03 - Acquisition and registration

Acknowledgements: Daniele Panozzo, Marco Tarini

In this lecture

- How to acquire 3D data from real objects
- How to obtain a point cloud

- Shape matching problem

Geometry Acquisition Pipeline

Scanning: results in range images → **Registration:** brings all range images to one coordinate system → **Stitching/ reconstruction:** integration of scans into a single mesh → **Postprocess:**
- Repairing
- Filtering
- Remeshing
- Compression

Geometry Acquisition Pipeline

Scanning: results in range images ⇨ **Registration:** Brings all range images to one coordinate system ⇨ **Stitching/ reconstruction:** Integration of scans into a single mesh ⇨ **Postprocess:**
- Repairing
- Filtering
- Remeshing
- Compression

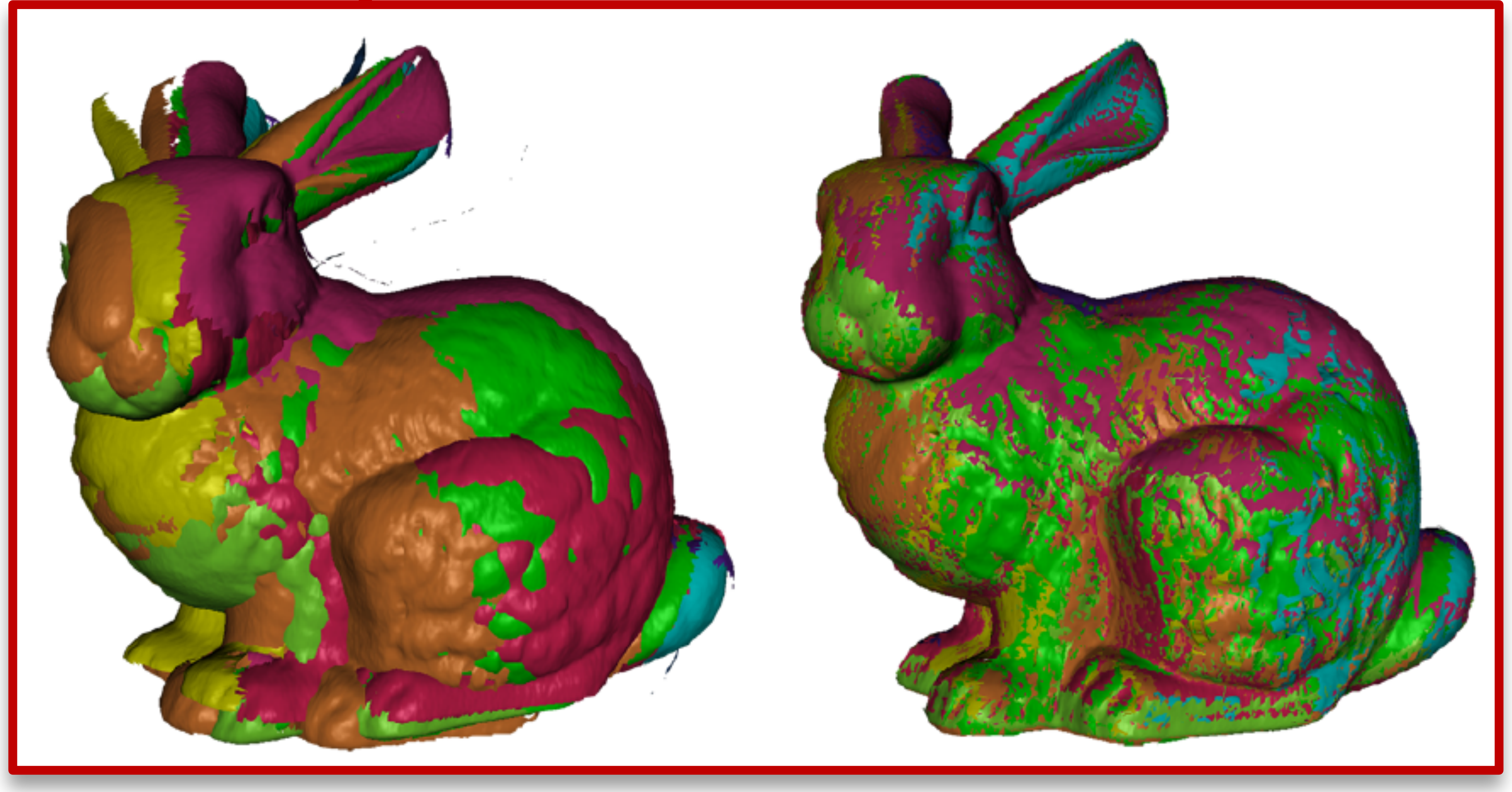
Geometry Acquisition Pipeline

Scanning:
results in range images

Registration:
Brings all range images to one coordinate system

Stitching/ reconstruction:
Integration of scans into a single mesh

Postprocess:
- Repairing
- Filtering
- Remeshing
- Compression



Geometry Acquisition Pipeline

Scanning:
results in range images

⇨

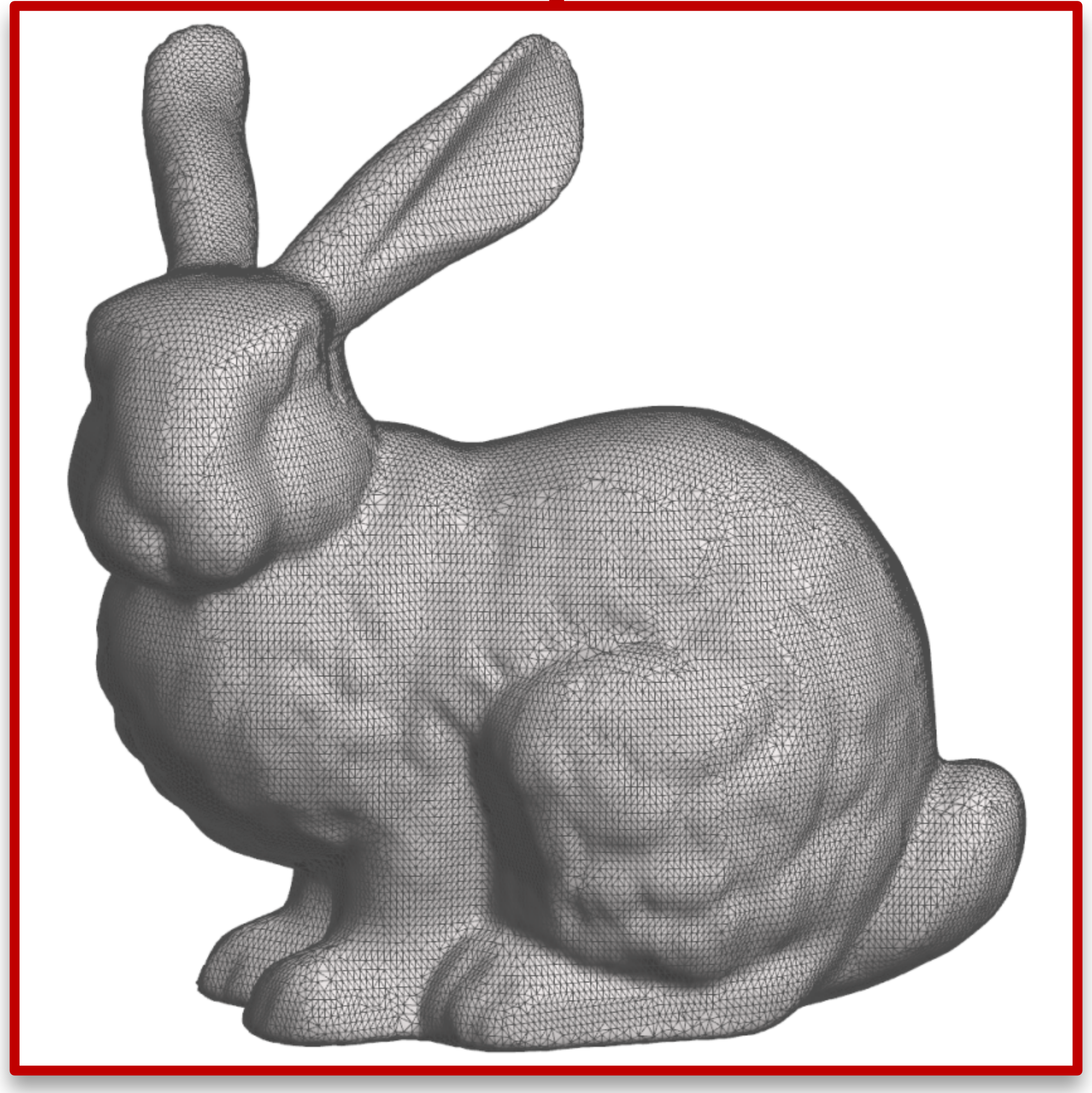
Registration:
Brings all range images to one coordinate system

⇨

Stitching/ reconstruction:
Integration of scans into a single mesh

⇨

Postprocess:
- Repairing
- Filtering
- Remeshing
- Compression



Geometry Acquisition Pipeline

Scanning: results in range images

⇨

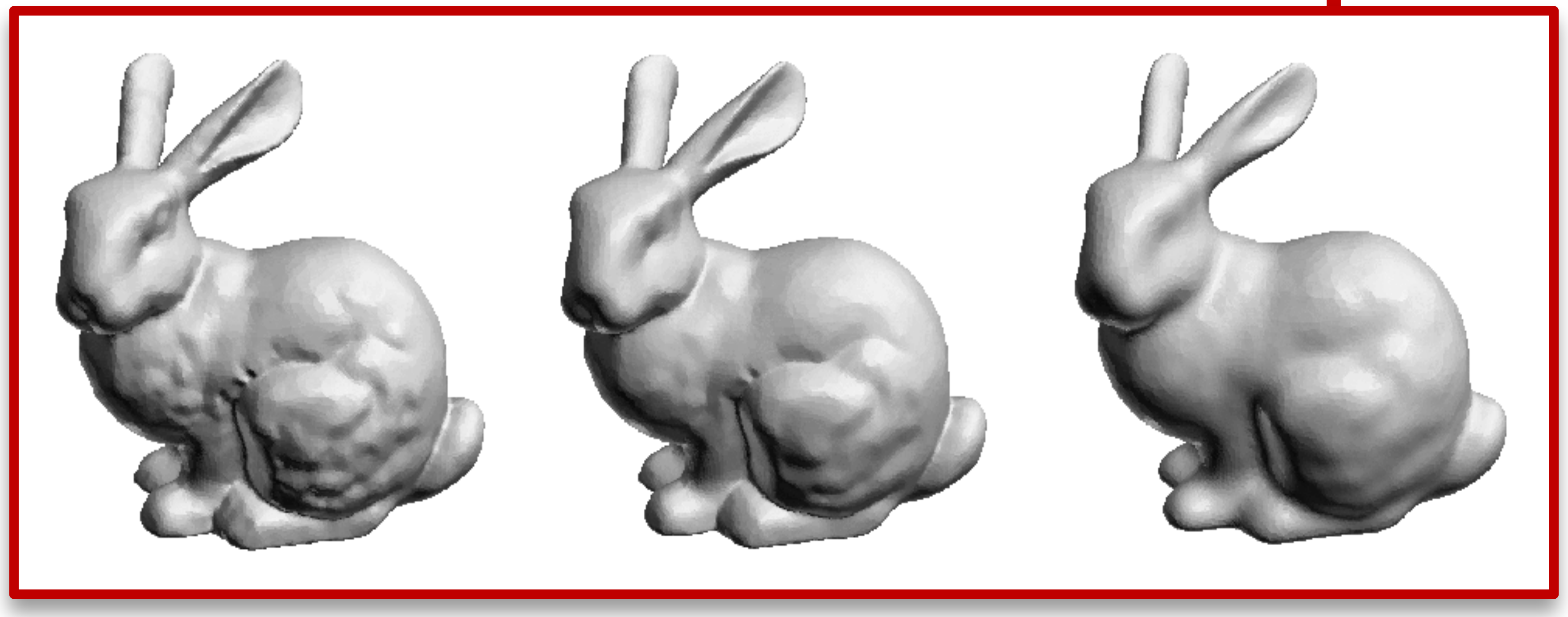
Registration: bring all range images to one coordinate system

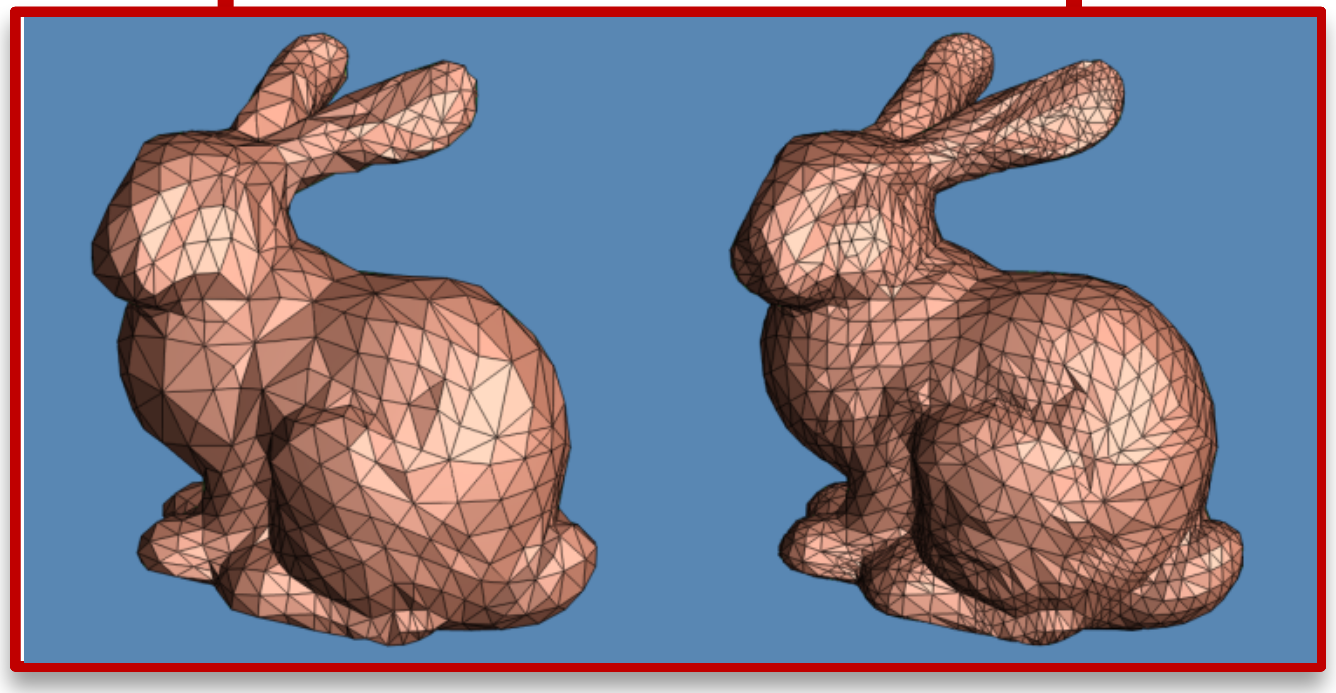
⇨

Stitching/ reconstruction: Integration of scans into a single mesh

⇨

Postprocess:
- Repairing
- Filtering
- Remeshing
- Compression





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
  - Can’t use on fragile objects
  - Holes and overhangs listed by size of probe

Optical Scanning

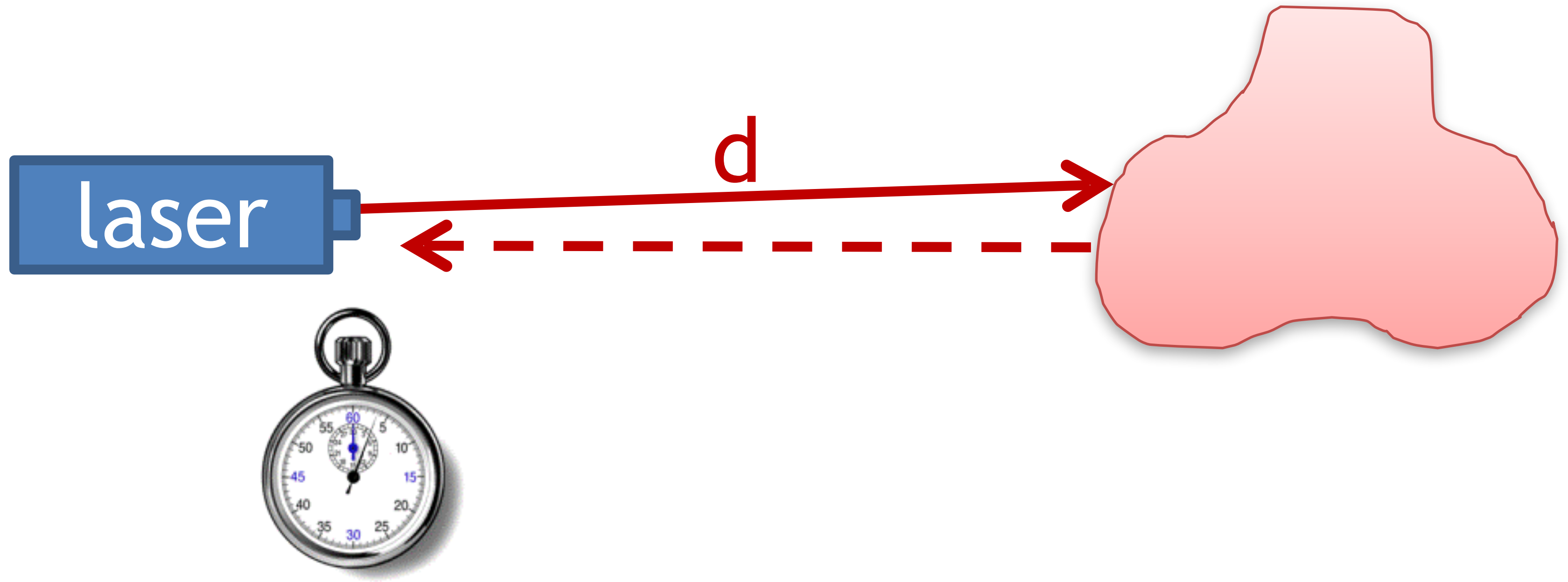
- Infer the geometry from light reflectance

- Advantages:
  - Less invasive than touch
  - Fast, large scale possible

- Disadvantages:
  - Difficulty with transparent and shiny objects

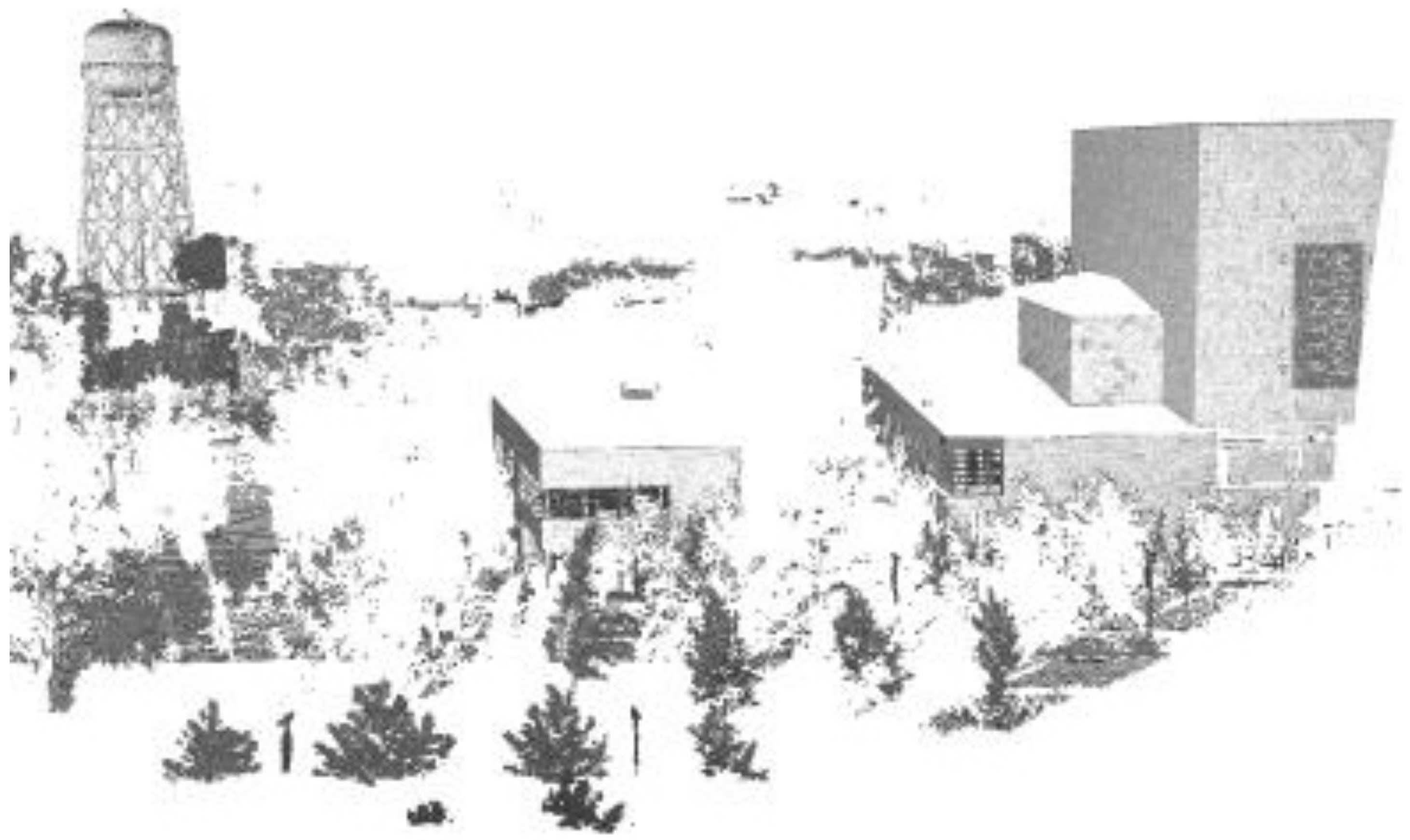
Optical scanning – active lighting
Time of flight laser

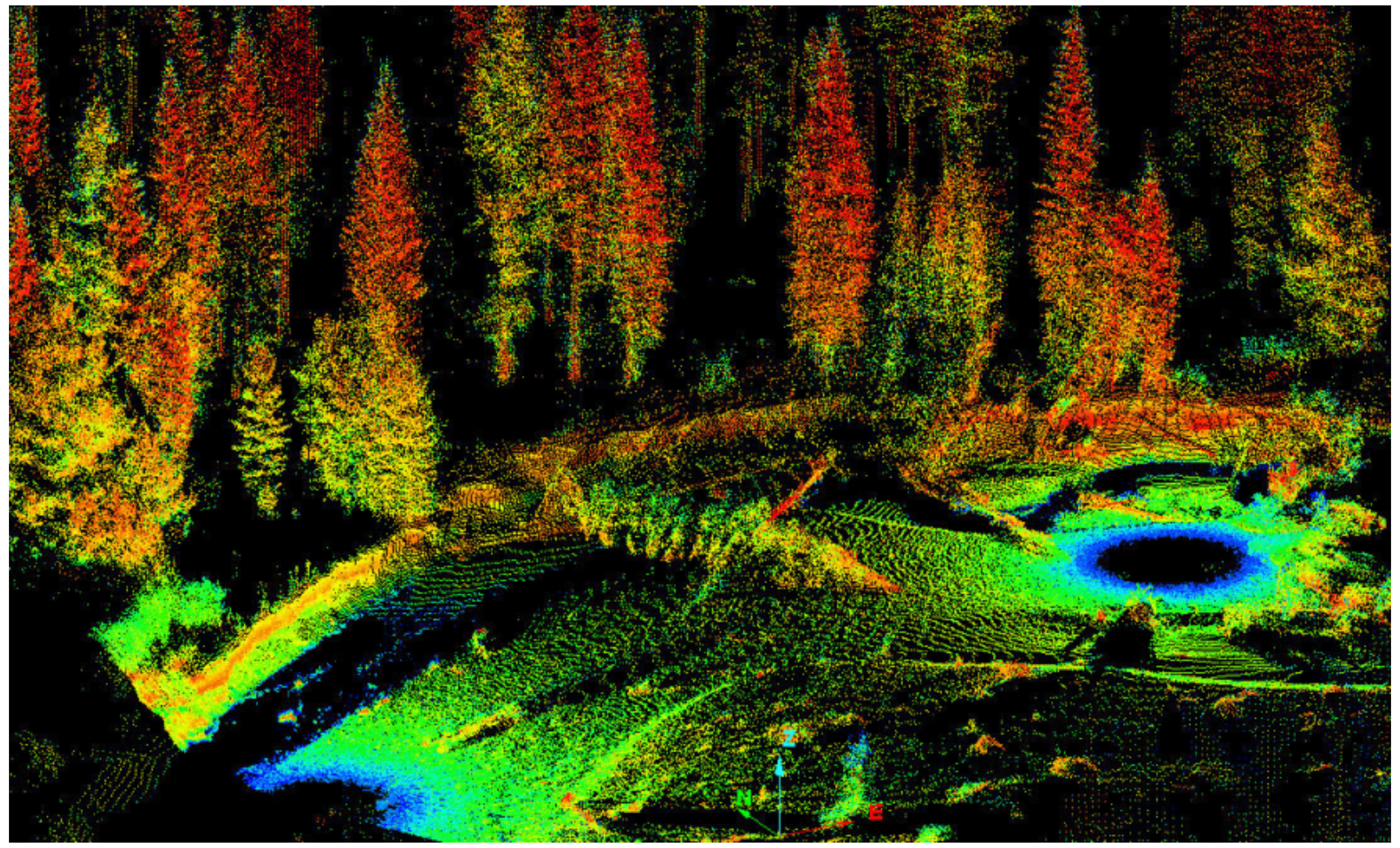
- A type of laser rangefinder (LIDAR)
- Measures the time it takes the laser beam to hit the object and come back



Optical scanning – active lighting

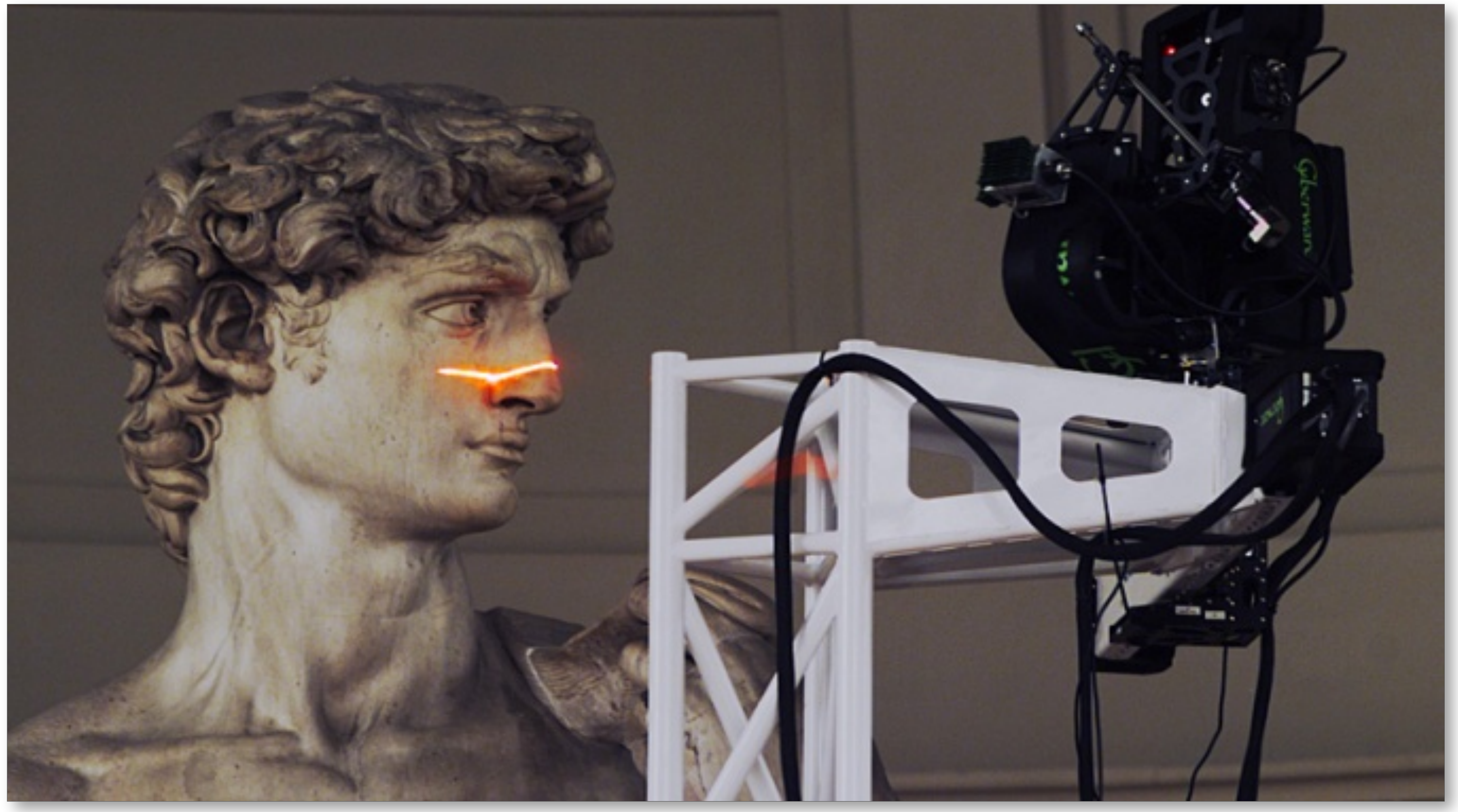
- Accommodates large range – up to several miles (suitable for buildings, rocks)
- Only for static scenes, object motion introduces noise



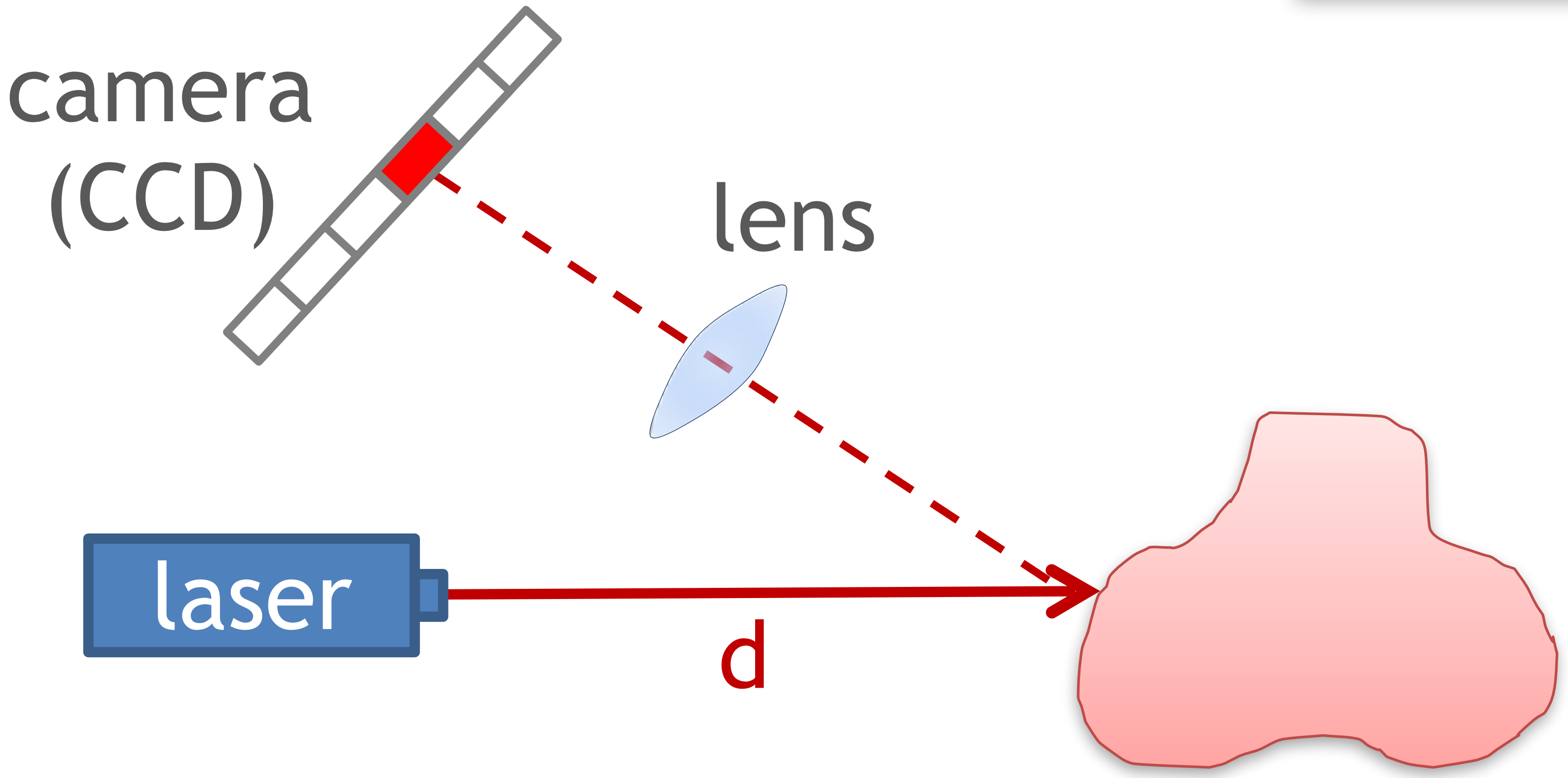


Optical scanning – active lighting
Triangulation laser

- 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

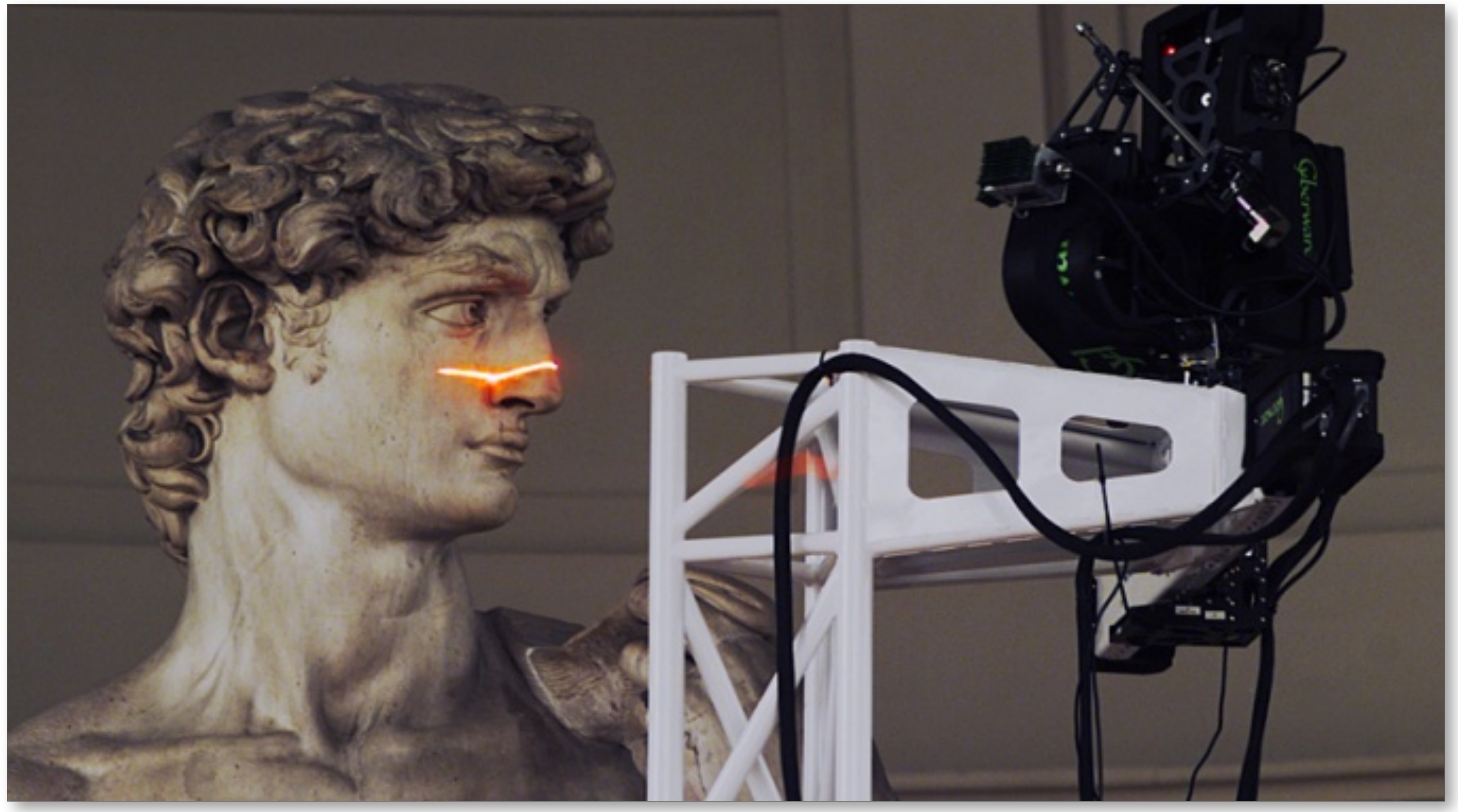


Digital Michelangelo Project

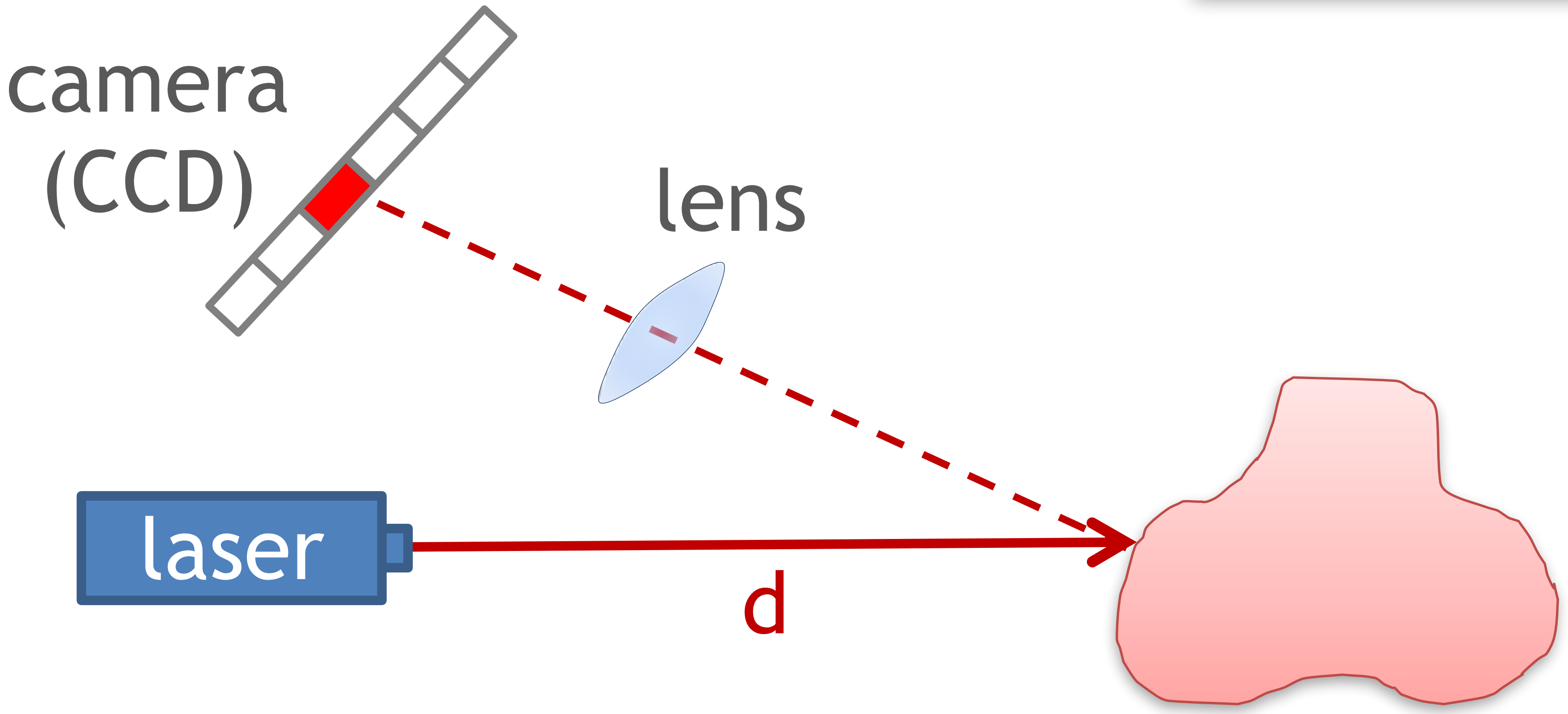


Optical scanning – active lighting
Triangulation laser

- 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

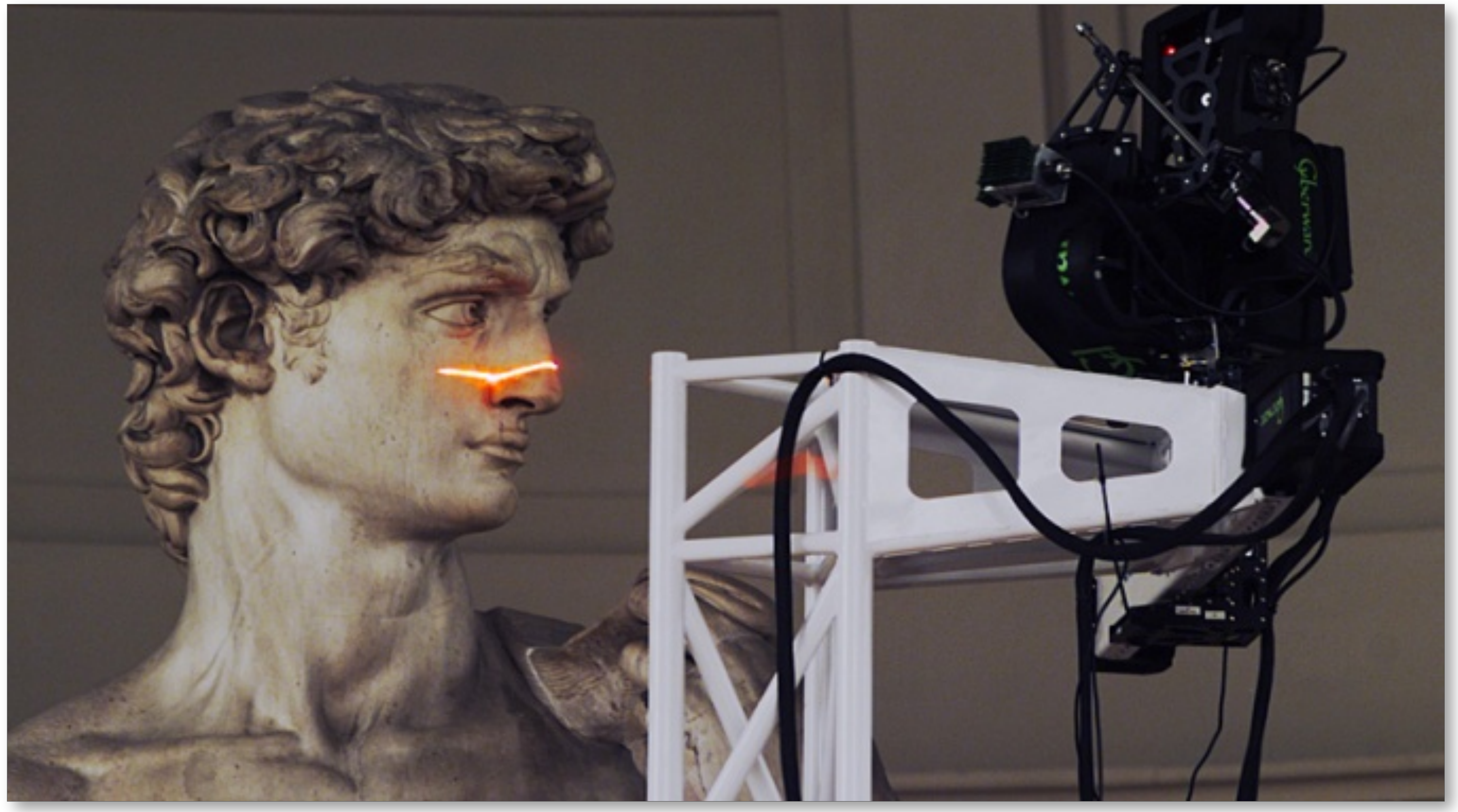


Digital Michelangelo Project

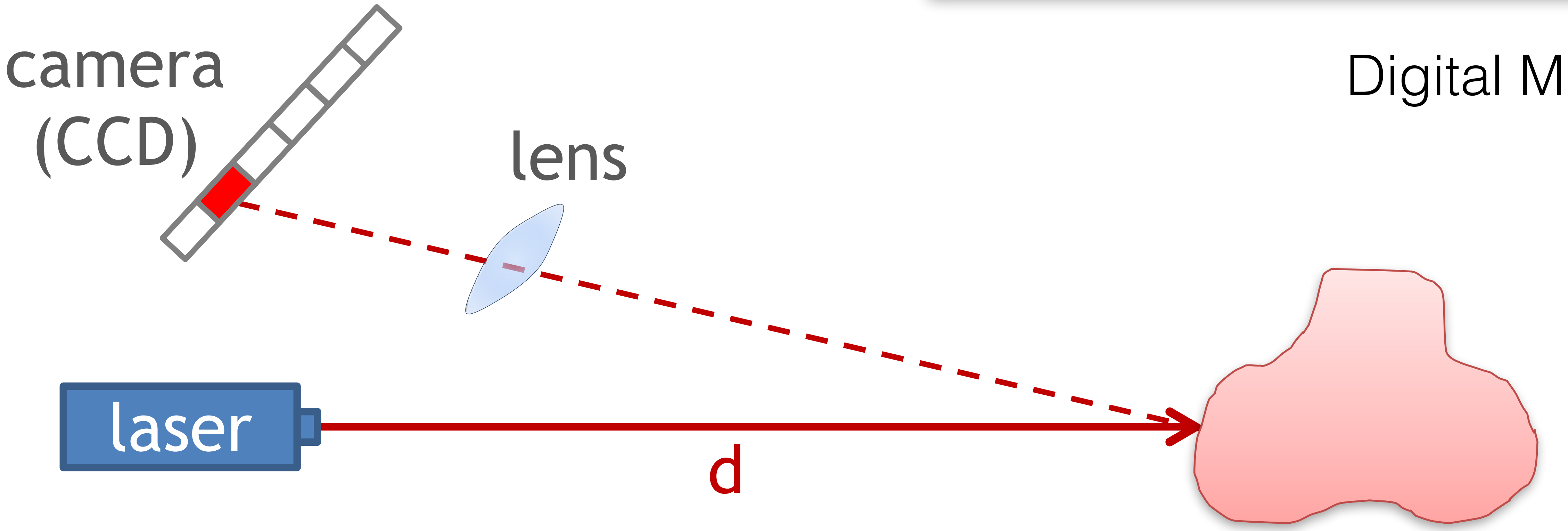


Optical scanning – active lighting
Triangulation laser

- 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

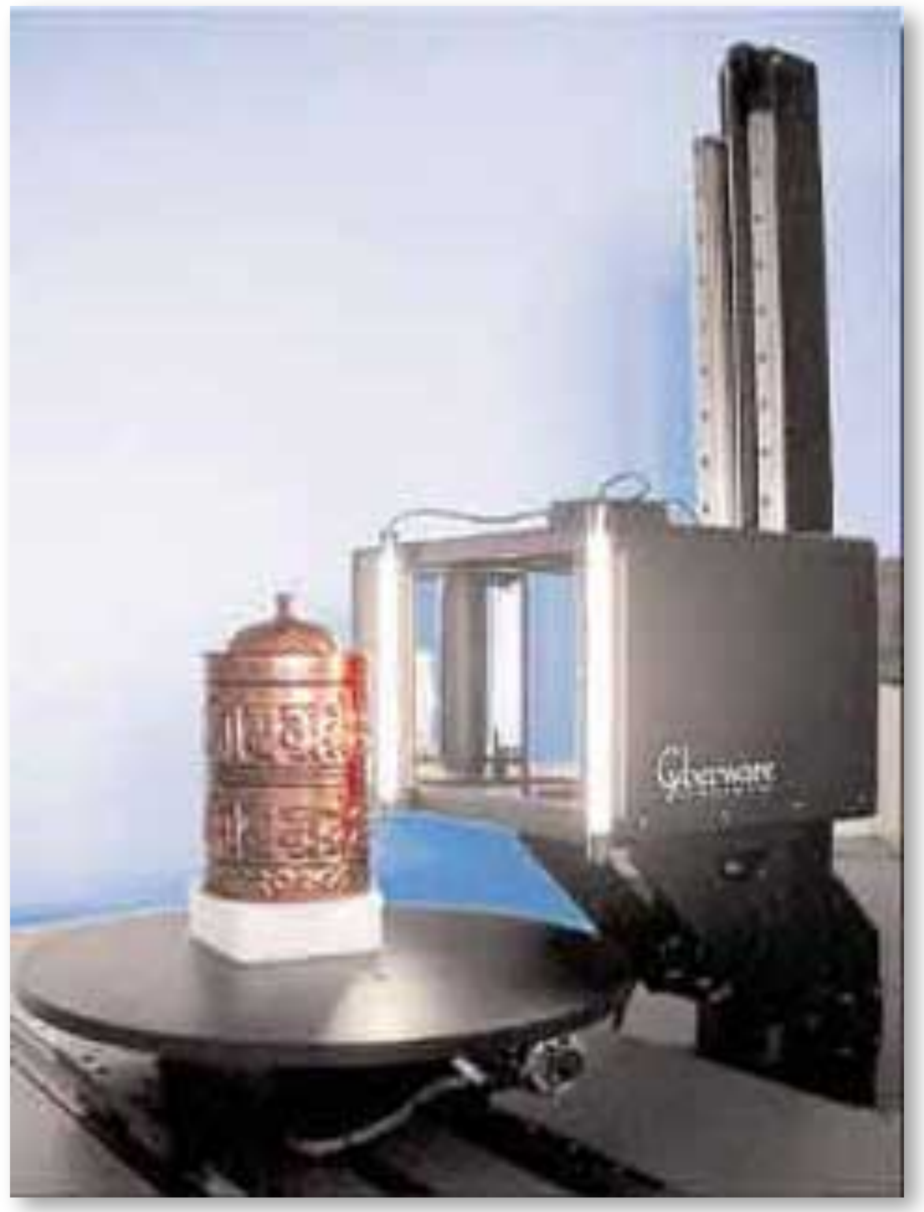


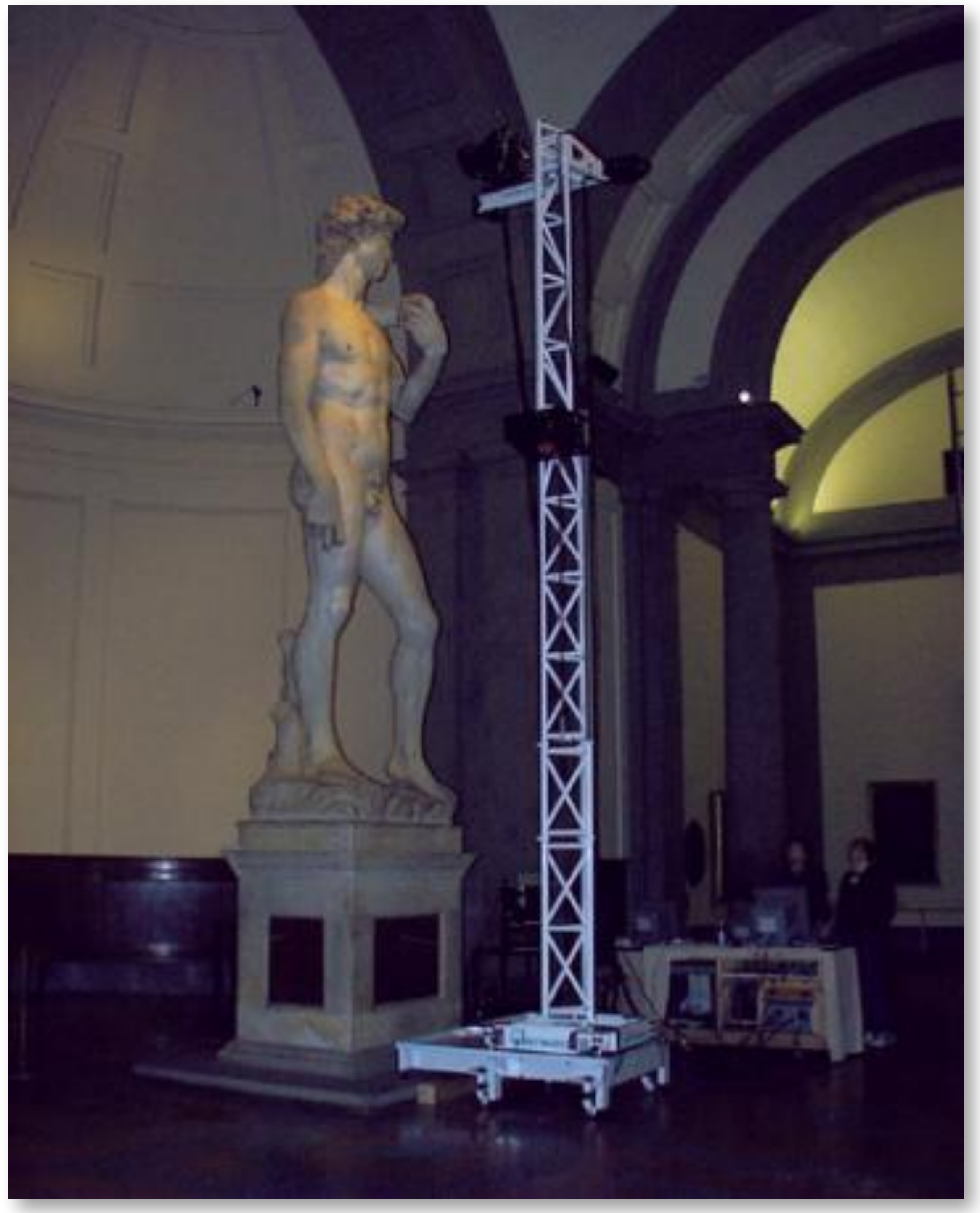
Digital Michelangelo Project



Optical scanning – active lighting
Triangulation laser

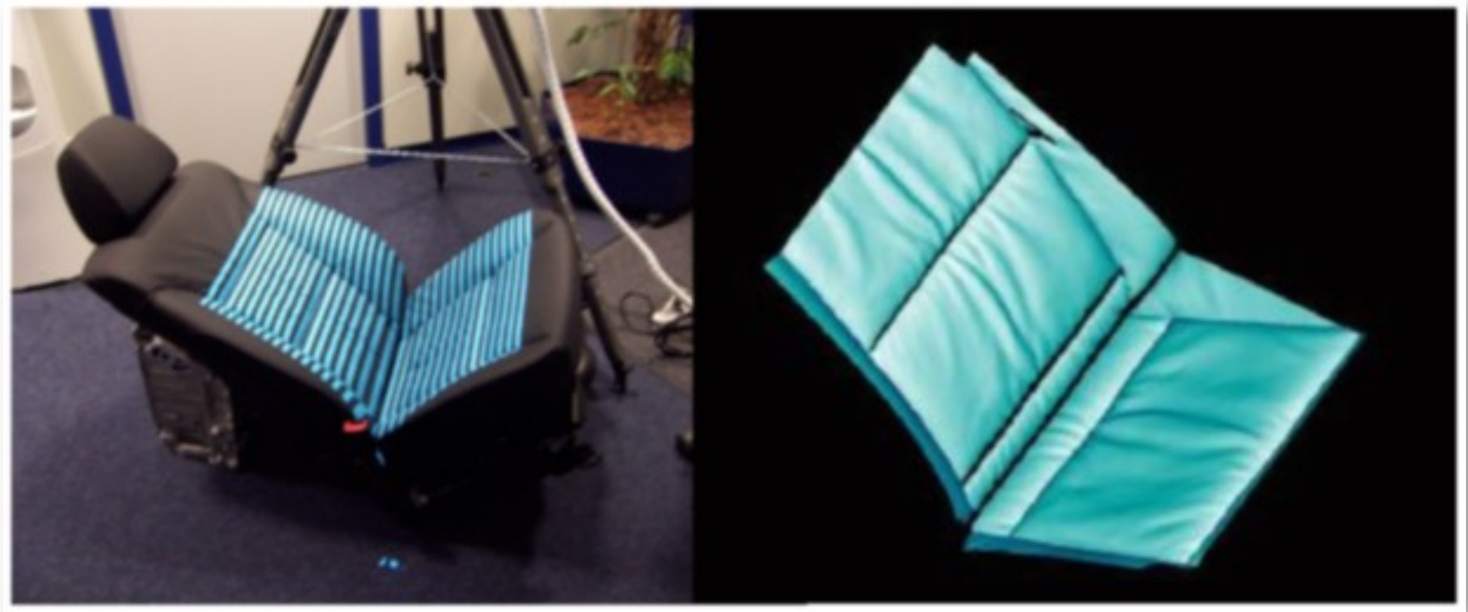
- Very precise (tens of microns)
- Small distance (meters)
- Small size of target (meters)





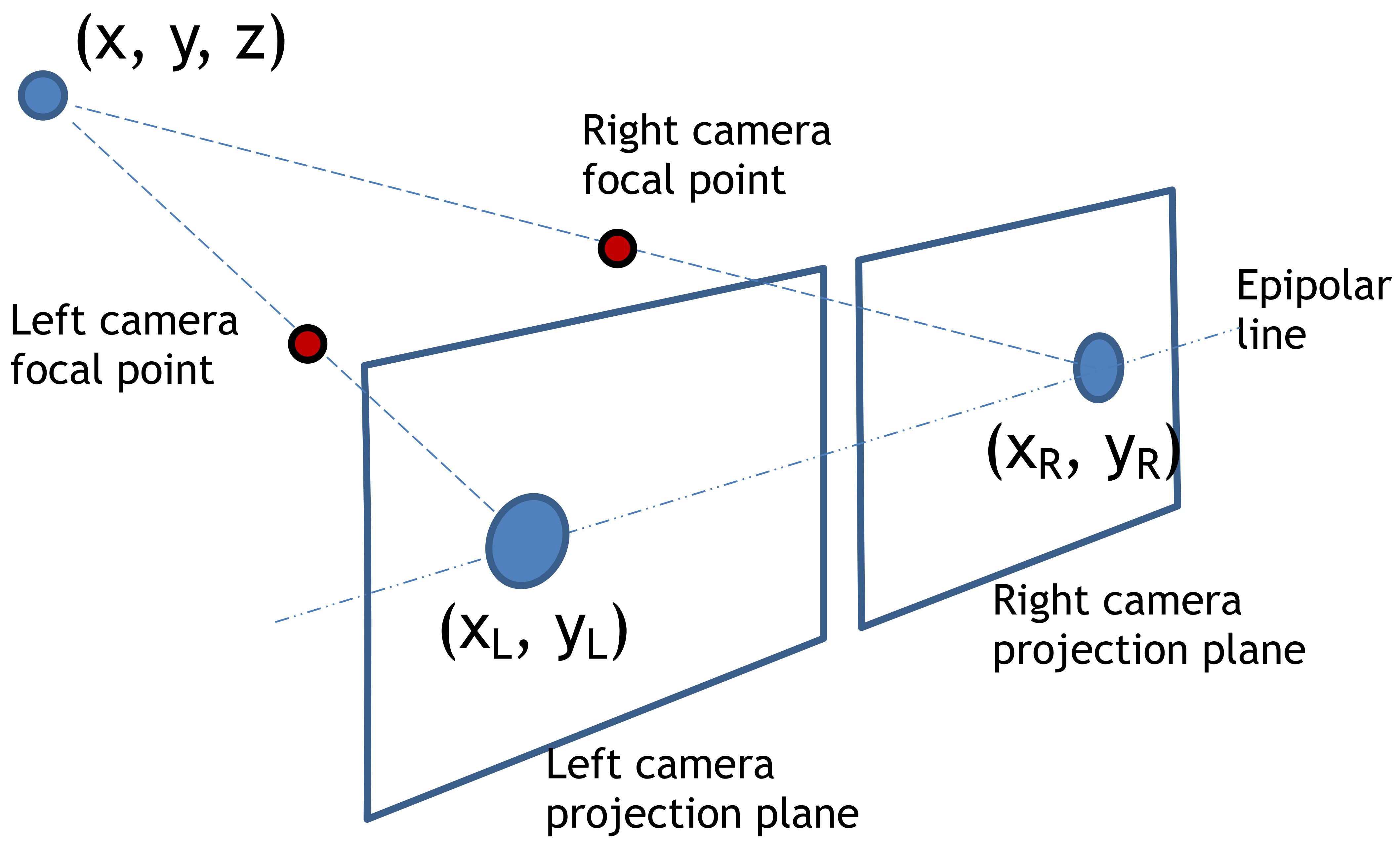
Optical scanning – active lighting
Structured light

- 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



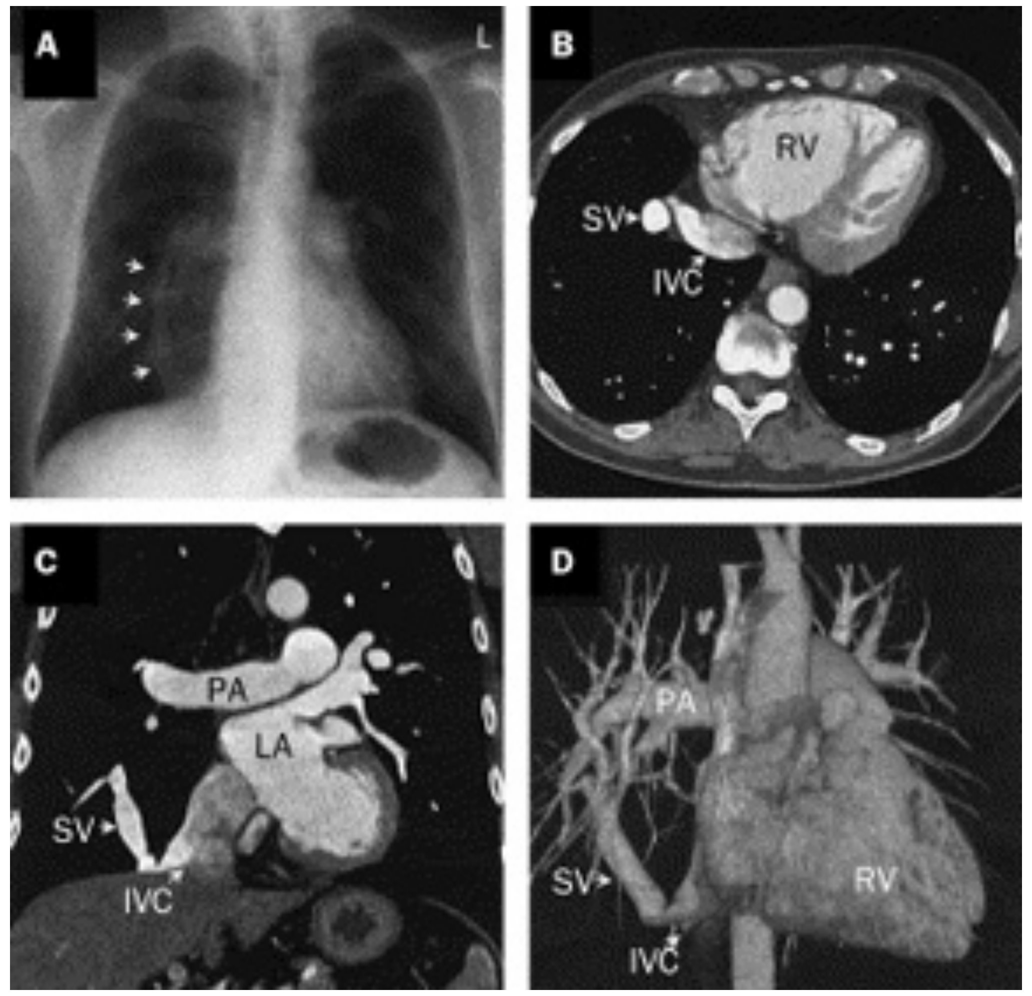
Optical scanning – passive stereo

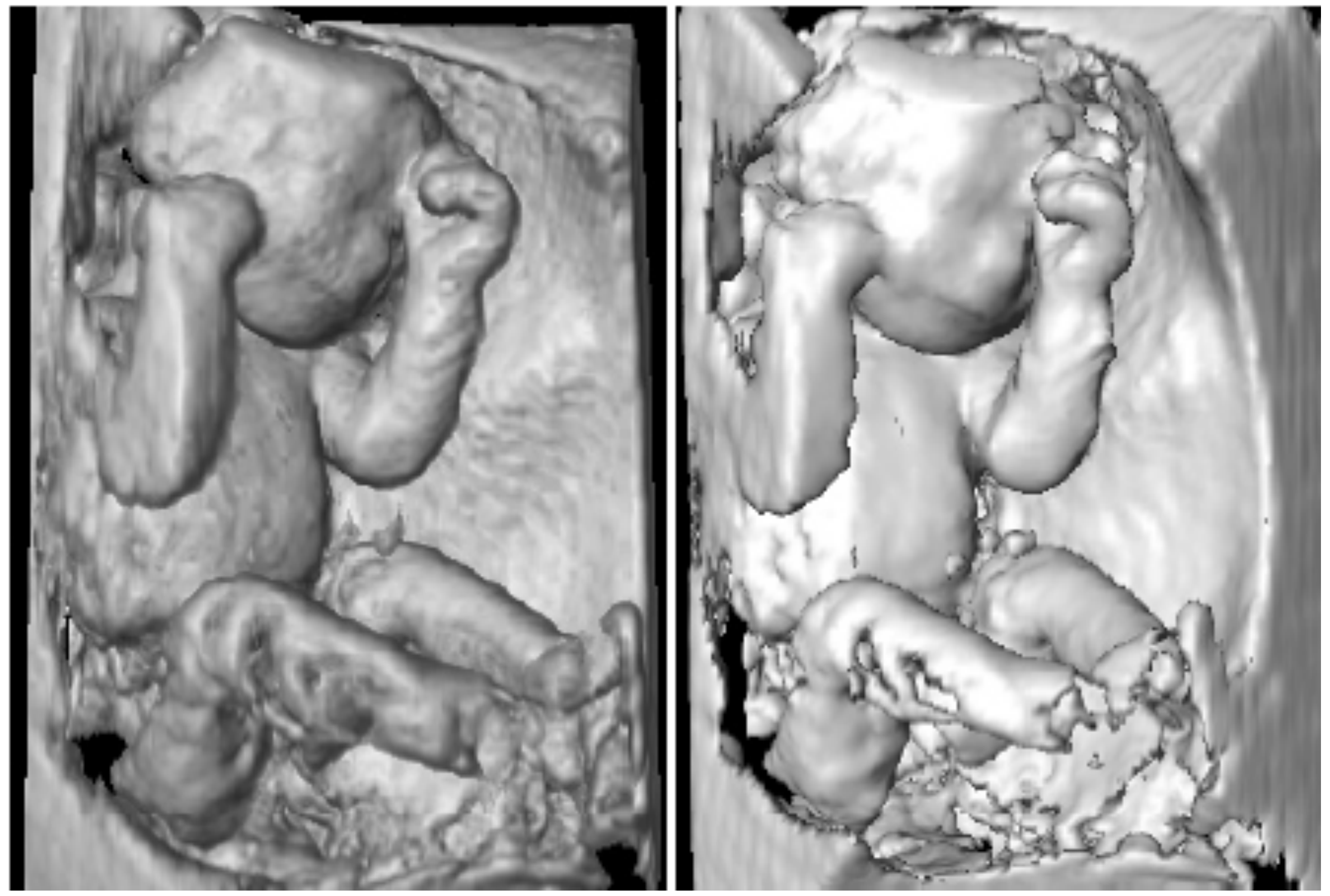
- No need for special lighting/radiation
- Two (or more) cameras
- Feature matching and triangulation

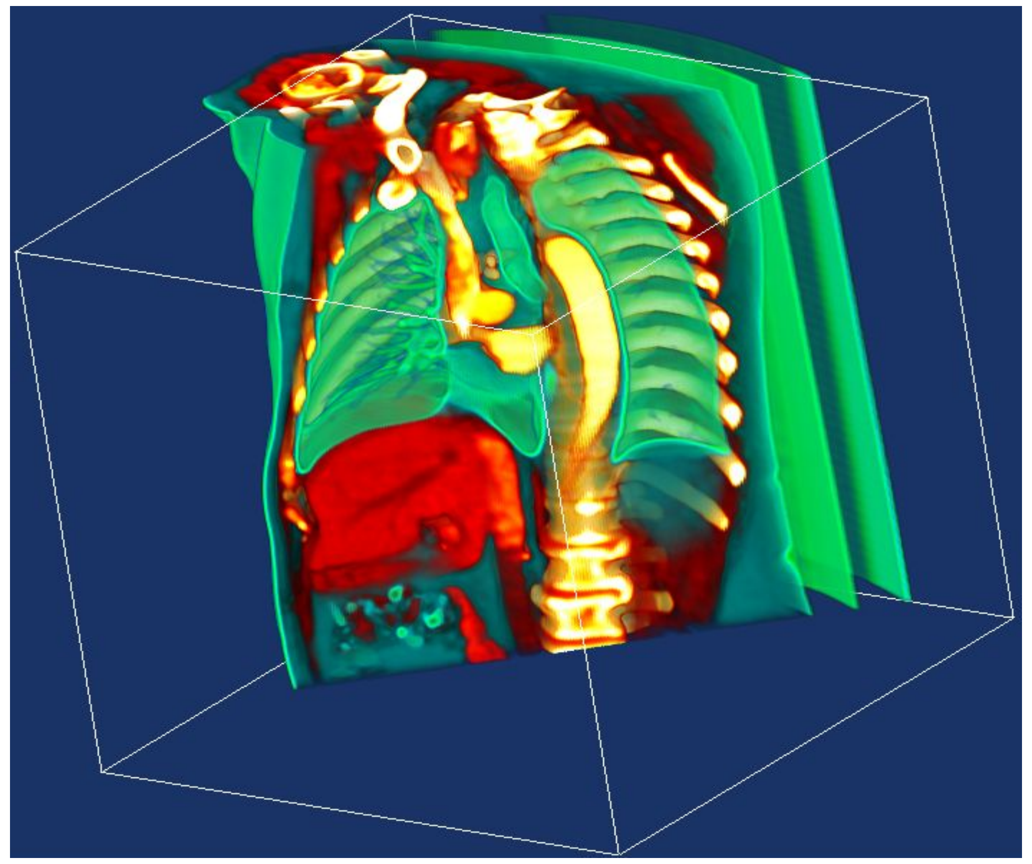


Imaging

- Ultrasound, CT, MRI
- Discrete volume of density data
- First need to segment the desired object (contouring)

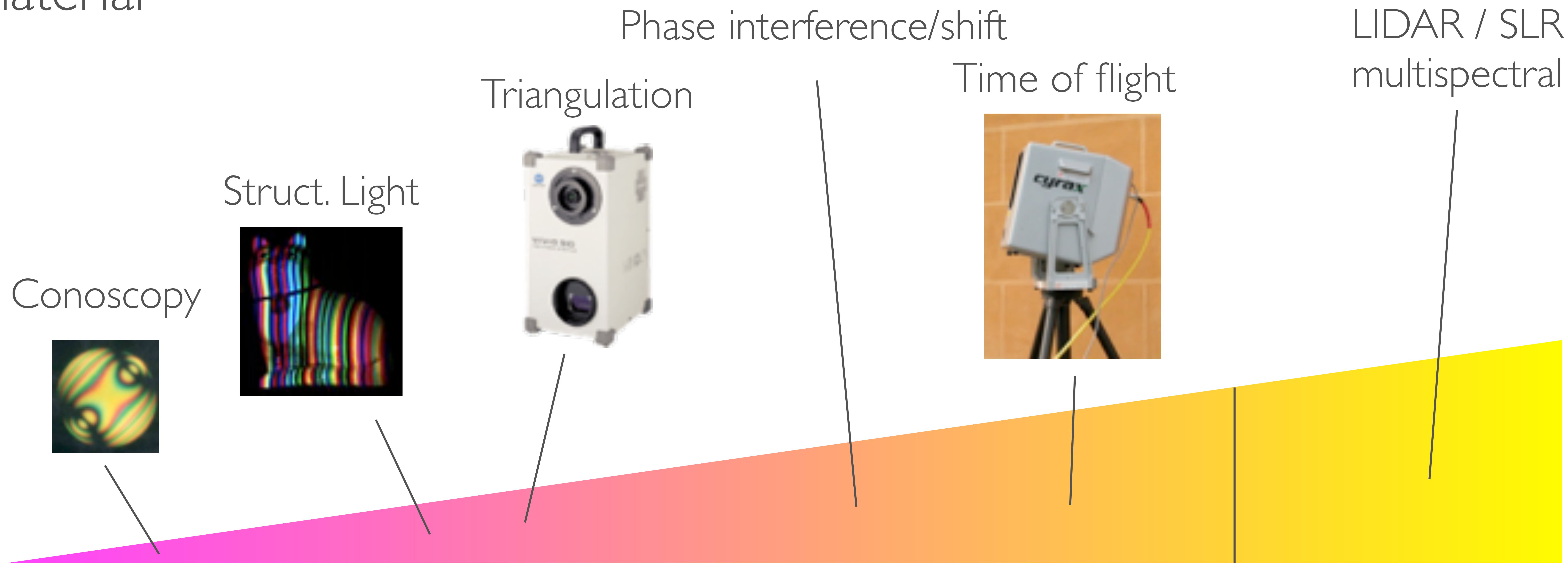






3D scanning devices

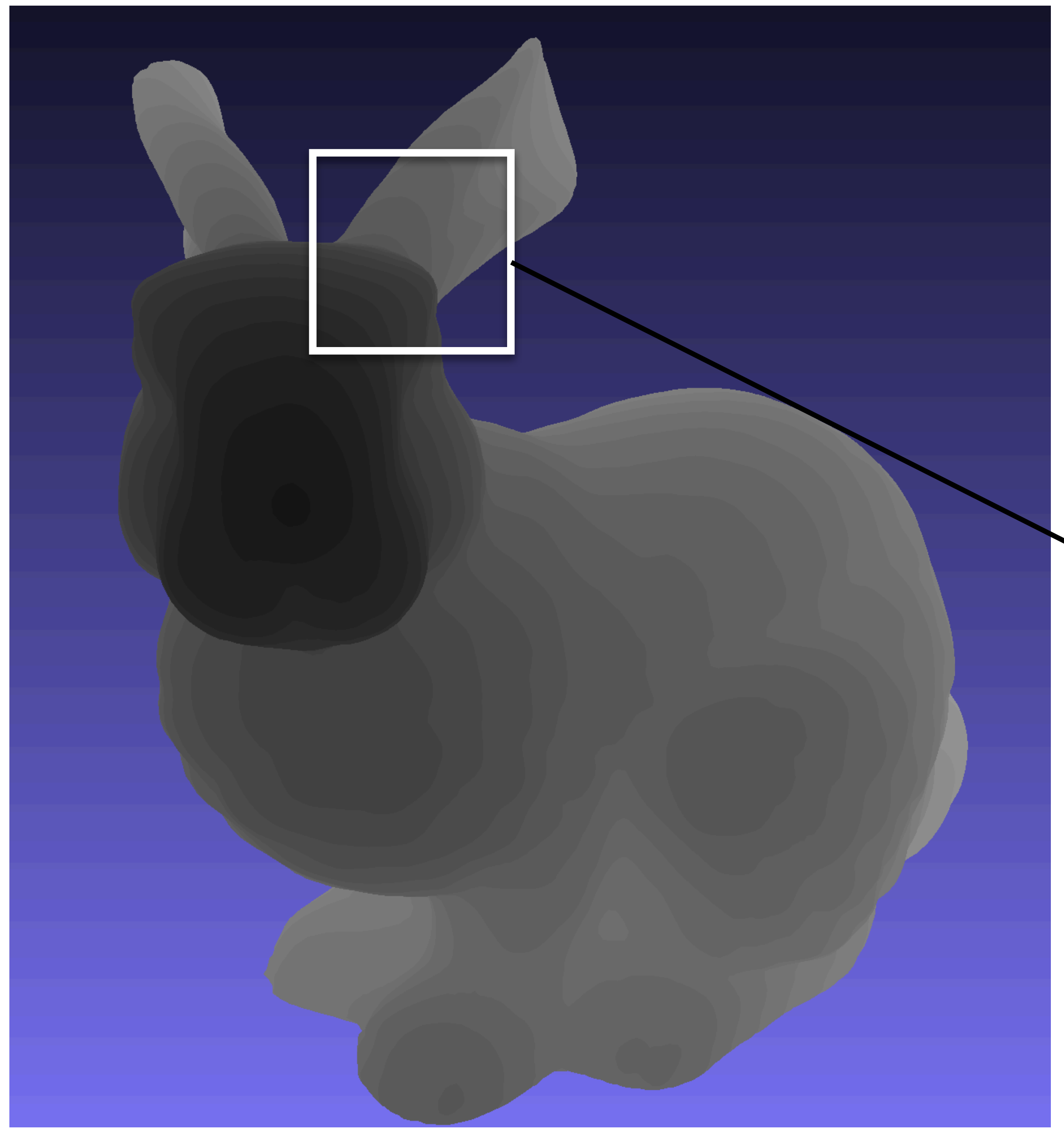
- Sensor is no longer the main problem...
- Gamut of measurable object is increasing, in terms of both size and material



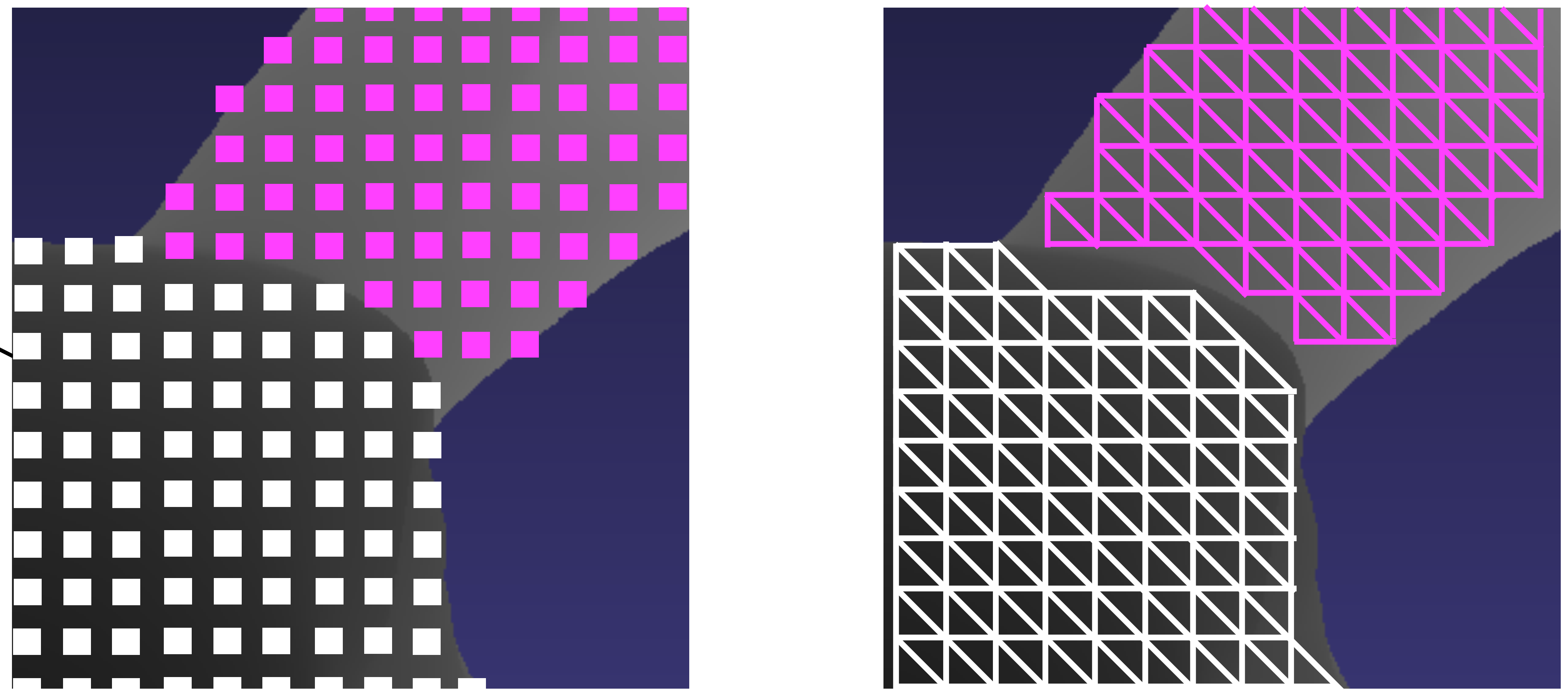
Range images

- A single shot from most scanning equipment provides a depth value for each pixel in its sensor

- Depths are converted to a point grid that is triangulated based on its intrinsic regularity



The result of a single scan is called a RANGE MAP



Registration

Acknowledgements: Niloy Mitra, Marco Tarini, Olga Sorkine Hornung
http://resources.mpi-inf.mpg.de/deformableShapeMatching/EG2012_Tutorial/

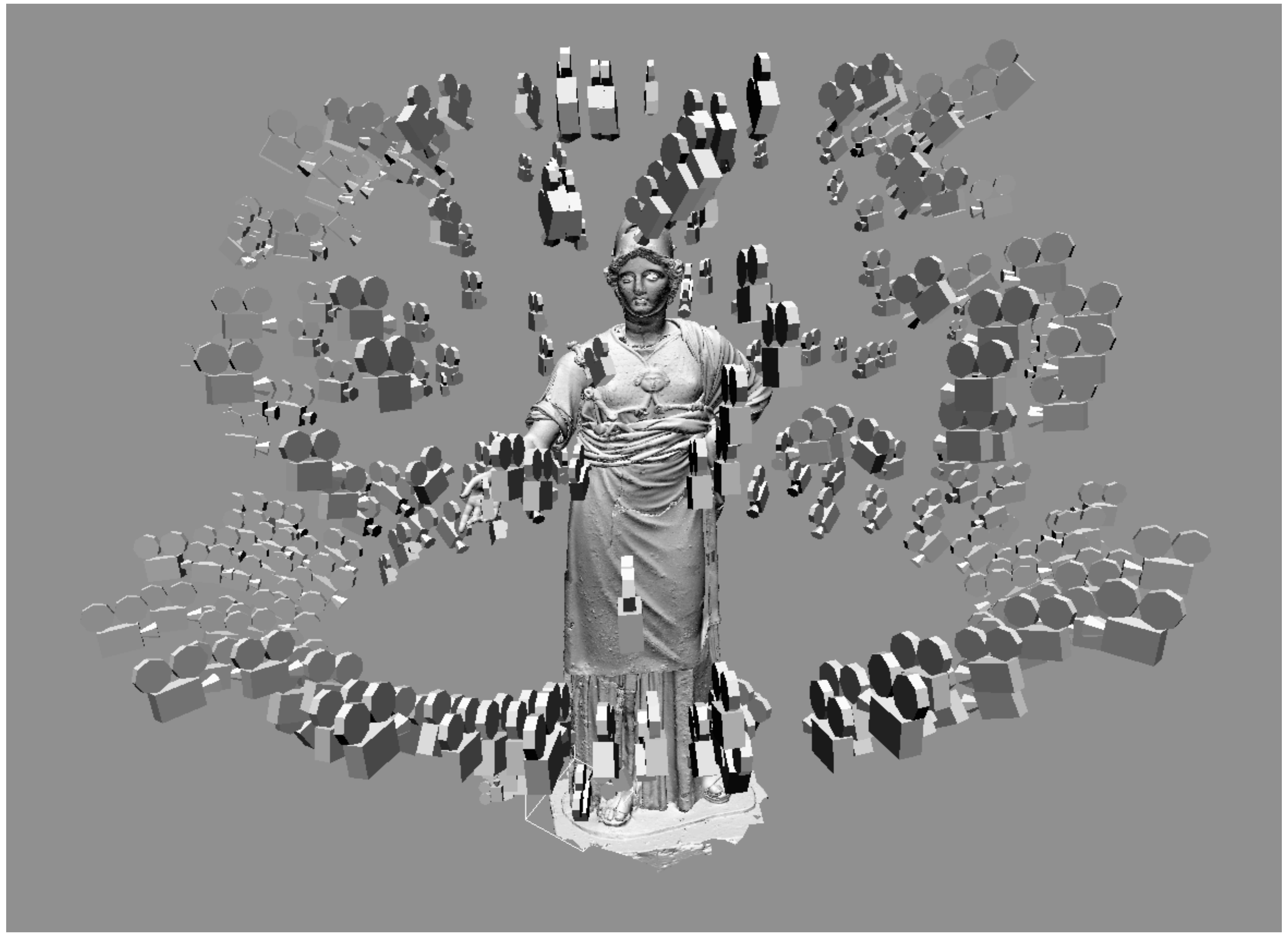
Problem Statement

Many range maps for a single object:

- Each one in its own 3D reference frame (camera frame)
- Need to put them in a common frame (world frame)

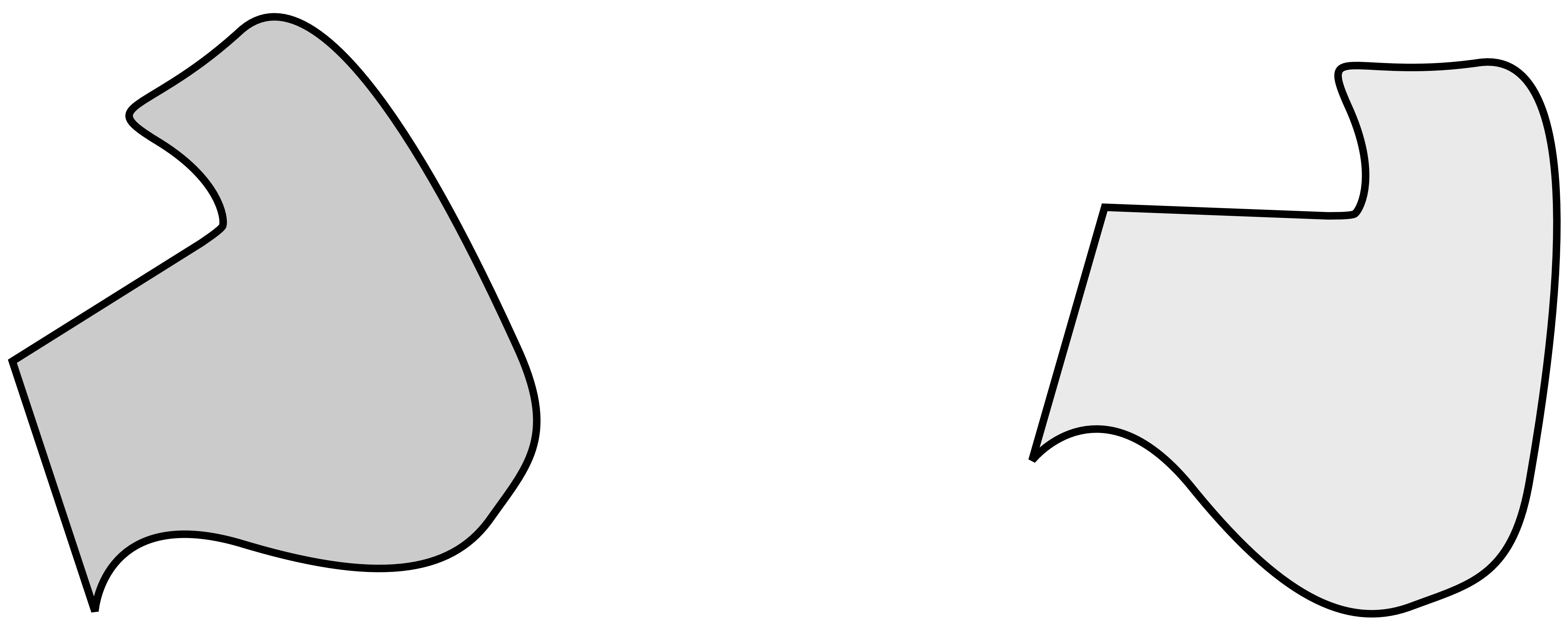
Example: Scanning the Minerva

- Minolta laser scanner (03/2002)
- No. range scans: 297
- Sampling resolution: ~0.3 mm



Problem Statement

M_1 M_2



$$M_1 \approx T(M_2)$$

T : Translation + Rotation

Problem Statement

Multiple range maps:

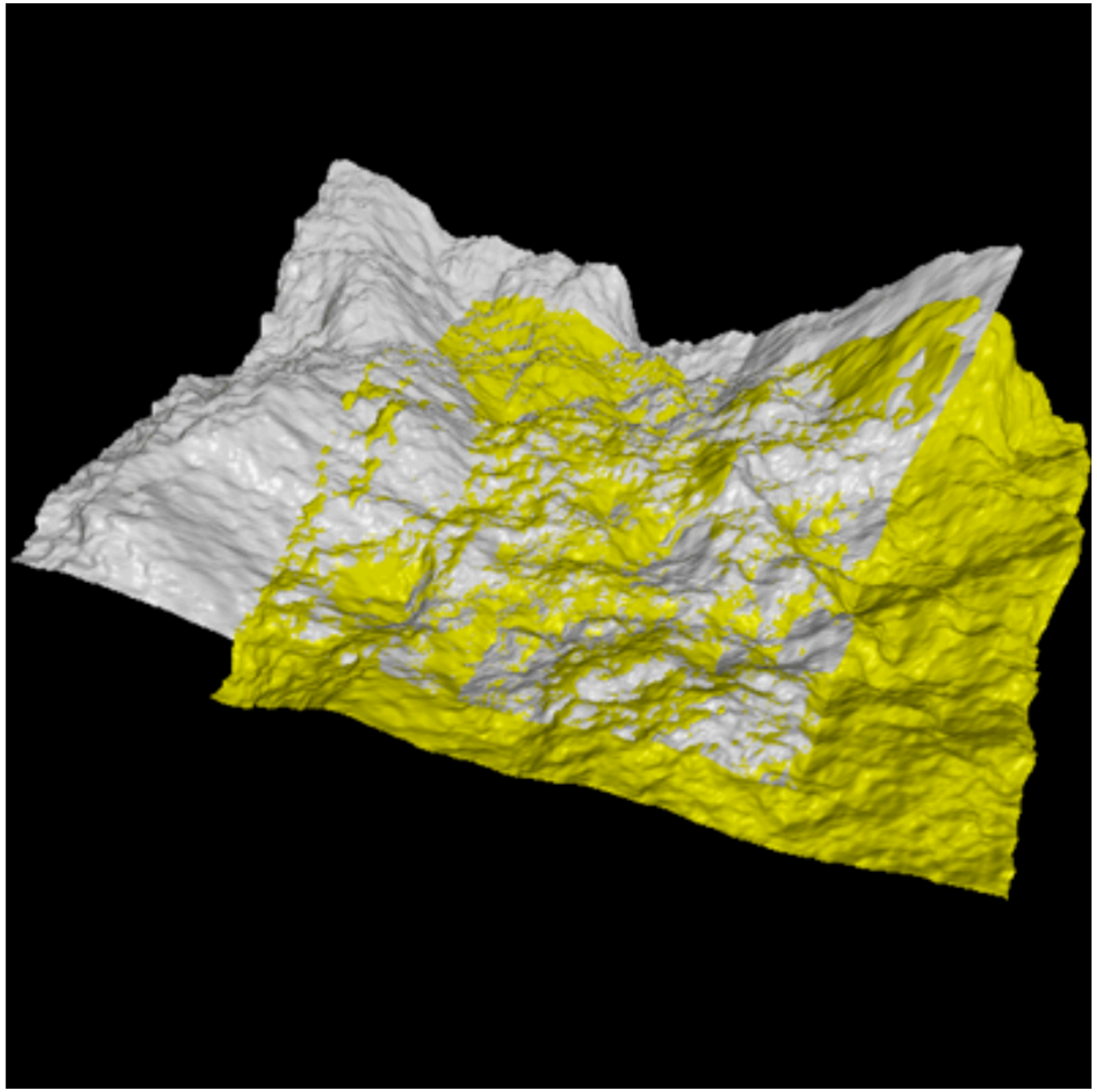
- Range maps $r_0, r_1, \ldots, r_k$
- Let $\mathbf{U}_i = \{u_{i1}, u_{i2}, u_{i3}, o_i\}$ be the reference frame of r_i
- Bring all maps $r_1, \ldots, r_k$ to the reference frame $\mathbf{U}_0 = \{u_{01}, u_{02}, u_{03}, o_0\}$
- Equivalently:
  - Find the rigid transformation taking r_i to position consistent with r_0

Correspondences

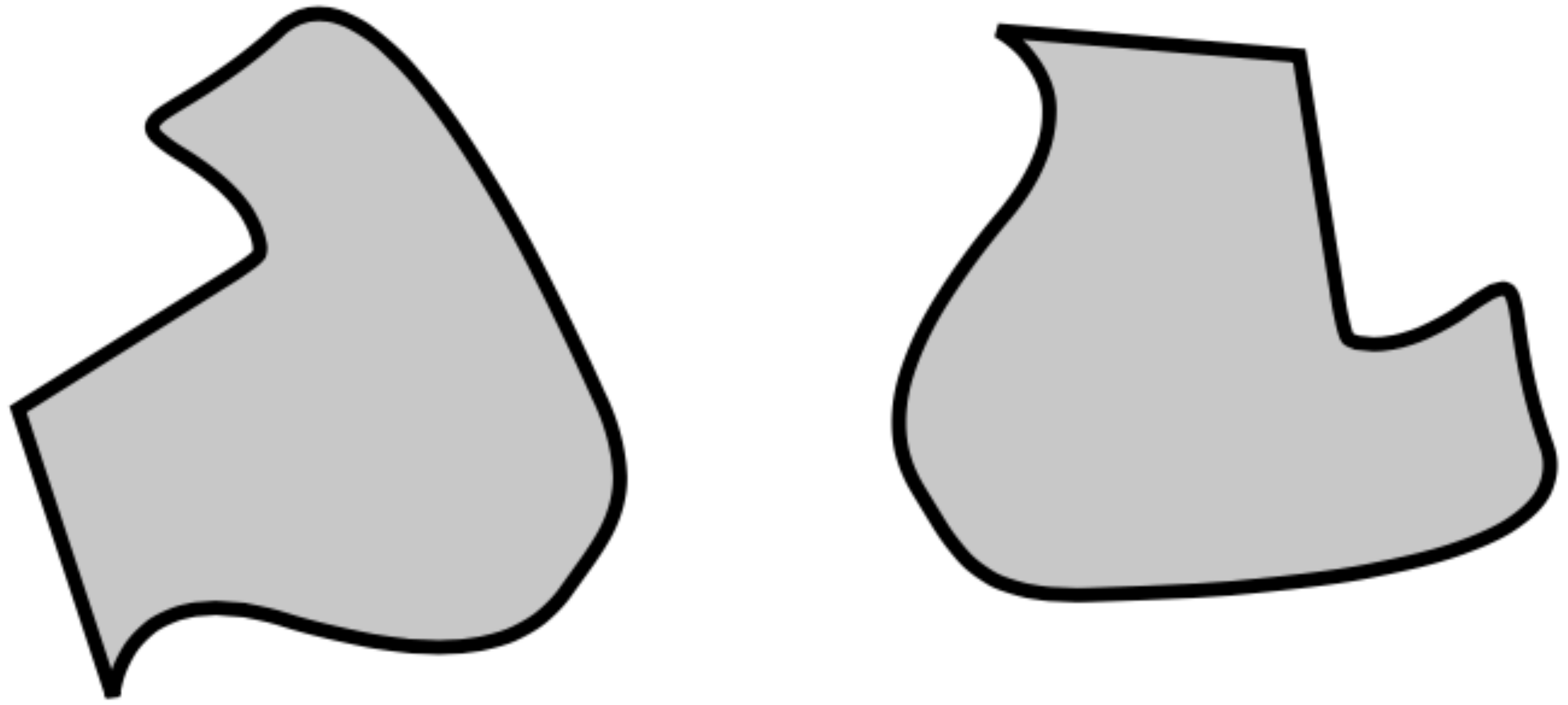
- Corresponding pairs: same object points in different range maps

$$\mathbf{p}_1 \rightarrow \mathbf{q}_1$$
$$\mathbf{p}_2 \rightarrow \mathbf{q}_2$$
$$\mathbf{p}_3 \rightarrow \mathbf{q}_3$$

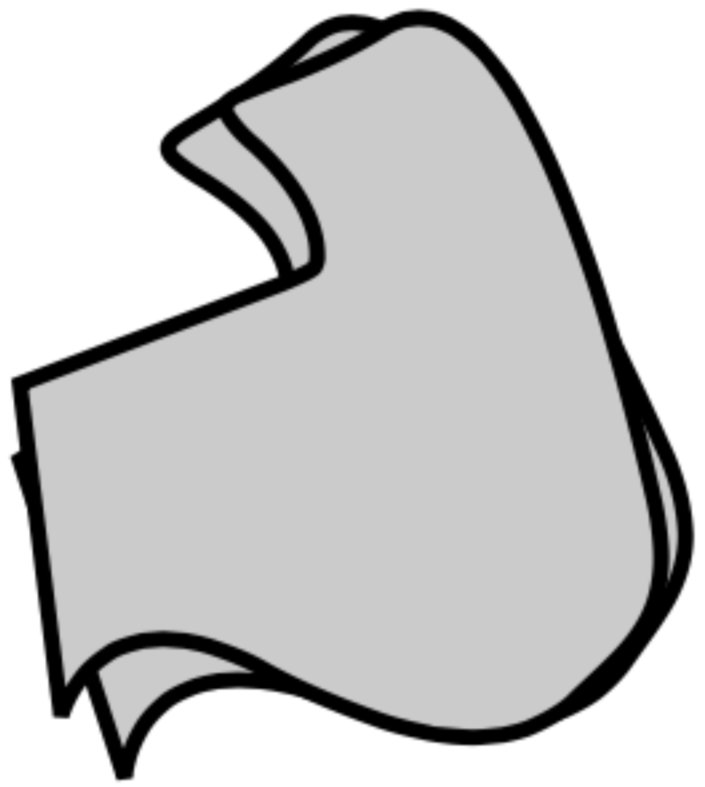
$$R\mathbf{p}_i + t \approx \mathbf{q}_i$$



Coarse vs Fine



Coarse Registration
Corresponding points are far from each other



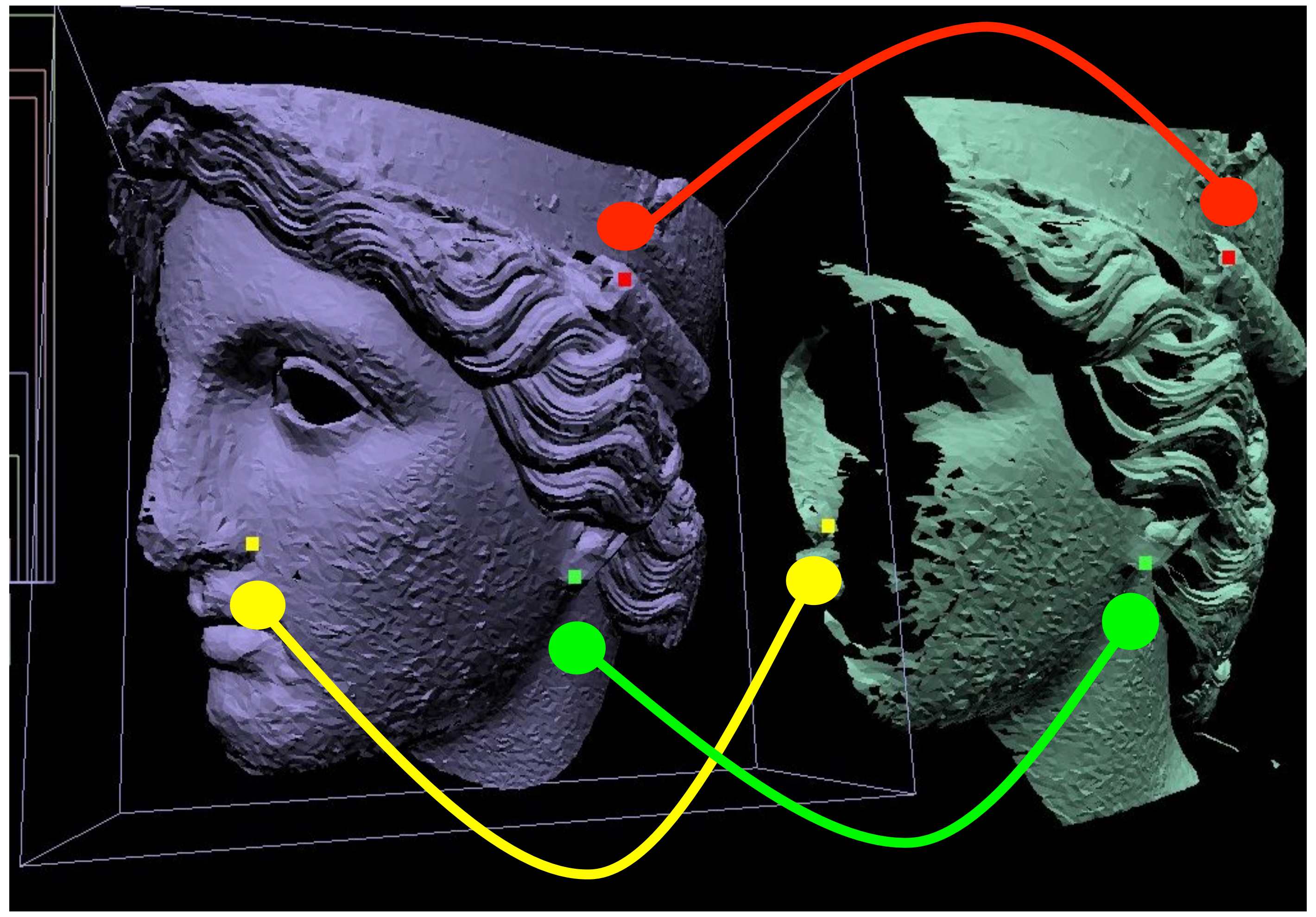
Fine Registration
Corresponding points are close

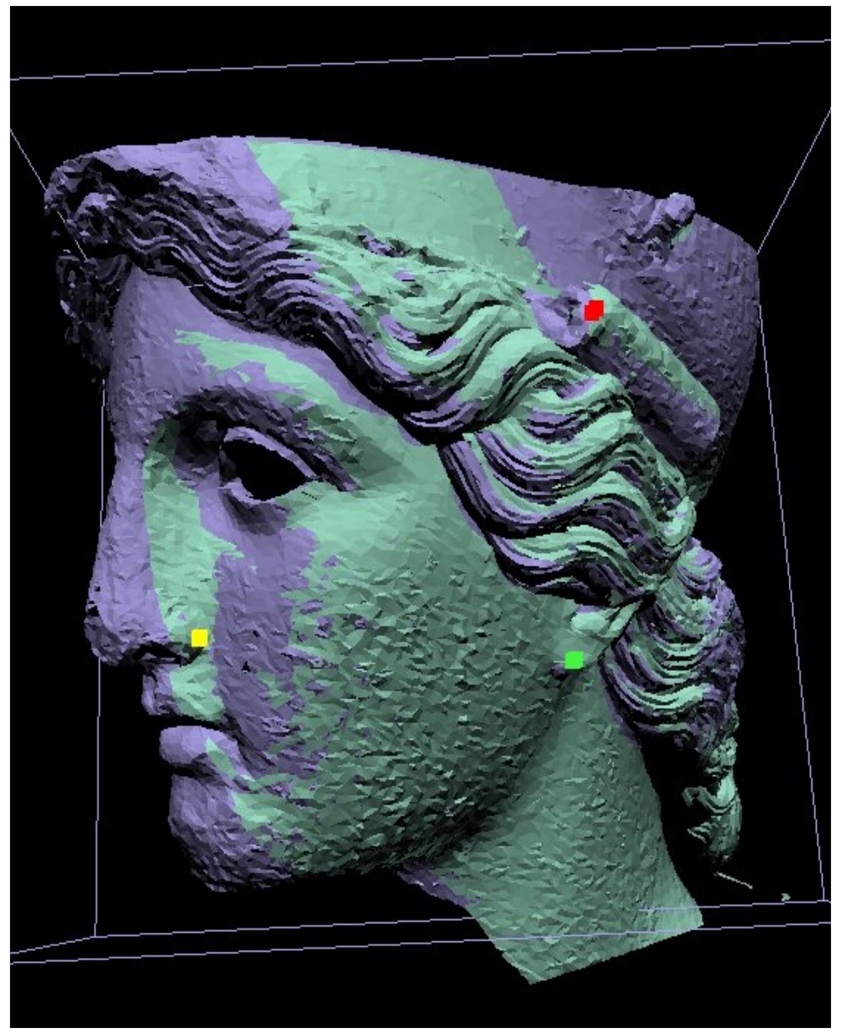
How can we find correspondences?

How many corresponding pairs do we need?

Coarse registration

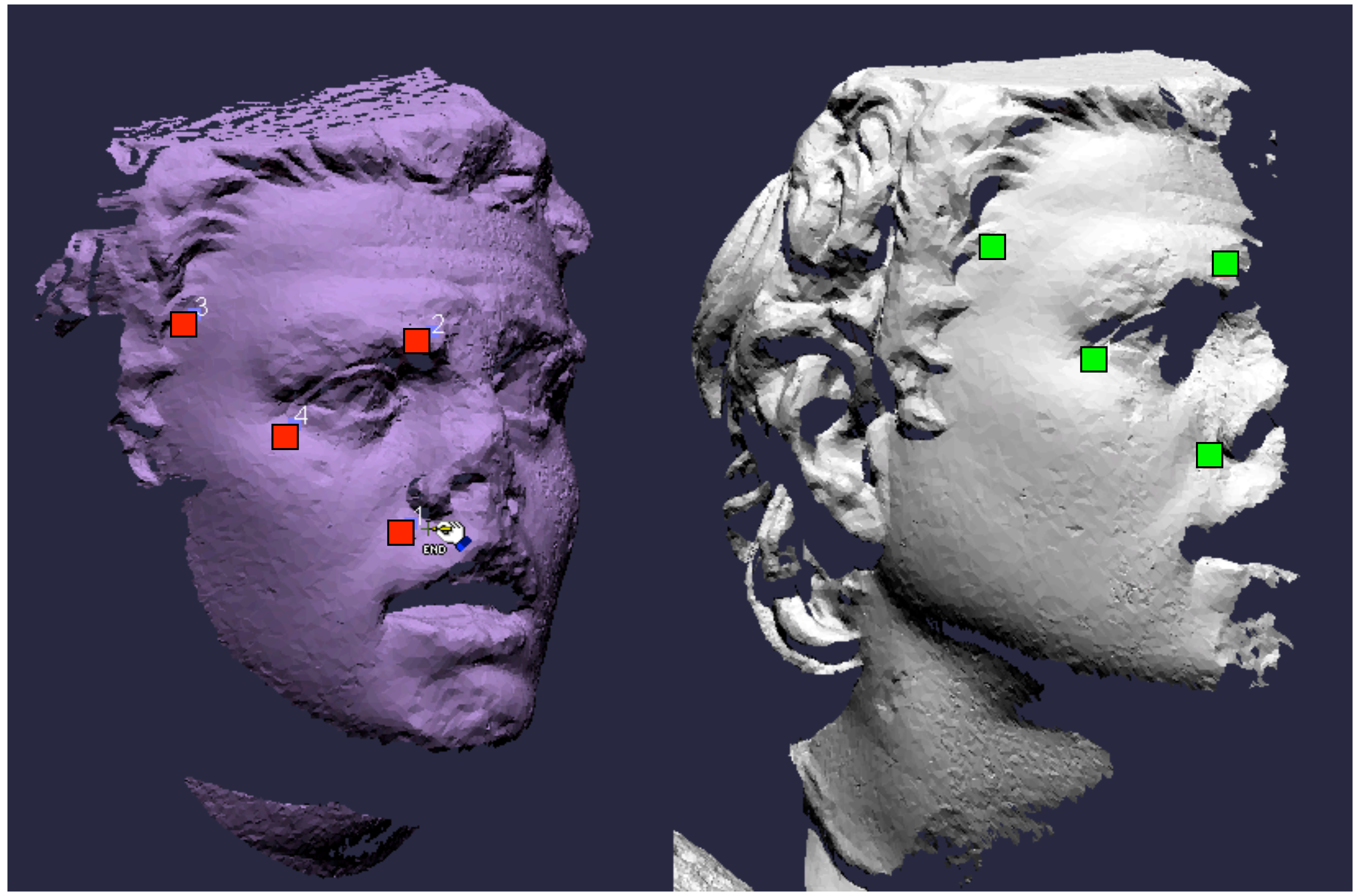
- Range maps must span overlapping portions of the object
- Pick shared reference point features
  - By hand
  - Through image processing techniques
- Models are roughly positioned according to point pairs selected
- Not perfect, but enough to warm start fine registration





Coarse registration

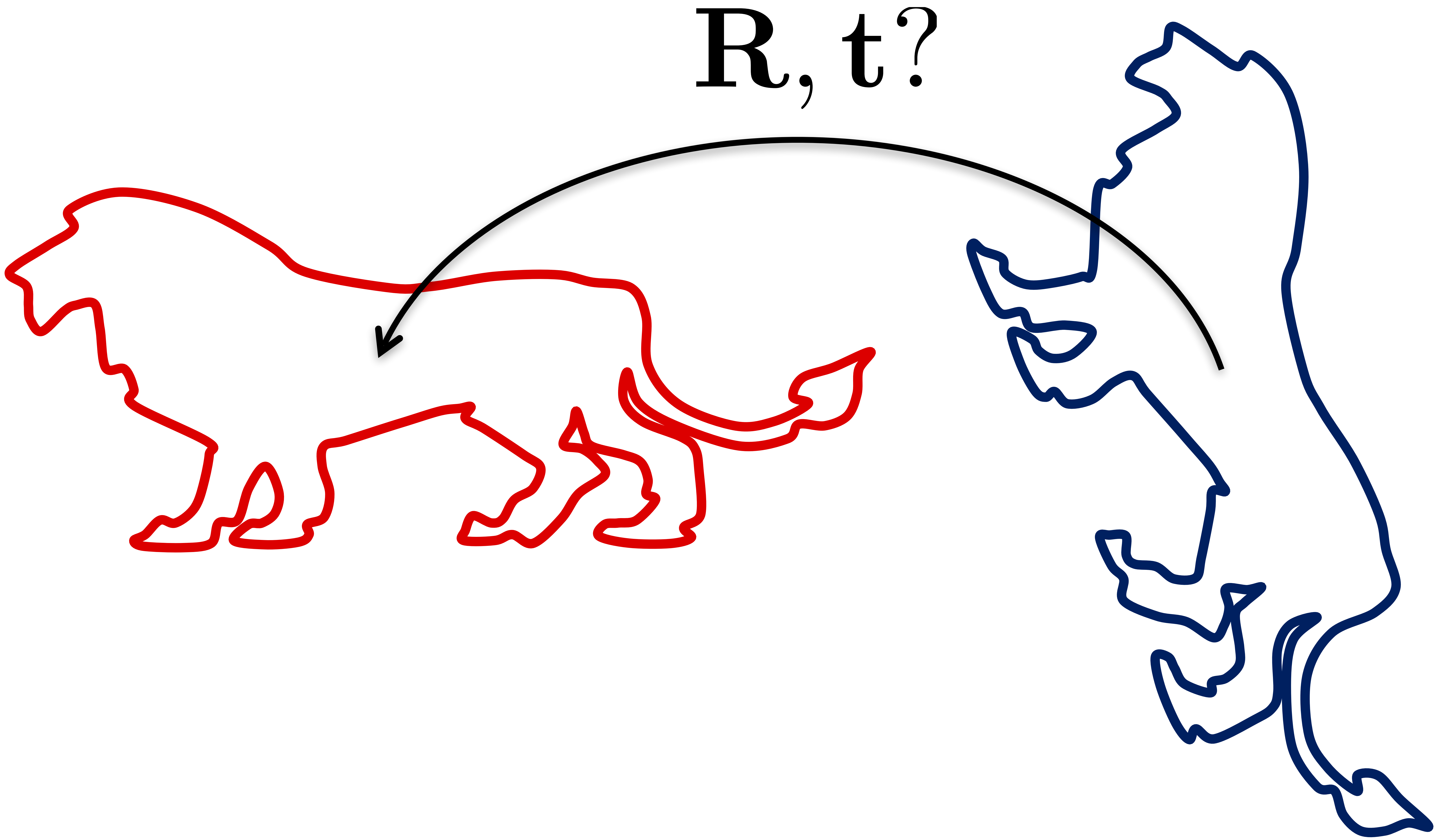
- How many pairs?
  - Three pairs are sufficient in principle, but not robust enough
  - We need several pairs to compensate for acquisition and identification error
  - Problem becomes over-determined



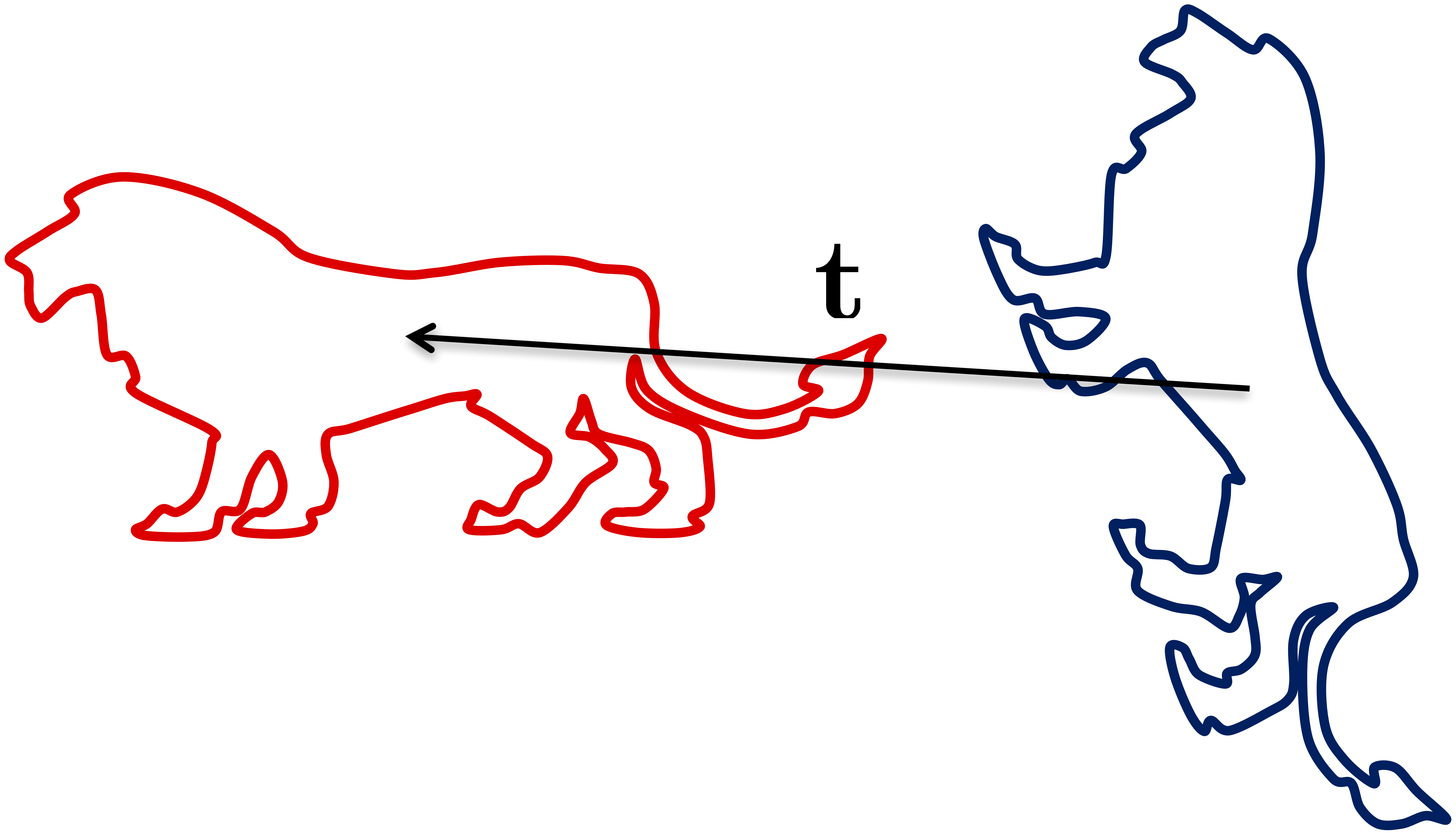
Finding Correspondences

- Manual pick is ok for 2 range maps, boring for 10, unbearable for 400…
- Automatic solutions: find “feature” points on two (or more) surfaces and their correspondences
- Feature descriptors from computer vision:
  - Spin images
  - SIFT
  - HOG
  - Curvature maps

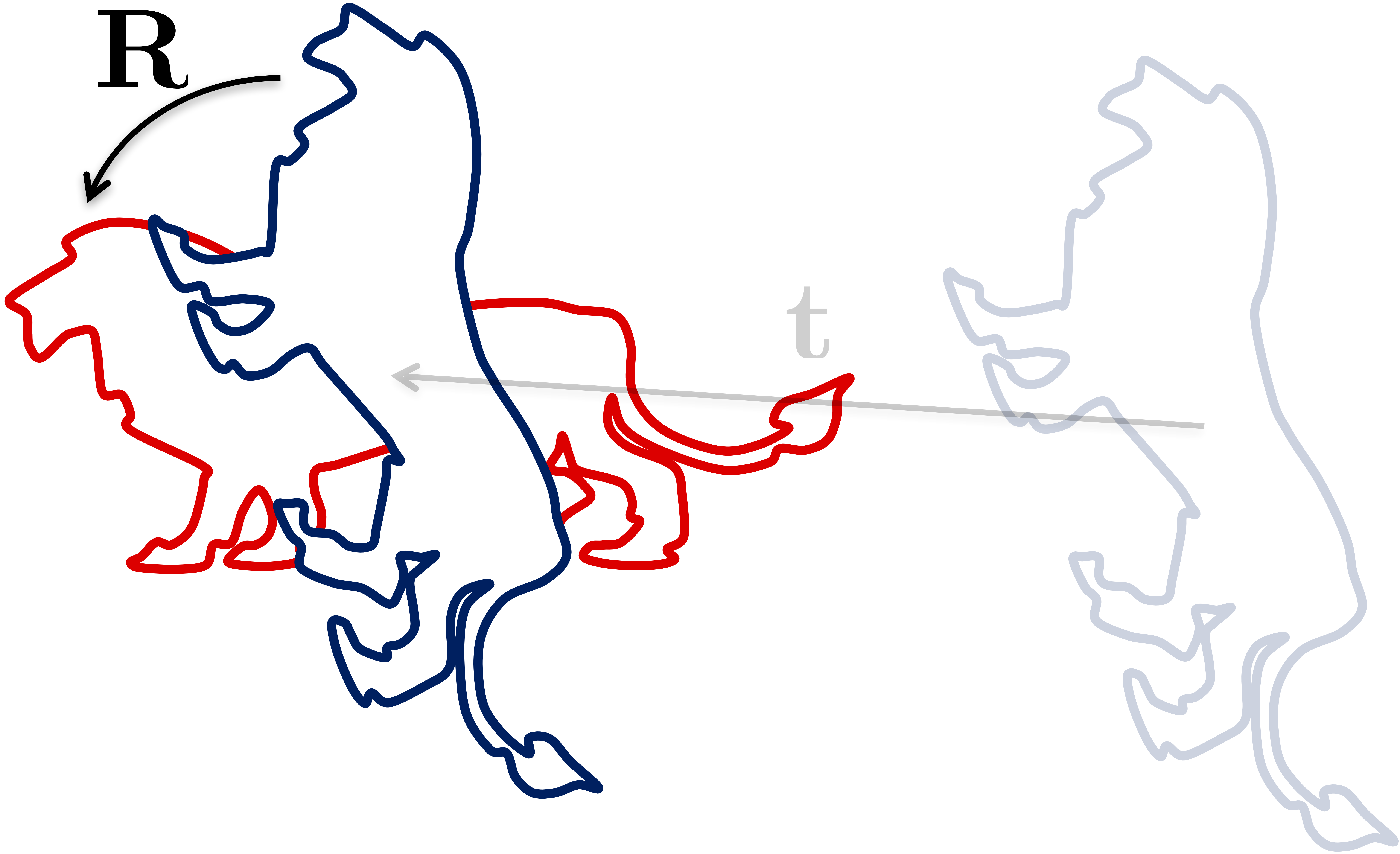
Shape Matching Problem



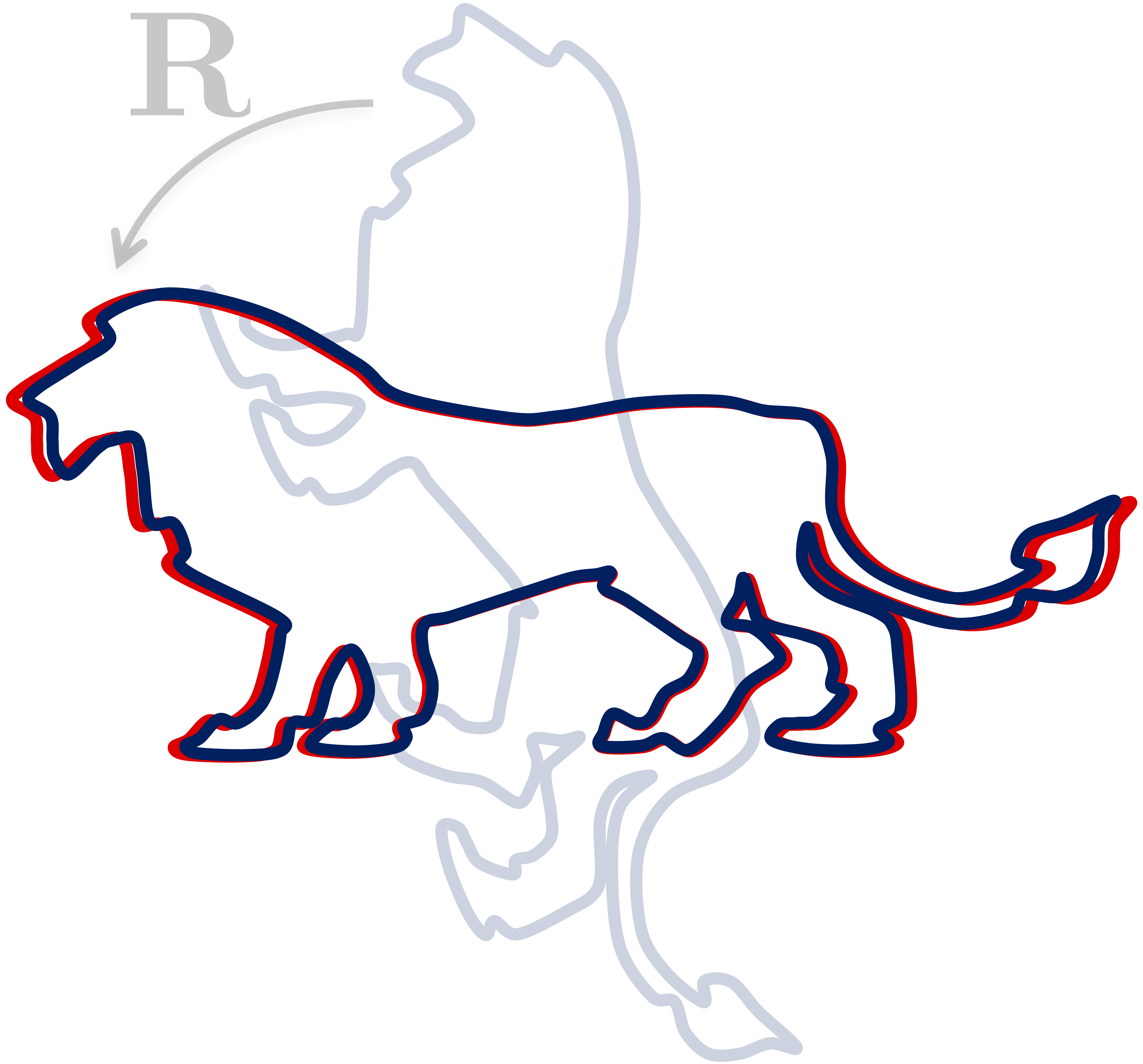
Shape Matching Problem



Shape Matching Problem



Shape Matching Problem



Shape Matching Problem

- Align two point sets

$$\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \text{ and } \mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$$

- Find a translation vector $\mathbf{t}$ and rotation matrix $\mathbf{R}$ so that

$$\sum_{i=1}^{n} \|(\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\|^2 \text{ is minimized}$$

Shape Matching – Solution

- Solve for translation first (w.r.t. **R**, **p**, and **q**)

$$\frac{\partial}{\partial \mathbf{t}} \sum_{i=1}^{n} \|(\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\|^2 = \sum_{i=1}^{n} 2\left((\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\right) \stackrel{!}{=} 0$$

$$\mathbf{R}\sum_{i=1}^{n}\mathbf{p}_i + \sum_{i=1}^{n}\mathbf{t} - \sum_{i=1}^{n}\mathbf{q}_i = 0$$

$$\mathbf{t} = \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{q}_i\right)}_{\bar{\mathbf{q}}} - \mathbf{R}\underbrace{\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_i\right)}_{\bar{\mathbf{p}}}$$

Point sets $\{\mathbf{q}_i\}$ and $\{\mathbf{R}\mathbf{p}_i\}$ have the same center of mass

Finding the Rotation $\mathbf{R}$

- To find the optimal $\mathbf{R}$, we bring the centroids of both point sets to the origin

$$\mathbf{v}_i = \mathbf{p}_i - \bar{\mathbf{p}}, \qquad \mathbf{v}'_i = \mathbf{q}_i - \bar{\mathbf{q}}$$

- We want to find $\mathbf{R}$ that minimizes

$$\sum_{i=1}^{n} \|\mathbf{R}\mathbf{v}_i - \mathbf{v}'_i\|^2$$

Finding the Rotation $\mathbf{R}$

$$\sum_{i=1}^{n} \|\mathbf{R}\mathbf{v}_i - \mathbf{v}'_i\|^2 = \sum_{i=1}^{n} (\mathbf{R}\mathbf{v}_i - \mathbf{v}'_i)^T (\mathbf{R}\mathbf{v}_i - \mathbf{v}'_i) =$$

$$= \sum_{i=1}^{n} \left(\mathbf{v}_i^T \underbrace{\mathbf{R}^T\mathbf{R}}_{\mathbf{I}} \mathbf{v}_i - {\mathbf{v}'_i}^T \mathbf{R}\mathbf{v}_i - \mathbf{v}_i^T \mathbf{R}^T \mathbf{v}'_i + {\mathbf{v}'_i}^T \mathbf{v}'_i \right)$$

These terms do not depend on **R**,
so we can ignore them in the minimization

Finding the Rotation $\mathbf{R}$

$$\underset{\mathbf{R}\in SO(3)}{\operatorname{argmin}} \sum_{i=1}^{n} \left(-\mathbf{v}_i'^T \mathbf{R} \mathbf{v}_i - \mathbf{v}_i^T \mathbf{R}^T \mathbf{v}_i' \right) = \underset{\mathbf{R}\in SO(3)}{\operatorname{argmax}} \sum_{i=1}^{n} \left(\mathbf{v}_i'^T \mathbf{R} \mathbf{v}_i + \underbrace{\mathbf{v}_i^T \mathbf{R}^T \mathbf{v}_i'} \right) =$$

$$= \boxed{\underset{\mathbf{R}\in SO(3)}{\operatorname{argmax}} \sum_{i=1}^{n} \mathbf{v}_i'^T \mathbf{R} \mathbf{v}_i}$$

$$\mathbf{v}_i^T \mathbf{R}^T \mathbf{v}_i' = \left(\mathbf{v}_i^T \mathbf{R}^T \mathbf{v}_i' \right)^T = \mathbf{v}_i'^T \mathbf{R} \mathbf{v}_i$$

It's a scalar!

Finding the Rotation $\mathbf{R}$

$$\sum_{i=1}^{n} \mathbf{v}_i'^{T} \mathbf{R} \mathbf{v}_i = tr\left(\mathbf{V}'^{T} \mathbf{R} \mathbf{V}\right)$$

$$\underbrace{\begin{bmatrix} \mathbf{v}_1'^{T} \\ \mathbf{v}_2'^{T} \\ \vdots \\ \mathbf{v}_n'^{T} \end{bmatrix}}_{\mathbf{V}'^{T}} \begin{bmatrix} \mathbf{R} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \end{bmatrix}}_{\mathbf{V}} = \begin{bmatrix} \mathbf{v}_1'^{T} \\ \mathbf{v}_2'^{T} \\ \vdots \\ \mathbf{v}_n'^{T} \end{bmatrix} \begin{bmatrix} \mathbf{R}\mathbf{v}_1 & \mathbf{R}\mathbf{v}_2 & \cdots & \mathbf{R}\mathbf{v}_n \end{bmatrix}$$

Finding the Rotation $\mathbf{R}$

$$\sum_{i=1}^{n} \mathbf{v}_i'^T \mathbf{R} \mathbf{v}_i = tr\left(\mathbf{V}'^T \mathbf{R} \mathbf{V}\right)$$

$$\begin{bmatrix} \mathbf{v}_1'^T \\ \mathbf{v}_2'^T \\ \vdots \\ \mathbf{v}_n'^T \end{bmatrix} \begin{bmatrix} \mathbf{R}\mathbf{v}_1 & \mathbf{R}\mathbf{v}_2 & \cdots & \mathbf{R}\mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1'^T \mathbf{R}\mathbf{v}_1 & & & \\ & \mathbf{v}_2'^T \mathbf{R}\mathbf{v}_2 & & \\ & & \ddots & \\ & & & \mathbf{v}_n'^T \mathbf{R}\mathbf{v}_n \end{bmatrix}$$

Finding the Rotation $\mathbf{R}$

Take a look at the Matrix Cookbook!

- Find $\mathbf{R}$ that maximizes

$$tr\left(\mathbf{V'}^T\mathbf{R}\mathbf{V}\right) = tr\left(\mathbf{R}\mathbf{V}\mathbf{V'}^T\right)$$

- SVD: $\mathbf{V}\mathbf{V'}^T = \mathbf{U}\mathbf{\Sigma}\tilde{\mathbf{U}}^T$

$$tr\left(\mathbf{R}\mathbf{V}\mathbf{V'}^T\right) = tr\left(\underbrace{\mathbf{R}\,\mathbf{U}}\underbrace{\mathbf{\Sigma}\tilde{\mathbf{U}}^T}\right) = tr\left(\mathbf{\Sigma}\,\underbrace{\tilde{\mathbf{U}}^T\mathbf{R}\mathbf{U}}_{\text{orthonormal matrix}}\right)$$

Finding the Rotation $\mathbf{R}$

- We want to maximize

$$tr\left(\Sigma\,\mathbf{M}\right)$$

$\mathbf{M}$: orthonormal matrix
all entries ≤ 1

$$\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} \begin{bmatrix} m_{11} & \cdots & \\ \vdots & m_{22} & \vdots \\ & \cdots & m_{33} \end{bmatrix}$$

$$tr\left(\Sigma\,\mathbf{M}\right) = \sum_{i=1}^{3} \sigma_i\, m_{ii} \leq \sum_{i=1}^{3} \sigma_i$$

Finding the Rotation $\mathbf{R}$

$$tr\,(\Sigma\,\mathbf{M}) = \sum_{i=1}^{3} \sigma_i\, m_{ii} \leq \sum_{i=1}^{3} \sigma_i$$

- Our best shot is $m_{ii} = 1$, i.e. to make **M = I**

$$\begin{aligned} \mathbf{M} = \tilde{\mathbf{U}}^T\mathbf{R}\mathbf{U} \quad &\overset{!}{=} \quad \mathbf{I} \\ \mathbf{R}\mathbf{U} \quad &= \quad \tilde{\mathbf{U}} \\ \boxed{\mathbf{R} \quad = \quad \tilde{\mathbf{U}}\mathbf{U}^T} \end{aligned}$$

Summary of Rigid Alignment

- Translate the input points to the centroids

$$\mathbf{v}_i = \mathbf{p}_i - \bar{\mathbf{p}}, \qquad \mathbf{v}'_i = \mathbf{q}_i - \bar{\mathbf{q}}$$

- Compute the "covariance matrix" $\mathbf{V}\mathbf{V}'^T$

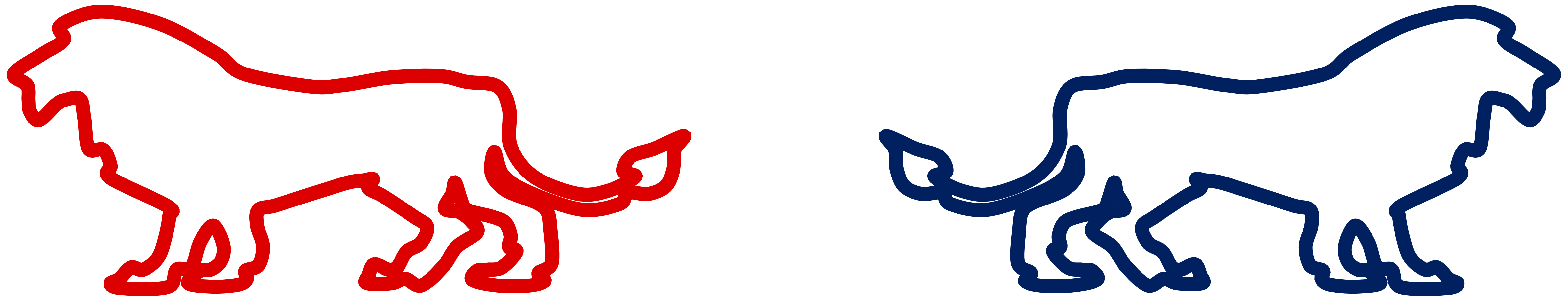
- Compute its SVD: $\mathbf{V}\mathbf{V}'^T = \mathbf{U}\mathbf{\Sigma}\tilde{\mathbf{U}}^T$

- The optimal orthonormal $\mathbf{R}$ is

$$\mathbf{R} = \tilde{\mathbf{U}}\mathbf{U}^T$$

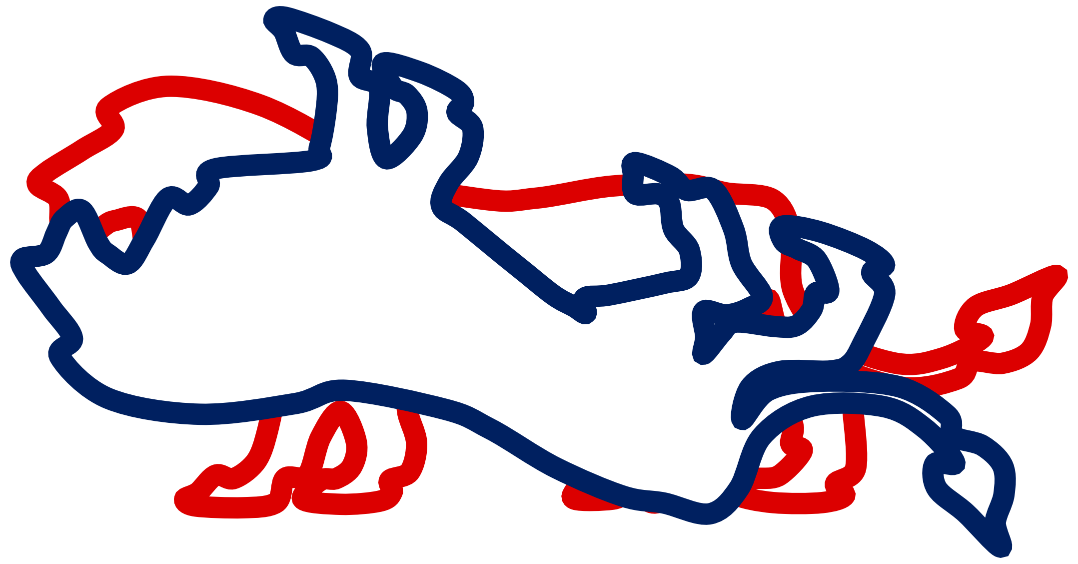
Sign Correction

- It is possible that $\det(\tilde{\mathbf{U}}\mathbf{U}^T) = -1$: sometimes reflection is the best orthonormal transform



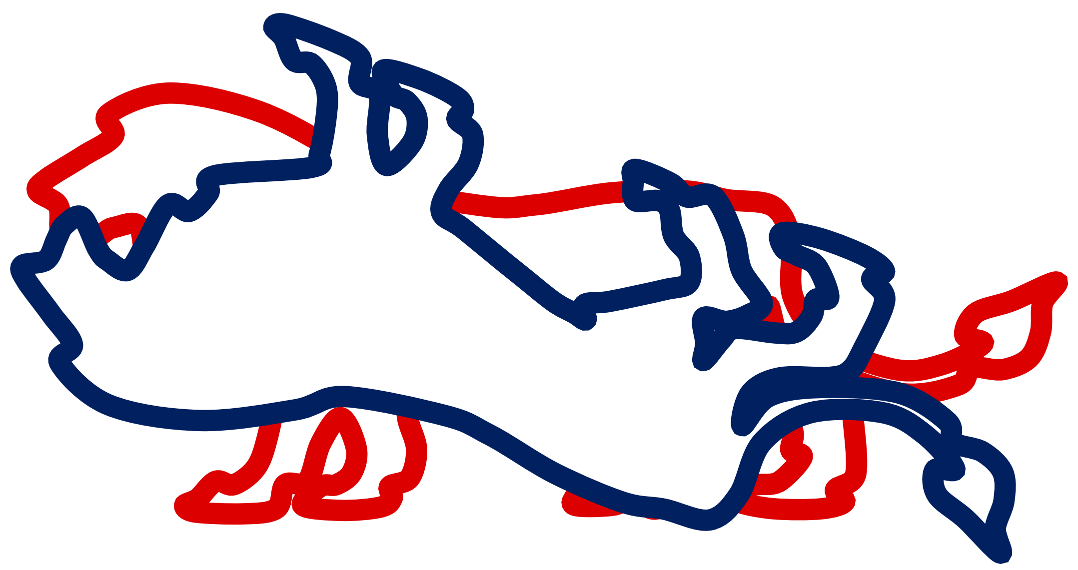
Sign Correction

- It is possible that $\det(\tilde{\mathbf{U}}\mathbf{U}^T) = -1$: sometimes reflection is the best orthonormal transform



Sign Correction

- To restrict ourselves to rotations only:
  take the last column of $\mathbf{U}$ (corresponding to the smallest singular value) and invert its sign.



- Why? See http://igl.ethz.ch/projects/ARAP/svd_rot.pdf

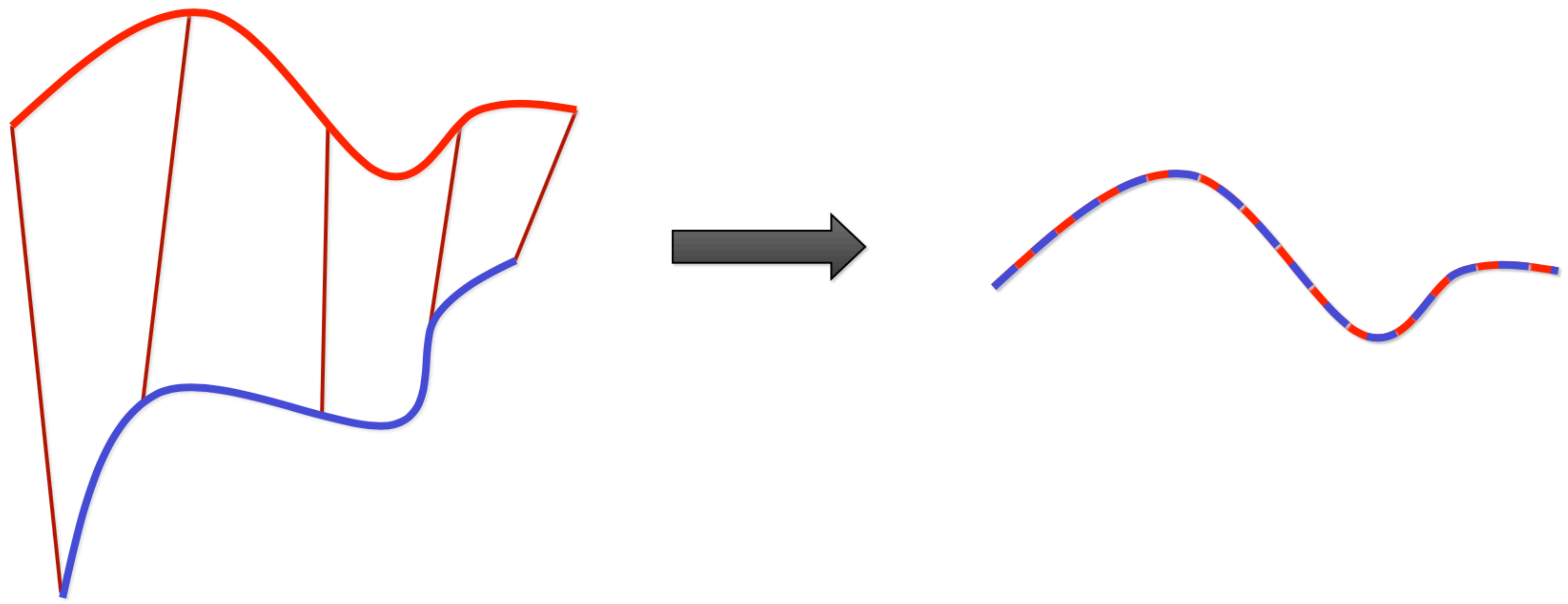
Fine registration

- Input: two overlapping range maps roughly aligned
- Output: fine alignment of range maps in the same reference frame
- Assumption: for any point **p** in a map its corresponding point **q** in the other map is near
- No need to use feature points
- Fully automatic approach

ICP: Iterative Closest Point

- Idea: Iteratively (1) find correspondences and (2) use them to find a transformation

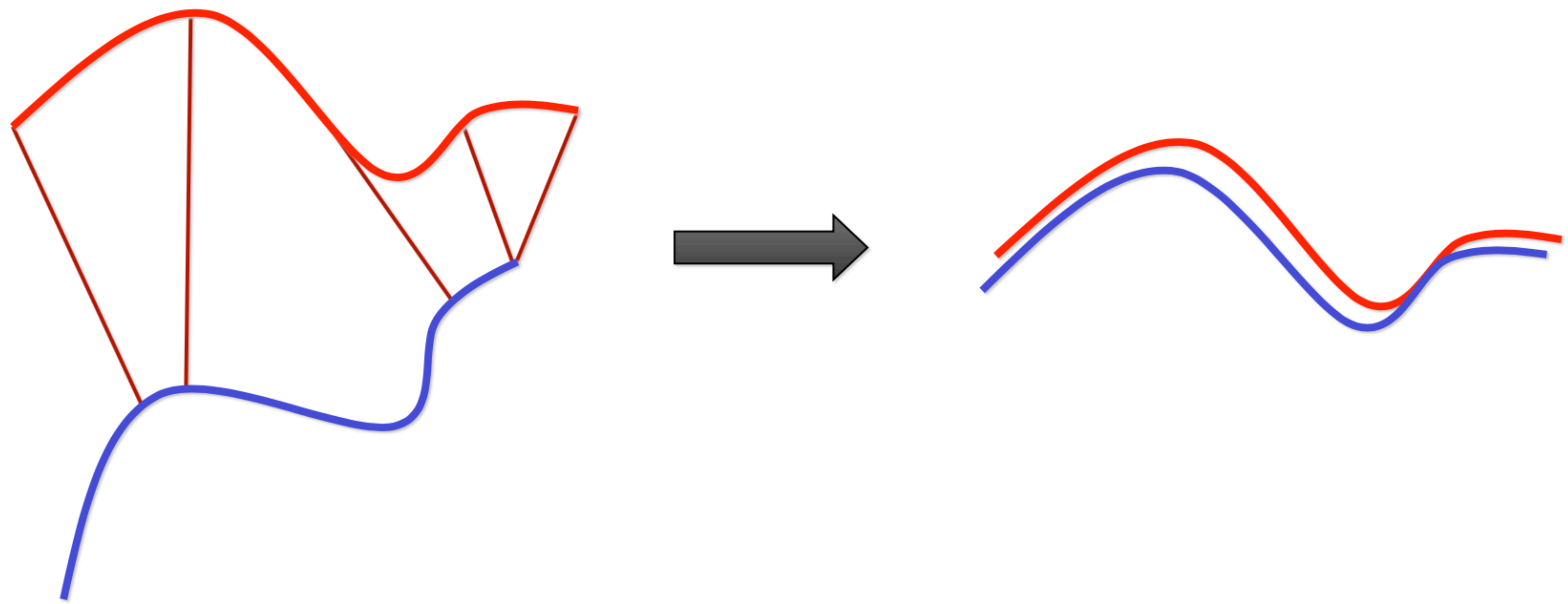
- Intuition: If you have the right correspondences, then the problem is easy



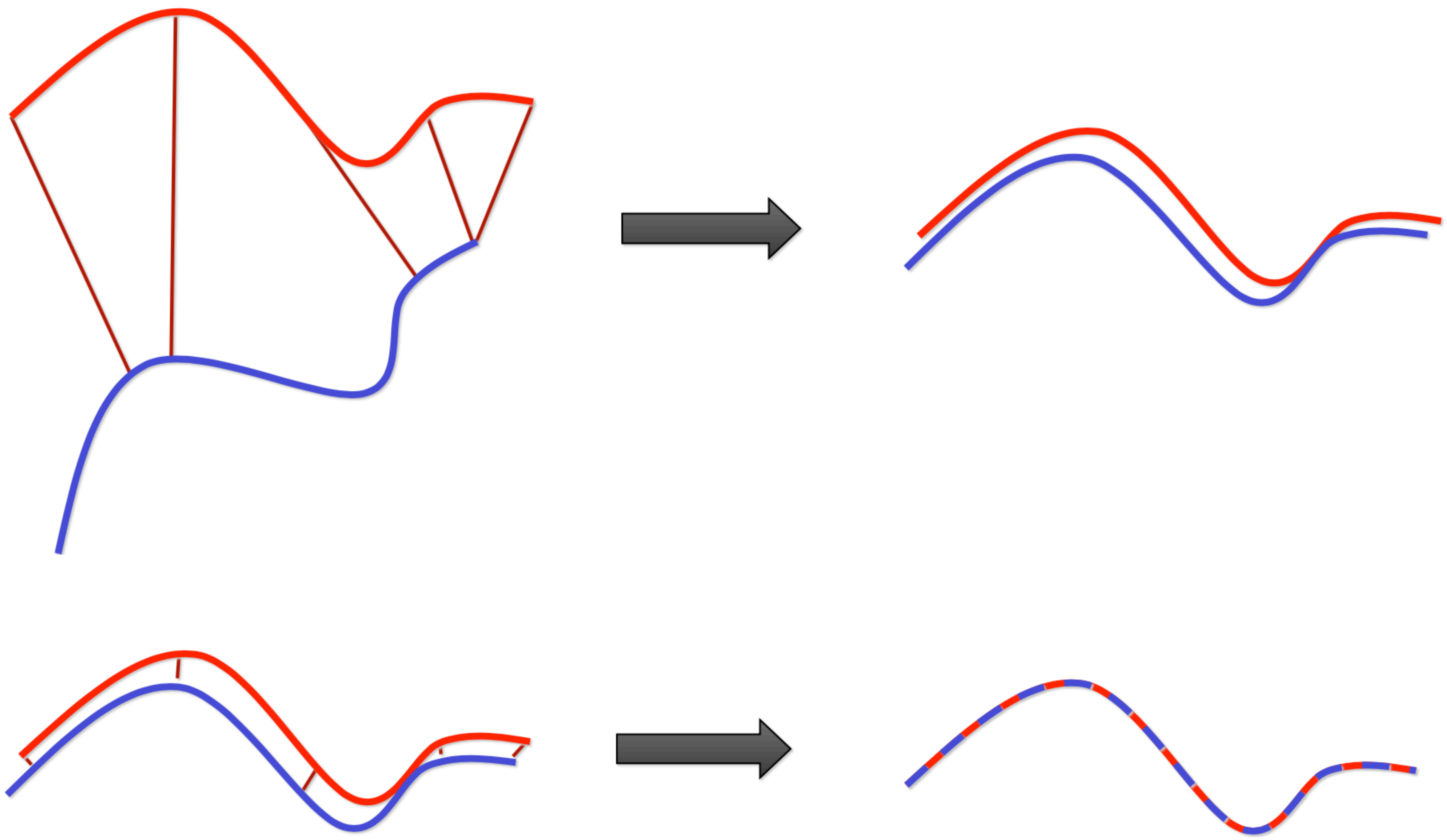
ICP: Iterative Closest Point

- 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”

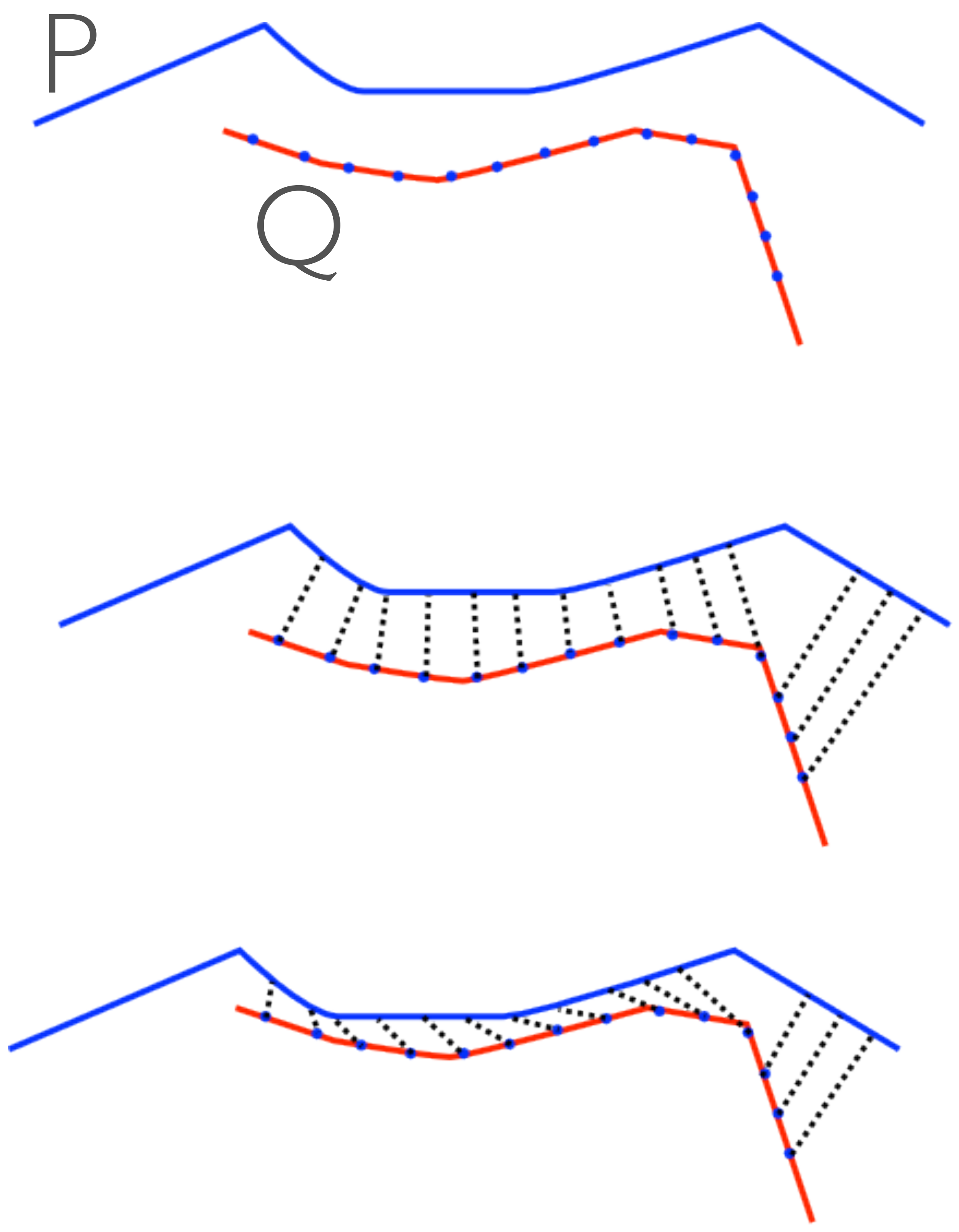
Basic Algorithm

- Select (e.g., 1000) random points
- Match each to closest point on other scan, using data structure such as *k*-d tree
- Reject pairs with distance > *k* times median
- Construct error function:

$$E := \sum_i (R\mathbf{p}_i + t - \mathbf{q}_i)^2$$

- Minimize as before (closed form solution in “Estimating 3-D rigid body transformations: a comparison of four major algorithms”, http://dl.acm.org/citation.cfm?id=250160)

Basic Algorithm



sample points on Q

$$Q_0 = Npoints \ \in Q, RT = \mathbf{I}$$

find the closest on P

$$\forall q_j \in Q_0, find\ closest(RT_k\ q_j)$$

minimize

$$M = argmin : \frac{1}{N} \sum_{j \in N} \|closest(RT_k q_j) - M\ \ RT_k q_j\|^2$$

$$RT_{k+1} = M\ \ RT_{k+1}$$

$$d_k = \frac{1}{N} \sum_{j \in N} \|closest(RT_k q_j) - RT_{k+1} q_j\|^2$$

until convergence

ICP: convergence proof

$$e_k = \frac{1}{N}\sum_{j \in N} \|closest(RT_k q_j) - RT_k q_j\|^2$$
$$d_k = \frac{1}{N}\sum_{j \in N} \|closest(RT_k q_j) - RT_{k+1}\ q_j\|^2$$
$$\Rightarrow d_k \leq e_k$$

i.e. if we kept the old correspondences

if $\boxed{closest(RT_k\ q_j) = closest(RT_{k+1}\ q_j)}$ we would have

$$e_{k+1} = d_k = \frac{1}{N}\sum_{j \in N} \|closest(RT_k q_j) - RT_{k+1} q_j\|^2$$

but, by construction of *closest*:

$$\|closest(RT_{k+1} q_j) - RT_{k+1} q_j\|^2 \leq \|closest(RT_k q_j) - RT_{k+1} q_j\|^2$$

Therefore:

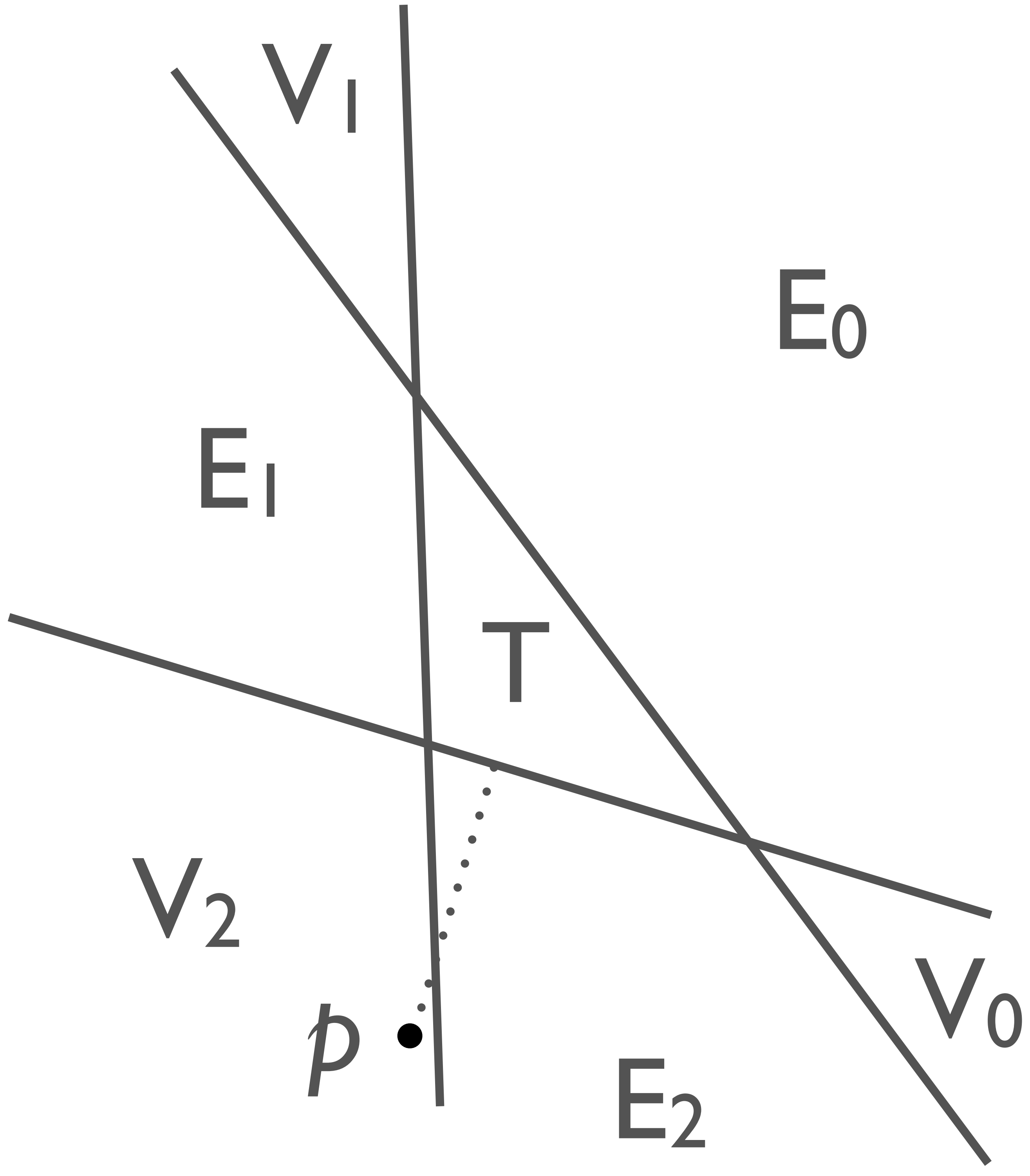
$$d_{k+1} \leq e_{k+1} \leq d_k$$

Finding closest point

- Problem: given a point q on Q, find its closest point on P
  - P made of triangles: distance point-to-triangle
  - P contains many triangles: which ones to test?

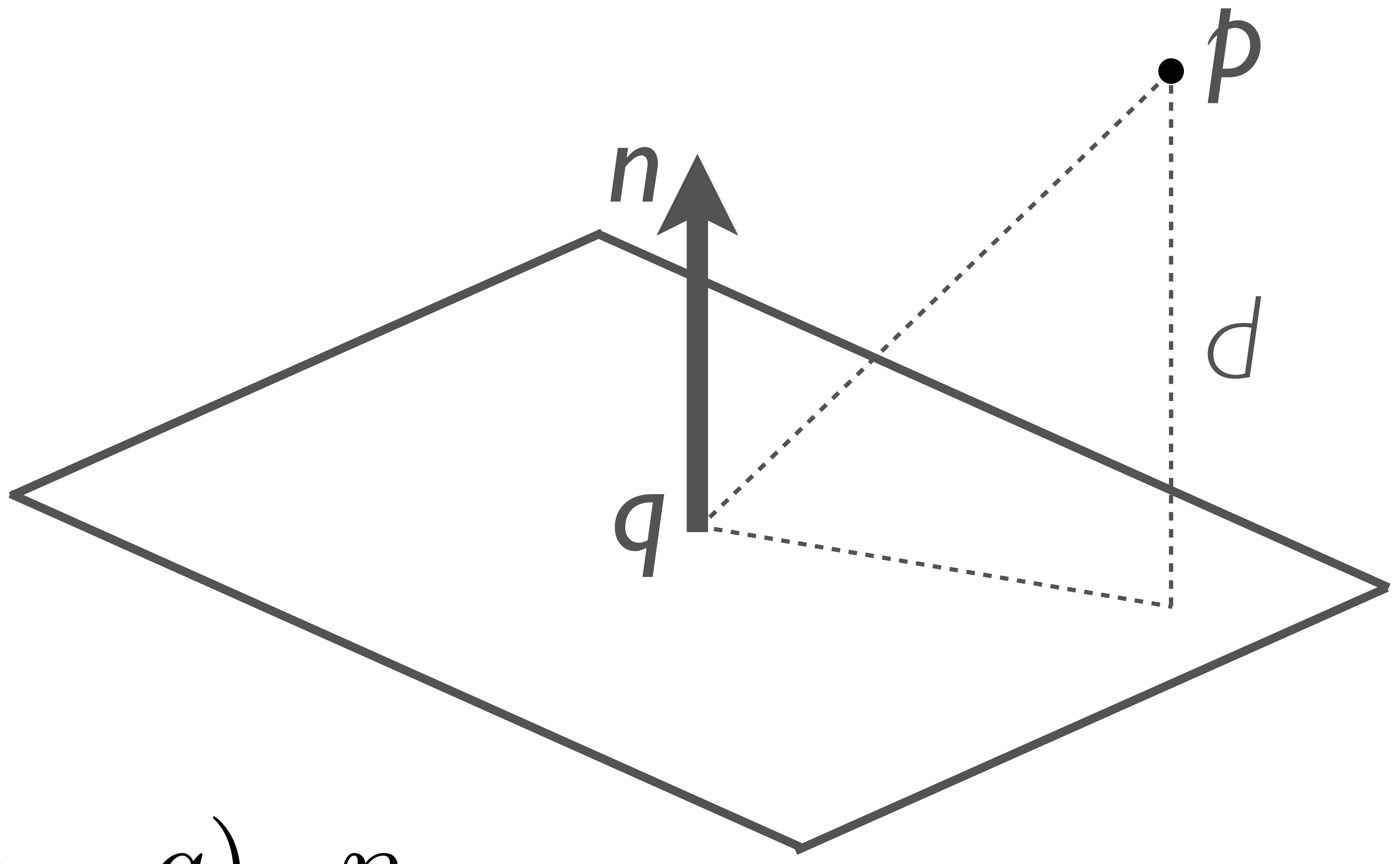
Finding closest point

- Given triangle t and point p, consider planes through edges of t and orthogonal to the plane of t
- they divide 3D space into 7 zones $V_0, V_1, V_2, E_0, E_1, E_2, T$
- If p inside T then take distance from p to the plane of t
- If p inside E_i then take distance from p to edge e_i
- If p inside V_i then take smallest distance from p to edges e_i and e_{i+2}



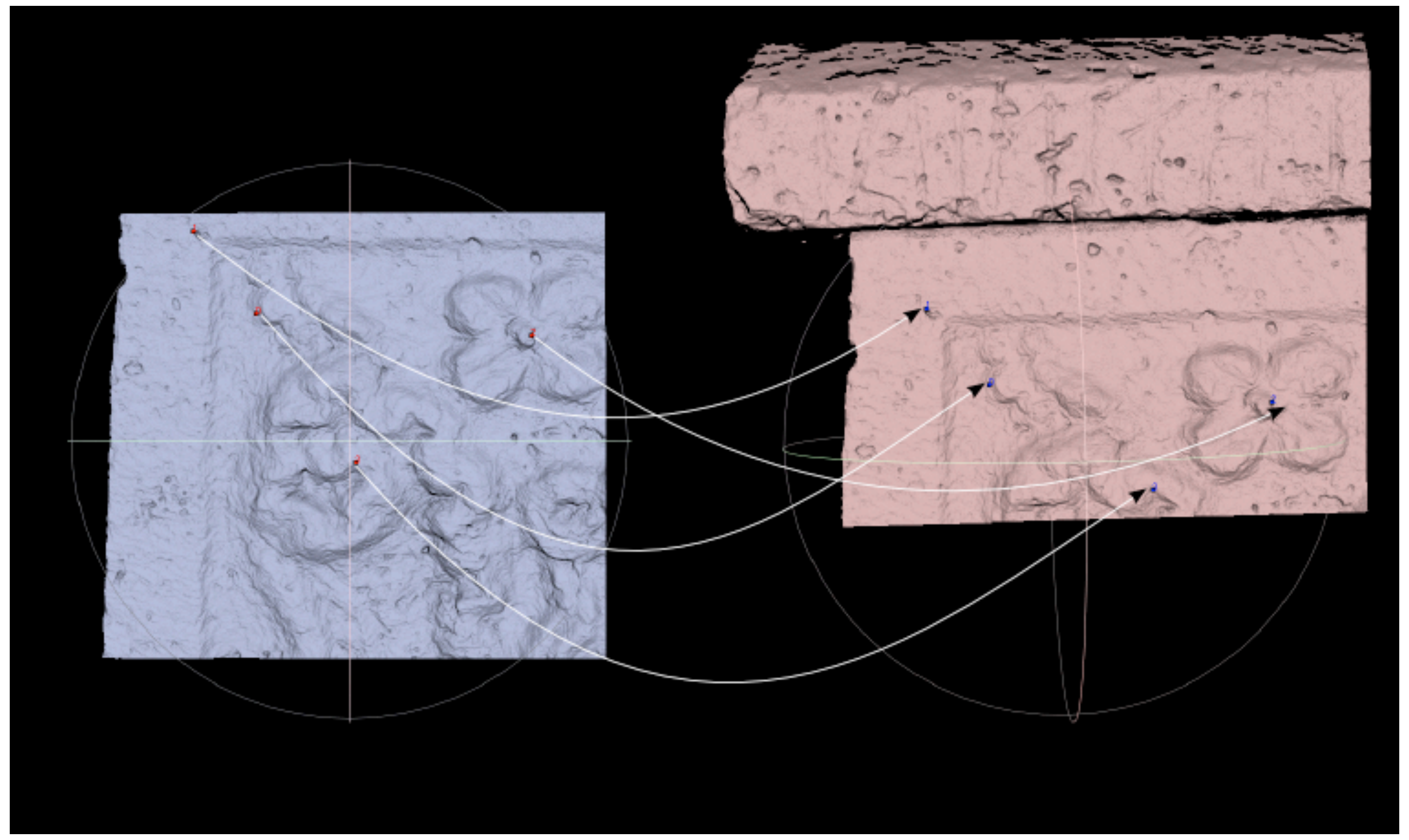
Finding closest point

- Distance point-to-plane:



$$d = (p - q) \cdot n$$

Local vs Global



Local Registration
Between two range maps
All-vs-one range map

Global Registration
All-vs-all

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

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

Thank you