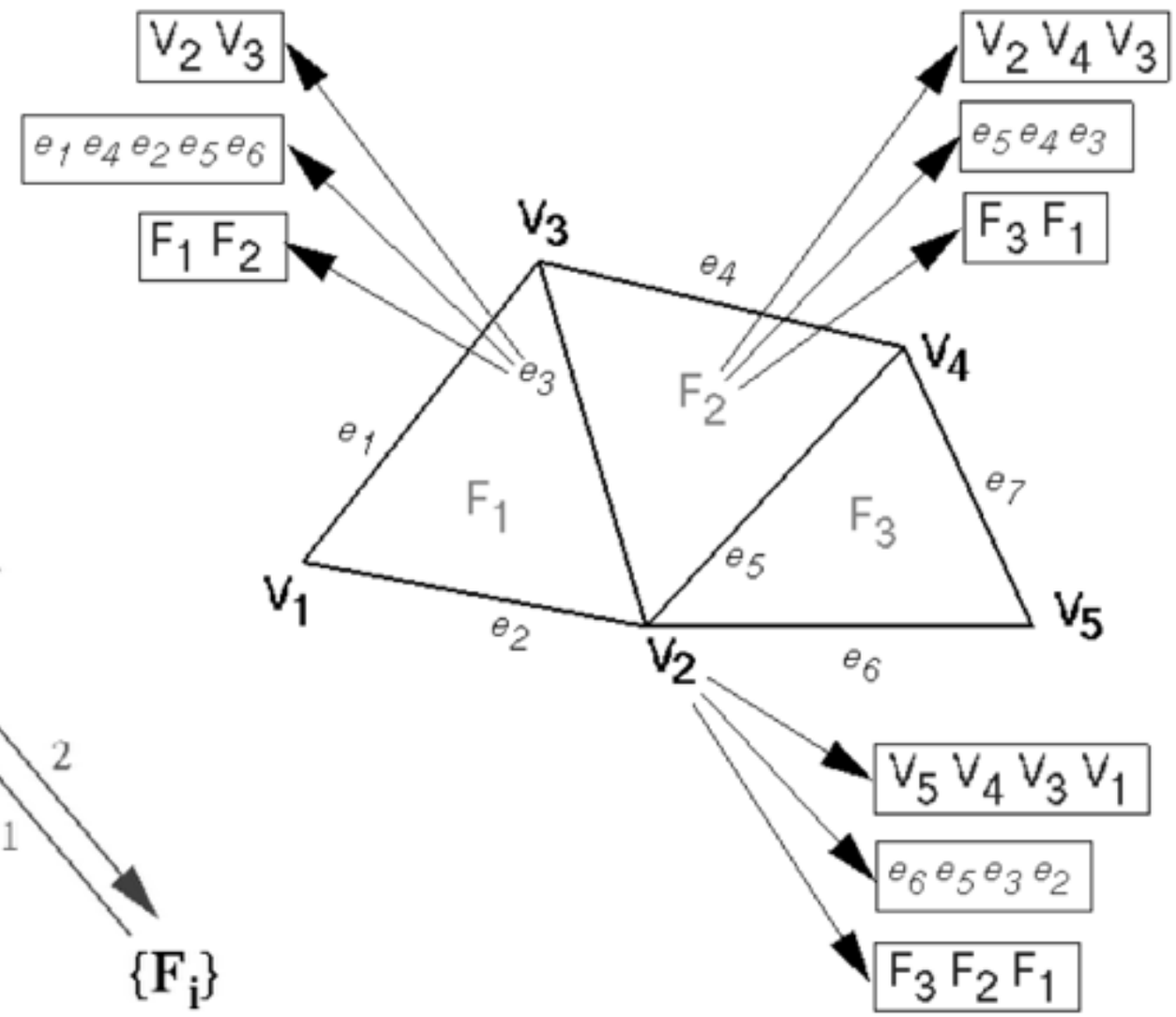

**Winged edge**

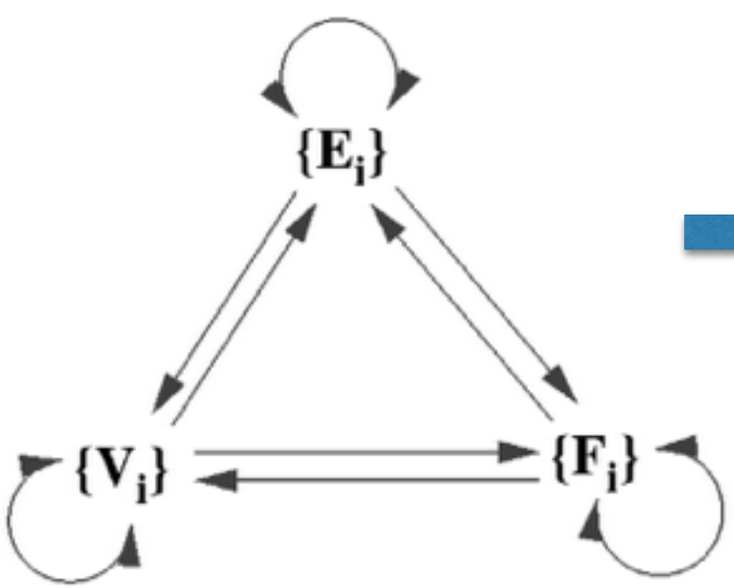
*Funkhouser*

# Mesh Representation

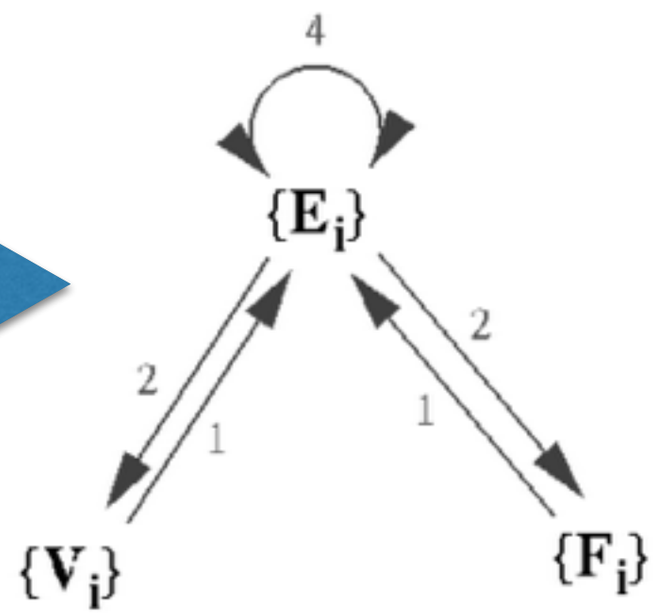
Datastructures

- List of faces
- Vertex and face tables
- Adjacency lists
- ➡ **Partial adjacency lists**




**All adjacency info**

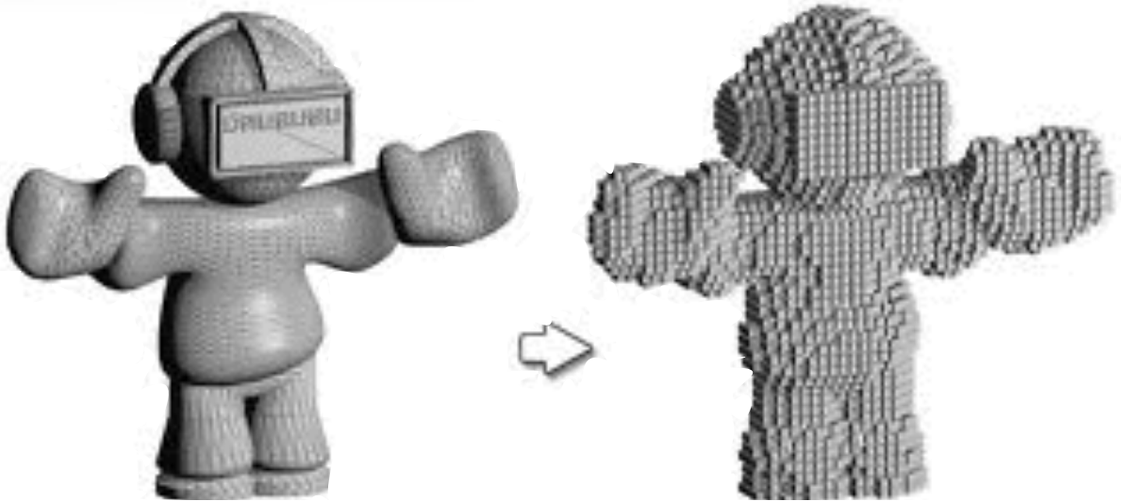



**Winged edge**

**Implement with caution**

*Funkhouser*

# Point Cloud & Voxel Grids



**Grid**

**Points**

# Point Cloud & Voxel Grids

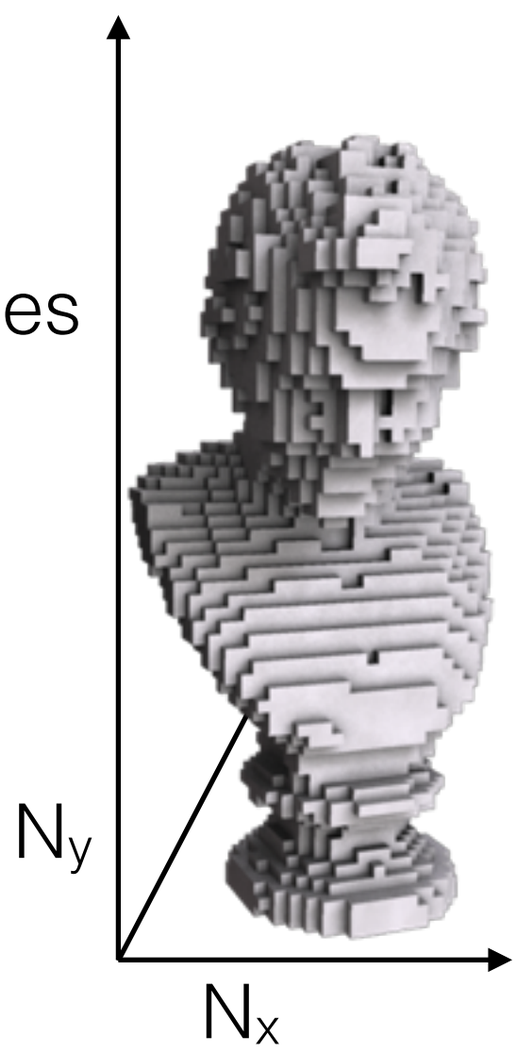
Datastructures

➡ **Voxel Grid**

- Point Cloud
- 3D k-d Tree

$N_x \times N_y \times N_z$ entries

**Voxel Grid**




**Expensive**

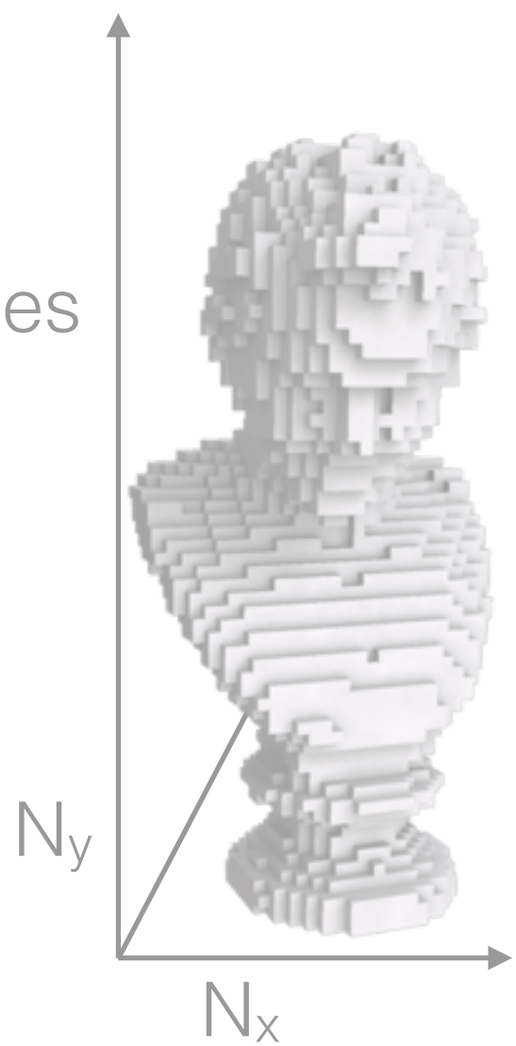
# Point Cloud & Voxel Grids

Datastructures

- Voxel Grid
- ➡ **Point Cloud**
- 3D k-d Tree

$N_x \times N_y \times N_z$ entries

**Voxel Grid**




**Expensive**

$\langle x_1, y_1, z_1 \rangle$
$\langle x_2, y_2, z_2 \rangle$
…
$\langle x_n, y_n, z_n \rangle$

**Point list**

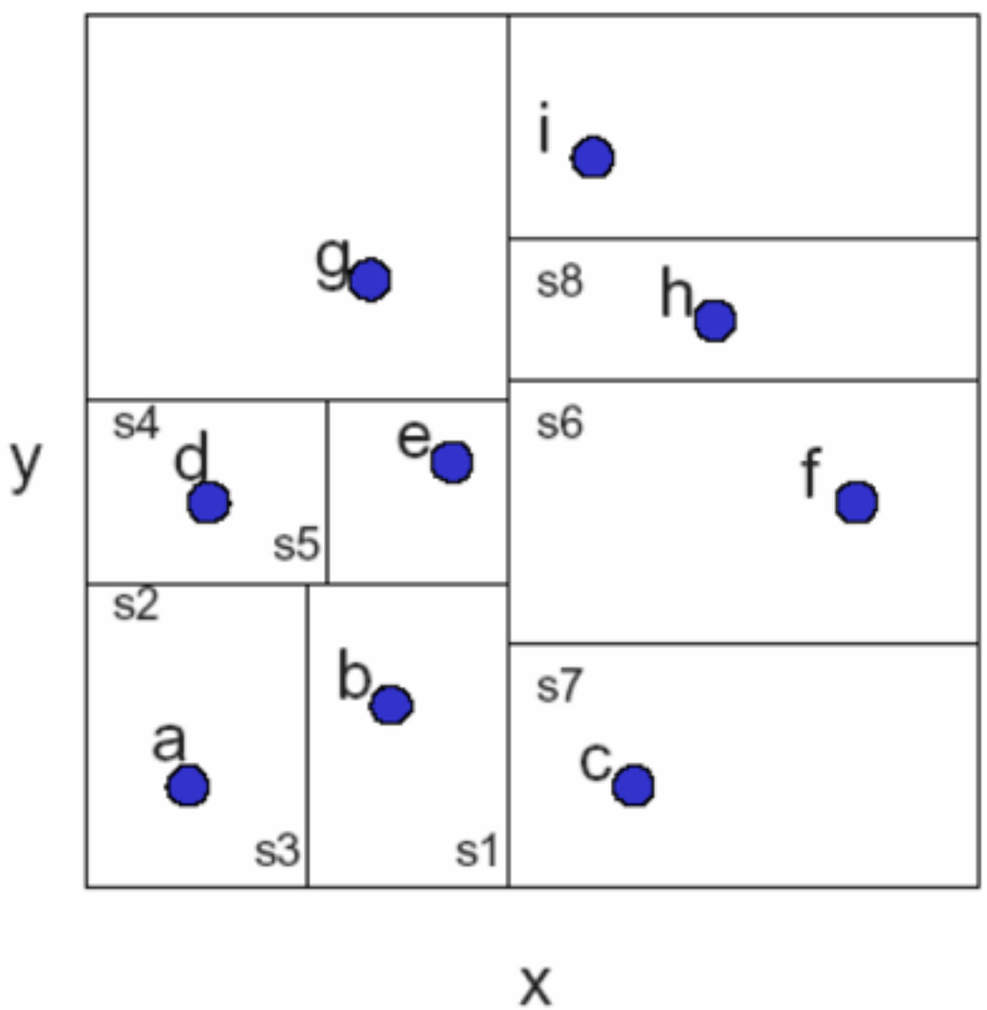
**No adjacency**

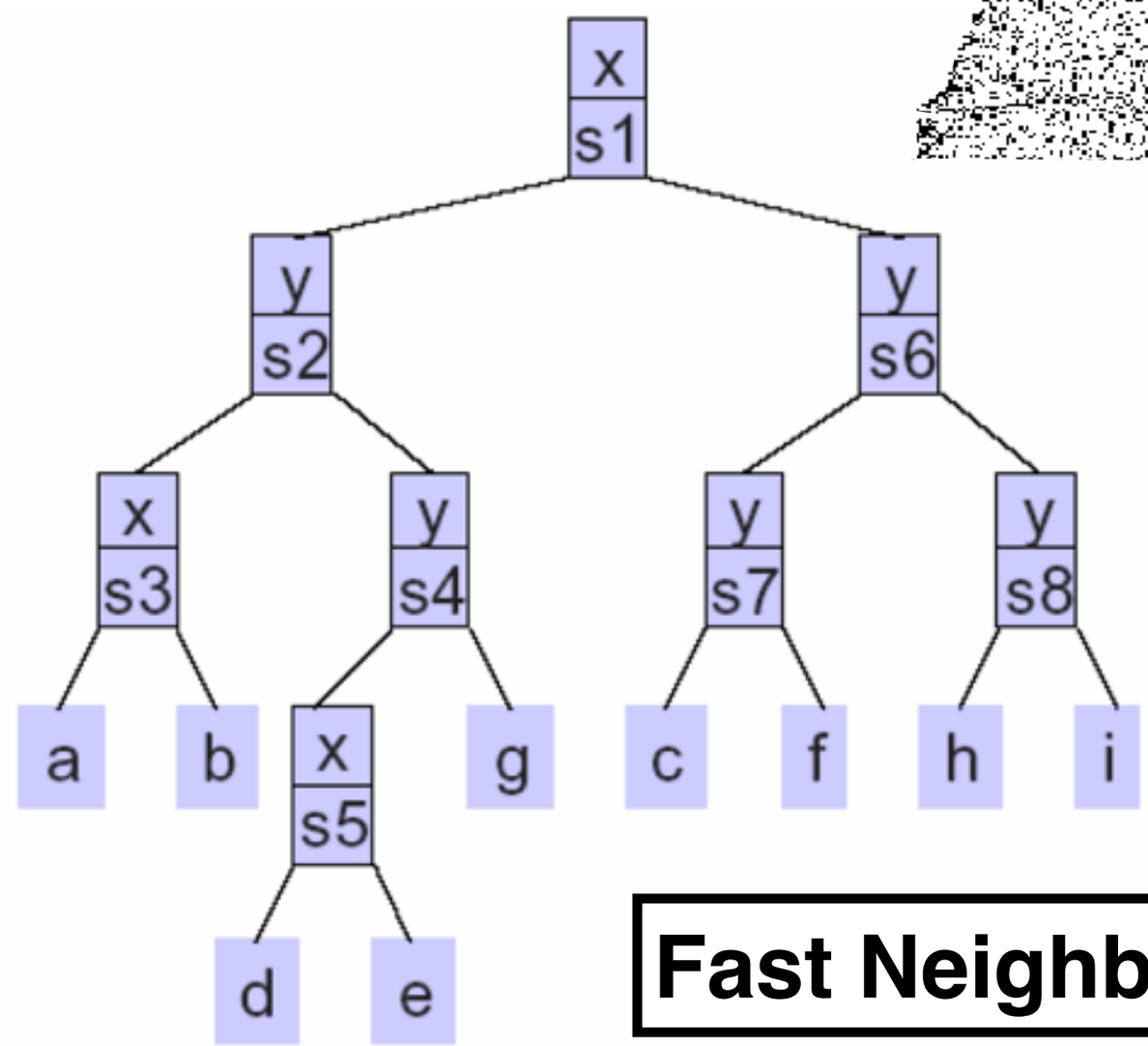
# Point Cloud & Voxel Grids

Datastructures

- Voxel Grid
- Point Cloud

➡ **3D k-d Tree**




**Fast Neighbors**

# Geometry Processing

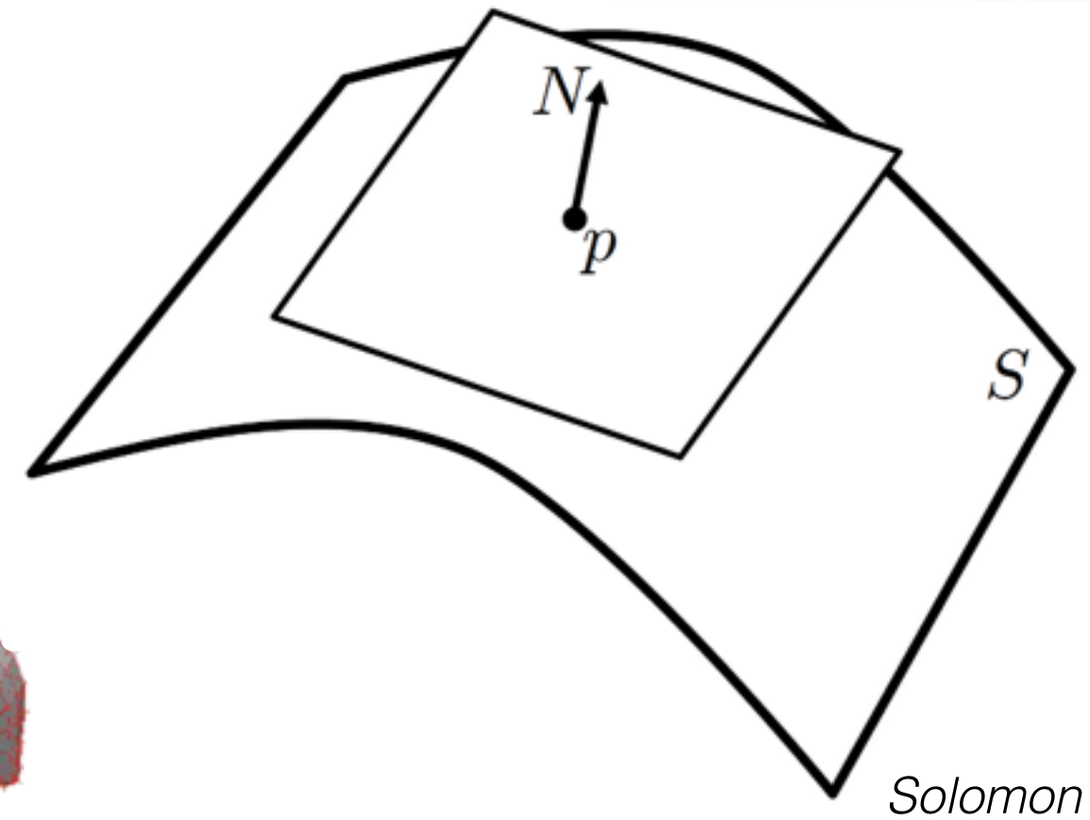
Analysis

➡ **Normals**

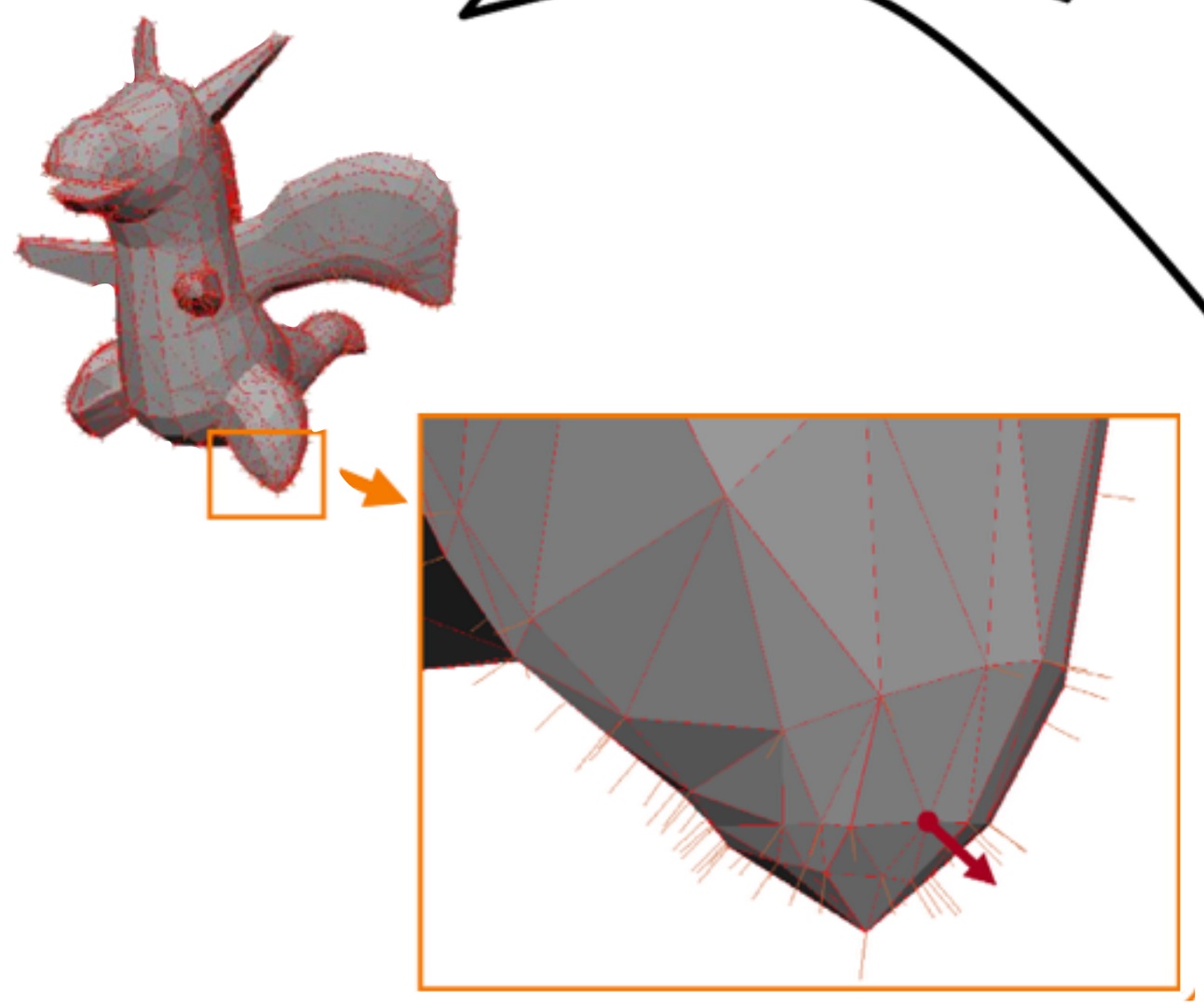
- Curvature

Warps

- Rigid x-form
- Deform




*Solomon*



# Geometry Processing

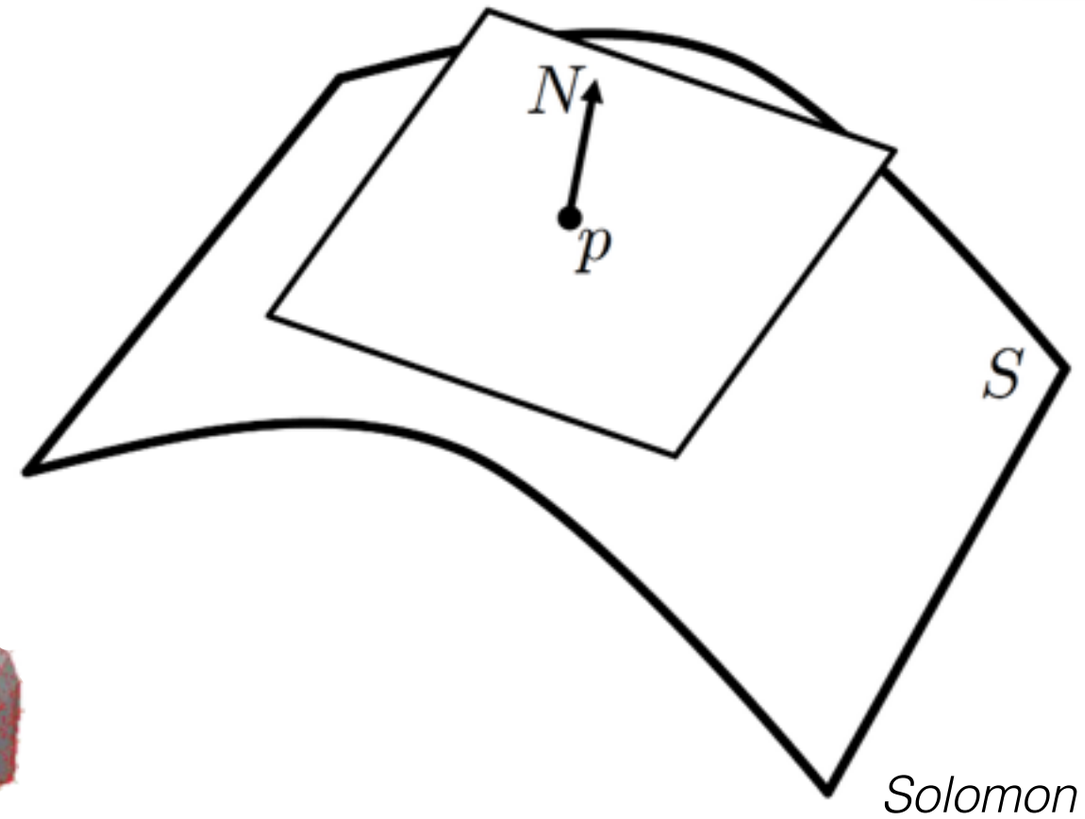
Analysis

➡ **Normals**

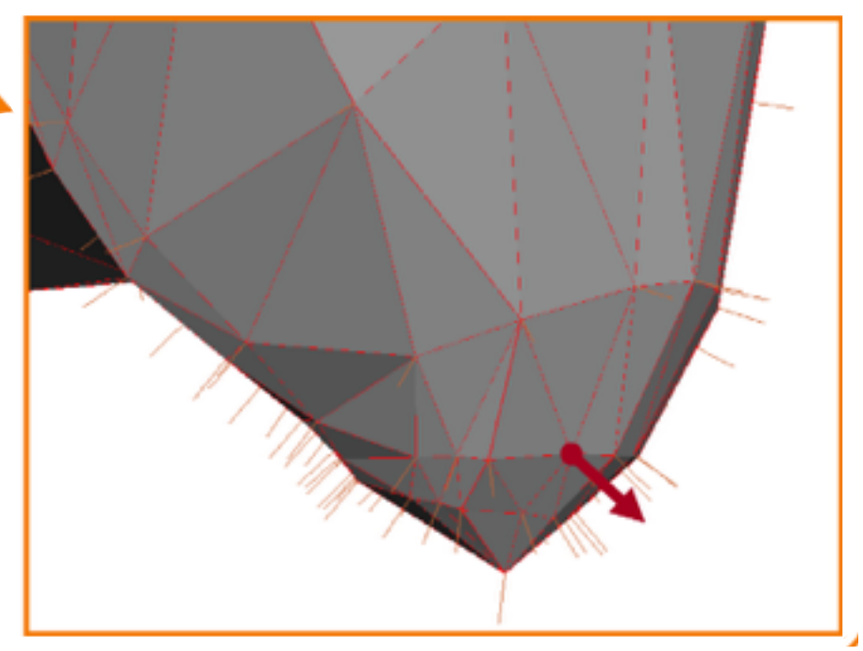
- Curvature

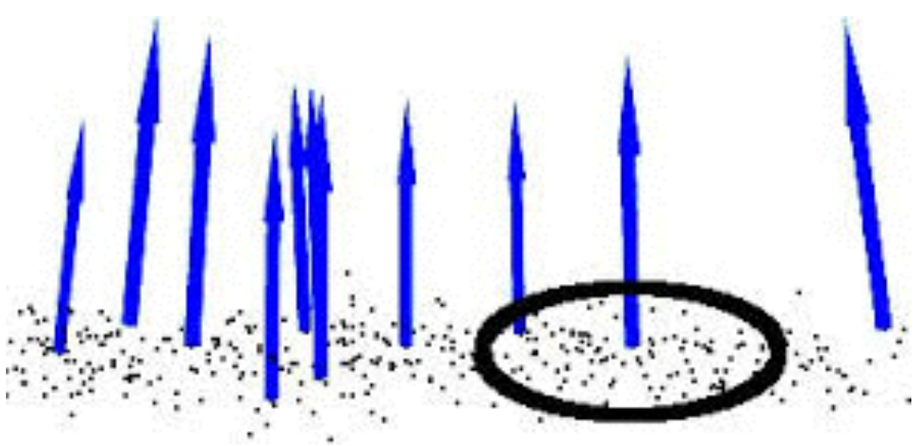
Warps

- Rigid x-form
- Deform




Solomon





**Also possible with point clouds**

# Geometry Processing

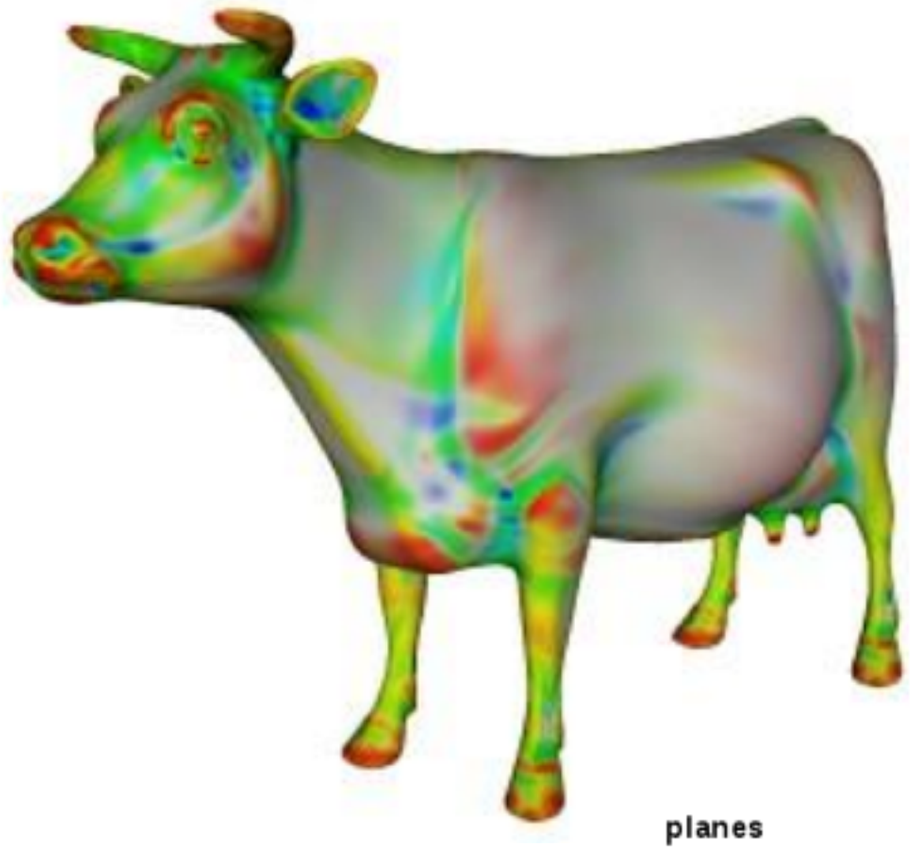
Analysis

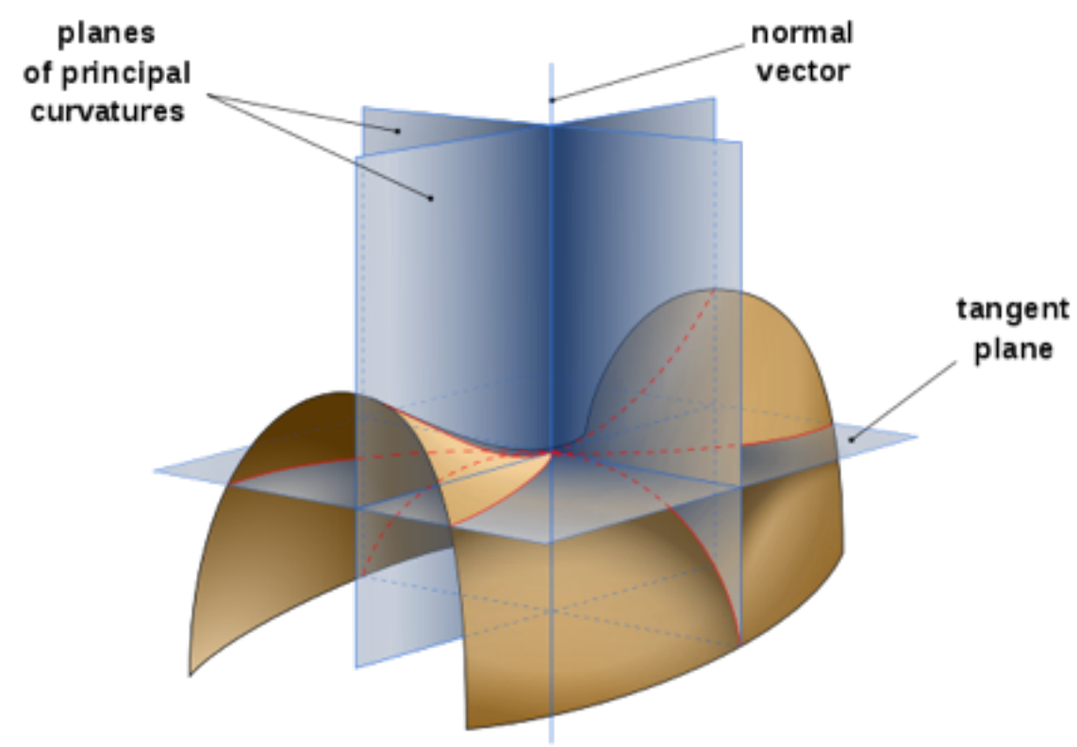
- Normals

➡ **Curvature**

Warps

- Rigid x-form
- Deform





# Geometry Processing

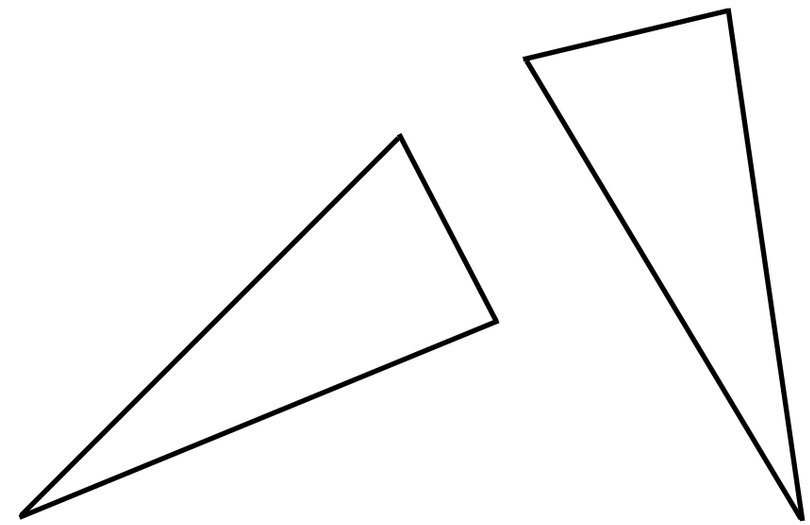
Analysis

- Normals
- Curvature

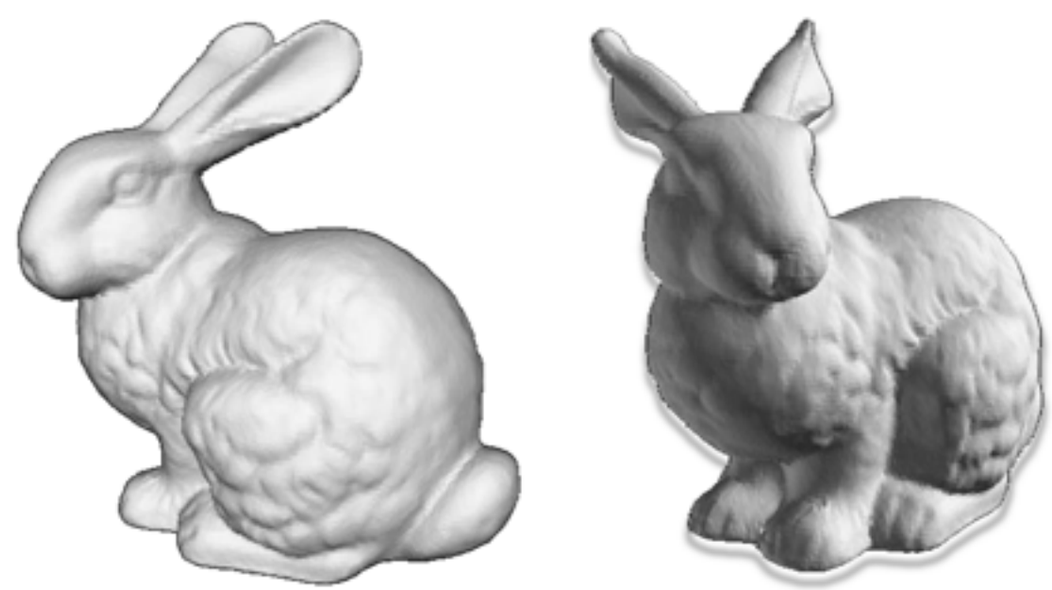
Warps

➡ **Rigid x-form**

- Deform



**Related by Rotation + Translation**



**E.g. same model scanned from different viewpoints**

# Geometry Processing

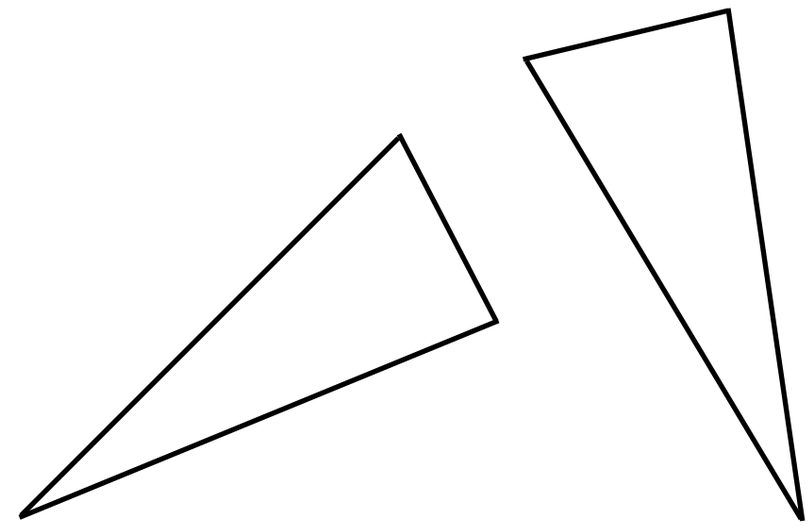
Analysis

- Normals
- Curvature

Warps

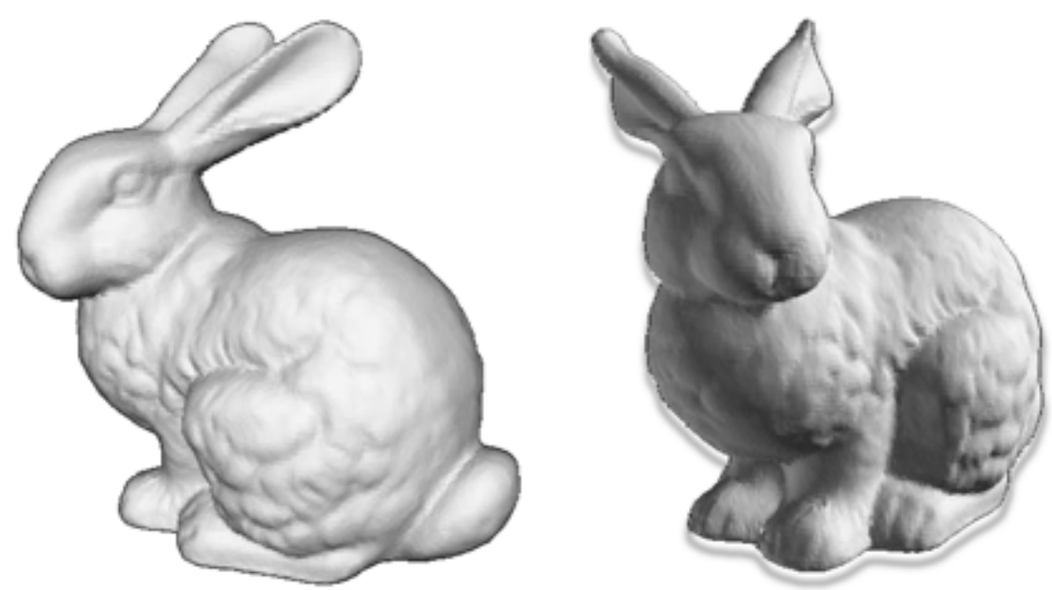
➡ **Rigid x-form**

- Deform



**Related by Rotation + Translation**

**NOTE: plenty of transformations!**



**E.g. same model scanned from different viewpoints**

# Geometry Processing

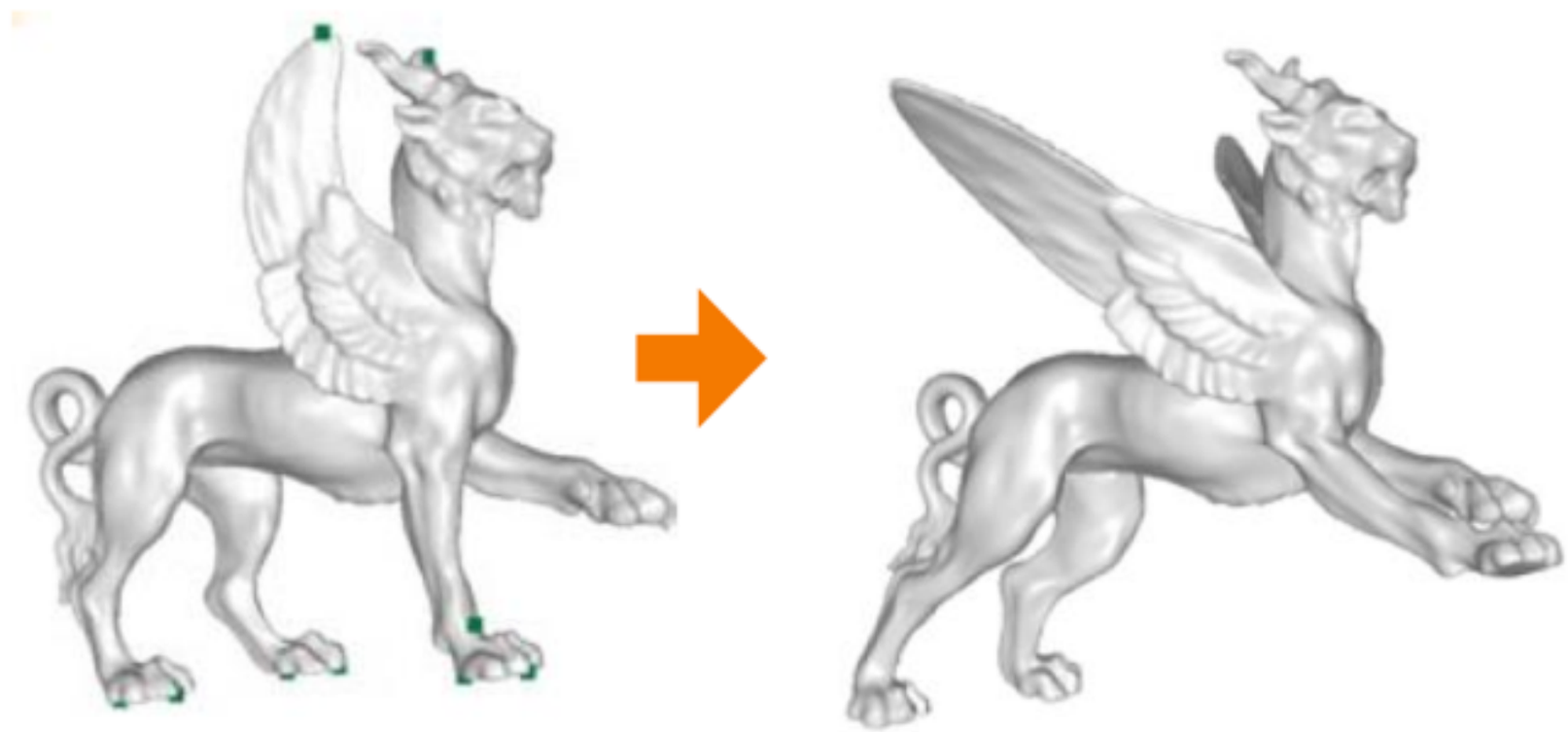
Analysis

- Normals
- Curvature

Warps

- Rigid x-form

➡ **Deform**



**NOTE: plenty of transformations!**

**Including higher-dimensional free-form deformations**

# Today's Lecture

Rest of today

- Basics: meshes & point clouds

➡ **Shape Descriptors (start)**