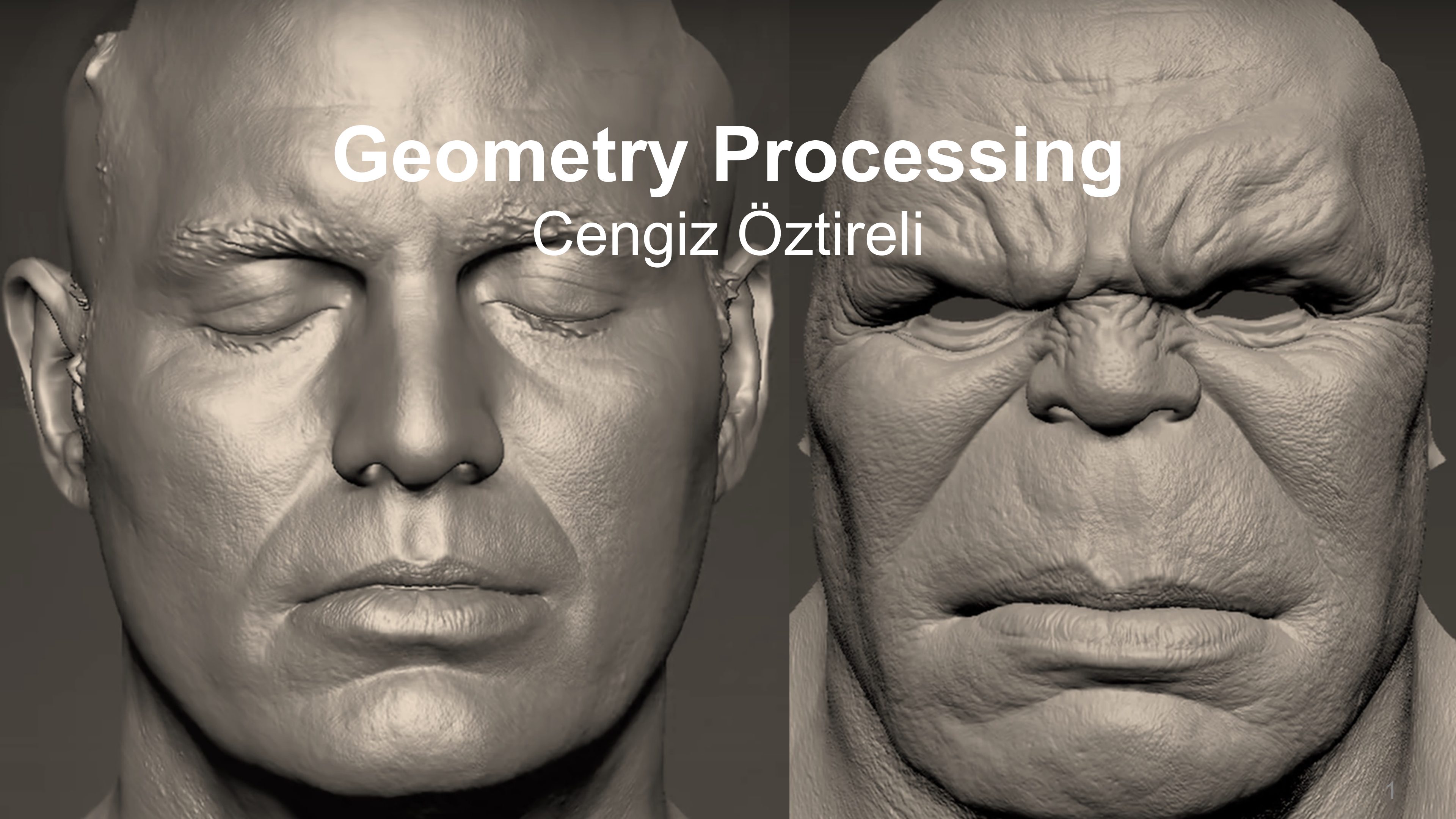


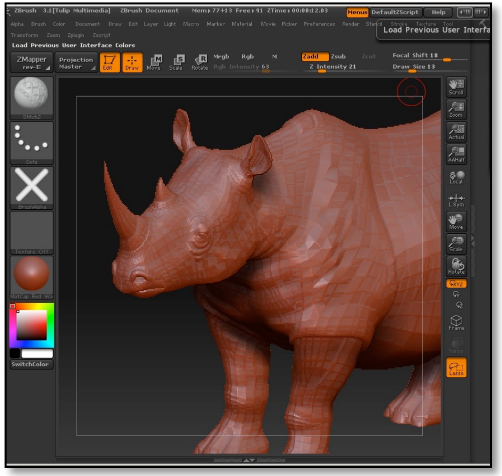
Geometry Processing

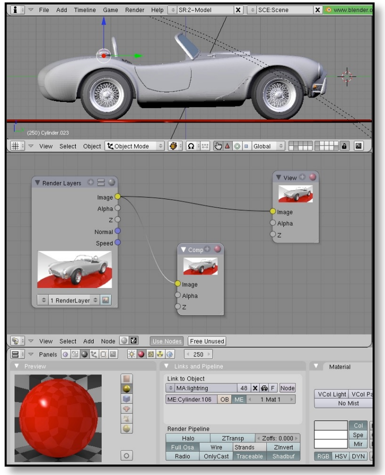
Cengiz Öztireli

Sources of Geometry

Acquisition from the real world

Modeling applications

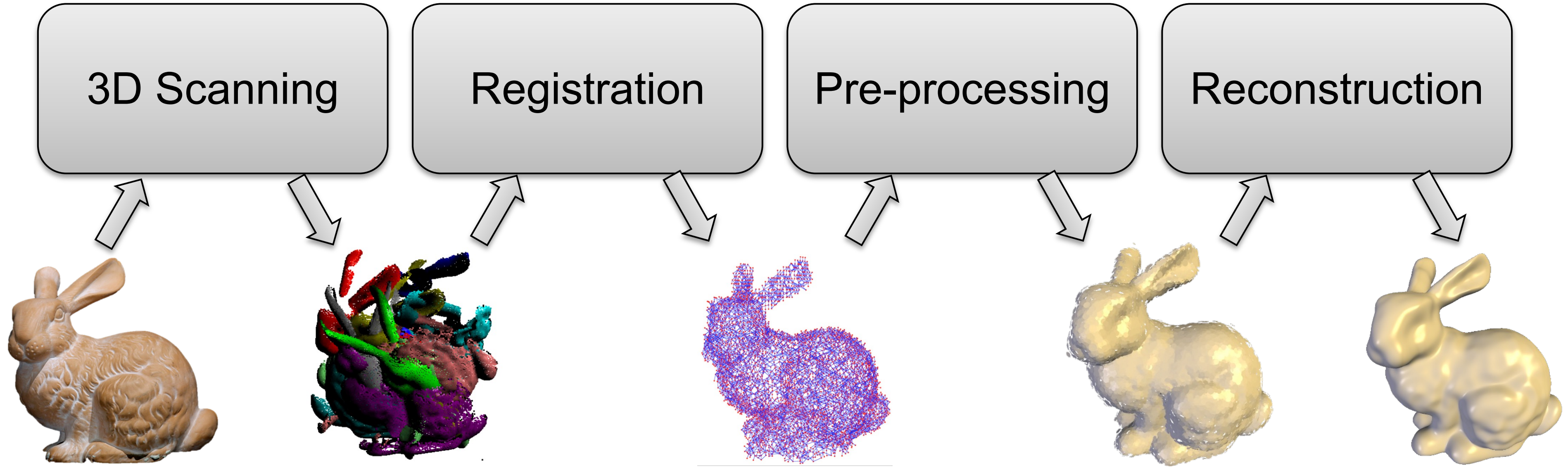




UNIVERSITY OF CAMBRIDGE

Shape Acquisition

- Digitizing real world objects



Shape Acquisition

Most modern 3D scanning systems are optical sensor based.
In optical scanning, active systems have a light source, laser, etc. that we can control.
Passive systems only have a sensor and we do not have control over the environment and object poses.

- 3D Scanning

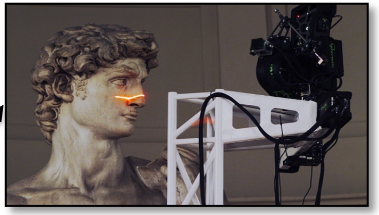
Touch Probes

+ Precise
- Small objects

Optical Scanning

+ Fast
- Glossy objects

Active



Passive

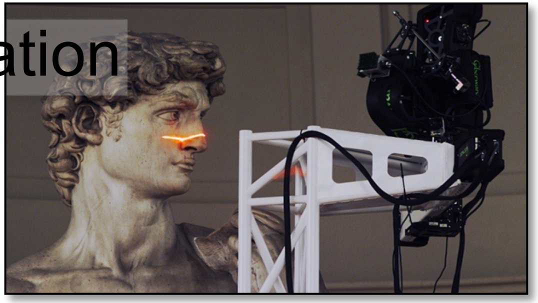
Shape Acquisition

- Optical Scanning – Active Systems

LIDAR

Measures the time it takes the laser beam to hit the object and come back

Triangulation Laser



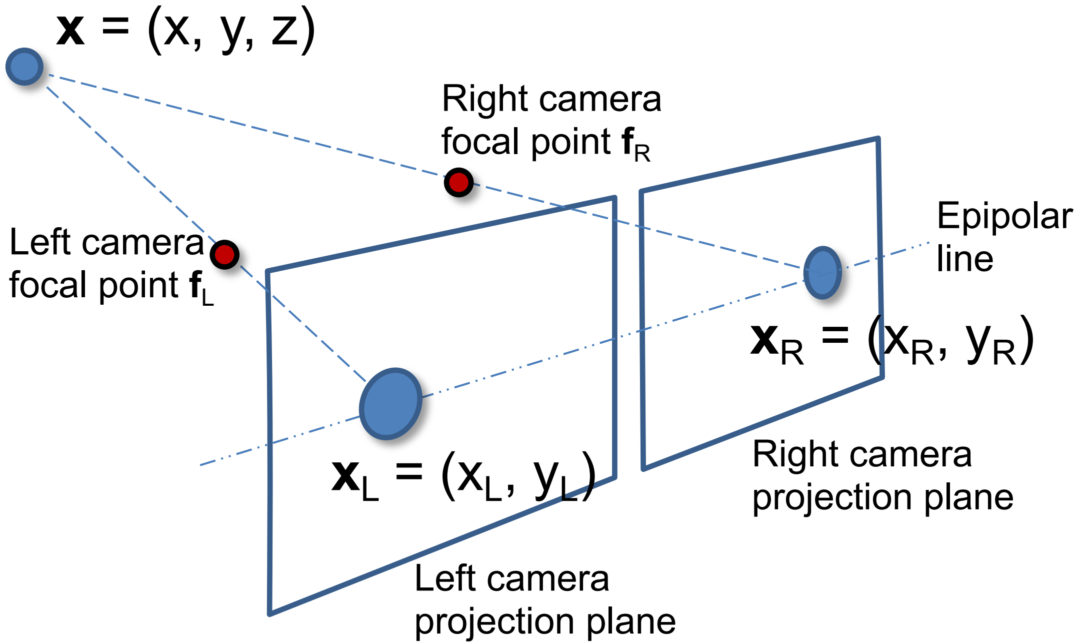
Projected laser beam is photographed, giving the distance of the pattern

UNIVERSITY OF CAMBRIDGE

An example of a passive system is multi-view stereo.
Each point $\mathbf{x}$ in 3D maps to two different points, $\mathbf{x}_L$ and $\mathbf{x}_R$.
We can find x, by intersecting the lines connecting $\mathbf{f}_L$ and $\mathbf{x}_L$, and $\mathbf{f}_R$ and $\mathbf{x}_R$.
Of course, this assumes we already know $\mathbf{x}_L$ and $\mathbf{x}_R$ corresponds to the same point in 3D.

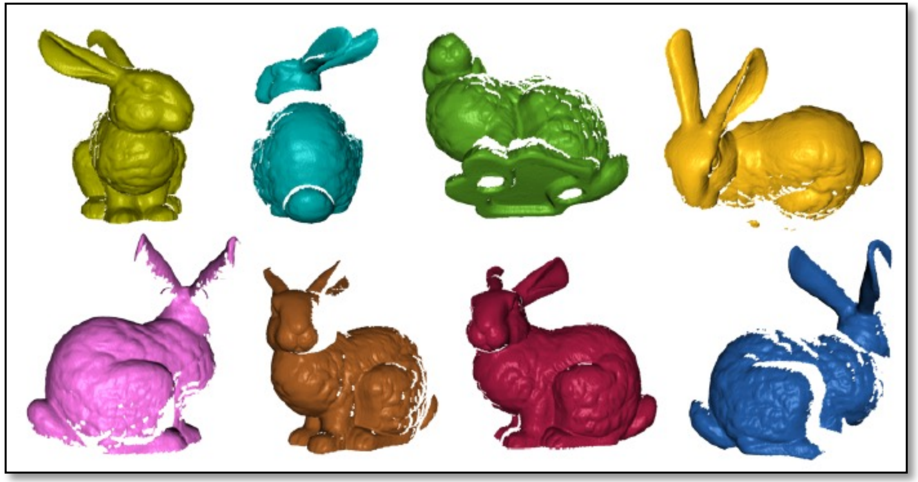
• Optical Scanning – Passive Systems

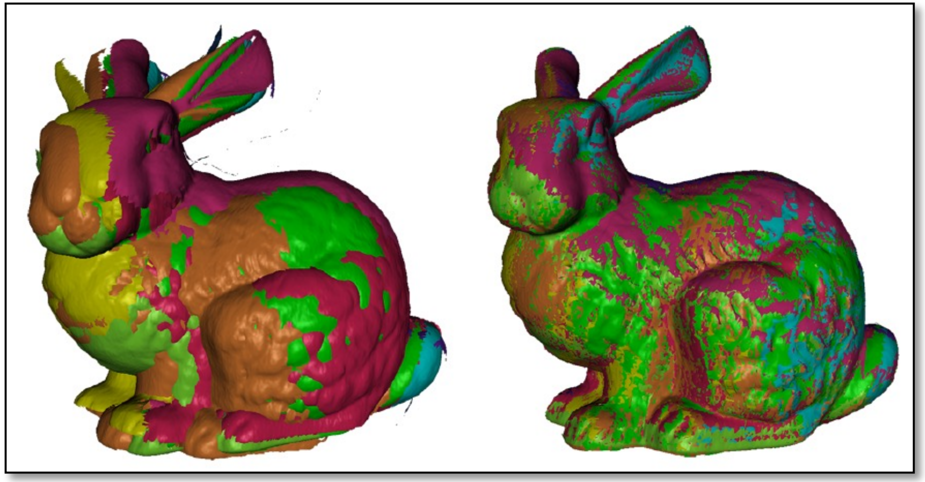
Multi-view Stereo


x = (x, y, z)
Right camera focal point $\mathbf{f}_R$
Left camera focal point $\mathbf{f}_L$
Epipolar line
$\mathbf{x}_R = (x_R, y_R)$
$\mathbf{x}_L = (x_L, y_L)$
Right camera projection plane
Left camera projection plane

When objects are acquired, the 3D point clouds are in different coordinate frames, i.e. rotated and translated. Registration is the process of bringing all of them into a common coordinate frame.

- Registration
  - Bringing scans into a common coordinate frame





UNIVERSITY OF CAMBRIDGE

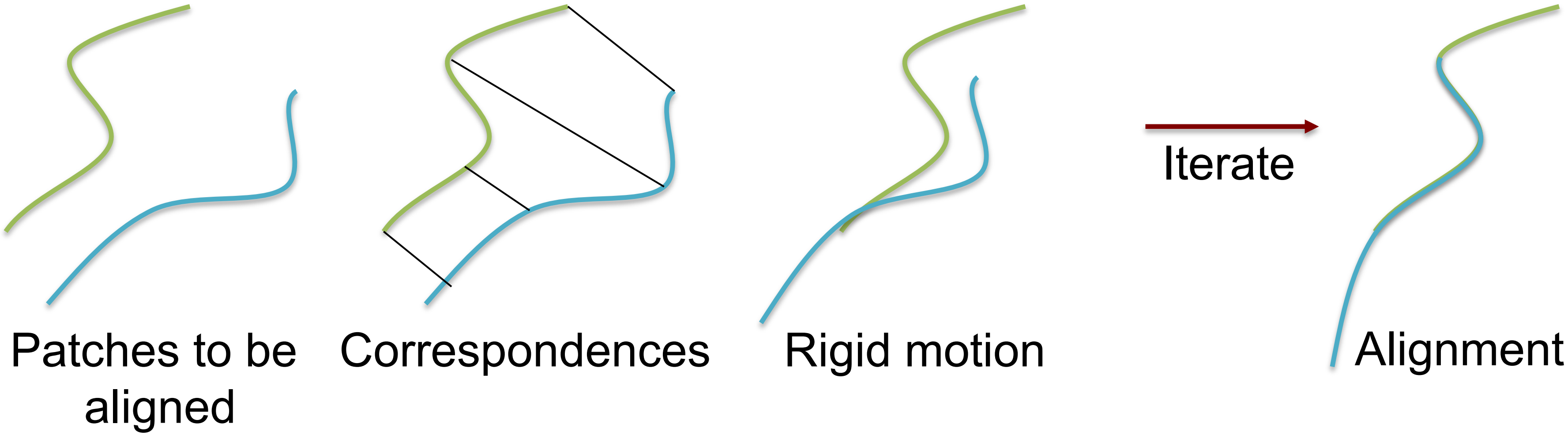
Shape Acquisition

Iterative closes point (ICP) algorithms are typically used for precise registration.

ICP: 1. establish point-to-point correspondences, 2. compute and apply the optimum rotation and translation, 3. iterate.

- Registration

Iterative Closest Point Algorithms

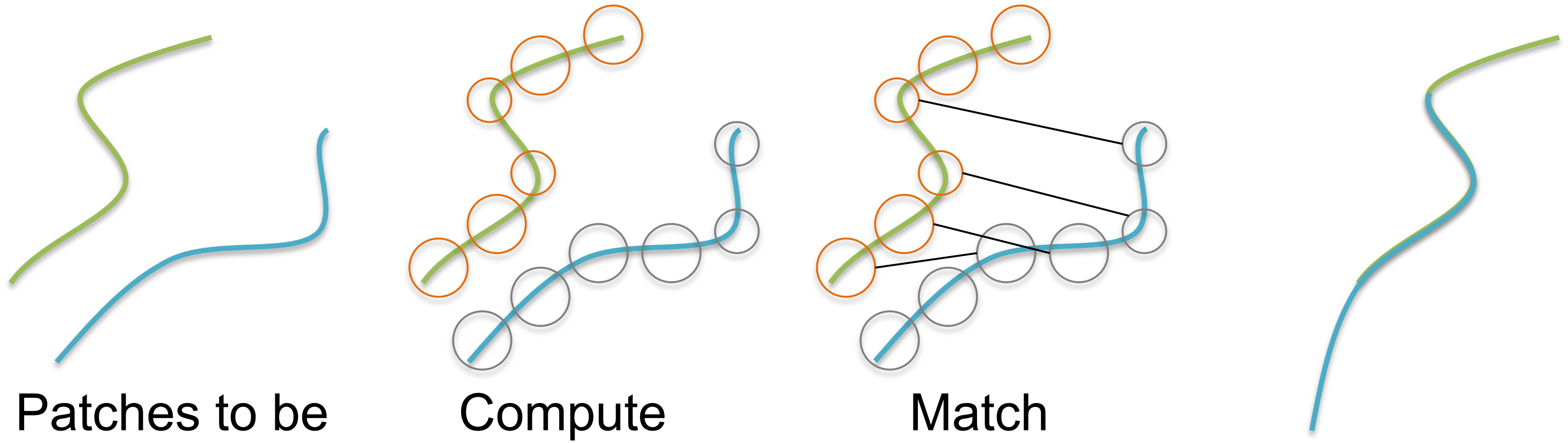


Shape Acquisition

If rotation/ translation of the initial patches are really off, establishing correspondences is not easy.
Feature-based methods compute rotation and translation invariant descriptors for points and match those.

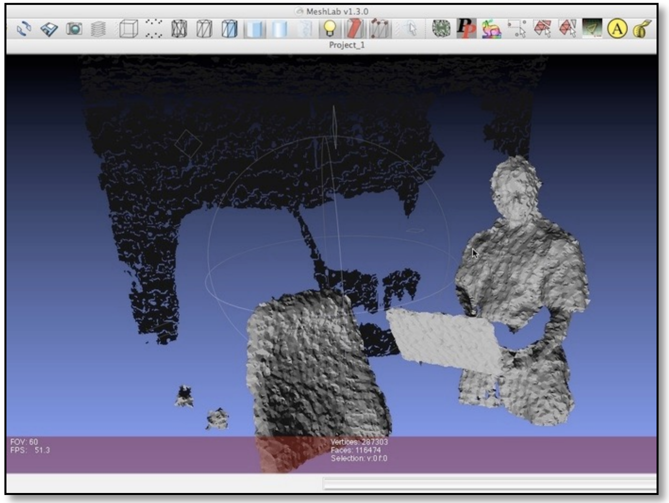
- Registration

Feature-based Methods



Shape Acquisition

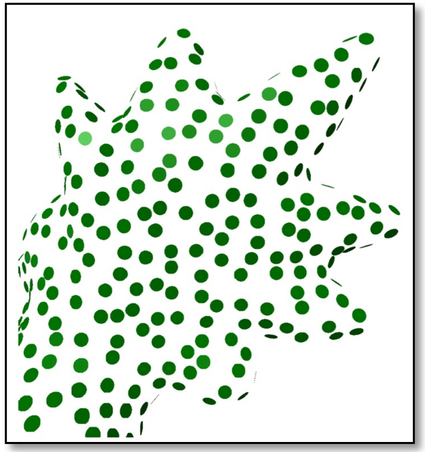
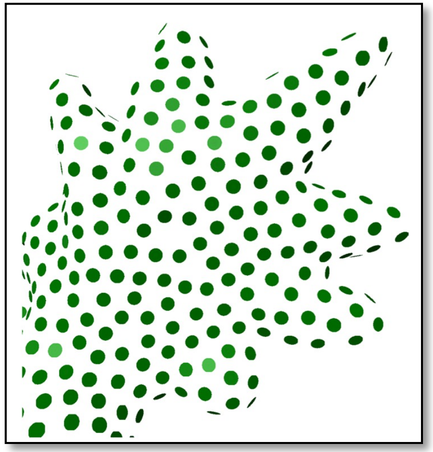
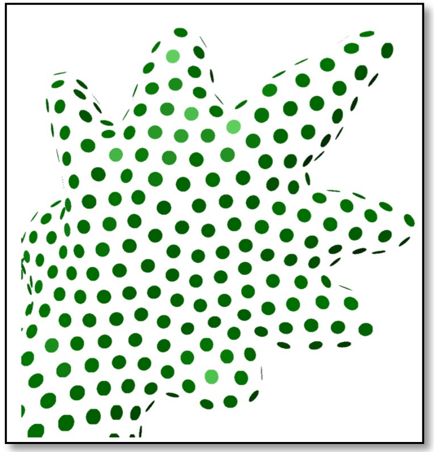
- Pre-processing
  - Cleaning, repairing, resampling



Accurate and efficient reconstructions of surfaces fundamentally depends on how we distribute points on the surface to represent the surface.

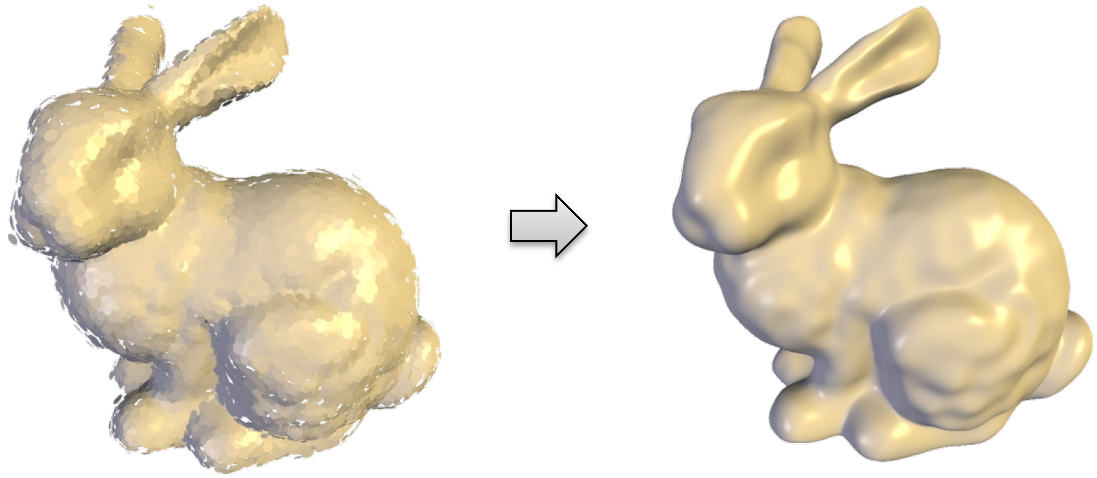
There are extensions of the sampling theorem to irregular sampling on surfaces.

- Pre-processing
  - Sampling for accurate reconstructions

Shape Acquisition

- Reconstruction
  - Mathematical representation for a shape

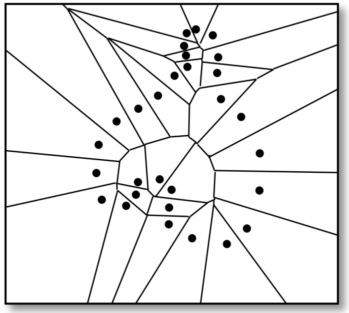
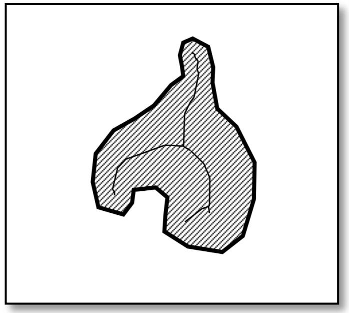


Shape Acquisition

There are two ways to reconstruct a mathematical definition of a surface.
The first one connects the sample points with lines/ triangles/ polygons, etc.
The second method approximates the underlying surface, e.g. via a least squares solution.

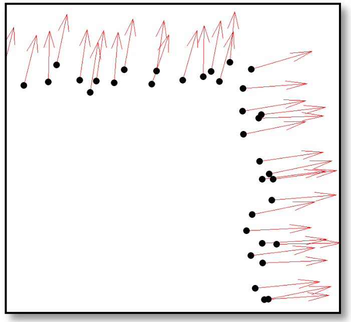
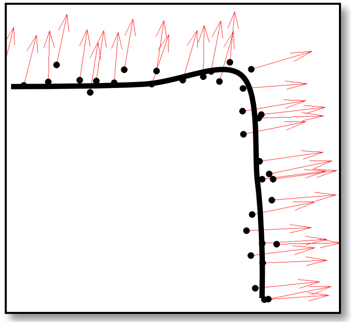
• Reconstruction

Connect-the-points Methods

+ Theoretical error bounds
– Expensive
– Not robust to noise

Approximation-based Methods

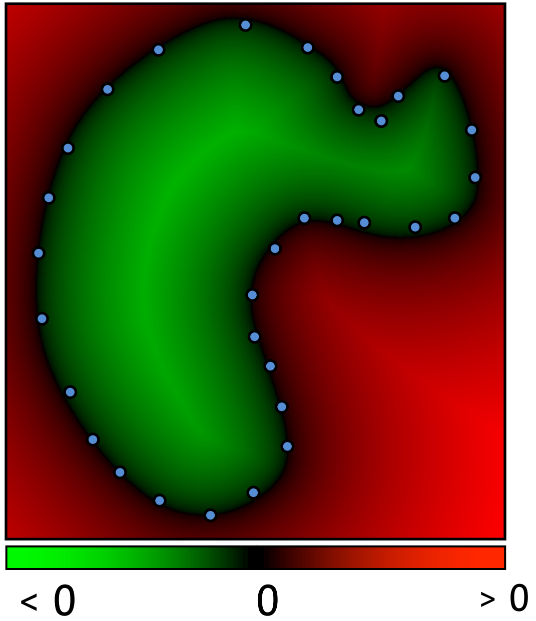
+ Efficient to compute
+ Robust to noise
– No theoretical error bounds

UNIVERSITY OF CAMBRIDGE

• Approximating an implicit function

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}$$

with value > 0 outside the shape and < 0 inside



Shape Acquisition

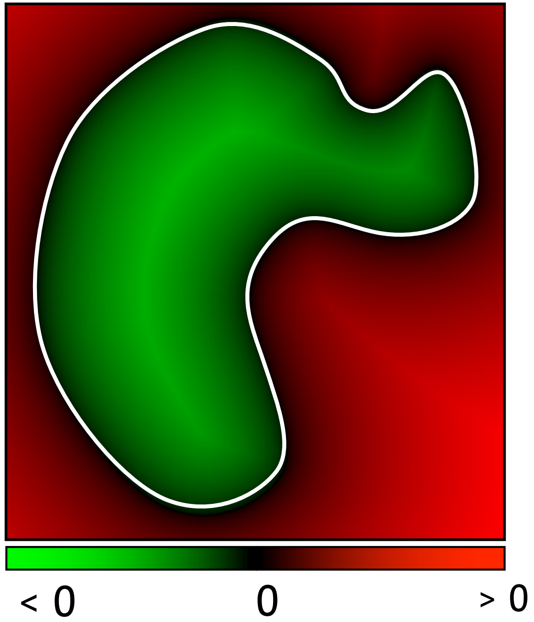
- Approximating an implicit function

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}$$

with value > 0 outside the shape and < 0 inside

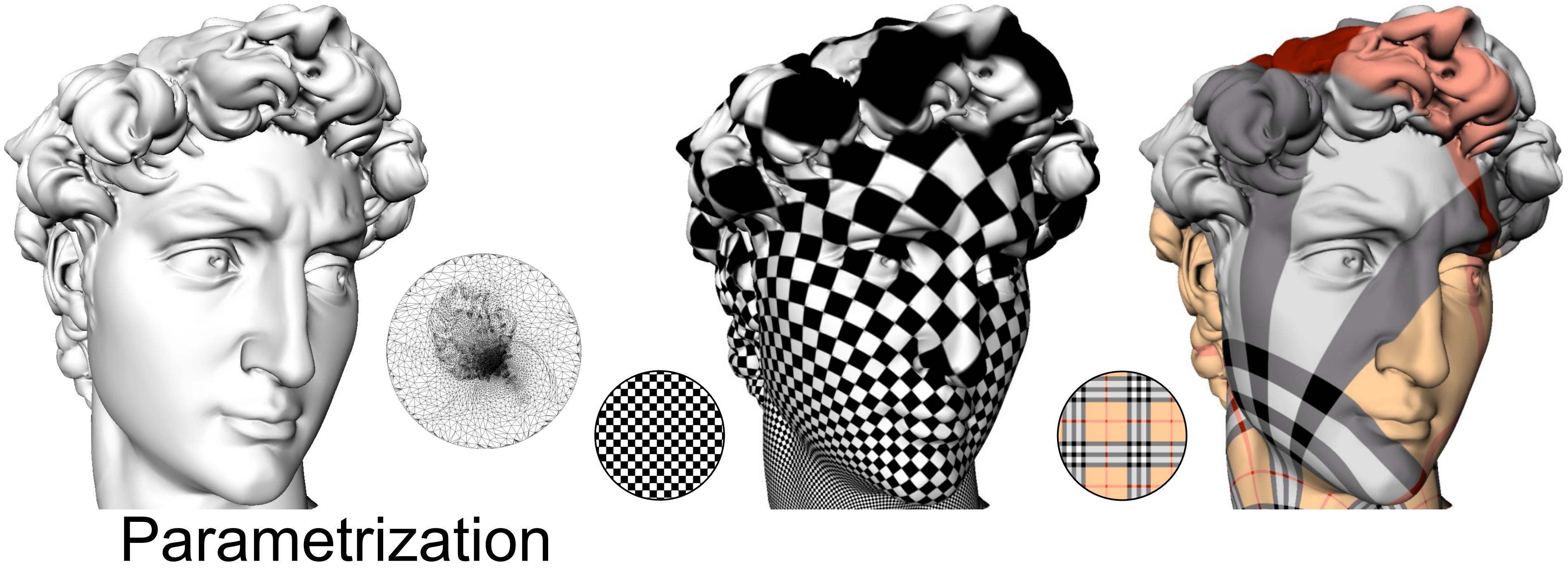
$$\{\mathbf{x} : f(\mathbf{x}) = 0\}$$

extract zero set



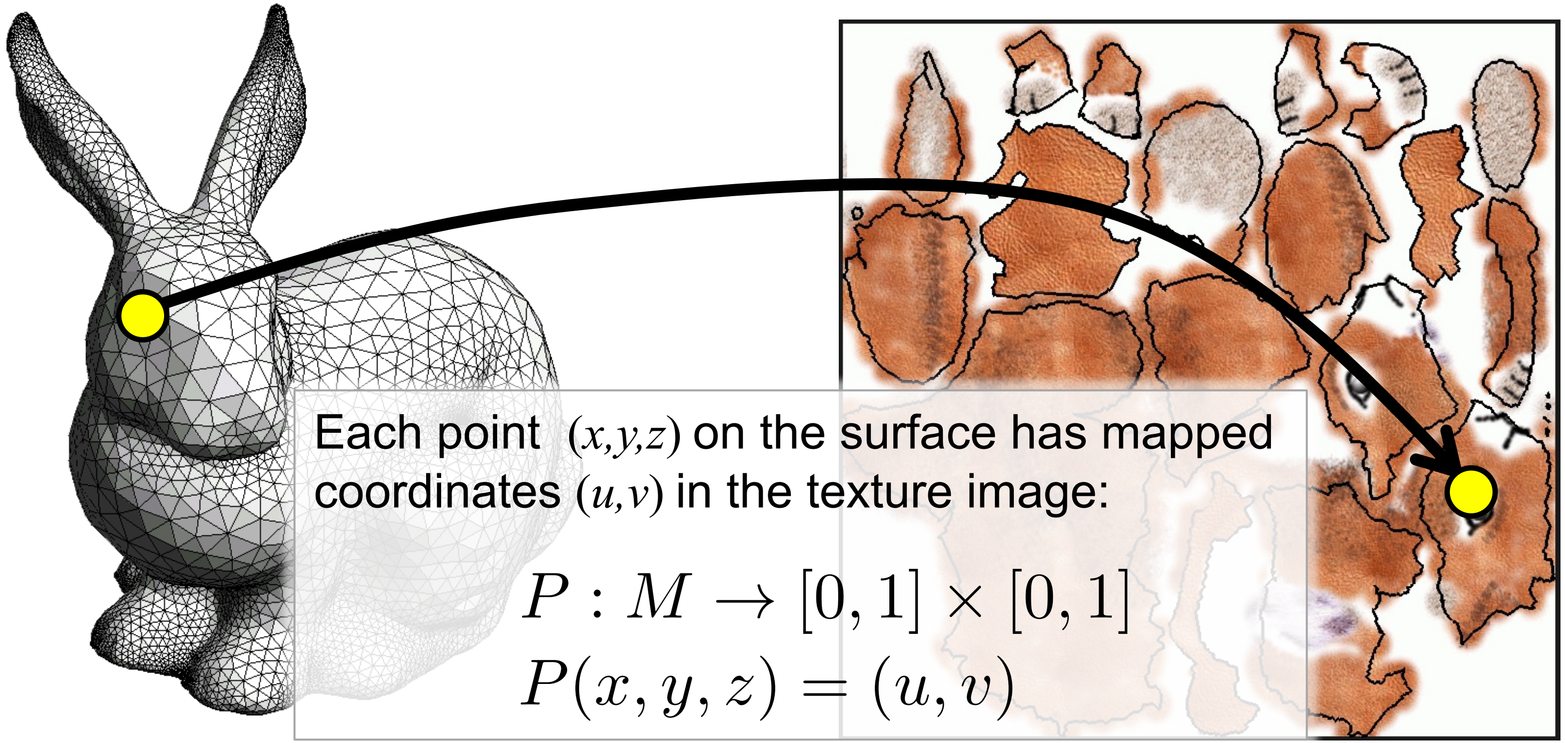
Texture Mapping

Once the surface of an object is reconstructed, we add color and detail using texture mapping.
This requires parametrizing the surface, i.e. finding a mapping from a plane to the surface.



Texture Mapping

The $(u,\ v)$ coordinates live on the image plane. For each $(u,\ v)$, we have the corresponding point $(x,\ y,\ z)$. We can assume the texture space is a unit square and define a map P from the surface to this square.



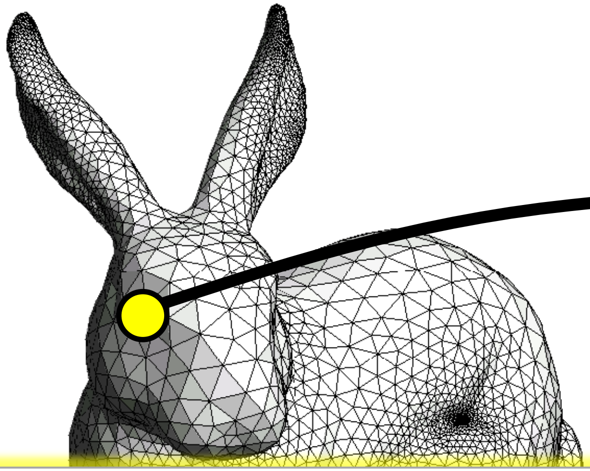
Each point (x,y,z) on the surface has mapped coordinates (u,v) in the texture image:

$$P : M \to [0,1] \times [0,1]$$
$$P(x,y,z) = (u,v)$$

UNIVERSITY OF CAMBRIDGE

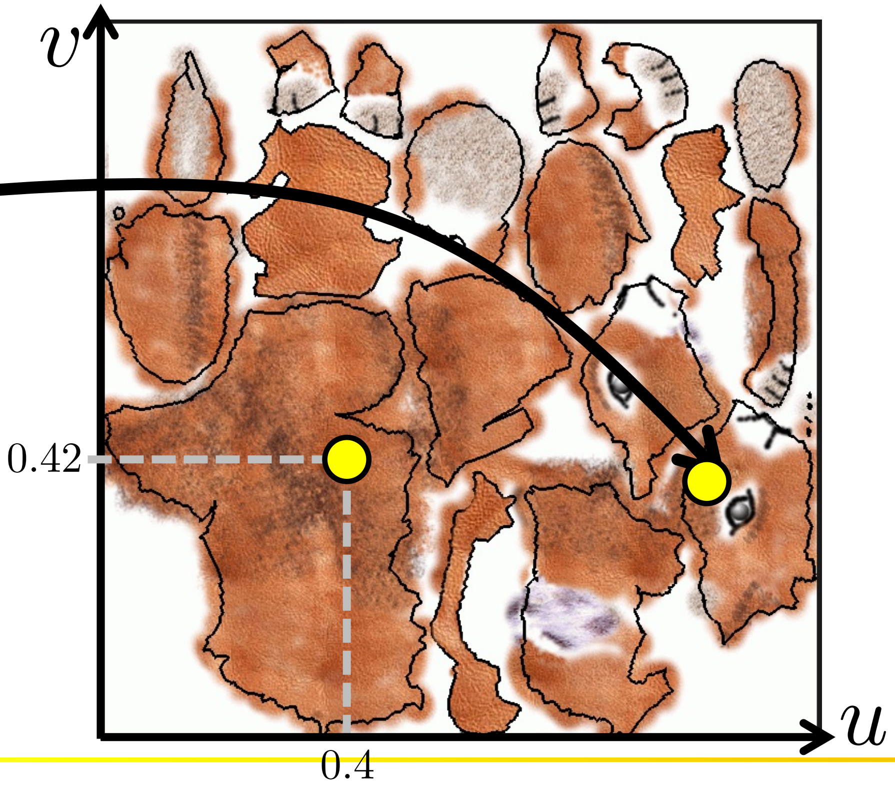
Texture Mapping

We further have an attribute, e.g. color, associated with each (u, v).
Let's denote this map with T.



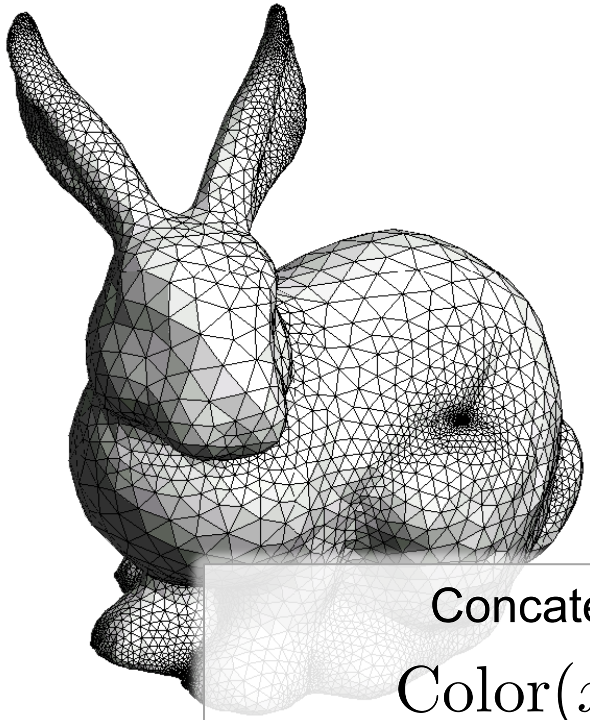
Texture itself is a function:

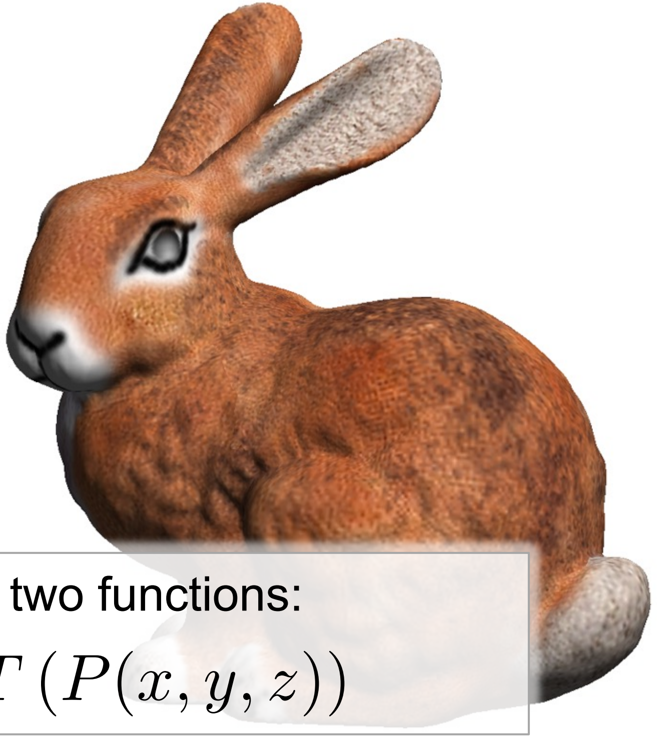
$$T : [0,1] \times [0,1] \to \text{RGB}$$
$$T(u,v) = (r,g,b)$$



Texture Mapping

These two maps define the final map that provides us with a color for each point on the surface.





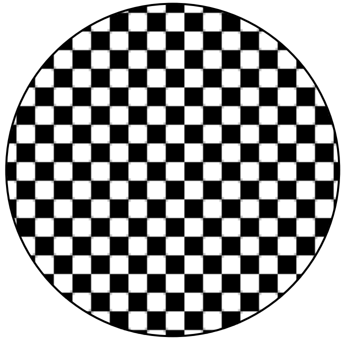
Concatenation of the two functions:

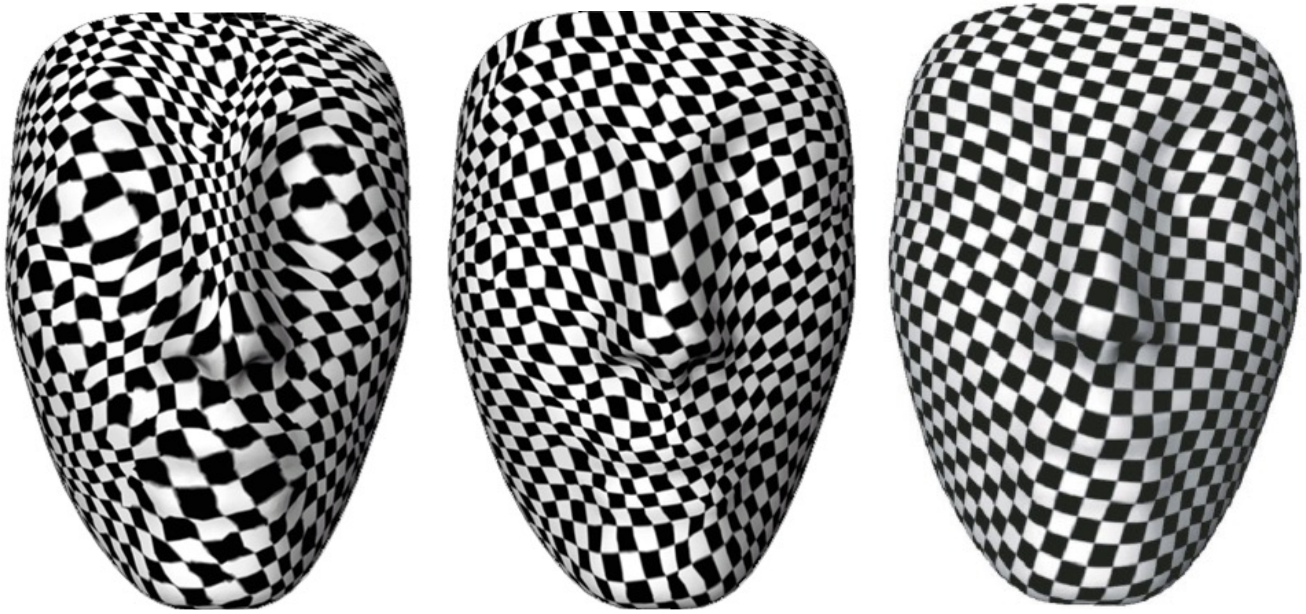
$$\text{Color}(x, y, z) = T\left(P(x, y, z)\right)$$

UNIVERSITY OF CAMBRIDGE

Texture Mapping

The critical part here is the parametrization P. In general, it is impossible to have a distortion-free map from a plane to a surface in 3D. We need to decide on what distortion we allow for.



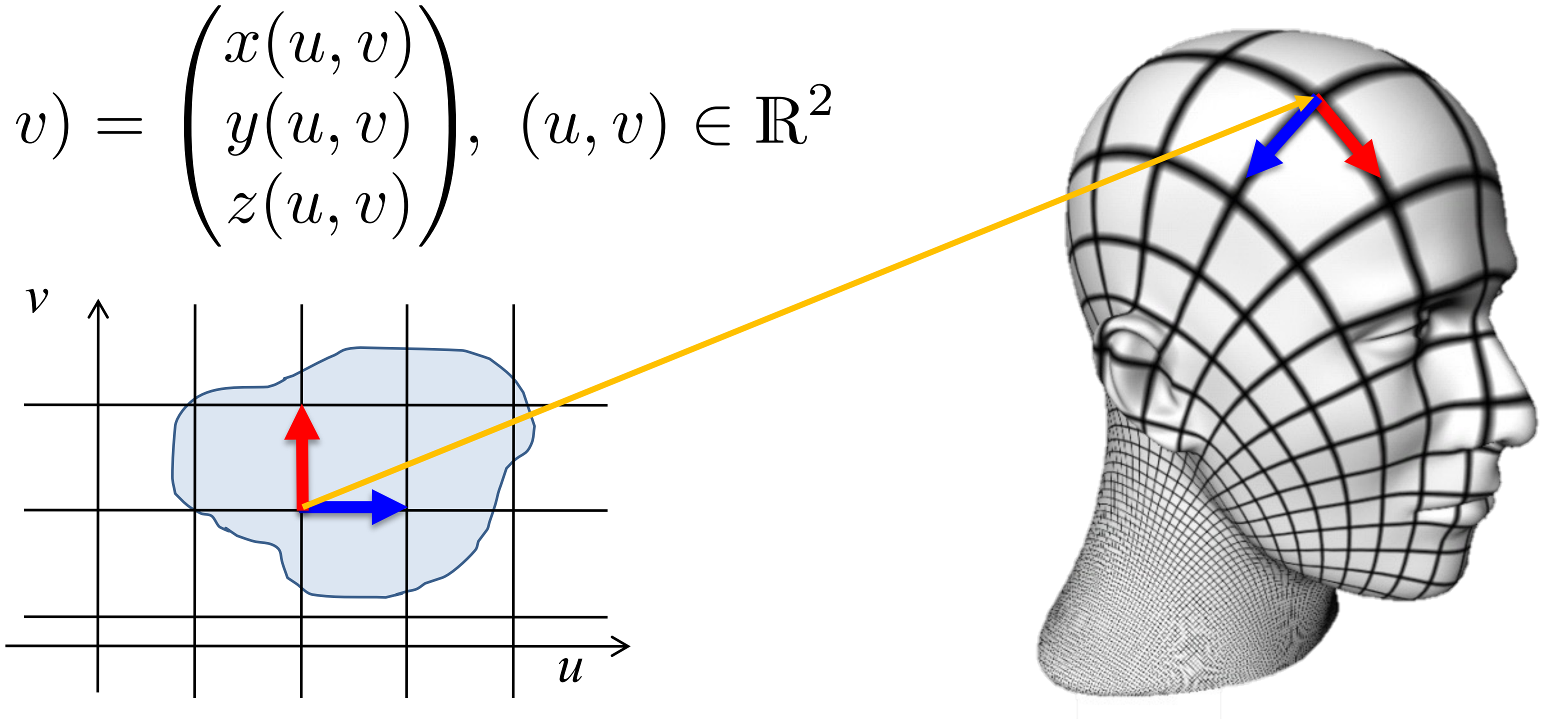


Parametrization

To define the properties of distortion, we can first explicitly write $\mathbf{p}$ as three functions, one for each component, as we did before.

Such a parametrization implies a map from vectors in the uv – space to those in the 3D space.

$$\mathbf{p}(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}, \quad (u,v) \in \mathbb{R}^2$$



UNIVERSITY OF CAMBRIDGE

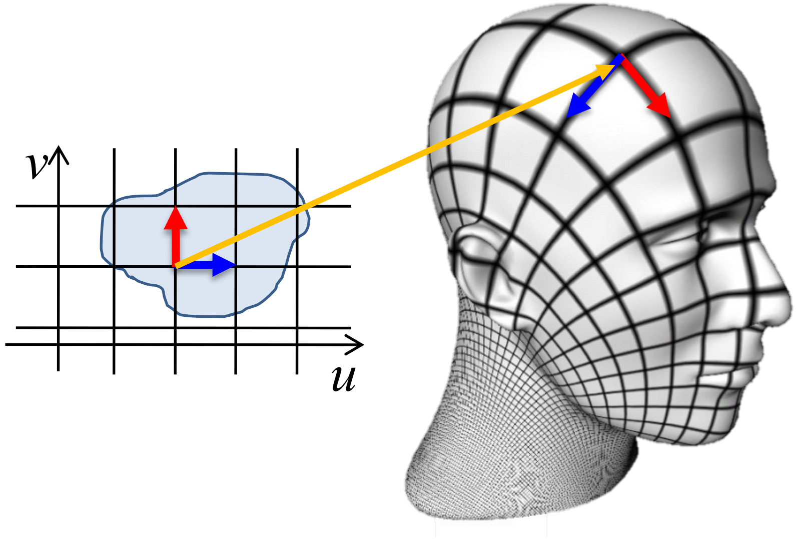
Parametrization

We can then define the so-called first fundamental form. This is a 2 x 2 matrix consisting of dot products of derivatives with respect to u and v.
Note that we denote the dot product with $^{\mathrm{T}}$, implying the transpose operator followed by matrix multiplication.

$$\mathbf{p}(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}, \quad (u,v) \in \mathbb{R}^2$$

$$\mathbf{p}_u = \frac{\partial \mathbf{p}(u,v)}{\partial u}, \quad \mathbf{p}_v = \frac{\partial \mathbf{p}(u,v)}{\partial v}$$

$$\mathbf{I} = \begin{pmatrix} E & F \\ F & G \end{pmatrix} = \begin{pmatrix} \mathbf{p}_u^T \mathbf{p}_u & \mathbf{p}_u^T \mathbf{p}_v \\ \mathbf{p}_u^T \mathbf{p}_v & \mathbf{p}_v^T \mathbf{p}_v \end{pmatrix}$$



UNIVERSITY OF CAMBRIDGE

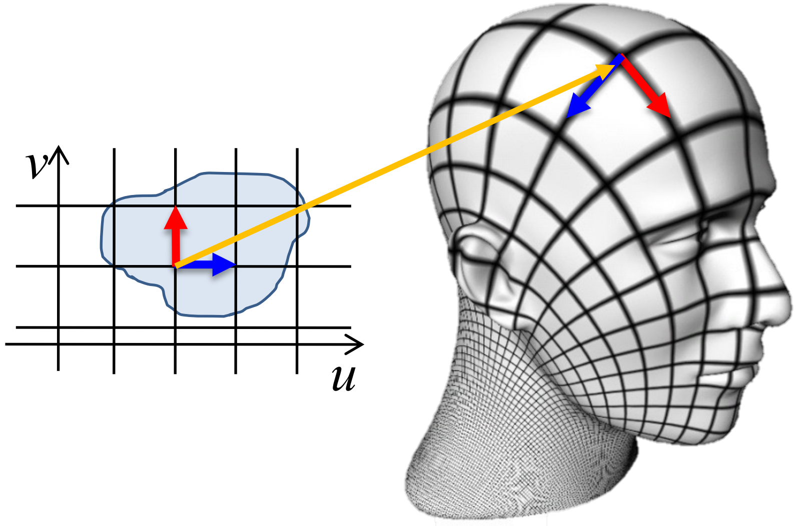
Parametrization

This matrix measures the angle and length change in different terms.
The area distortion can also be expressed in terms of these terms. It is defined as the change in areas when applying the map.

$$\mathbf{I} = \begin{pmatrix} E & F \\ F & G \end{pmatrix} = \begin{pmatrix} \mathbf{p}_u^T \mathbf{p}_u & \mathbf{p}_u^T \mathbf{p}_v \\ \mathbf{p}_u^T \mathbf{p}_v & \mathbf{p}_v^T \mathbf{p}_v \end{pmatrix}$$

Angle change

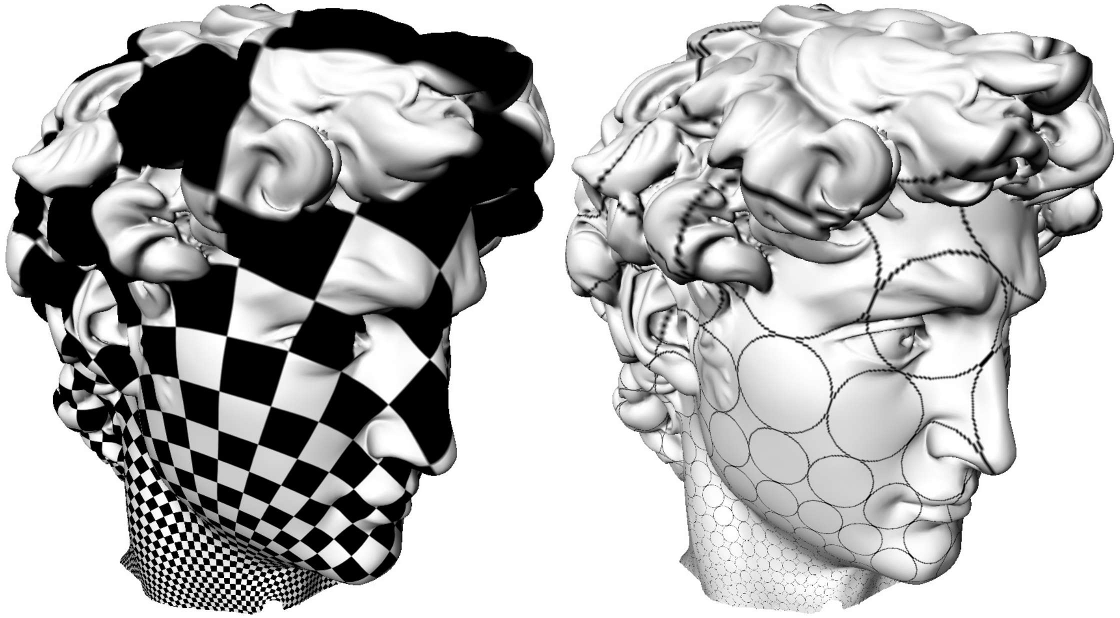
Length change



Area distortion: $dA = \sqrt{EG - F^2}\, du dv$

Conformal parametrization (angle preservation)

$$\mathbf{I} = \begin{pmatrix} \mathbf{p}_u^T\mathbf{p}_u & \mathbf{p}_u^T\mathbf{p}_v \\ \mathbf{p}_u^T\mathbf{p}_v & \mathbf{p}_v^T\mathbf{p}_v \end{pmatrix} = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$$

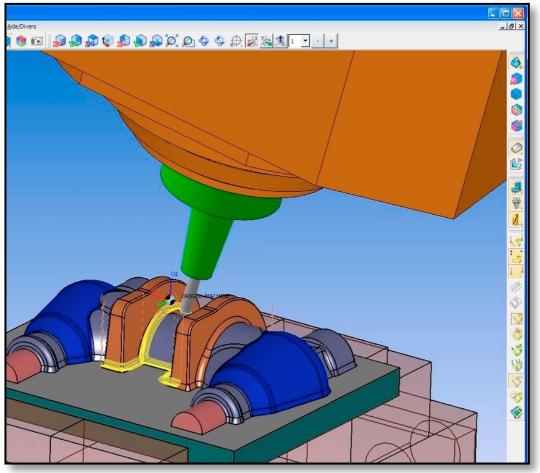


Editing Geometry
There are many established tools to further edit geometry.
Some are artistic, e.g. via sculpting, others are engineering related, e.g. designing parts of a machine.
Procedural geometry, where we define the rules for geometry creation, is becoming more popular, e.g. infinite virtual cities in games.

- Modeling tools

Sculpting

CAD/CAM



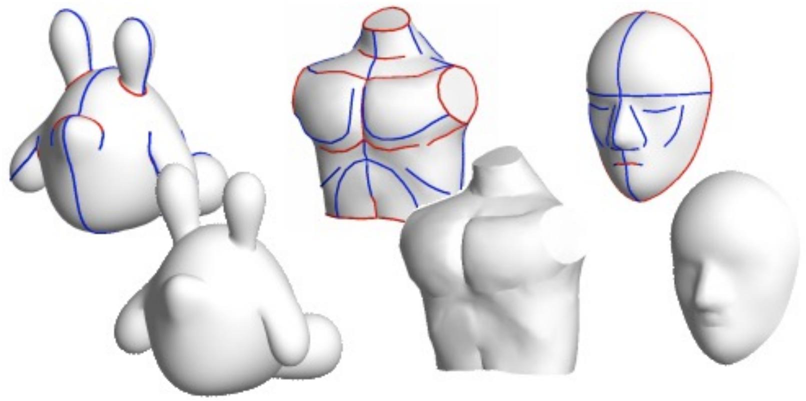
Procedural

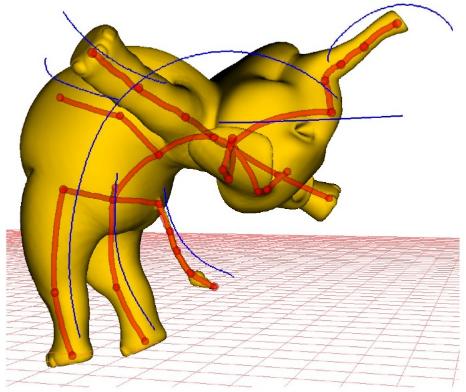
UNIVERSITY OF CAMBRIDGE

Editing Geometry

There are further novel interfaces based on e.g. contour or motion curve sketching.

- Interactive & sketch-based interfaces



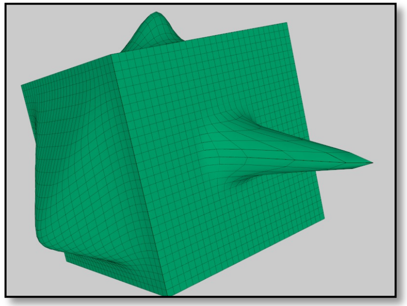


Editing Geometry

Deforming an initial surface is a typical workflow in digital modeling tools.
Such deformations can be based on low-level curvature related characteristics (e.g. elastic deformation) or high level controls (e.g. skeletal deformations as we will see in the next lectures).

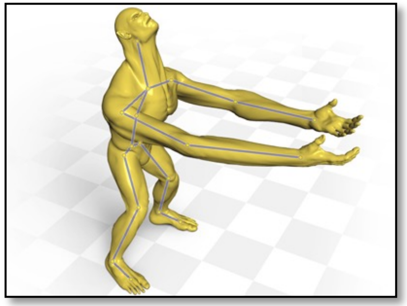
- Deformations

Free-form

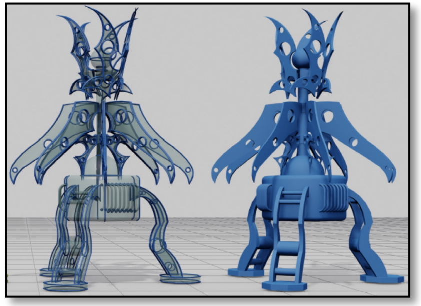


Elastic

Skeletal



Structure-aware

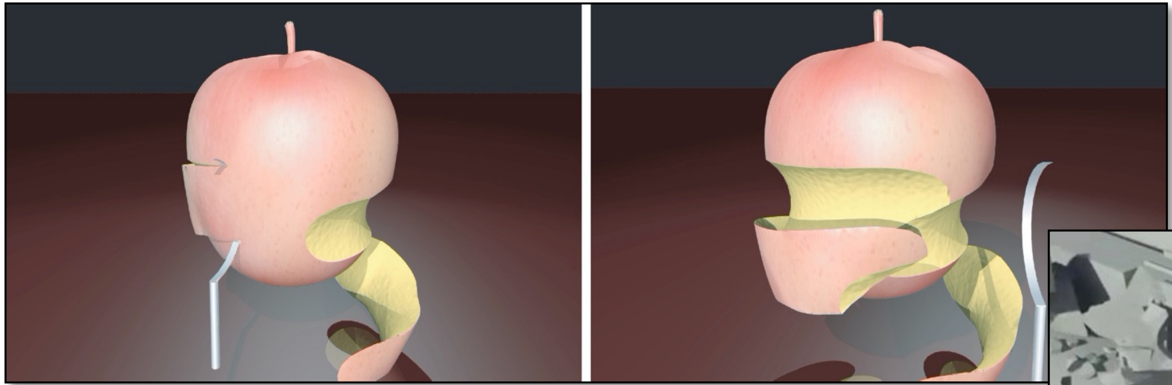
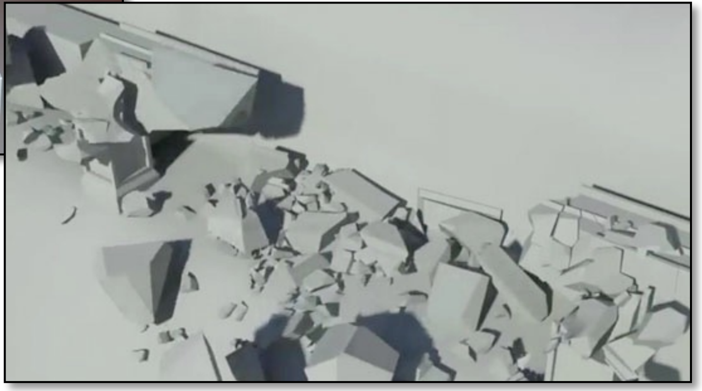


More structure

UNIVERSITY OF CAMBRIDGE

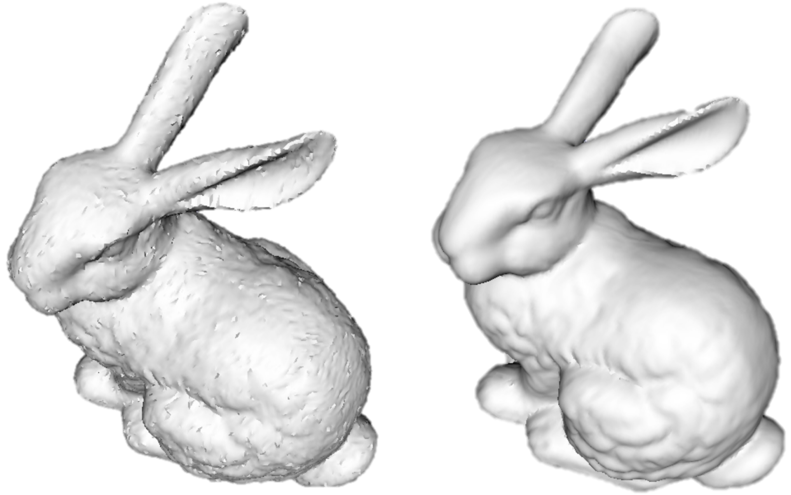
Editing Geometry
Cutting and fracturing leads to changing mesh connectivity.
This is a rather hard operation and one of the reasons why connectivity-free geometry representations are useful.

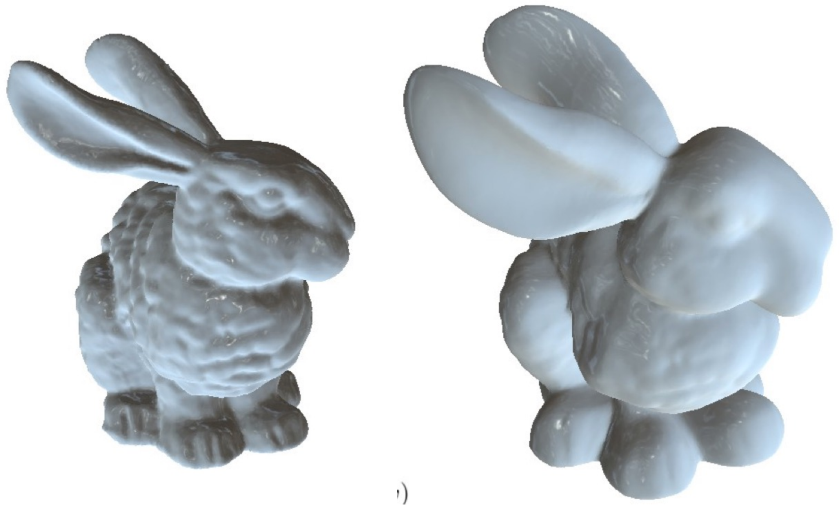
- Cutting & fracturing

Editing Geometry
Many algorithms exist for smoothing geometry or amplifying certain features of surfaces.
A lot of them are curvature-based, e.g. reducing local curvature leads to smoothing.

- Smoothing & filtering



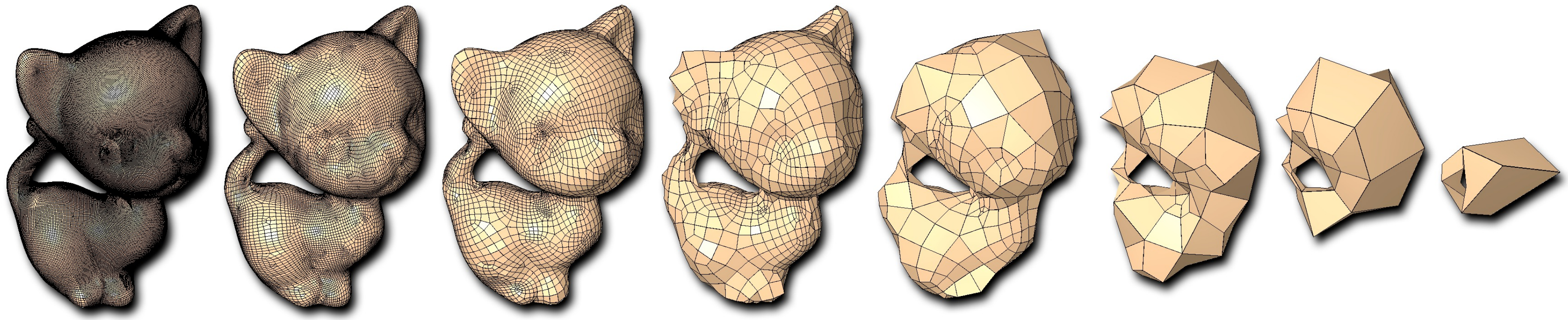


Editing Geometry

Compression of geometric representations, e.g. reducing the number of vertices in a mesh or quantizing the vertex locations, is important for modern large data processing with billions of triangles.

Many compression objectives can be reduced to preserving local properties given by curvature.

- Compression & Simplification



UNIVERSITY OF CAMBRIDGE