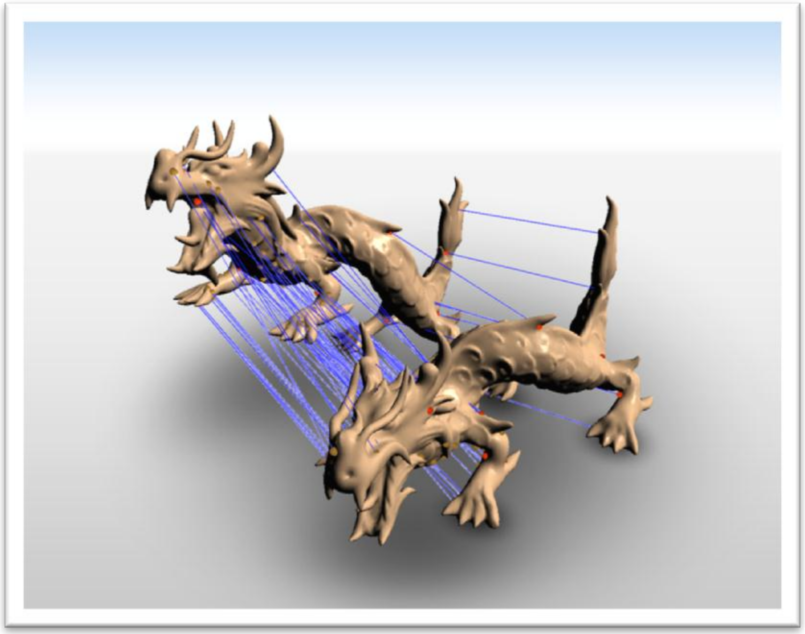
Statistical Geometry Processing

Winter Semester 2011/2012

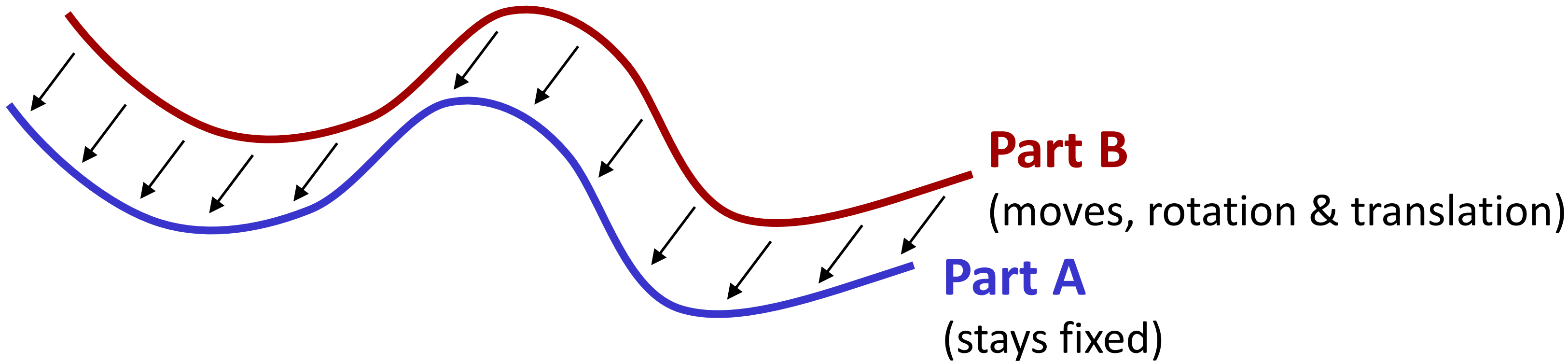


Global Shape Matching

UNIVERSITÄT DES SAARLANDES
CLUSTER OF EXCELLENCE
max planck institut informatik

Rigid Global Matching

Iterated Closest Points (ICP)



Problems

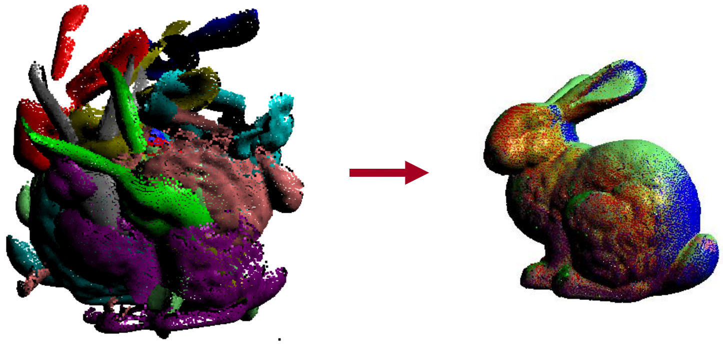
- Need good intialization
  - Non-convex problem
  - Runs into local minima
- Deformable shape matching
  - Even worse: bad initialization even more problematic
  - Reason: more degrees of freedom

Global Matching

How to assemble the bunny (globally)?

Pipeline (rough sketch):

- Feature detection
- Feature descriptors
- Spectral validation



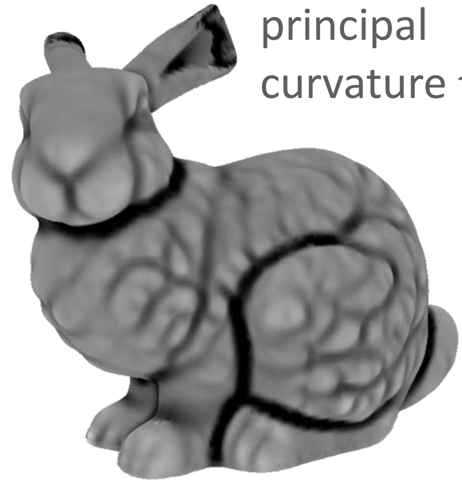
Feature Detection

Feature points (keypoints)

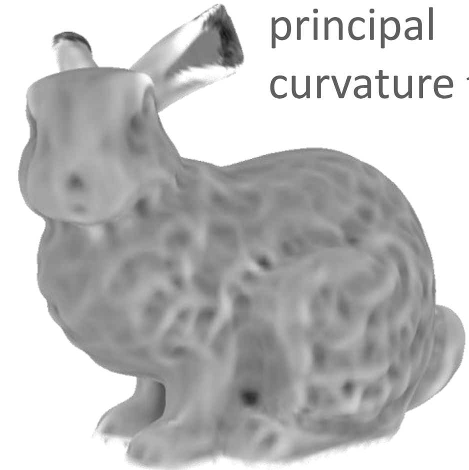
- Regions that can be identified locally
- "Bumps", i.e. points with maximum curvature
  - "curvature" ∈ $\left\{\kappa_1, \kappa_2, \frac{1}{2}(\kappa_1 + \kappa_2), \kappa_1 \cdot \kappa_2\right\}$
  - Mean/principal curvature most stable
    (κ_2 often inaccurate when computed by least-squares fitting)
  - "SIFT" features – compute bumps at multiple scales:
    - With with different radii
    - Search for maxima in 3D surface-scale space
  - Output: list of keypoints

Bunny Curvature

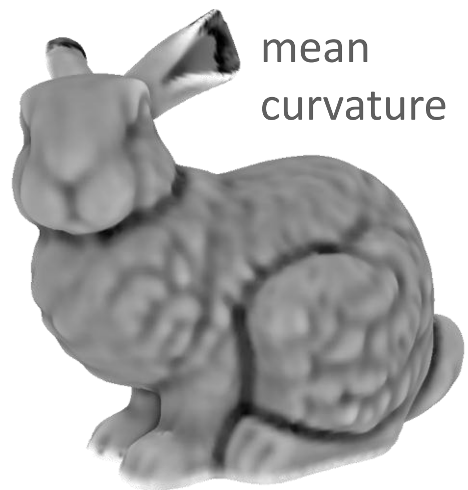
Stanford Bunny
(dense point cloud)



principal curvature κ_1



principal curvature κ_2



mean curvature

Gaussian curvature

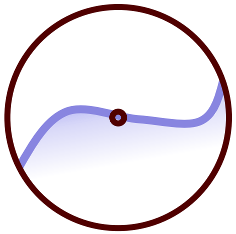
[courtesy of Martin Bokeloh]

Descriptors

Feature descriptors:

- Rotation invariant description of local neighborhood (within scale of the feature point)
  - Translation already fixed by feature point
- Used to find match candidates
- Not 100% reliable (typically 3x – 5x outlier ratio)

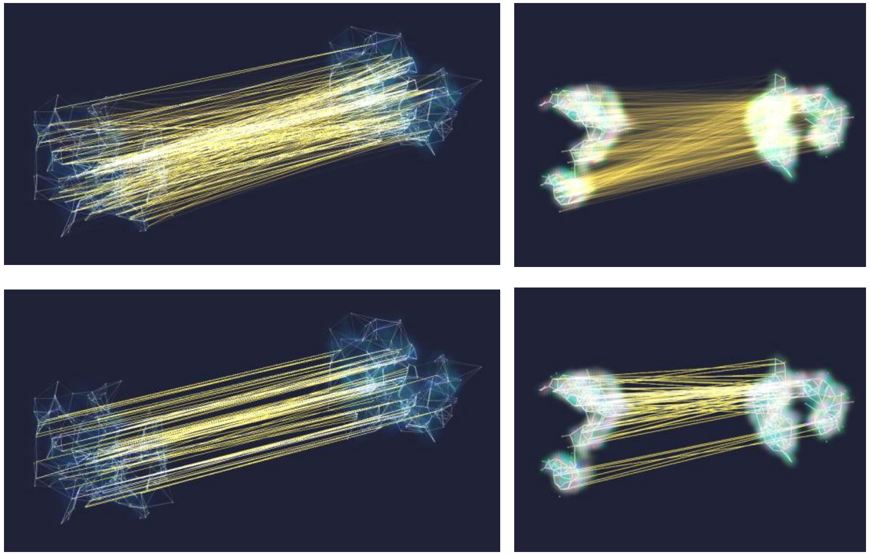
Descriptors



Rotation invariant descriptors:

- Curvatures $\{\kappa_1, \kappa_2\}$, derived properties
  - Curvature histograms in spherical neighborhood
- Pairwise distances
  - “d2-Histograms”: Histogram of pairwise distance within sphere
  - Histogram of distances to medial axis
- Spin images
  - Use surface normal
  - Cut-out sphere
  - Rotate geometry around sphere and splat into “spin-image”
- Spherical harmonics power spectrum, Zernicke descriptors

Correspondence Validation



We have:

- Candidate matches
- But every keypoint matches 5 others on average
- At most one of these is correct

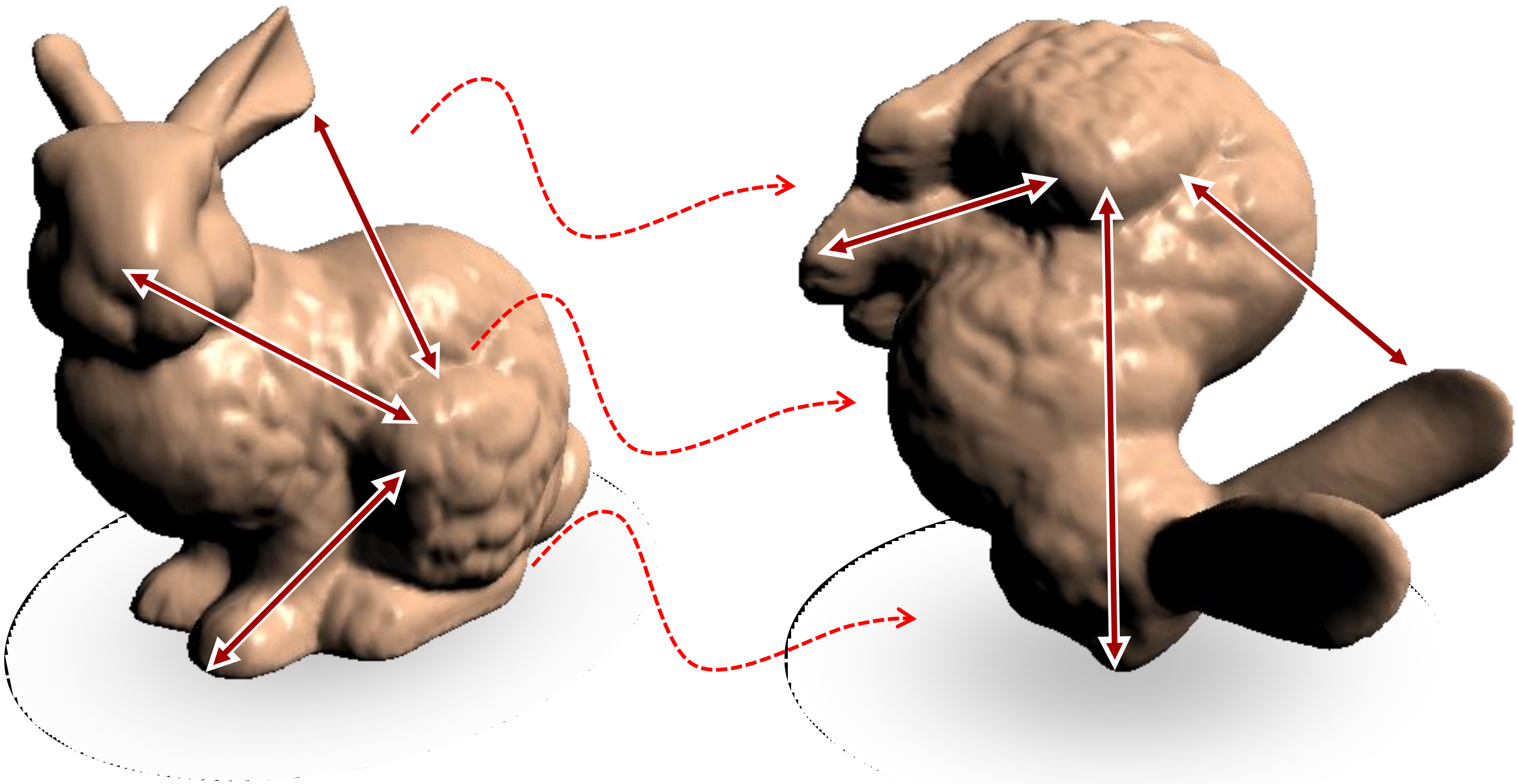
Validation Criterion:

- Euclidian distance should be preserved

Invariants

Rigid Matching

- Invariant: Euclidean distances are preserved



Branch and Bound

Simple Algorithm:

- Branch-and-bound [Gelfand et al. 2005]
- Fix correspondences, prune all incompatible ones (i.e., violation of Euclidian distance)
- Try all possibilities

Efficiency:

- Efficient for sparse (widely spaced) features
  - Only few combinations work
- Possibly exponential for dense features (try many equivalent solutions)

Alternatives

Alternatives: We will look at

- Spectral matching
- Randomized search

Further alternatives:

- Loopy belief propagation
  (“Correlated Correspondences”, Anguelov 2005).
- Quadratic assignment heuristics

Important:

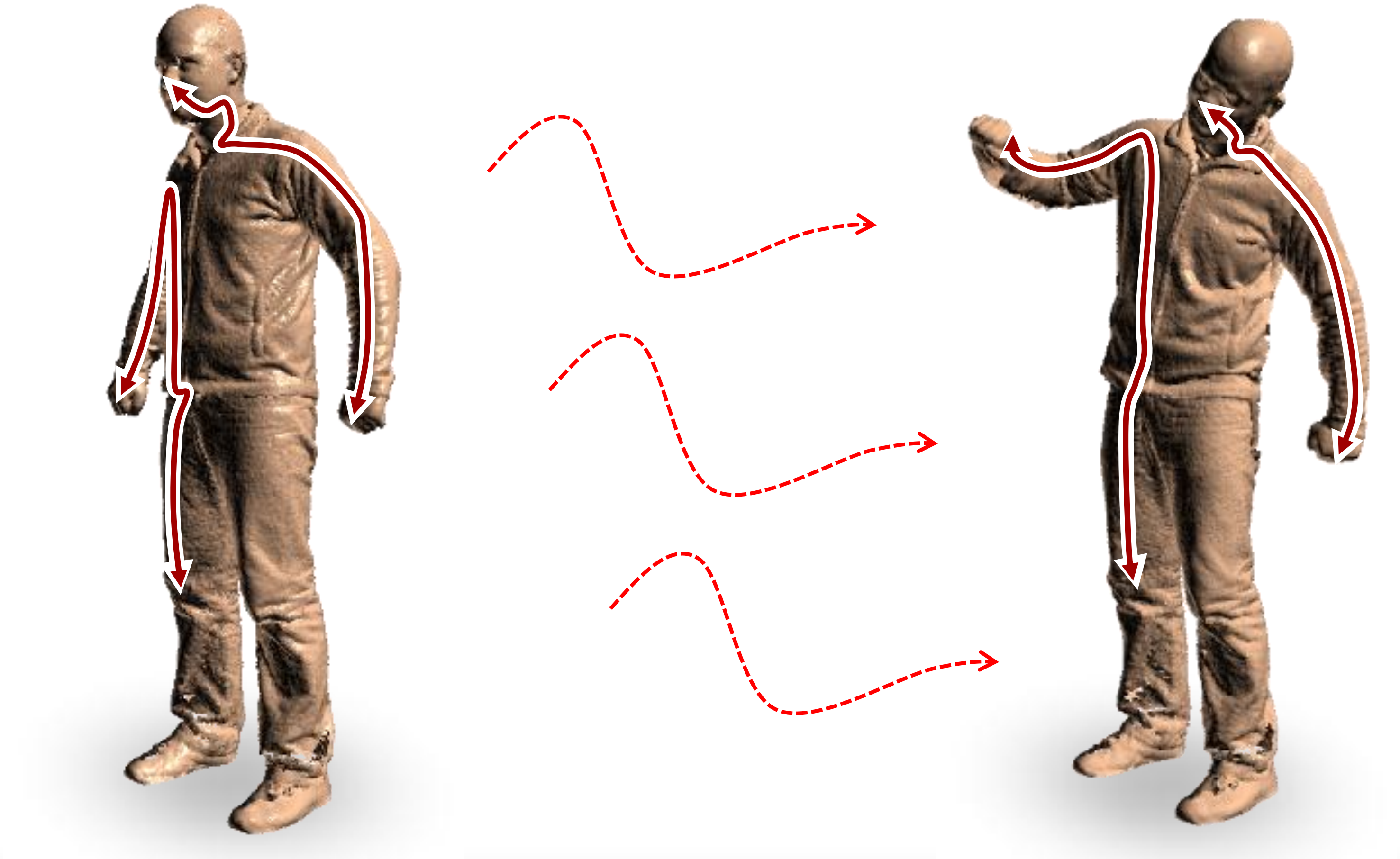
- Structure: Pairwise optimization problem

Isometric Matching

Invariants

Intrinsisc Matching

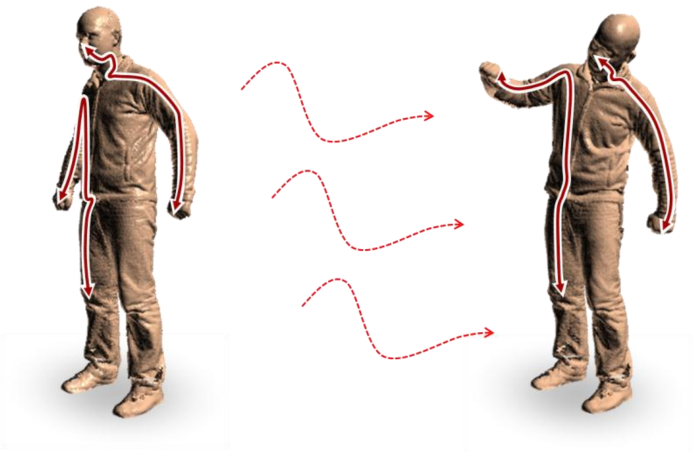
- Invariants: All geodesic distances are preserved



Invariants

Intrinsisc Matching

- Presevation of geodesic distances („intrinsic distances“)
- Approximation
  - Cloth is almost unstretchable
  - Skin does not stretch a lot
  - Most live objects show approximately isometric surfaces
- Accepted model for deformable shape matching
  - In cases where one subject is presented in different poses
  - Accross different subjects: Other assumptions necessary
  - Then: global matching is an open problem



Feature Based Matching

Quadratic Assignment Model

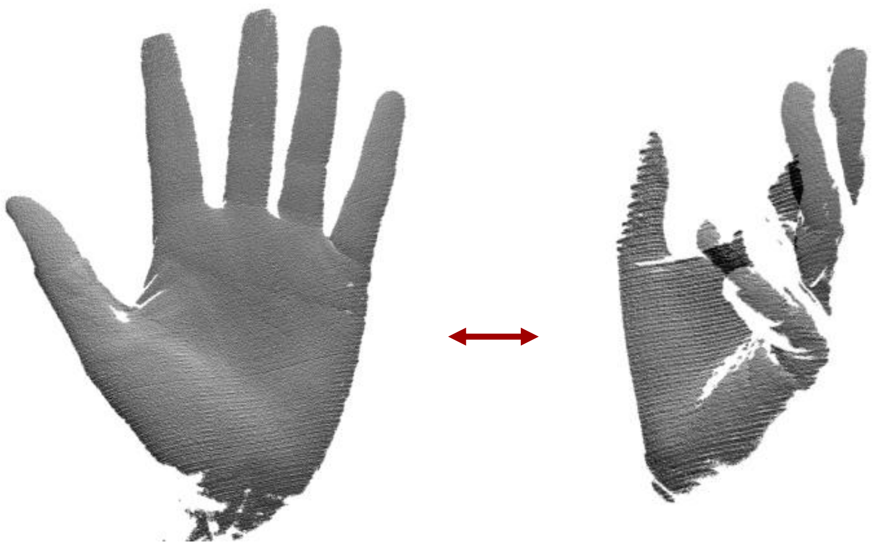
Problem Statement

Deformable Matching

- Two shapes: original, deformed
- How to establish correspondences?
- Looking for global optimum
  - Arbitrary pose

Assumption

- Approximately isometric deformation

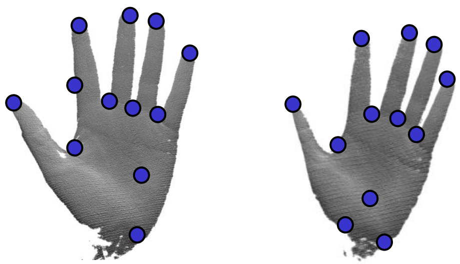


[data set: S. König, TU Dresden]

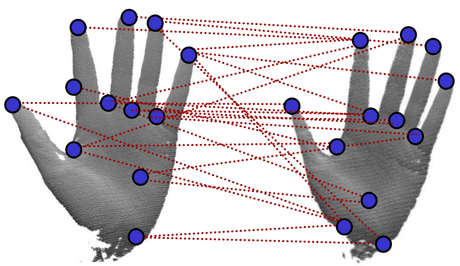
Algorithm

Feature-Matching

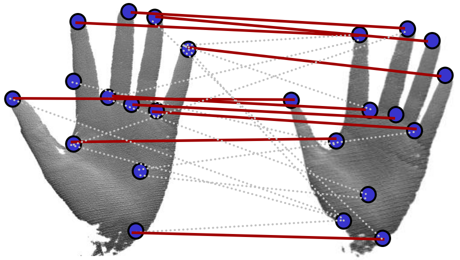
- Detect feature points



- Local matching: potential correspondences



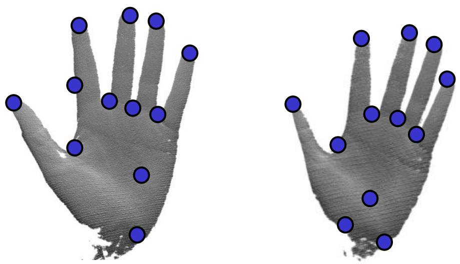
- Global filtering: correct subset



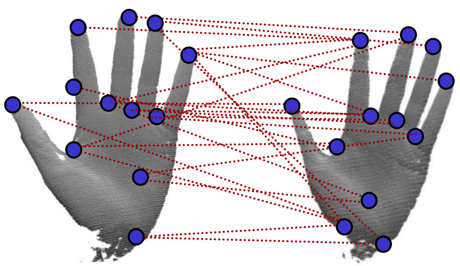
Algorithm

Feature-Matching

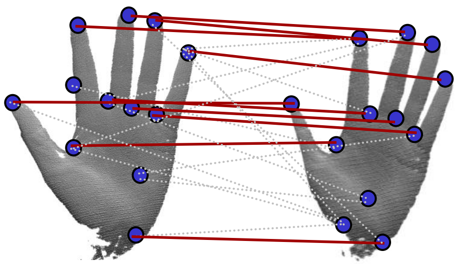
- Detect feature points
  - Maxima of Gaussian curvature
  - Locally unique descriptors



- Local matching: potential correspondences



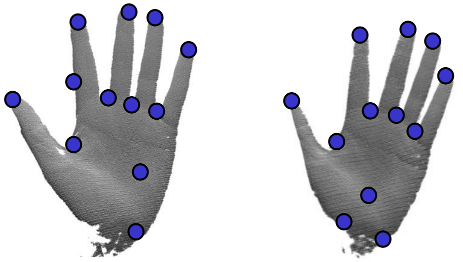
- Global filtering: correct subset



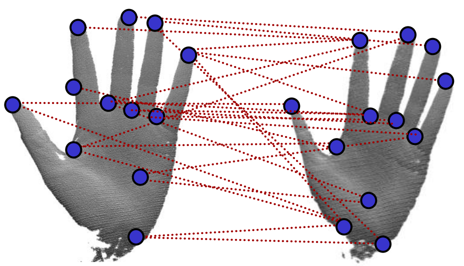
Algorithm

Feature-Matching

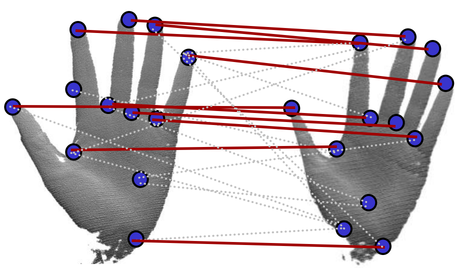
- Detect feature points
  - Maxima of Gaussian curvature
  - Locally unique descriptors



- Local matching: potential correspondences
  - Curvature histograms
  - Heat-kernels, geodesic waves



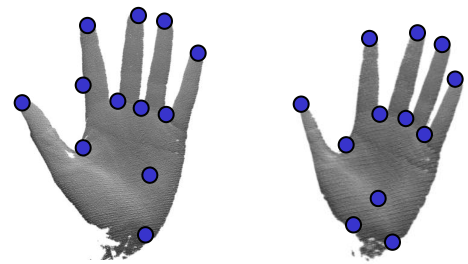
- Global filtering: correct subset



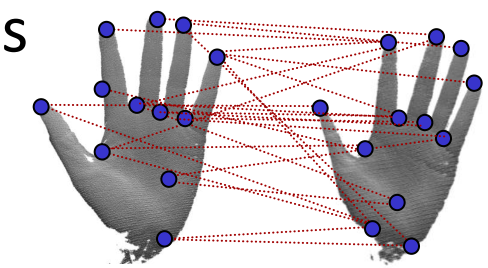
Algorithm

Feature-Matching

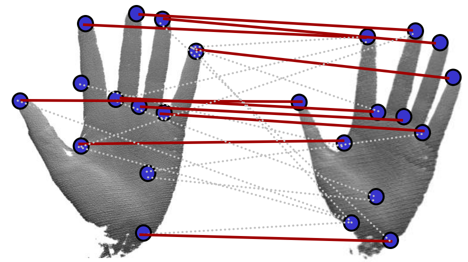
- Detect feature points
  - Maxima of Gaussian curvature
  - Locally unique descriptors



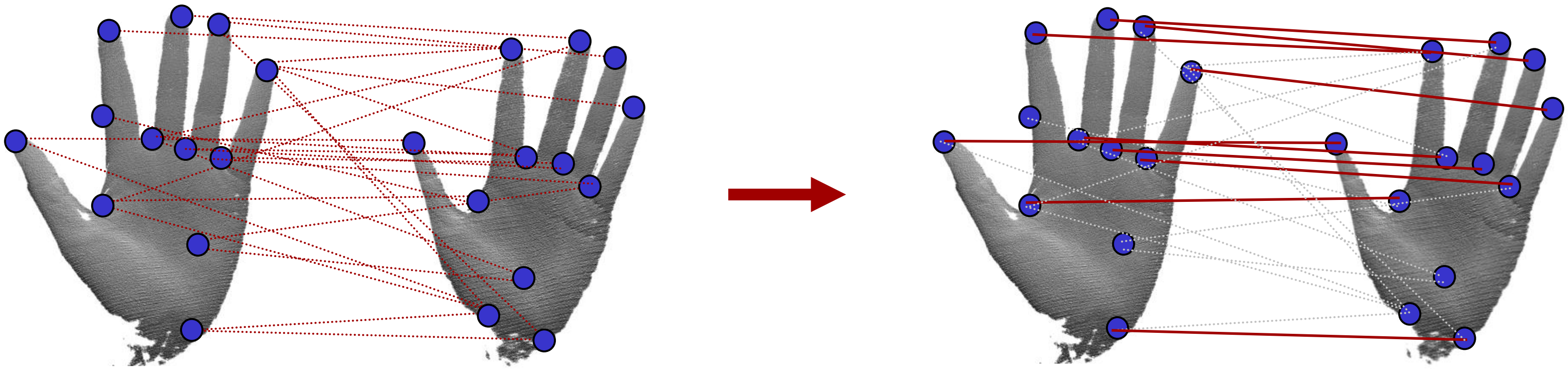
- Local matching: potential correspondences
  - Curvature histograms
  - Heat-kernels, geodesic waves



- Global filtering: correct subset
  - Quadratic assignment
  - Spectral relaxation [Leordeanu et al. 05]
  - RANSAC



Quadratic Assignment



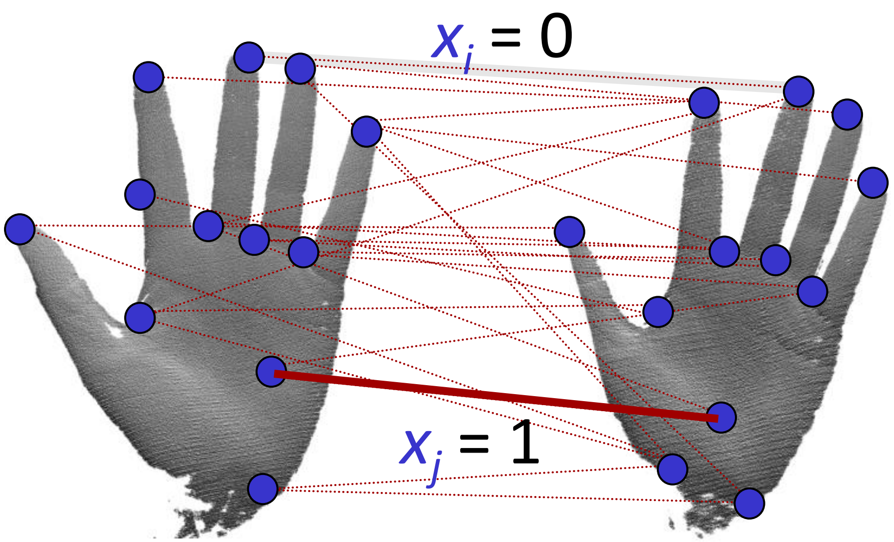
Most difficult part: Global filtering

- Find a consistent subset
- Pairwise consistency:
  - Correspondence pair must preserve intrinsic distance
- Maximize number of pairwise consistent pairs
  - Quadratic assignment (in general: NP-hard)

Quadratic Assignment Model

Quadratic Assignment

- *n* potential correspondences
- Each one can be turned on or off
- Label with variables x_i
- Compatibility score:



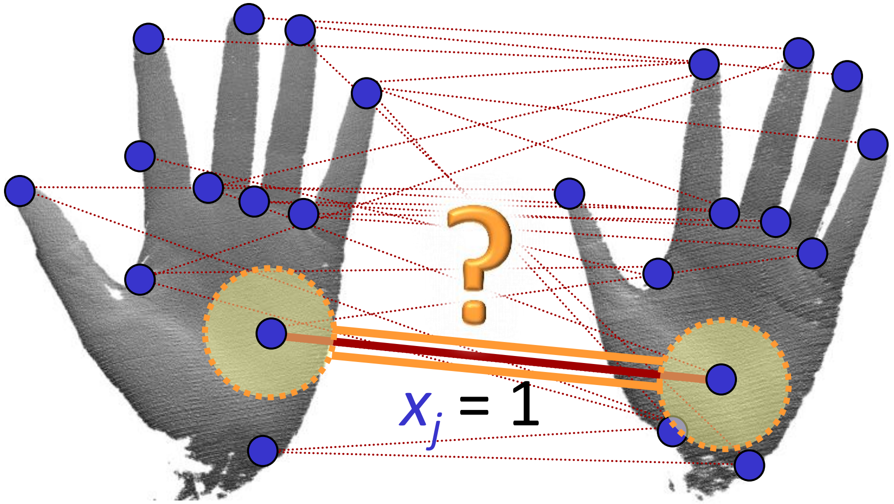
$$P^{(match)}(x_1,\ldots,x_n) = \prod_{i=1}^{n} P_i^{(single)} \prod_{i,j=1}^{n} P_{i,j}^{(compatible)}, x_i \in \{0,1\}$$

(incomplete model; details later)

Quadratic Assignment Model

Quadratic Assignment

- Compatibility score:
  - **Singeltons:** Descriptor match

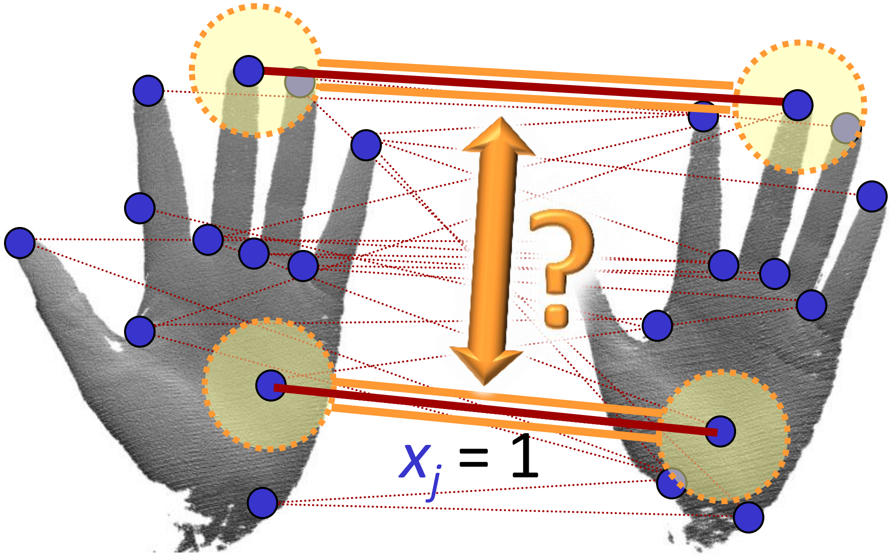


$$P^{(match)}(x_1,...,x_n) = \prod_{i=1}^{n} P_i^{(single)} \prod_{i,j=1}^{n} P_{i,j}^{(compatible)}, x_i \in \{0,1\}$$

Quadratic Assignment Model

Quadratic Assignment

- Compatibility score:
  - **Singeltons:** Descriptor match
  - **Doubles:** Compatibility



$$P^{(match)}(x_1,\ldots,x_n) = \prod_{i=1}^{n} P_i^{(single)} \prod_{i,j=1}^{n} P_{i,j}^{(compatible)}, x_i \in \{0,1\}$$

Quadratic Assignment Model

Quadratic Assignment

- Matrix notation:

$$P^{(match)}(x_1,\ldots,x_n) = \prod_{i=1}^{n} P_i^{(single)} \prod_{i,j=1}^{n} P_{i,j}^{(compatible)}$$

$$\log P^{(match)}(x_1,\ldots,x_n) = \sum_{i=1}^{n} \log P_i^{(single)} + \sum_{i,j=1}^{n} \log P_{i,j}^{(compatible)}$$

$$= \quad \mathbf{xs} \quad + \quad \mathbf{x}^T \mathbf{D} \mathbf{x}$$

- Quadratic scores are encoded in Matrix **D**
- Linear scores are encoded in Vector **s**

Quadratic Assignment Model

Quadratic Assignment

- Task: find optimal binary vector **x**

Regularization:

- No trivial solution x = 0

Examples

- As many „1“s as possible without exceeding error threshold
- Fixed norm of **x**-vector

Spectral Matching

Simple & Effective Approximation:

- Spectral matching [Leordeanu & Hebert 05]
- Form compatibility matrix:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{21} & a_{31} & \\ a_{12} & a_{22} & & \\ a_{13} & & \ddots & \\ & & & \ddots \end{pmatrix}$$

Diagonal:
Descriptor match

Off-Diagonal:
Pairwise compatibility

All entries within [0..1]
= [no match...perfect match]

Spectral Matching

Approximate largest clique:

- Compute eigenvector with largest eigenvalue
- Maximizes Rayleigh quotient:

$$\arg\max \frac{\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}}{\|\mathbf{x}\|^2}$$

- “Best yield” for bounded norm
  - The more consistent pairs (rows of 1s), the better
  - Approximates largest clique
- Implementation
  - For example: power iteration

Spectral Matching

Post-processing

- Greedy quantization
  - Select largest remaining entry, set it to 1
  - Set all entries to 0 that are not pairwise consistent with current set
  - Iterate until all entries are quantized

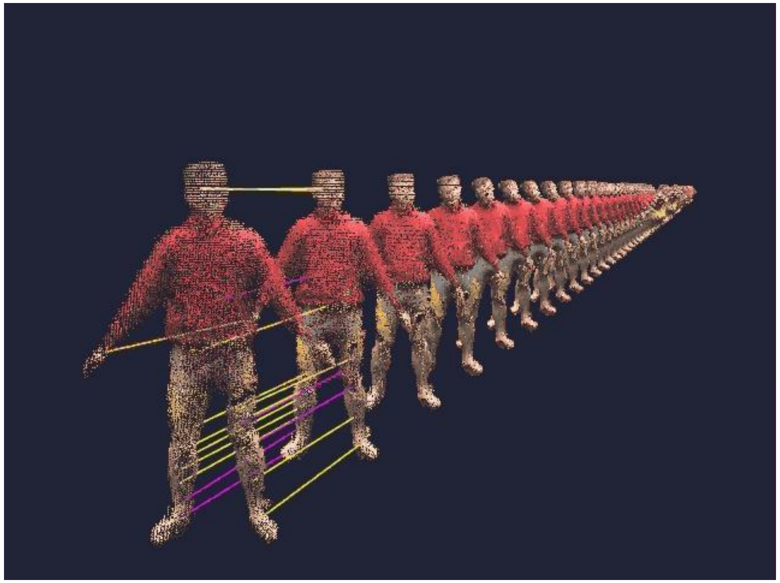
In practice...

- This algorithm turns out to work quite well.
- Very easy to implement
- Limited to (approx.) quadratic assignment model

Spectral Matching Example

Application to Animations

- **Feature points:**
  Geometric MLS-SIFT features [Li et al. 2005]
- **Descriptors:**
  Curvature & color ring histograms
- **Global Filtering:**
  Spectral matching
- **Pairwise animation matching:**
  Low precision passive stereo data



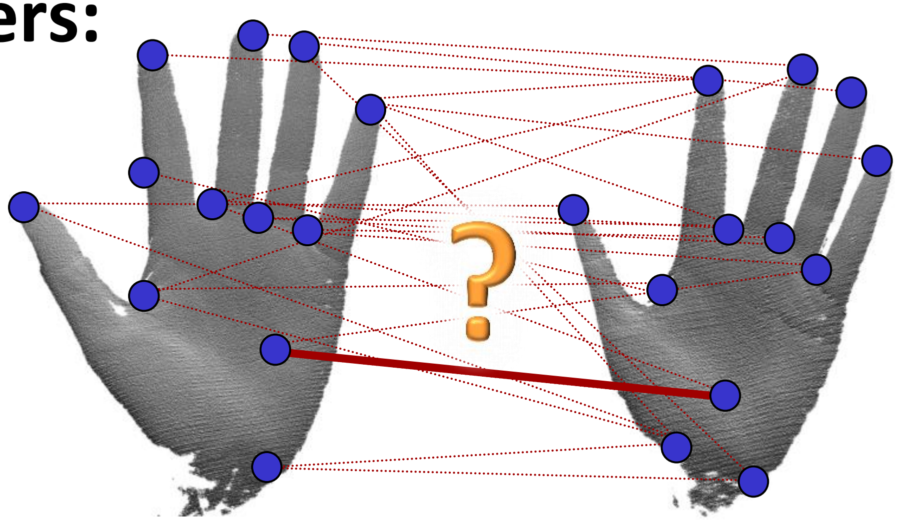
[Data set: Christian Theobald, Implementation: Martin Bokeloh]

Ransac and Forward Search

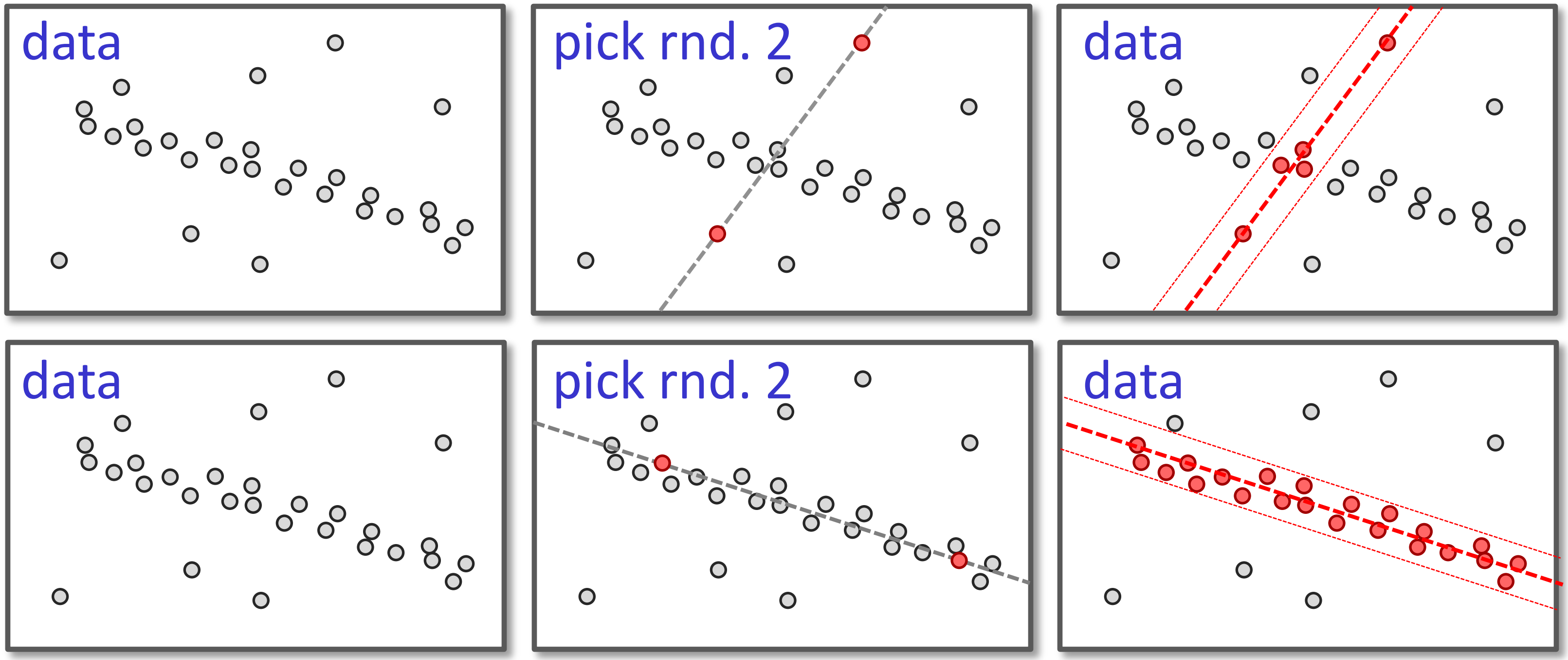
Random Sampling Algorithms

Estimation subject to outliers:

- We have candidate correspondences
- But most of them are bad
- Standard vision problem
- Standard tools:
  Ransac & forward search



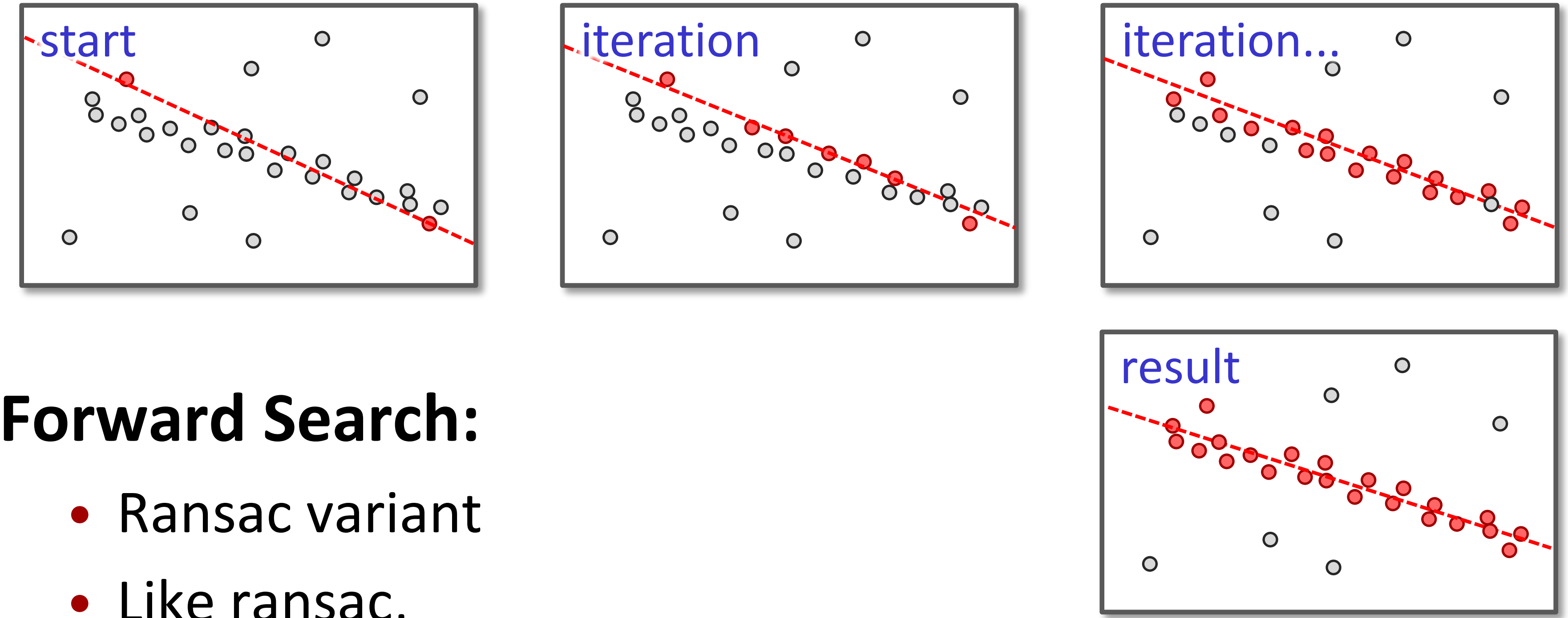
RANSAC



„Standard“ RANSAC line fitting example:

- Randomly pick two points
- Verify how many others fit
- Repeat many times and pick the best one (most matches)

Forward Search



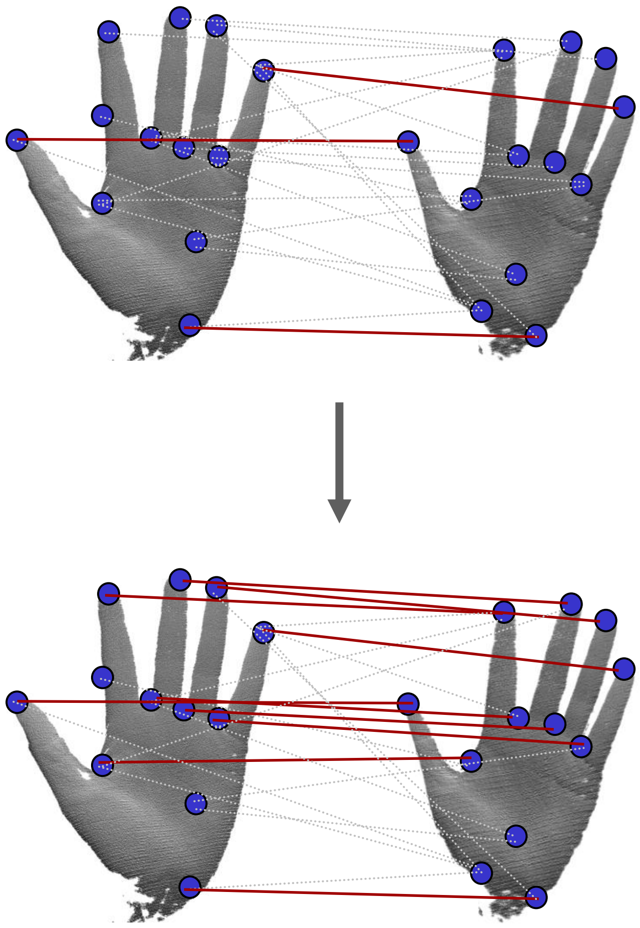
Forward Search:

- Ransac variant
- Like ransac,
  but refine model by „growing“
- Pick best match, then recalculate
- Repeat until threshold is reached

RANSAC/FWS Algorithm

Idea

- Starting correspondence
- Add more that are consistent
  - Preserve intrinsic distances
- Importance sampling algorithm



Advantages

- Efficient (small initial set)
- General (arbitrary criteria)

Ransac/FWS Details

Algorithm: Simple Idea

- Select correspondences with probability proportional to their plausibility
- First correspondence: Descriptors
- Second: Preserve distance (distribution peaks)
- Third: Preserve distance (even fewer choices)

  ...
- Rapidly becomes deterministic
- Repeat multiple times (typ.: 100x)
  - Choose the largest solution (larges #correspondences)

Ransac/FWS Details

Provably Efficient:

- Theoretically efficient (details later)
- Faster in practice (using descriptors)

Flexible:

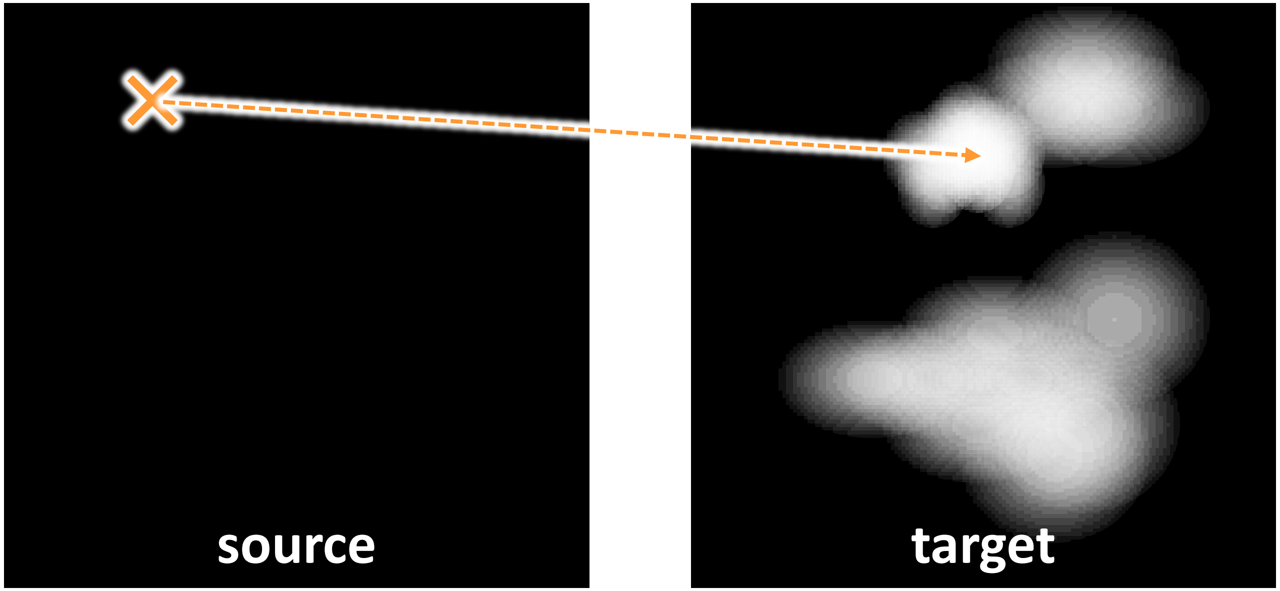
- In later iterations (> 3 correspondences), allow for outlier geodesics
- Can handle topological noise

Foreward Search Algorithm

Forward Search

- Add correspondences incrementally
- Compute match probabilities given the information already decided on
- Iterate until no more matches can found that meet a certain error threshold
- Outer Loop:
  - Iterate the algorithm with random choices
  - Pick the best (i.e., largest) solution

Foreward Search Algorithm

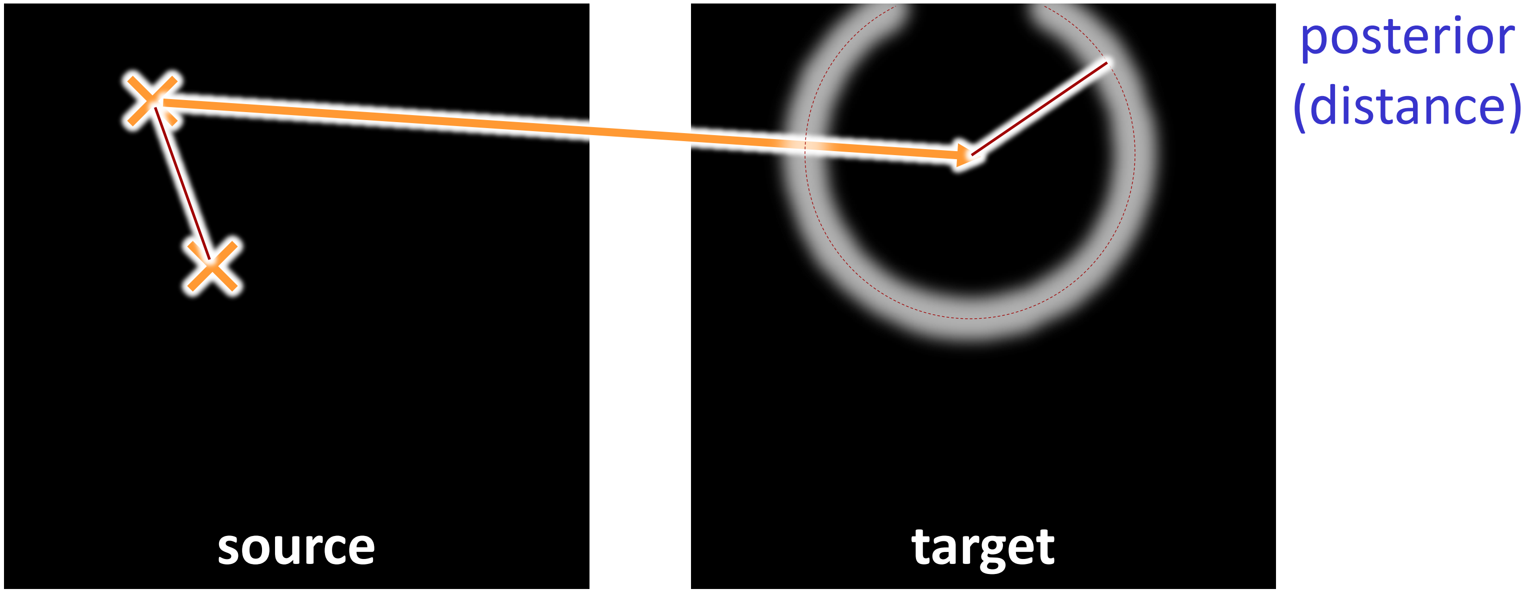


Descriptor matching scores

Step 1:

- Start with one correspondence
  - Target side importance sampling: prefer good descriptor matches
  - Optional source side imp. sampl: prefer unique descriptors

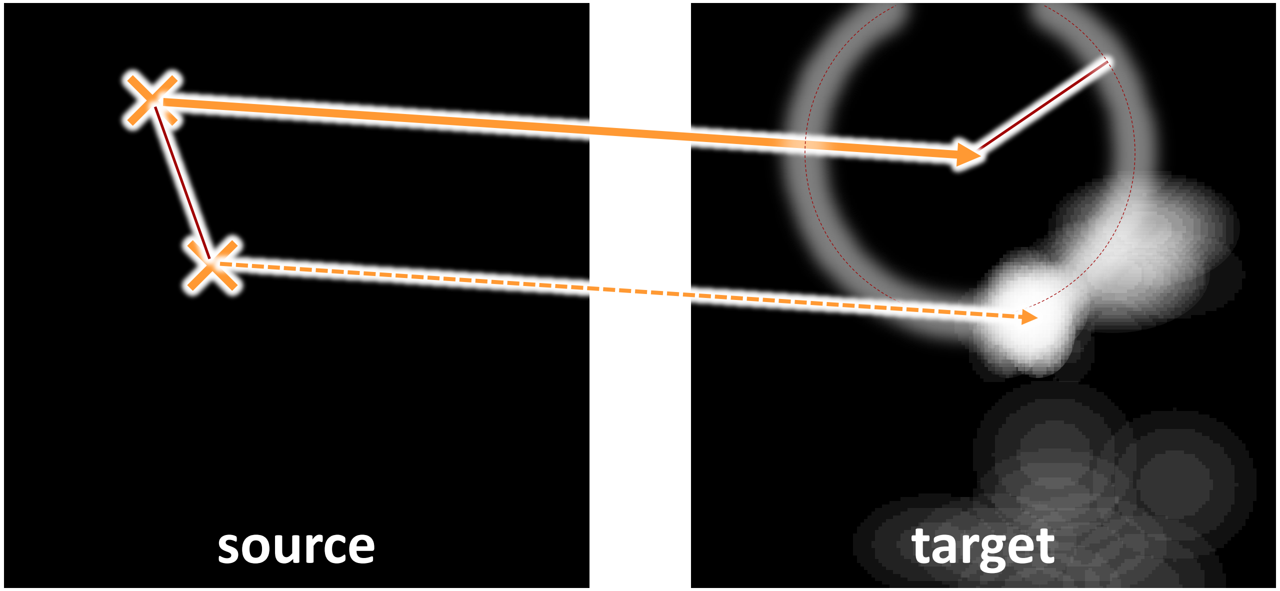
Foreward Search Algorithm



Step 2:

- Compute „posterior“ incorporating geodesic distance
  - Target side importance sampling:
    sample according to descriptor match × distance score
  - Again: optional source side imp. sampl: prefer unique descriptors

Foreward Search Algorithm

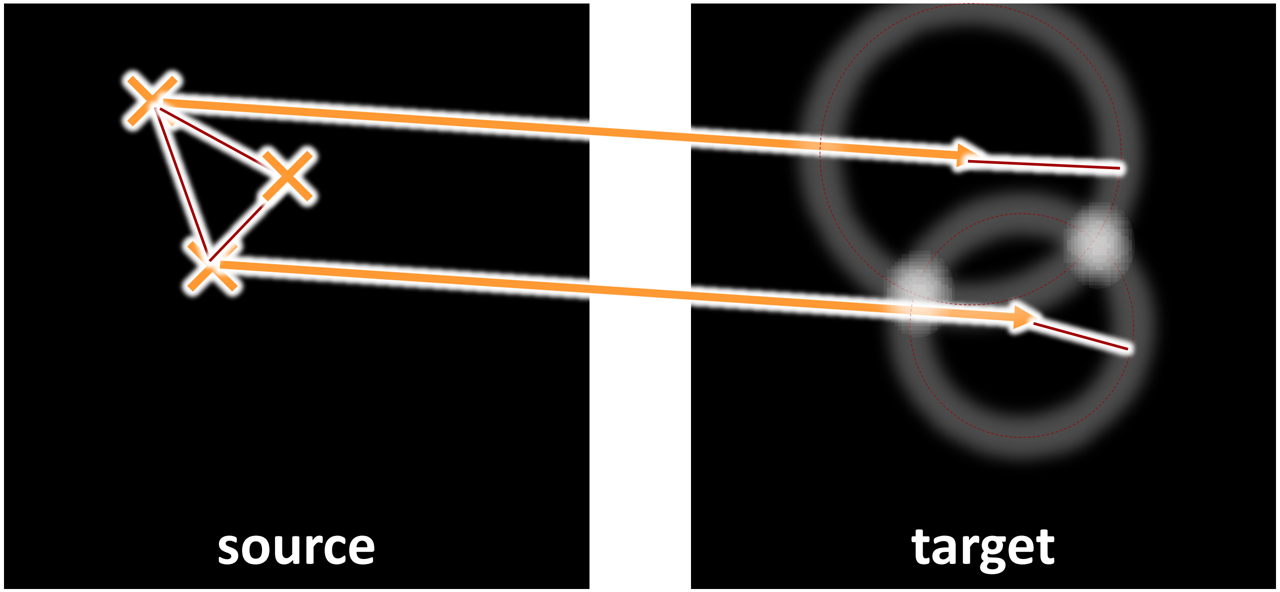


posterior (distance & descriptors)

Step 2:

- Compute „posterior“ incorporating geodesic distance
  - Target side importance sampling: sample according to descriptor match × distance score
  - Again: optional source side imp. sampl: prefer unique descriptors

Foreward Search Algorithm

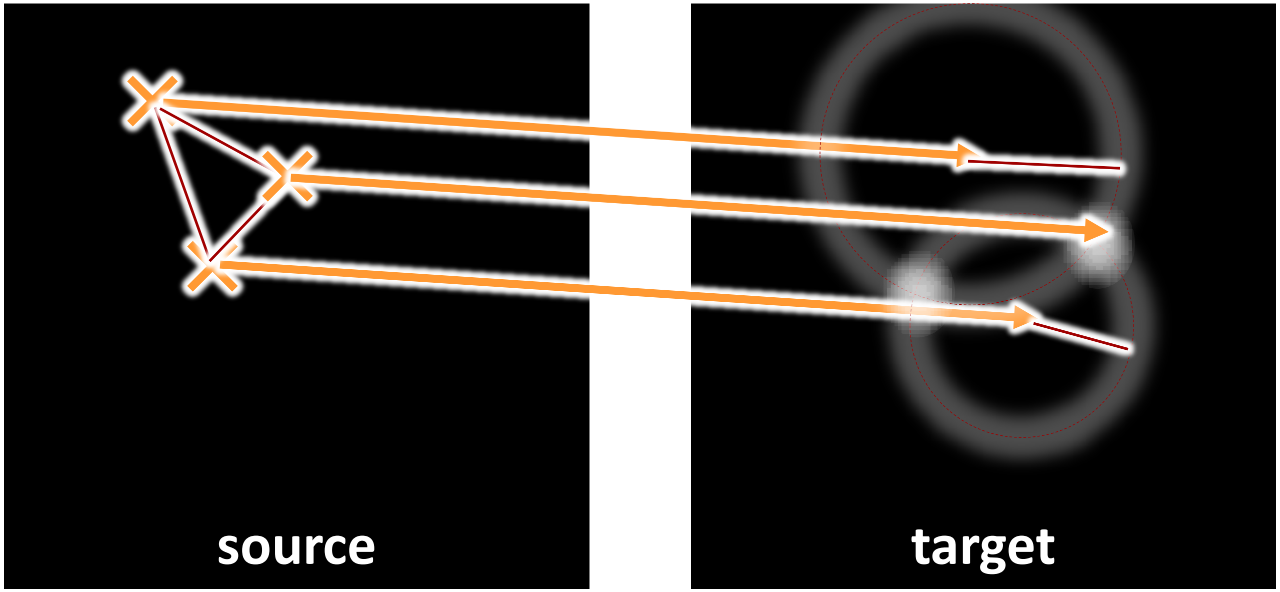


posterior (distance & descriptors)

Step 3:

- Same as step 2, continue sampling...

Foreward Search Algorithm

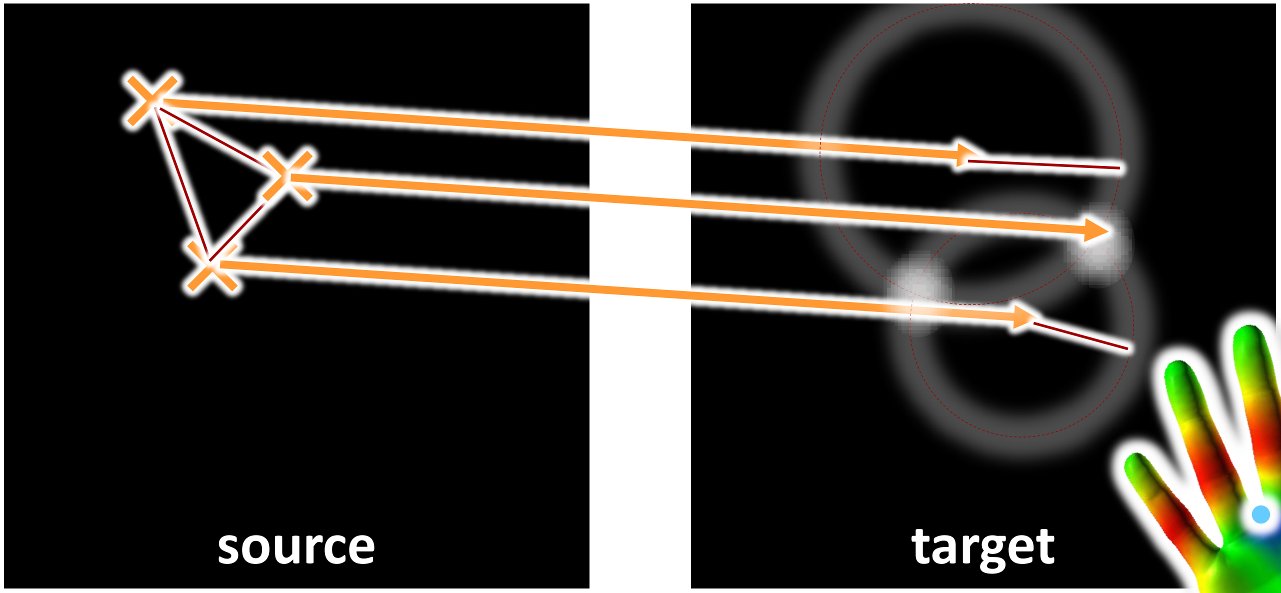


posterior (distance & descriptors)

Step 3:

- Same as step 2, continue sampling...

Foreward Search Algorithm



posterior (distance & descriptors)

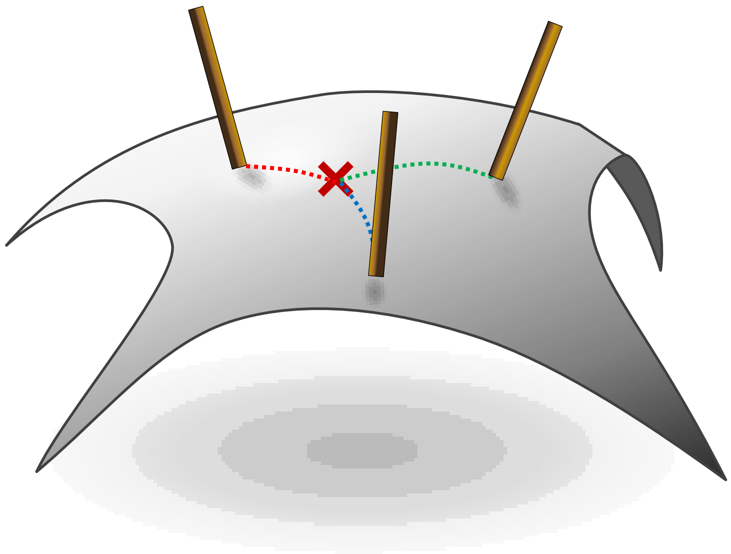
Source side:

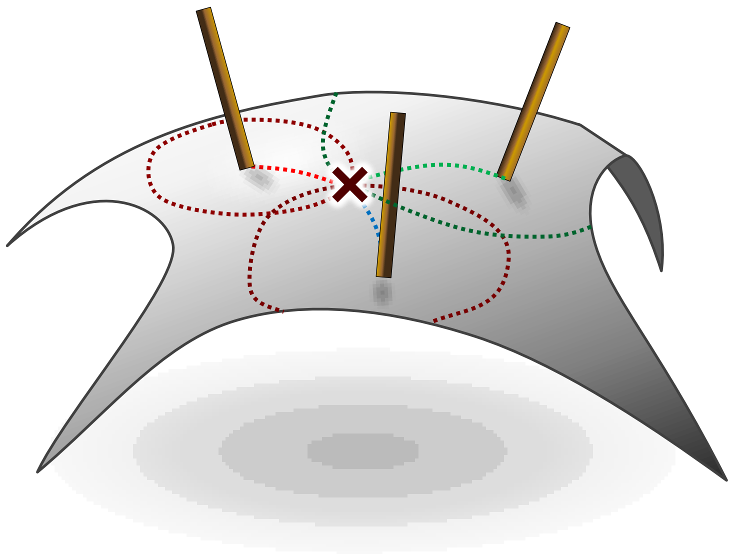
- Match all descriptors, compute entropy
- Choose minimum entropy features for start
- Subsequent features: consider entropy of all matches in addition

Another View

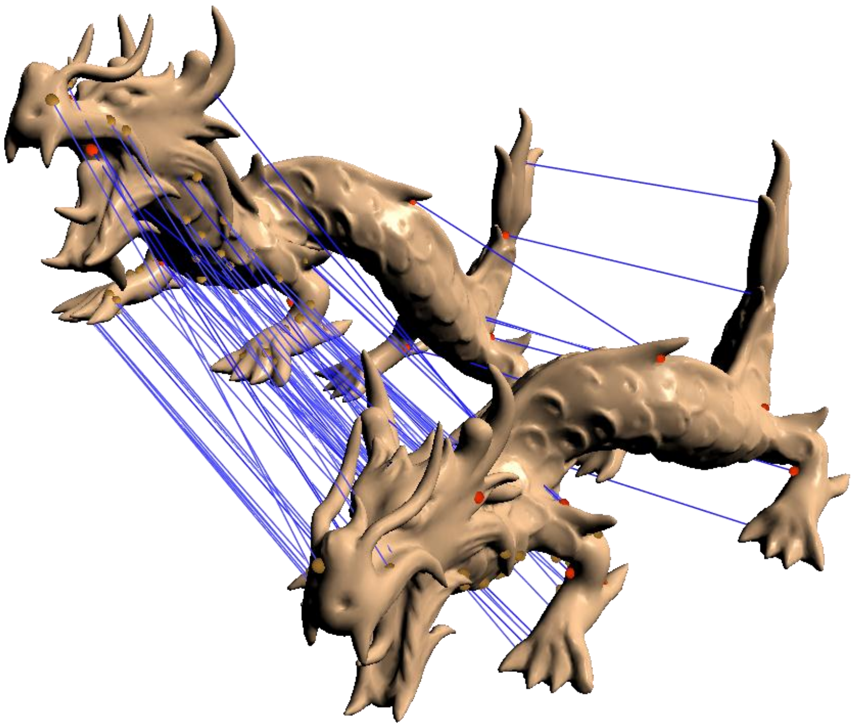
Landmark Coordinates

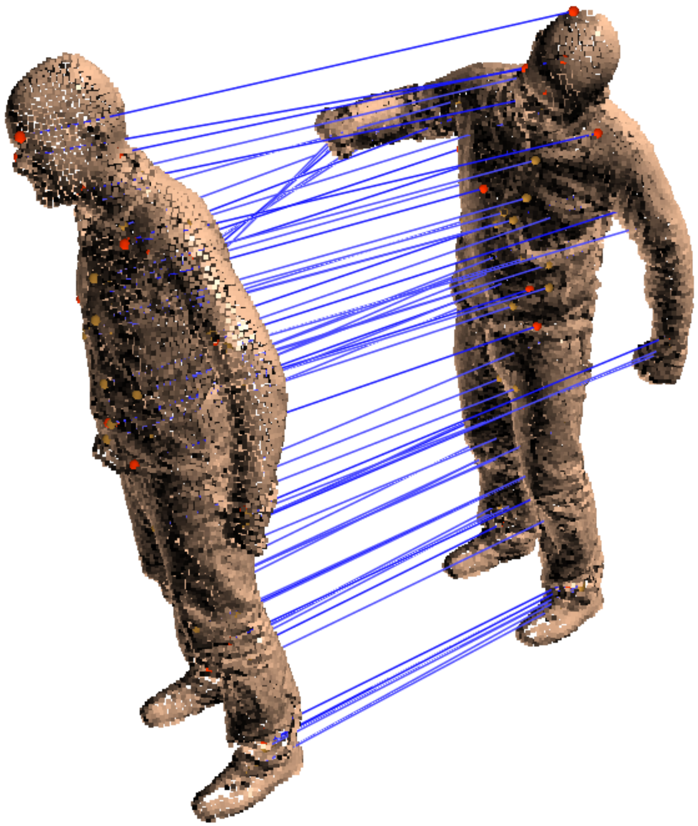
- Distance to already established points give a charting of the manifold





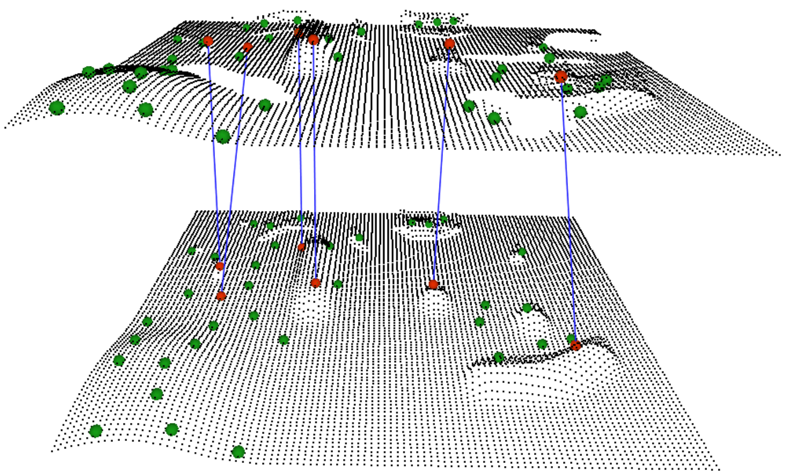
Results



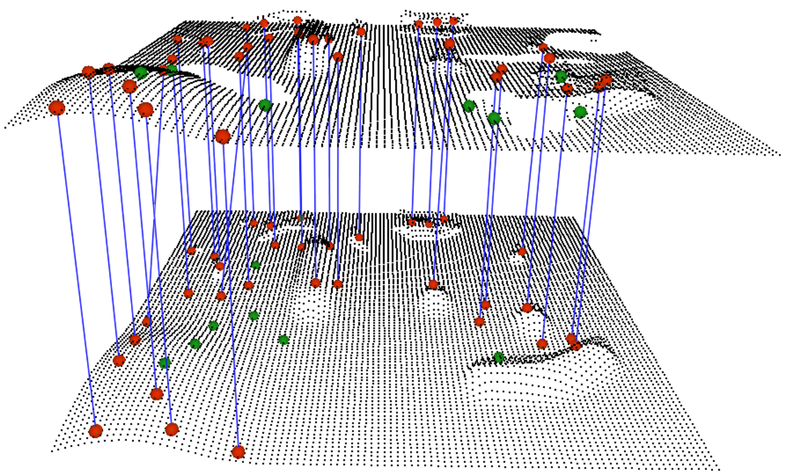


[data sets: Stanford 3D Scanning Repository / Carsten Stoll]

Results: Topological Noise



Spectral Quadratic Assignment
[Leordeanu et al. 05]



Ransac Algorithm
[Tevs et al. 09]

Complexity

How expensive is all of this?

Cost analysis:

- How many rounds of sampling are necessary?

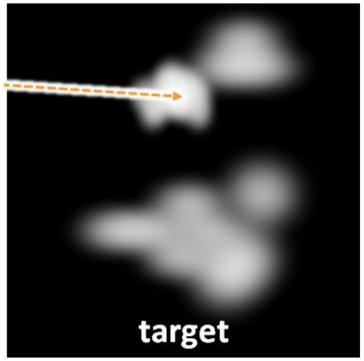
Constraints [Lipman et al. 2009]:

- Assume disc or sphere topology
- An isometric mapping is in particular a conformal mapping
- A conformal mapping is determined by 3 point-to-point correspondences

How expensive is it..?

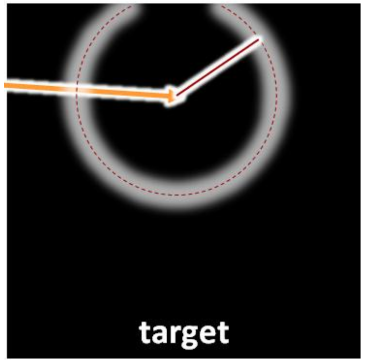
First correspondence:

- Worst case: n trials (n feature points)
- In practice: $k \ll n$ good descriptor matches (typically $k \approx$ 5-20)



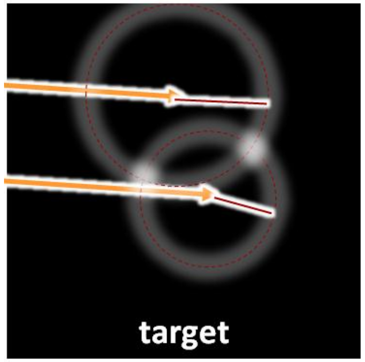
Second correspondence:

- Worst case: n trials, expected: $\sqrt{n}$ trials
- In practice: very few (due to descriptor matching, maybe 1-3)



Last match:

- At most two matches



Costs...

Overall costs:

- Worst case: $O(n^2)$ matches to explore
- Typical: $O(n^{1.5})$ matches to explore

Randomization:

- Exploring m items costs expected $O(m \log m)$ trials
- Worst case bound of $O(n^2 \log n)$ trials
- Asymptotically sharp: $O(c)$-times more trials for shrinking failure probability to $O(\exp(-c^2))$

Costs...

Surface discretization:

- Assume ε-sampling of the manifold (no features): O(ε^{-2}) sample points
- Worst case O($\varepsilon^{-4} \log \varepsilon^{-1}$) sample correspondences for finding a match with accuracy ε.
- Expected: O($\varepsilon^{-3} \log \varepsilon^{-1}$).

In practice:

- Importance sampling by descriptors is very effective
- Typically: Good results after 100 iterations
- Entropy-based planning: 1-10 iteartions

General Case

Numerical errors:

- Noisy surfaces, imprecise features: reflected in probability maps (we know how little we might know)

Topological noise:

- Use robust constraint potentials
- For example: account for 5 best matches only

Topologically complex cases:

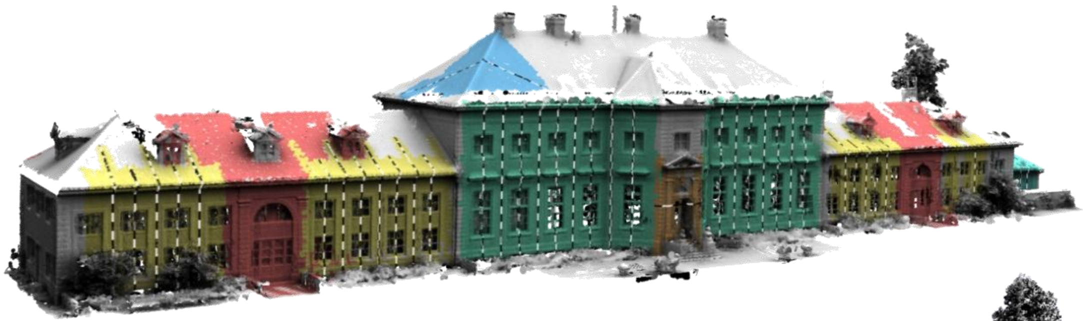
- No analysis beyond disc/spherical topology
- However: the algorithm will work in the general case (potentially, at additional costs)

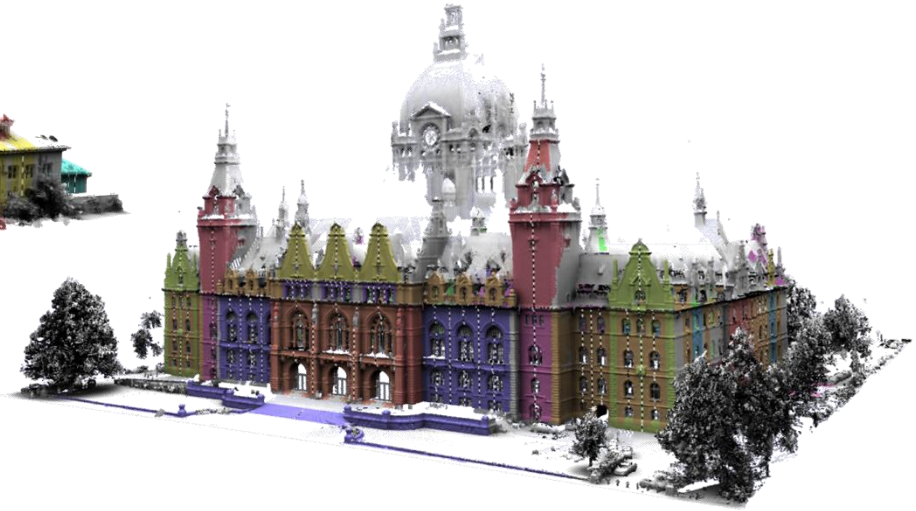
Other Application:
Symmetry Detection

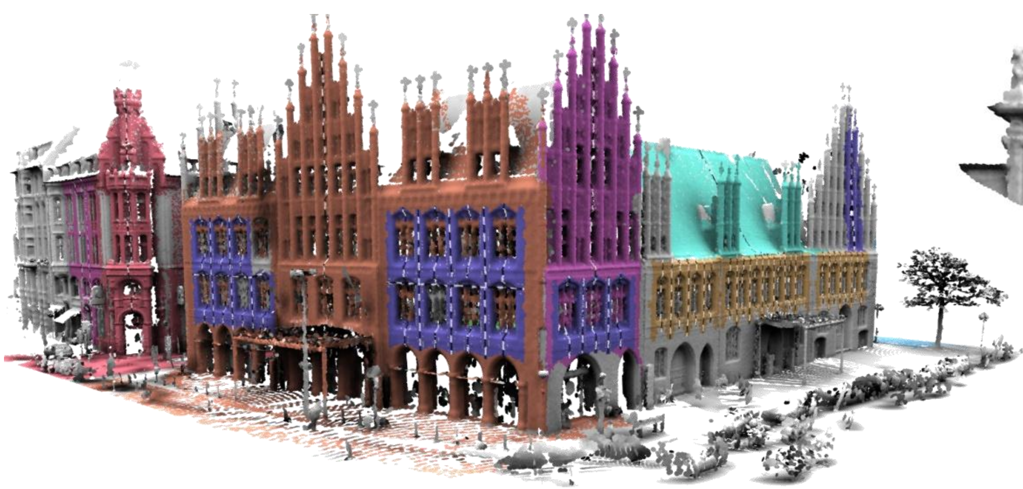
Symmetry Detection

[data set: M. Wacker, HTW Dresden]

Symmetry Detection







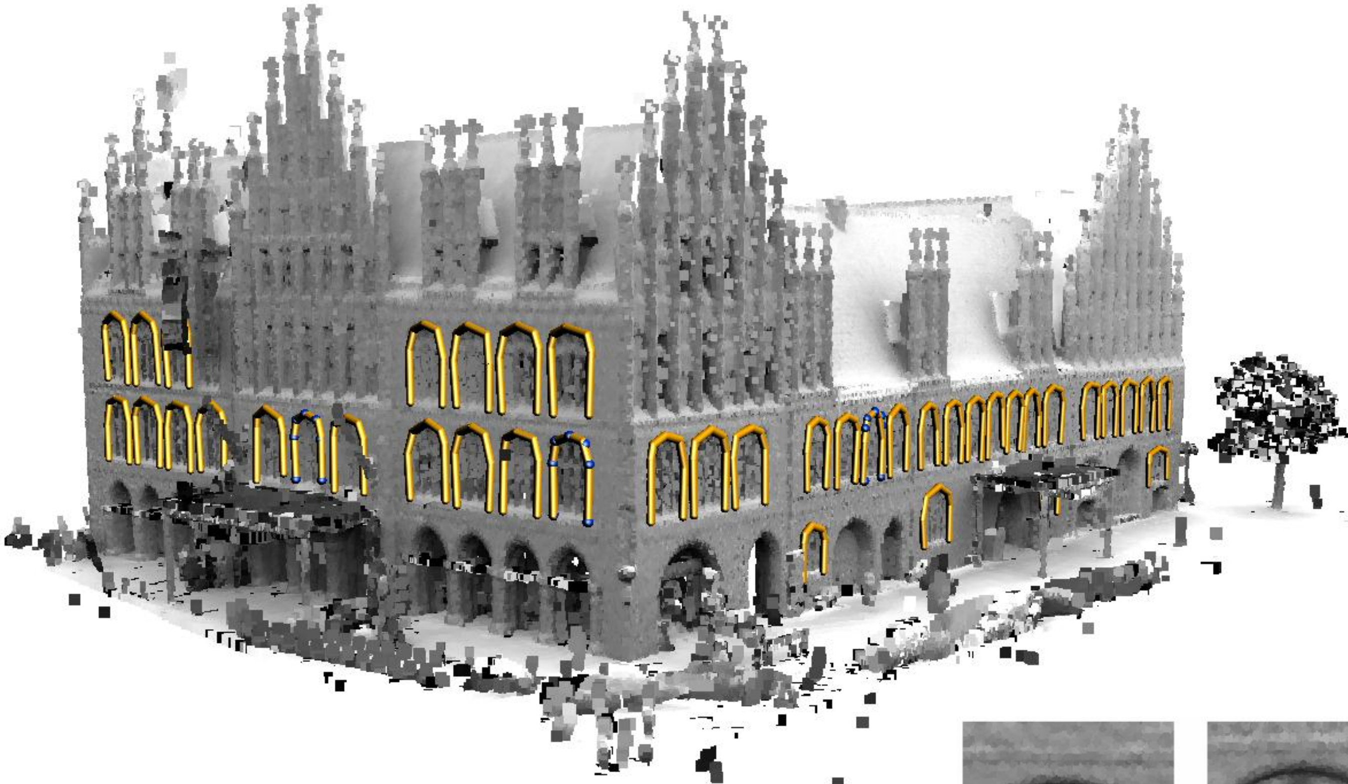
[data sets: IKG, Leibnitz University Hannover / M. Wacker, HTW Dresden]

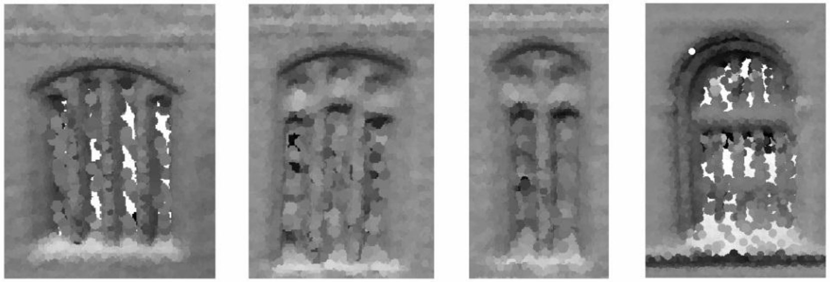
Rigid, Isometic, Relaxed Isometric

rigid isometric relaxed isometric

Learning Correspondences

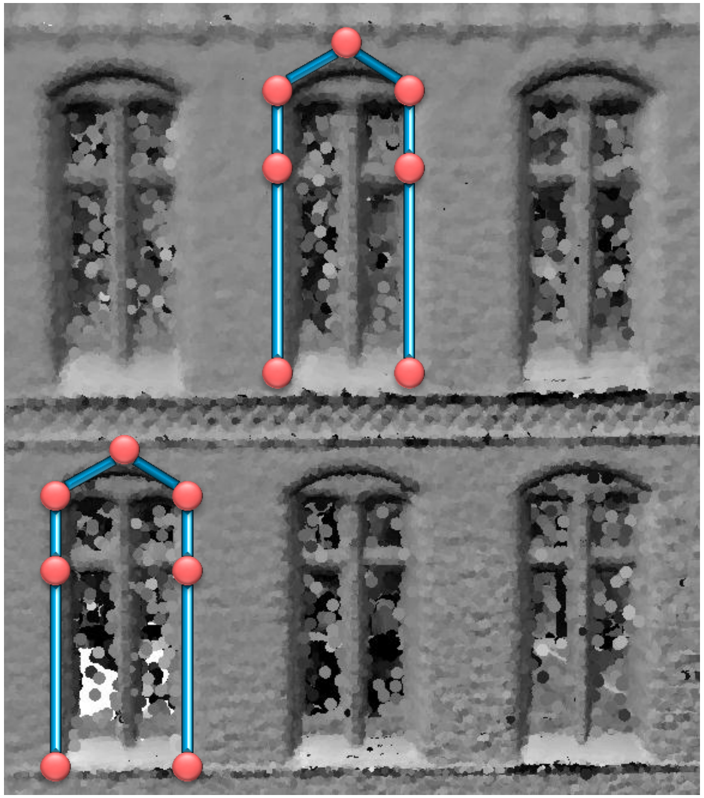
Objective





Window Variants

Objective



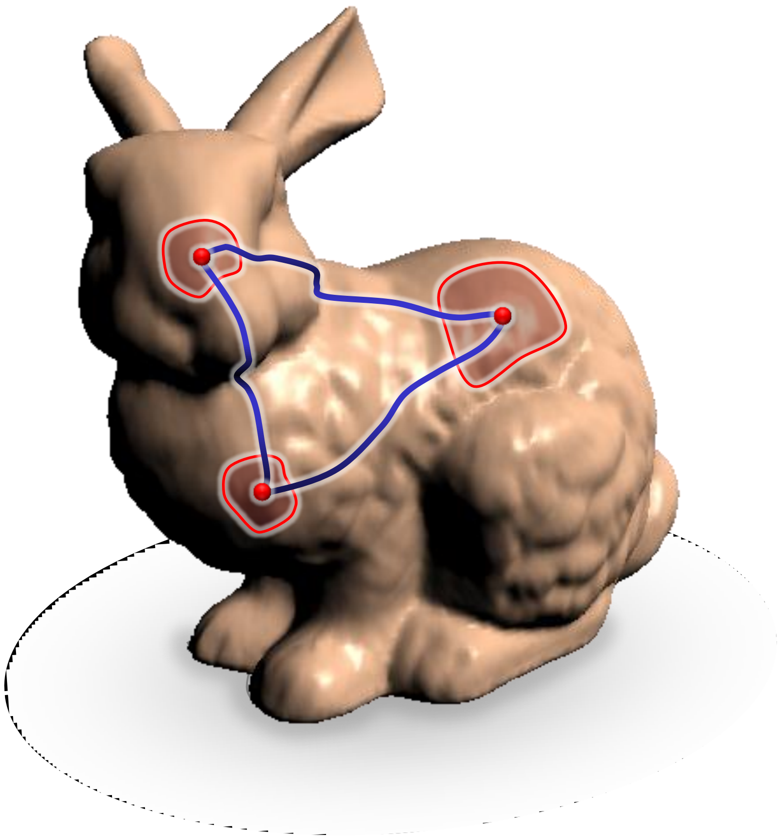
User: a few sparse sketches

Find similar elements

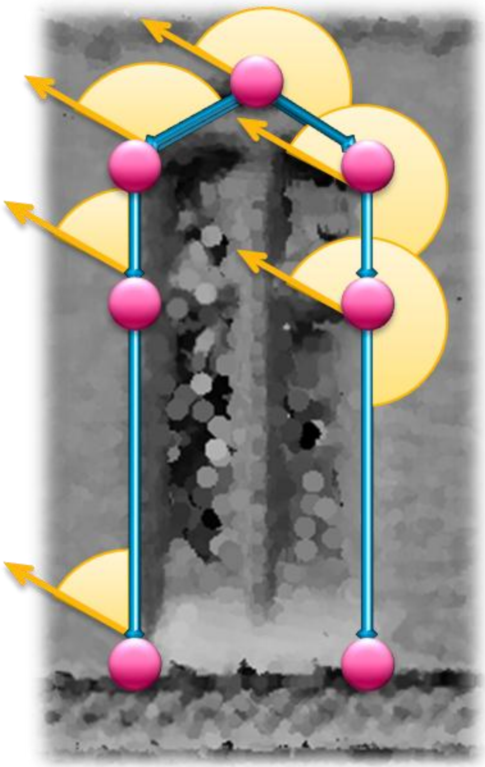
Learning a Matching Model

Learning a matching model

- Learn *descriptors*
- Learn *geometric relations*



Energy Function

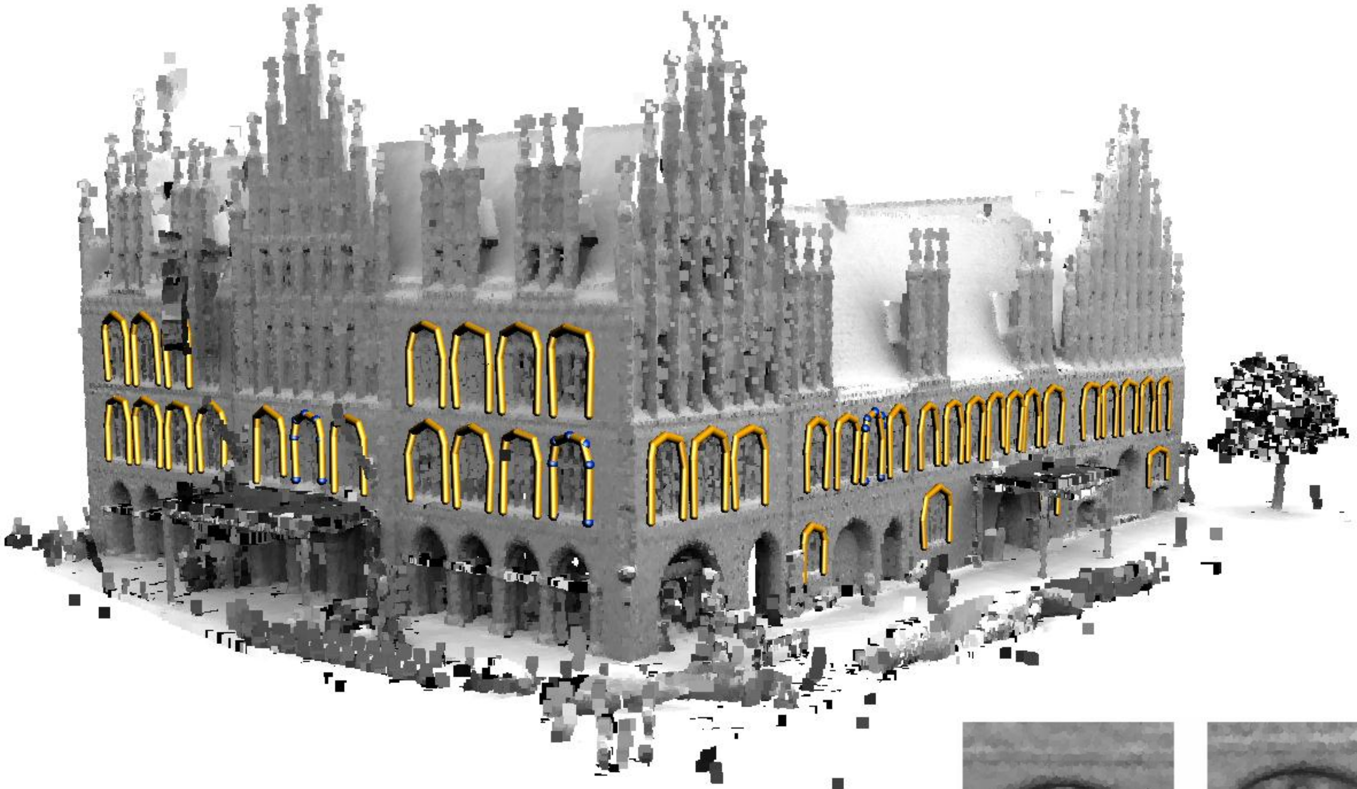


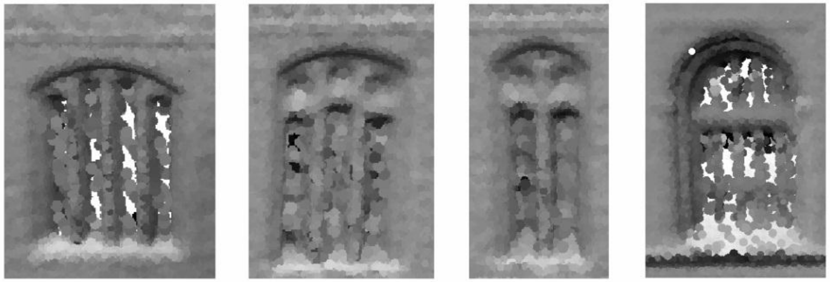
$$\frac{1}{Z}\prod_{i=1}^{k}\Phi_{\mathrm{i}}(\mathbf{x}_{\mathrm{i}})\prod_{i=1}^{k-1}\Psi_{\mathrm{i}}(\mathbf{x}_{\mathrm{i}},\mathbf{x}_{\mathrm{i+1}})$$

Markov Chain Model

- Global optimum: Belief propagation
- Symmetry: Enumerate local optima

Result: Single-Class Learning





Window Variants

Result: Multi-Class Learning

Results: Ludwigskirche