Ransac & ICP

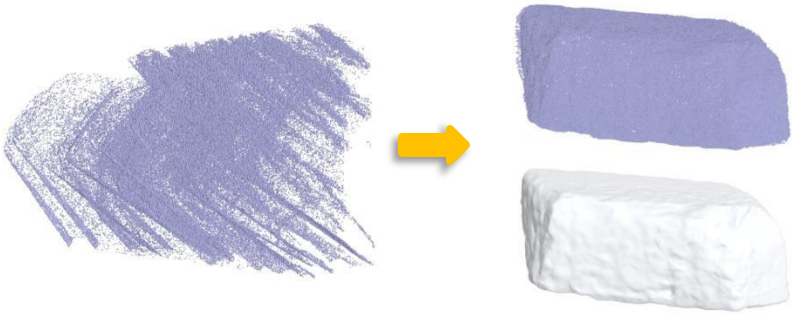
Paolo Cignoni

3D GEOMETRIC MODELING & PROCESSING

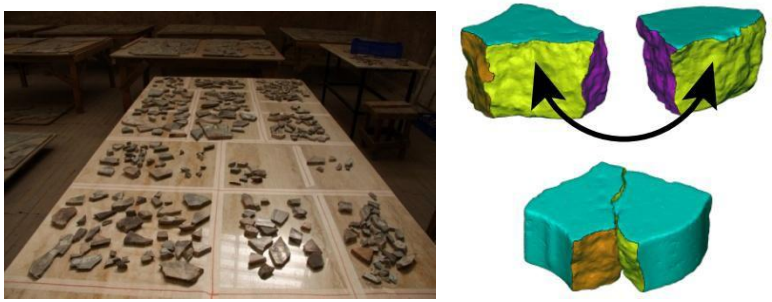
RANSAC

- RANSAC:
  Random Sample Consensus.
- It is a statistical method that is used to **estimate parameters** of a mathematical model from a set of observed data that contains outliers.
- Iterative method (can be interpreted as an outlier detection method)

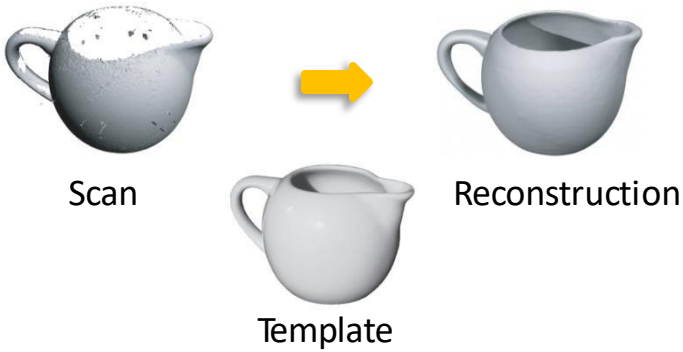
Applications



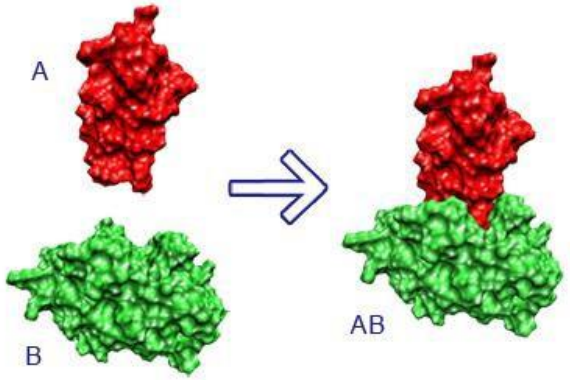
Surface reconstruction



Fragment assembly

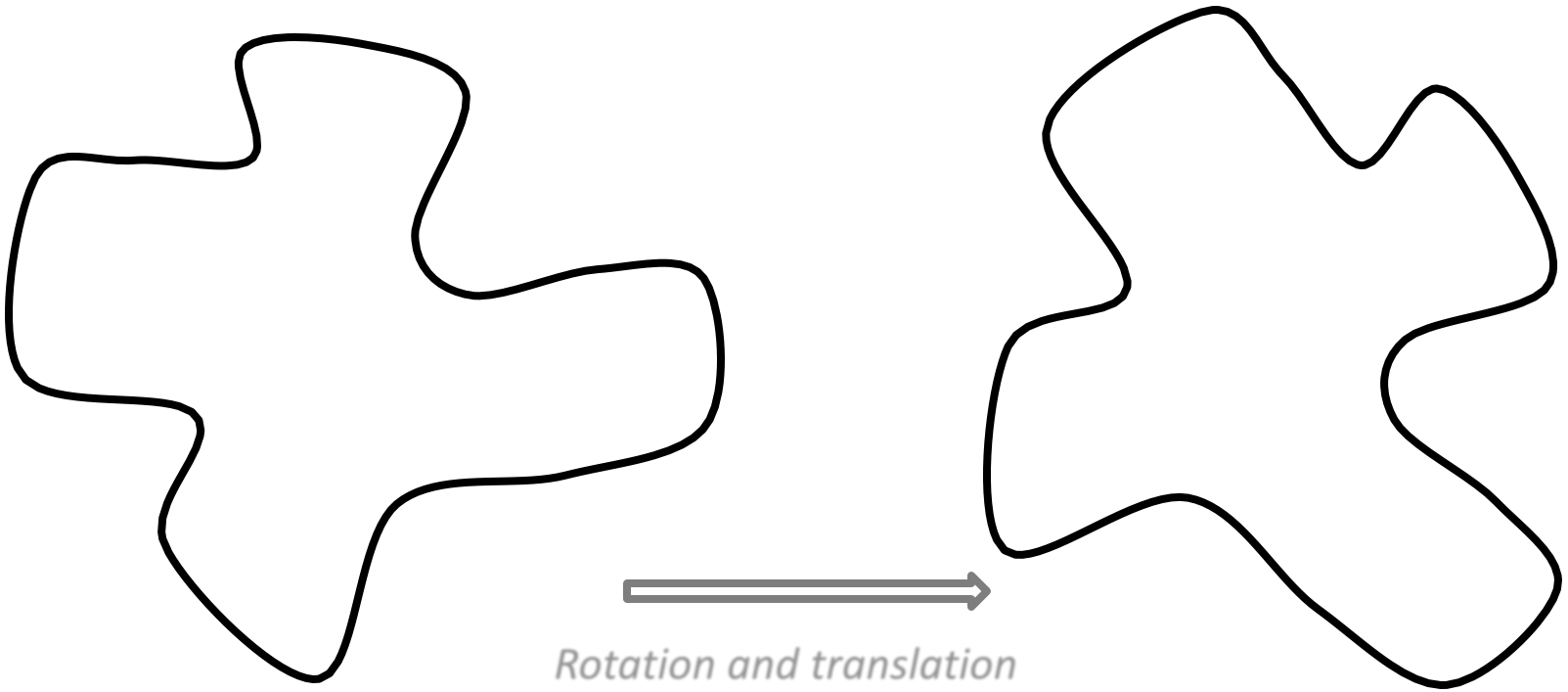


Object completion



Protein docking

Rigid Matching



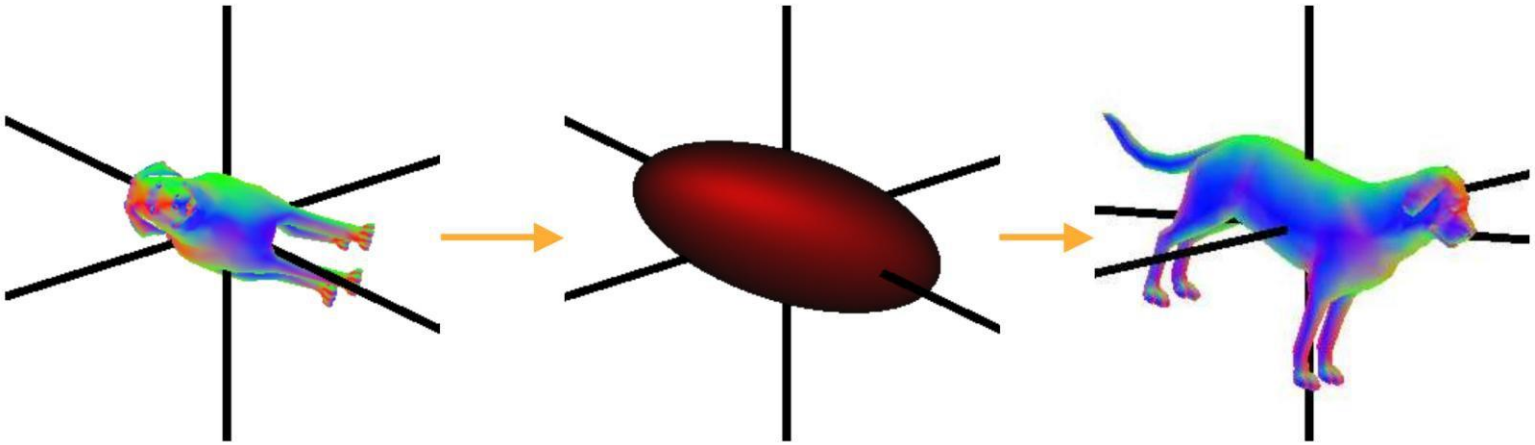
Rigid matching is a problem in computer vision, pattern recognition and computer graphics communities. It is the process of finding the transformation that maps one rigid object onto another

Approach --- PCA

- Use PCA to place models into a canonical coordinate frame

Covariance matrix computation

Principal Axis alignment



Principal Axis Computation

- Given a collection of points $\{\mathbf{p}_i\}$, form the co-variance matrix:

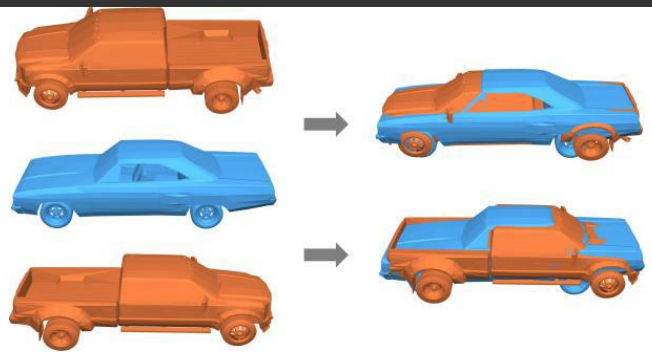
$$\mathbf{c} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{p}_i$$

$$C = \frac{1}{N} \sum_{i=1}^{N} \mathbf{p}_i \mathbf{p}_i^T - \mathbf{c}\mathbf{c}^T$$

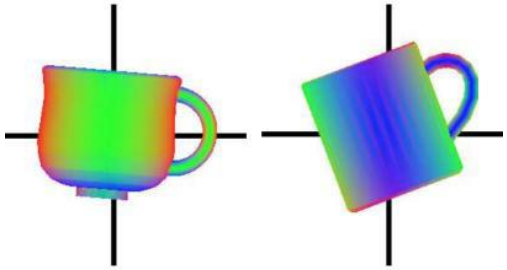
- Compute eigenvectors of matrix C

Issues with PCA

- Principal axes are not oriented



- Axes are unstable when principal values are similar



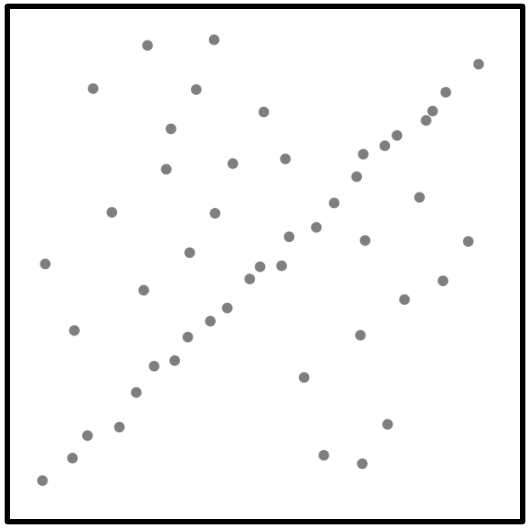
- Partial similarity

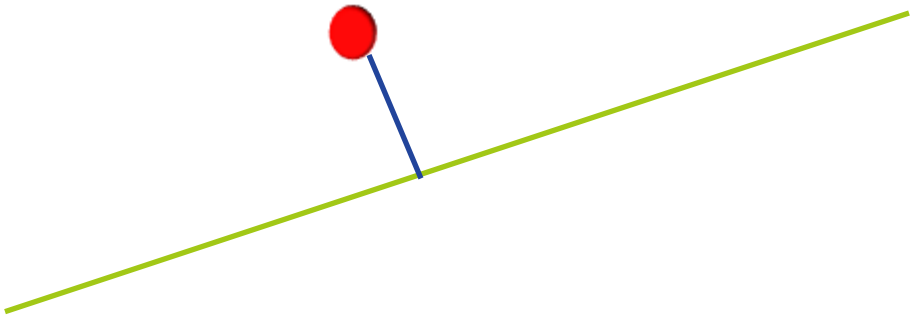
Ransac: Basis

- **Ran**dom **Sa**mple **C**onsensus
  - Hypothesize and test.
- Used for Parametric Matching
  - Want to match two things.
  - Hypothesized match can be described by parameters (eg., translation, affine.)
- Match enough features to determine a hypothesis.
  - See if it is good.
  - Repeat.

Ransac Example

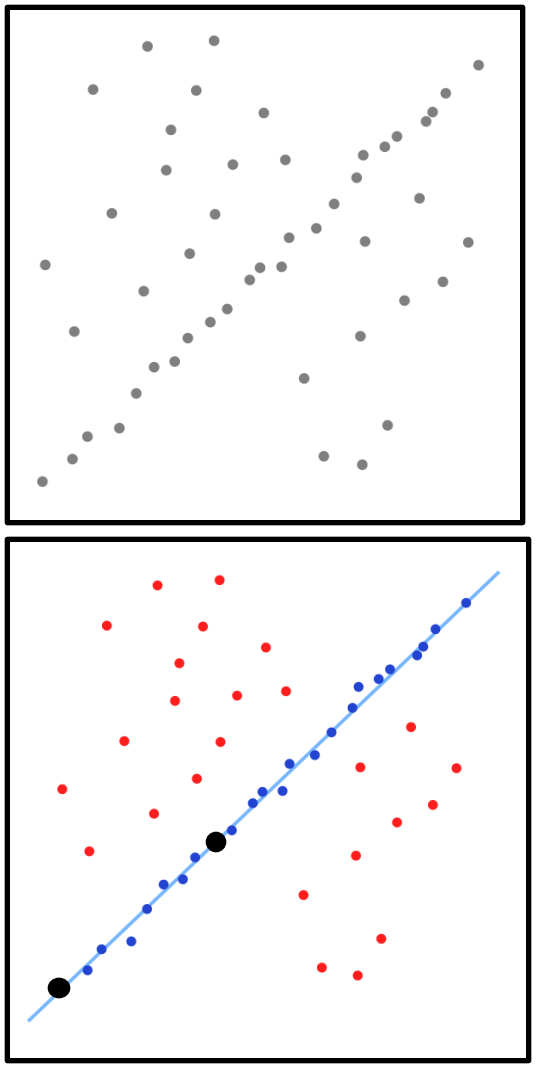
- Grouping Points into Lines
- Basic Info on lines





Ransac

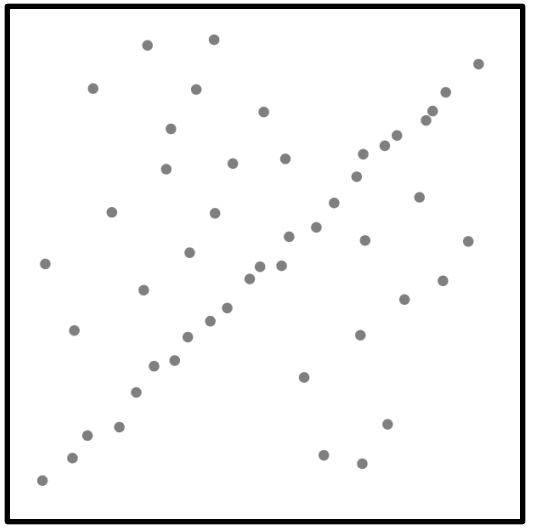
- Select a random subset of the original data. Call this subset the hypothetical inliers.
- A model is fitted to the set of hypothetical inliers.
- All other data are then tested against the fitted model.
- The estimated model is reasonably good if enough points have been classified as part of the consensus set.
- Afterwards, the model may be improved by re-estimating it using all members of the consensus set.

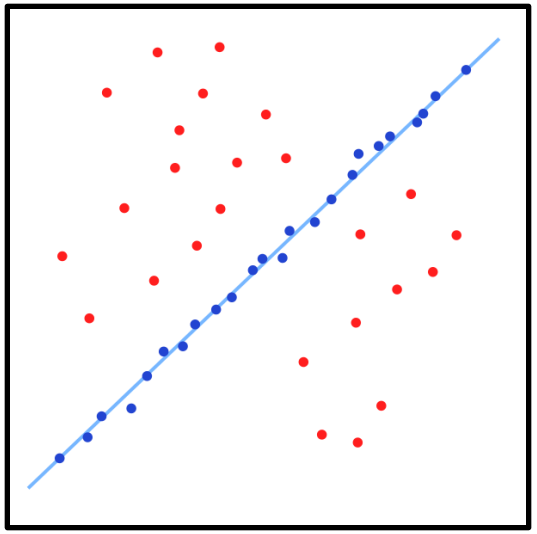


Ransac

- Complexity?
- How many samples?
  - **p** is fraction of points on the line.
    - Fraction of inlayer/total
  - **n** points needed to define hypothesis (2 for lines)
  - **k** number of trials.
- Probability that after **N** trials I've the correct solution is:

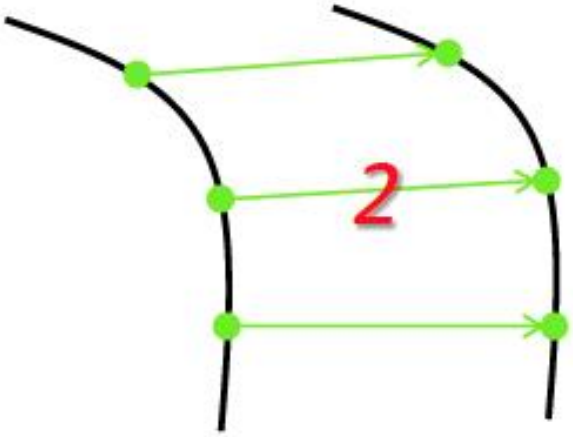
$$1-(1-p^n)^N$$

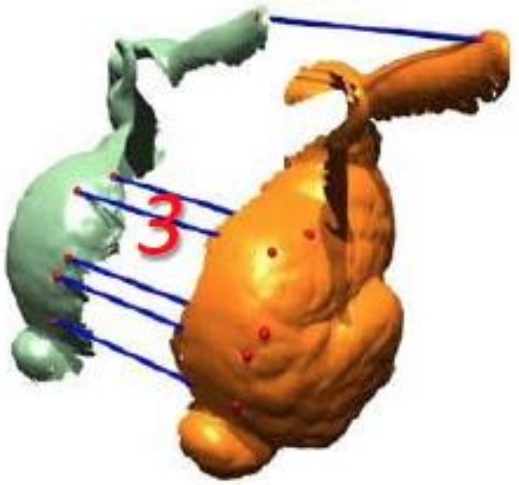




Ransac

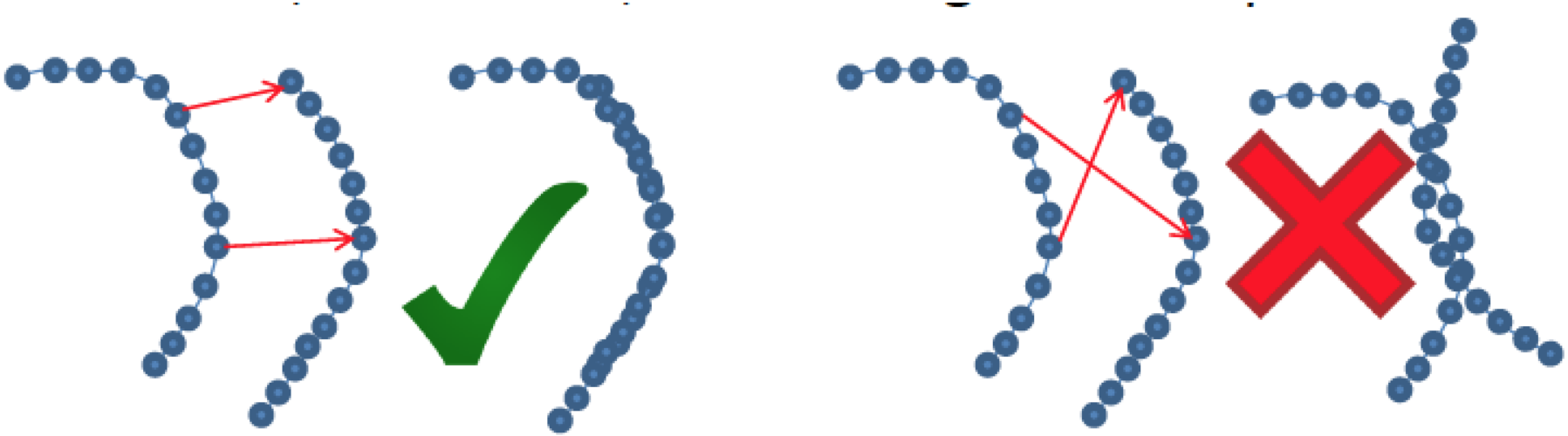
- How many point-pairs specify a rigid transform?
  - In R2?
  - In R3?
- Additional constraints?
  - Distance preserving
  - Stability?





Ransac

- Preprocessing: sample each object
- Iterate
  - Step I: Sample three (two) pairs, check distance constraints
  - Step II: Fit a rigid transform
  - Step III: Check how many point pairs agree. If above threshold, terminates; otherwise goes to Step I

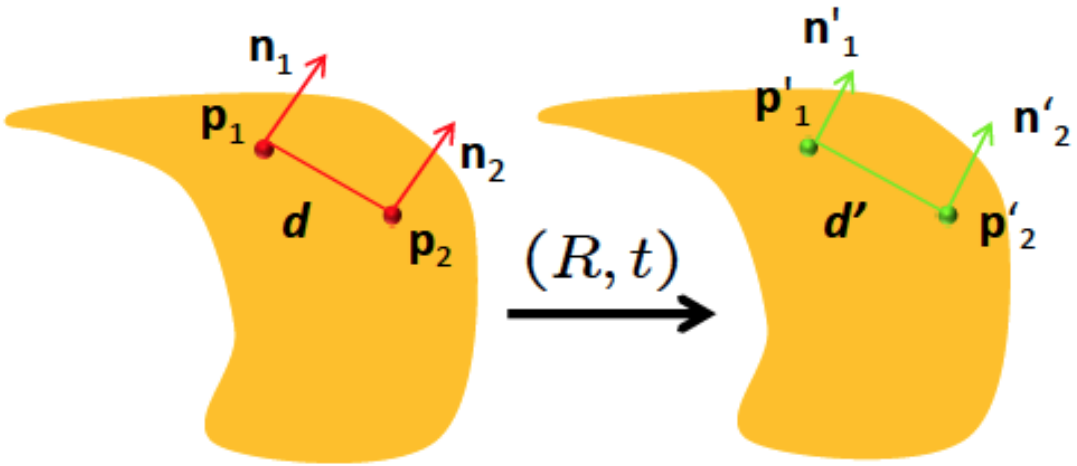


Ransac

- Sampling
  - Feature point detection
- Correspondences
  - Use feature descriptors
  - Denote a larger success rate p
  - Probability a descriptor identifies the correct match
  - Try only candidated made by pair of samplese with similar descriptor.
- Basic analysis
  - The probability of having a valid triplet $\mathbf{p^3}$
  - The probability of having a valid triplet in **N** trials is $\mathbf{1-(1-p^3)^N}$

Ransac +

- How many surfel (position + normal) correspondences specify a rigid transform?



Constraints:

1. $\|\mathbf{p}_1 - \mathbf{p}_2\| \approx \|\mathbf{p}'_1 - \mathbf{p}'_2\|$
2. $\angle(\mathbf{n}_1, \mathbf{d}) = \angle(\mathbf{n}'_1, \mathbf{d}')$
3. $\angle(\mathbf{n}_2, \mathbf{d}) = \angle(\mathbf{n}'_2, \mathbf{d}')$
4. $\angle(\mathbf{n}_1, \mathbf{n}_2) = \angle(\mathbf{n}'_1, \mathbf{n}'_2)$

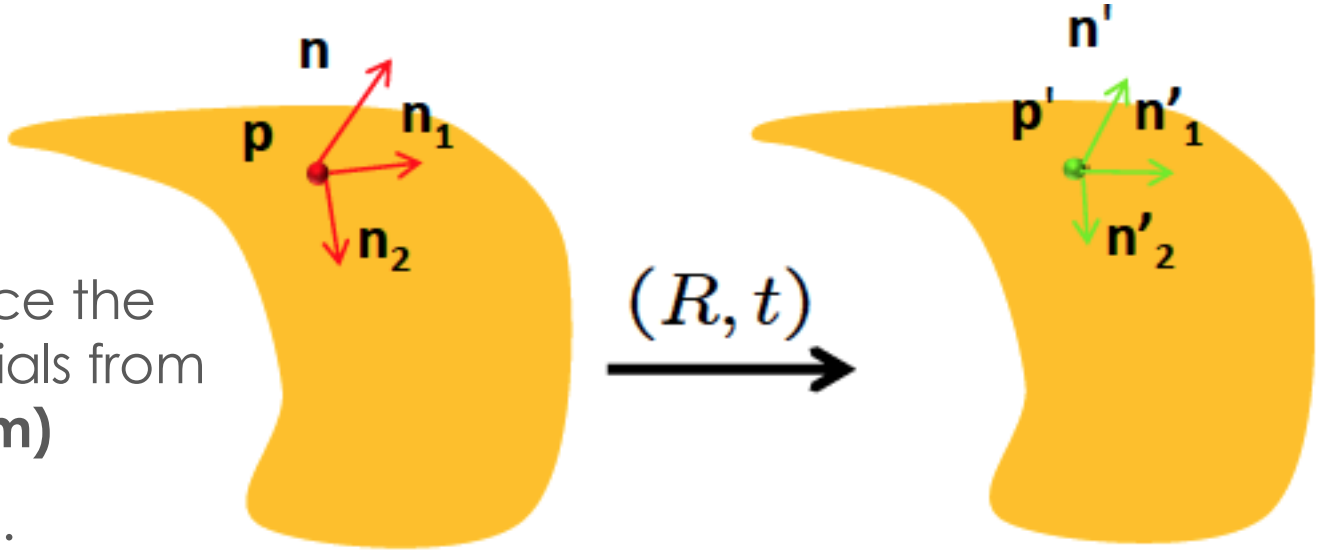
- Reduce the number of trials from **O(m³)** to **O(m²)**
- Success rate: **1-(1-p²)^N**

$$\mathbf{t} = \frac{\mathbf{p}'_1 + \mathbf{p}'_2}{2} - \frac{\mathbf{p}_1 + \mathbf{p}_2}{2}$$

$$[\mathbf{n}_1, \mathbf{n}_2, d] \xrightarrow{R} [\mathbf{n}'_1, \mathbf{n}'_2, \mathbf{d}']$$

Ransac ++

- How many frame correspondences specify a rigid transform?
  - Principal curvatures
  - Local PCA



- Further reduce the number of trials from **O(m²)** to **O(m)**
- Success rate: **1-(1-p)^N**

$$\mathbf{t} = \mathbf{p}' - \mathbf{p}$$

$$R(\mathbf{n}, \mathbf{n}_1, \mathbf{n}_2) \approx (\mathbf{n}', \mathbf{n}'_1, \mathbf{n}'_2)$$

Post Processing

- Refine the match via ICP

Input

After matching

After registration

icp

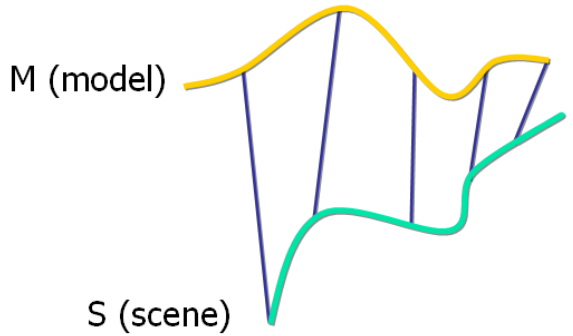
- **i**terative **c**losest **p**oint
- “a method for registration of 3-D shapes”, besl et al. - 1992
- “object modelling by registration of multiple range images”, chen et al. – 1991
- a lot of variants have been proposed for the original algorithm
- icp algorithm works with
  - point clouds
  - polygonal surfaces

corresponding point set alignment

- just consider for the moment the problem in 2D
- let M be a model point set.
- let S be a scene point set.

We assume :

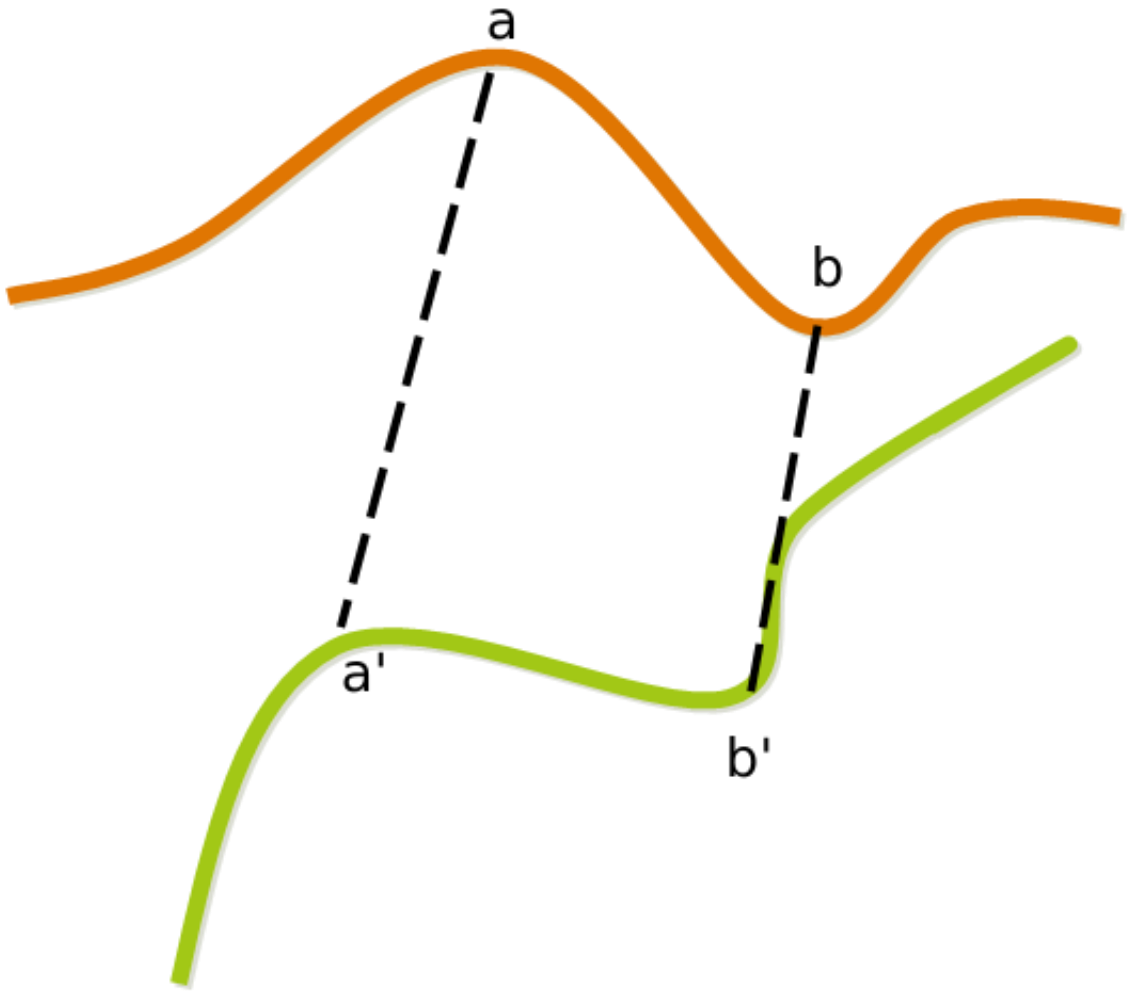
1. $n_M = n_S$. With $n \geq 2$
2. each point S_i correspond to M_i .



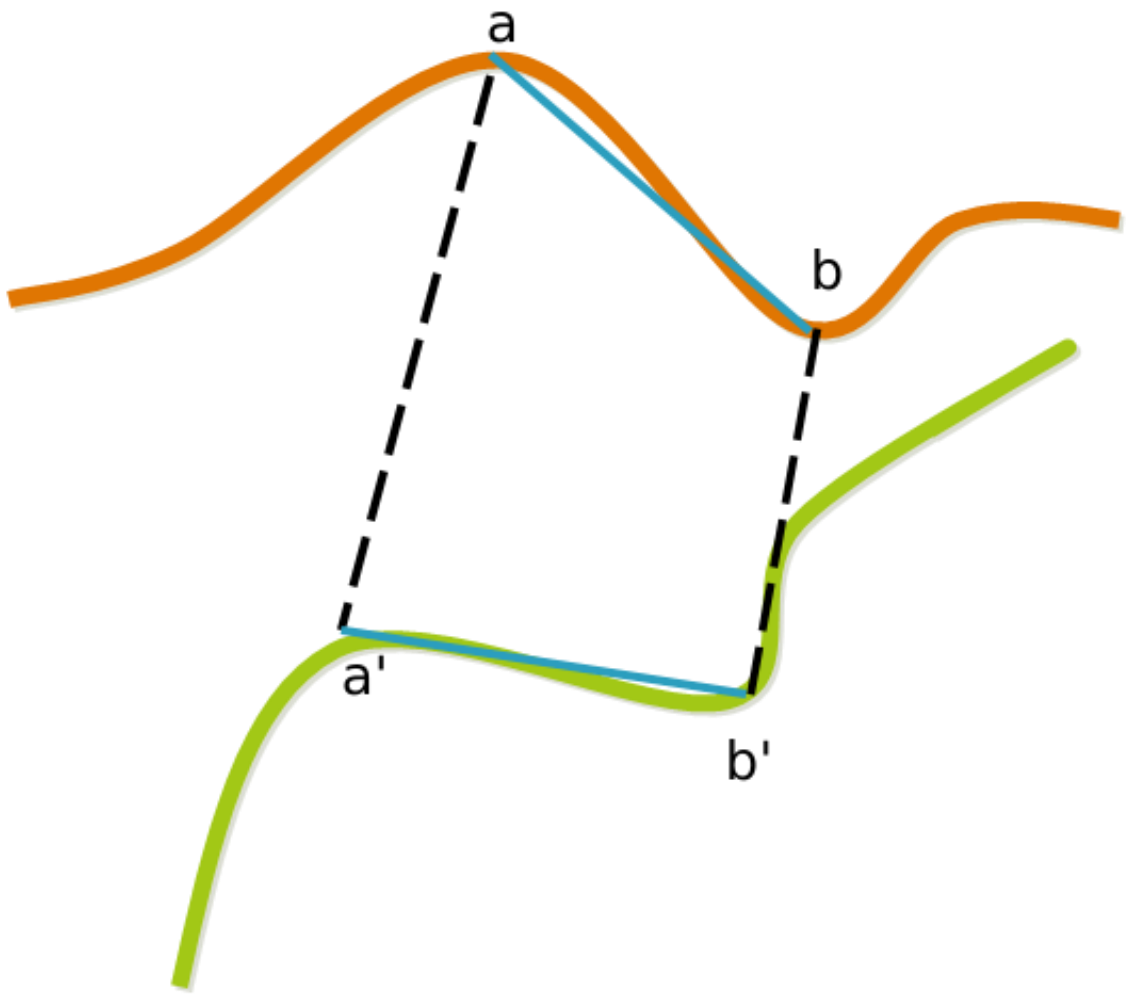
- What do we need to align M to S?
  - Compute a **transformation**
  - In mathematics a transformation is represented by a matrix

Alignment: an overview of the problem in 2D

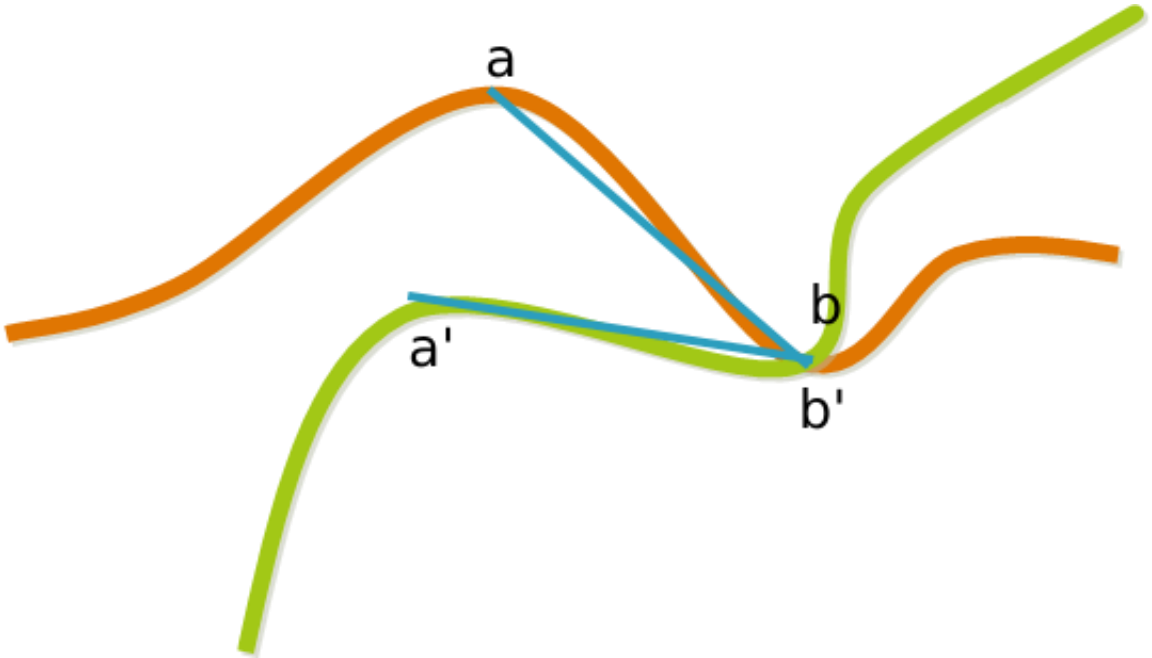
corresponding point set alignment



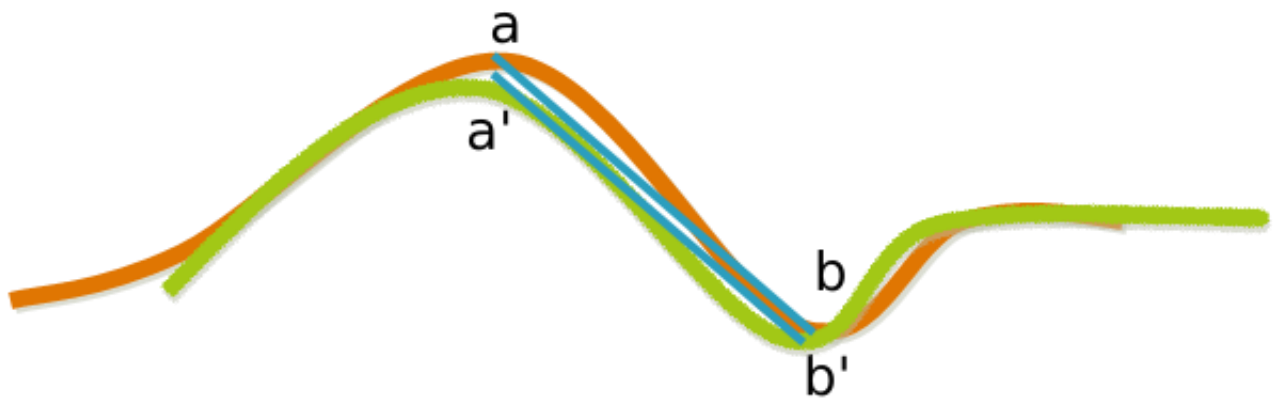
corresponding point set alignment



corresponding point set alignment

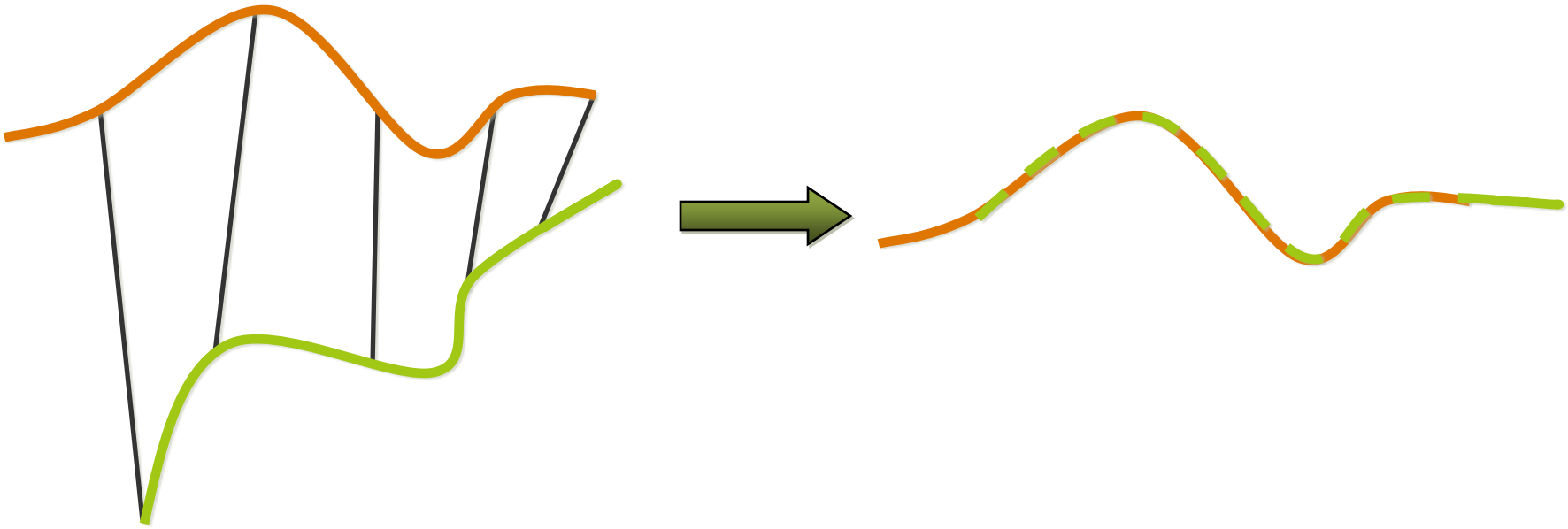


corresponding point set alignment



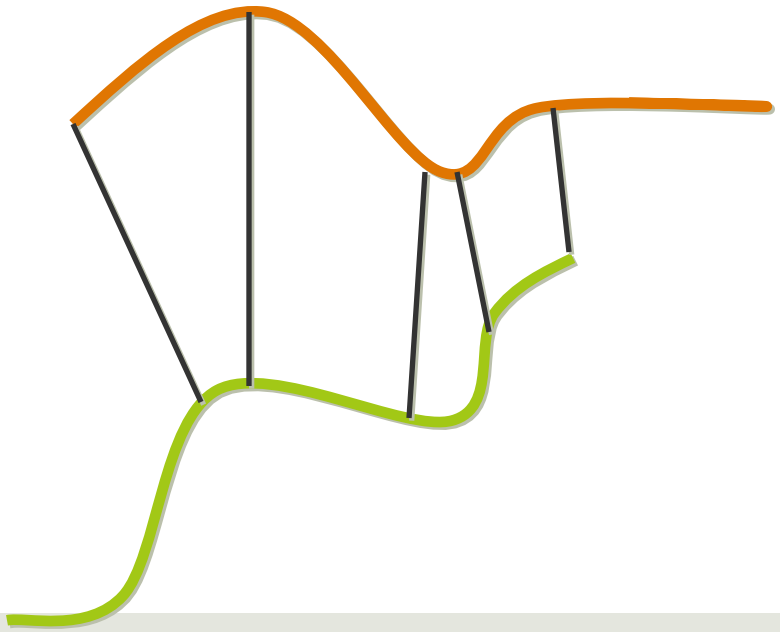
aligning 3D data

- if correct correspondences are known, can find correct relative rotation/translation



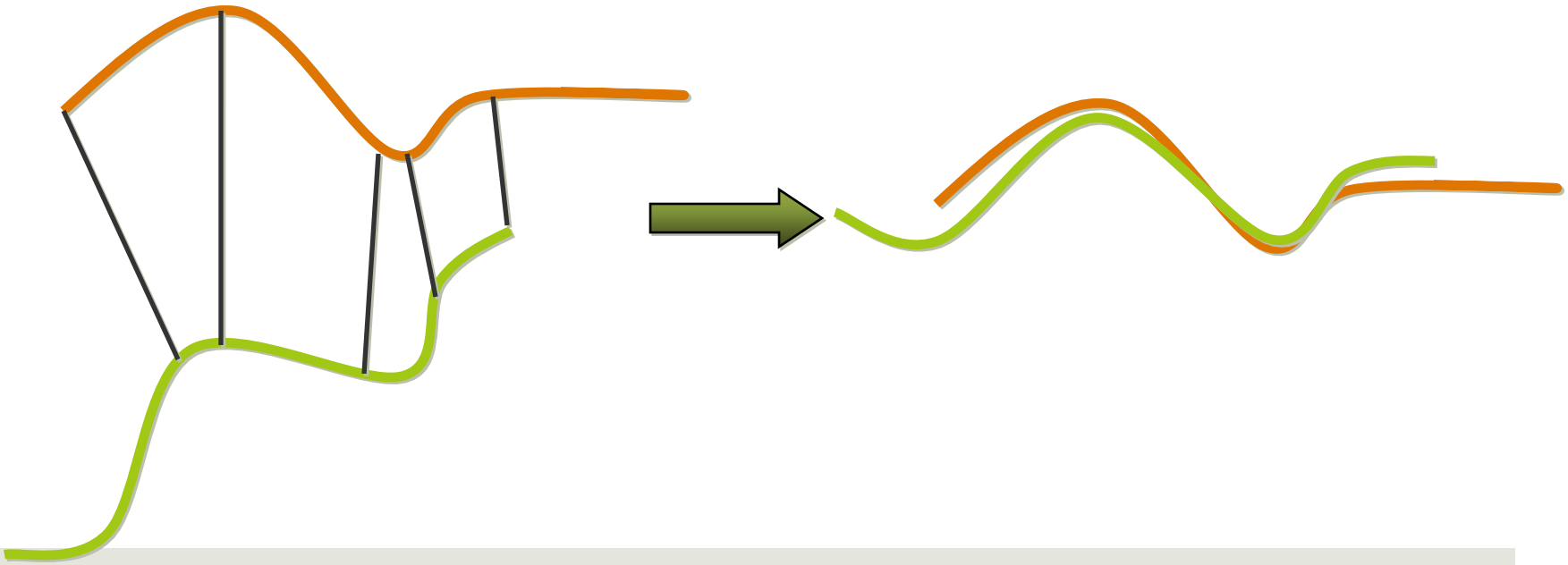
aligning 3D data

- how to find correspondences: user input? feature detection?
- alternative: assume **closest points** correspond



aligning 3D data

- how to find correspondences: user input? feature detection?
- alternative: assume **closest points** correspond



mean square error

- mse

$$MSE = \frac{1}{N_s}\sum_{i=1}^{N_s}\left\|\hat{Y} - Y\right\|^2$$

where $\hat{Y}$ is a prediction value

and Y the measured value

- imagine you are the weather forecast reporter and every day you predict the temperature for the next day
- mse will give you an average extimation on how much your predictions were wrong

corresponding point set alignment

- the mse objective function :

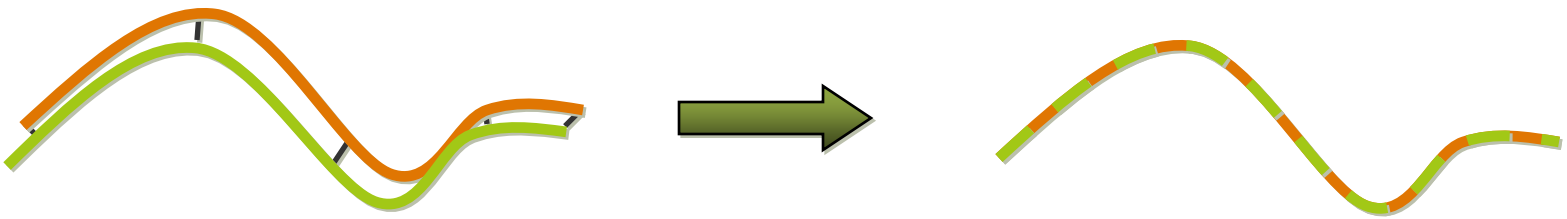
$$f(R,T) = \frac{1}{N_s}\sum_{i=1}^{N_s} \|m_i - rot(s_i) - trans\|^2$$

- a possible alignment is :

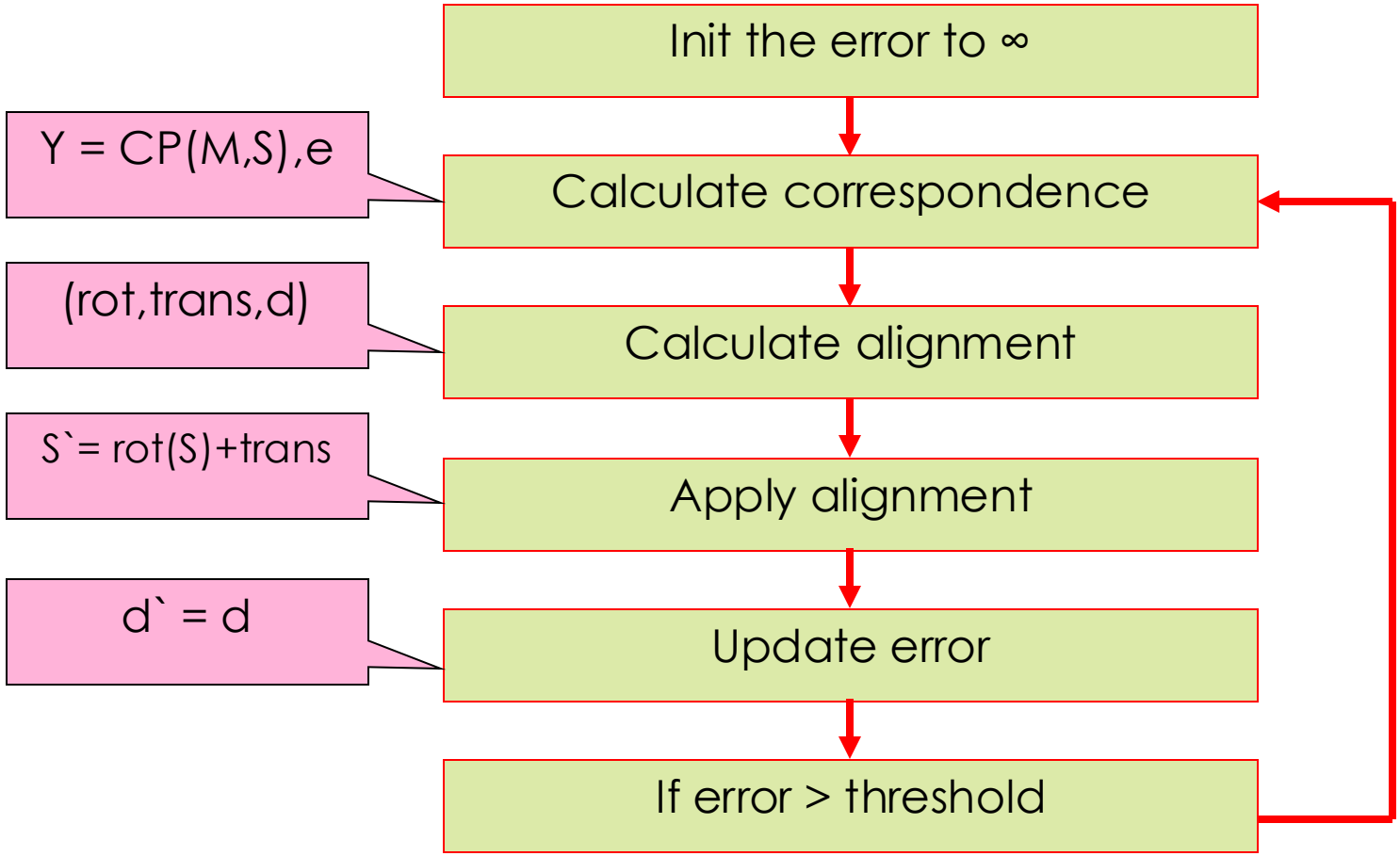
$$(rot, trans, d_{mse}) = \Phi(M, S)$$

aligning 3D data

- **Converges if starting position "close enough"**
  - **THIS IS REALLY IMPORTANT!!!!!!!!**
- …remember that we are talking about an iterative method…



The Algorithm



the algorithm

```
function ICP(Scene,Model)
begin
E` ← + ∞;
(Rot,Trans) ← In Initialize-Alignment(Scene,Model);
repeat
    E ← E`;
    Aligned-Scene ← Apply-Alignment(Scene,Rot,Trans);
    Pairs ← Return-Closest-Pairs(Aligned-Scene,Model);
    (Rot,Trans,E`) ← Update-Alignment(Scene,Model,Pairs,Rot,Trans);
Until |E`- E|  < Threshold
return (Rot,Trans);
end
```

ICP Variants

- Variants on the following stages of ICP have been proposed:

1. Selecting sample points (from one or both meshes)
2. Matching to points in the other mesh
3. Weighting the correspondences
4. Rejecting certain (outlier) point pairs

ICP Variants

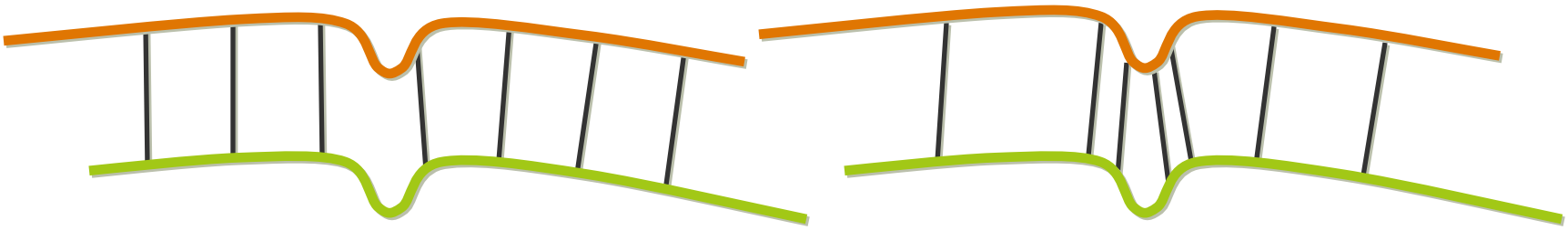
1. Selecting sample points (from one or both meshes).
2. Matching to points in the other mesh.
3. Weighting the correspondences.
4. Rejecting certain (outlier) point pairs.

Selection of points

- Use all available points [Besl 92].
- Uniform subsampling [Turk 94].
- Random sampling in each iteration

  [Masuda 96].
- Ensure that samples have normals distributed as uniformly as possible [Rusinkiewicz 01].

Selection of points



Uniform Sampling

Normal-Space Sampling

ICP Variants

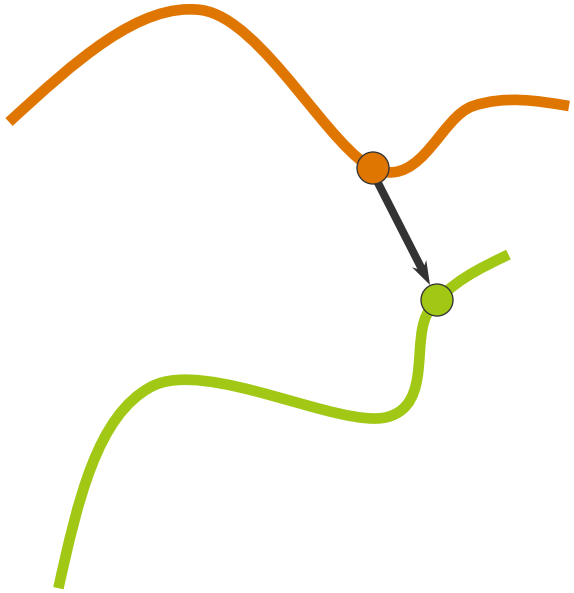
1. Selecting sample points (from one or both meshes).
2. Matching to points in the other mesh.
3. Weighting the correspondences.
4. Rejecting certain (outlier) point pairs.

Points matching

- Closest point in the other mesh [Besl 92].
- Normal shooting [Chen 91].
- Reverse calibration [Blais 95].
- Restricting matches to compatible points (color, intensity , normals , curvature ..) [Pulli 99].

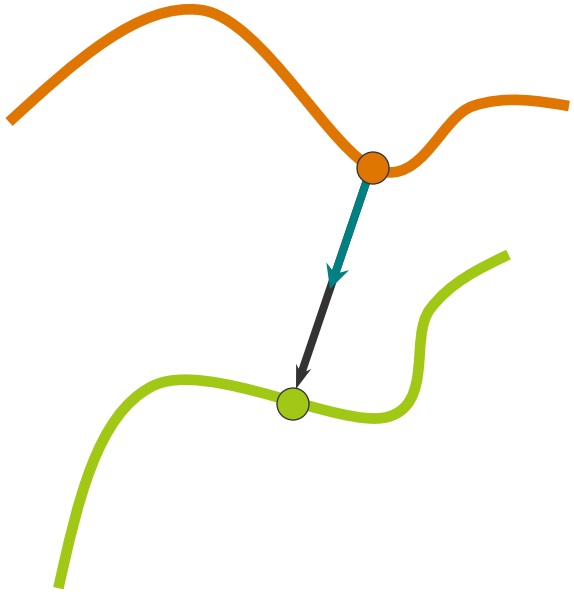
Points matching

- Closest point :



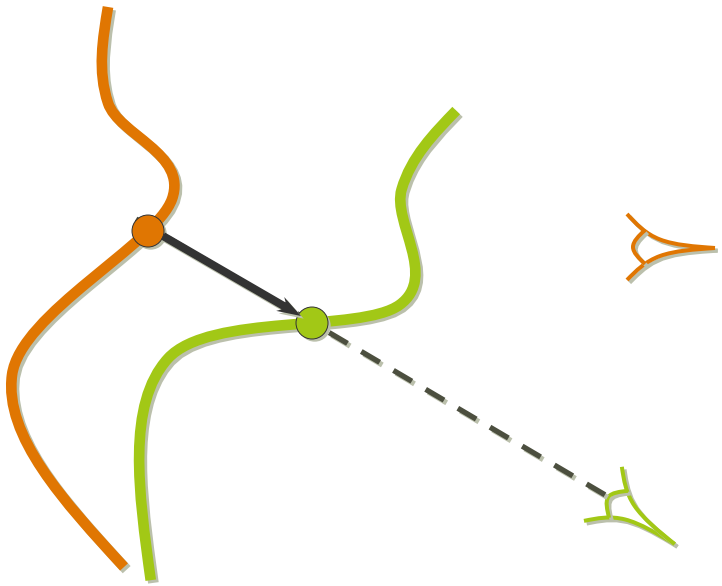
Points matching

- Normal Shooting



Points matching

- Projection (reverse calibration)
  Project the sample point onto the destination mesh , from the point of view of the destination mesh's camera.



ICP Variants

1. Selecting sample points (from one or both meshes).
2. Matching to points in the other mesh.
3. Weighting the correspondences.
4. Rejecting certain (outlier) point pairs.

Weighting of pairs

- Constant weight.
- Assigning lower weights to pairs with greater point-to-point distance :

$$Weight = 1 - \frac{Dist(p_1, p_2)}{Dist_{\max}}$$

- Weighting based on compatibility of normalized normals :

$$Weight = n_1 \bullet n_2$$

ICP Variants

1. Selecting sample points (from one or both meshes).
2. Matching to points in the other mesh.
3. Weighting the correspondences.
4. Rejecting certain (outlier) point pairs.

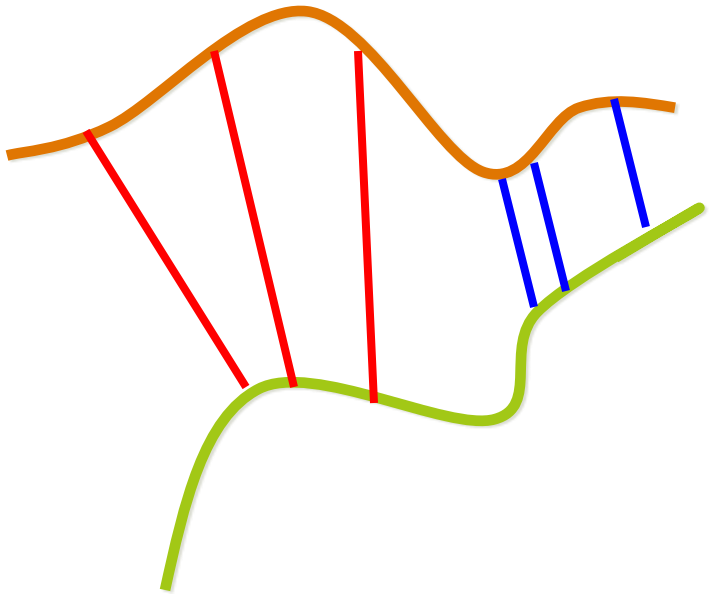
Rejecting Pairs

- Corresponding points with point to point distance higher than a given threshold.
- Rejection of worst n% pairs based on some metric.
- Pairs containing points on end vertices.
- Rejection of pairs that are not consistent with their neighboring pairs [Dorai 98] :
  (p_1,q_1) , (p_2,q_2) are inconsistent iff

$$\left|Dist(p_1,p_2)-Dist(q_1,q_2)\right| > threshold$$

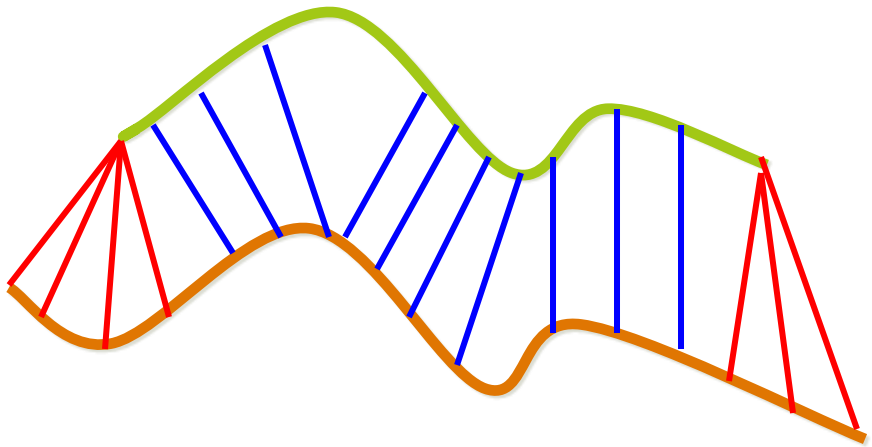
Rejecting Pairs

Distance thresholding



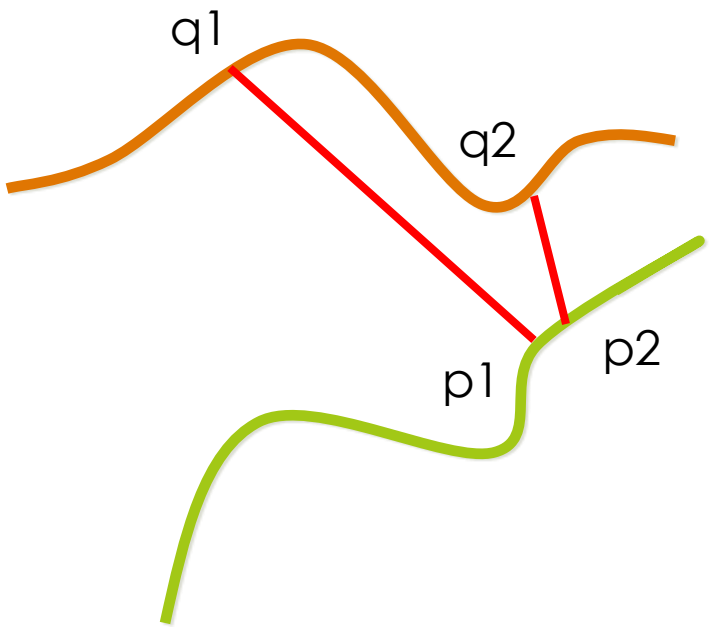
Rejecting Pairs

Points on end vertices



Rejecting Pairs

Inconsistent Pairs

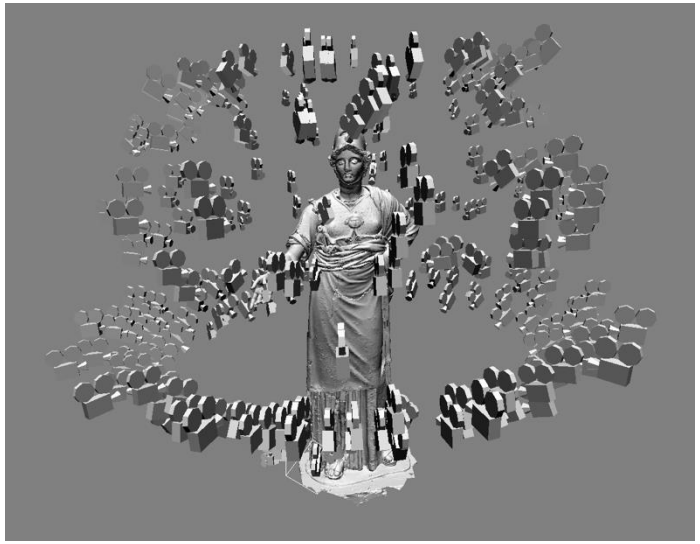


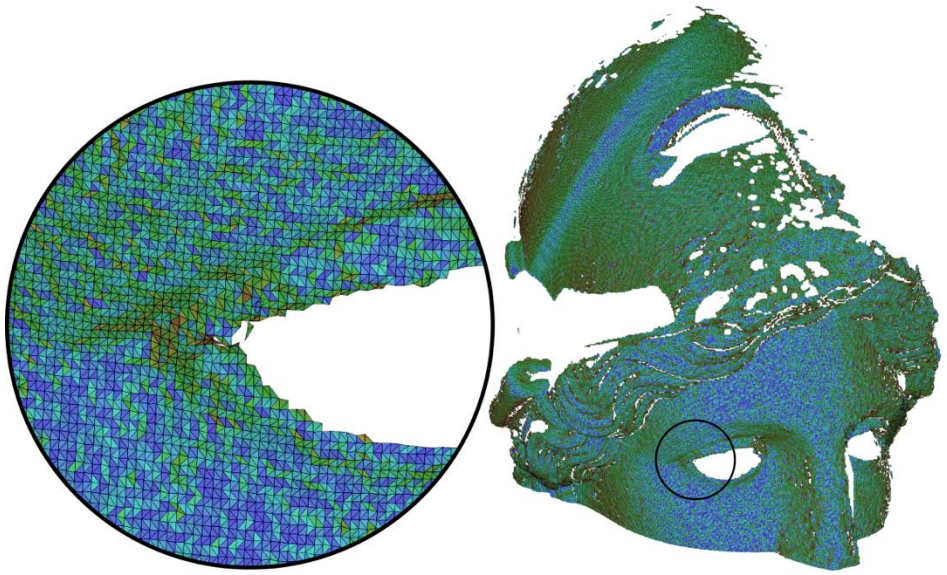
icp summary

- iterative closest point
- with a good initial solution I have the guaranteed to converge to a local minimum
  - ok….but which is a good initial solution?
  - how can I find one?
  - in MeshLab the user assists the alignment plugin providing an initial solution to the ICP algoritm!
    - semi automatic approach
      - pro: reliable
      - cons: **time consuming…the time of a man is precious**

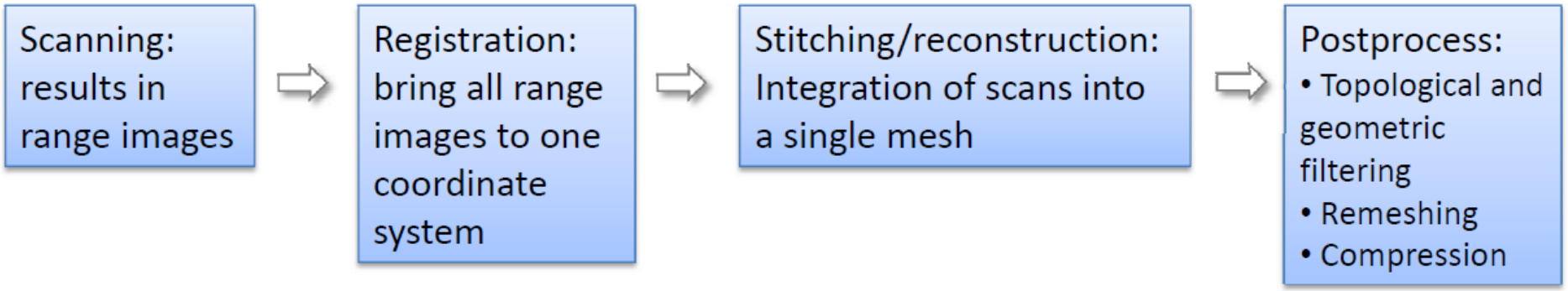
Two Notes on 3D scanning

- almost all scanning technologies will internally store the same type of data
  - **range map**: 2D map of the sampled 3D points
- **at the end what a scanner will output are n^{th} set of 3d coordinates expressend on the camera coordinates system**
- **our goal:** a final 3d model





fast overview on scanning pipeline



acquisition

Scanning: results in range images

Registration: bring all range images to one coordinate system

Stitching/reconstruction: Integration of scans into a single mesh

Postprocess:
- Topological and geometric filtering
- Remeshing
- Compression

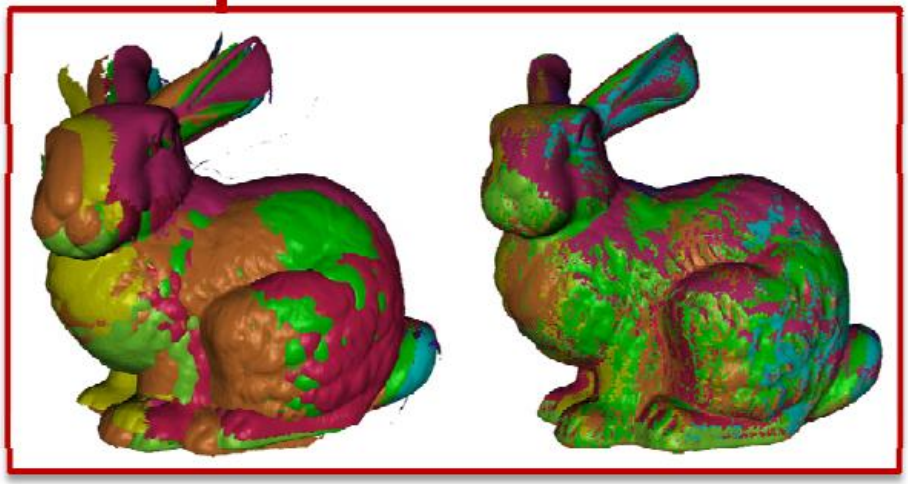
alignment

Scanning: results in range images

Registration: bring all range images to one coordinate system

Stitching/reconstruction: Integration of scans into a single mesh

Postprocess:
- Topological and geometric filtering
- Remeshing
- Compression

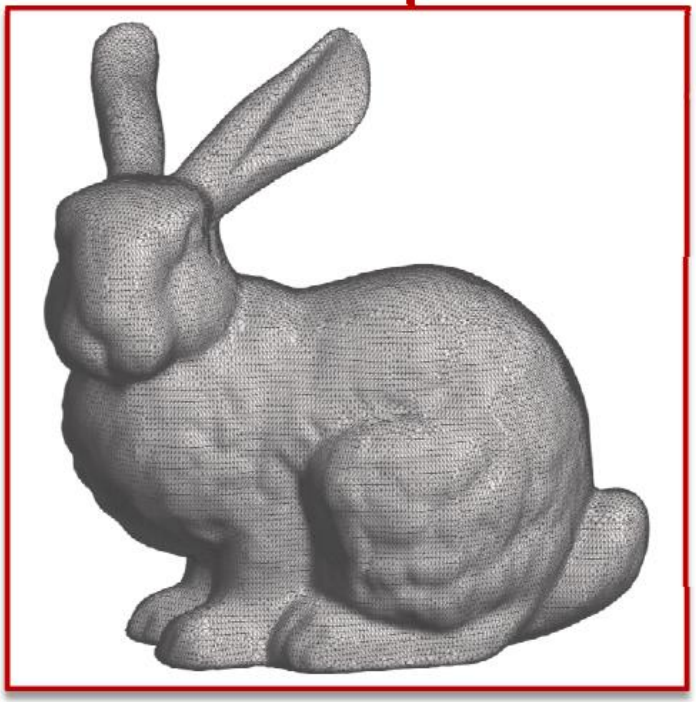


merging

Scanning: results in range images

Registration: bring all range images to one coordinate system

Stitching/reconstruction: Integration of scans into a single mesh

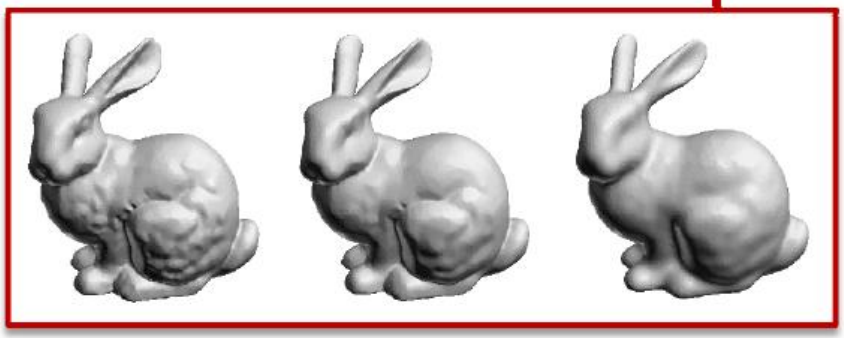
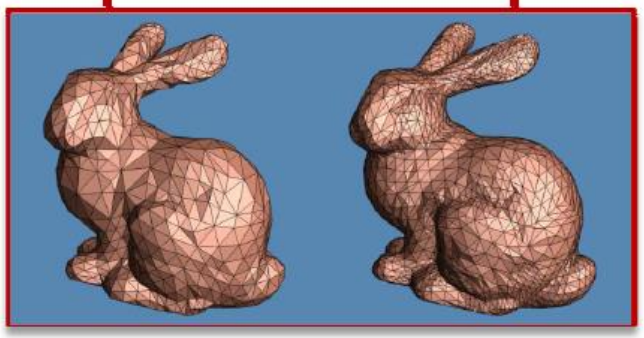


Postprocess:
- Topological and geometric filtering
- Remeshing
- Compression

post processing

Scanning: results in range images

Registration: bring all range images to one coordinate system

Stitching/reconstruction: Integration of scans into a single mesh

Postprocess:
- Topological and geometric filtering
- Remeshing
- Compression