# Application of graph matching in Computer Vision

## Master Seminar

Ekaterina Tikhoncheva

University of Heidelberg
Faculty of Mathematics and Computer Science
Computer Vision group
at
Heidelberg Collaboratory for Image Processing

November 2015

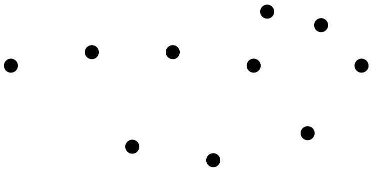
# Agenda

1. Graph matching
   - Introduction
   - Graph matching
   - Exact graph matching
   - Inexact graph matching
2. 2LevelGM
3. Evaluation
   - Synthetic data
   - Real data
4. Conclusions

# Attributed undirected graph I

Attributed undirected graph $G = (V,$

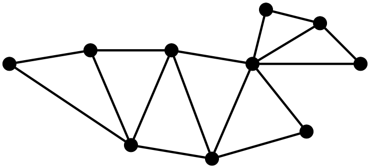
- set of nodes $V = \{v_i\}_{i=1}^{n}$



# Attributed undirected graph II

Attributed undirected graph $G = (V, E,$

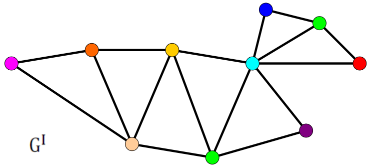
- set of nodes $V = \{v_i\}_{i=1}^n$
- set of edges $E \subseteq \{\{u, v\} | u, v \in V\}$



# Attributed undirected graph

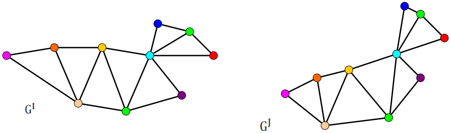
Attributed undirected graph $G = (V, E, D)$

- set of nodes $V = \{v_i\}_{i=1}^{n}$
- set of edges $E \subseteq \{\{u, v\} | u, v \in V\}$
- node attributes $D = \{d_i\}_{i=1}^{n}$

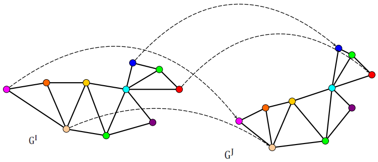


# Graph matching

Let us consider two undirected attributed graphs $G^I = (V^I, E^I, D^I)$ and $G^J = (V^J, E^J, D^J)$:



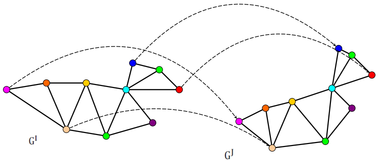
# Graph matching




A matching function between $G^I$ and $G^J$ is a mapping

$$m : V^I \to V^J$$

# Graph matching

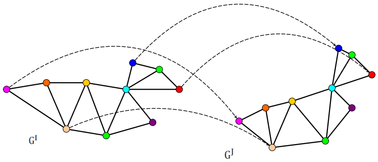



A matching function between $G^I$ and $G^J$ is a mapping

$$m : V^I \to V^J$$

not unique!

# Graph matching




A matching function between $G^I$ and $G^J$ is a mapping

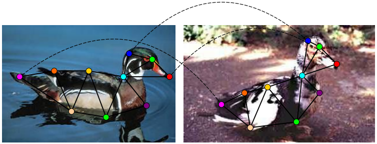
$$m : V^I \to V^J$$

Define a function $S(G^I, G^J, m)$ to measure the quality of matching $m$ that fulfills some conditions

$\Rightarrow$ Graph matching problem between $G^I$ and $G^J$

$$m = \operatorname*{argmax}_{\hat{m}} S(G^I, G^J, \hat{m})$$

# Graph matching in computer vision

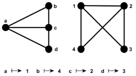


- image matching
- shape matching
- object detection
- object tracking
- ...

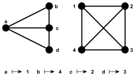
# Exact graph matching I

Edge preserving mapping $m$: $\{v_i, v_{i'}\} \in E^I \Rightarrow \{m(v), m(v_{i'})\} \in E^J$

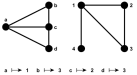
- mapping $m$ is bijective $\rightarrow$ graph isomorphism (GI)




- mapping $m$ is injective $\rightarrow$ graph monomorphism




- mapping $m$ is total $\rightarrow$ graph homomorphism

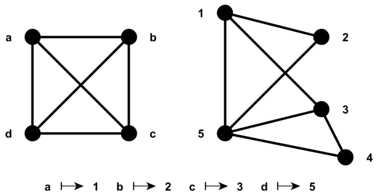



NP complete (except GI) [9]

# Exact graph matching II

Exact graph matching:

- too strict
- time/memory consuming
- cannot handle object deformation



# Inexact graph matching I

$$m = \operatorname*{argmax}_{\hat{m}} S(G^I, G^J, \hat{m})$$

- second-order (edge) similarity $s_E(e_{ii'}, e_{jj'})$, $e_{ii'} \in E^I, e_{jj'} \in E^J$
- first-order (node) similarity $s_V(v_i, v_j)$, $v_i \in V^I, v_j \in V^J$

$$S(G^I, G^J, m) = \sum_{\substack{m(v_i)=v_j \\ m(v_i')=v_j'}} s_E(e_{ii'}, e_{jj'}) + \sum_{m(v_i)=v_j} s_V(v_i, v_j)$$

- Assignment matrix $x \in \{0,1\}^{n_1 \times n_2}$, $x_{ij} = 1 \iff m(v_i) = v_j$

# Inexact graph matching II

The most common problem formulation:

## Quadratic Assignment Problem (NP complete) [3]

$$x^* = \arg\max \sum_{\substack{x_{ij}=1 \\ x_{i'j'}=1}} s_E(e_{ii'}, e_{jj'}) + \sum_{x_{ij}=1} s_V(v_i, v_j)$$

$$s.t. \begin{cases} x \in \{0,1\}^{n_1 n_2} \\ \sum_{i=1}^{n_1} x_{ij} \leq 1 \\ \sum_{j=1}^{n_2} x_{ij} \leq 1 \end{cases}$$

Using matrix notation : $\arg\max_x x^T S x$, $S-$similarity (or affinity) matrix

# Inexact graph matching III

Solution techniques [8]

- discrete optimization
  - tree search [2, 21, 22, 25]
  - simulated annealing [11]
- continuous optimization
  - constraint relaxation [10, 14, 15, 24, 26]
  - spectral methods [13, 23]
  - probabilistic frameworks [1, 12, 16, 20]
  - clustering [4, 6, 19, 17]

# Drawback of the existing algorithms

- most of them were developed for matching relative small graphs ($\sim$ 100 nodes)
- scale badly due to the polynomial increase of time and storage demand
- algorithms for the big graphs use another formulation of the graph matching optimization problem

  $$P = \operatorname{argmin}_{\hat{P} \in \Pi_n} \|A^I - \hat{P} A^J \hat{P}^T\|^2 + \|D^I - \hat{P} D^J\|_2^2$$

# Complexity reduction

$$x^* = \arg\max x^T S x$$

$$s.t. \begin{cases} x \in \{0,1\}^{n_1 n_2} \\ \sum_{i=1}^{n_1} x_{ij} \leq 1 \\ \sum_{j=1}^{n_2} x_{ij} \leq 1 \end{cases}$$

- set of candidate correspondences
- sparse affinity matrix
- subdivide problem into a set of smaller subproblems

# Complexity reduction

$$x^* = \arg\max x^T S x$$

$$s.t. \begin{cases} x \in \{0,1\}^{n_1 n_2} \\ \sum_{i=1}^{n_1} x_{ij} \leq 1 \\ \sum_{j=1}^{n_2} x_{ij} \leq 1 \end{cases}$$

- set of candidate correspondences
- sparse affinity matrix
- subdivide problem into a set of smaller subproblems $\leftarrow$

# Complexity reduction

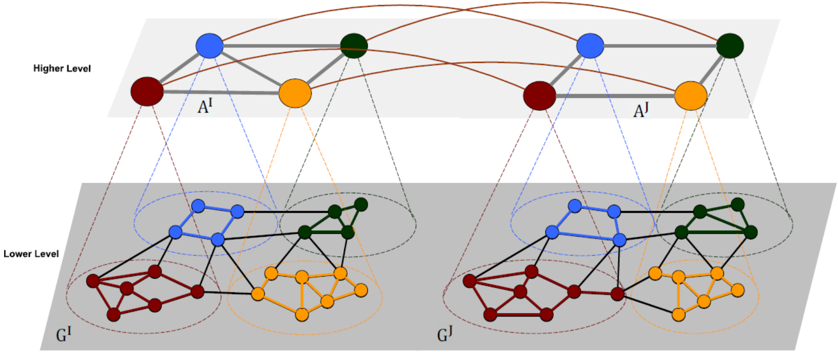
$$x^* = \arg\max x^T Sx$$

$$s.t. \begin{cases} x \in \{0,1\}^{n_1 n_2} \\ \sum_{i=1}^{n_1} x_{ij} \leq 1 \\ \sum_{j=1}^{n_2} x_{ij} \leq 1 \end{cases}$$

- set of candidate correspondences
- sparse affinity matrix
- subdivide problem into a set of smaller subproblems $\leftarrow$
  Similar works:
  - semisupervised case [17]
  - another objective function [4, 19]
  - special kind of subproblem [19, 18]

# Two level graph matching framework

Lower level: initial graphs $G^I$, $G^J$
Higher level: simplified graphs (anchor graphs $A^I$, $A^J$)



# Anchor graph construction

Goal: $G^I = (V^I, E^I, D^I) \rightarrow A^I = (V^{Ia}, E^{Ia}, U^{Ia})$
Equivalent: partitioning of $G^I \supset (G_1^I \cup \cdots \cup G^I_{|V^{Ia}|})$
Done by:

- grid with $r$ rows and $c$ columns
- graph coarsening algorithms: Heavy Edge Matching (HEM) and Light Edge Matching (LEM)

# Anchor graph and subgraph matching

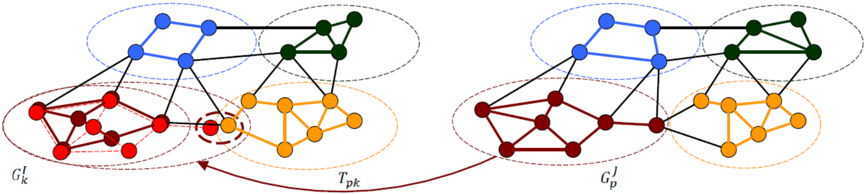
Find correspondences between two anchor graphs
$A^I = (V^{Ia}, E^{Ia}, U^{Ia})$ and $A^J = (V^{Ja}, E^{Ja}, U^{Ja})$

- edge similarity: compare length of the edges beween anchors
- node similarity:
  - score of the matching of $G^I_k$ and $G^J_p$
  - define anchor attributes based on the $D^I, D^J$ and/or on the geometry of $G^I, G^J$

Match anchor graphs and subgraph using some existing algorithm (e.g. RRWM [7])

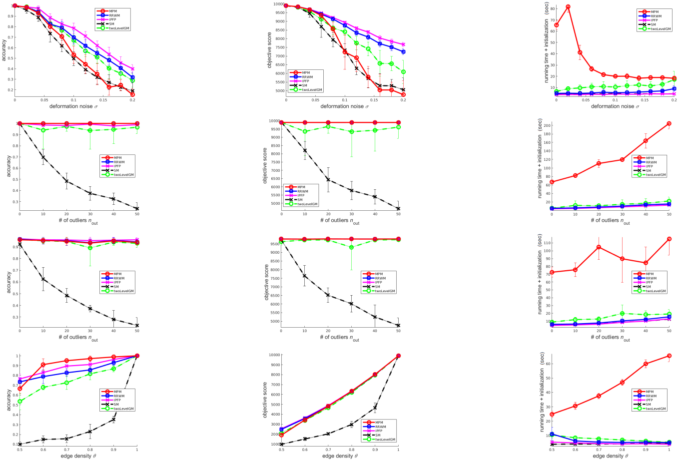
# Graph partition update

1. estimate an affine transformation between matched subgraphs (point set registration problem)
2. let nodes "vote" to which subgraphs they should belong to

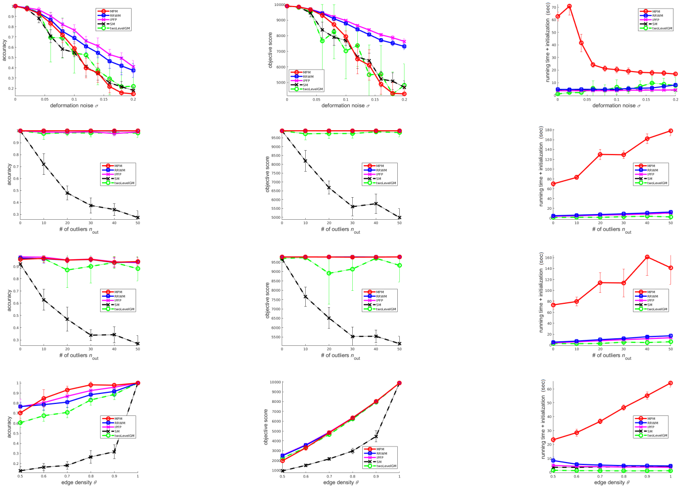


# Evaluation

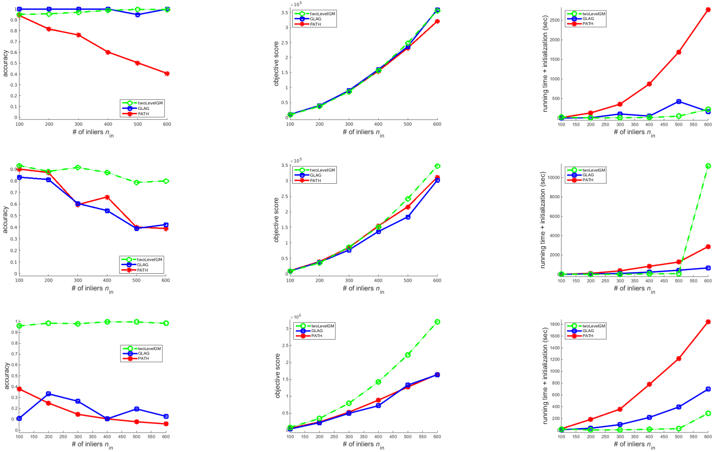
# Synthetic data I



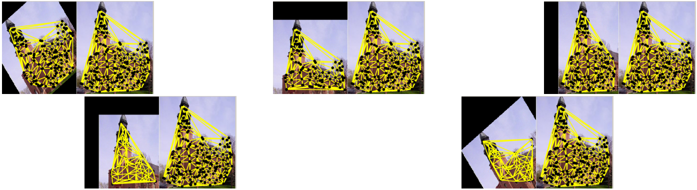
# Synthetic data II

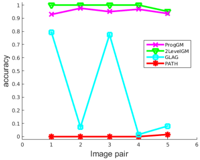


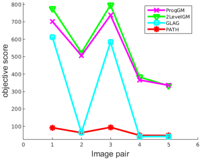
# Synthetic data III

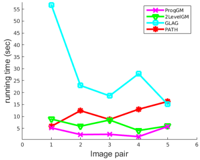


# Image affine transformation I

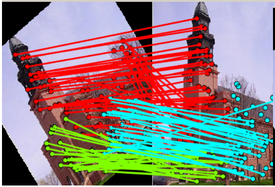




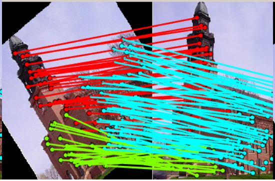



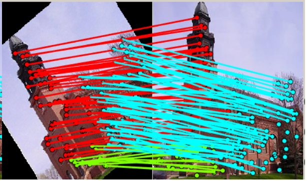
# Image affine transformation II



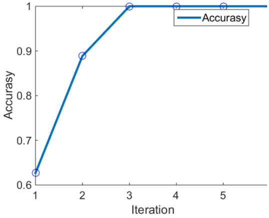
Iteration 1



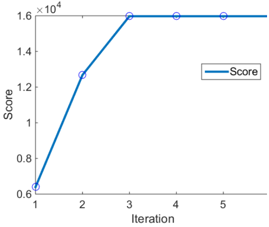
Iteration 2



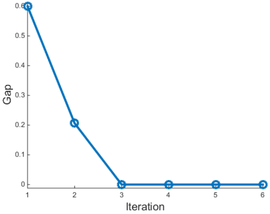
Iteration 3




Accuracy

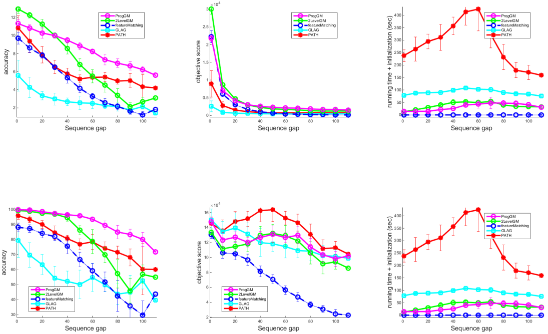


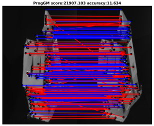

Matching score



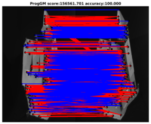

Gap between current and optimal solutions

# House data set I




accuracy 11.63%




accuracy 100%

# Conclusions

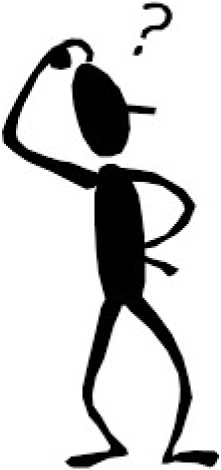
- inexact graph matching problem
- application of existing algorithms to bigger graphs
- very good results for (sub)graph isomorphism problems
- sensible to the deformations, but not more then other algorithms
- good results in case of affine deformations
- troubles with non-affine transformations
- complexity depends on the number of iterations and on the size of the anchor graphs and subgraphs
- fast
- anchor attributes for anchor graph matching are preferred

# Future work

- more sophisticated graph partitioning techniques
- improvement of anchor attributes
- further improvement of the update rule
- probabilistic matching framework
- hierarchical method

# The end

Thank you for your attention!



# References I

[1] Ayser Armiti and Michael Gertz. Geometric graph matching and similarity: A probabilistic approach. In *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*, SSDBM '14, pages 27:1–27:12, New York, NY, USA, 2014. ACM.

[2] H Bunke and G Allermann. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245–253, 1983.

[3] R. E. Burkard, E. Çela, P. M. Pardalos, and L. Pitsoulis. The quadratic assignment problem. In P. M. Pardalos and D.-Z Du, editors, *Handbook of Combinatorial Optimization*, pages 241–338. Kluwer Academic Publisher, 1998.

# References II

[4] Marco Carcassoni and Edwin R. Hancock. Correspondence matching with modal clusters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(12):1609–1615, 2003.

[5] Minsu Cho, Karteek Alahari, and Jean Ponce. Learning Graphs to Match. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[6] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Feature Correspondence and Deformable Object Matching via Agglomerative Correspondence Clustering. In *The IEEE International Conference on Computer Vision (ICCV)*, 2009.

[7] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted Random Walks for Graph Matching. *ECCV*, 2010.

# References III

[8] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty Years of Graph Matching in Pattern Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.

[9] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[10] S. Gold and Anand Rangarajan. A Graduated assignment algorithm for graph matching. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 18, pages 377–388, 1996.

# References IV

[11] L. H'erault, R. Horaud, F. Veillon, and J. Niez. Symbolic image matching by simulated annealing. In *Procedings of the British Machine Vision Conference*, pages 319–324, 1990.

[12] J. Kittler and E. R. Hancock. International journal of pattern recognition and artificial intelligence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 3(1):29–51, 1989.

[13] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005.

[14] Marius Leordeanu, Martial Hebert, Rahul Sukthankar, and Martial Herbert. An Integer Projected Fixed Point Method for Graph Matching and MAP Inference. In *NIPS*, 2009.

# References V

[15] Yao Lu, Kaizhu Huang, and Cheng-Lin Liu. A fast projected fixed-point algorithm for large graph matching, 2012. Available at `http://arxiv.org/abs/1207.1114v3`, last access on 17/10/2015.

[16] Bin Luo and Edwin R. Hancock. Structural graph matching using the EM algorithm and singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(10):1120–1136, 2001.

# References VI

[17] Vince Lyzinski, Daniel L Sussman, Donniell E Fishkind, Henry Pao, Li Chen, Joshua T Vogelstein, Youngser Park, and Carey E Priebe. Spectral Clustering for Divide-and-Conquer Graph Matching. 2011/14. Available at `http://arxiv.org/abs/1310.1297v5`, last access on 17/10/2015.

[18] Wei-Zhi Nie, An-An Liu, Zan Gao, and Yu-Ting Su. Clique-graph Matching by Preserving Global & Local Structure. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[19] Huaijun Qiu and Edwin R. Hancock. Graph matching and clustering using spectral partitions. *Pattern Recognition*, 39(1):22–34, 2006.

# References VII

[20] Gerard Sanromà, René Alquézar, and Francesc Serratosa. A new graph matching method for point-set correspondence using the EM algorithm and Softassign. *Computer Vision and Image Understanding*, 116:292–304, 2012.

[21] L G Shapiro and R M Haralick. Structural descriptions and inexact matching. *IEEE transactions on pattern analysis and machine intelligence*, 3(5):504–519, 1981.

[22] Wen-Hsiang Tsai and King-Sun Fu. Error-Correcting Isomorphisms of Attributed Relational Graphs for Pattern Analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(12):757–768, 1979.

# References VIII

[23] Shinji Umeyam. An Eigendecomposition Approach to Weighted Graph Matching Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10, 1988.

[24] J. T. Vogelstein, J. M. Conroy, L. J. Podrazik, S. G Kratzer, E. T. Harley, D. E. Fishkind, R. J. Vogelstein, and C. E. Priebe. Large (Brain) Graph Matching via Fast Approximate Quadratic Programming, 2011/14. Available at `http://arxiv.org/abs/1112.5507v5`, last access on 17/10/2015.

[25] Jason T.L. Wang, Kaizhong Zhang, and Gung-Wei Chirn. Algorithms for approximate graph matching. *Information Sciences*, 82(1-2):45–74, 1995.

# References IX

[26] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A Path Following Algorithm for the Graph Matching Problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2227–2242, 2009.