Many to Many Matching in Graphs

Ali Shokoufandeh,

Department of Computer Science, Drexel University

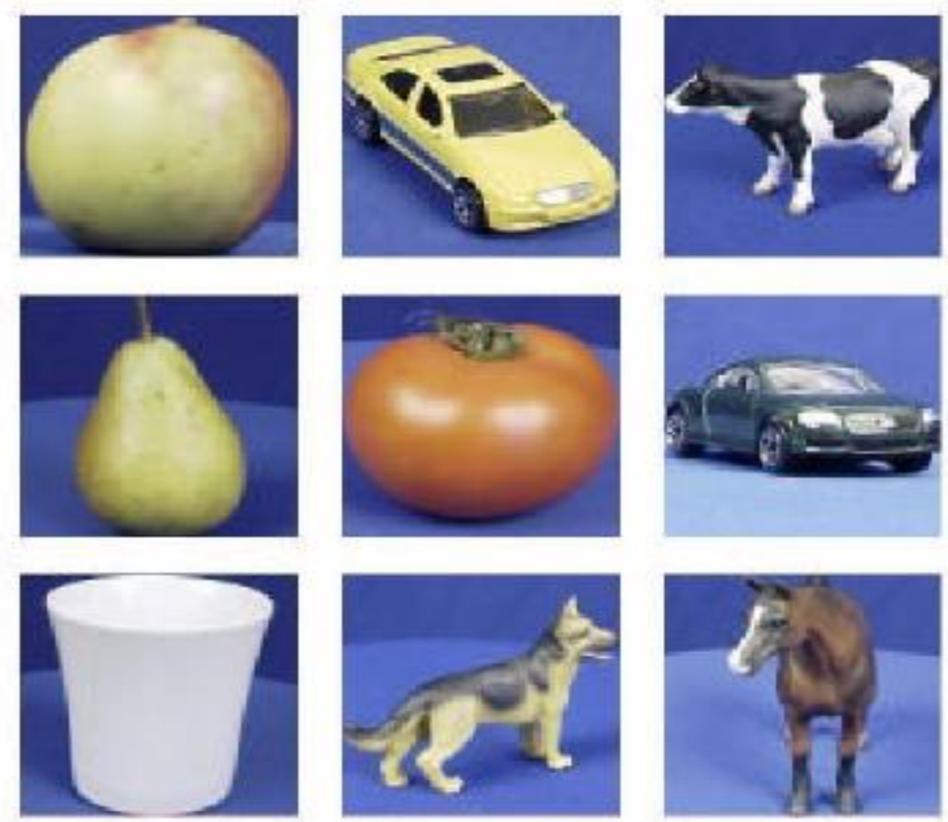
Object Recognition

Query

Database

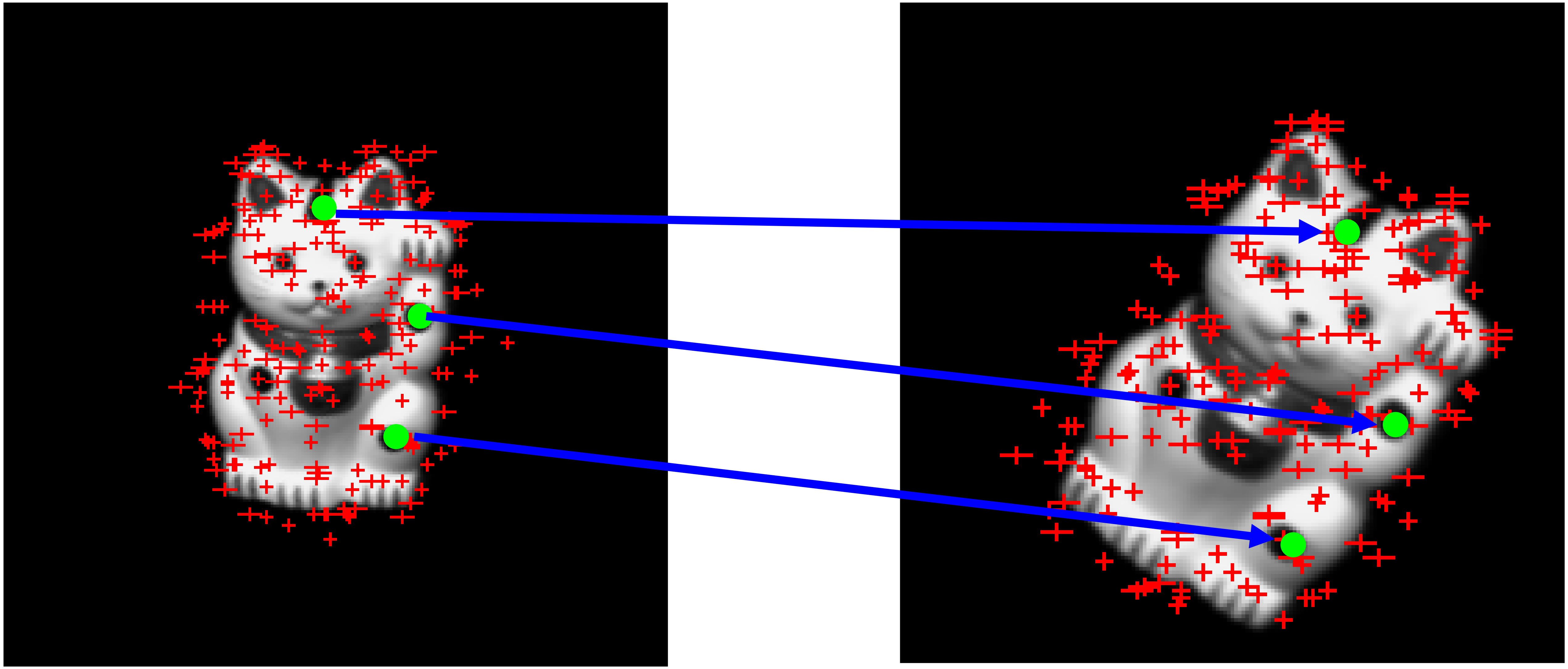
Recognition

?

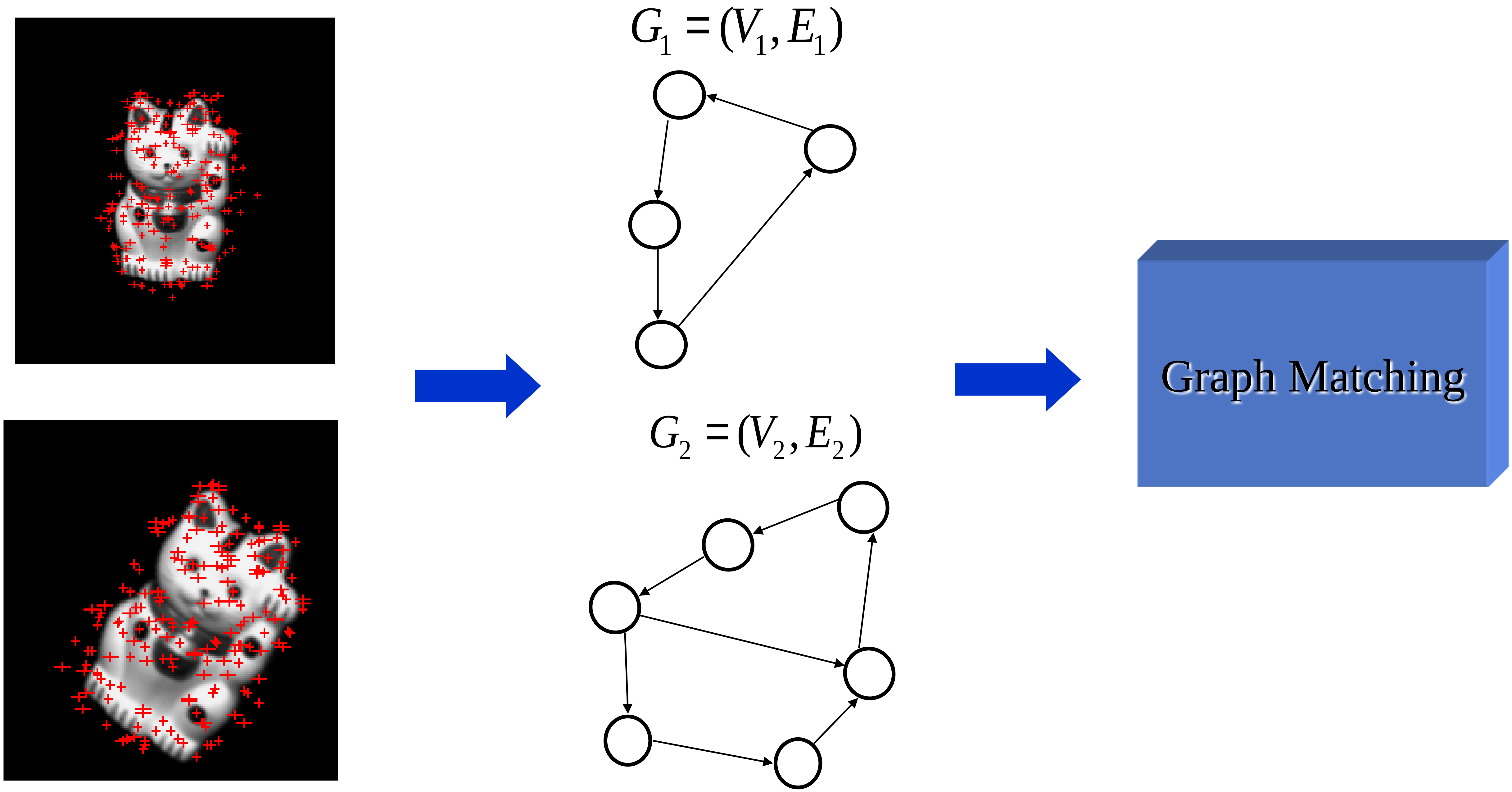


[Lebie et al., CVPR, 2003].

Feature Extraction and Correspondence

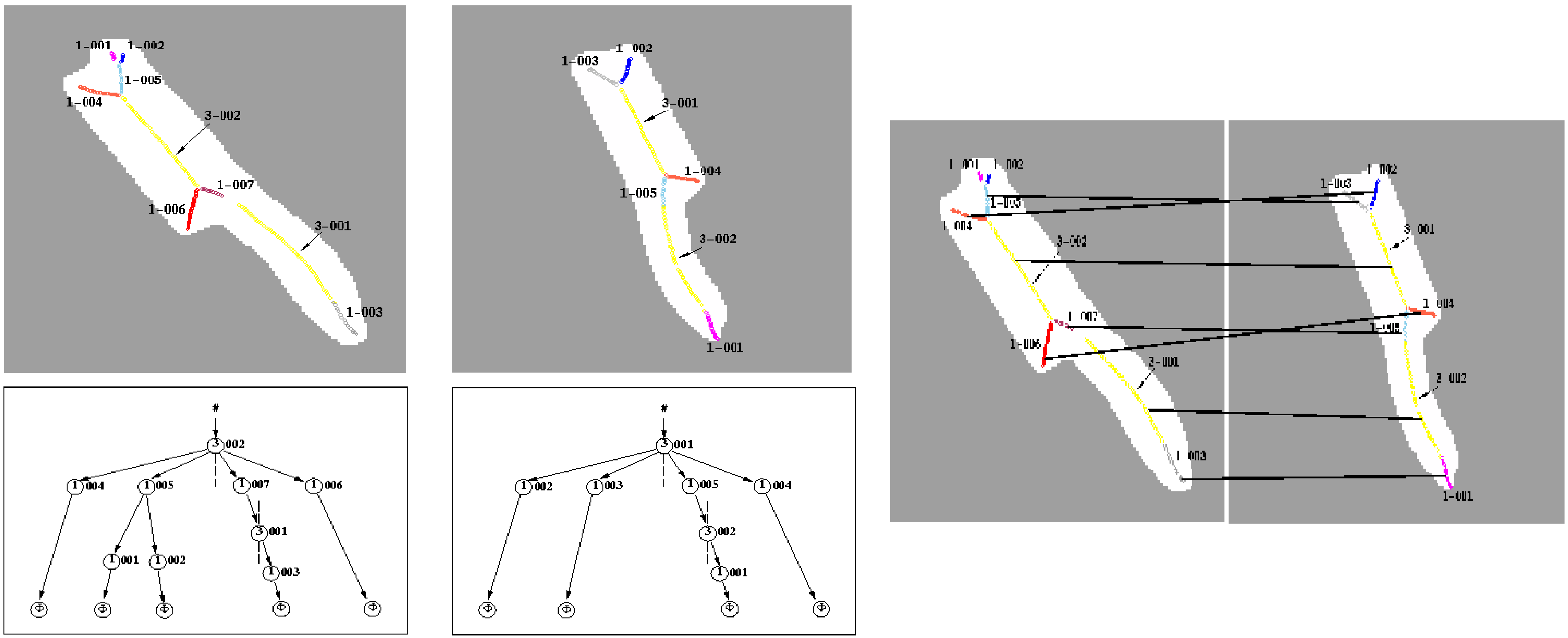


The Role of Graph Matching



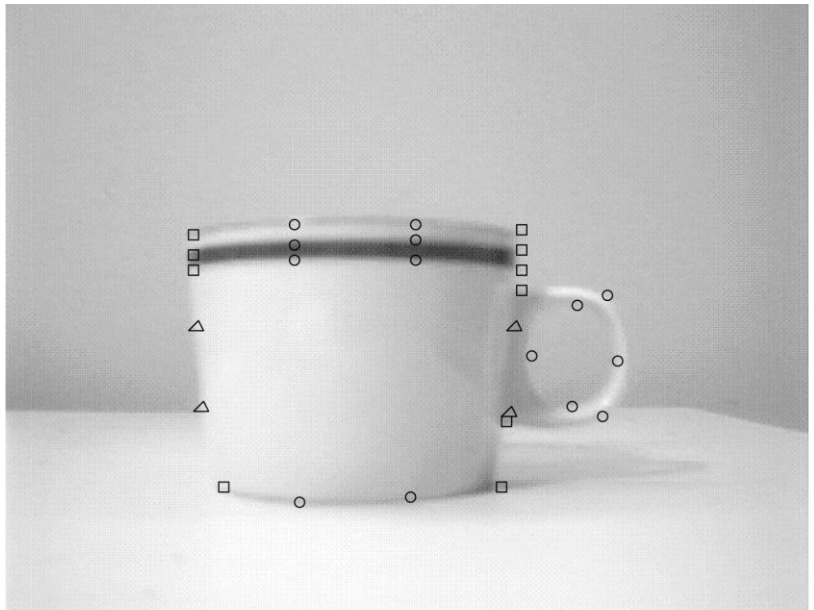
Matching Problem

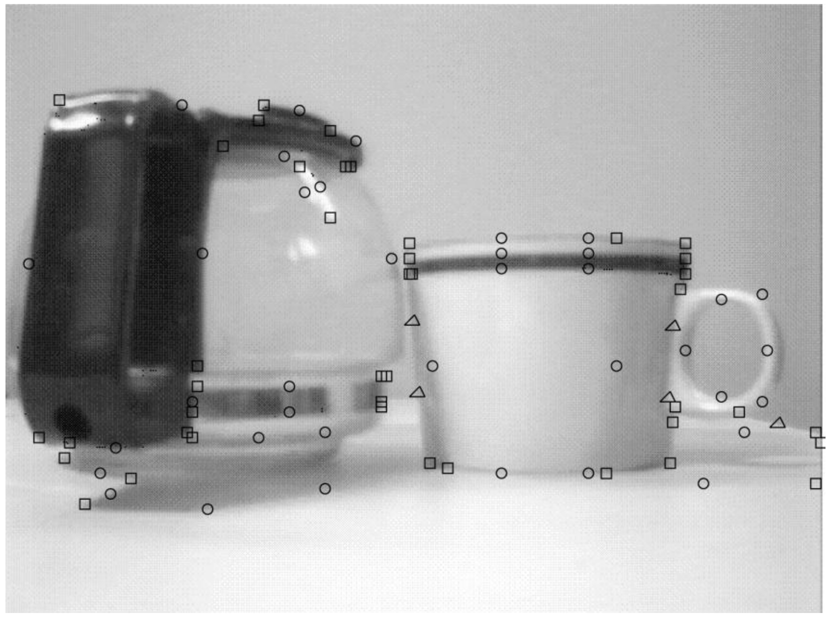
- Feature matching is an important step in recognition systems.



One to one matching

- Most approaches to shape matching assume a **one-to-one** correspondence between image features and model features.
- This restriction pushes object recognition toward **exemplar-based** recognition.

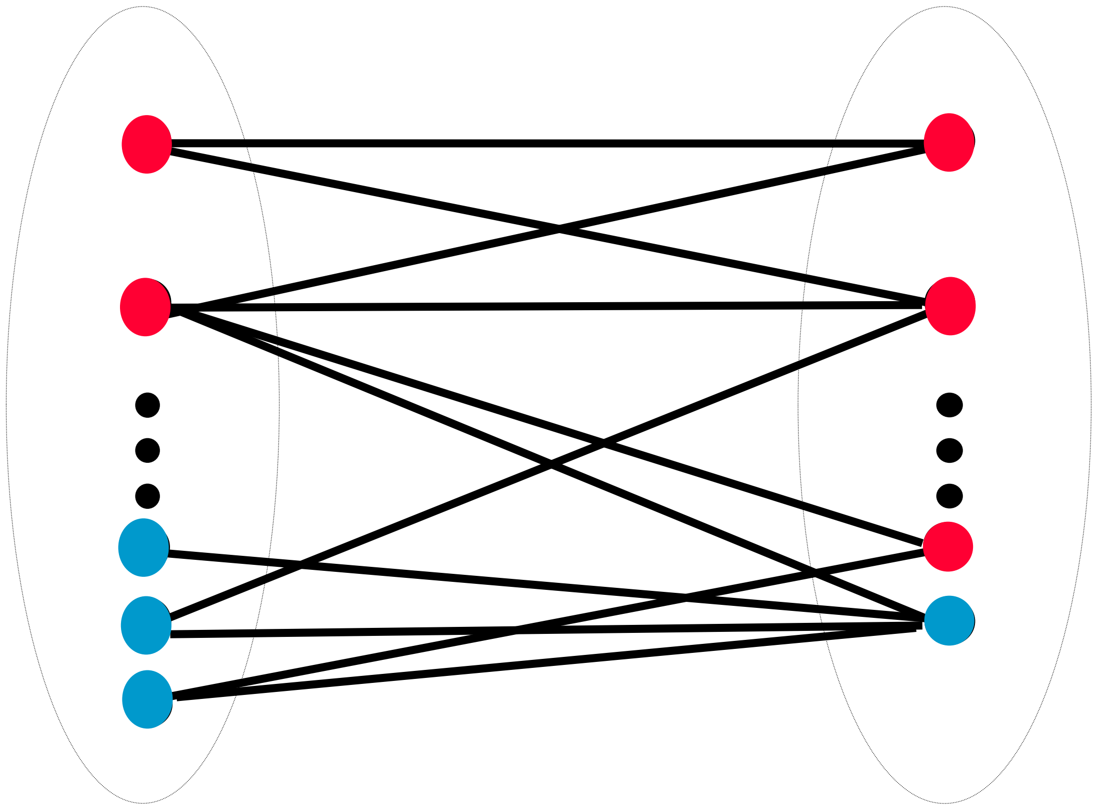




Gold and Rangarajan, A Graduated Assignment Algorithm for Graph Matching, IEEE PAMI, Volume 18, Number 4, April 1996.

Graph Matching

- Graph matching is an important component in many object recognition algorithms.
- Two main types of graph matching algorithms:
  - One-to-One
  - Many-to-Many

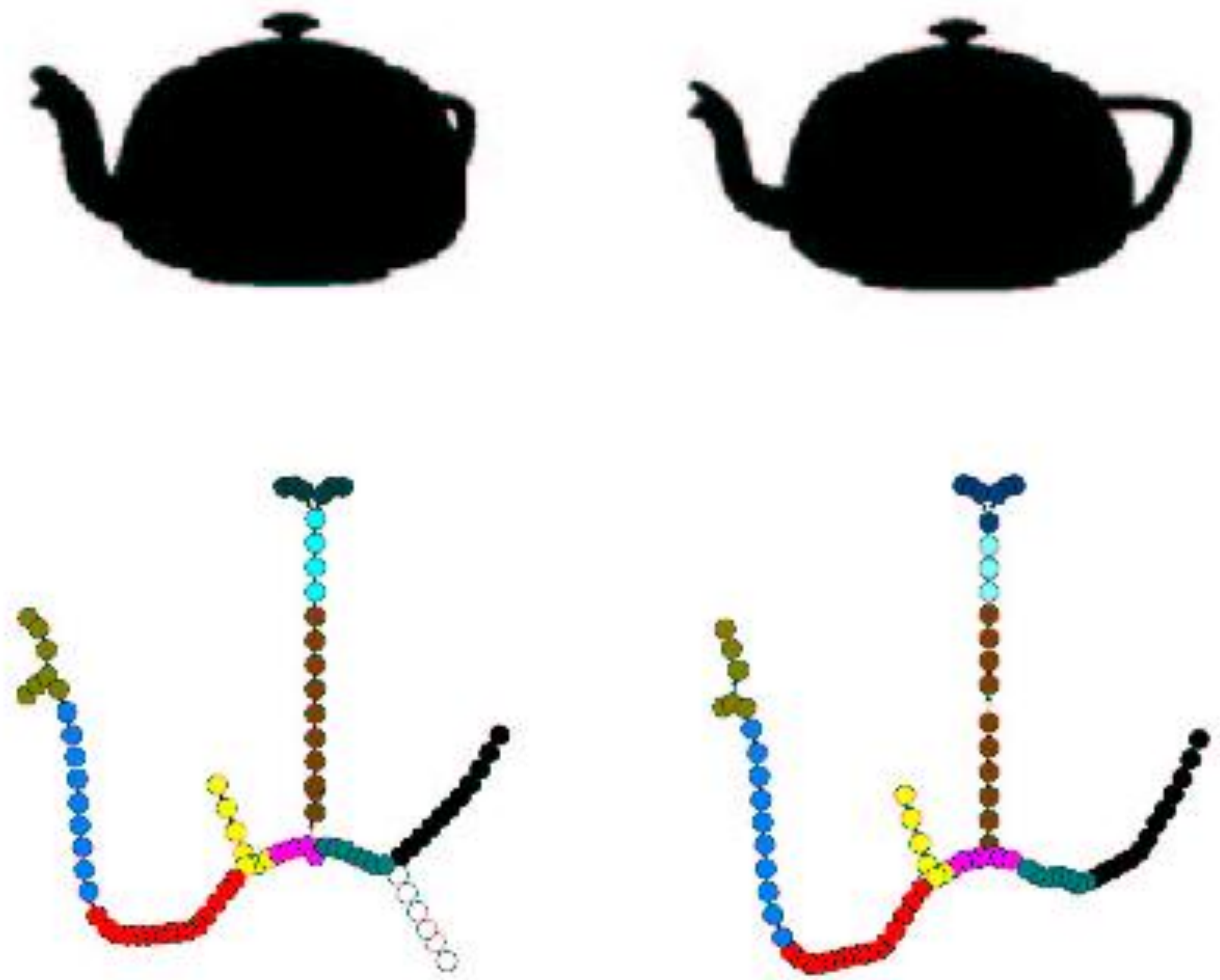


Motivation

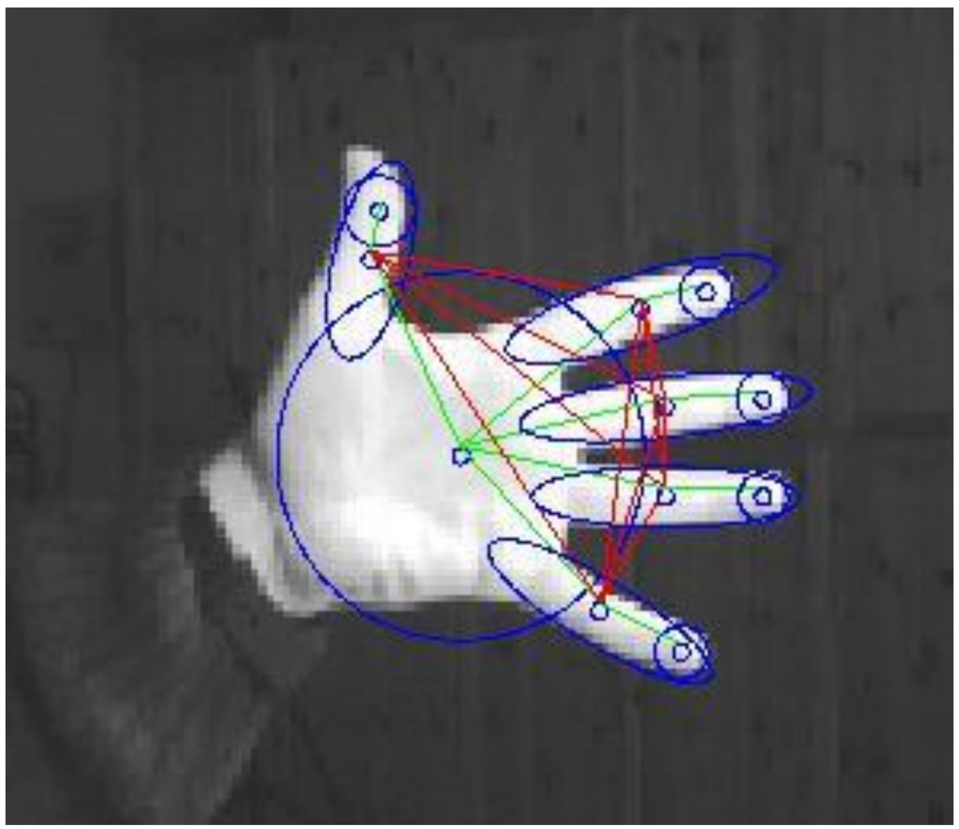
- But different exemplars belonging to the same **category** may not share a single low-level feature (e.g., interest point, contour, region, etc.).

- Only at higher-levels of **abstraction** does within-class one-to-one feature correspondence occur.

Medial Axis Representation



Multi-Scale Qualitative Shape Description



Shokoufandeh, Dickinson, Jönsson, Bretzner, Lindeberg (ECCV '02)

- Scale-space representation of image signal

$$L(\cdot\,;t) = g(\cdot\,;t) * f(\cdot)$$

- **Blobs** (compact regions) are detected as local maxima in scale-space of the square of:

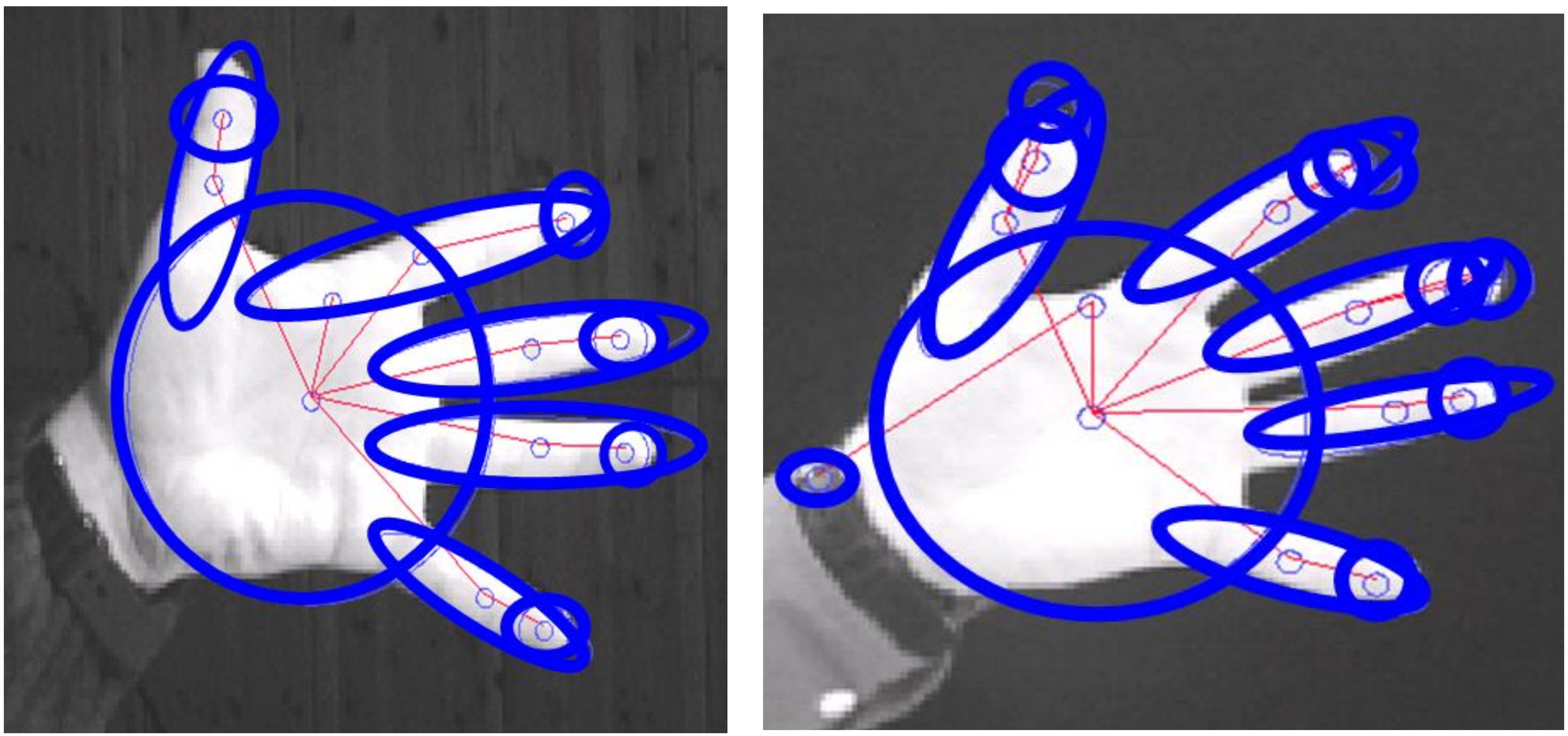
$$\nabla^2_{norm} L = t(L_{xx} + L_{yy})$$

- **Ridges** (elongated structures) are detected as local maxima of

$$R_{norm} L = t^{3/2}(L_{pp} - L_{qq})^2$$
$$= t^{3/2}((L_{xx} - L_{yy})^2 + 4L_{xy}^2)$$

The Need for Many-to-Many Matching

- One-to-one correspondence … may be not!

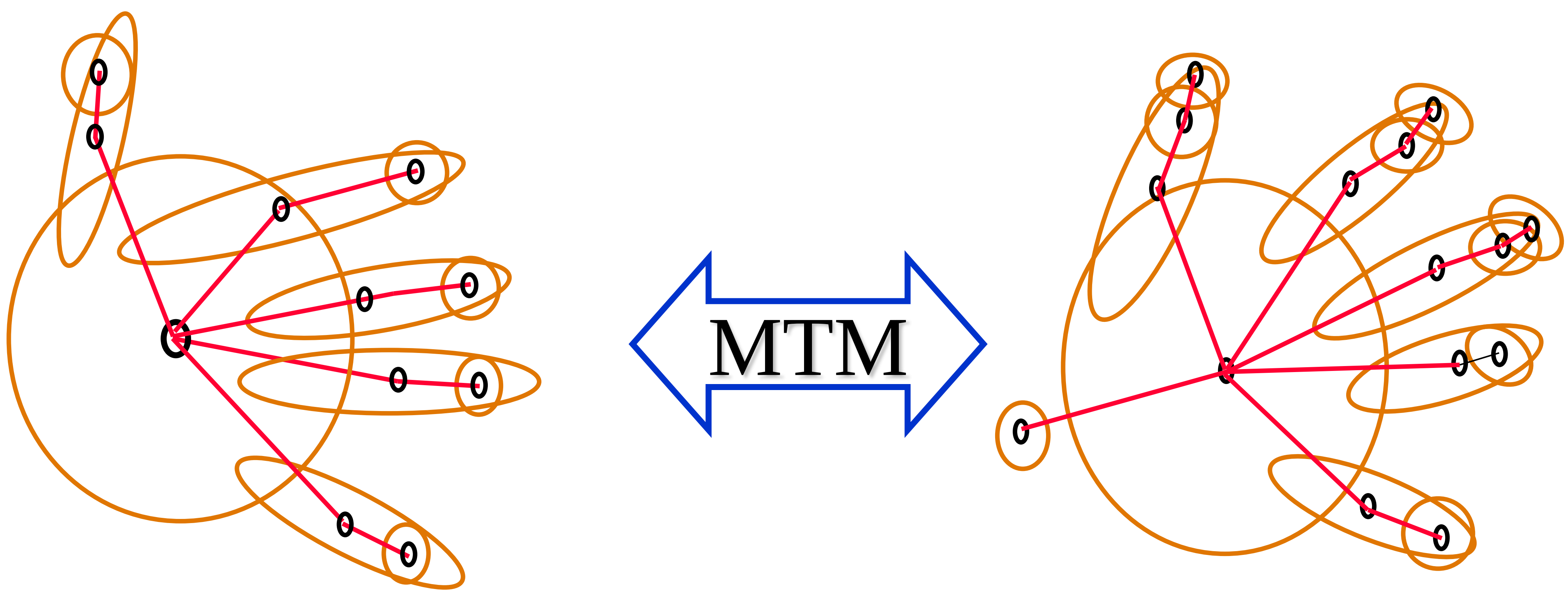


Similar objects, but extracted features do not match one-to-one!

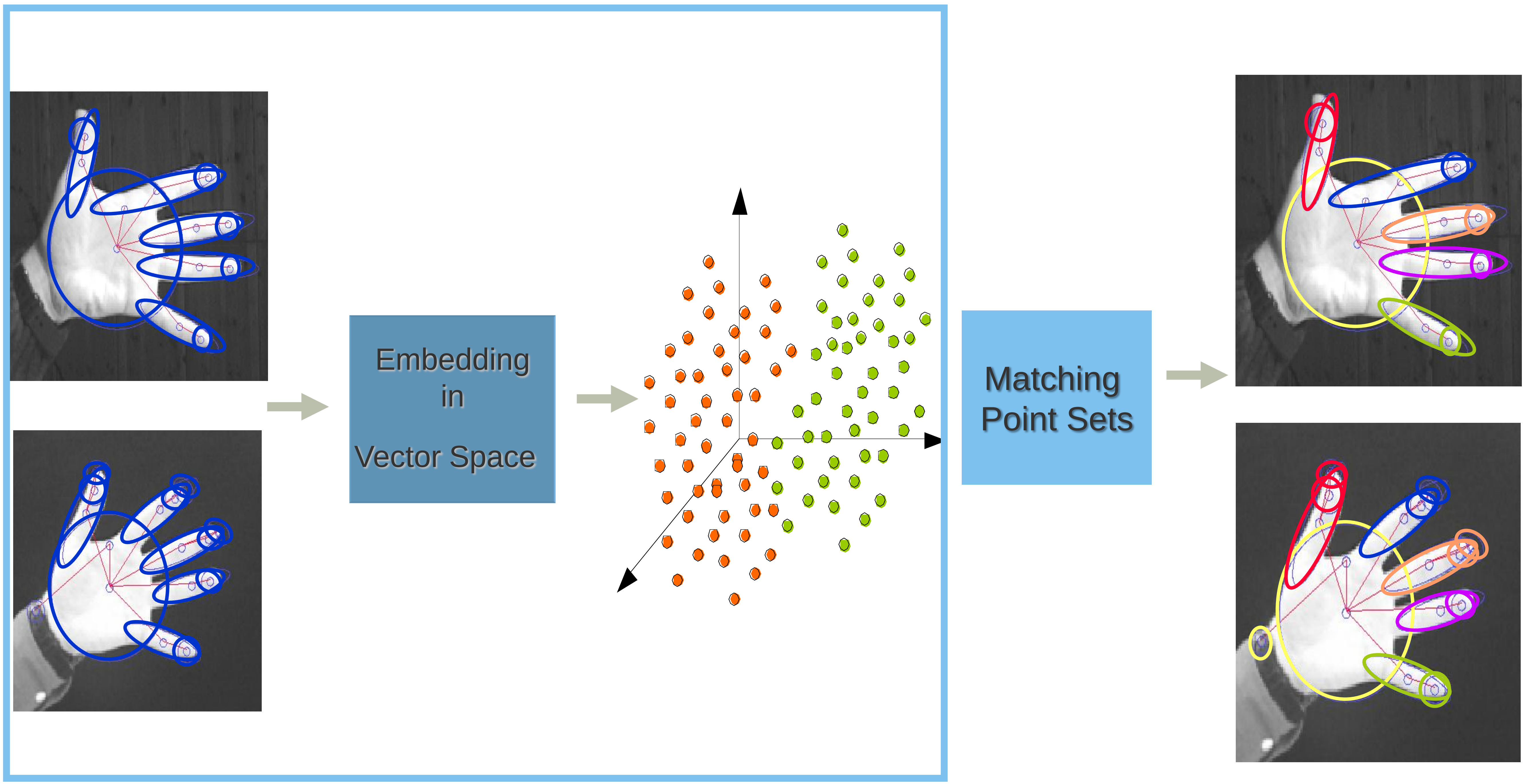
Problem Statement

- Compute many-to-many feature correspondences between graph pairs.

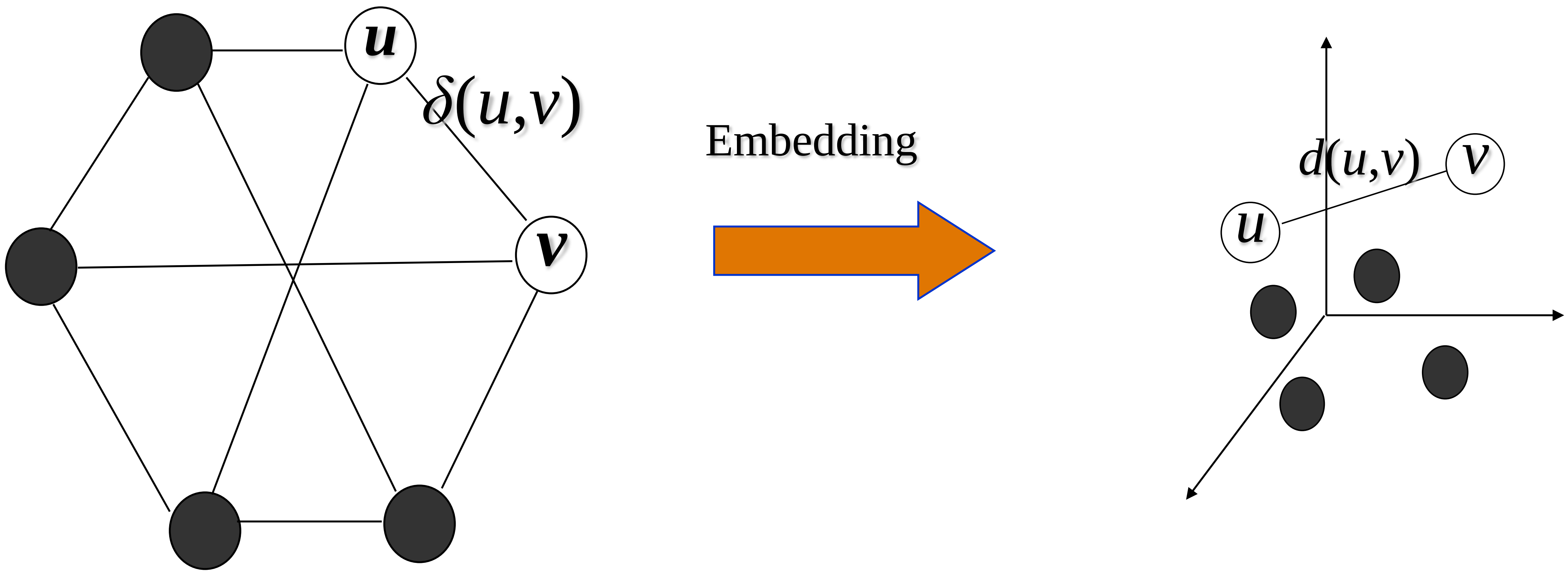
Combinatorial challenge!



Proposed Many-to-Many Approach



Embedding

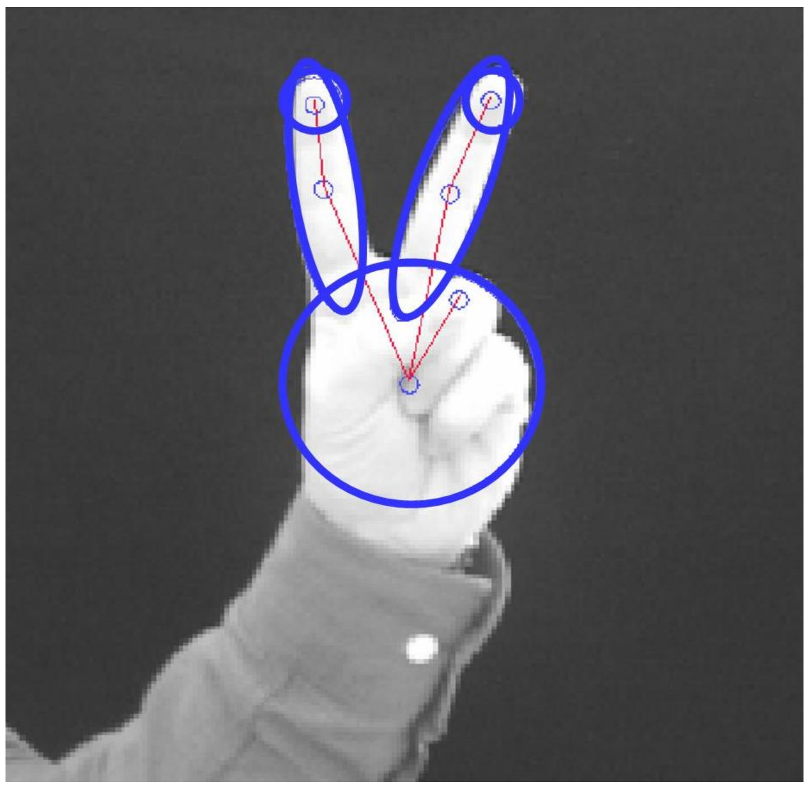


Two important isues:

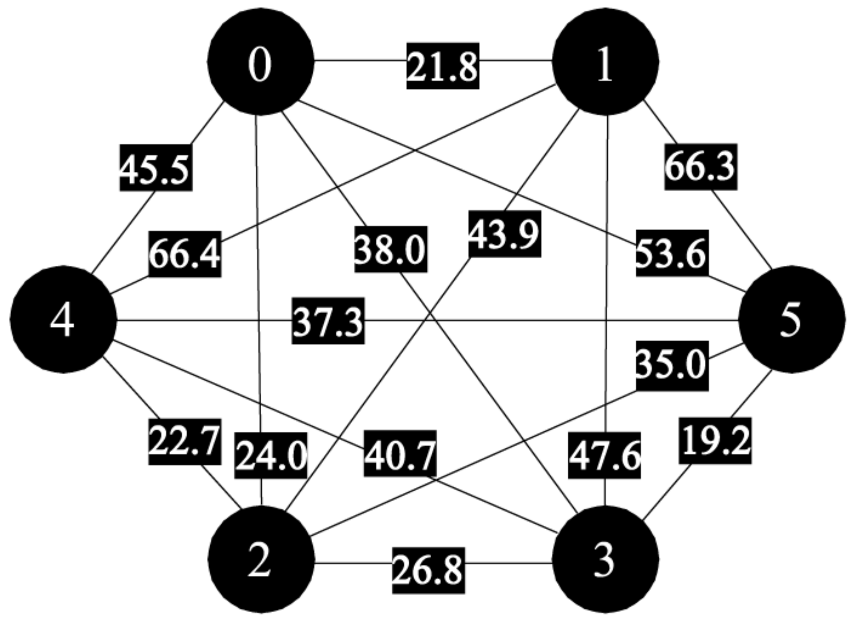
1. Distortion: $\left(\max_{x,y\in G}\frac{d(x,y)}{d(x,y)}\right)\cdot\left(\max_{z,t\in G}\frac{d(z,t)}{d(z,t)}\right)$

2. Dimension of the target space

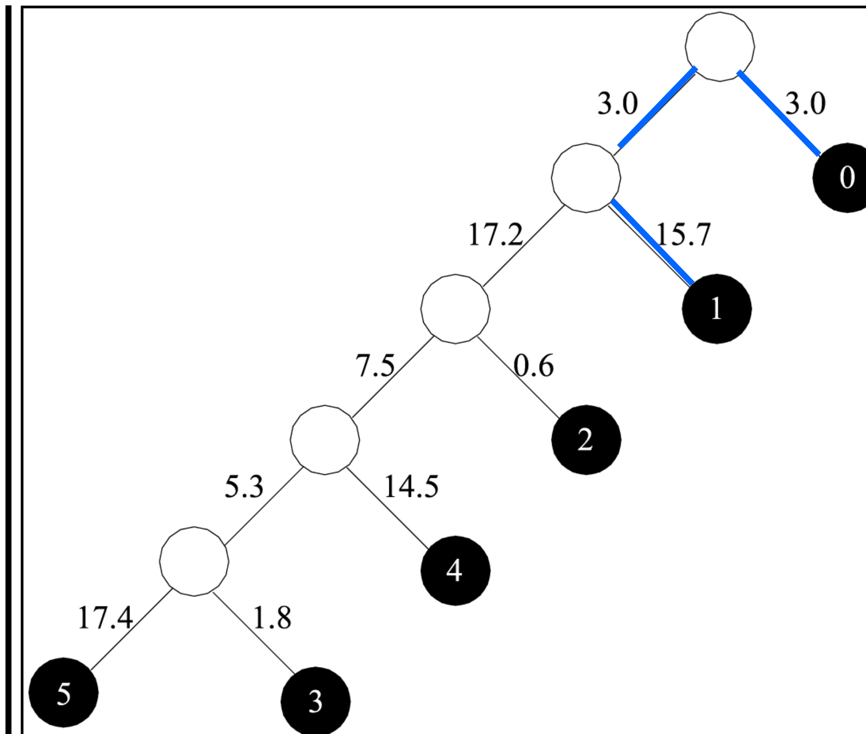
Tree Metric Construction Example



original graph

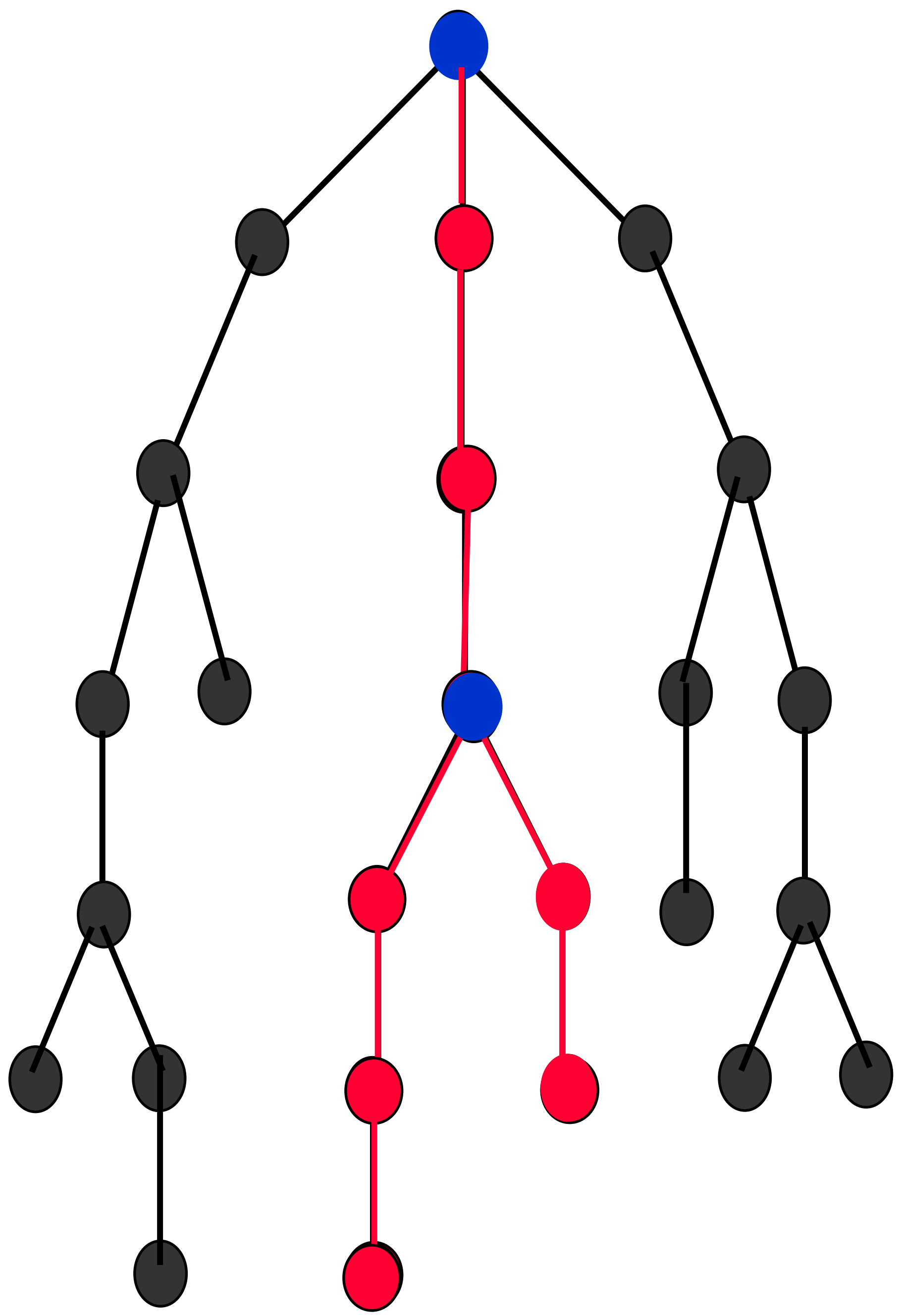


complete graph with edge weights equal to Euclidean distance between region centroids



resulting low-distortion tree metric with additional vertices.

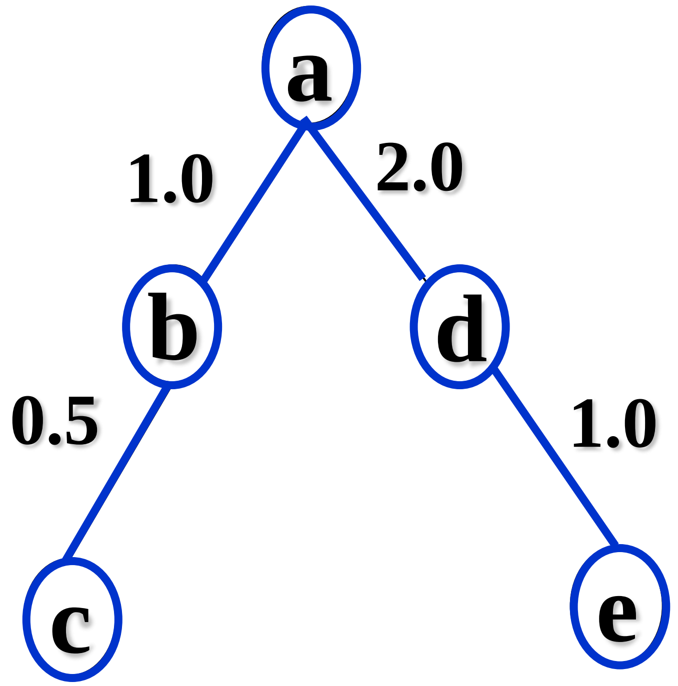
Caterpillar Decomposition



Assume all edge weights are equal to 1.

Repeat the same algorithm for every sub tree

Graph-Dependent Embedding



a: (0,0)

b: (1,0)

c: (1.5,0)

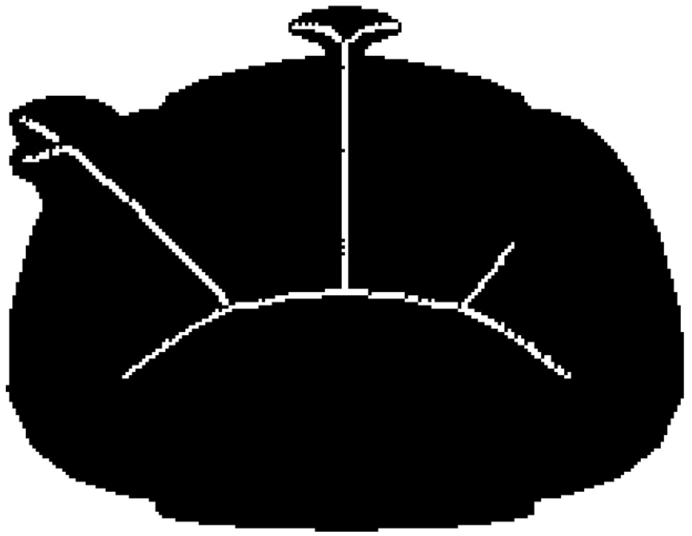
d: (0,2)

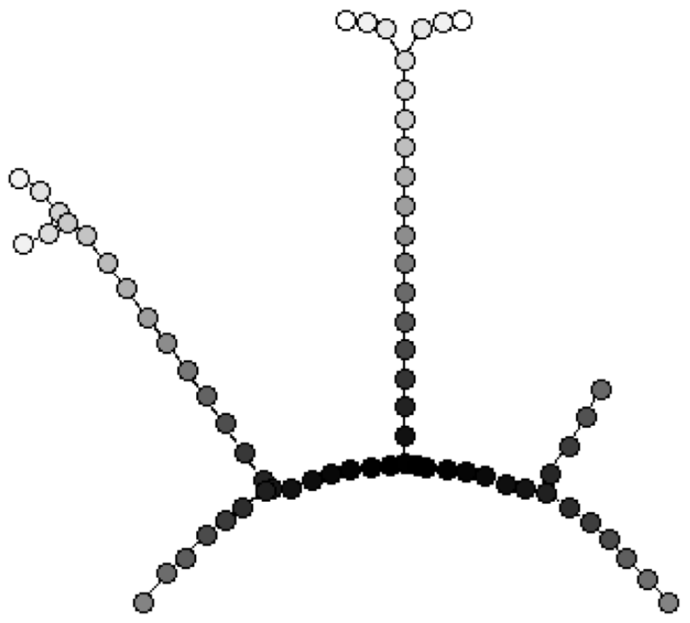
e: (0,3)

Better Cases:

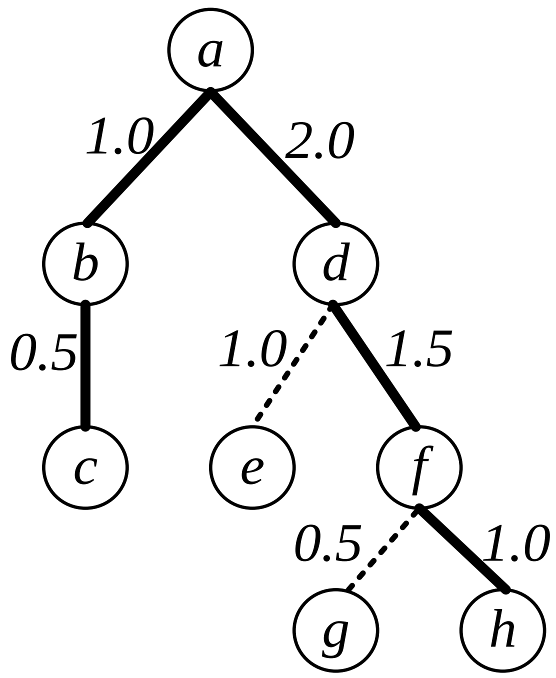
- The features have a natural representation as a tree!

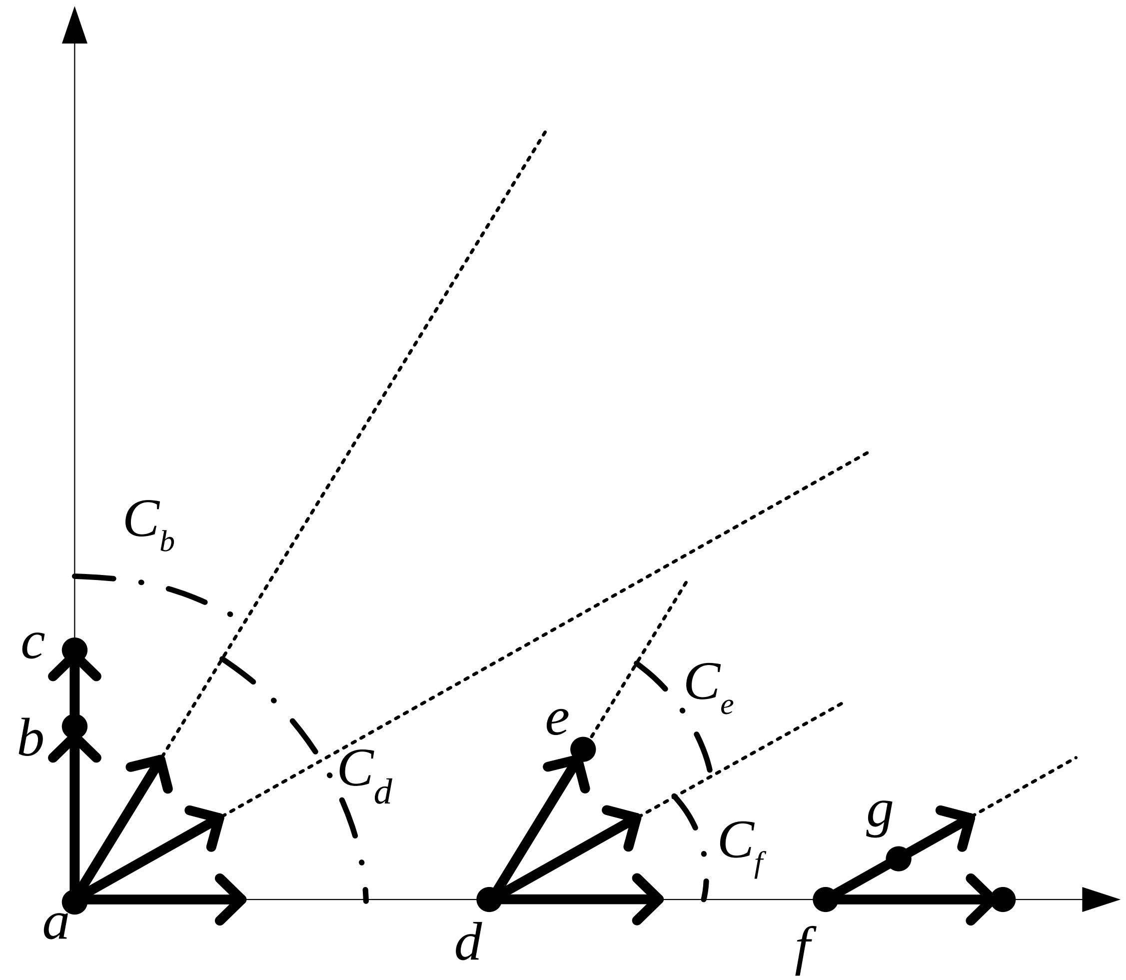
There are more efficient combinatorial algorithms for embedding metric distances defined on a tree!





Embedding through Spherical Coding



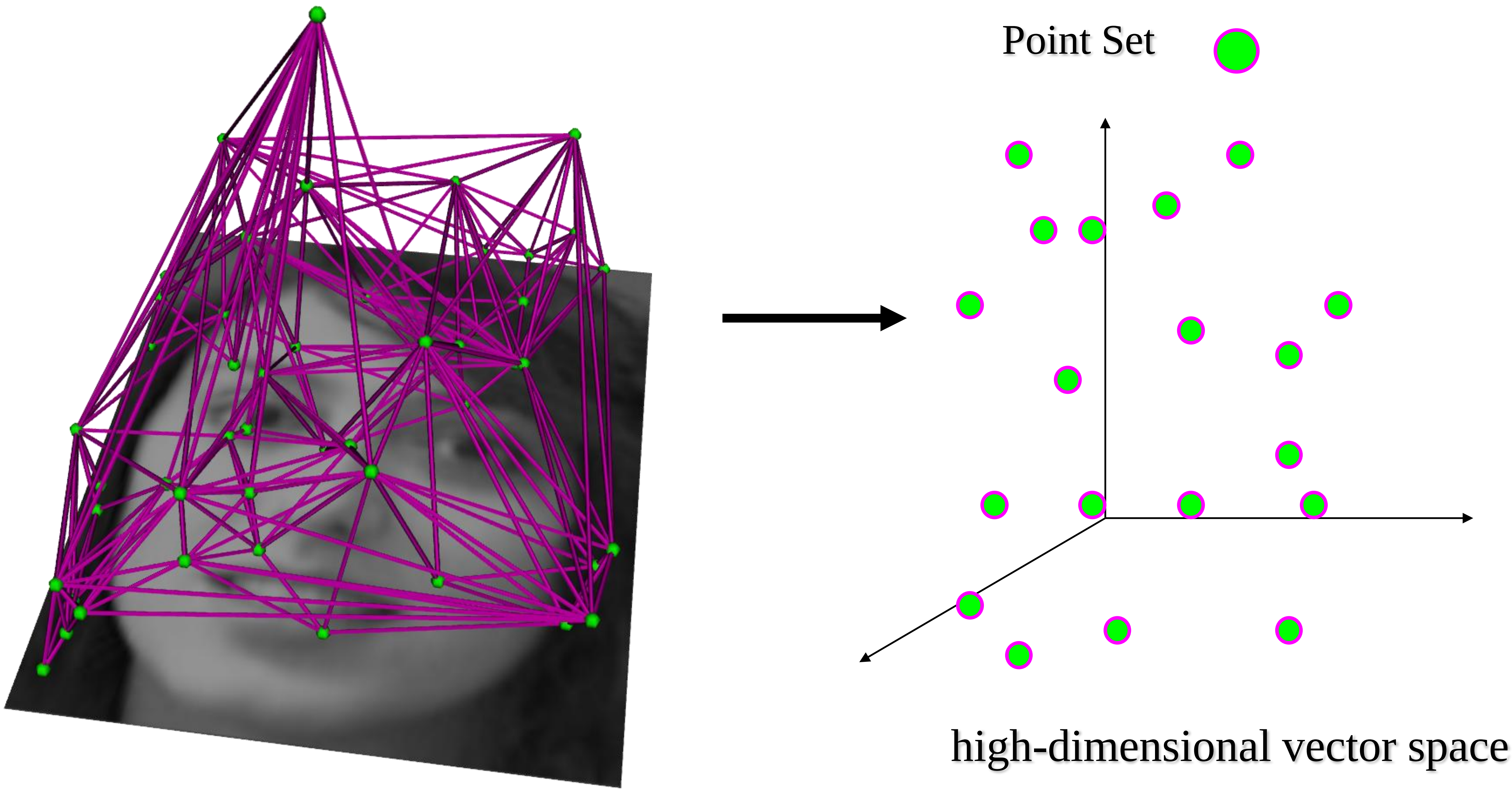


a(0,0), b(0,1.0), c(0,1.5), d(2.0,0) , e(2.5,0.87), f(3.5,0) , g(3.93,0.25) , and h(4.5,0)

Graph-Dependent Embedding

- Problems :
  - Dimensionality of the embedding is graph dependent!
  - Dimensionality reduction technique is required prior to matching.

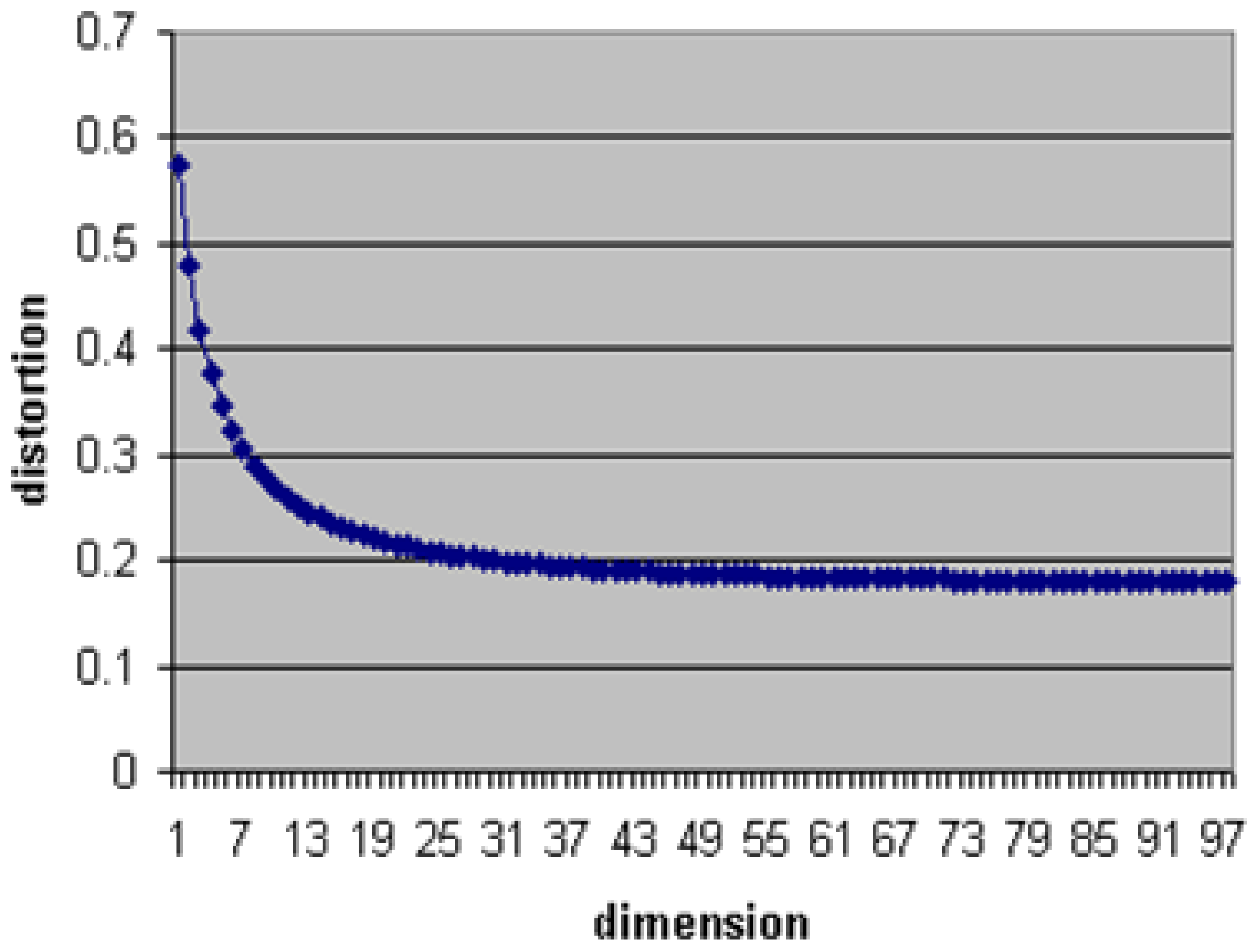
Embedding



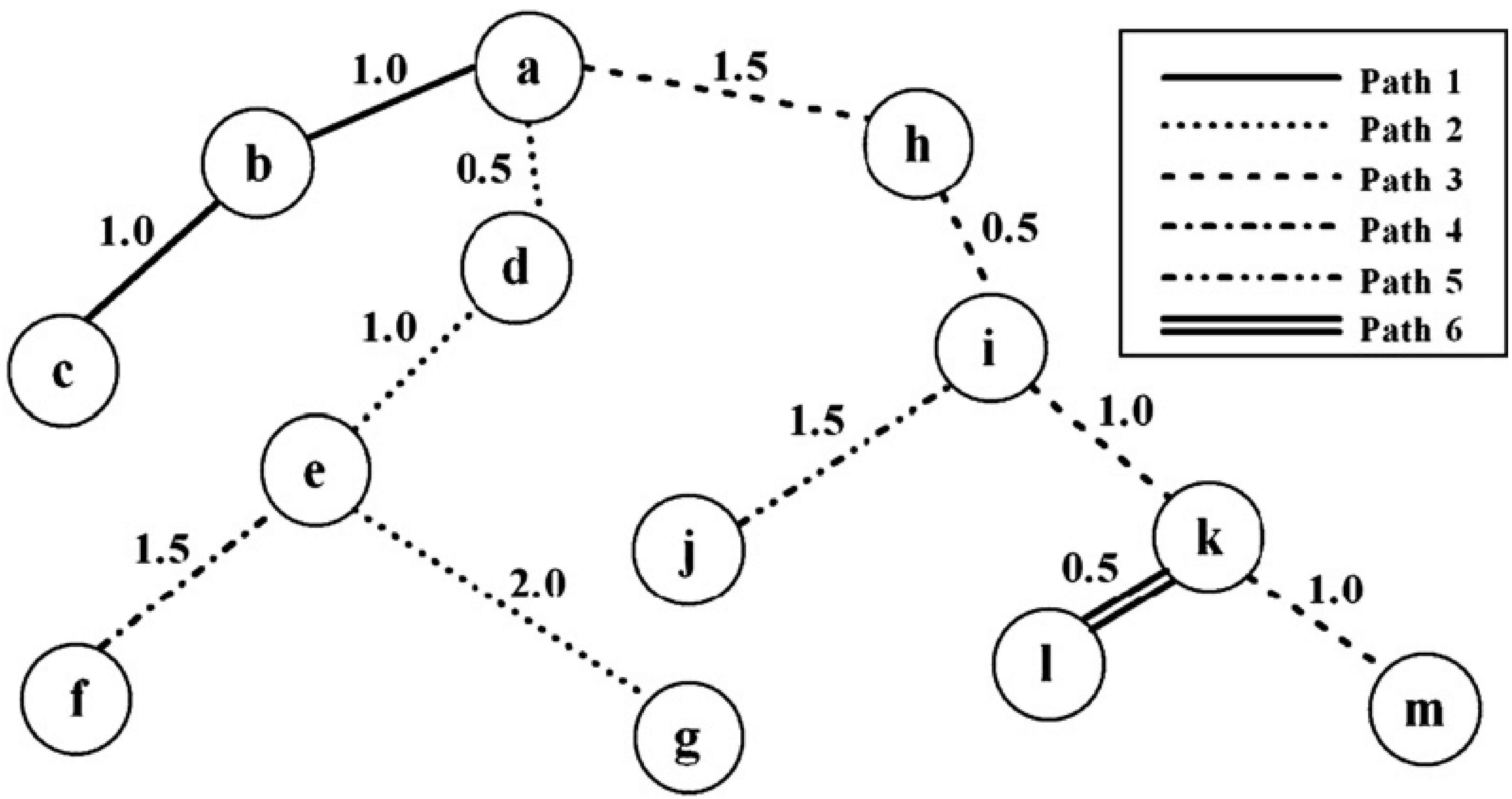
[*]Many-to-Many Feature Matching Using Spherical Coding of Directed Graphs, CVPR 2004

Dimensionality of the Embedding Space

- The distortion rapidly decreases when increasing the dimensionality in the beginning, but hardly decreases after the 30th dimension.

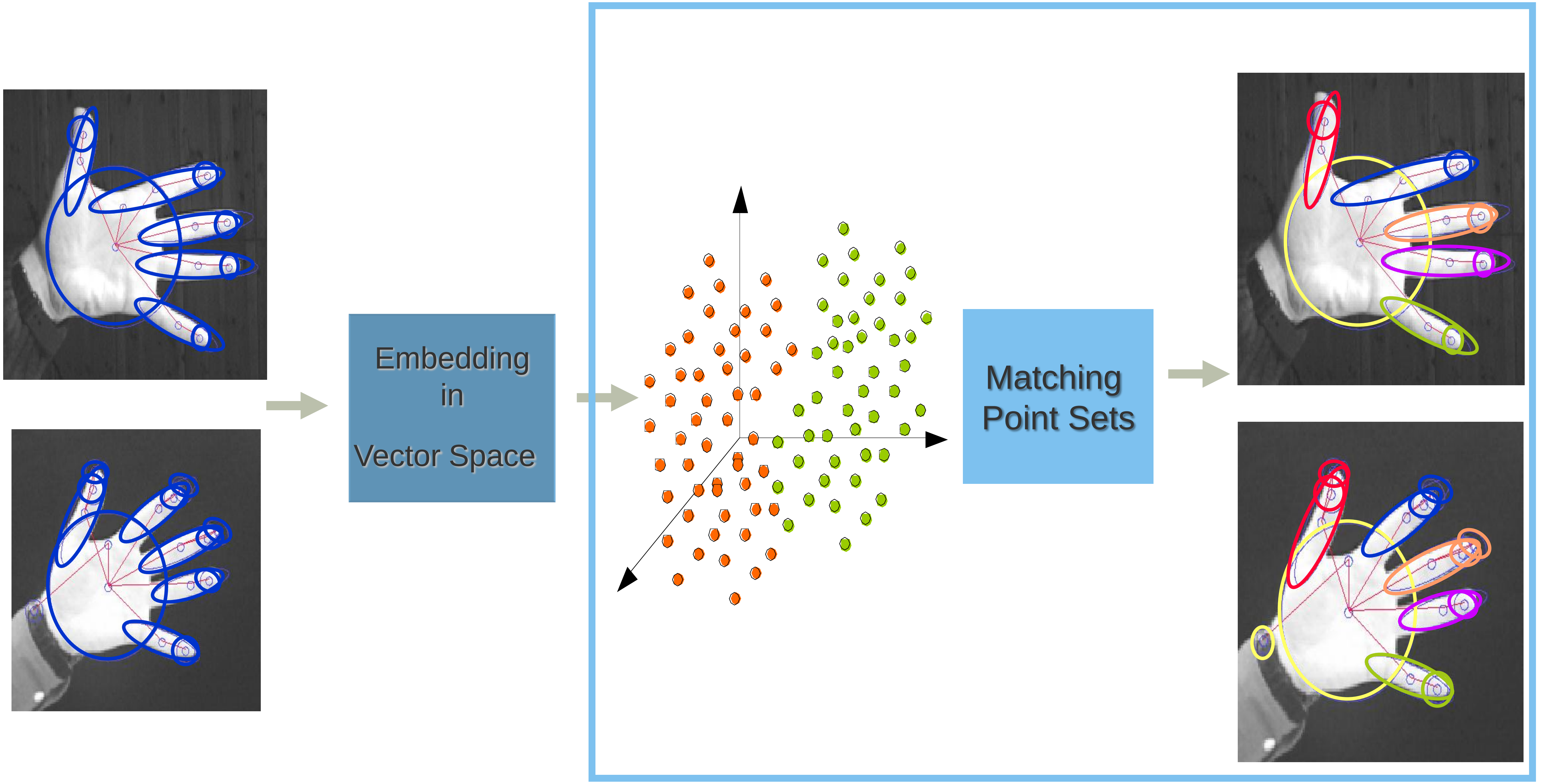


Caterpillar Decomposition under l_1

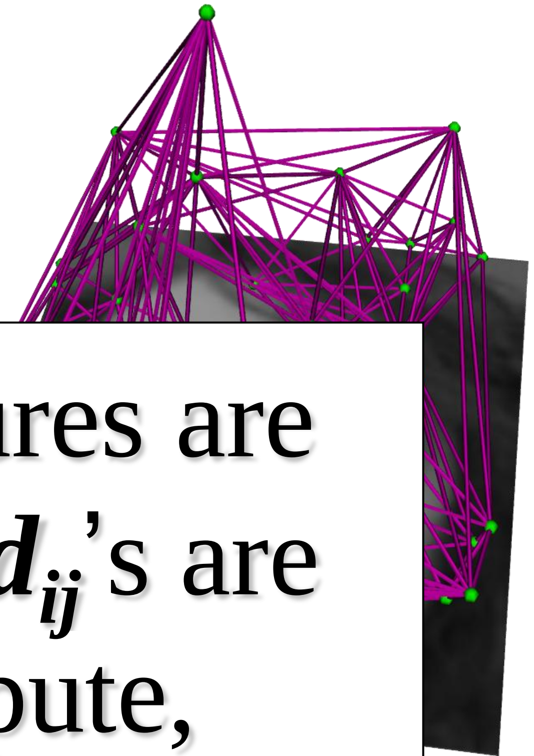


M. F. Demirci, Y. Osmanoglou, A. Shokoufandeh, and S. Dickinson. cviu 2011

Proposed Many-to-Many Approach



- Either the features are in a space that d_{ij}'s are easy to compute,
- Or, will be embedded to a vector space.



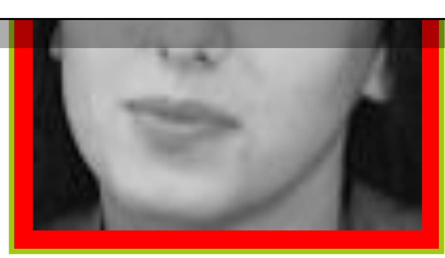
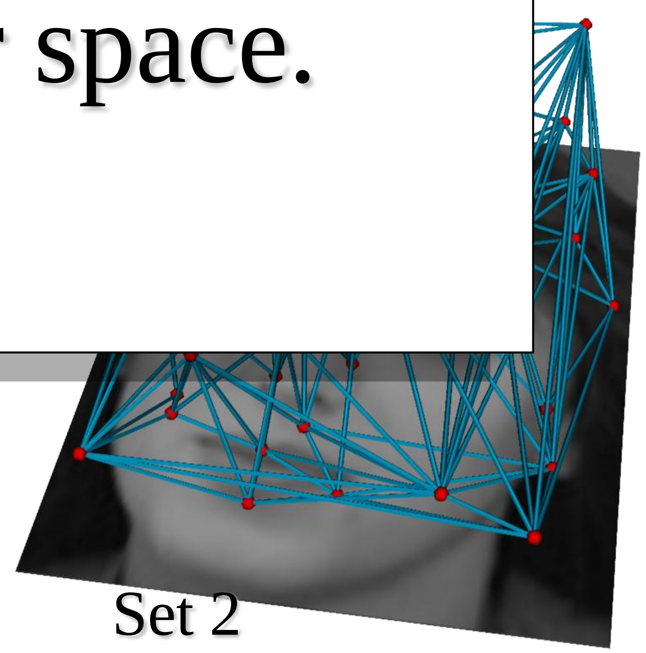
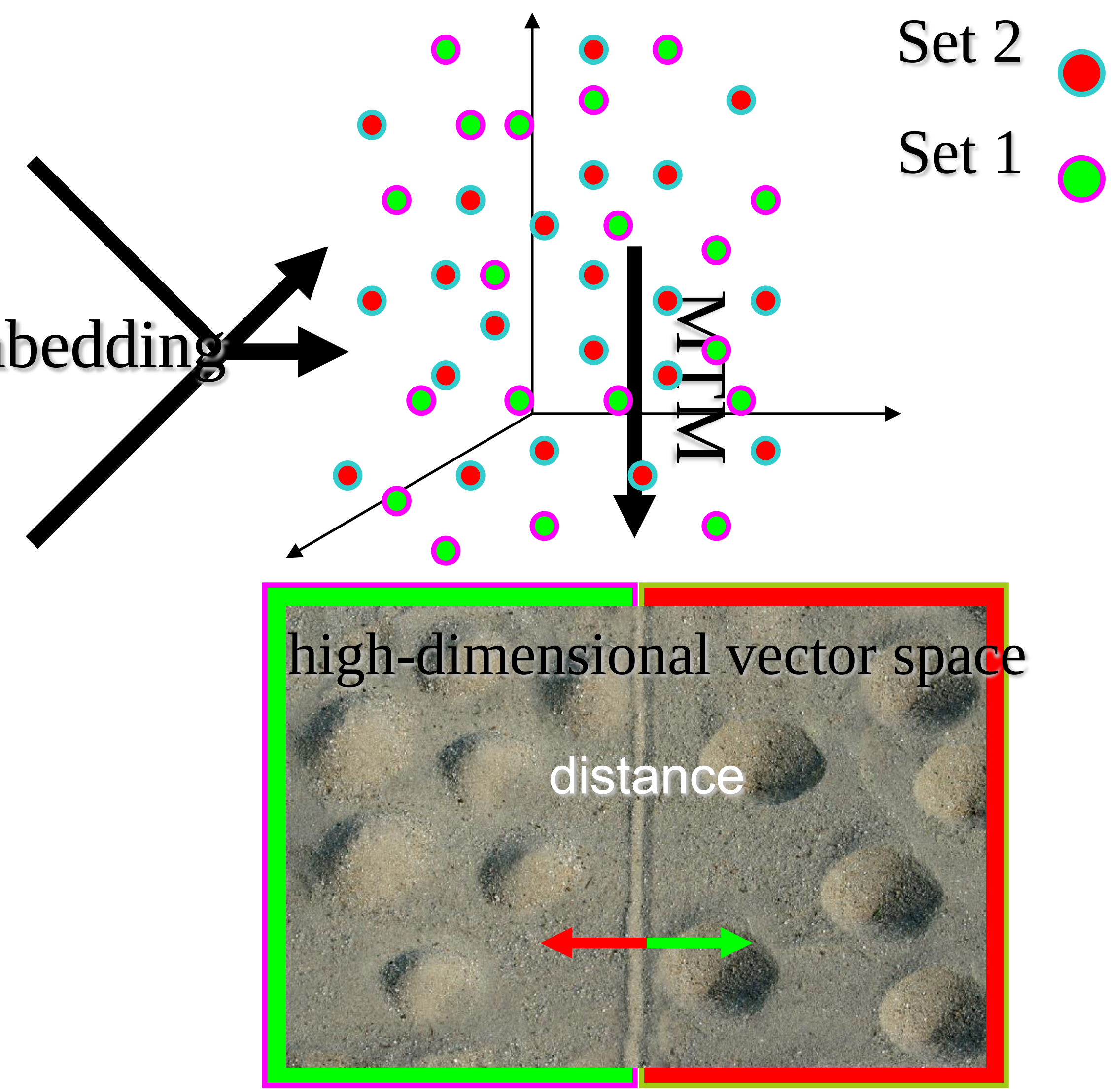


Image 2



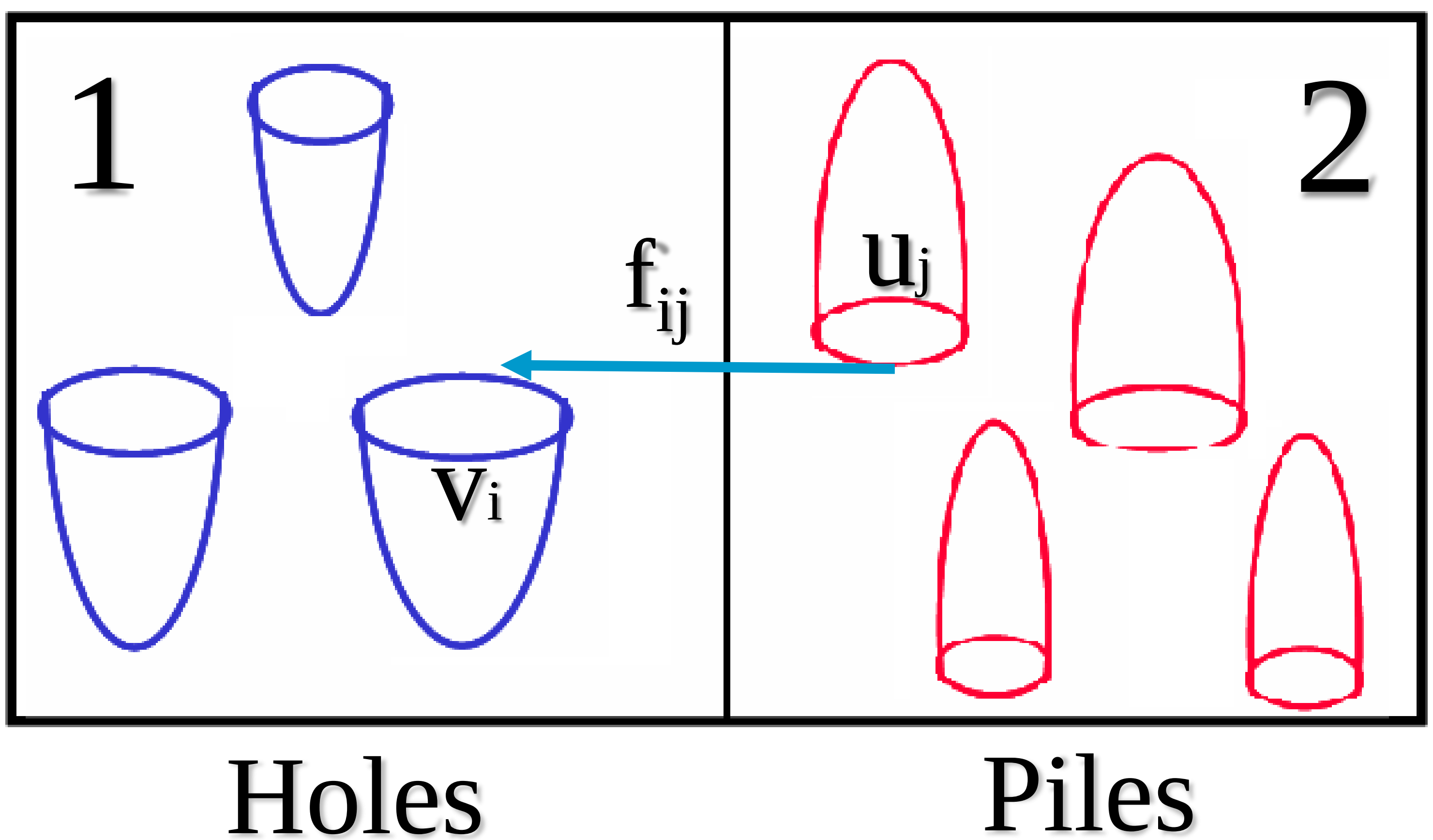
Set 2



Many-to-Many Feature Matching Using Spherical Coding of Directed Graphs, Demirci et. al. ECCV2004

Matching Distributions

- The Earth Mover's Distance (EMD)
  - Distance between two point sets.
  - Many-to-many point correspondences



[Rubner et al., IJCV, 2000].

Matching – Mathematical Formulation

- Let $\boldsymbol{P}=\{(\boldsymbol{p_1},\boldsymbol{w_{p_1}}),\ldots,(\boldsymbol{p_m},\boldsymbol{w_{pm}})\}$ and $\boldsymbol{Q}=\{(\boldsymbol{q_1},\boldsymbol{w_{q_1}}),\ldots(\boldsymbol{q_n},\boldsymbol{w_{qn}})\}$ be distributions.

Matching – Mathematical Formulation

- Let $\boldsymbol{P}=\{(\boldsymbol{p_1},\boldsymbol{w_{p_1}}),\ldots,(\boldsymbol{p_m},\boldsymbol{w_{pm}})\}$ and $\boldsymbol{Q}=\{(\boldsymbol{q_1},\boldsymbol{w_{q_1}}),\ldots(\boldsymbol{q_n},\boldsymbol{w_{qn}})\}$ be distributions.
- Find a flow matrix $F=[fij]$ that minimizes:

$$Work(P,Q,F)=\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}$$

Matching – Mathematical Formulation

- Let $\boldsymbol{P}=\{(\boldsymbol{p_1},\boldsymbol{w_{p1}}),\ldots,(\boldsymbol{p_m},\boldsymbol{w_{pm}})\}$ and $\boldsymbol{Q}=\{(\boldsymbol{q_1},\boldsymbol{w_{q1}}),\ldots(\boldsymbol{q_n},\boldsymbol{w_{qn}})\}$ be distributions.
- Find a flow matrix $F=[fij]$ that minimizes:

$$Work(P,Q,F)=\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}$$

- subject to:

$$f_{ij} \geq 0, 1 \leq i \leq m, 1 \leq j \leq n$$

Matching – Mathematical Formulation

- Let $\boldsymbol{P}=\{(\boldsymbol{p_1},\boldsymbol{w_{p1}}),\ldots,(\boldsymbol{p_m},\boldsymbol{w_{pm}})\}$ and $\boldsymbol{Q}=\{(\boldsymbol{q_1},\boldsymbol{w_{q1}}),\ldots(\boldsymbol{q_n},\boldsymbol{w_{qn}})\}$ be distributions.
- Find a flow matrix $F=[fij]$ that minimizes:

$$Work(P,Q,F)=\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}$$

- subject to:

$$f_{ij} \ge 0, 1 \le i \le m, 1 \le j \le n$$

$$\sum_{j=1}^{n} f_{ij} \le w_{pi}, 1 \le i \le m$$

$$\sum_{i=1}^{m} f_{ij} \le w_{qi}, 1 \le j \le n$$

Matching – Mathematical Formulation

- Let $\boldsymbol{P}=\{(\boldsymbol{p_1},\boldsymbol{w_{p1}}),\ldots,(\boldsymbol{p_m},\boldsymbol{w_{pm}})\}$ and $\boldsymbol{Q}=\{(\boldsymbol{q_1},\boldsymbol{w_{q1}}),\ldots(\boldsymbol{q_n},\boldsymbol{w_{qn}})\}$ be distributions.
- Find a flow matrix $F=[fij]$ that minimizes:

$$Work(P,Q,F)=\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}$$

- subject to:

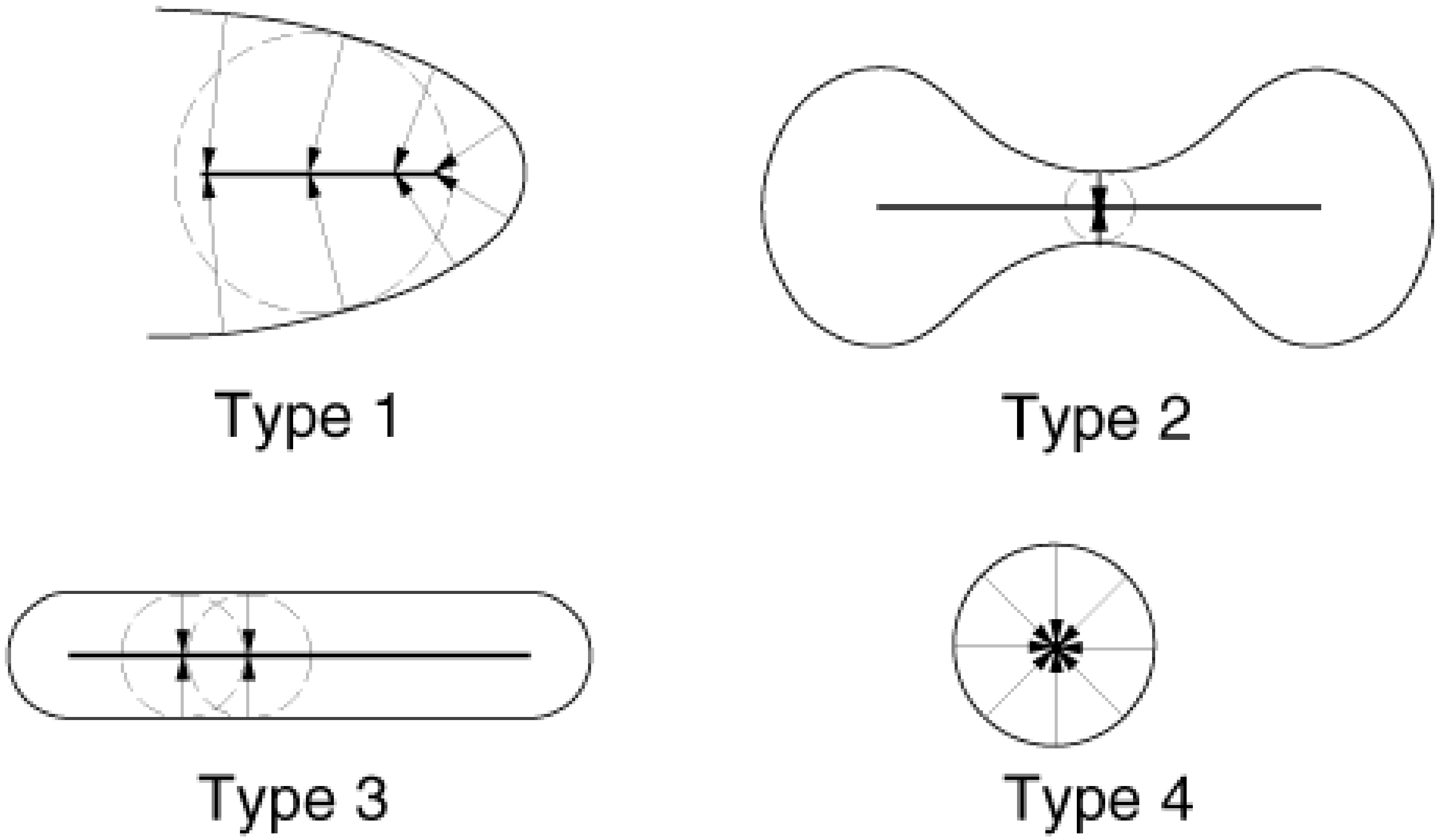
$$f_{ij} \geq 0, 1 \leq i \leq m, 1 \leq j \leq n$$

$$\sum_{j=1}^{n} f_{ij} \leq w_{pi}, 1 \leq i \leq m$$

$$\sum_{i=1}^{m} f_{ij} \leq w_{qi}, 1 \leq j \leq n$$

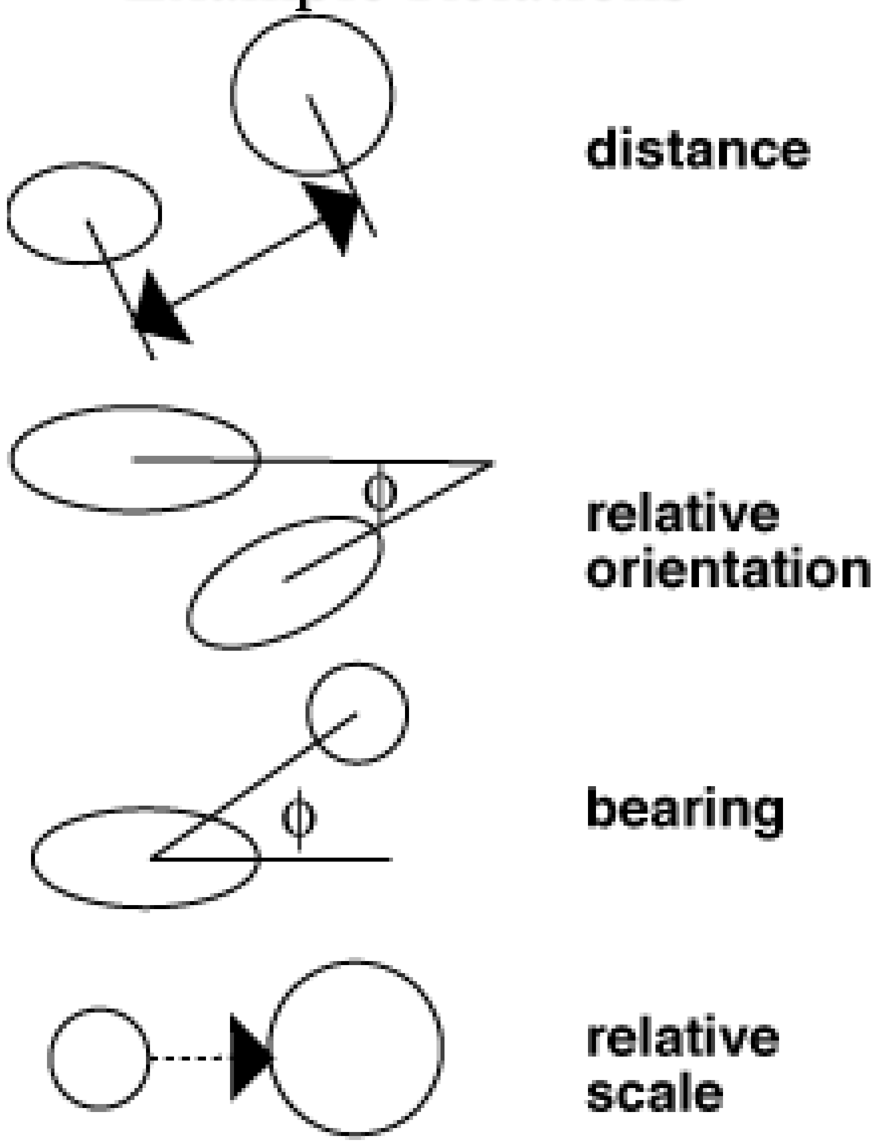
$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} w_{pi}, \sum_{j=1}^{n} w_{qj}\right)$$

Mass for Shock points

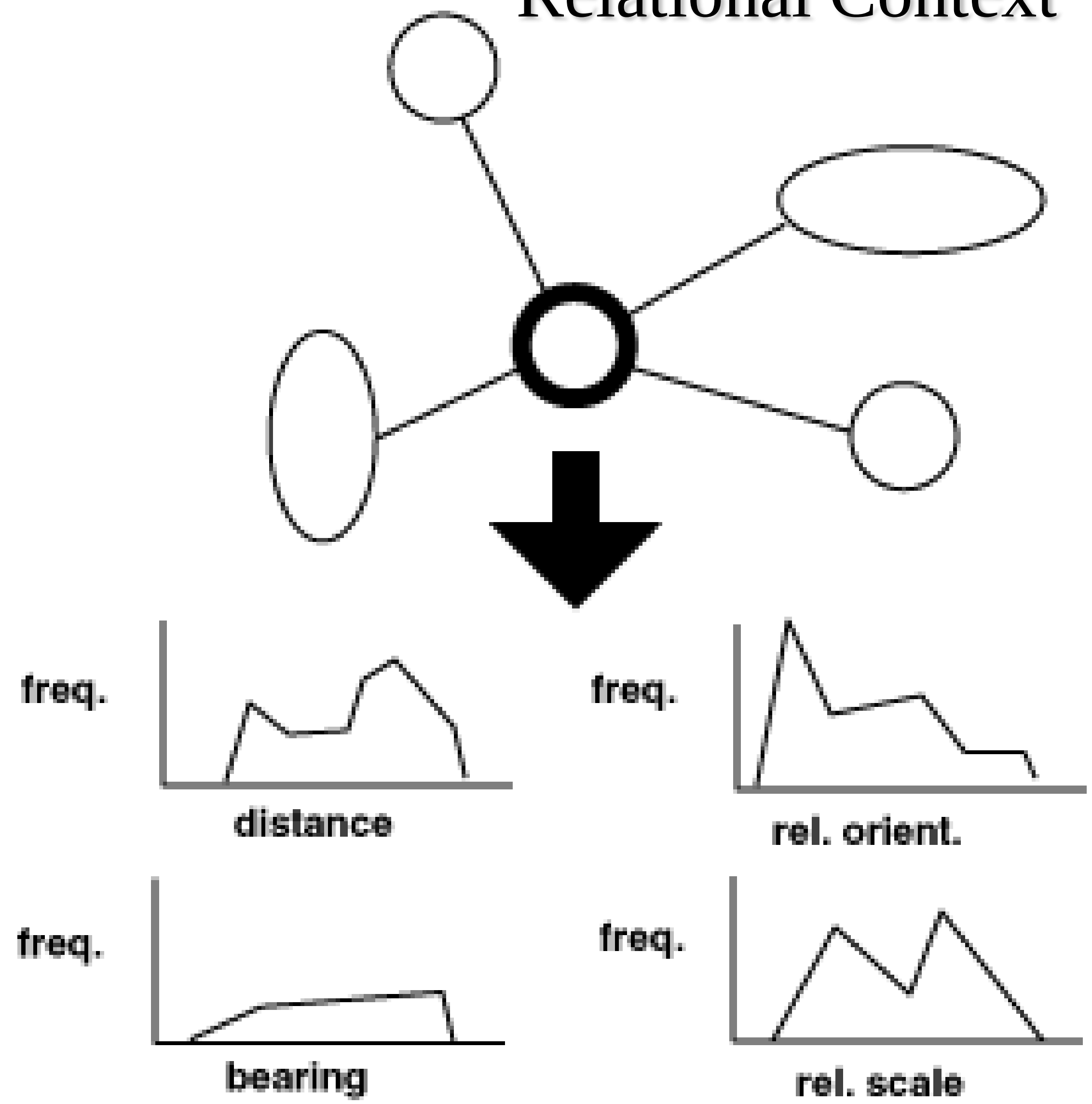


Mass fo blob and Ridge Rigons

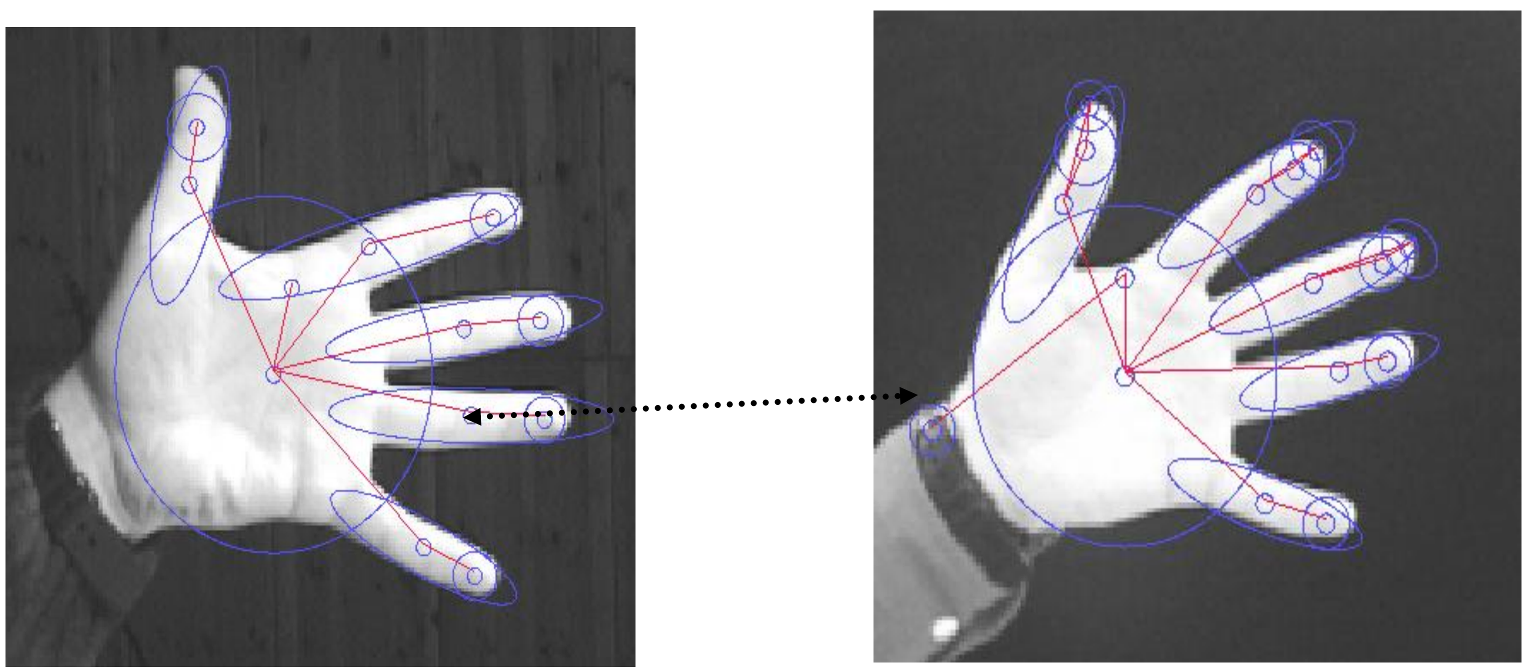
Example Relations

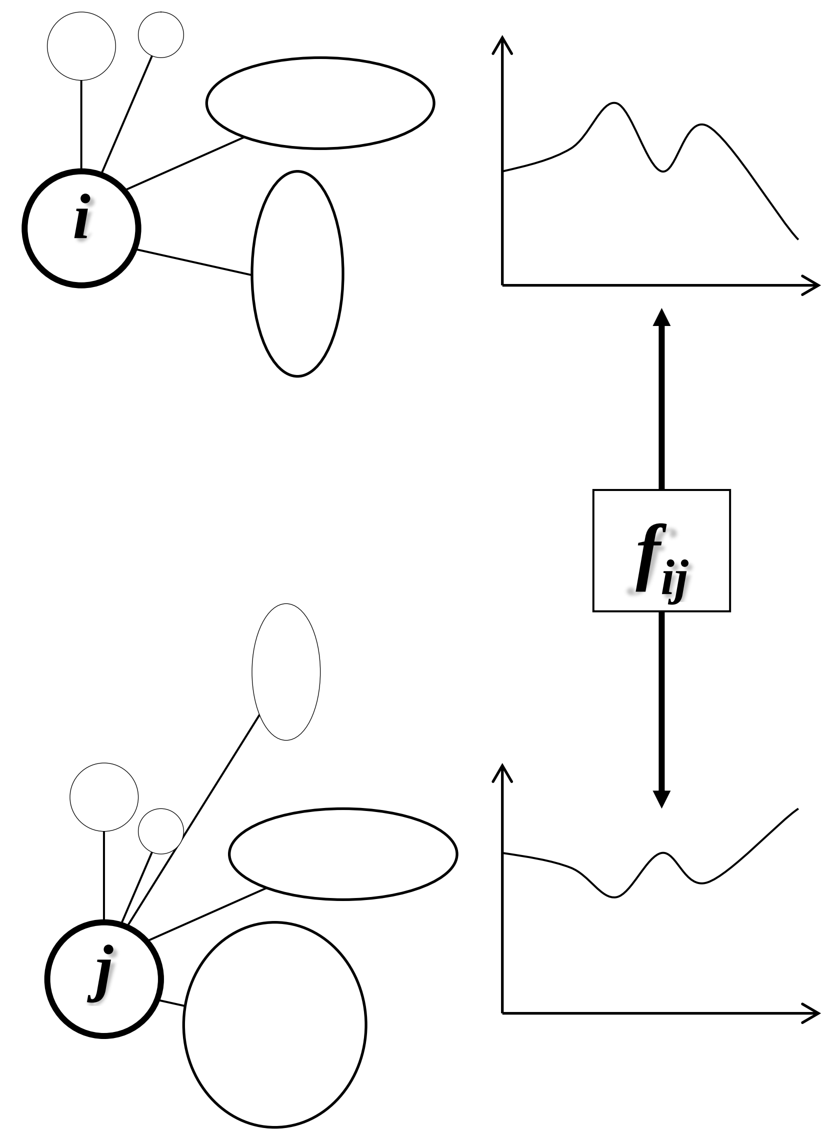


Relational Context



A simple Matching Setup





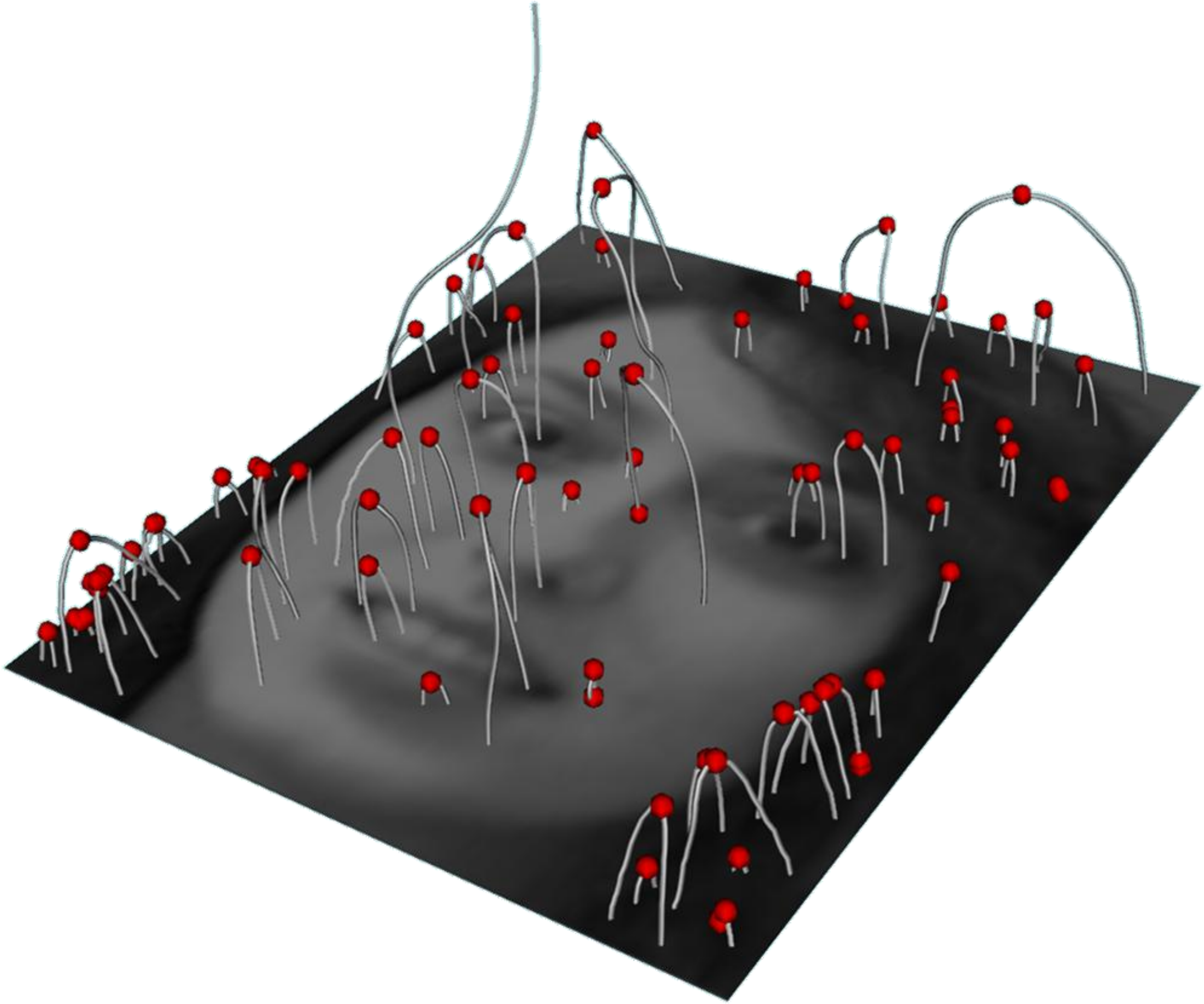
EMD under transformation

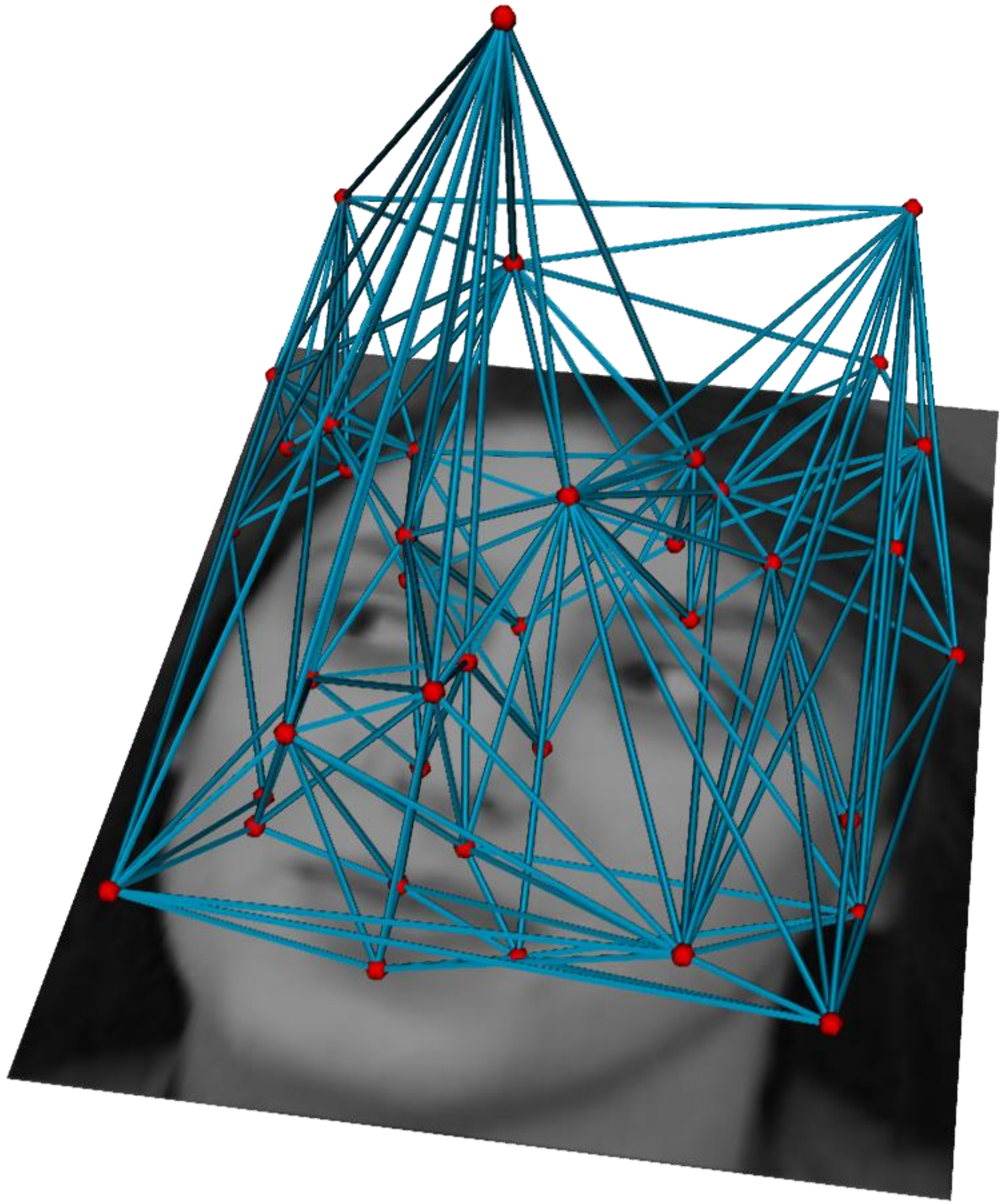
- Iterative process for an optimal **F**low and optimal **T**ransformation.
- Finding transformation for weighted point sets.
- Least Squares Estimation

[Cohen et al., ICCV, 1999].

[Umeyama, PAMI, 1991].

[Keselman, Shokoufandeh, Demirci, Dickinson, CVPR, 2003].





Mass of Top-Points

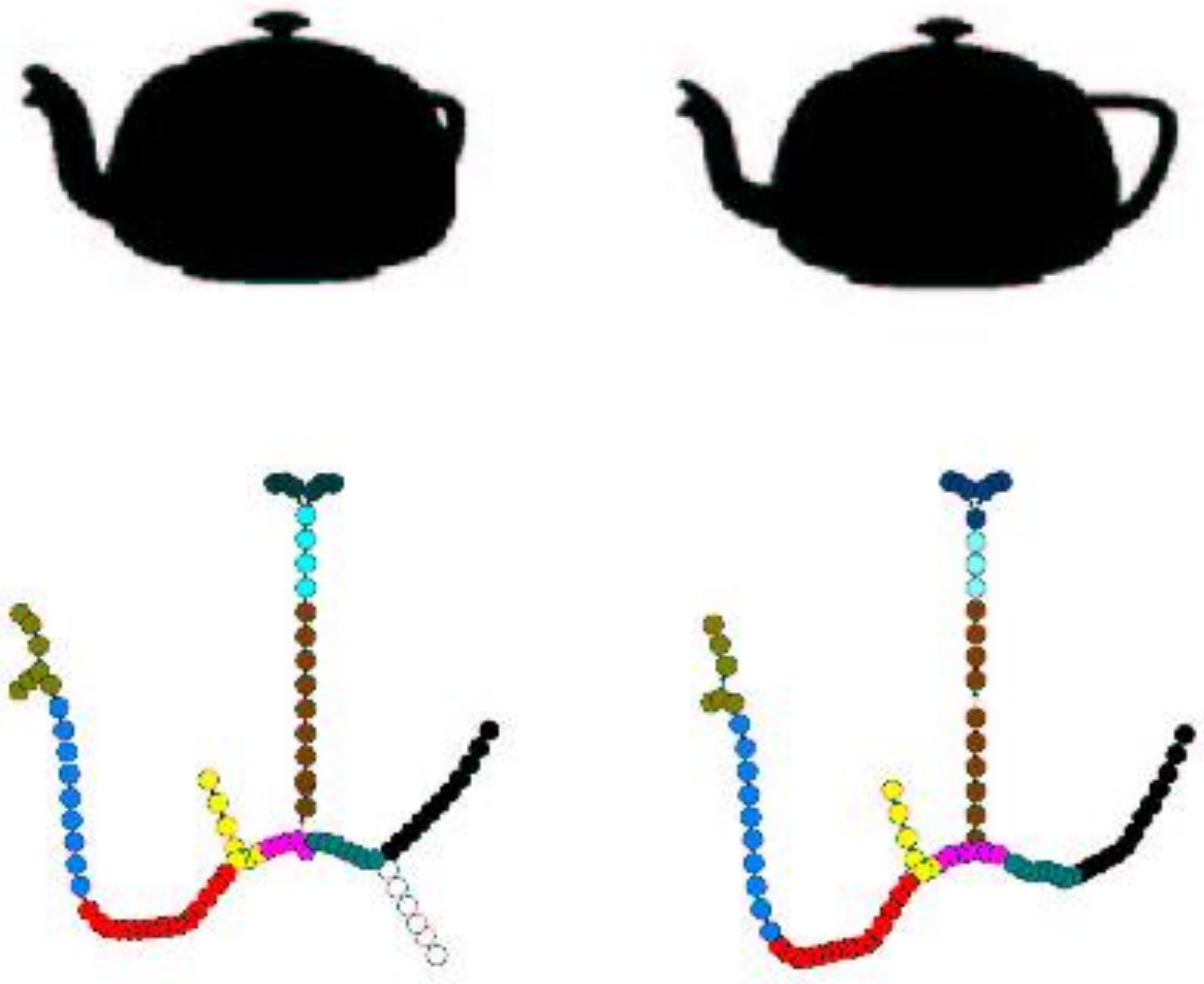
- We can calculate the variance of the displacement of top-points under noise.
- We define the mass of the top-points as:

  Exp[-(stability volume)]

- This yields a mass between 0 and 1.

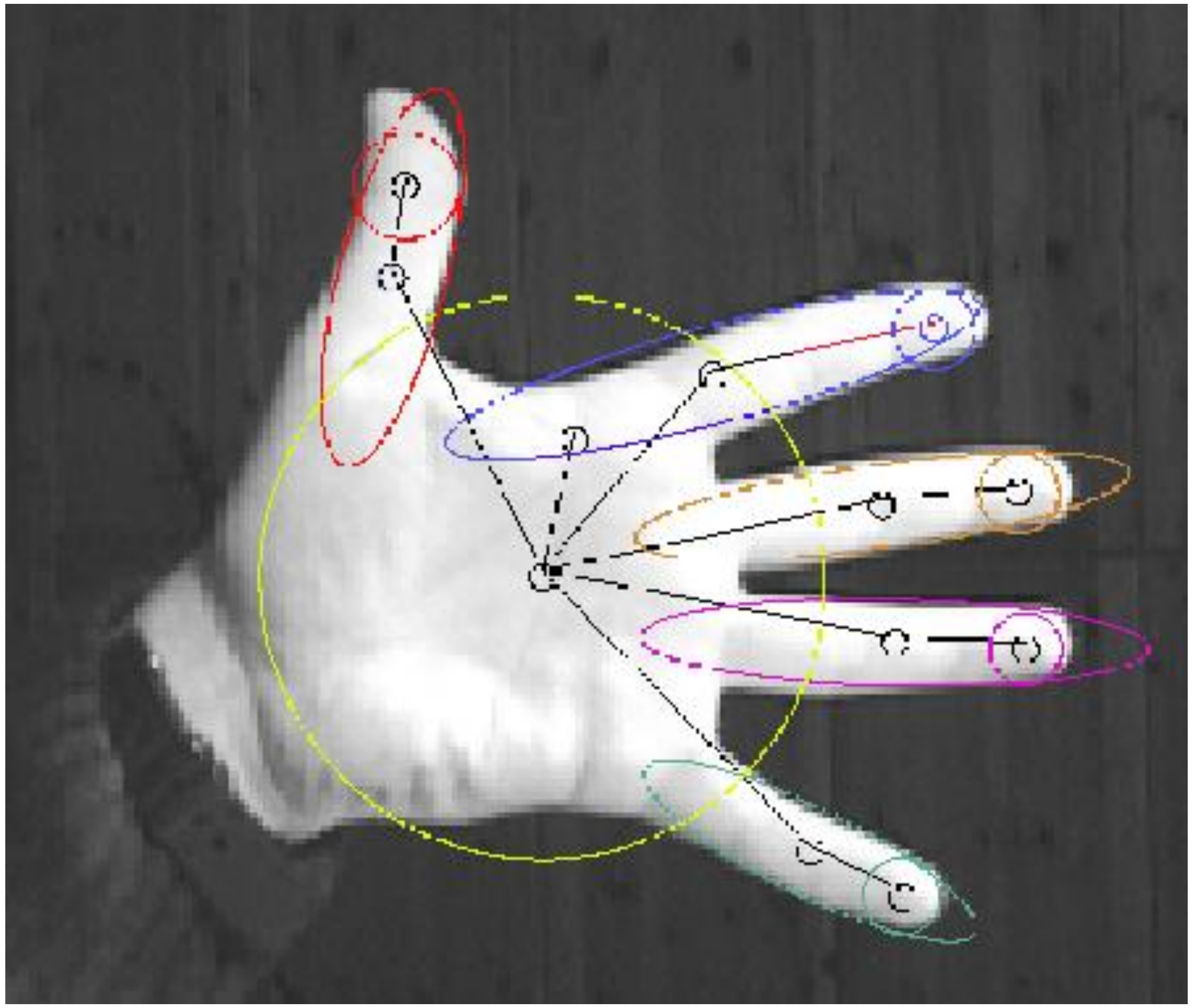
Experiments

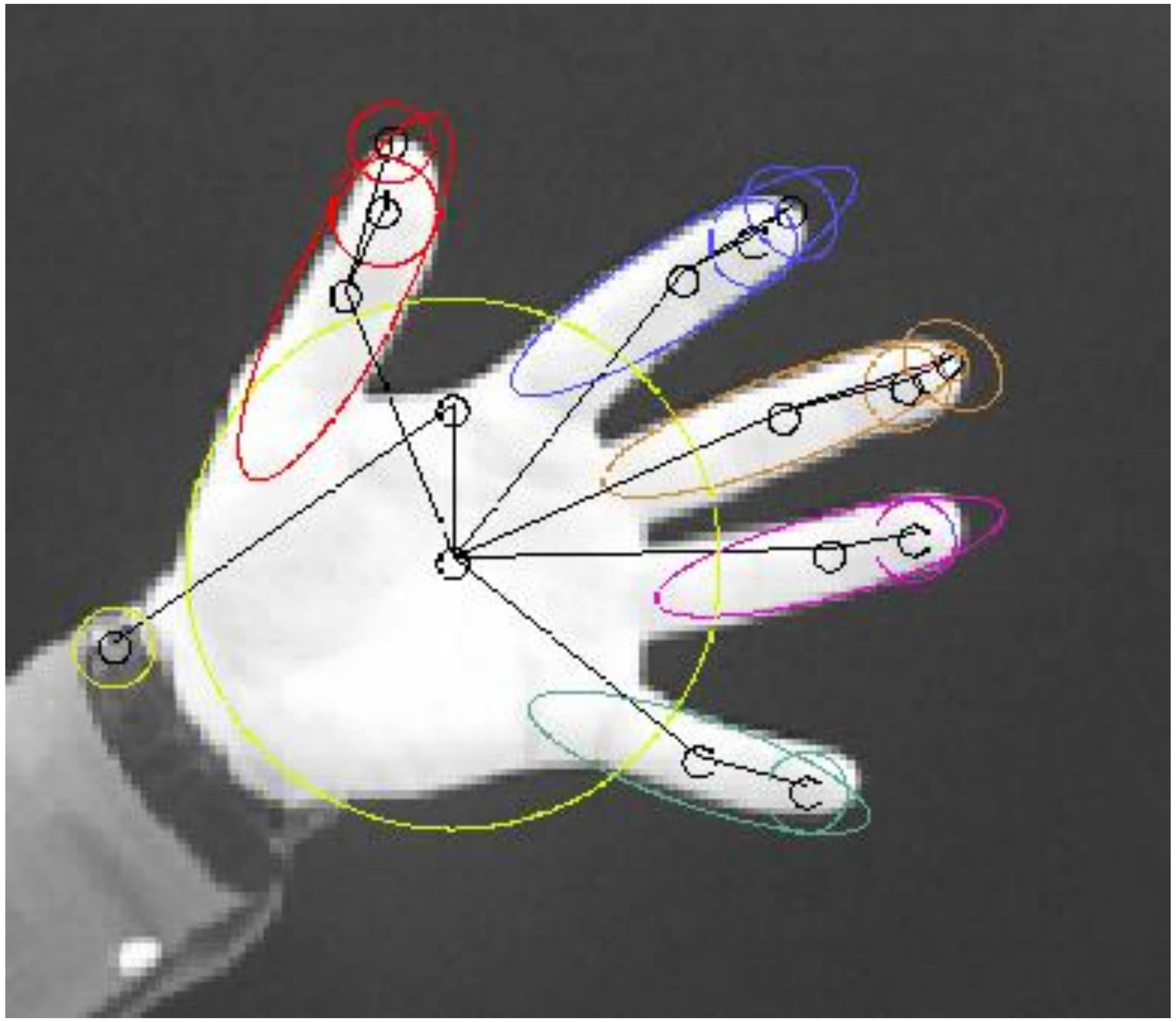
Example Correspondences



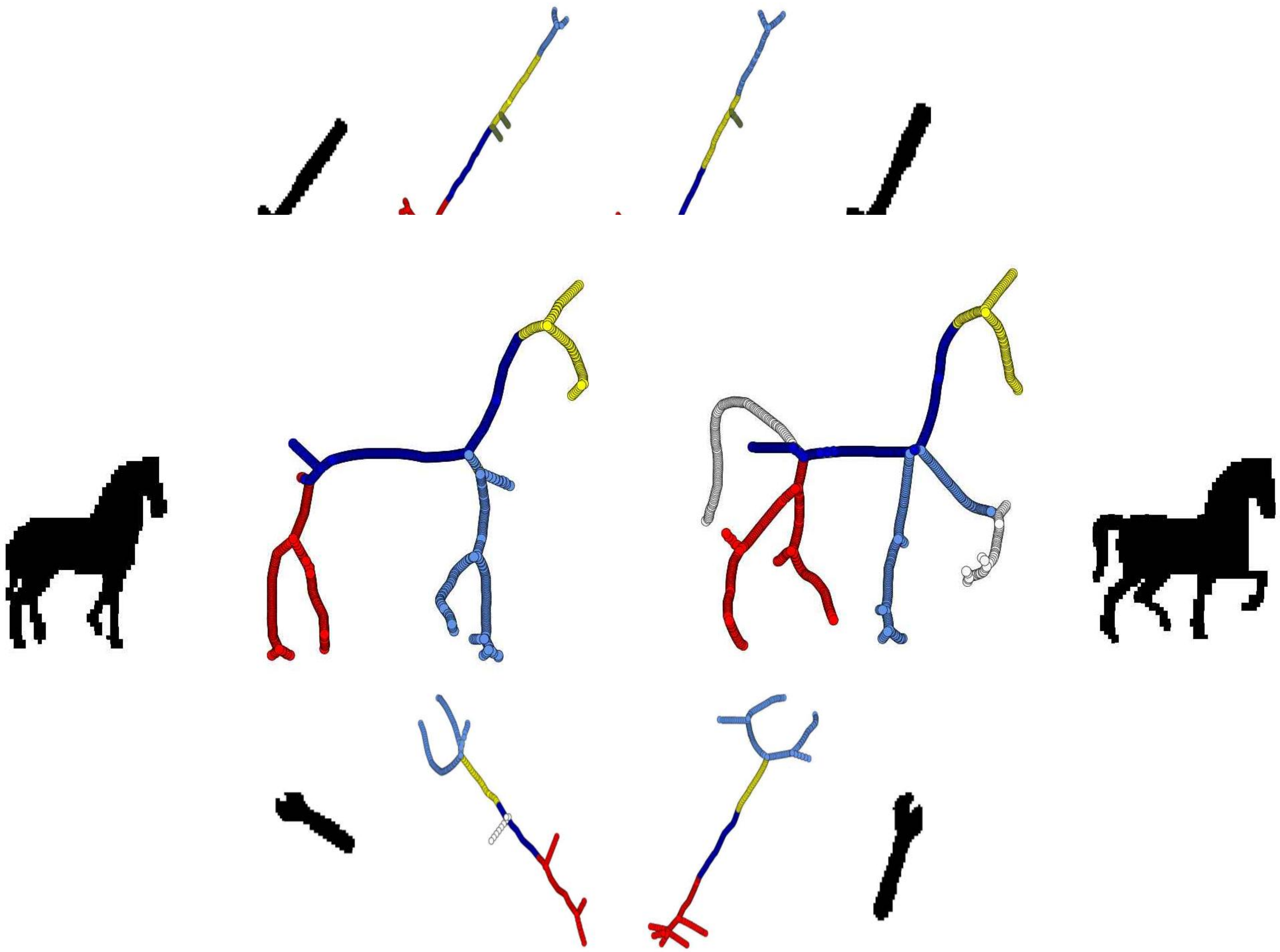
Demirci, Shokoufandeh, Dickinson, IJCV 2006

Demonstration





Results of Many-to-Many Matching



Demirci, Shokoufandeh, Dickinson, IJCV 2006

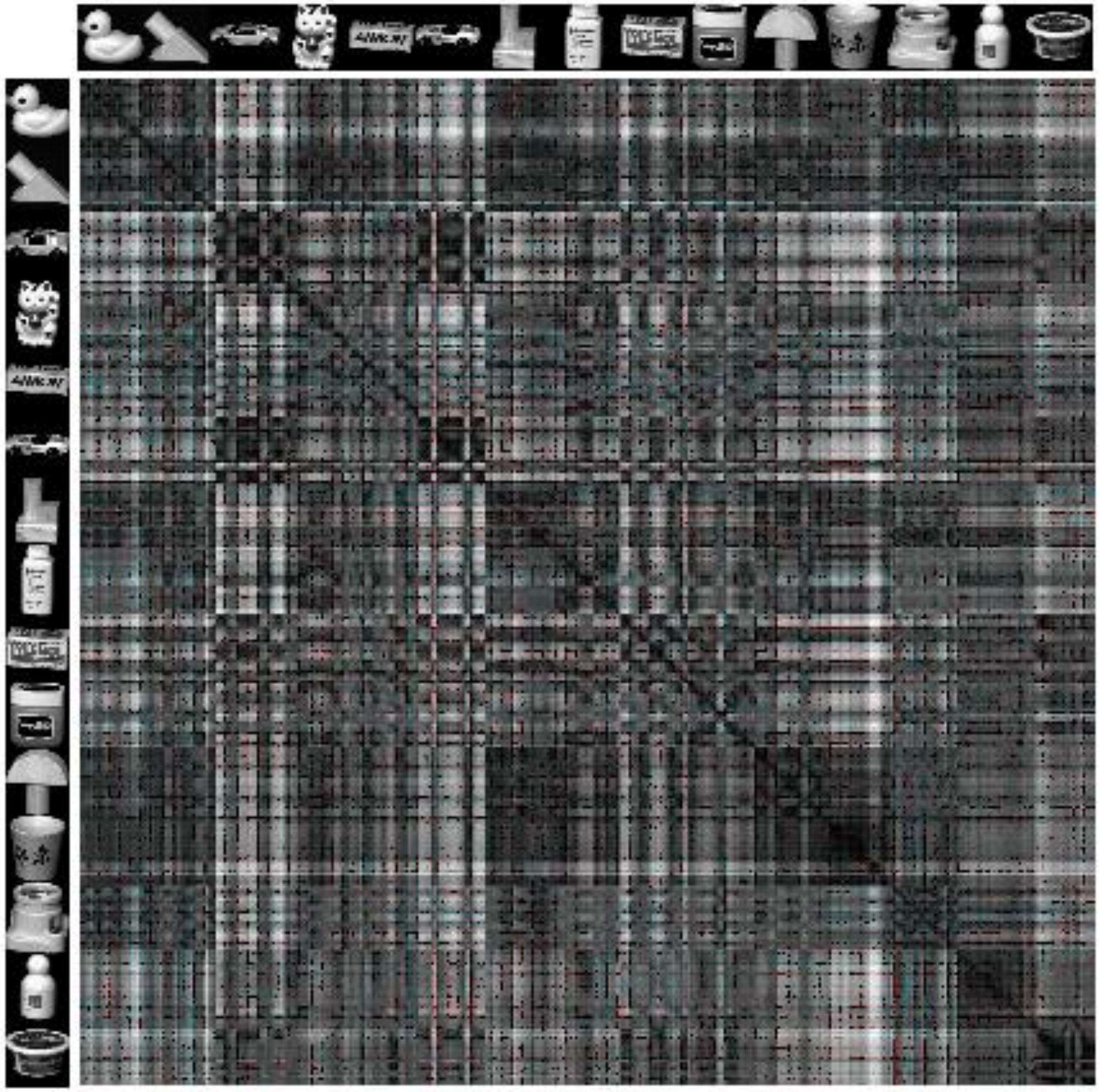
Experiments

COIL-20 (Columbia University Image Library) database consisting of 72 views per object.

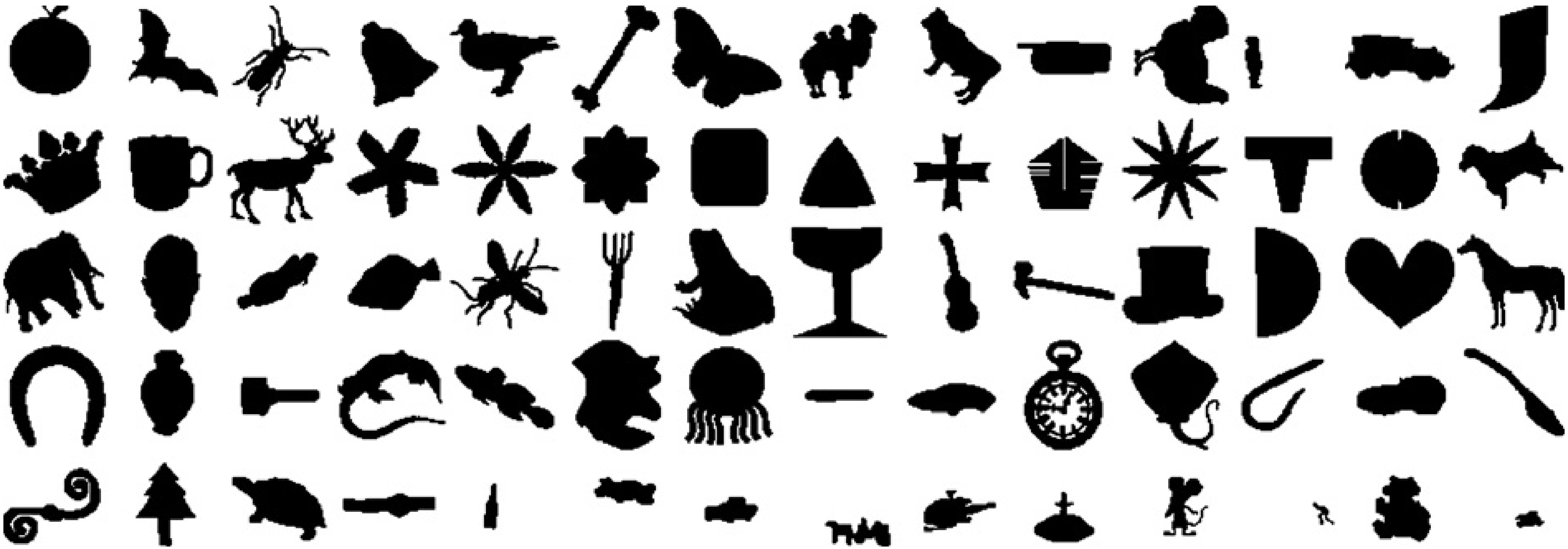
Experiments (cont'd)

- For the 72 views of each object, every second view serves as a query view, with remaining 36 views added to the database.
- Compute the distance between each query view and each database view.
- Ideally, for view i of object j, recognition trial is correct if closest view is $v_{i+1,j}$ or $v_{i-1,j}$.

- In all but 10.7% of the experiments, the closest match selected by our algorithm was a neighboring view.
- Among the mismatches, the closest view belonged to the correct object 80% of the time.
- These results ignore the effects of symmetry, and can be considered worst-case.



View-Based Object Recognition

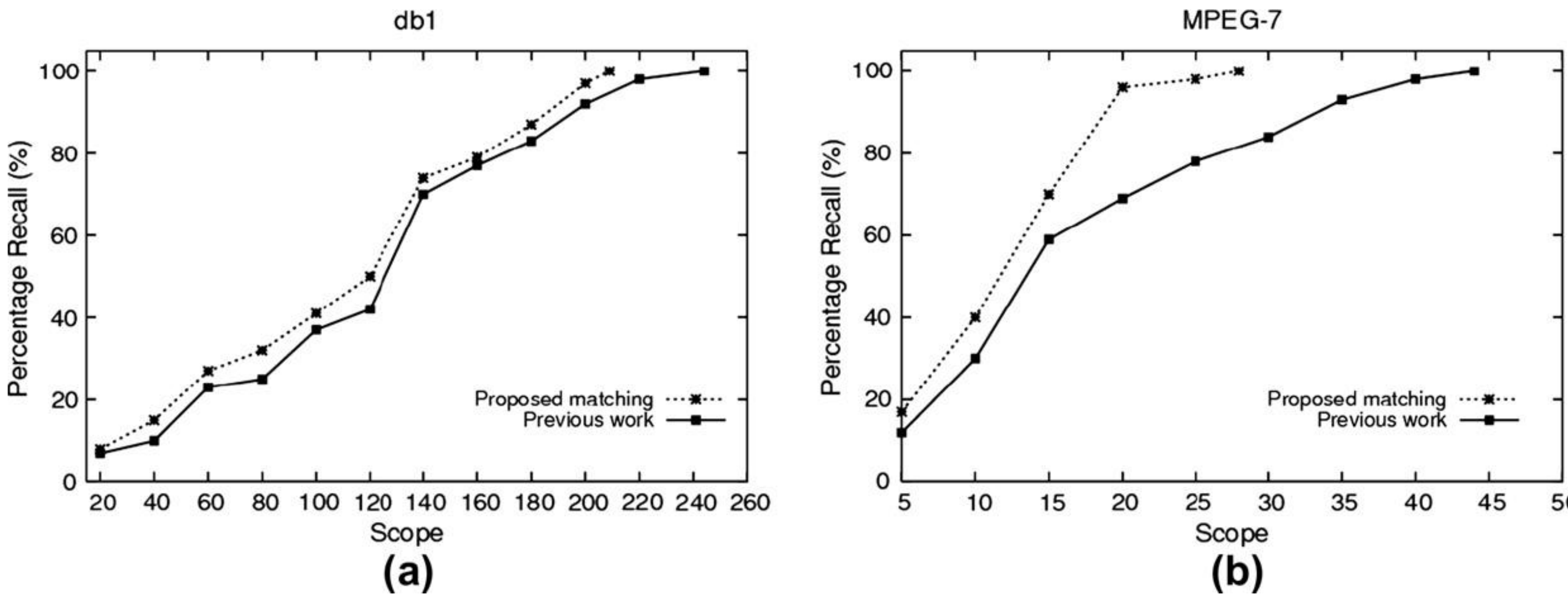


MPEG-7 (1400 SHAPES, 70 CLASSES)

View-Based Object Recognition

DB1-7 (16250 VIEWS)

Results

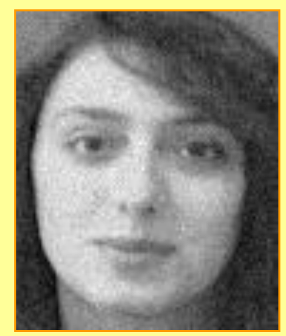


Percentage recall values of the no distortion using L1and low-distortion (baseline) techniques for Db1-7 (a) and the MPEG-7 dataset (b)

Sensitivity to Noise

- The set of query images is the database with 1%, 2%, 4%, 8% and 16% Gaussian noise added.
- Use the original database to match against.
- We count the match as correct if the distance between the perturbed image and the original image is the smallest of all images in the database.


Results of Matching under Noise

Noise	1% 	2% 	4% 	8% 	16% 
Score	97%	93%	87%	83.5%	74%

Stability under Within-class Variation

matching

Query Image

ETC.

If the graph representation is invariant to within-class deformation, then a query should match to another image from the same individual. Which is the case here.

Table 2.

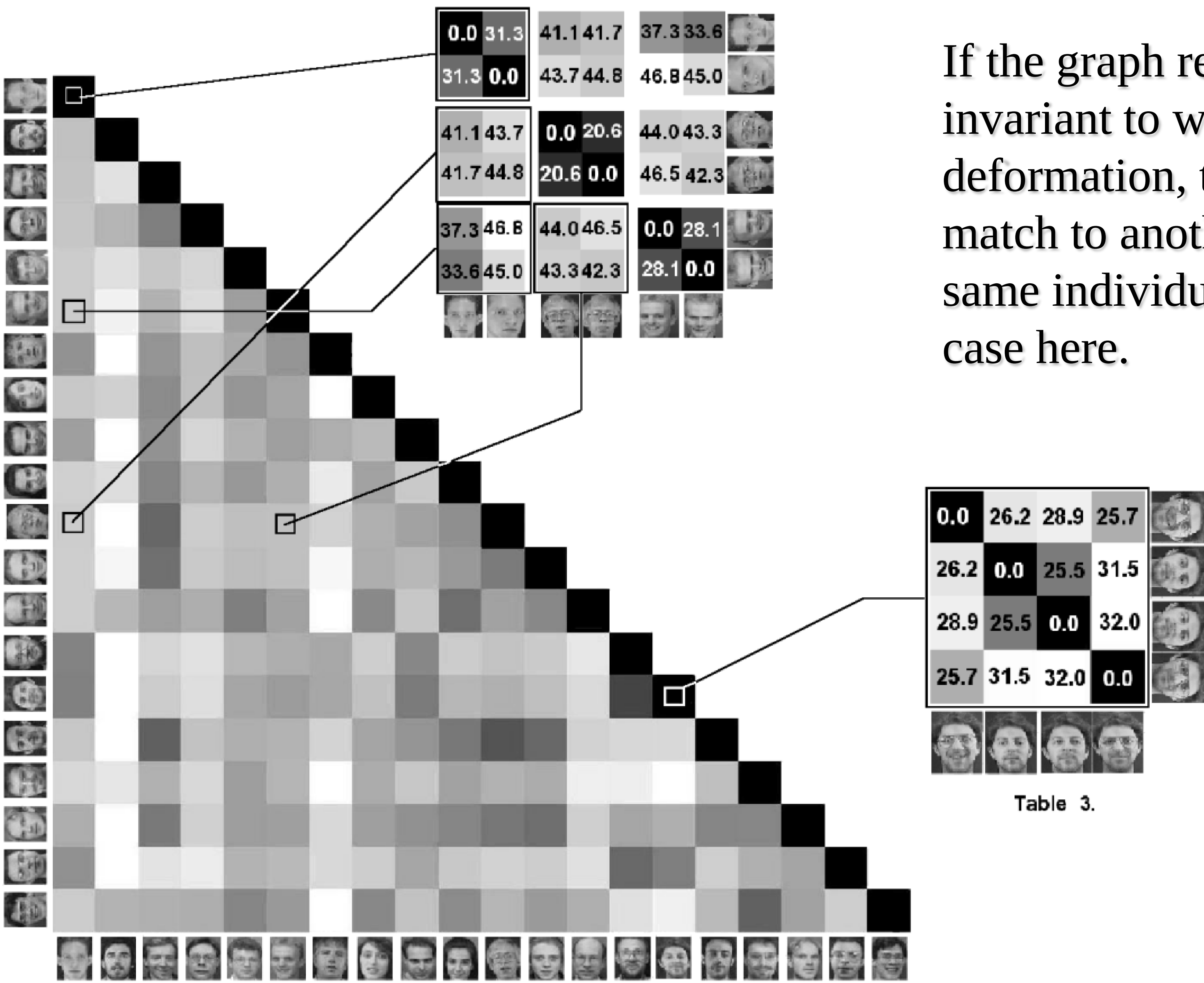


Table 3.

Table 1.

Best match

Worst match

Contributions

- Developed a general framework for many-to-many matching:
  - Proposed a deterministic variation of spherical embedding.
  - Embedding into a fixed dimension under $\boldsymbol{l_1}$.
  - Precluding the need for a dimensionality reduction process.
  - Showed that MTM matching with EMD achieves meaningful MTM matching between graph nodes.
  - Showed that directed edge information can be folded into the nodes as local histograms and can be used in the matching process.

Credits:

- Fatih Demirci (Drexel)
- Sven Dickinson (U. of Toronto)
- Yakov Keselman (Mircosoft)
- Tony Lindeberg , Lars Bretzner (KTH)
- Kaleem Siddiqi (McGill)
- Steven Zucker (Yale)
- Luc Florack (Eindhoven)
- Bram Platel

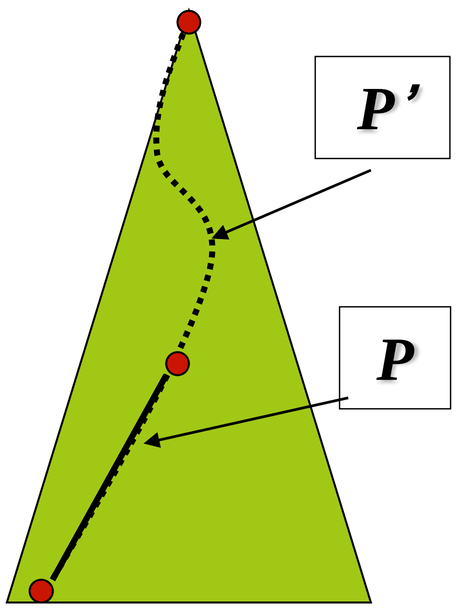
Acknowledgment:

Office of Naval Research

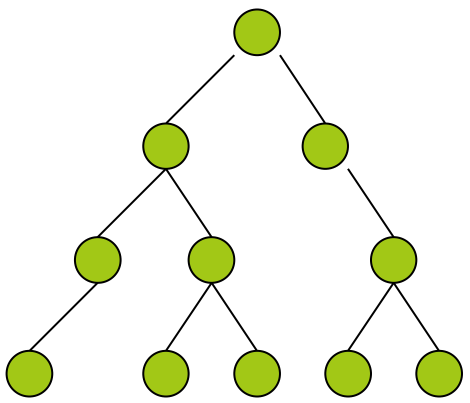
NSF

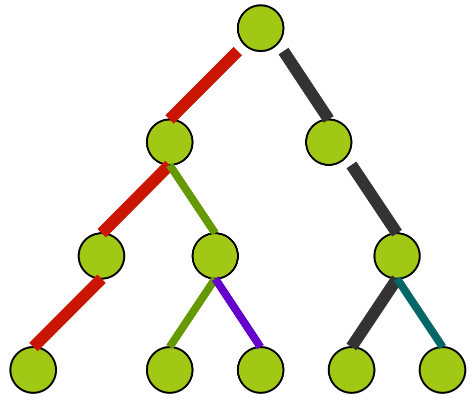
Caterpillar Decomposition:

- A monotone path ***P*** in tree ***T*** is a sunset of some root-leaf path ***P'***.



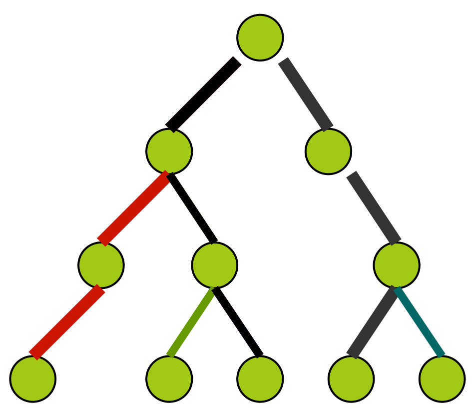
- A caterpillar decomposition is a partition $\boldsymbol{F}=\{\boldsymbol{P}_1,\ldots,\boldsymbol{P}_t\}$ of the edge set of ***T*** such that each $\boldsymbol{P}_i$ is a monotone path .





Caterpillar Dimension:

- A decomposition ***F*** has dimension ***m*** if for any root-leaf path ***P*** in ***T***, ***P*** has a nonempty intersection with at most ***m*** of P_i's.



- We will use ***cdmin***(***T***) to denote the smallest width of any decomposition of tree T, and refer to this quantity as ***caterpillar dimension***.

- The ***cdmin***(***T***) is bounded by ***O***(log ***l***(***T***)), ***l***(***T***) is the number of leaves in ***T***, and can be computed in polynomial time using dynamic programming [Matousek 1998].

Using Caterpillar Decomposition:

- The reason for large contraction for naïve embedding is that we have a large number of bends on long paths.
- The contraction is propositional to square root of number of bends.
- Let $F=\{P_1,\ldots,P_t\}$ be caterpillar decomposition of width $O(\log l(T))$, and associate a dimension u_i with the path P_i .
- For each edge $e\in E$ we will find the path P_i with $e\in P_i$, and set $v(e)=w(e)u_i$.

Distortion:

- As before the expansion will be again 1.
- Since the number of bends of any path is at most $O(\log l(T))$, the shrinkage will be at most $O((\log l(T))^{1/2})$.
- So, the distortion will be at most $O((\log l(T))^{1/2})$.
- The problem is we are using orthogonal vectors for u_i's , and this is the limit for distortion.
- Modification: Use $l(T)$ vectors with small inner products. This will decrease the distortion by a log factor.

Finally

- Assume we have already determined about the dimensionality of host space ***d*.**
- Modification: Use ***l*(*T*)** vectors with small inner products. This will decrease the distortion by a log factor
- For a given fixed ***d***, use the previous idea:
  - Find a set of ***l*(*T*)** vectors (in d dimensions) which are as far apart as they can be, and use them to embed the tree.