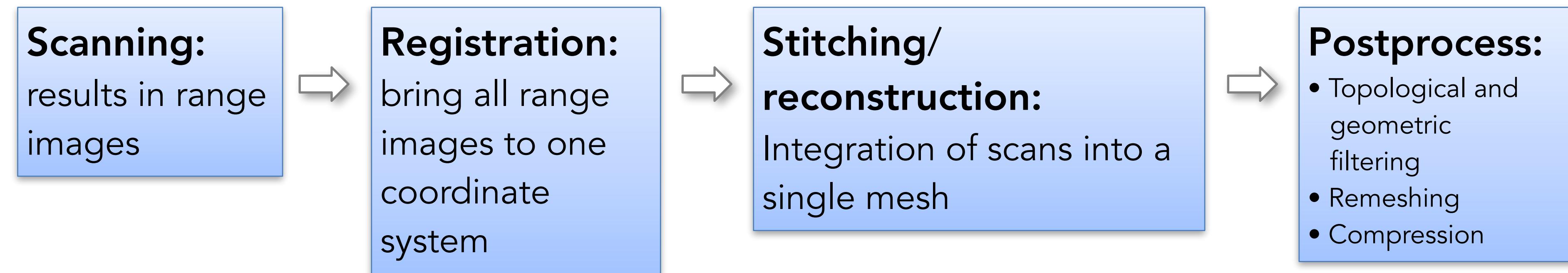


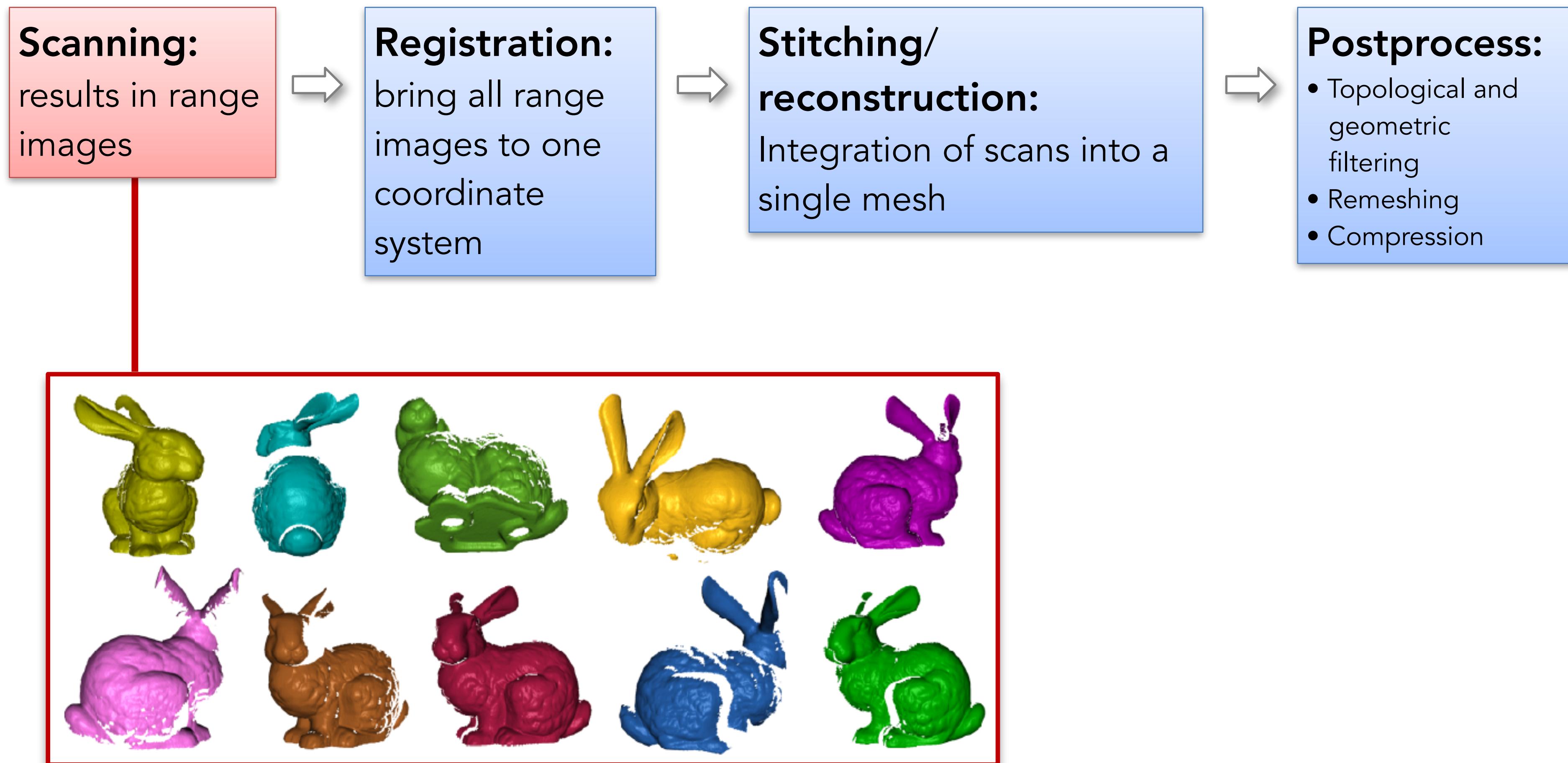
# Acquisition

Acknowledgements: Olga Sorkine-Hornung and Daniele Panozzo  
CSC 486B/586B - Geometric Modeling - Teseo Schneider

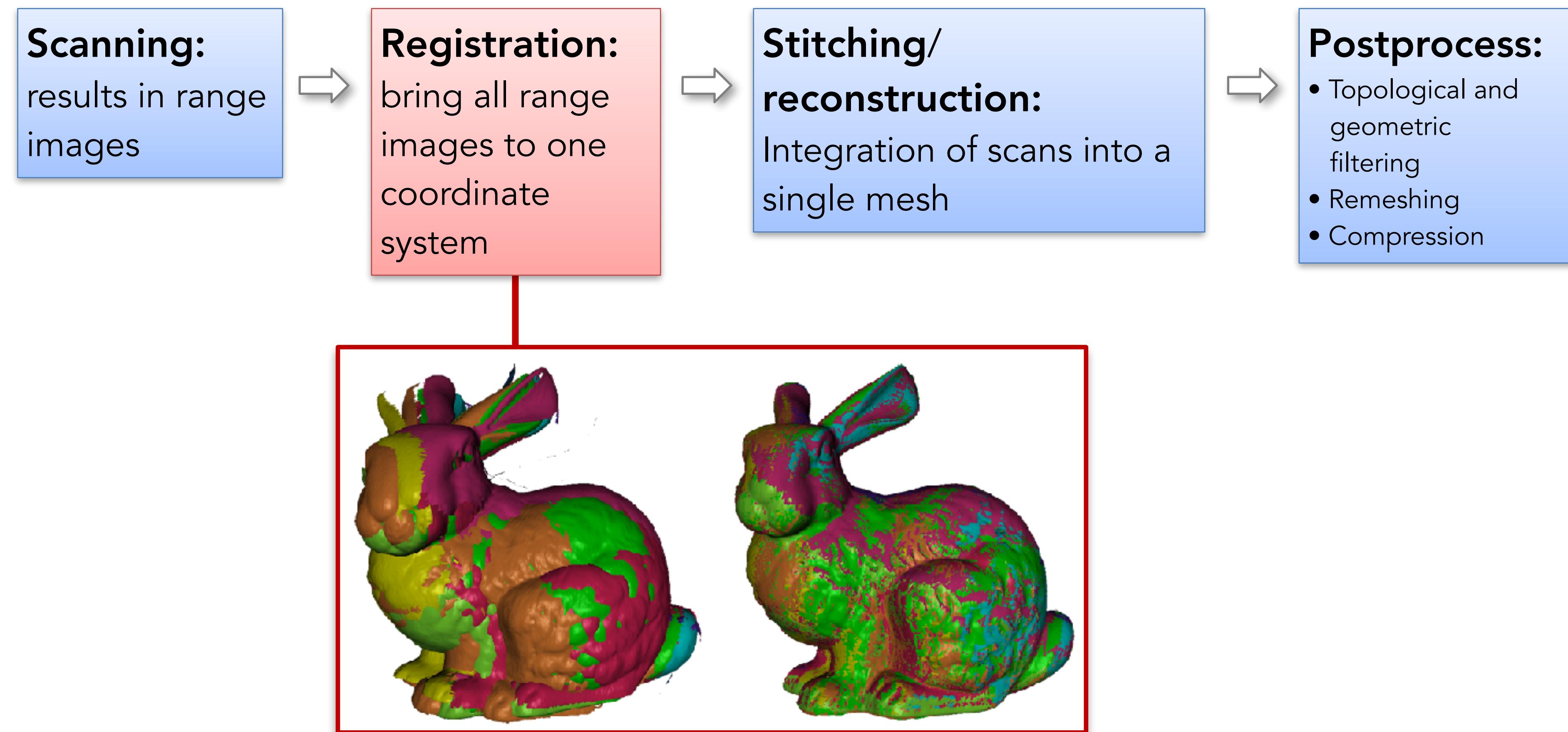
# Geometry Acquisition Pipeline



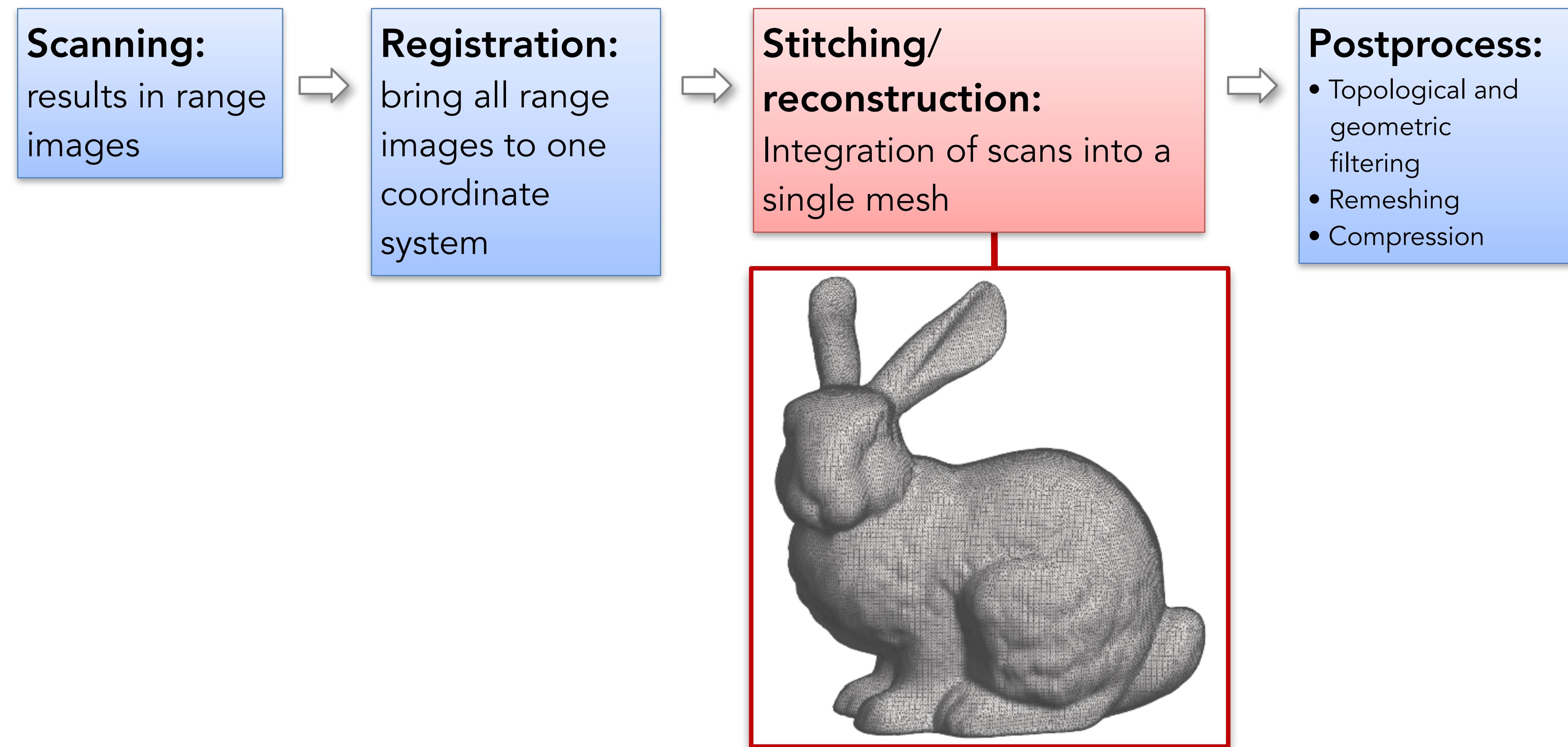
# Geometry Acquisition Pipeline



# Geometry Acquisition Pipeline

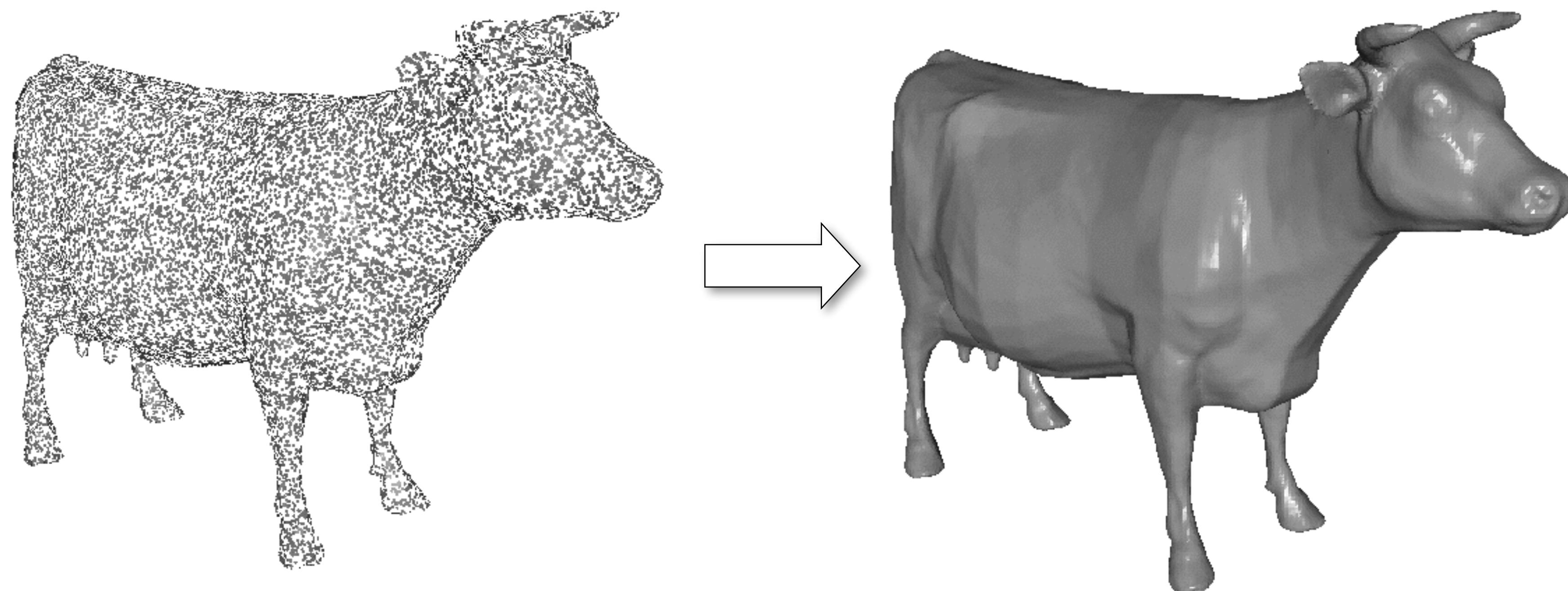


# Geometry Acquisition Pipeline

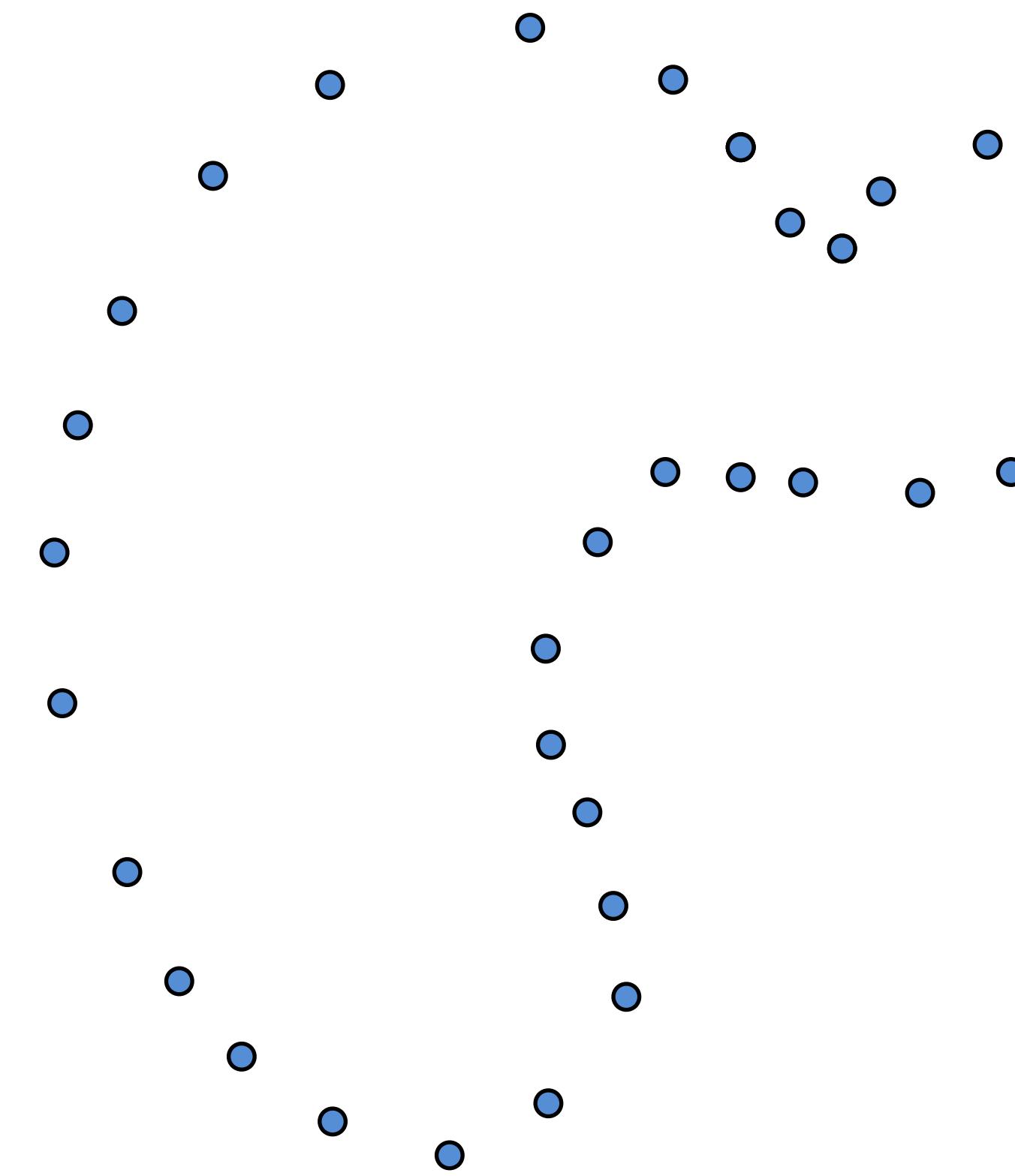


# Surface Reconstruction

- Generate a mesh from a set of surface samples



# Implicit Function Approach

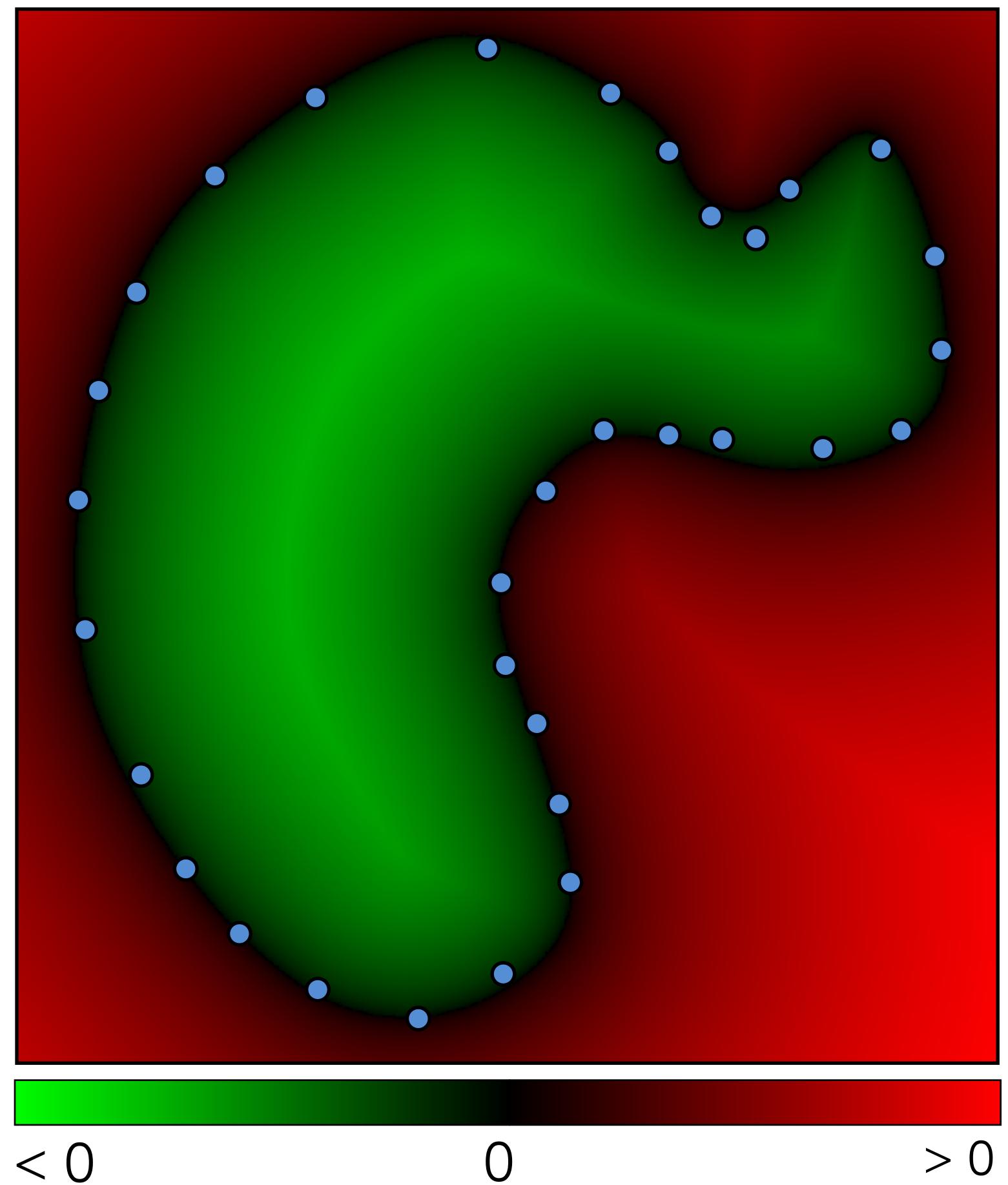


# Implicit Function Approach

- Define a function

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

with value  $> 0$  outside the shape and  
 $< 0$  inside



# Implicit Function Approach

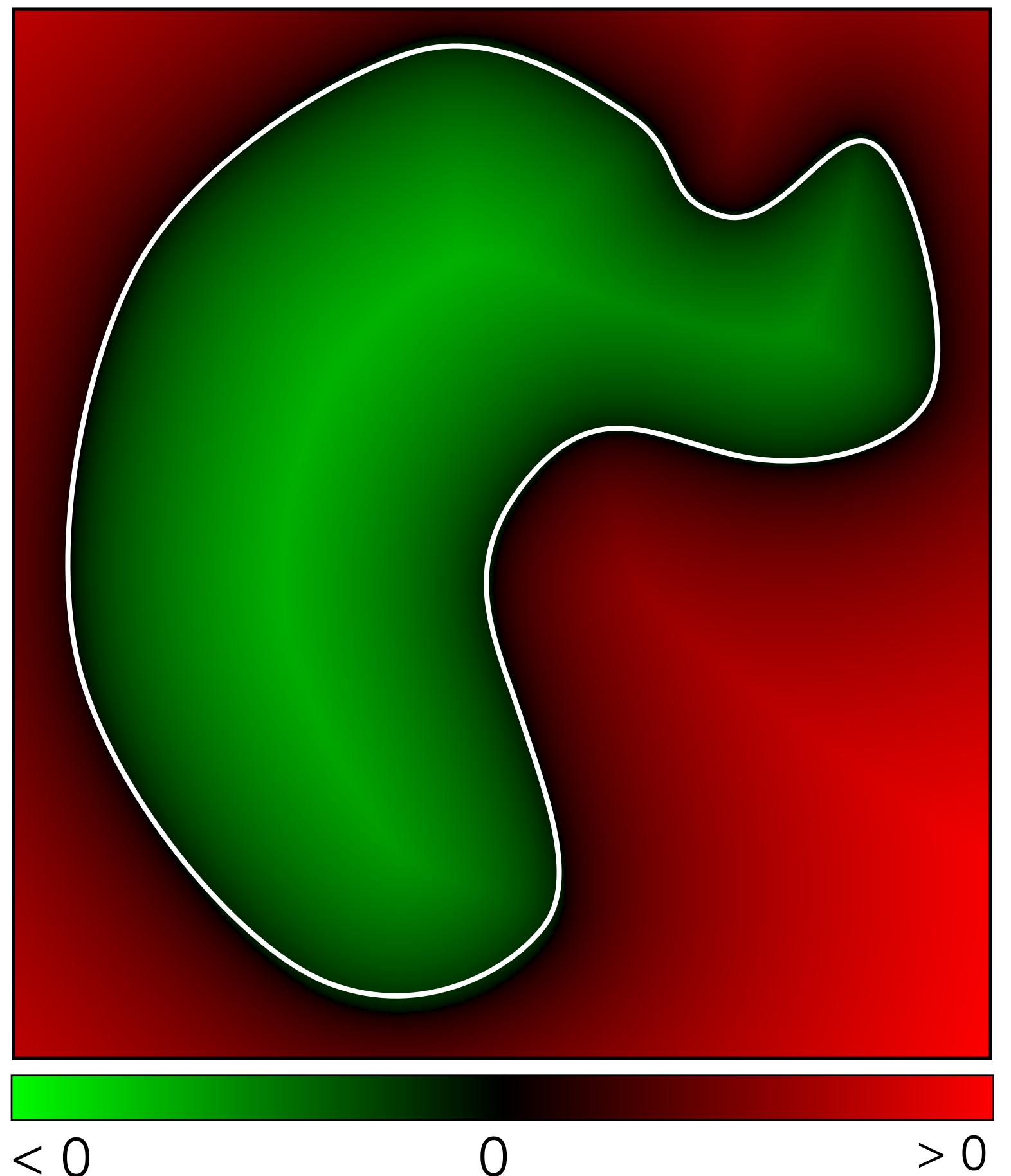
- Define a function

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

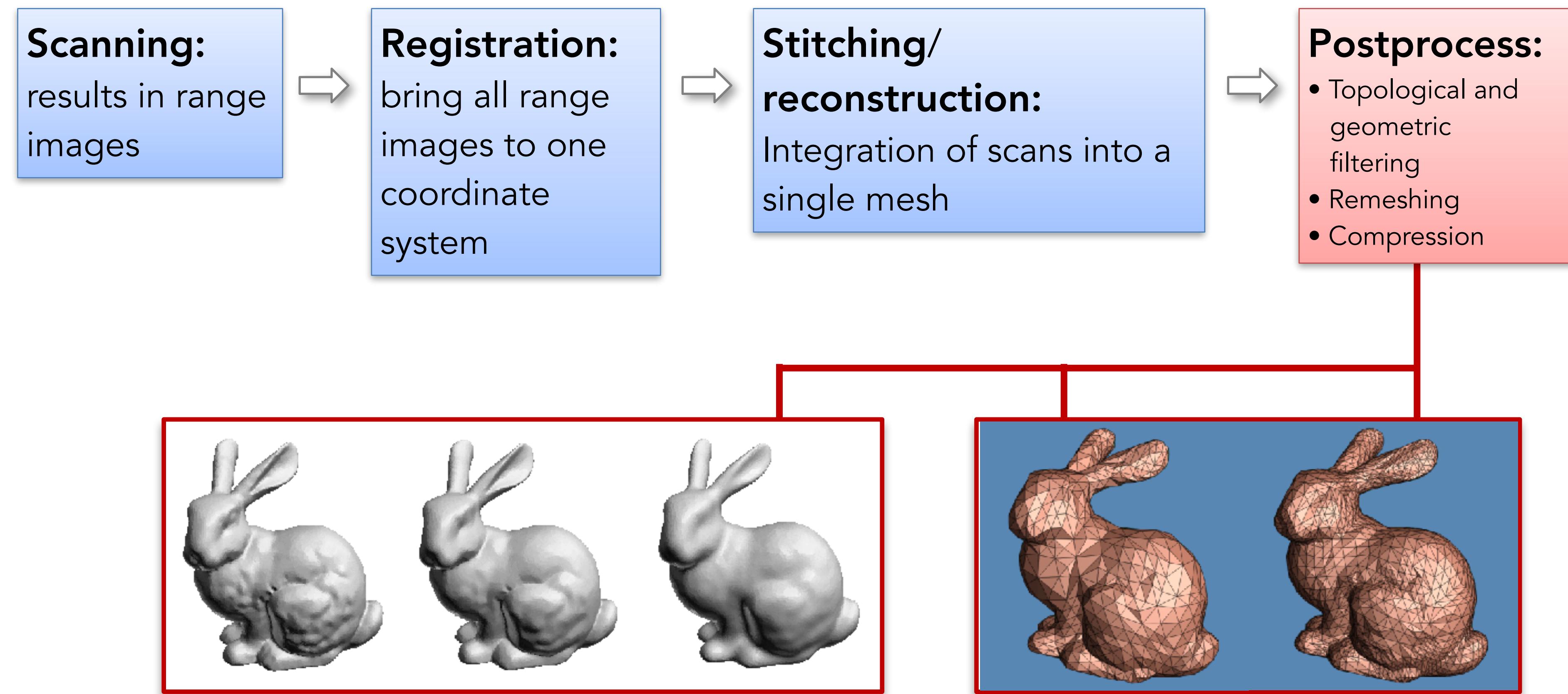
with value  $> 0$  outside the shape and  
 $< 0$  inside

- Extract the zero-set

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



# Geometry Acquisition Pipeline



# Scanning

# Touch Probes



# Touch Probes

- Physical contact with the object
- Manual or computer-guided
- Advantages:
  - Can be very precise
  - Can scan any solid surface
- Disadvantages:
  - Slow, small scale
  - Cannot be used on fragile objects



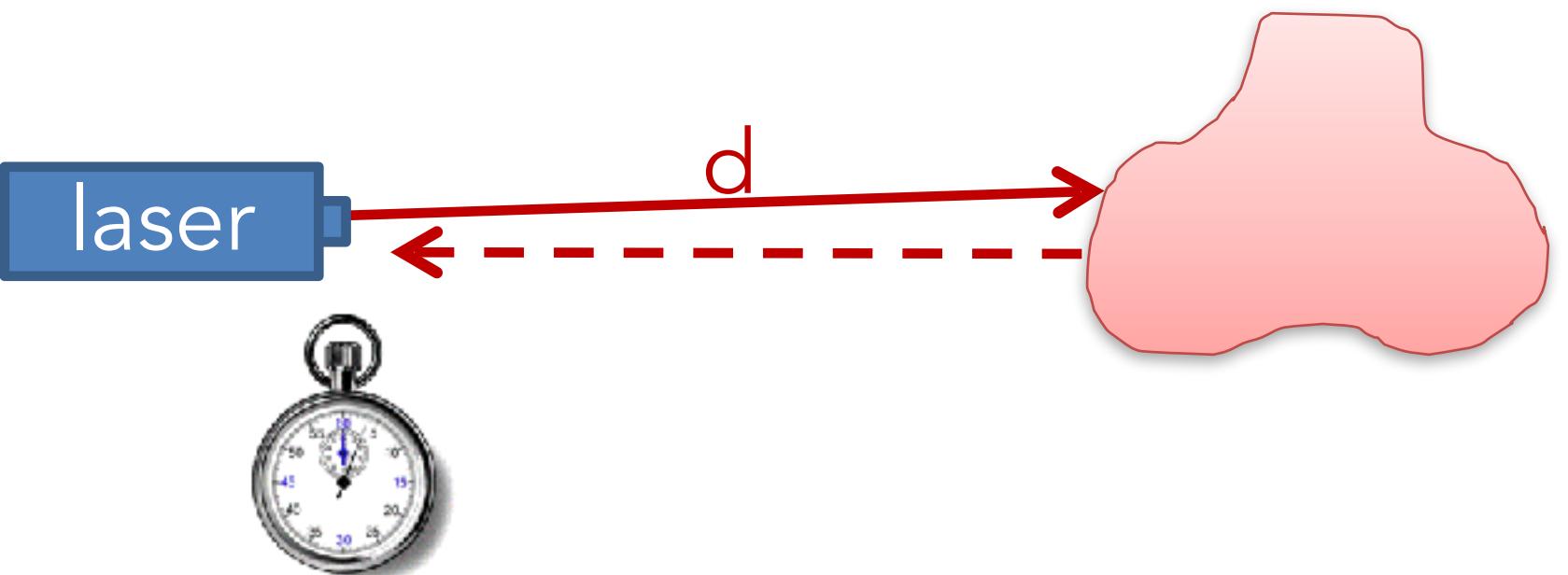
# Optical Scanning

- Infer the geometry from light reflectance
- Advantages:
  - Less invasive than touch
  - Fast, large scale possible
- Disadvantages:
  - Difficulty with transparent and shiny objects



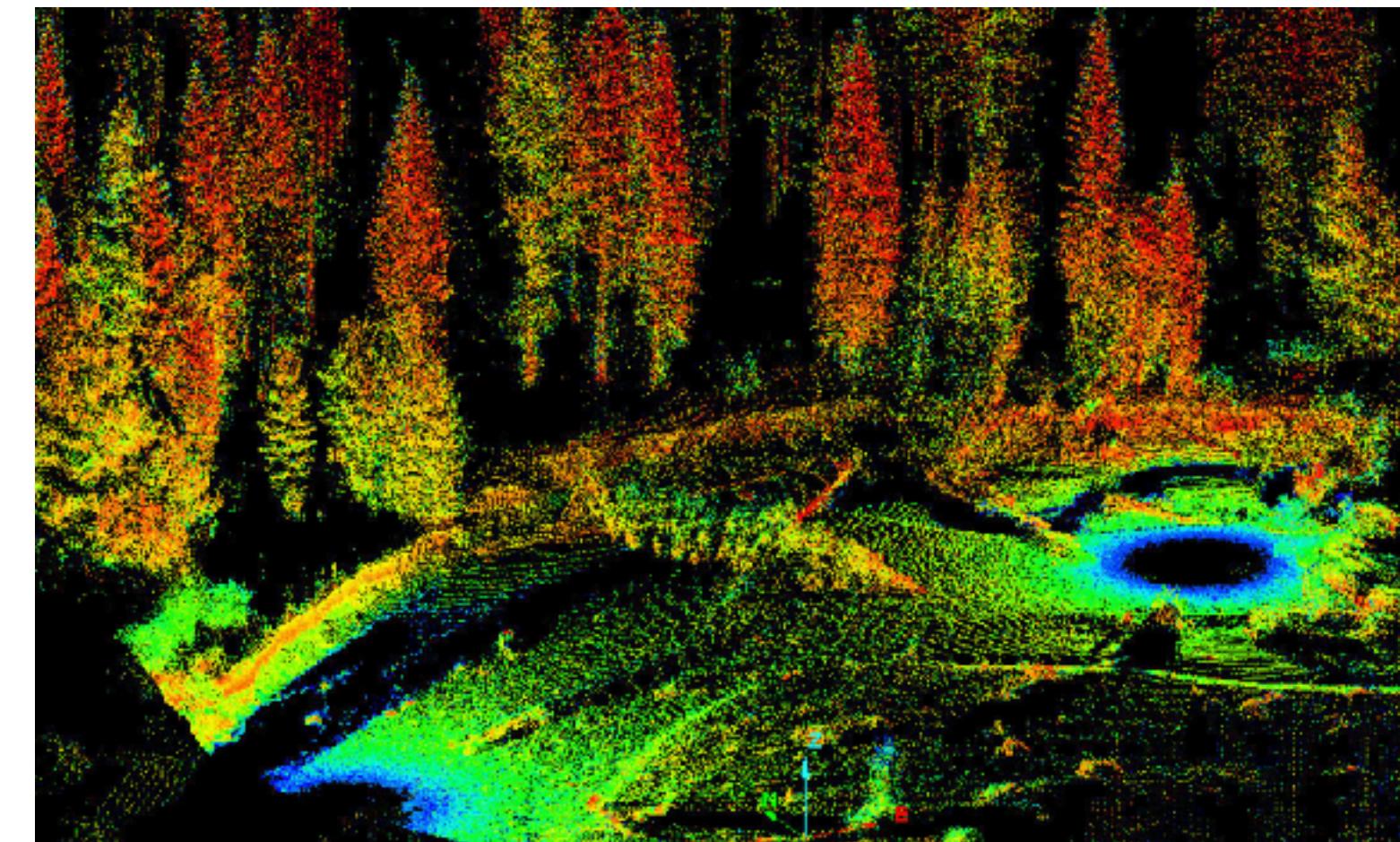
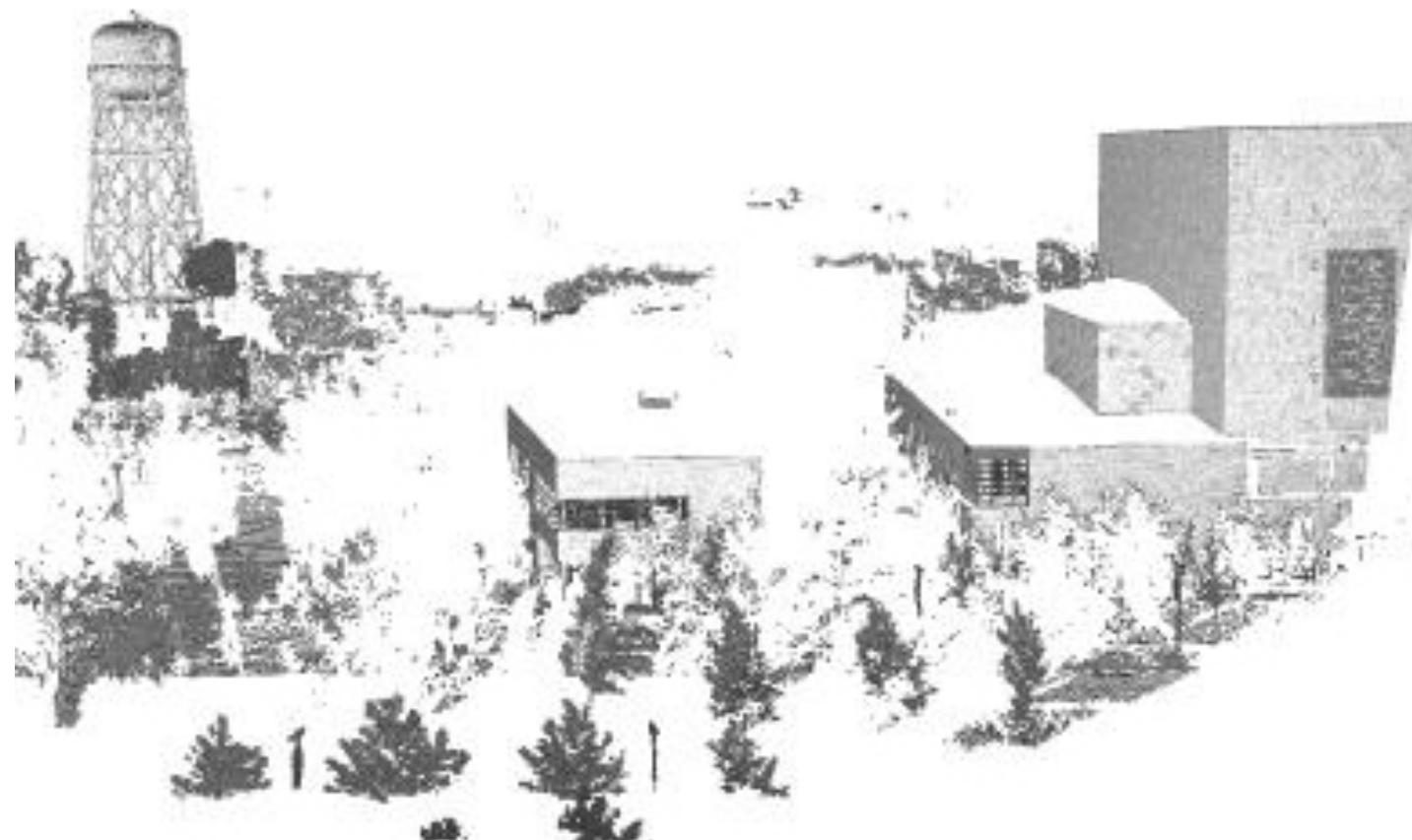
# Time of Flight Scanner

- A type of laser rangefinder (LIDAR)
- Measures the time it takes the laser beam to hit the object and come back



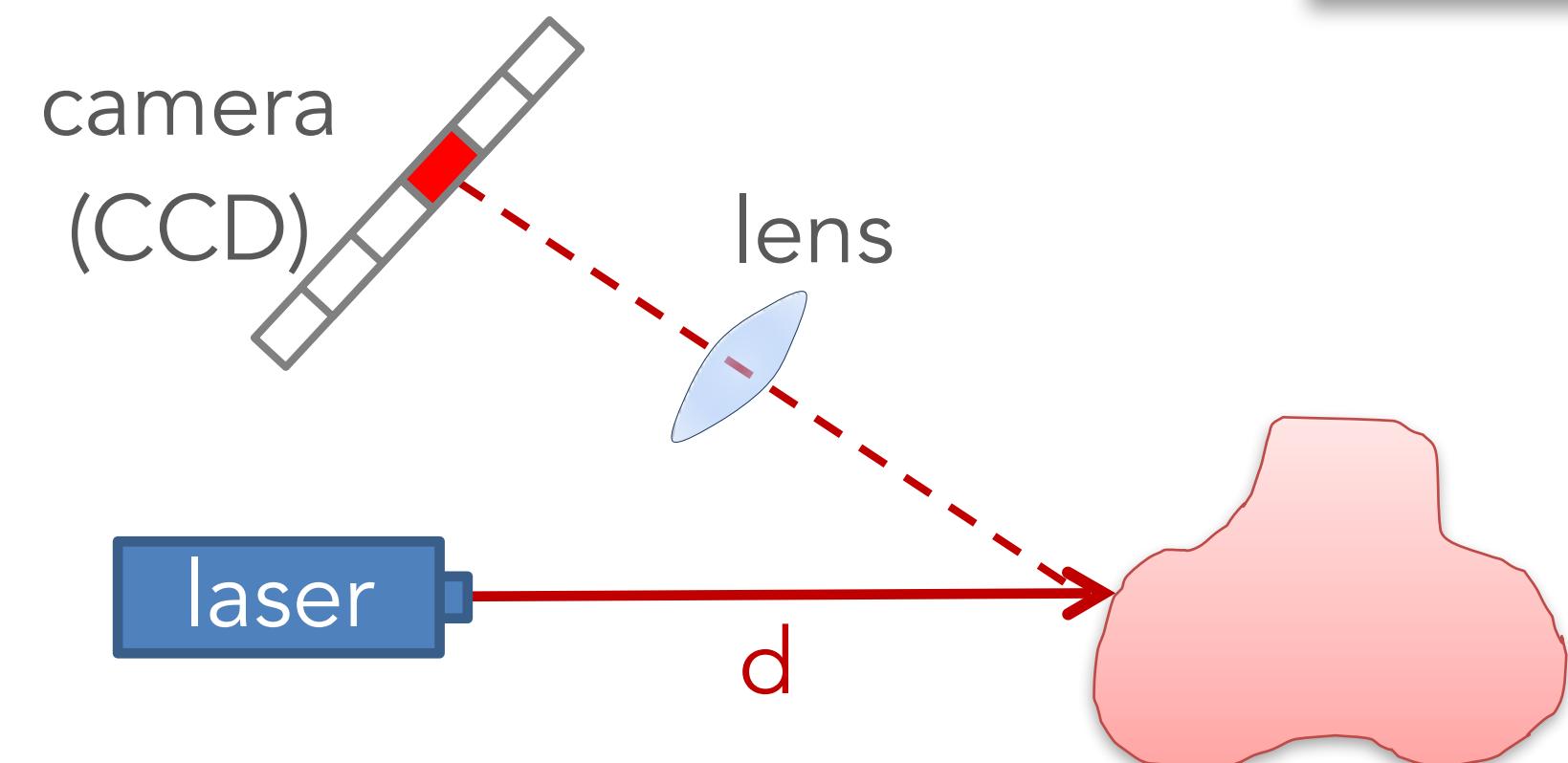
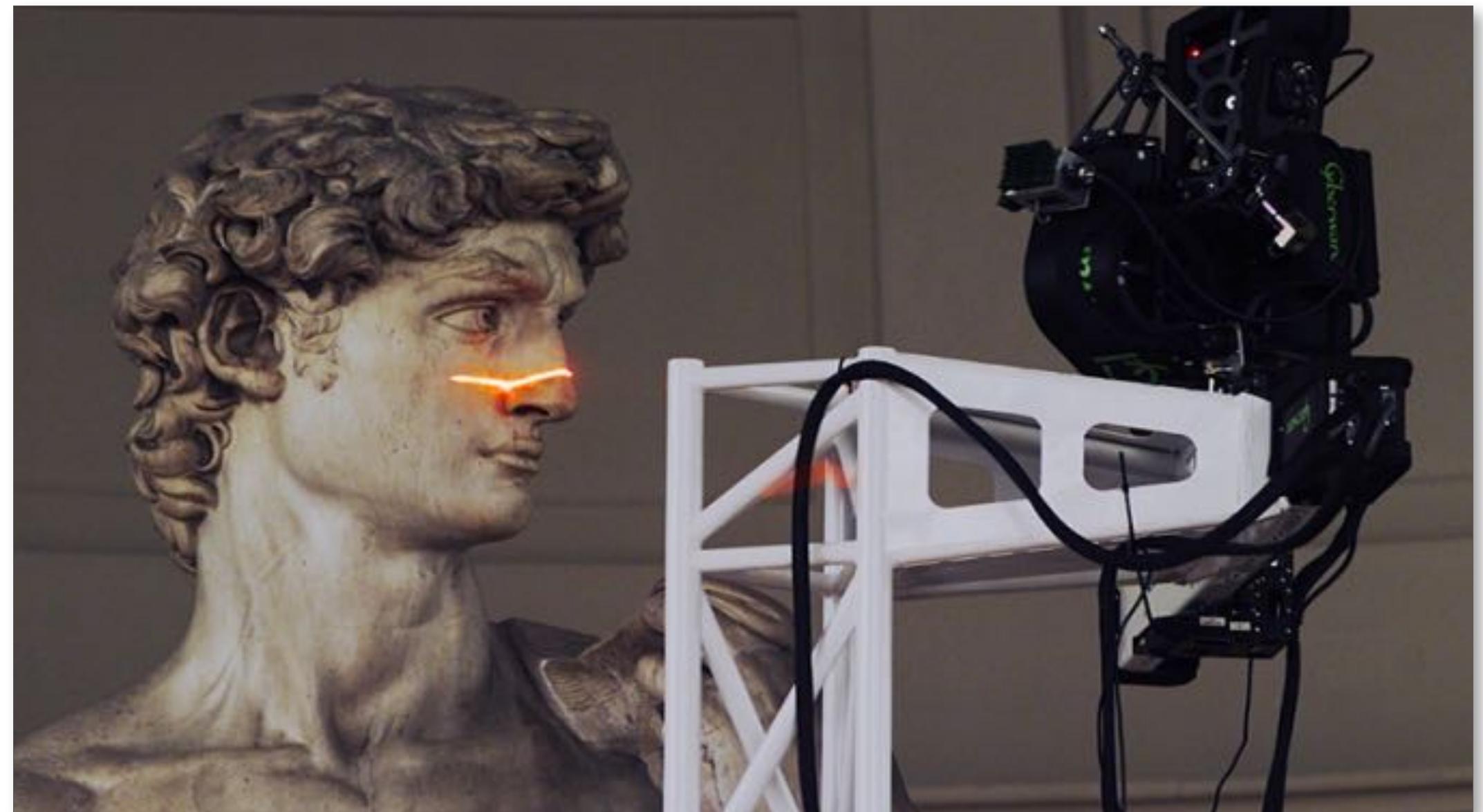
# Time of Flight Scanner

- Accommodates large range – up to several miles (suitable for buildings, rocks)
- Only for static scenes, object motion introduces noise



# Triangulation Laser

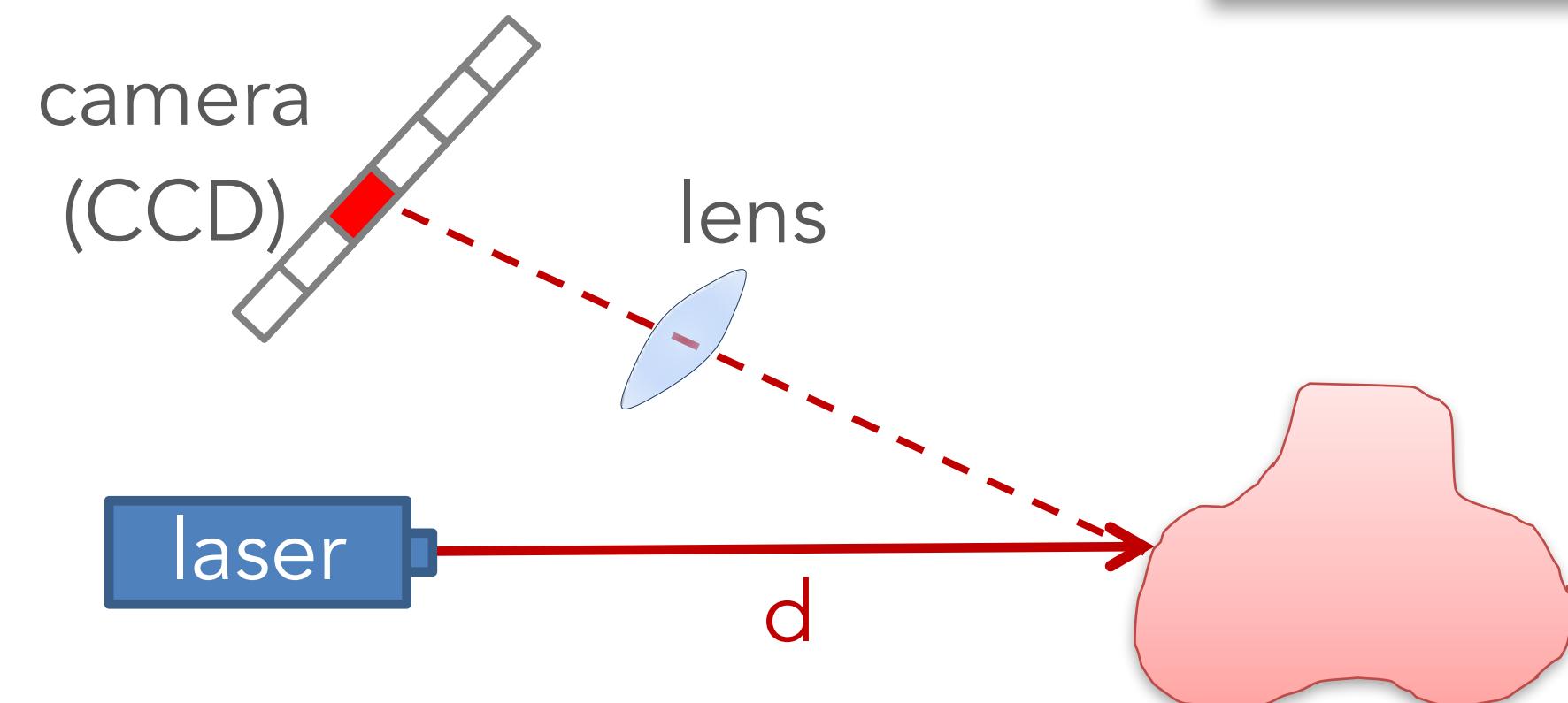
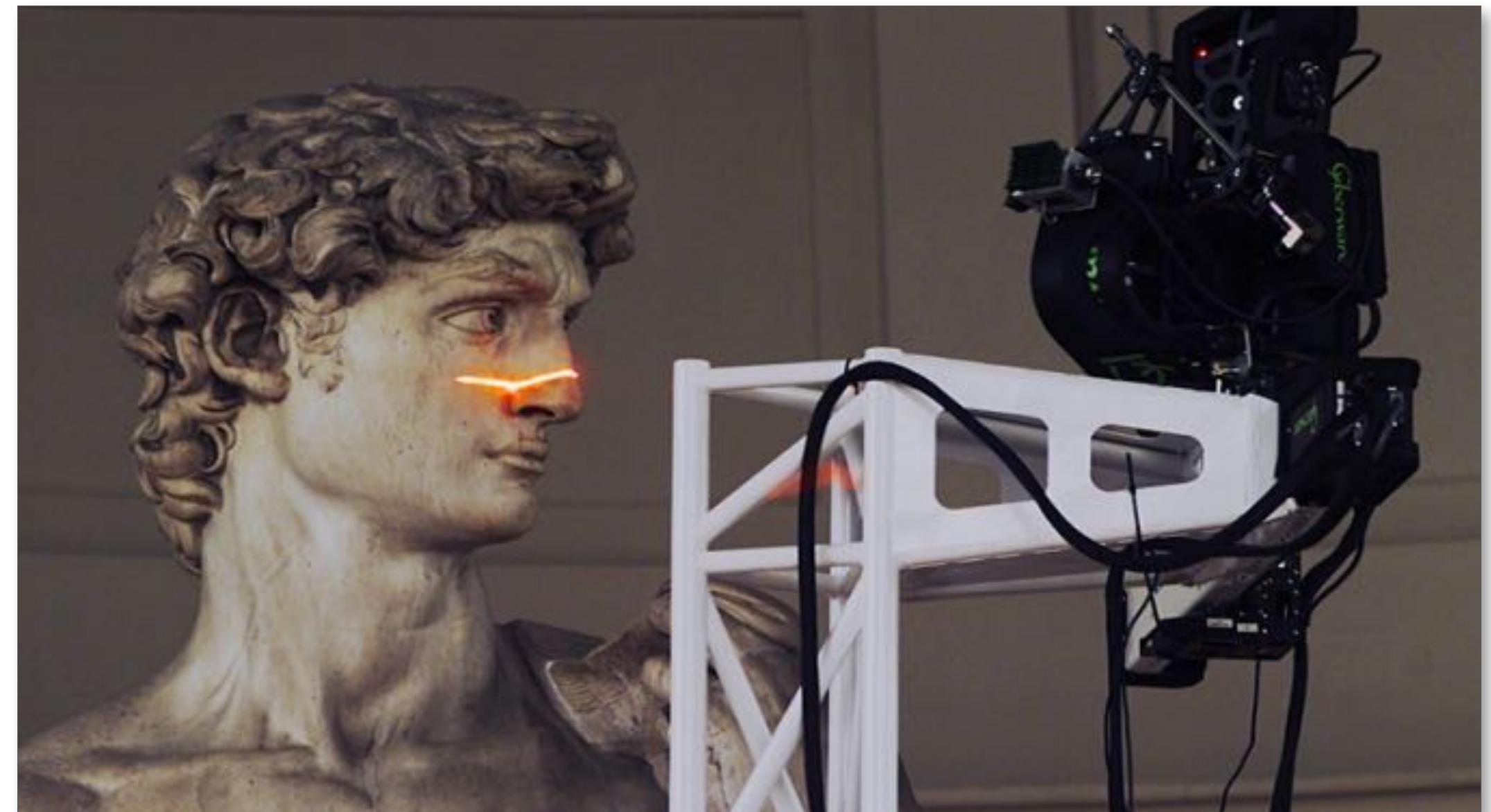
- Laser beam and camera
- Laser dot is photographed
- The location of the dot in the image allows triangulation:  
we get the distance to the object



Digital Michelangelo Project

# Triangulation Laser

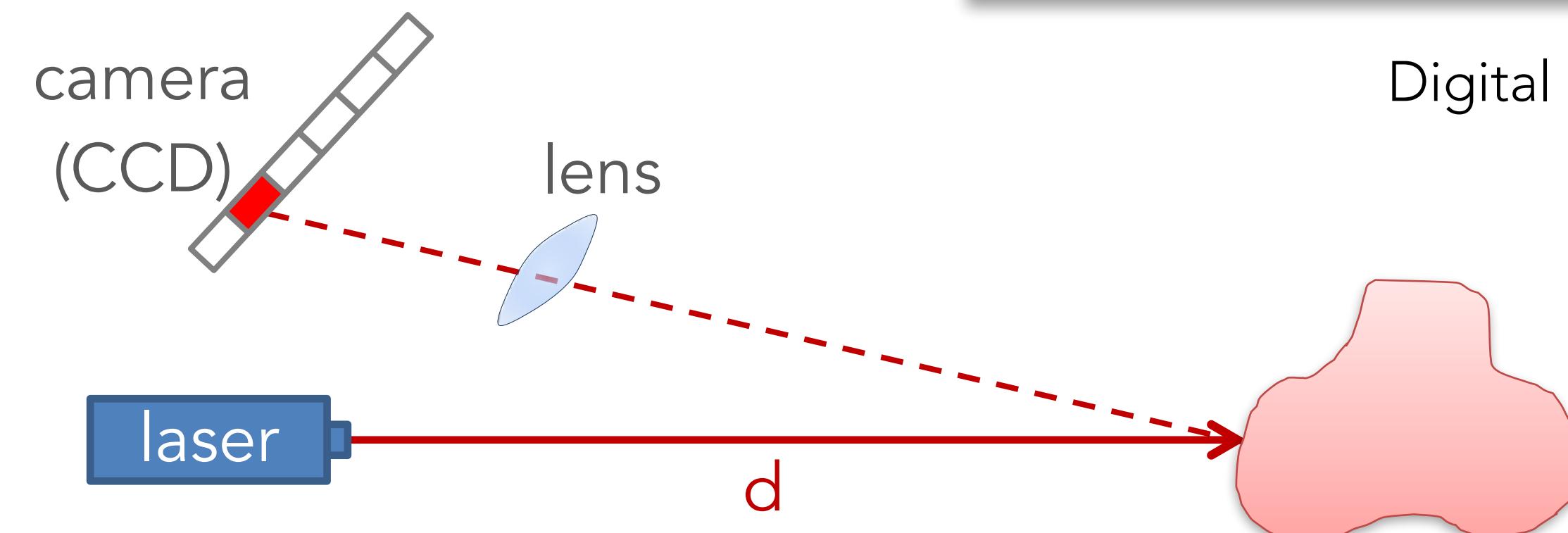
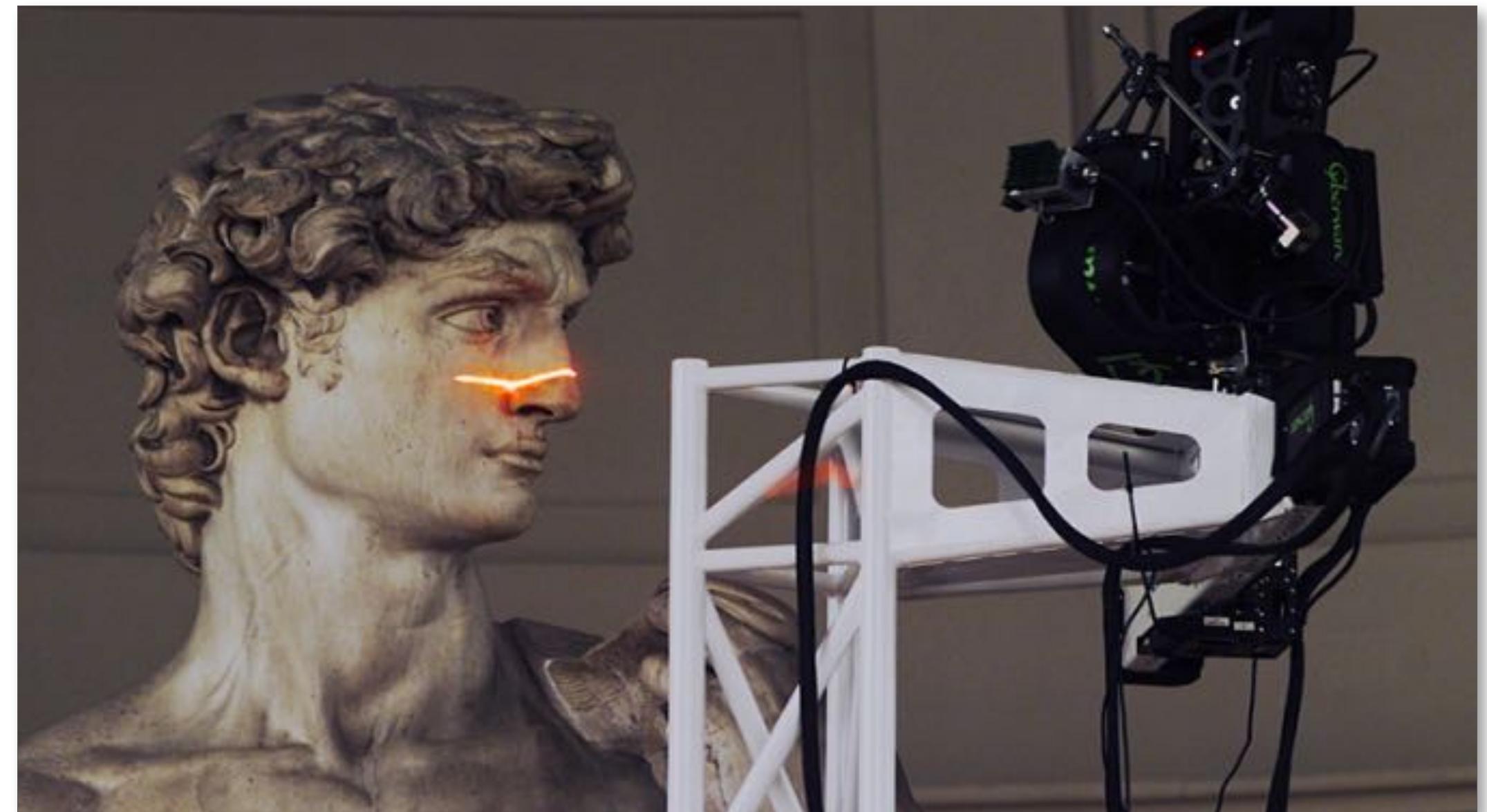
- Laser beam and camera
- Laser dot is photographed
- The location of the dot in the image allows triangulation:  
we get the distance to the object



Digital Michelangelo Project

# Triangulation Laser

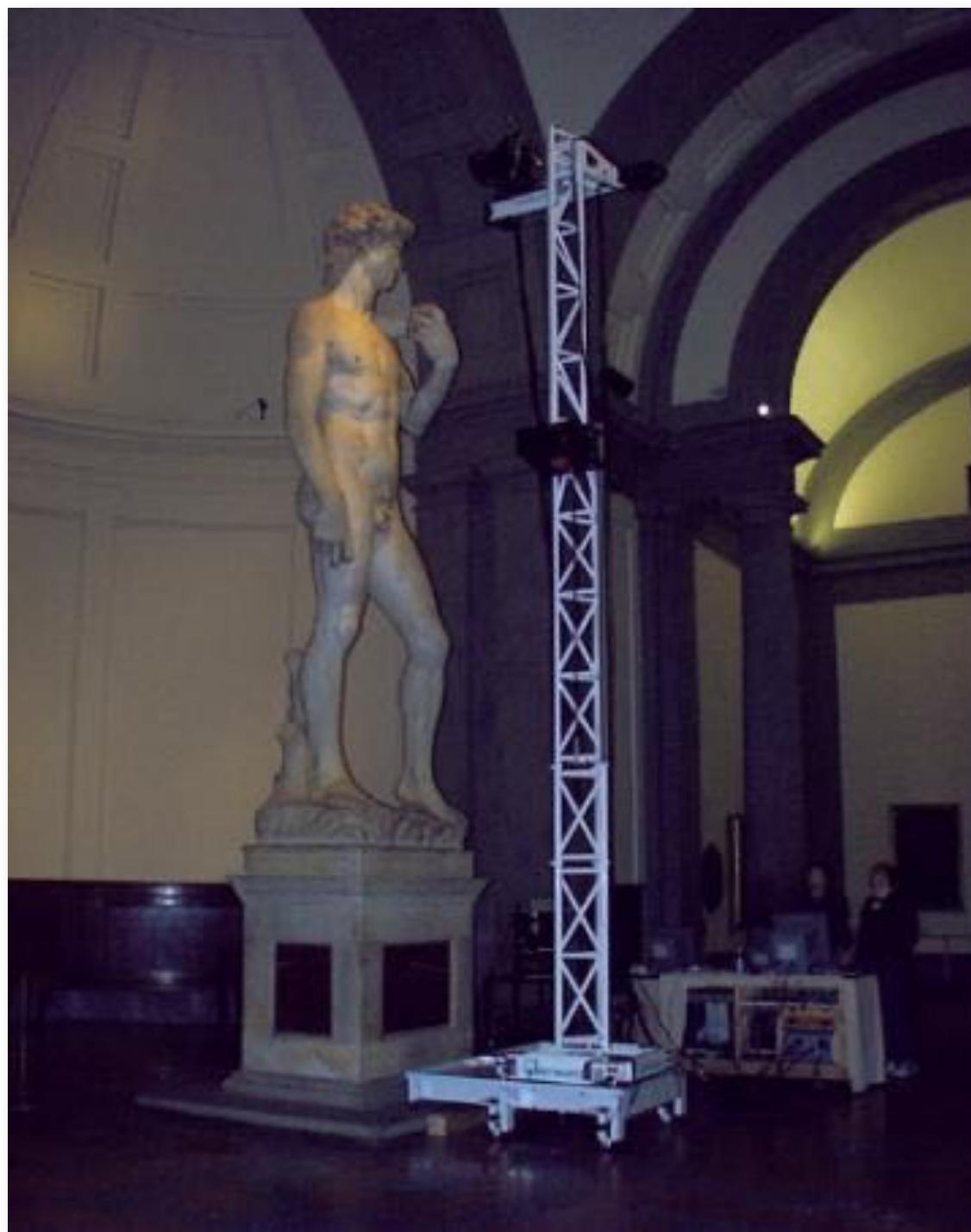
- Laser beam and camera
- Laser dot is photographed
- The location of the dot in the image allows triangulation:  
we get the distance to the object



Digital Michelangelo Project

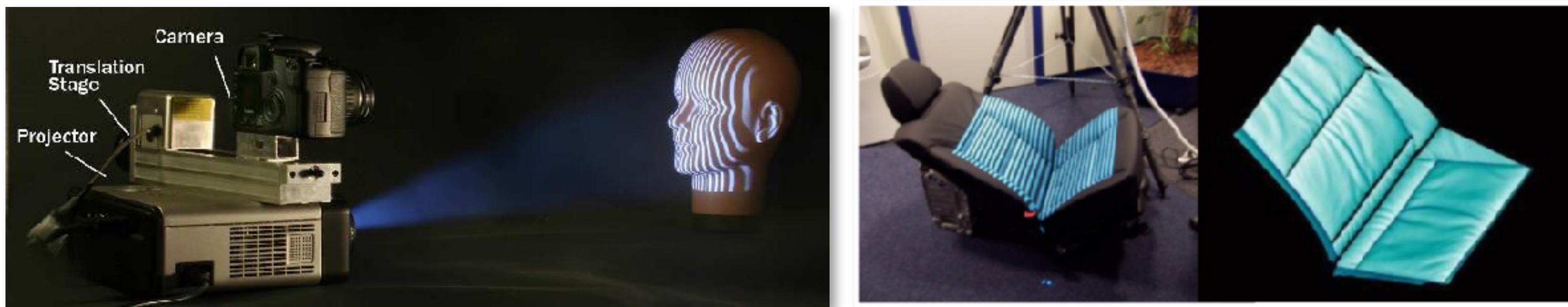
# Triangulation Laser

- Very precise (tens of microns)
- Small distances (meters)



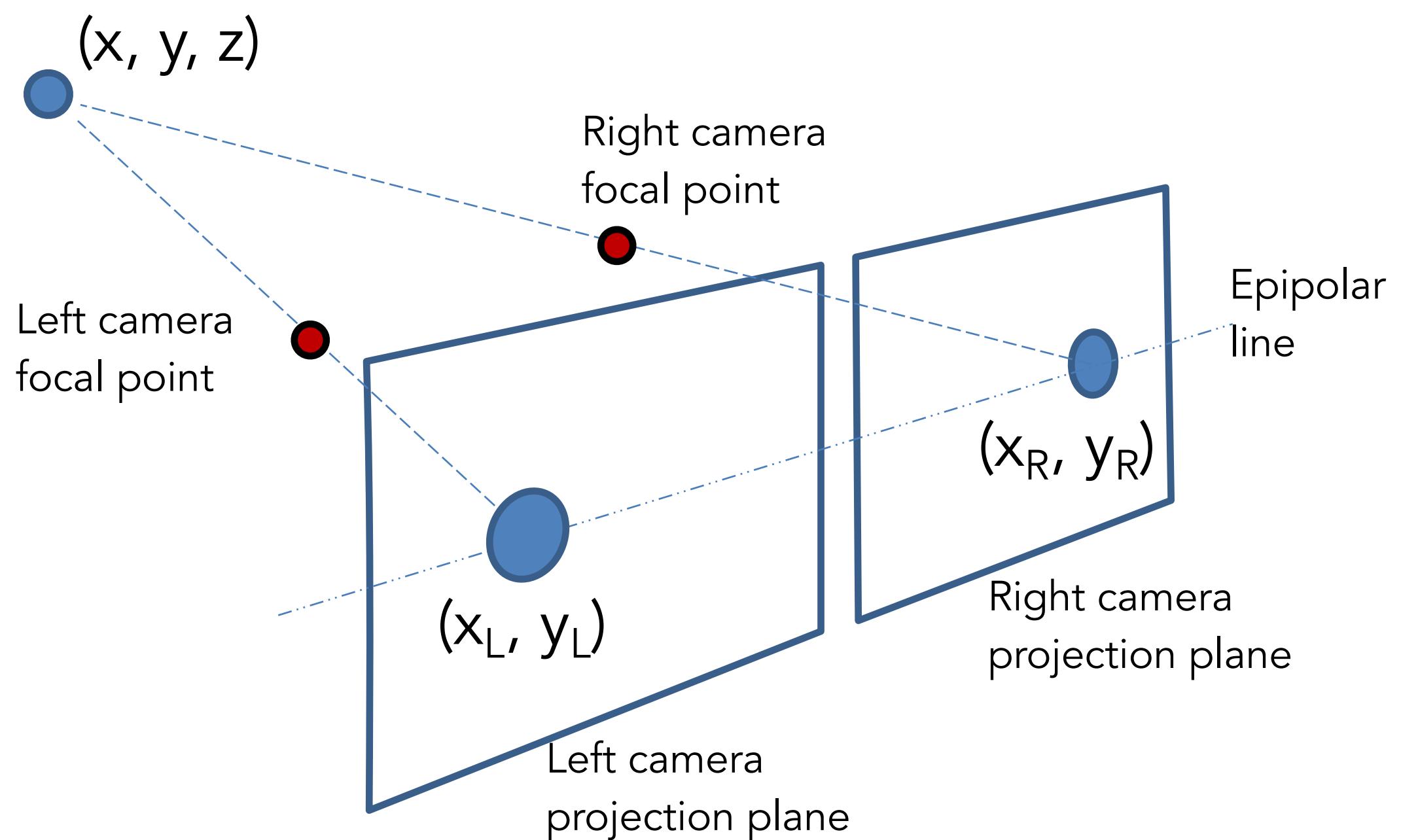
# Structured Light

- Pattern of visible or **infrared** light is projected onto the object
- The distortion of the pattern, recorded by the camera, provides geometric information
- Very fast – 2D pattern at once
  - Even in real time, like Kinect 1.0
- Complex distance calculation, prone to noise



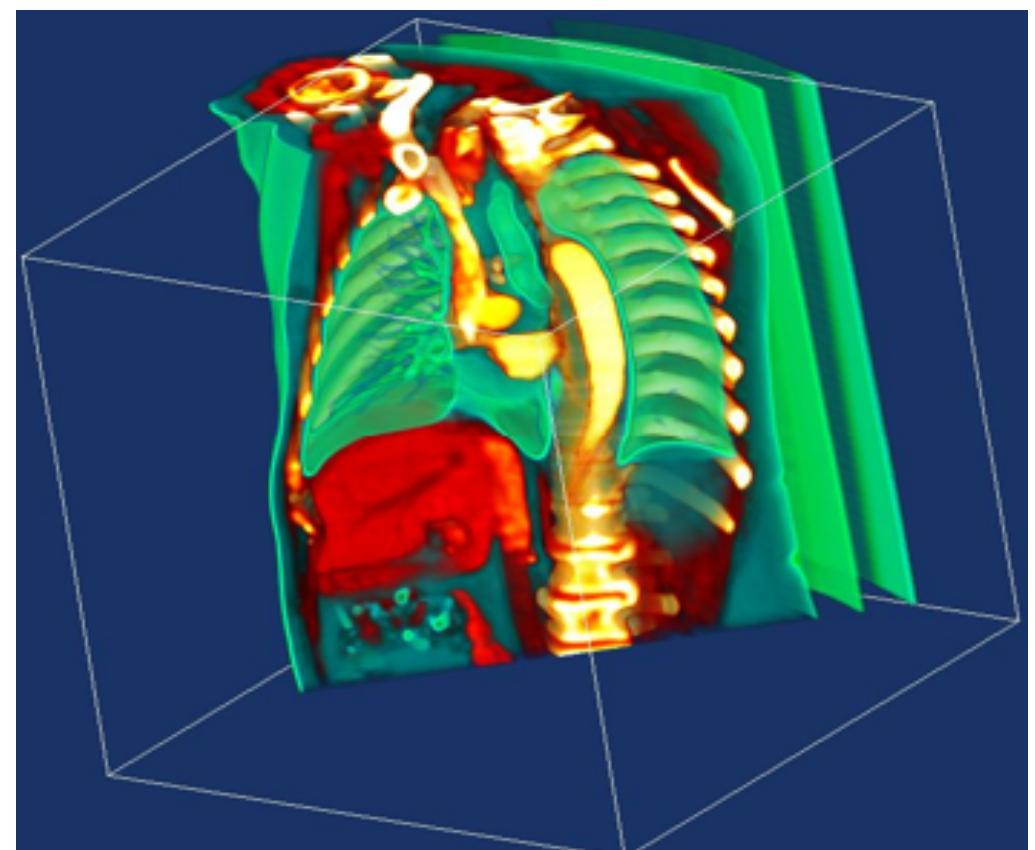
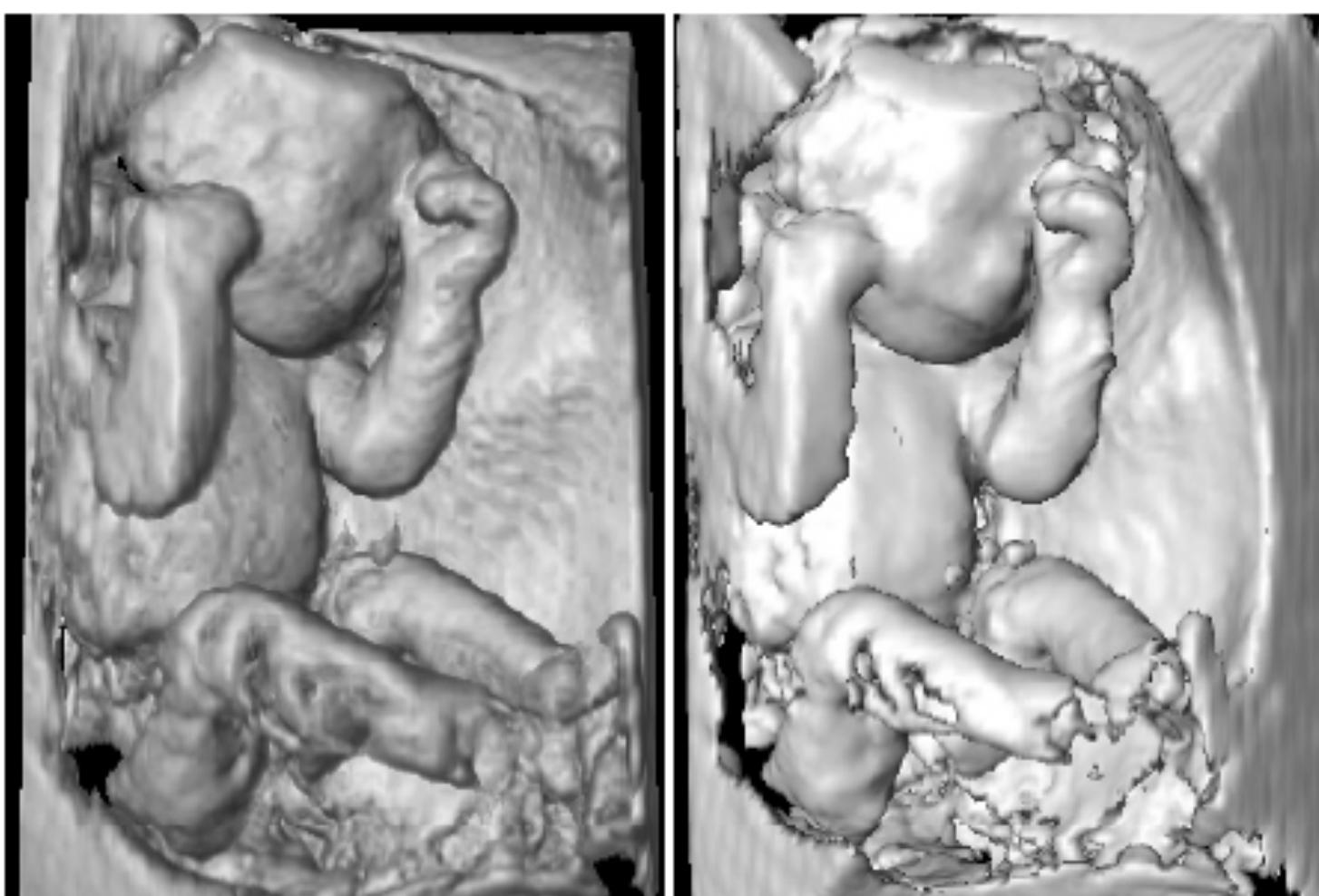
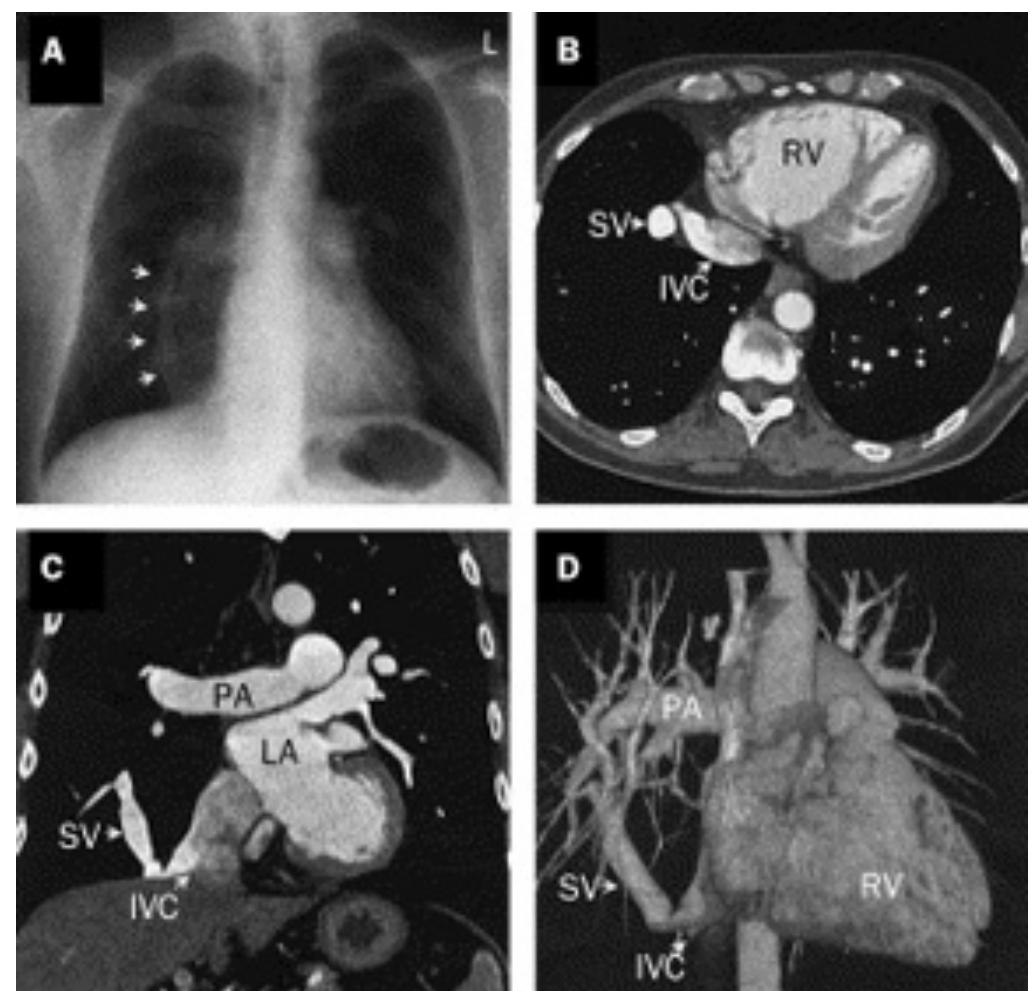
# Passive Stereo

- No need for special lighting/radiation
- Two (or more) cameras
- Feature matching and triangulation



# Medical Imaging

- Ultrasound, CT, MRI
- Discrete volume of density data
- Requires segmenting the desired object (contouring)

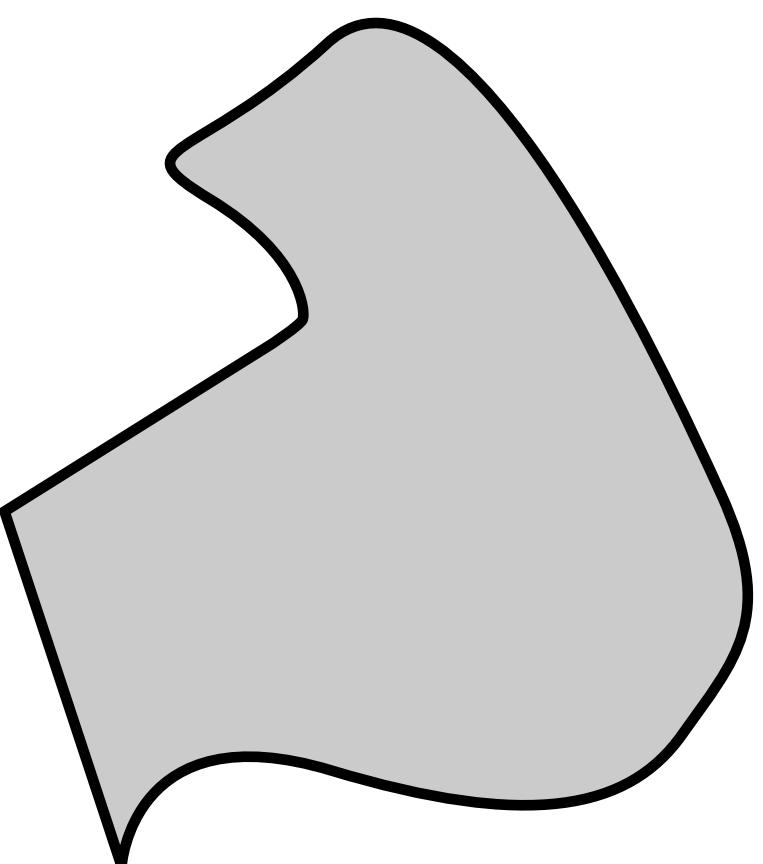


# Registration

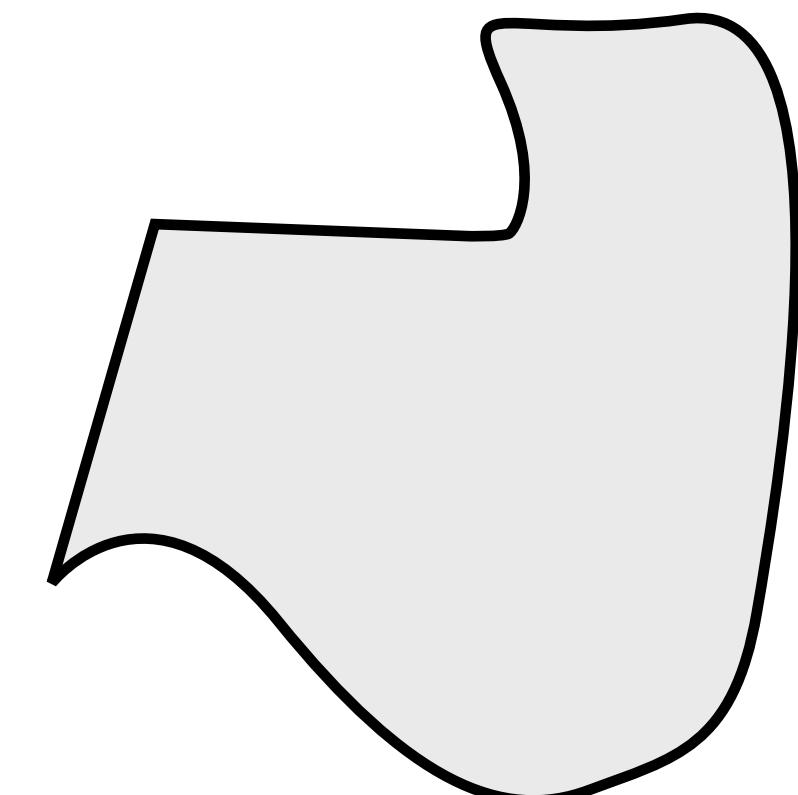
Acknowledgement: Niloy Mitra  
[http://resources.mpi-inf.mpg.de/deformableShapeMatching/EG2012\\_Tutorial/](http://resources.mpi-inf.mpg.de/deformableShapeMatching/EG2012_Tutorial/)  
CSC 486B/586B - Geometric Modeling - Teseo Schneider

# Problem Statement

$M_1$



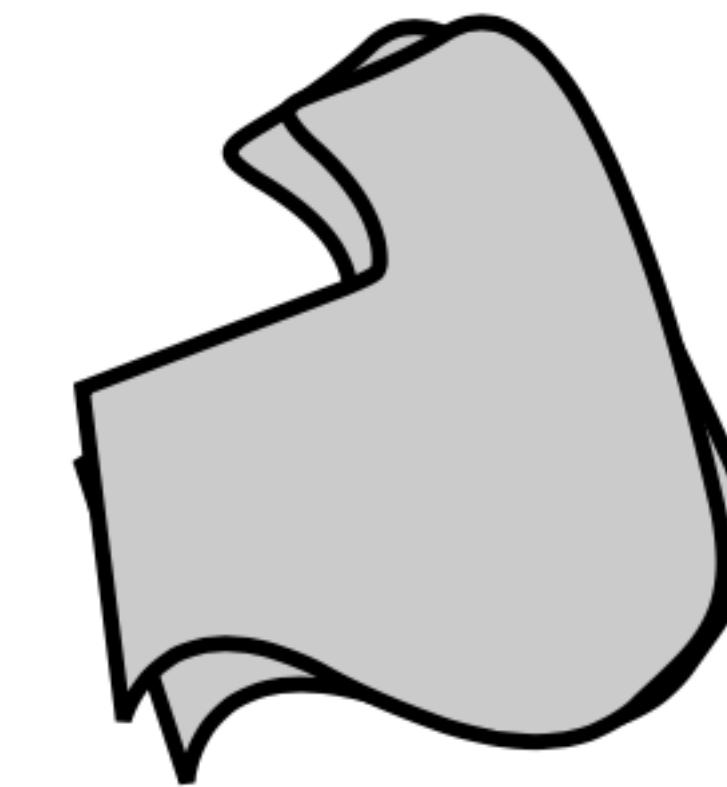
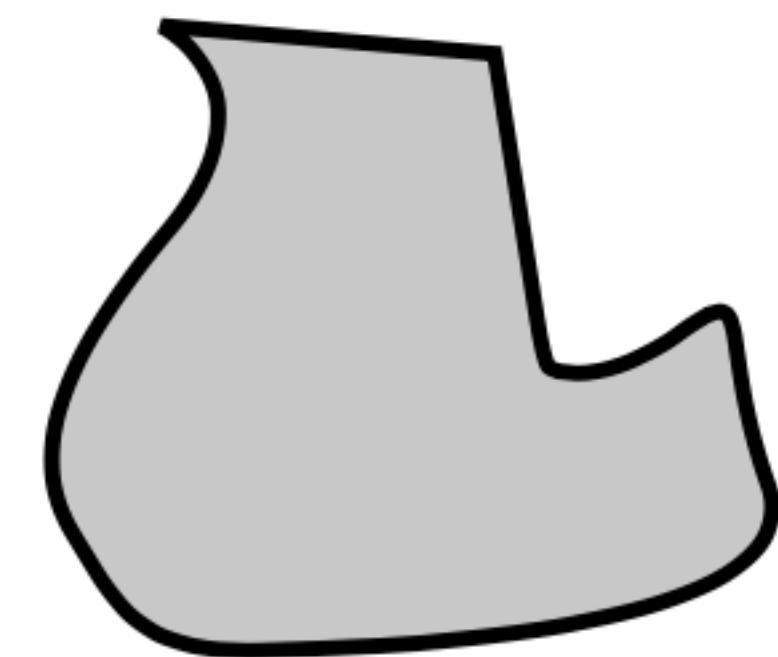
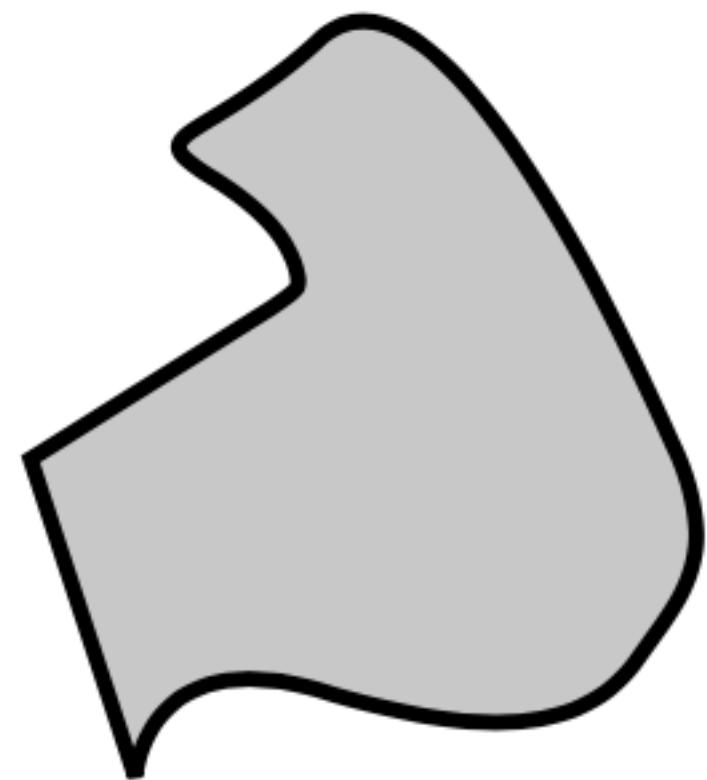
$M_2$



$$M_1 \approx T(M_2)$$

T: Translation + Rotation

# Local vs Global



## Global Registration

Arbitrary Transformation

## Local Registration

“Small” Transformation

Given  $M_1, \dots, M_n$ , find  $T_2, \dots, T_n$  such that

$$M_1 \approx T_2(M_2) \cdots \approx T_n(M_n)$$



University  
of Victoria

Computer Science

# Correspondences

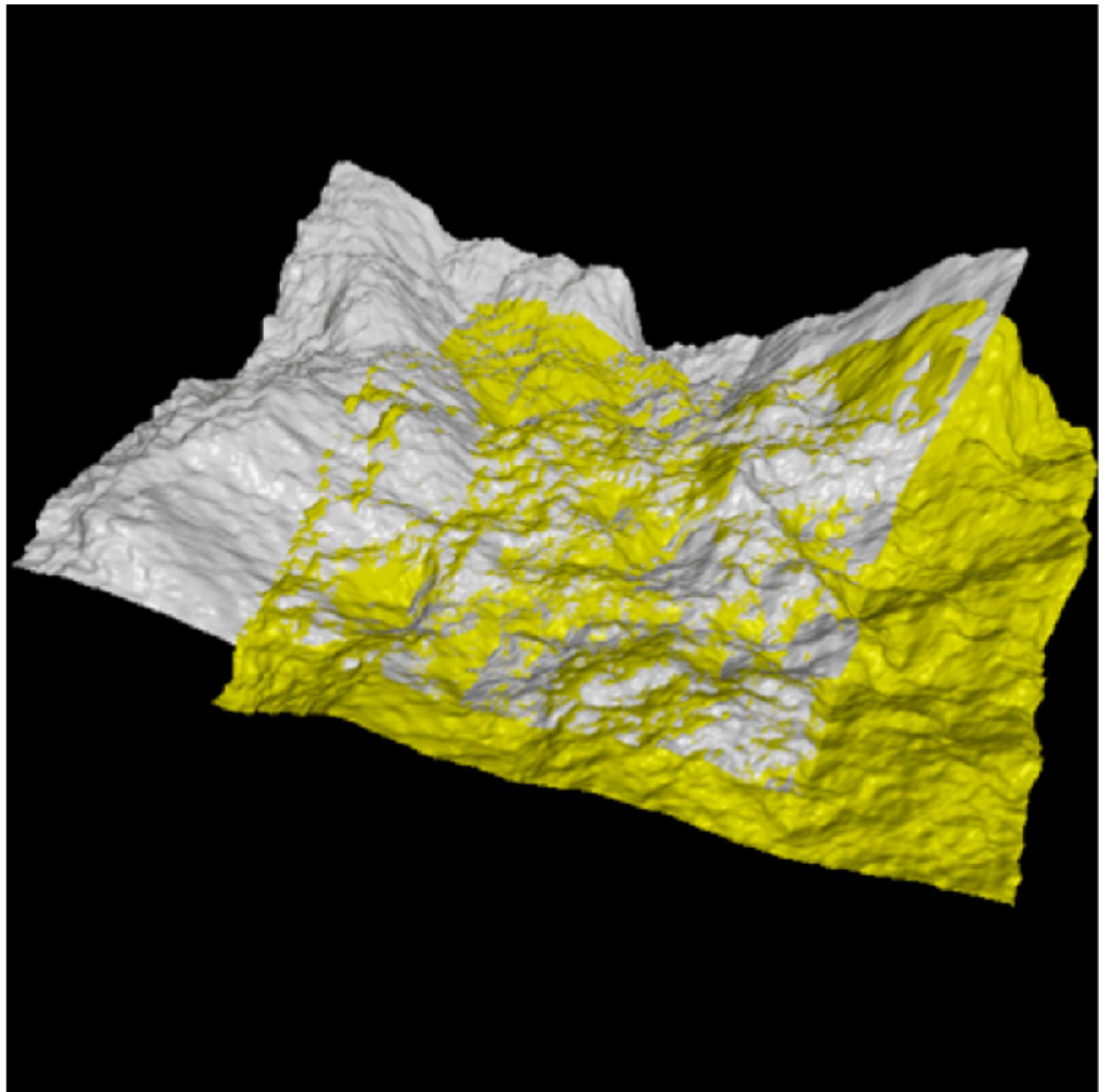
- How many points are needed to define a unique rigid transformation?
- The first problem is finding corresponding pairs!

$$\mathbf{p}_1 \rightarrow \mathbf{q}_1$$

$$\mathbf{p}_2 \rightarrow \mathbf{q}_2$$

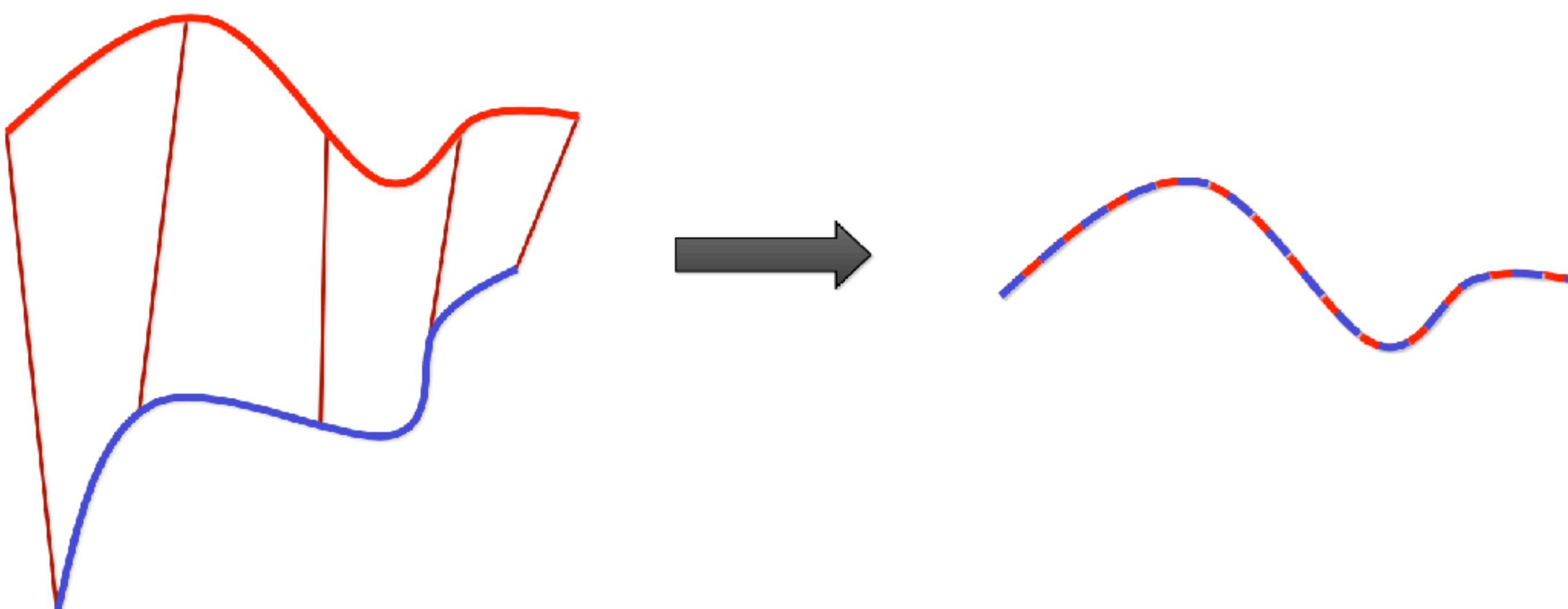
$$\mathbf{p}_3 \rightarrow \mathbf{q}_3$$

$$R\mathbf{p}_i + t \approx \mathbf{q}_i$$



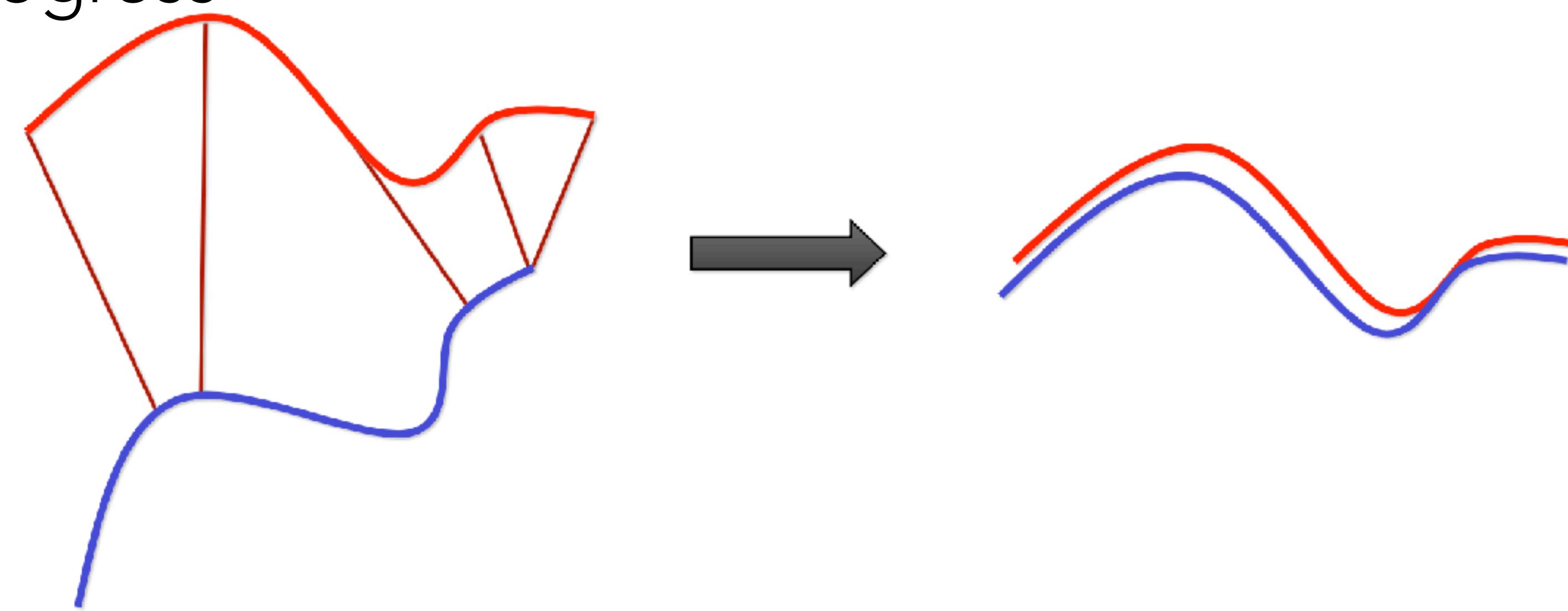
# ICP: Iterative Closest Point

- Idea: Iteratively (1) find correspondences and (2) use them to find a transformation
- Intuition: If you have the right correspondences, then the problem is easy

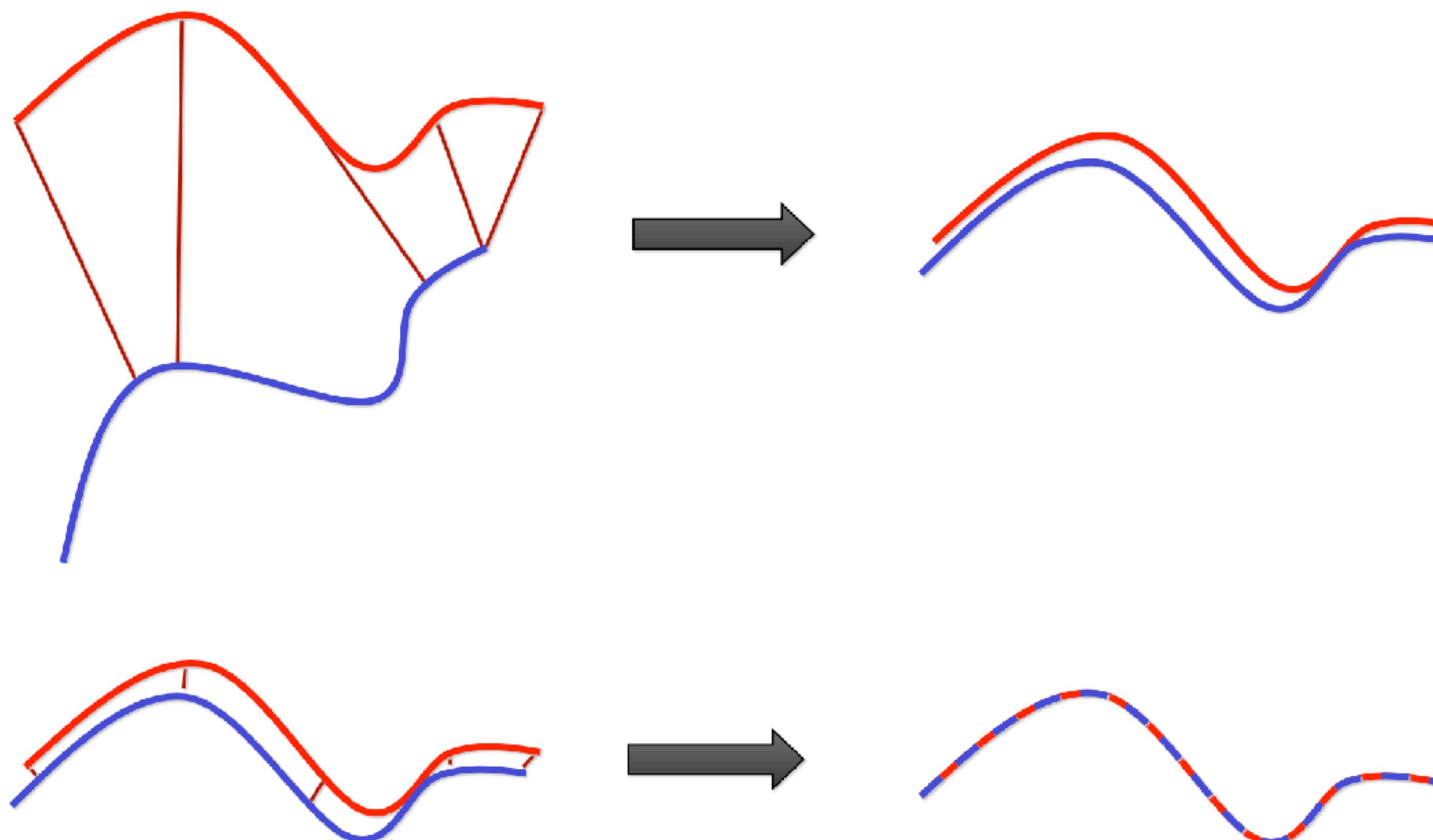


# ICP: Iterative Closest Point

- Idea: Iteratively (1) find correspondences and (2) use them to find a transformation
- Intuition: If you don't have the right correspondences, you still can make progress



# ICL: Iterative Closest Point



This algorithm converges to the correct solution only  
if the starting scans are “close enough”



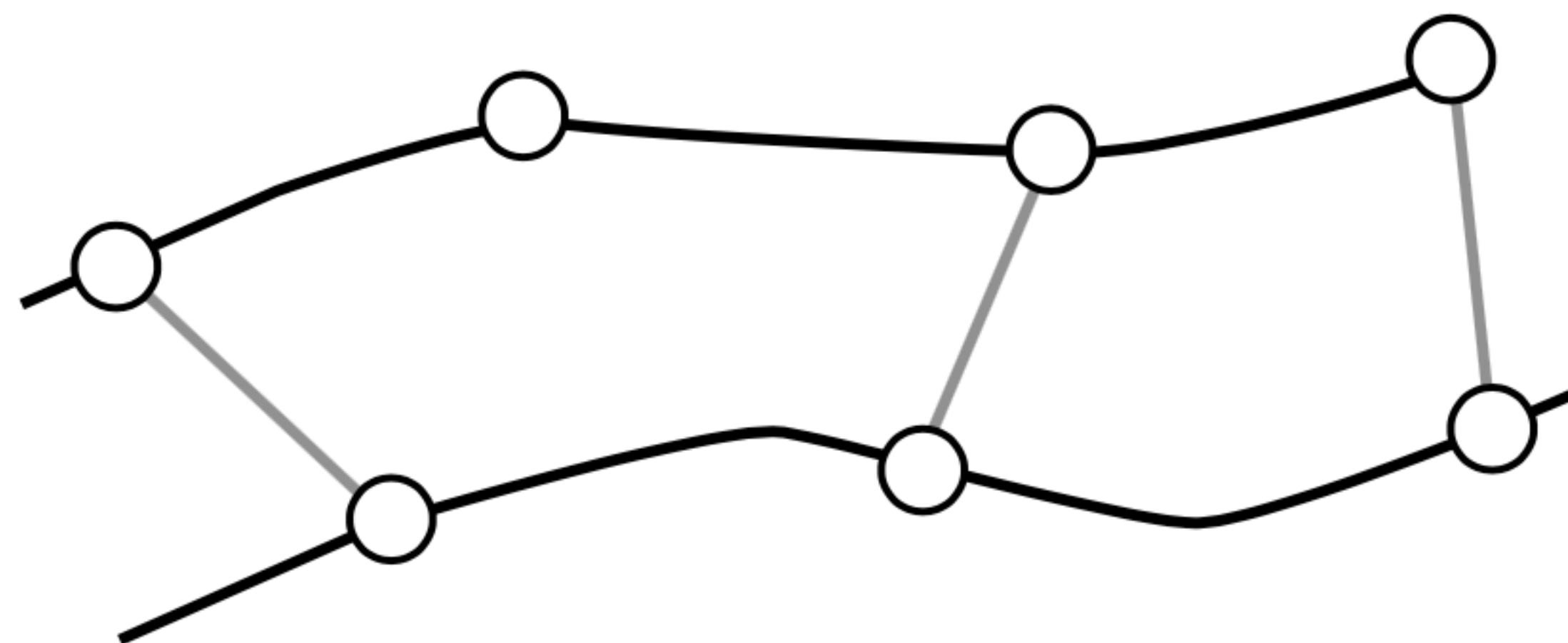
University  
of Victoria

Computer Science

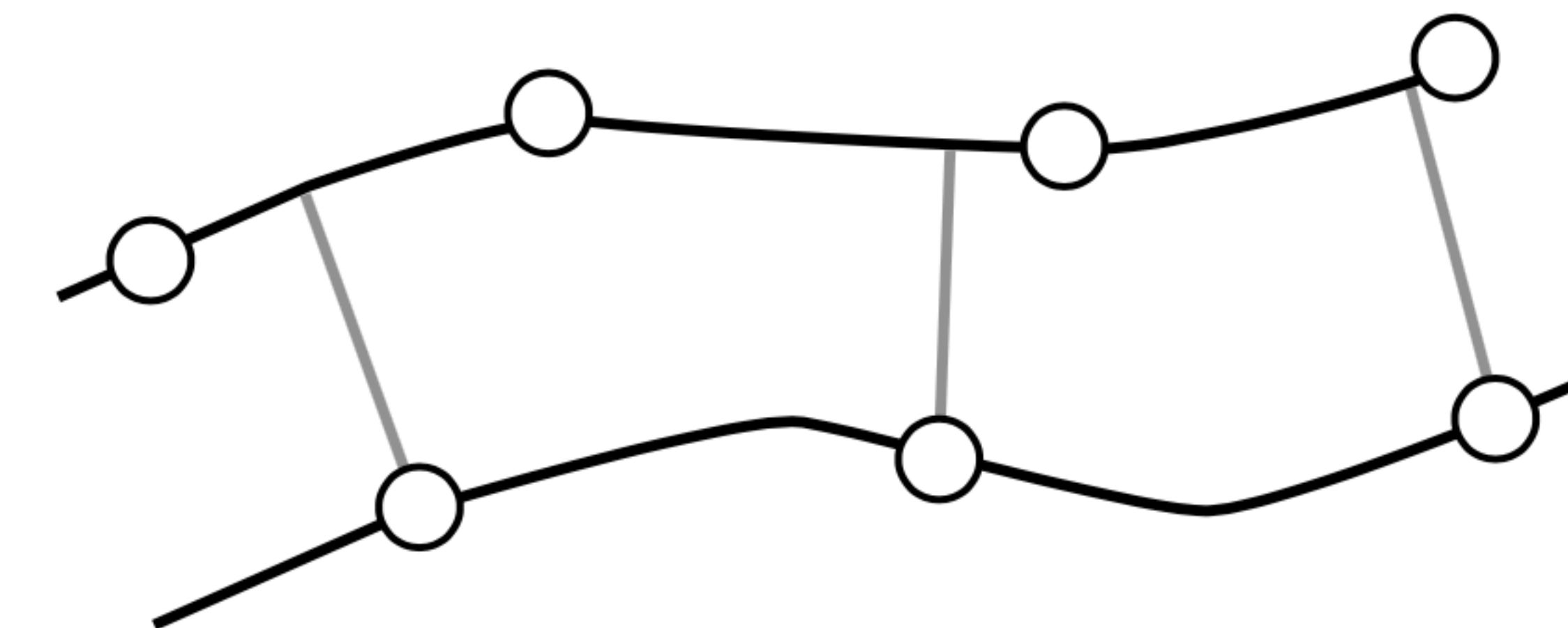
# Basic Algorithm

- Select (e.g., 1000) random points
- Match each to closest point on other scan, using a spatial data structure
- Reject pairs with distance  $> k$  times median
- Construct error function:  
$$E := \sum_i (R\mathbf{p}_i + t - \mathbf{q}_i)^2$$
- Minimize (closed form solution in “Estimating 3-D rigid body transformations: a comparison of four major algorithms”, <http://dl.acm.org/citation.cfm?id=250160>)  
.

# Important Variant



Point-to-Point



Point-to-Plane

See [http://resources.mpi-inf.mpg.de/deformableShapeMatching/EG2012\\_Tutorial/](http://resources.mpi-inf.mpg.de/deformableShapeMatching/EG2012_Tutorial/) for details

# References

- [http://resources.mpi-inf.mpg.de/deformableShapeMatching/  
EG2012\\_Tutorial/](http://resources.mpi-inf.mpg.de/deformableShapeMatching/EG2012_Tutorial/)
- <http://mesh.brown.edu/byo3d/>