Shape Matching

Brandon Smith and Shengnan Wang
Computer Vision – CS766 – Fall 2007

Outline

- Introduction and Background
    - Uses of shape matching
    - Kinds of shape matching
    - Support Vector Machine (SVM)

- Matching with Shape Contexts
    - Shape Context
    - Bipartite Graph Matching
    - Modeling Transformations
    - Invariance and Robustness
    - Results

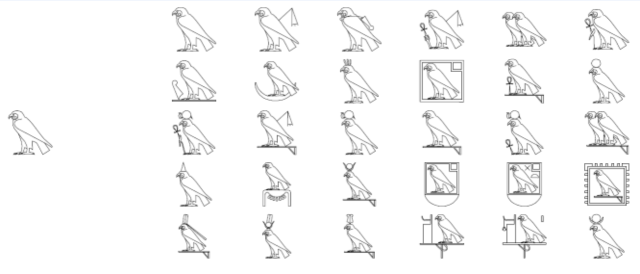
- Questions

- Shengnan’s part…

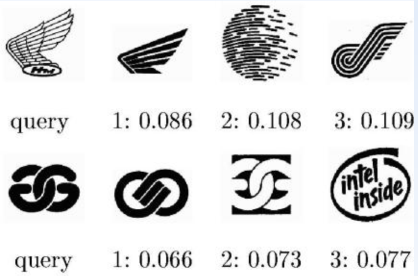
Introduction and Background

Shape matching examples

Hieroglyph Lookup

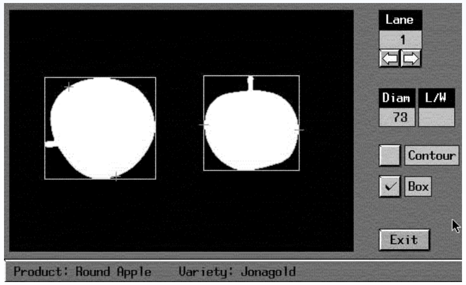


Trademark Lookup



Fingerprint Matching

Fruit Inspection



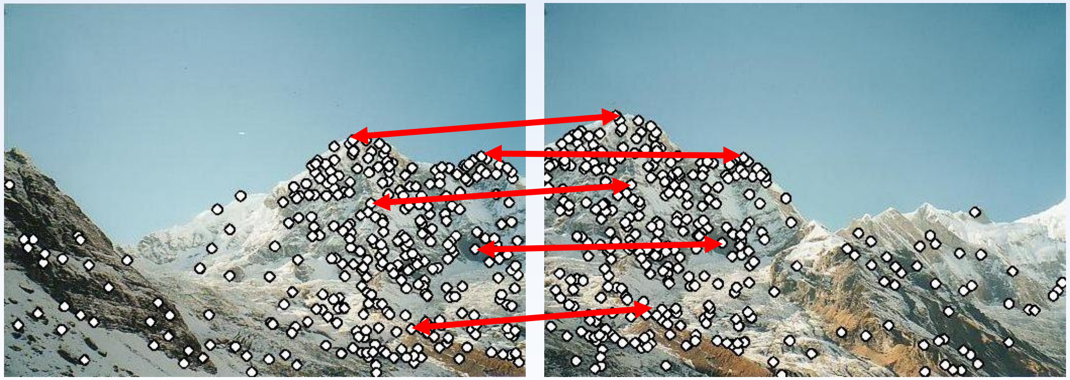
Introduction and Background

Introduction and Background

- Feature-Based Methods
- Brightness-Based Methods

Introduction and Background

Feature-Based Methods



Introduction and Background

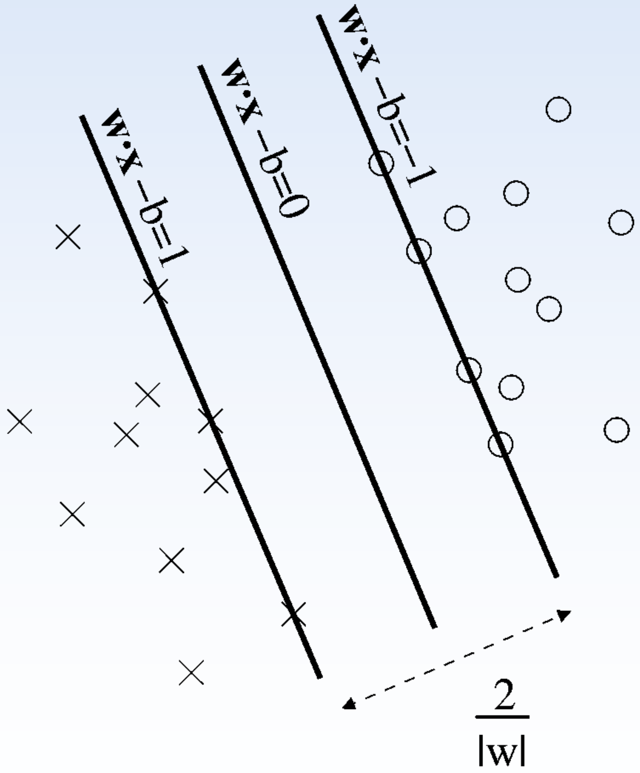
Brightness-Based Methods

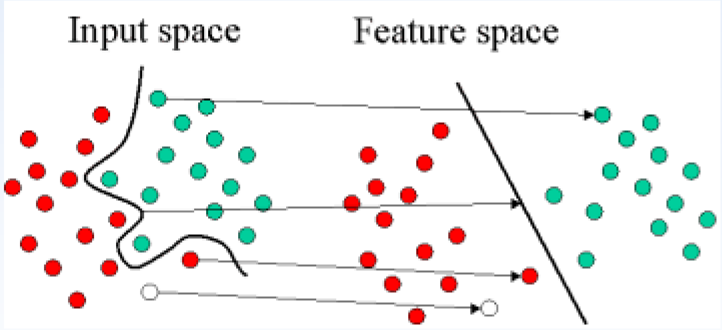
Two different frameworks:

- Explicitly find correspondences
- Build classifiers without explicitly finding correspondences.

Introduction and Background

Support Vector Machine (SVM)





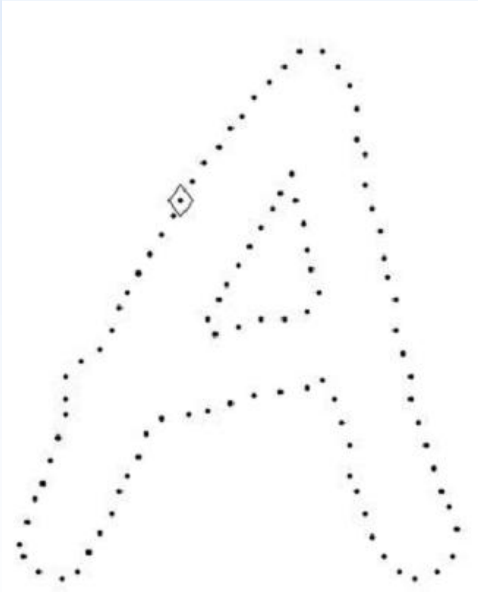
Introduction and Background

Approach:

1. Find correspondences between shapes
2. Estimate an aligning transform
3. Measure similarity

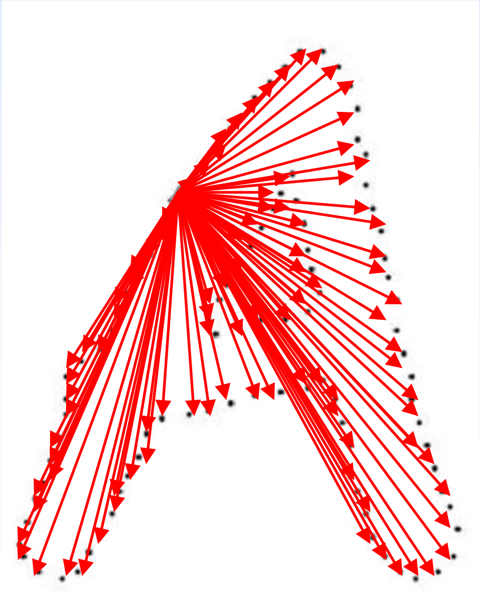
Matching with Shape Contexts

Shape Context



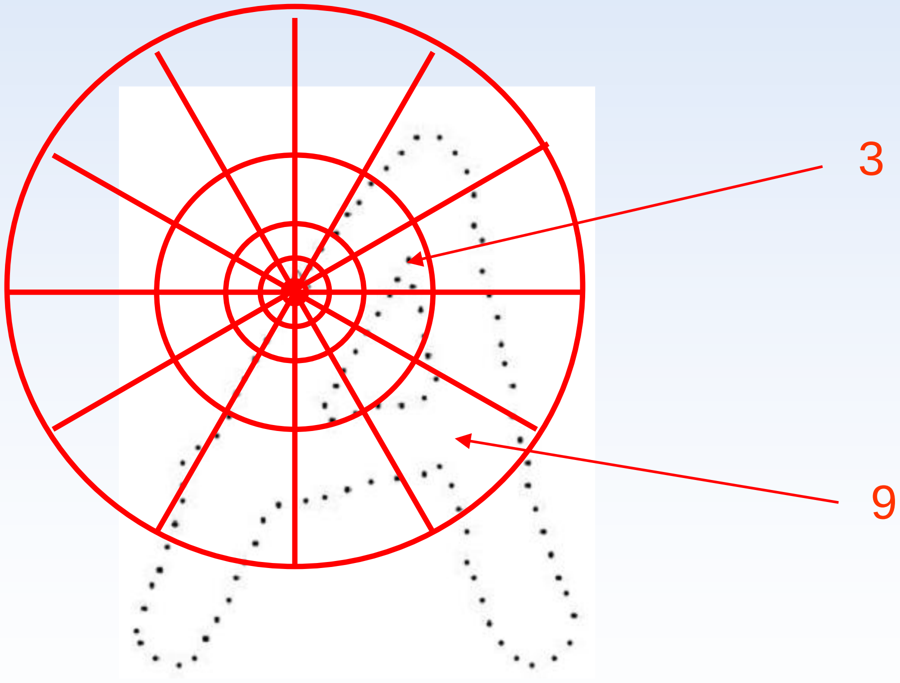
Matching with Shape Contexts

Shape Context



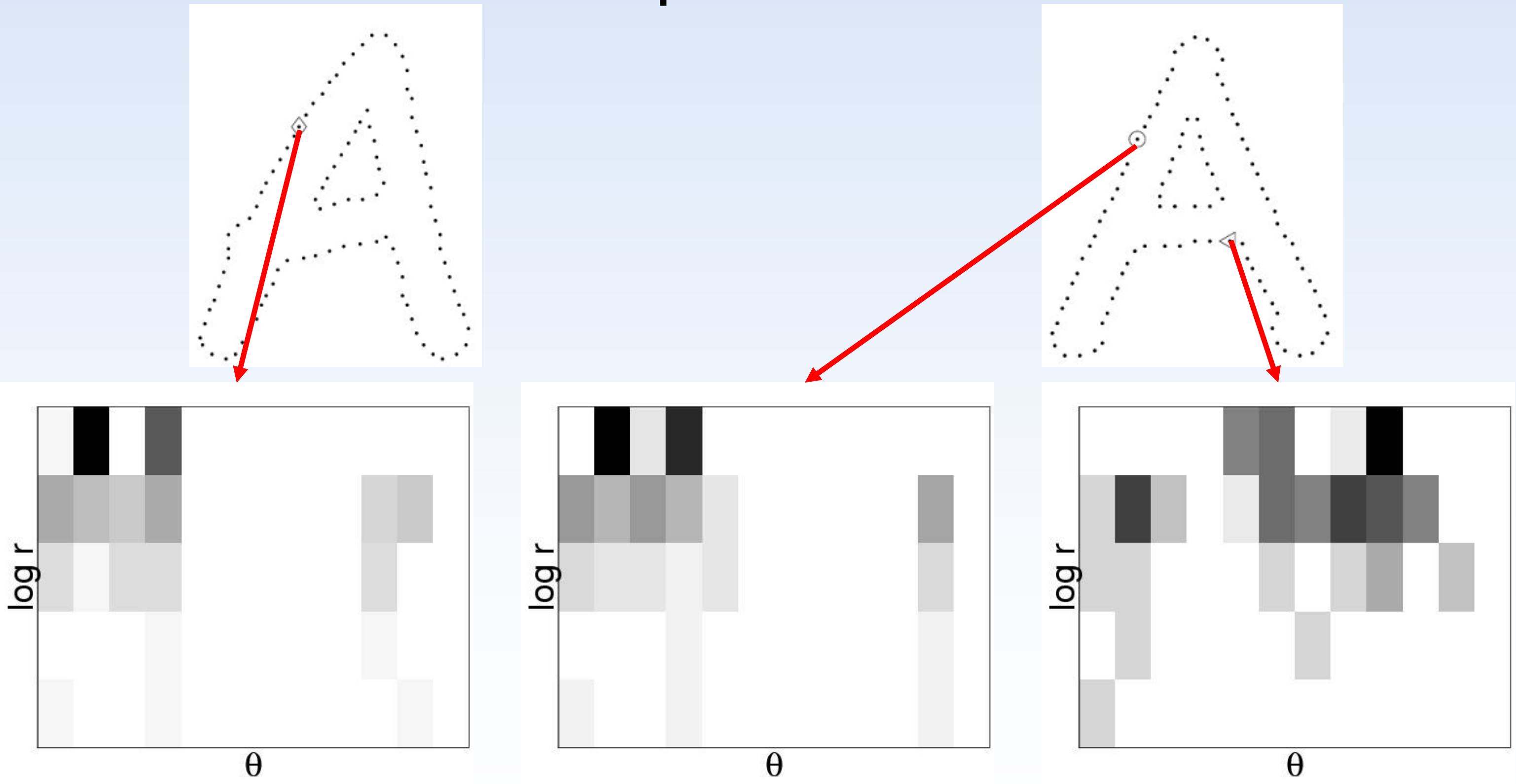
Matching with Shape Contexts

Shape Context



Matching with Shape Contexts

Shape Context



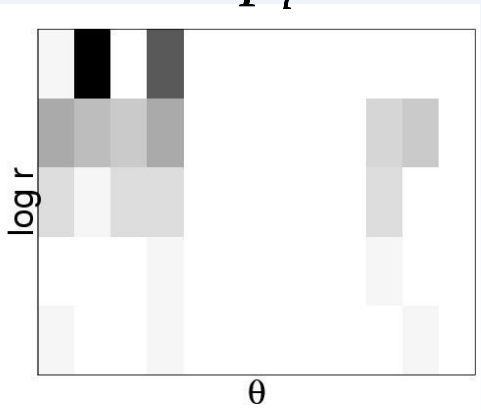
Belongie, et al, PAMI 2002, [Shape matching and object recognition using shape contexts]()

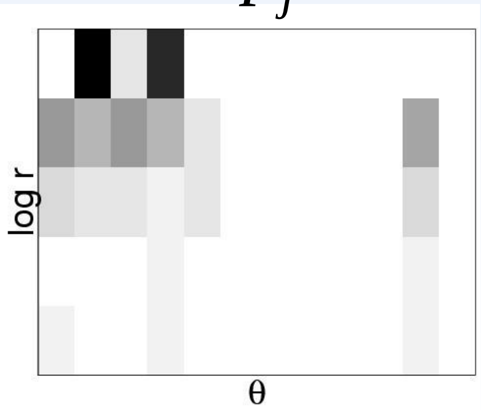
Matching with Shape Contexts

Shape Context

$$C_{ij} \equiv C(p_i, p_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{\left[h_i(k) - h_j(k)\right]^2}{h_i(k) + h_j(k)}$$

p_i p_j





Belongie, et al, PAMI 2002, Shape matching and object recognition using shape contexts

Matching with Shape Contexts

Bipartite Graph Matching

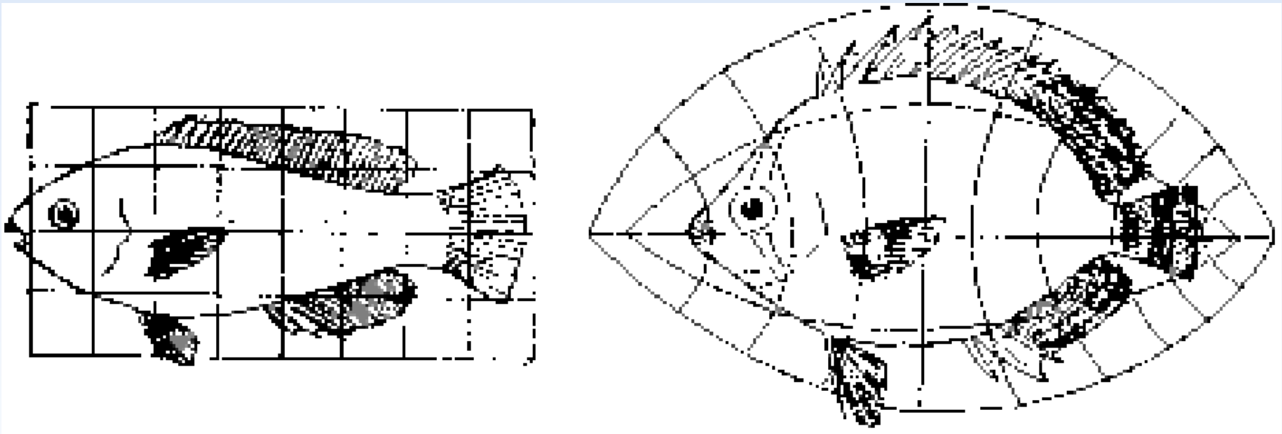
$$H(\pi) = \sum_i C(p_i, q_{\pi(i)})$$

Solved in about $O(N^3)$

Belongie, et al, PAMI 2002, [Shape matching and object recognition using shape contexts]

Matching with Shape Contexts

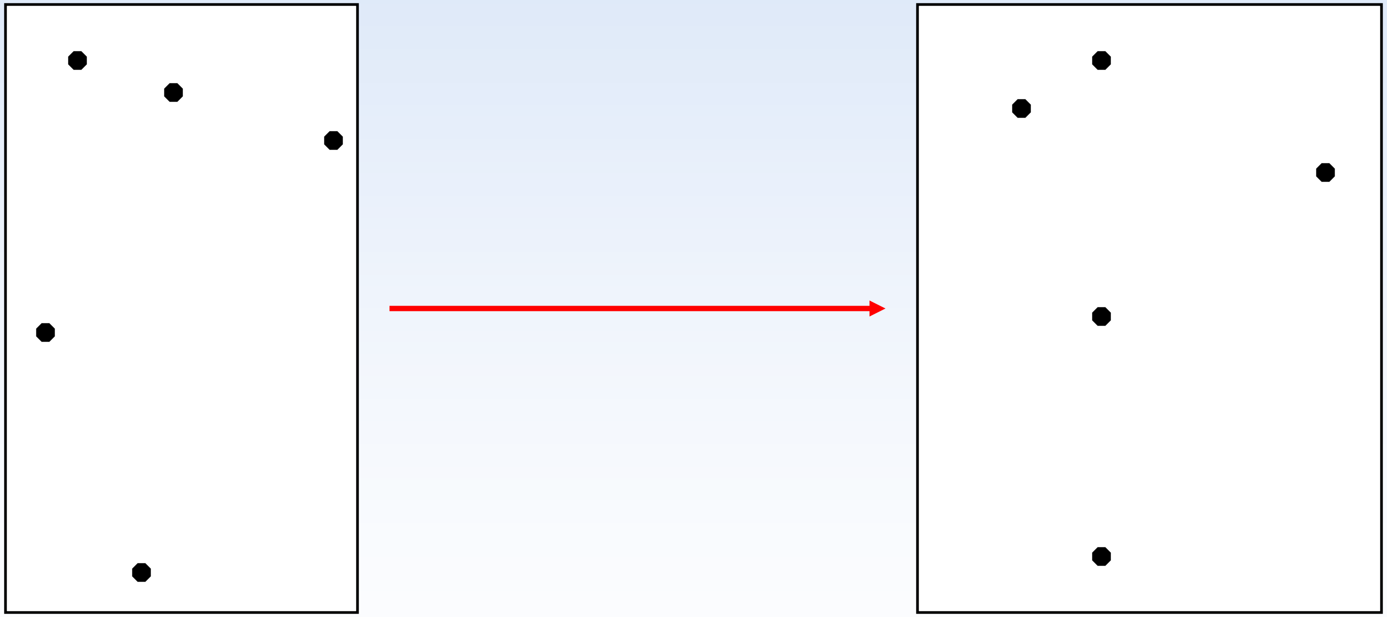
Modeling Transformations



Belongie, et al, PAMI 2002, Shape matching and object recognition using shape contexts

Matching with Shape Contexts

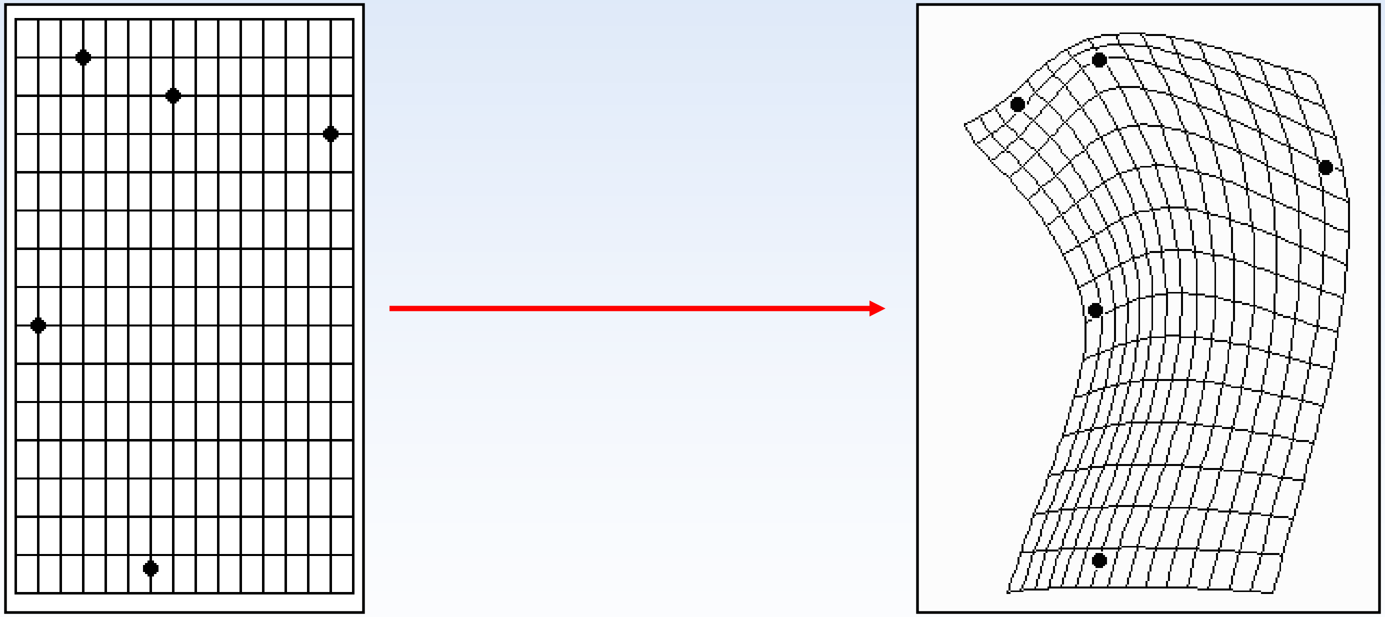
Thin Plane Spline (TPS) Model
(2D Generalization of Cubic Spline)



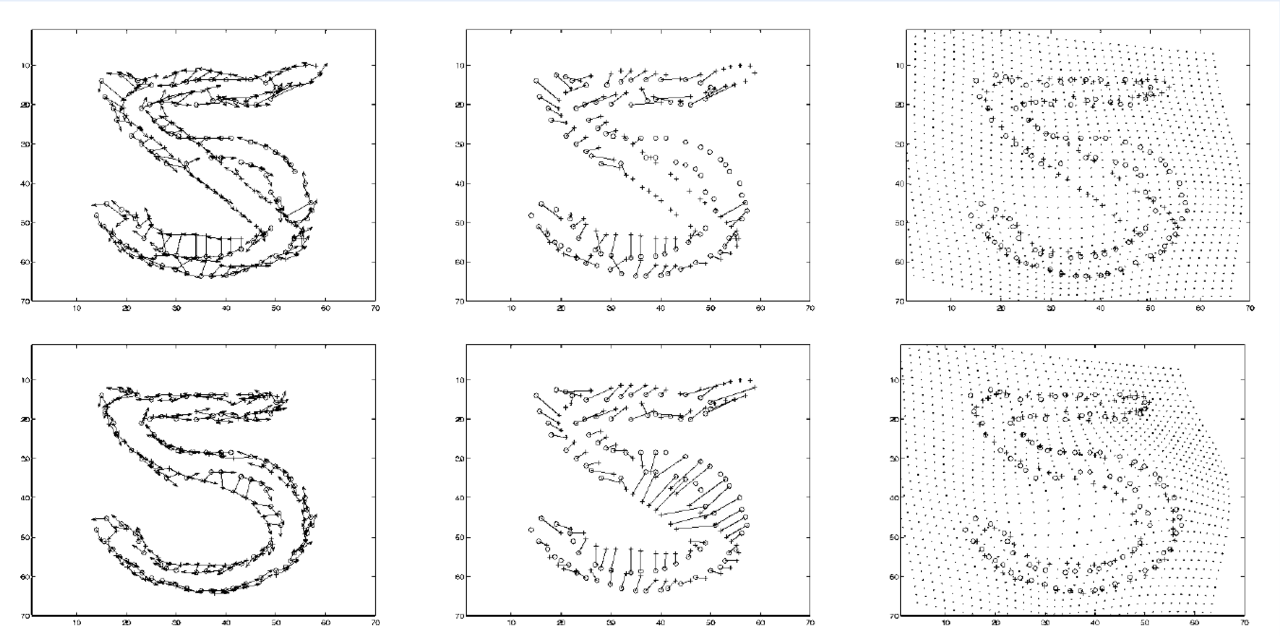
Belongie, et al, PAMI 2002, Shape matching and object recognition using shape contexts

Matching with Shape Contexts

Thin Plane Spline (TPS) Model
(2D Generalization of Cubic Spline)



Matching with Shape Contexts



Belongie, et al, PAMI 2002, [Shape matching and object recognition using shape contexts]()

Matching with Shape Contexts

Invariance and Robustness

- Invariant under translation and scaling
- Insensitive to small affine distortion
- Can be made invariant to rotation

Belongie, et al, PAMI 2002, [Shape matching and object recognition using shape contexts]

Matching with Shape Contexts

Evaluation and Results

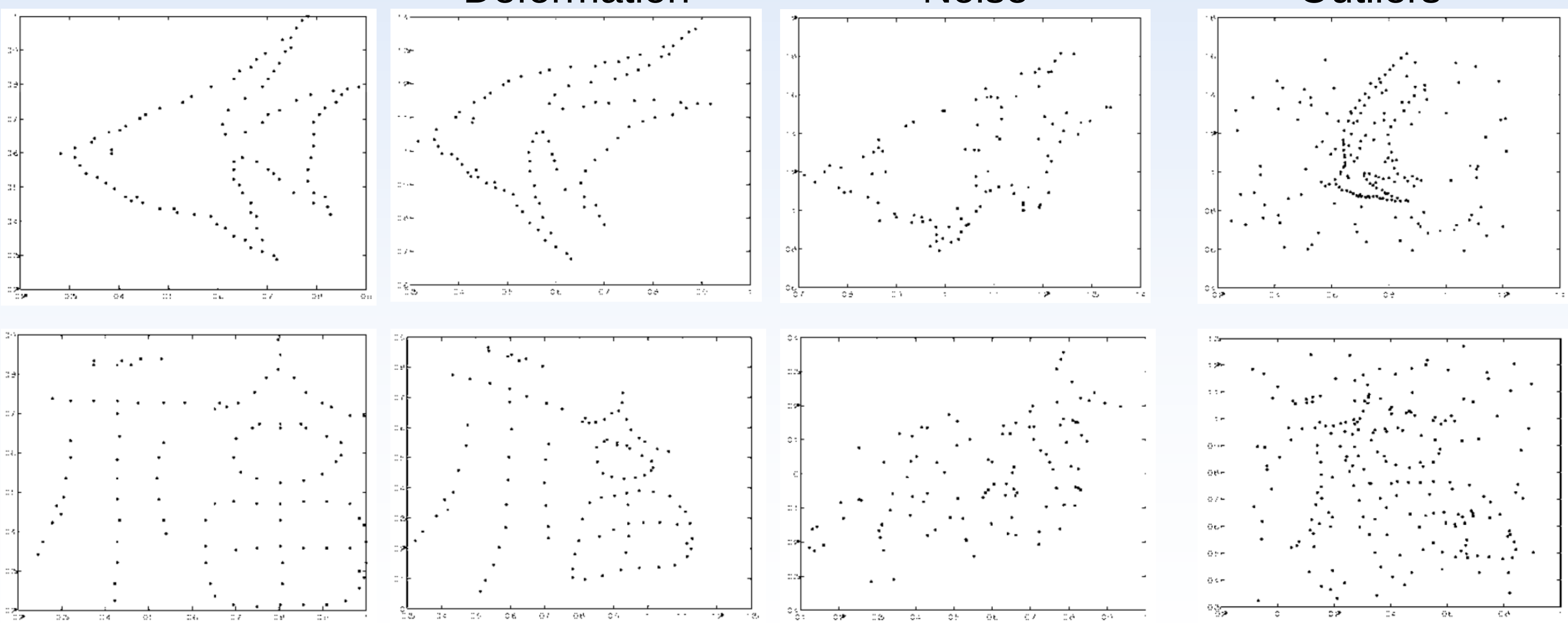
Model Point Sets

Target Point Sets

Deformation

Noise

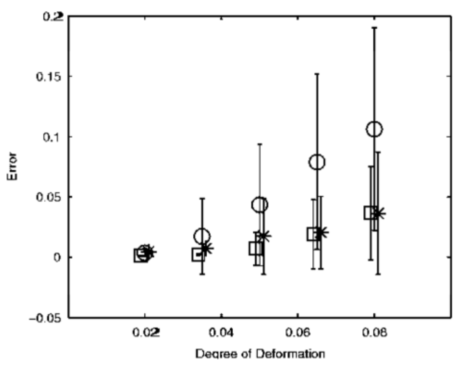
Outliers

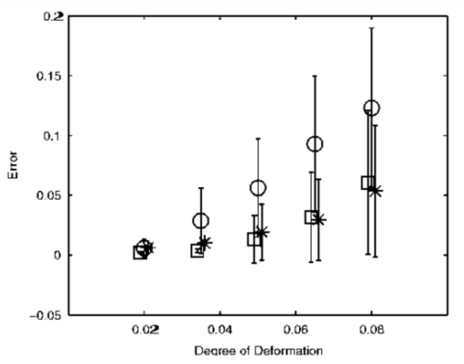


Matching with Shape Contexts

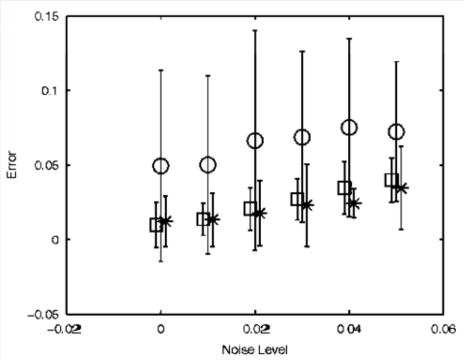
Evaluation and Results

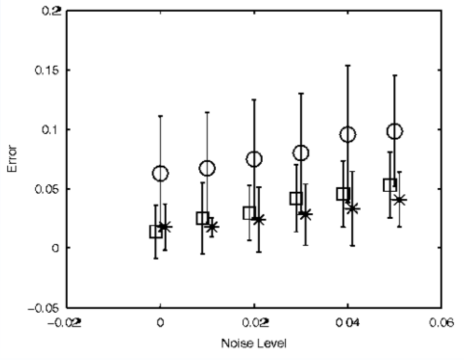
Deformation



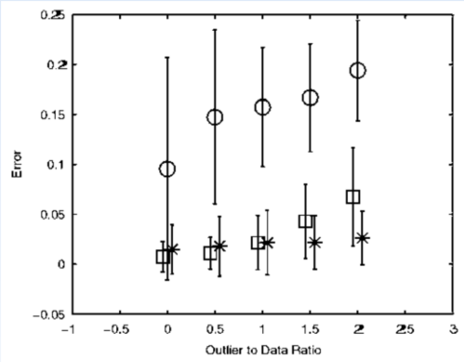


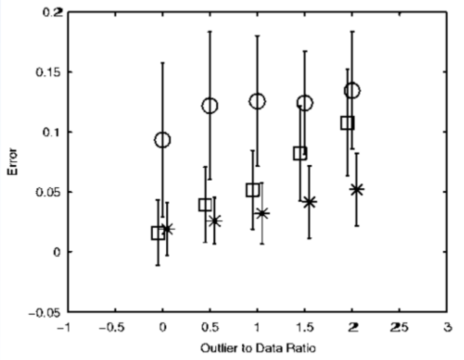
Noise





Outliers

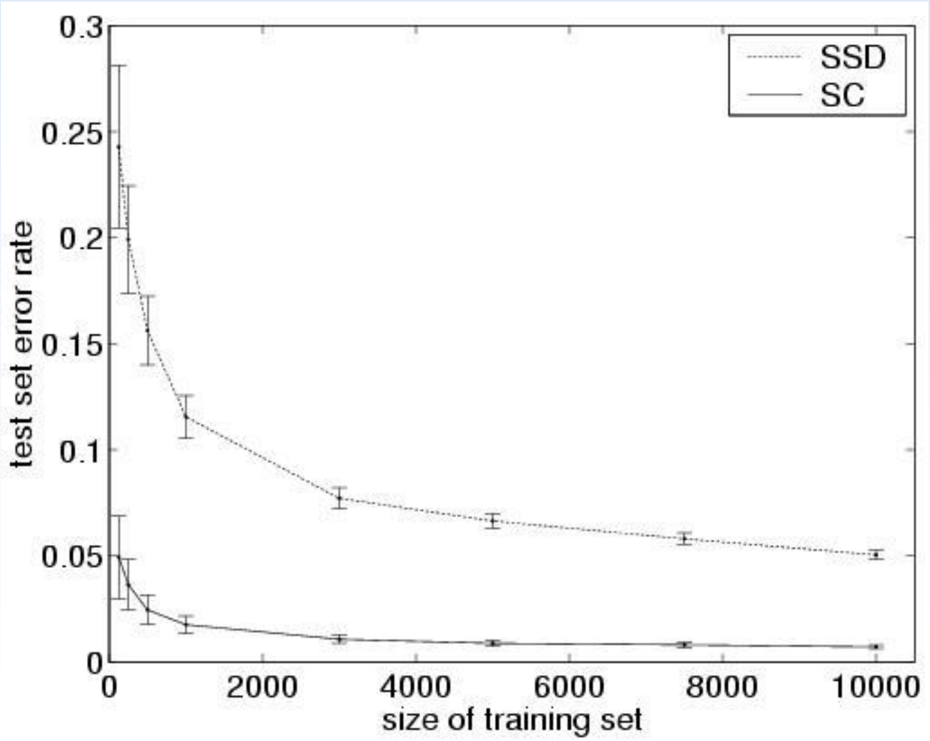




□ Belongie et al. * Chui and Rangarajan ○ Iterated closed point

Matching with Shape Contexts

Evaluation and Results



Conclusion

Questions

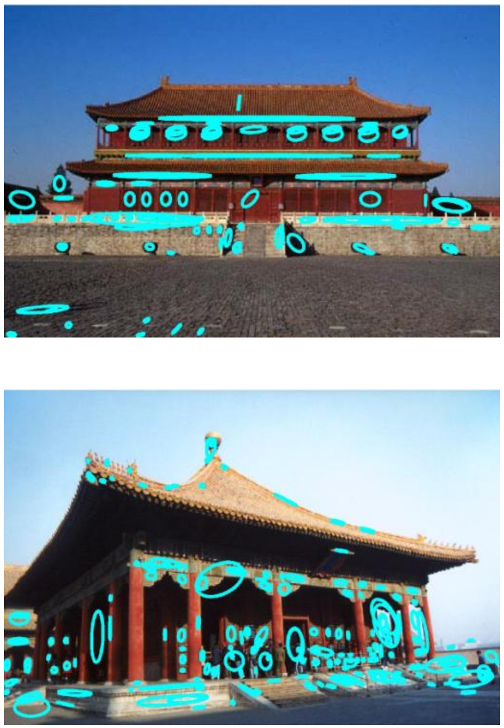
Shape and Image Matching

Shengnan Wang
shengnan@cs.wisc.edu

Today

- The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features
  - Kristen Grauman & Trevor Darrell
  - MIT
- Matching Local Self-Similarities across Images and Videos
  - Eli Shechtman & Michal Irani
  - @ CVPR07

Set Representation



invariant region descriptors

local shape features

examples under varying conditions

Motivation

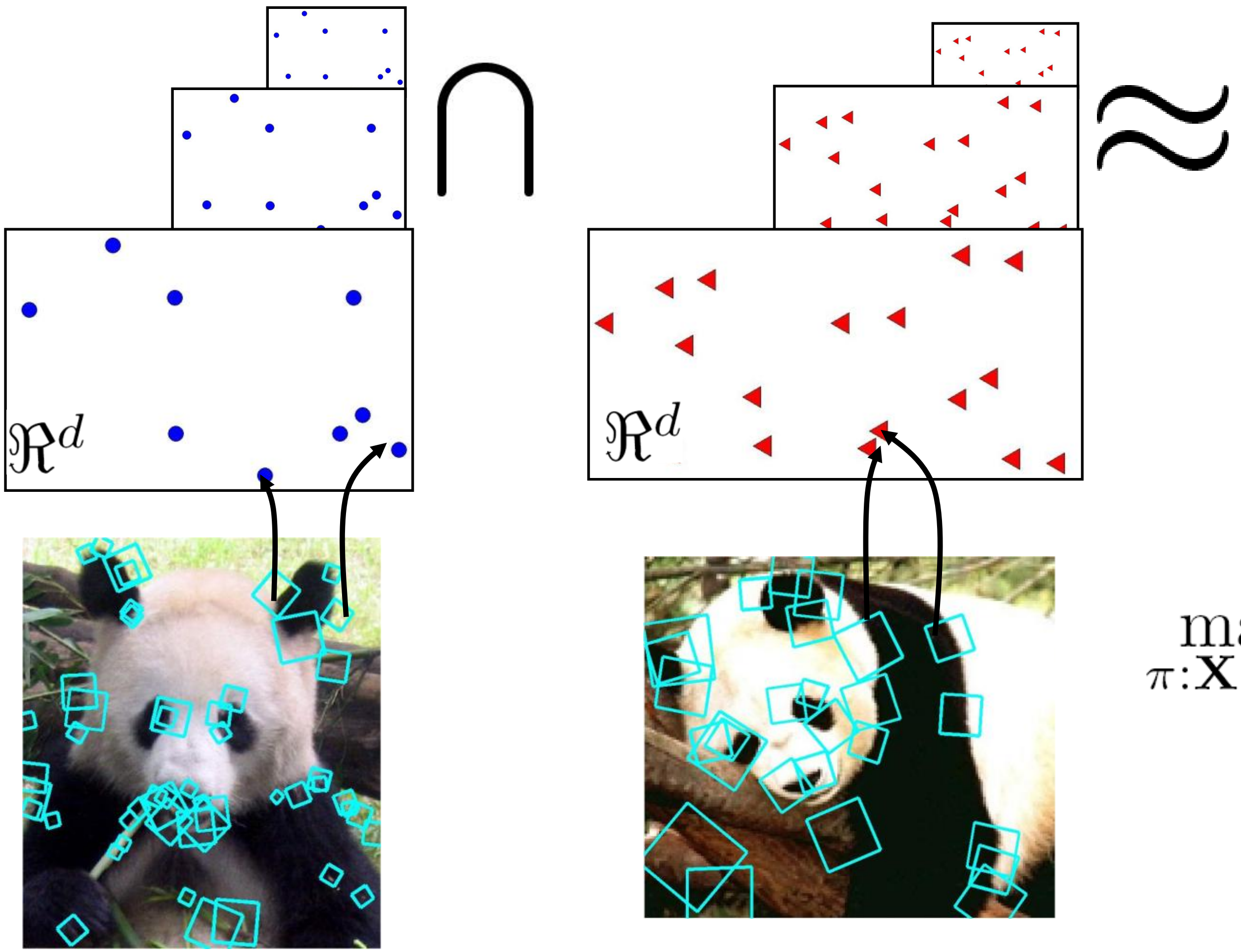
- How to build a **discriminative classifier** using the set representation?

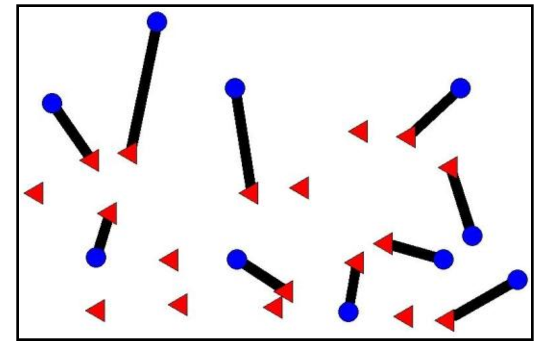
- Kernel-based methods (e.g. SVM) are appealing for efficiency and generalization power…

- What determines the appropriates of a kernel?

  - Each instance is unordered set of vectors
  - Varying number of vectors per instance

Pyramid Match Kernel



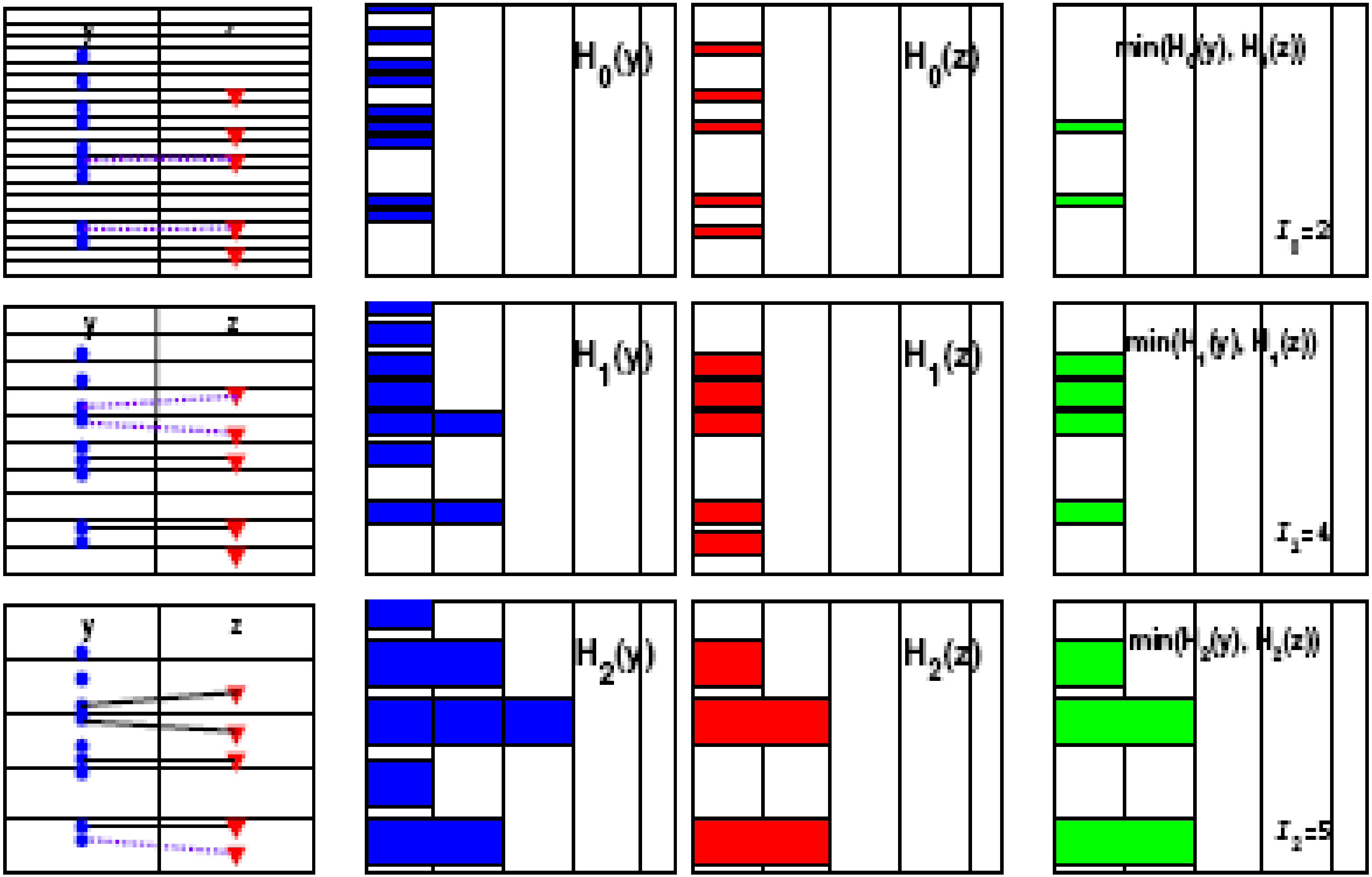


optimal partial matching

$$\max_{\pi:\mathbf{X}\rightarrow\mathbf{Y}} \sum_{\mathbf{x}_i\in\mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

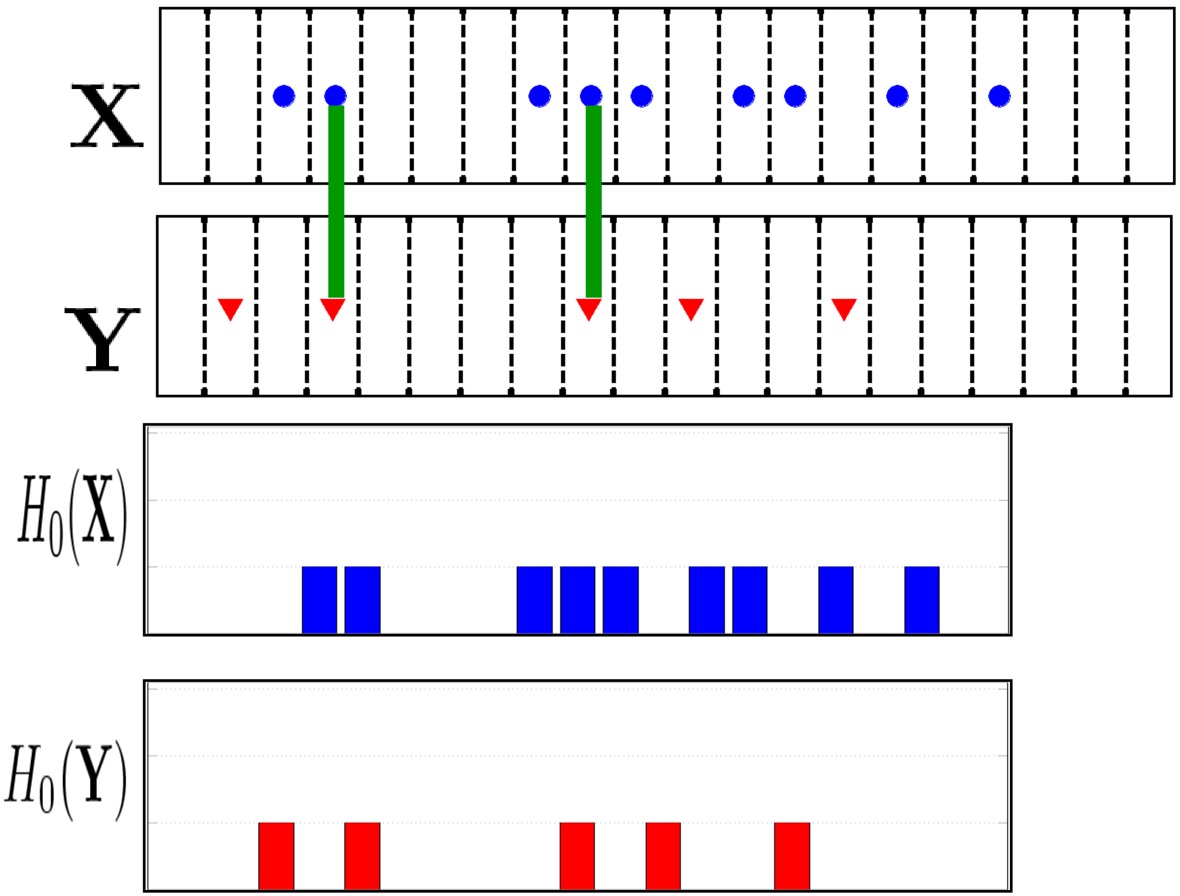
$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$$
$$\vec{\mathbf{x}}_i \in \Re^d$$

$$\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$$
$$\vec{\mathbf{y}}_i \in \Re^d$$



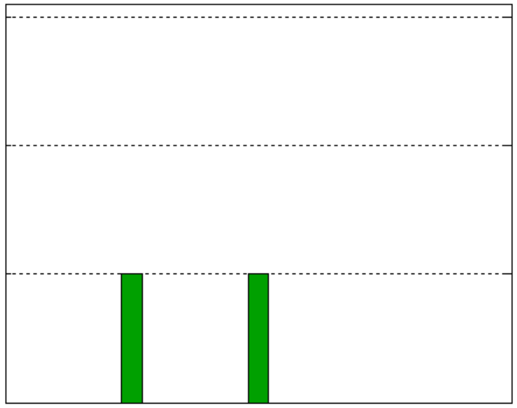
(a) Point sets (b) Histogram pyramids (c) Intersections

Example pyramid match



Level 0

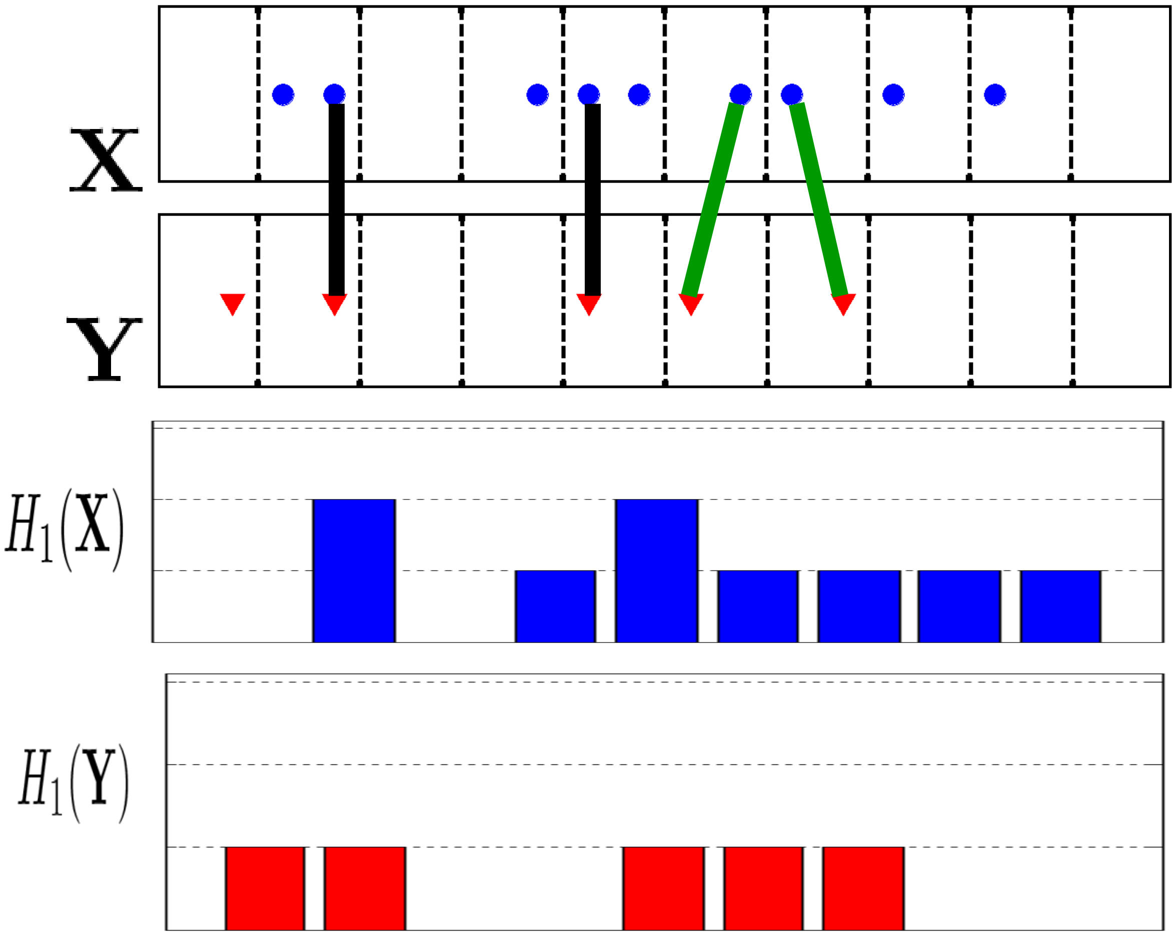
$$N_0 = 2$$
$$w_0 = 1$$



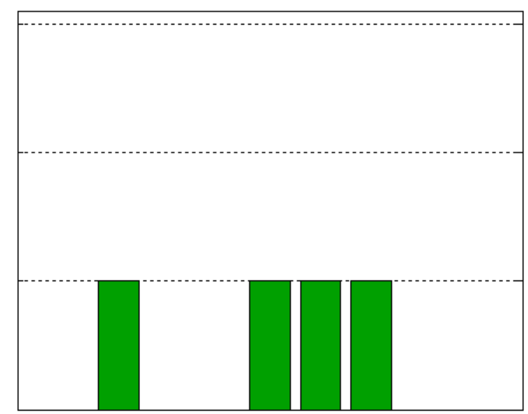
$$\mathcal{I}_0 = 2$$

Example pyramid match

Level 1

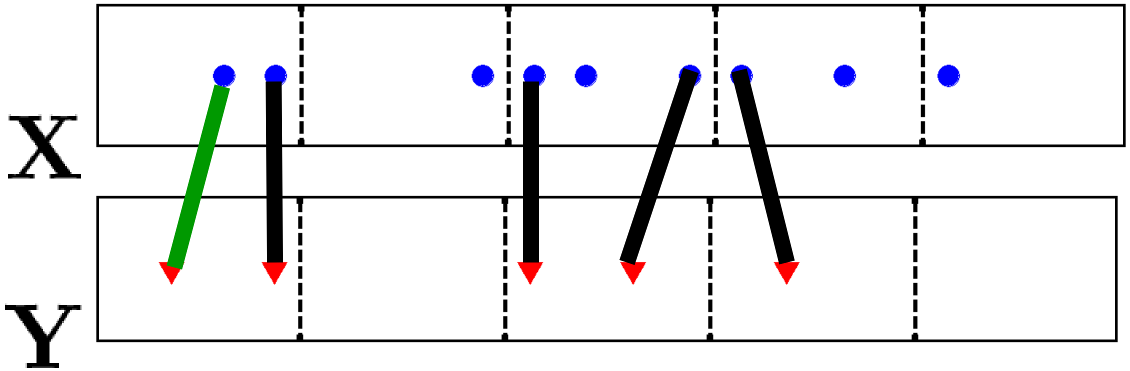


$$N_1 = 4 - 2 = 2$$
$$w_1 = \frac{1}{2}$$



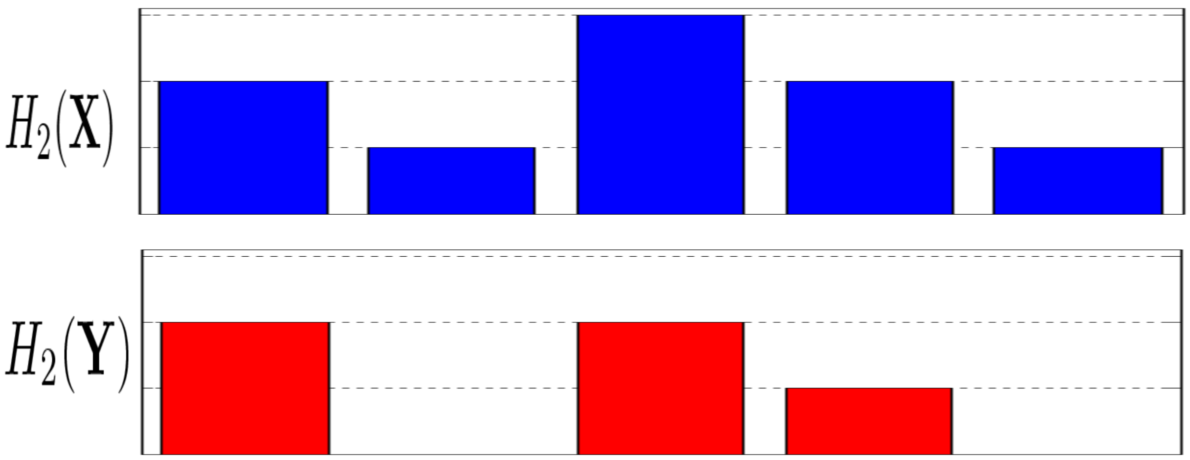
$$\mathcal{I}_1 = 4$$

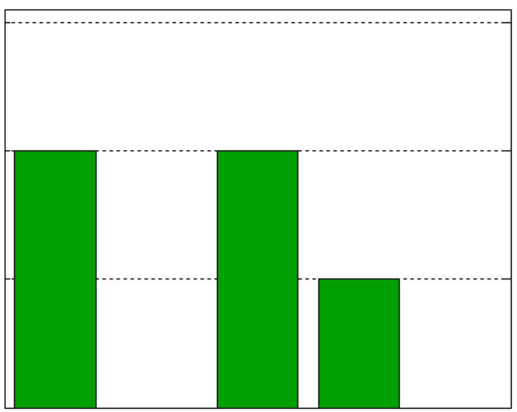
Example pyramid match



Level 2

$$N_2 = 5 - 4 = 1$$
$$w_2 = \frac{1}{4}$$





$$\mathcal{I}_2 = 5$$

Pyramid match kernel

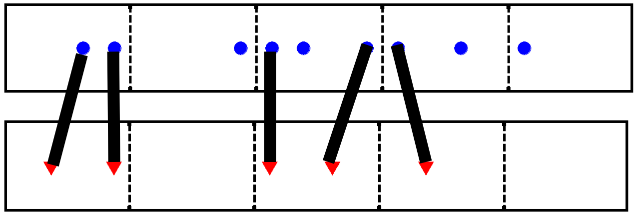
$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$

$$N_i = \mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)$$

$$K_\Delta = \sum_{i=0}^{L} w_i N_i$$

Example pyramid match

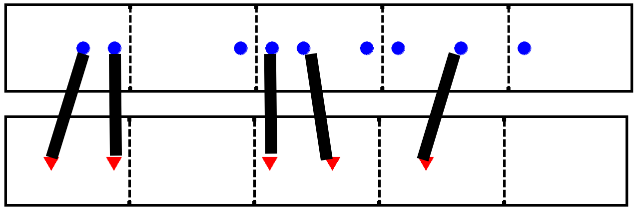
pyramid match



$$K_\Delta = \sum_{i=0}^{L} w_i N_i$$

$$= 1(2) + \tfrac{1}{2}(2) + \tfrac{1}{4}(1) = 3.25$$

optimal match



$$K = \max_{\pi : \mathbf{X} \to \mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

$$= 1(2) + \tfrac{1}{2}(3) = 3.5$$

Summary: Pyramid match kernel

- linear time complexity: $O(dmL)$
  m features of dimension *d*, *L*-level pyramid
- model-free
- insensitive to clutter
- positive-definite function
- no independence assumption
- fast, effective object recognition

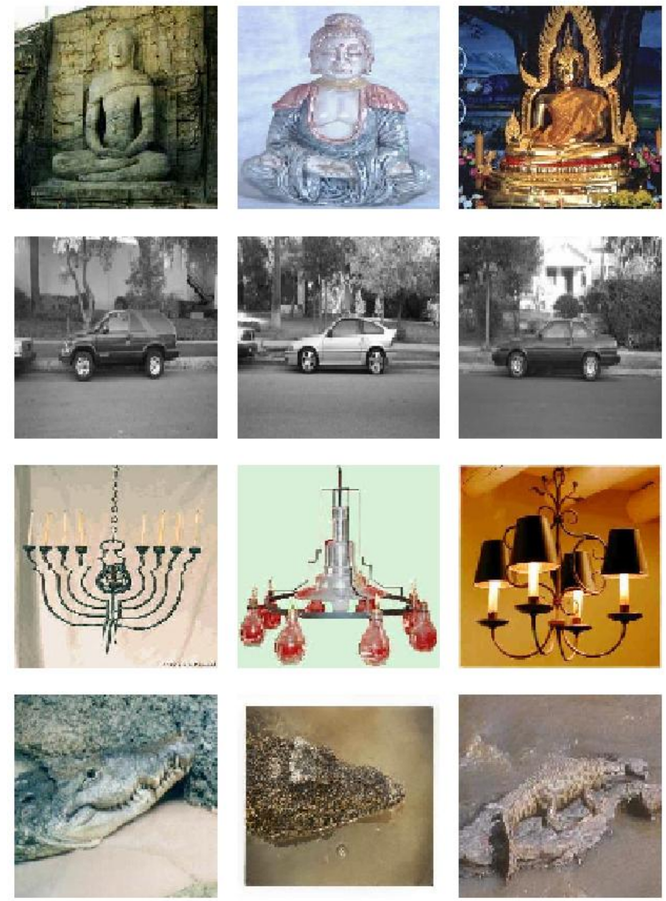
Object recognition results

- ETH-80 database :8 object classes
- Features:
  - Harris detector
  - PCA-SIFT descriptor, *d*=10

Kernel	**Complexity**	**Recognition rate**
Match *[Wallraven et al.]*	$O(dm^2)$	84%
Bhattacharyya affinity *[Kondor & Jebara]*	$O(dm^3)$	85%
Pyramid match	$O(dmL)$	84%

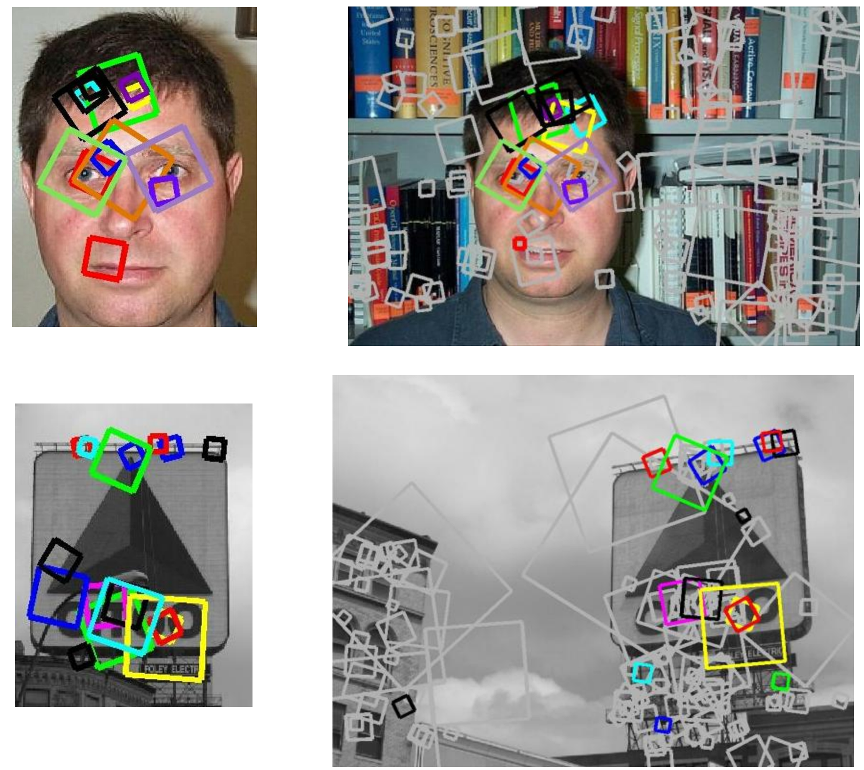
Object recognition results

- Caltech objects database 101 object classes
- Features:
  - SIFT detector
  - PCA-SIFT descriptor, d=10
- 30 training images / class
- 43% recognition rate
  (1% chance performance)
- 0.002 seconds per match



Localization

- Inspect intersections to obtain correspondences between features
- Higher confidence correspondences at finer resolution levels



target observation

Future work

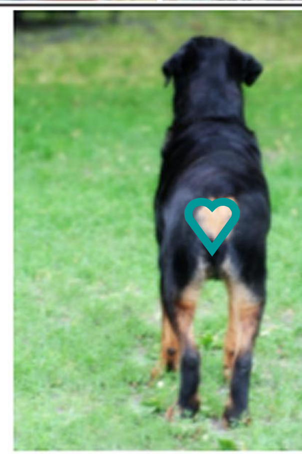
- Geometric constraints
- Fast search of large databases with the pyramid match for image retrieval
- Use as a filter for a slower, explicit correspondence method
- Alternative feature types and classification domains

Next

- Matching Local Self-Similarities across Images and Videos
  - Eli Shechtman & Michal Irani
  - @ CVPR07

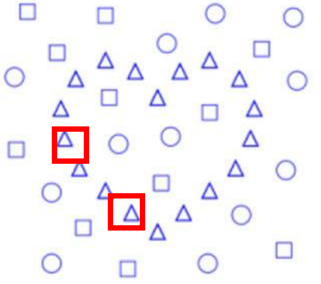
What do they do?

How to measure similarity between visual entities (images or videos)



What's new?

.VS.

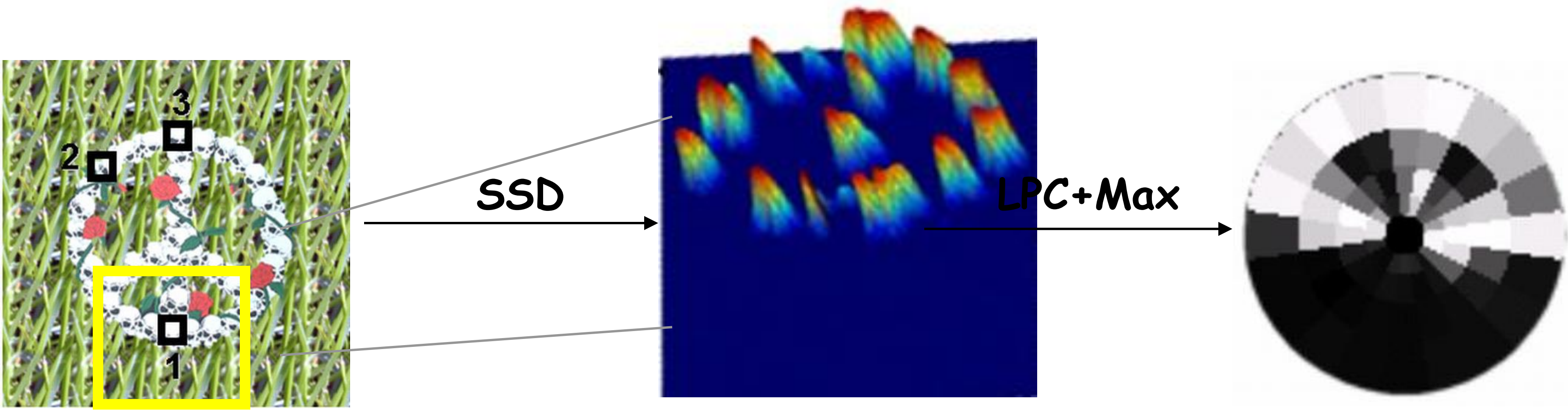


- Traditional idea: they share some common visual properties (colors, intensity, gradients, edges or other filter response)

- New idea: they share the same local geometry layout.

Local self repeat

Self-similarity descriptor



Patch **Local Similarity Map** **Self-similarity descriptor**

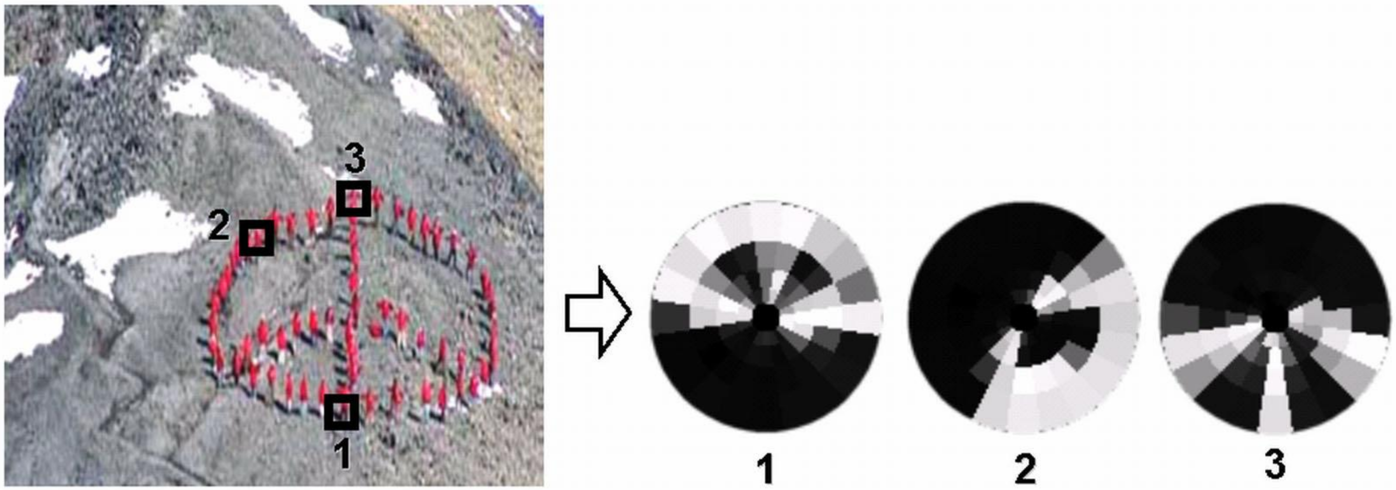
$$S_q(x,y) = \exp\left(-\frac{SSD_q(x,y)}{\max\left(var_{noise}, var_{auto}(q)\right)}\right)$$

$$r = [(x-x_c)^2 + (y-y_c)^2]^{1/2}, \qquad \theta = \tan^{-1}\left(\frac{y-y_c}{x-x_c}\right)$$

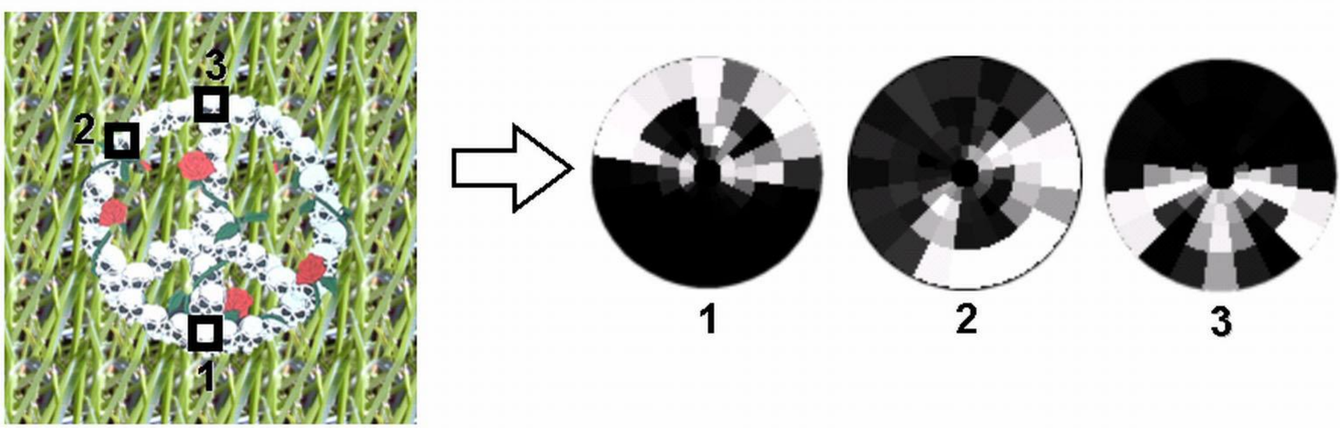
$$R = \frac{(n_r - 1)\log(r/r_{min}),}{\log(r_{max}/r_{min})} \qquad W = \frac{n_w \theta}{2\pi}$$

Corresponding Self-similarity descriptor

F



G

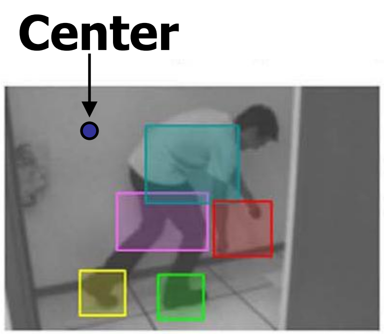


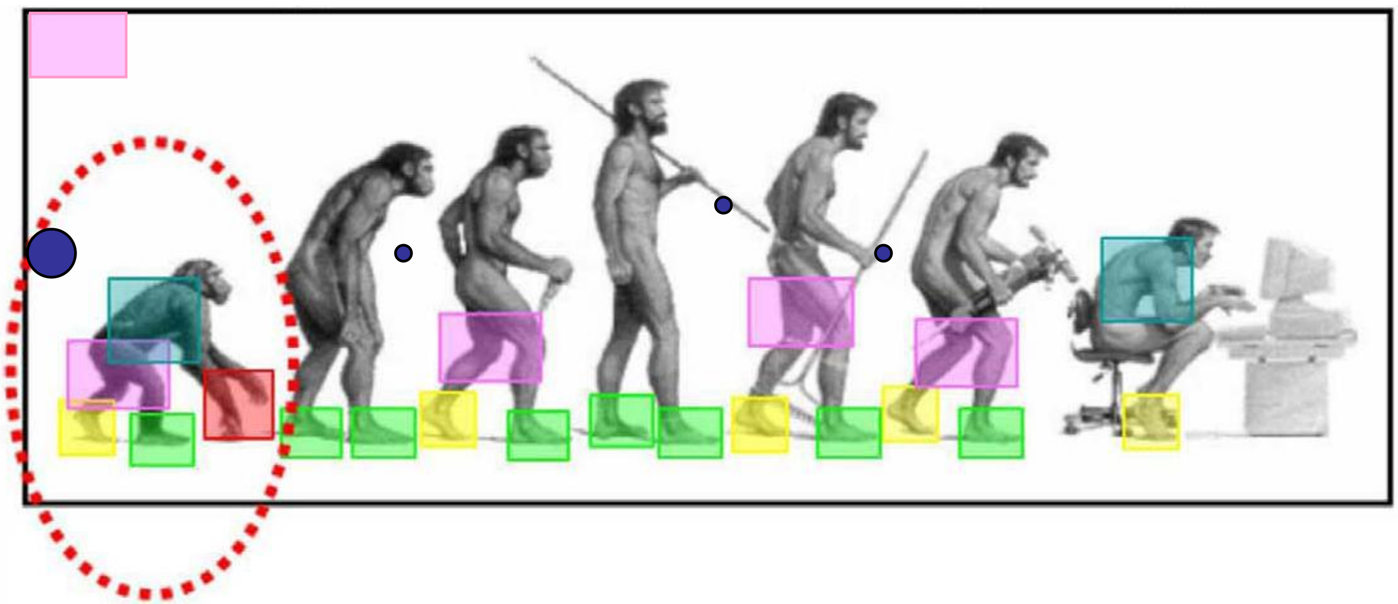
Self-similarity descriptor

- Benefits
  - self-similarity descriptor: local descriptors, wider applicability
  - log-polar: accounts for local affine deformations
  - maximal correlation value: insensitive to the exact best match position*
  - use of patches: more meaningful image patterns

***T. Wolf, L. Bileschi, S. Riesenhuber, M. Poggio, T., Robust Object Recognition with Cortex-Like Mechanisms Serre, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007**

Matching ensemble of patches*





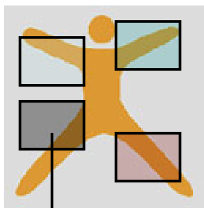
$$\max_{X} P\left(c_x, d_x^1, \ldots, l_x^1, \ldots, c_y, d_y^1, \ldots, l_y^1\right) = \alpha \prod_i \max_{l_x^i} P\left(l_y^i | l_x^i, c_x, c_y\right) \max_{d_x^i} P\left(d_y^i | d_x^i\right) P\left(d_x^i | l_x^i\right)$$

*O. Boiman and M. Irani. Detecting irregularities in images and in video. In IEEE International Conference on Computer Vision, Beijing, October 2005.

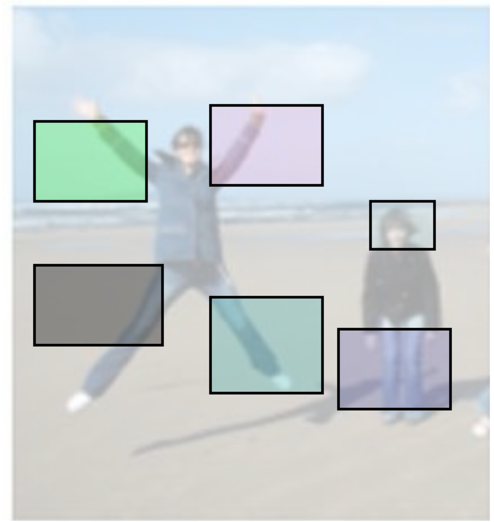
Matching ensemble of patches

- Two Implementation Details
  - Filter out non-informative descriptors
    - Descriptors that do not capture any local self-similarity (salient points)
    - Descriptors that contain high self-similarity everywhere (homogeneous region)
  - Scale space representation

Overview



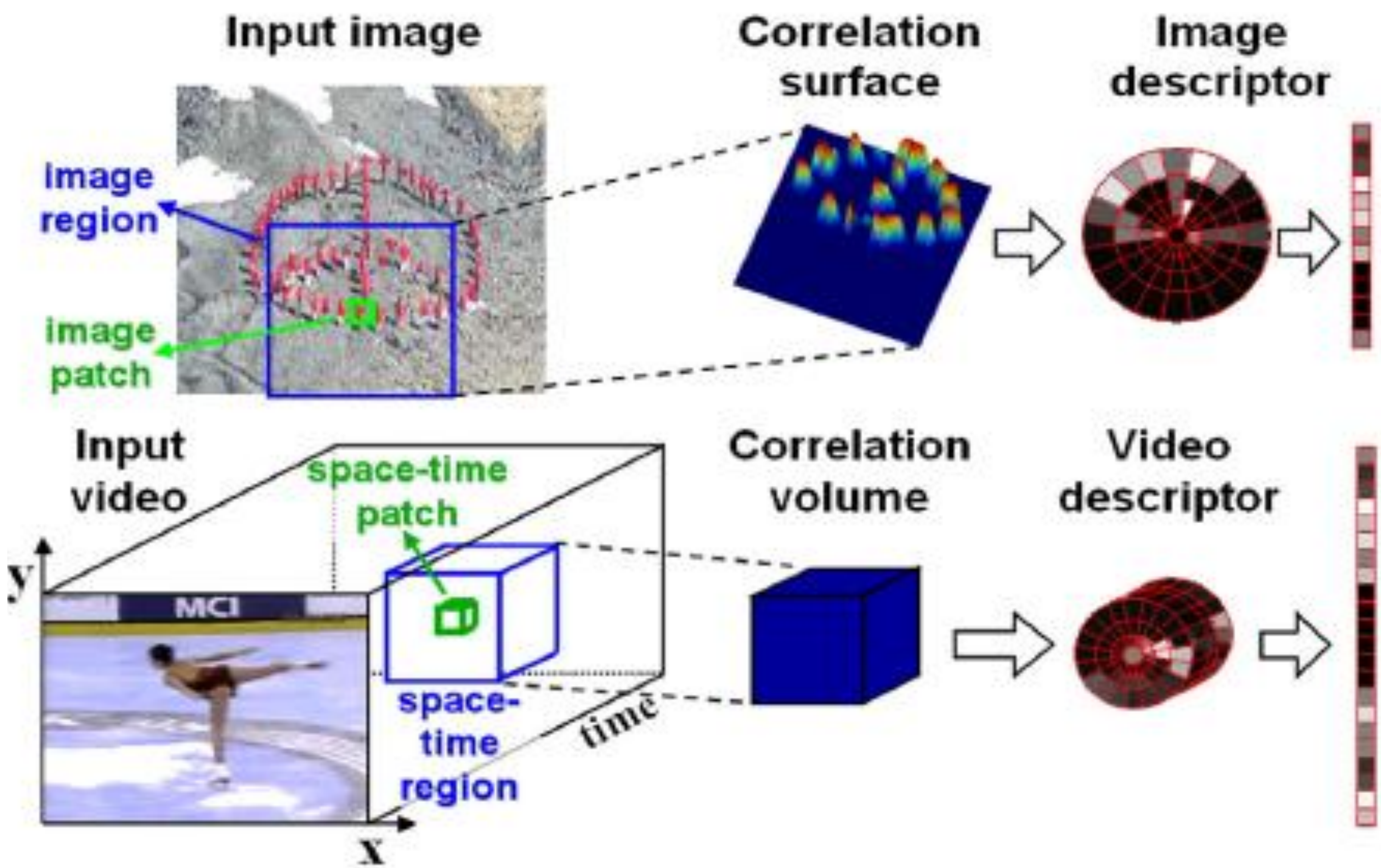
Matching



1. Self-similarity descriptor

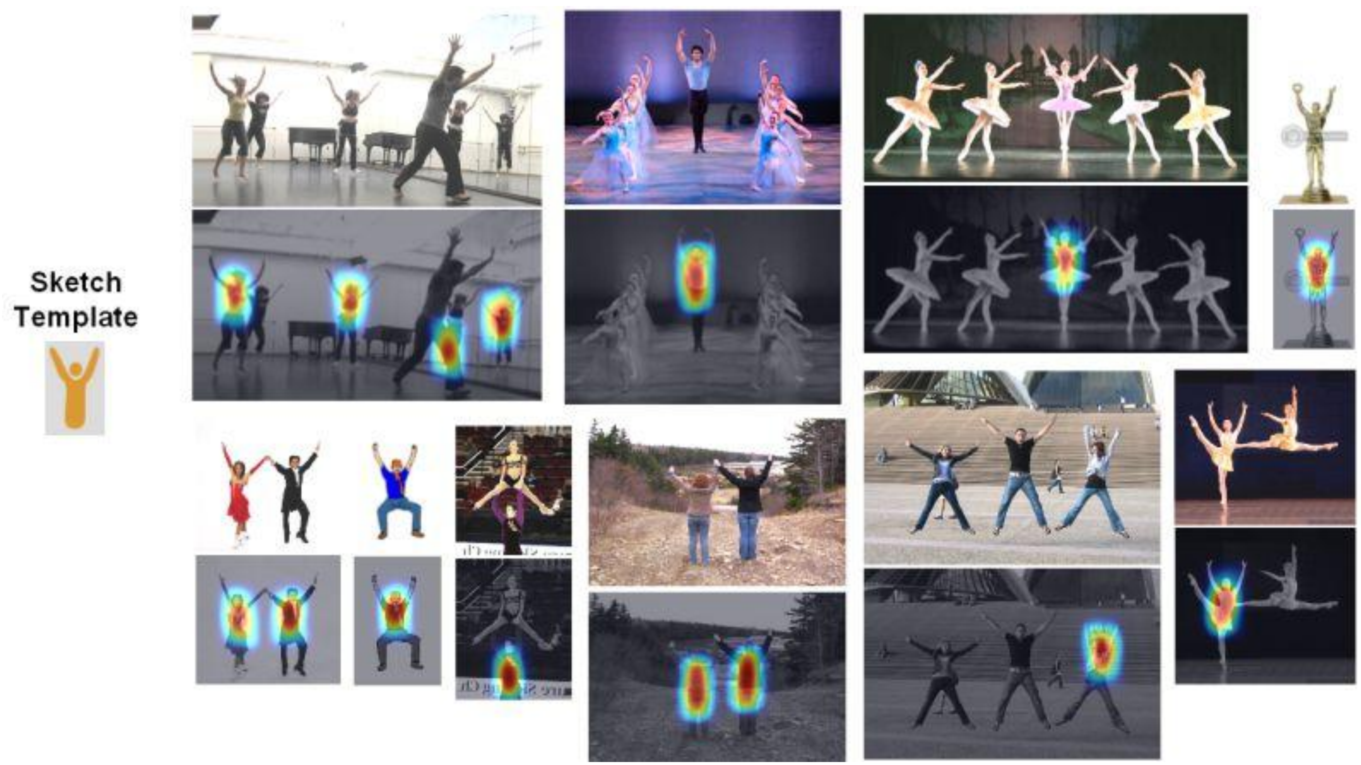
2. Matching ensemble of patches

Self-similarity descriptor



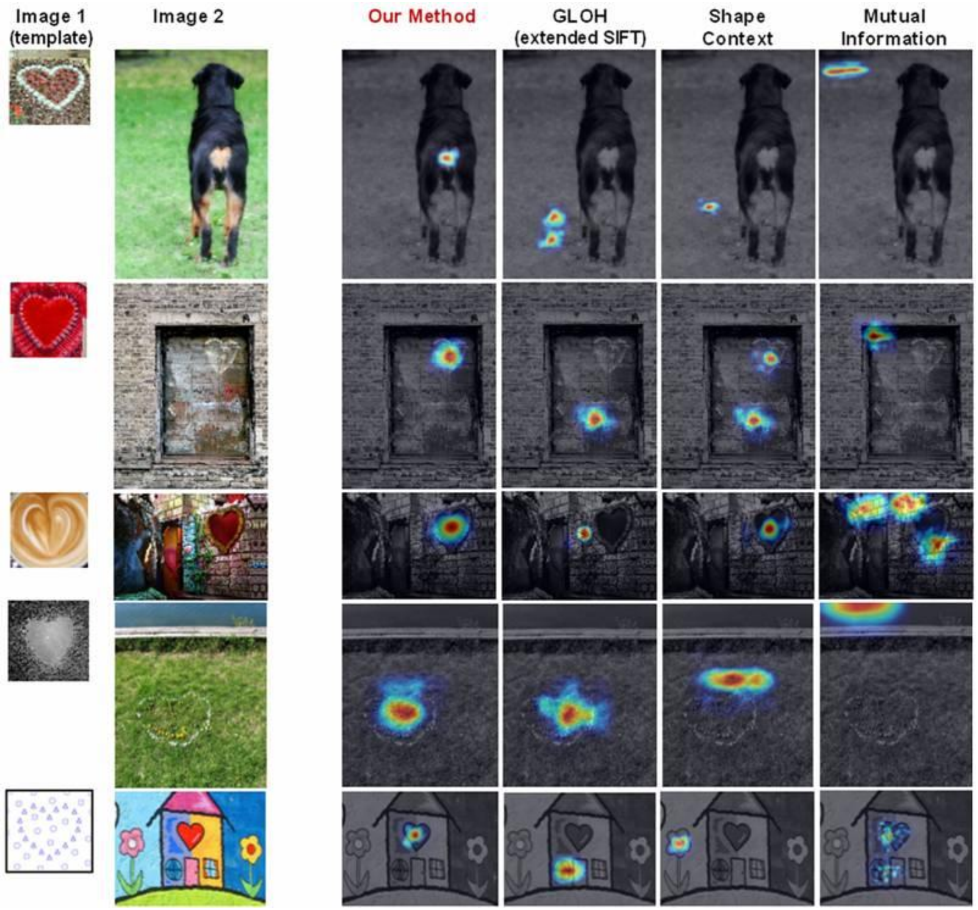
Experiments

- Sketch Detection



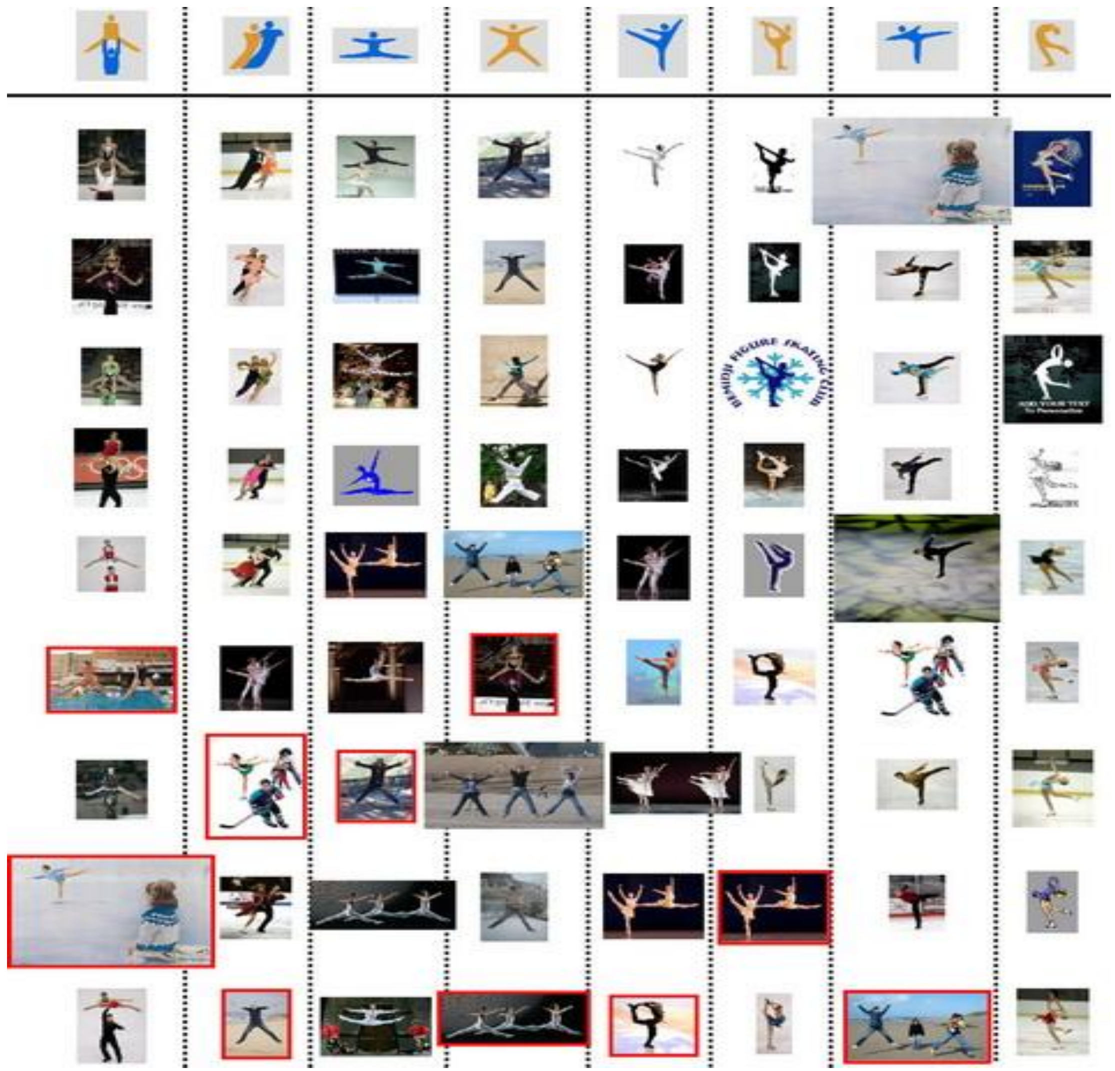
Experiments (1)

- Object Detection



Experiments (2)

- Image Retrieval



Experiments (3)

- Action detection

Experiments (4)

- Action detection

Questions?

Thank you!