Incorporating Domain Knowledge in Matching Problems via Harmonic Analysis

Deepti Pachauri
(joint work with Maxwell Collins, Risi Kondor, Vikas Singh)

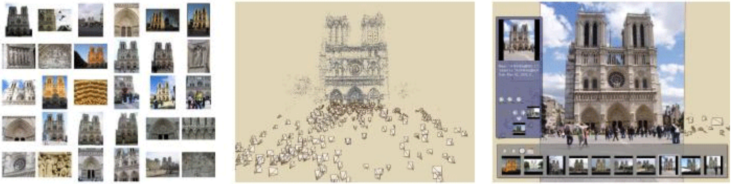
University of Wisconsin-Madison

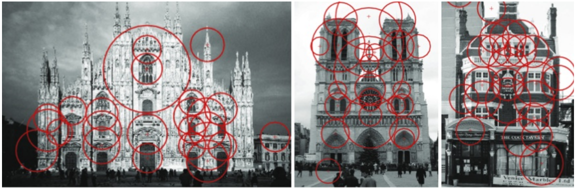
University of Chicago

International Conference on Machine Learning 2012

Matching Problems are Ubiquitous

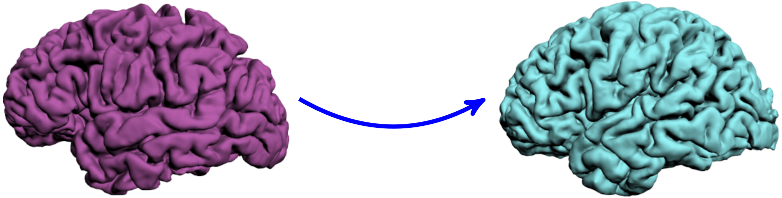
Photo Tourism





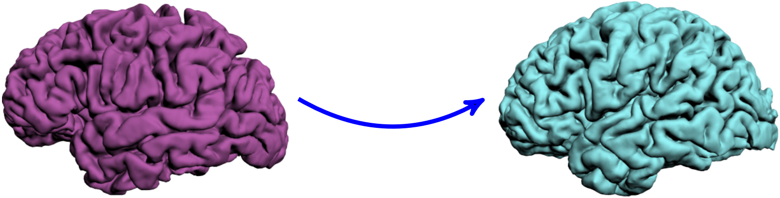
Matching Problems are Ubiquitous

Shape Matching



Matching Problems are Ubiquitous

Shape Matching

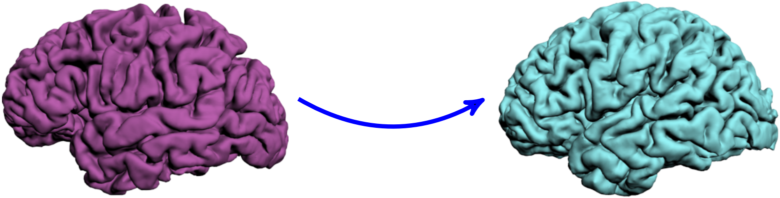


General Strategy

Write the functional form of the matching problem and then use an appropriate optimization engine to find a solution.

Matching Problems are Ubiquitous

Shape Matching



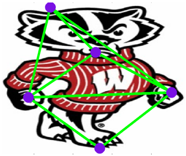
General Strategy

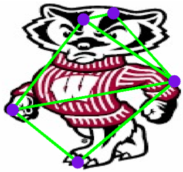
Write the functional form of the matching problem and then use an appropriate optimization engine to find a solution.

Use past knowledge to make future instances easier . . . ?

Overview

- ~~Motivation~~
- Problem Setup
  - Graph Matching and QAPs
- Why learn QAPs?
- Algebraic Structure of $\mathbb{S}_n$ and Harmonic Analysis
- Learning in Fourier Space
- Evaluations





$$G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad G' = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

$$G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad G' = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

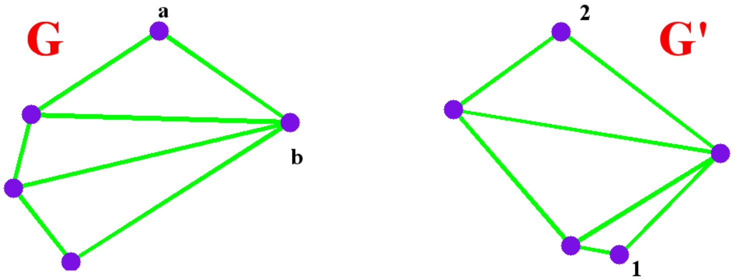
Solution of matching problem is a permutation matrix $\mathbf{y}$

$$y = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad \sigma := (51342)$$

such that $yGy^\top = G'$

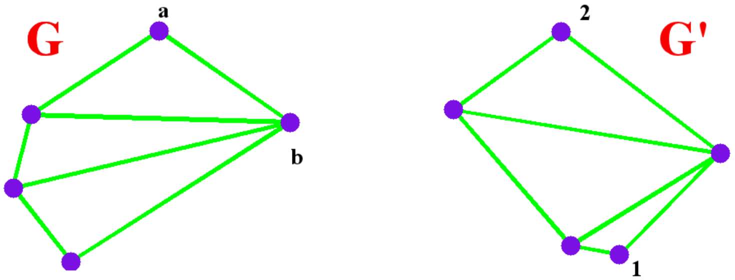
Quadratic Assignment Problem (QAP)

$$\mathbf{y}^* = \arg\max_y \sum_{ii'} c_{ii'} y_{ii'} + \sum_{ii'jj'} d_{ii'jj'} y_{ii'} y_{jj'}$$



Quadratic Assignment Problem (QAP)

$$\mathbf{y}^* = \arg\max_y \sum_{ii'} c_{ii'} y_{ii'} + \sum_{ii'jj'} d_{ii'jj'} y_{ii'} y_{jj'}$$



- Computationally expensive: $n \geq 40$ infeasible in general.

Supervised Learning

Given

Training data : $((x_1, y_1), \ldots, (x_m, y_m))$

$$f^{\omega}(x_i) \approx y_i$$

$$(x_1, y_1) : f^{\omega}(x_1) \approx y_1$$

$$(x_2, y_2) : f^{\omega}(x_2) \approx y_2$$

$$(x_3, y_3) : f^{\omega}(x_3) \approx y_3$$

and so on ....

Learning for QAPs?

Given

Training data : $((x_1, \sigma_1), ..., (x_m, \sigma_m))$

$$\arg\max f^{\omega}(x_i) \approx \sigma_i$$

$$(x_1, \sigma_1) : \arg\max f^{\omega}(x_1) \approx \sigma_1$$

$$(x_2, \sigma_2) : \arg\max f^{\omega}(x_2) \approx \sigma_2$$

$$(x_3, \sigma_3) : \arg\max f^{\omega}(x_3) \approx \sigma_3$$

and so on ....

Learning for QAPs?

Given

Training data : $((x_1, \sigma_1), ..., (x_m, \sigma_m))$

$$\arg\max f^{\omega}(x_i) \approx \sigma_i$$

$$(x_1, \sigma_1) : \arg\max f^{\omega}(x_1) \approx \sigma_1$$

$$(x_2, \sigma_2) : \arg\max f^{\omega}(x_2) \approx \sigma_2$$

$$(x_3, \sigma_3) : \arg\max f^{\omega}(x_3) \approx \sigma_3$$

and so on ....

. . . and we want to solve $\arg\max f^{\omega}(x_i)$ *cheaply*.

Inspired in part by

Caetano et al., PAMI 2009

- Structure learning approach to find most violated constraints using linear assignment.

Xu et al., JMLR 2009

- Use disciminative learning to acquire a domain–specific heuristic for controlling beam–search.

Stobbe et al., AISTATS 2012

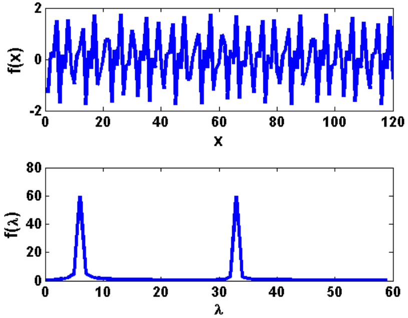
- Fourier space sparsity to recover a set function from very few samples.

Structure of $\sigma \in \mathbb{S}_\mathbf{n}$

Harmonic Analysis

Fourier transform of a function $f : \mathbb{R} \mapsto \mathbb{C}$

$$\hat{f}(\lambda) = \sum_{x \in \mathbb{R}} f(x) e^{2\pi i x \lambda} \qquad \lambda \in \mathbb{R},$$



Structure of $\sigma \in \mathbb{S}_\mathbf{n}$

Harmonic Analysis on Symmetric Groups $\mathbb{S}_n$

$$\hat{f}(\rho_\lambda) = \sum_{\sigma \in \mathbb{S}_n} f(\sigma)\rho_\lambda(\sigma) \qquad \rho_\lambda \in \mathcal{R}$$

- λ is the integer partition of n, $\lambda \vdash n$
- $\rho_\lambda(\sigma)$ is the irreducible representation of $\mathbb{S}_n$

$$\rho_\lambda(\sigma) = \begin{pmatrix} \rho_{1,1} & \cdot & \cdot & \rho_{1,d_\lambda} \\ \cdot & \cdot & \cdot & \cdot \\ \rho_{d_\lambda,1} & \cdot & \cdot & \cdot \end{pmatrix}$$

Properties $\mathbb{S}_n$

Convolution

$(f * g)(\sigma) = \sum_{\tau \in \mathbb{S}_n} f(\sigma\tau^{-1})\, g(\tau) \quad \widehat{f * g}(\lambda) = \hat{f}(\lambda)\hat{g}(\lambda)$

Correlation

$(f \star g)(\sigma) = \sum_{\tau \in \mathbb{S}_n} f(\sigma\tau) g(\tau)^* \quad \widehat{f \star g}(\lambda) = \hat{f}(\lambda)\hat{g}(\lambda)^\dagger$

- $\mathbb{S}_{n-1}$ is a subgroup of $\mathbb{S}_n$

Properties $\mathbb{S}_n$

Convolution

$(f * g)(\sigma) = \sum_{\tau \in \mathbb{S}_n} f(\sigma\tau^{-1})\, g(\tau) \quad \widehat{f * g}(\lambda) = \hat{f}(\lambda)\hat{g}(\lambda)$

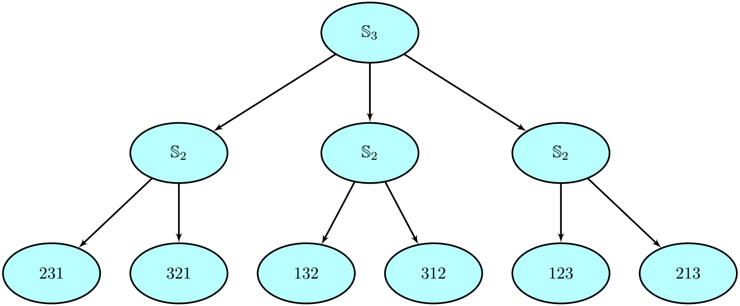
Correlation

$(f \star g)(\sigma) = \sum_{\tau \in \mathbb{S}_n} f(\sigma\tau) g(\tau)^* \quad \widehat{f \star g}(\lambda) = \hat{f}(\lambda)\hat{g}(\lambda)^\dagger$

- $\mathbb{S}_{n-1}$ is a subgroup of $\mathbb{S}_n$
- The set $\sigma\mathbb{S}_{n-1}$ is called a **left coset** of σ
- Two left (right) cosets are either disjoint or the same

Coset Tree

Cosets provide a partition of $\mathbb{S}_n$:



$f : \mathbb{S}_n \to \mathbb{C}$

Graph function of G

$$f_A(\sigma) = A_{\sigma(n),\sigma(n-1)}$$

Properties:

- $\mathbb{S}_{n-2}$-invariant function on adjacency matrix A (Kondor, 2010)
- *Band-limited* in Fourier domain (Rockmore, 2002)
- Under relabeling, $f_{A^\pi} = f_A^\pi$

Graph Matching Problem

Standard QAP:
Given a pair of graphs

$$\max_{\sigma \in \mathbb{S}_n} f(\sigma) = \sum_{i,j=1}^{n} A_{i,j} A'_{\sigma(i),\sigma(j)}$$

Graph Correlation:

$$f(\sigma) = \frac{1}{(n-2)!} \sum_{\pi \in \mathbb{S}_n} f_A(\sigma\pi) f_{A'}(\pi)$$

(A, A') could be weighted or unweighted adjacency matrices.

Learning Graph Matching

Given: A training set of related graph pairs with D encodings of adjacency matrices : (G_m, G'_m), $m = \{1, \cdots, M\}$.

Goal: "Learn" parameters ω such that QAP procedure finds a *good* solution (*quickly*) for the test case (unseen graph pairs).

Learning Graph Matching

Given: A training set of related graph pairs with D encodings of adjacency matrices : (G_m, G'_m), $m = \{1, \cdots, M\}$.

Goal: "Learn" parameters ω such that QAP procedure finds a *good* solution (*quickly*) for the test case (unseen graph pairs).

- Define parameter vector $\omega \in \mathbb{R}^D$

Learning Graph Matching

Given: A training set of related graph pairs with D encodings of adjacency matrices : (G_m, G'_m), $m = \{1, \cdots, M\}$.

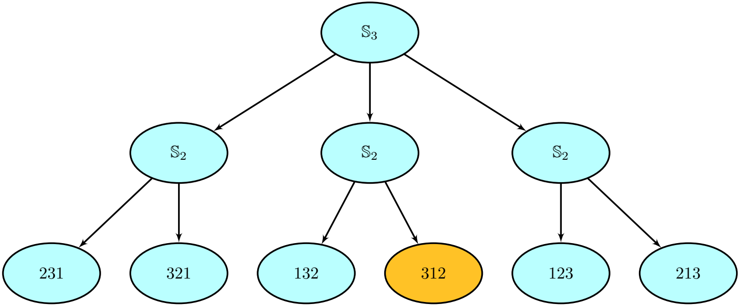
Goal: "Learn" parameters ω such that QAP procedure finds a *good* solution (*quickly*) for the test case (unseen graph pairs).

- Define parameter vector $\omega \in \mathbb{R}^D$

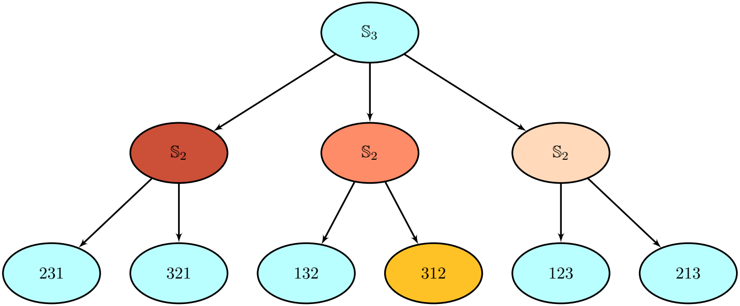
QAP Objective for Learning: $f^{\omega}(\sigma) = \sum_{d=1}^{D} \omega_d f^d(\sigma)$

where $f^d(\sigma) = \frac{1}{(n-2)!} \sum_{\pi \in \mathbb{S}_n} f_{A^d}(\sigma\pi) f_{A'^d}(\pi) = \sum_{i,j} A^d_{ij} A'^d_{\sigma(i)\sigma(j)}$

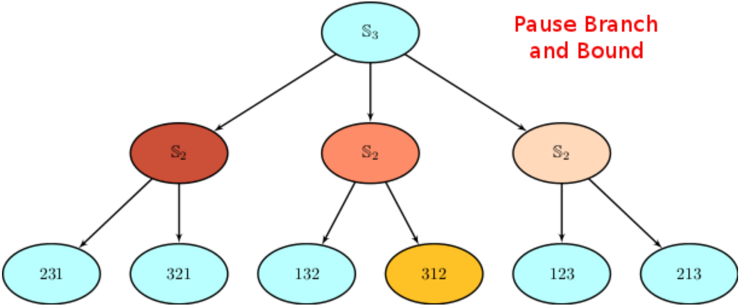
Learning Correct bounds on Coset Tree



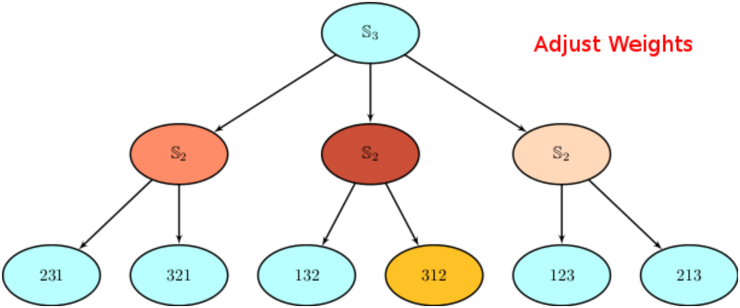
Learning Correct bounds on Coset Tree



Learning Correct bounds on Coset Tree



Learning Correct bounds on Coset Tree



Fourier Domain QAP Solver

Fast Fourier Transform

$$\hat{f}^{\omega}(\lambda) = \sum_{i=1}^{n} \frac{d_\lambda}{n d_\mu} \rho_\lambda([[i,n]]) \bigoplus_{\mu \in \lambda \downarrow n-1} \hat{f}_i^{\omega}(\mu)$$

Fourier Space Bounds [Kondor et.al.]

$$B_{n \to i} = \sum_{\mu \vdash n-1} \|\hat{f}_i^{\omega}(\mu)\|_*$$

Risk Minimization

Loss Function

$$\sum_{k=1}^{n} \sum_{i \in \text{children}((n-k+1)^*)} \left[\|\hat{f}_i^{\omega}(\mu)\|_* - \|\hat{f}_{i^*_{n-k}}^{\omega}(\mu)\|_* + 1 \right]^+$$

- i^*_{n-k} is the correct node at level $n-k$ in coset tree.

Risk Minimization

Jensen's Inequality

For parameterization: $\hat{f}_i^{\omega}(\mu) = \sum_{d=1}^{D} \omega_d \hat{f}_i^d(\mu)$

$$\|\hat{f}_i^{\omega}(\mu)\|_* = \|\sum_{d=1}^{D} \omega_d \hat{f}_i^d(\mu)\|_* \leq \sum_{d=1}^{D} \omega_d \|\hat{f}_i^d(\mu)\|_*$$

Risk Minimization

Jensen's Inequality

For parameterization: $\hat{f}_i^{\omega}(\mu) = \sum_{d=1}^{D} \omega_d \hat{f}_i^{d}(\mu)$

$$\|\hat{f}_i^{\omega}(\mu)\|_* = \|\sum_{d=1}^{D} \omega_d \hat{f}_i^{d}(\mu)\|_* \leq \sum_{d=1}^{D} \omega_d \|\hat{f}_i^{d}(\mu)\|_*$$

Fourier space Stochastic Gradient Descent Solver

Each update takes the form

$$\omega_d \leftarrow \omega_d - \eta \begin{cases} \|\hat{f}_i^{d}(\mu)\|_* - \|\hat{f}_{i^*_{n-k}}^{d}(\mu)\|_* + \frac{\nu}{M\mathcal{O}(n^2)}\omega_d \\ \frac{\nu}{M\mathcal{O}(n^2)}\omega_d \end{cases}$$

Risk Minimization

Jensen's Inequality

For parameterization: $\hat{f}_i^{\omega}(\mu) = \sum_{d=1}^{D} \omega_d \hat{f}_i^d(\mu)$

$$\|\hat{f}_i^{\omega}(\mu)\|_* = \|\sum_{d=1}^{D} \omega_d \hat{f}_i^d(\mu)\|_* \leq \sum_{d=1}^{D} \omega_d \|\hat{f}_i^d(\mu)\|_*$$

Fourier space Stochastic Gradient Descent Solver

Each update takes the form

$$\omega_d \leftarrow \omega_d - \eta \begin{cases} \|\hat{f}_i^d(\mu)\|_* - \|\hat{f}_{i^*_{n-k}}^d(\mu)\|_* + \frac{\nu}{M\mathcal{O}(n^2)}\omega_d \\ \frac{\nu}{M\mathcal{O}(n^2)}\omega_d \end{cases}$$

Convergence: emulate proof for D-dimensional *Perceptron*.

Experimental Results

Setup

- **Edge:** *Delaunay triangulation* on interest points
- **Distance:** Euclidean distance between interest points
- **Shape Context (60 in all):** Similarities based on local shape-based appearance of interest points

Experimental Results

Setup

- **Edge:** *Delaunay triangulation* on interest points
- **Distance:** Euclidean distance between interest points
- **Shape Context (60 in all):** Similarities based on local shape-based appearance of interest points

Task

- Learn ω using training instances
- Solve the learnt problem “cheaply” (e.g., greedy or linear assignment)
- Evaluate compromise on accuracy?
- Evaluate improvements in running time?

Experimental Results: CMU House

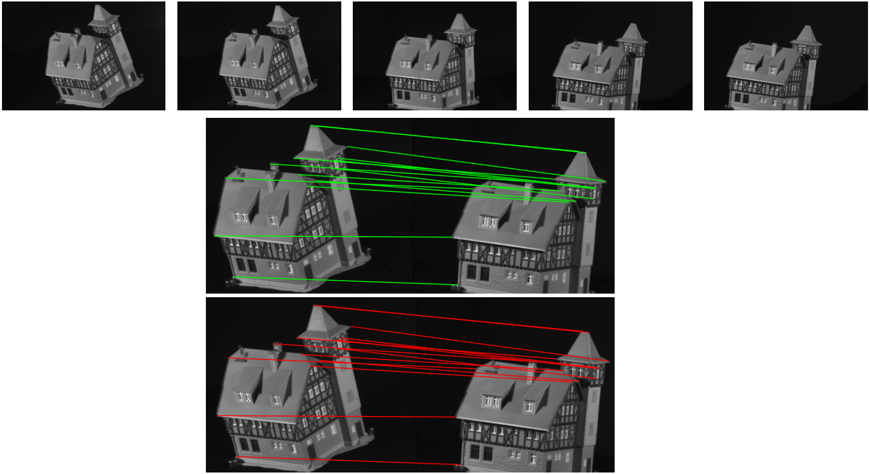


Figure: (Green) the ground truth and (red) the learnt correspondences.

Experimental Results: CMU Hotel

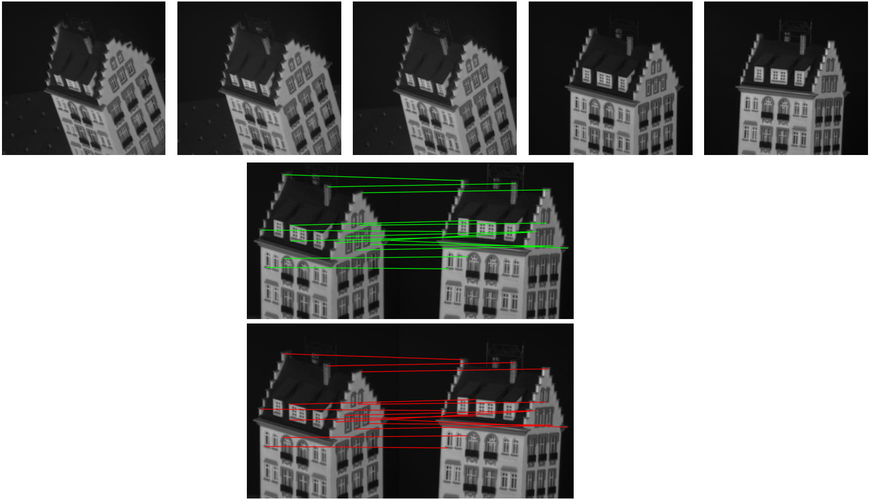
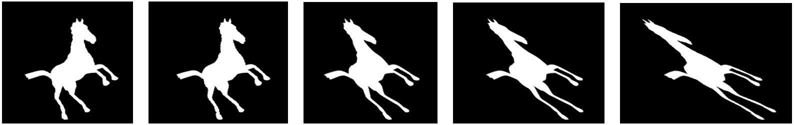


Figure: (Green) the ground truth and (red) the learnt correspondences.

Experimental Results: Silhouette



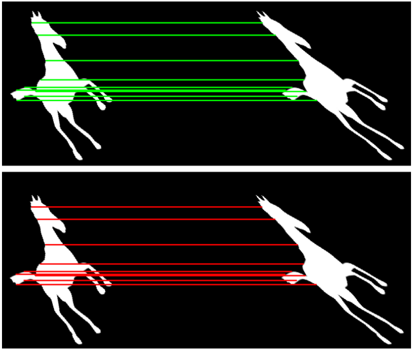


Figure: (Green) the ground truth and (red) the learnt correspondences.

Accuracy vs. Offset: CMU House

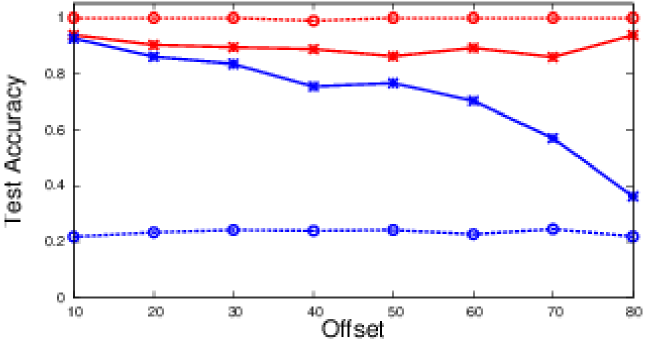


Figure: Our method compared with no-learn baseline. (Red) learning and (blue) no-learning.

Accuracy vs. Offset: CMU Hotel

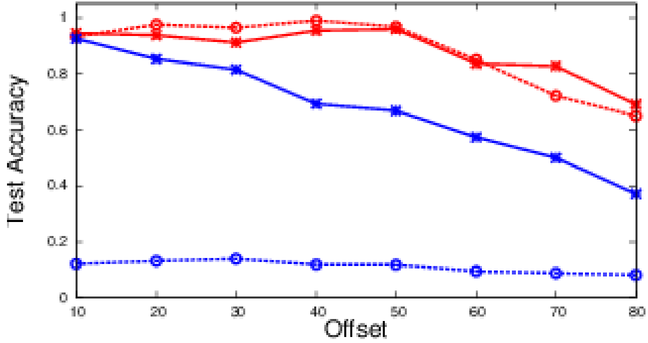


Figure: Our method compared with no-learn baseline. (Red) learning and (blue) no-learning.

Accuracy vs. Offset: Silhouette

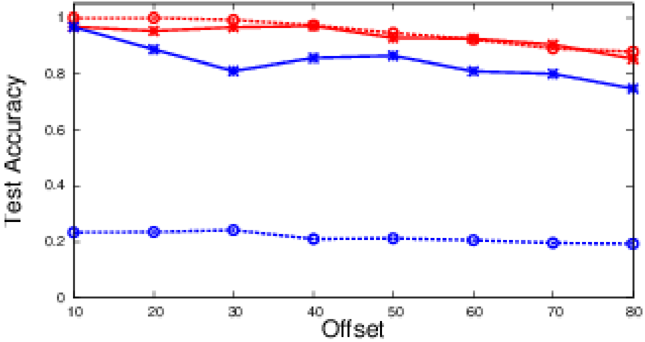


Figure: Our method compared with no-learn baseline. (Red) learning and (blue) no-learning.

Conclusions

- Incorporating domain knowledge help solving hard problems.
- Harmonic analysis provide nice structure for matching problems.
- Other parameterization schemes might provide further insights.
- Please come to the poster session. Poster 15 in Informatics Forum.

Thank You!