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

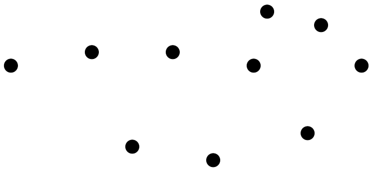
November 2015

Agenda

Attributed undirected graph I

Attributed undirected graph $G = (V,$

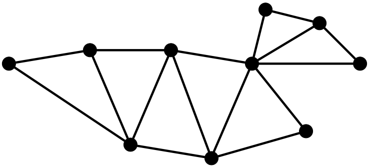
- set of nodes $V = \{v_i\}_{i=1}^{n}$



Attributed undirected graph II

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

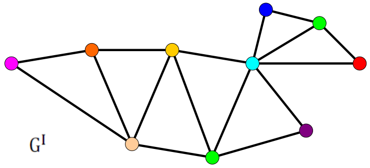
- set of nodes $V = \{v_i\}_{i=1}^{n}$
- set of edges $E \subseteq \{\{u, v\} | u, v \in V\}$



Attributed undirected graph

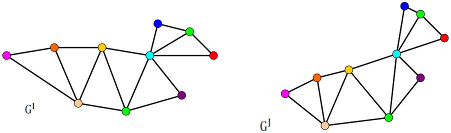
Attributed undirected graph $G = (V, E, D)$

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

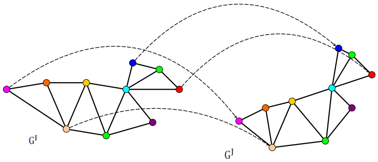


Graph matching

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



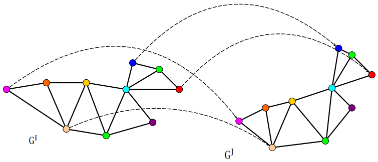
Graph matching



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

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

Graph matching

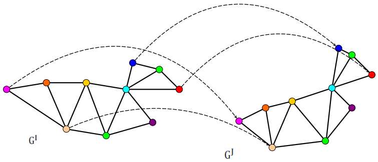


A matching function between G^I and G^J is a mapping

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

not unique!

Graph matching



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

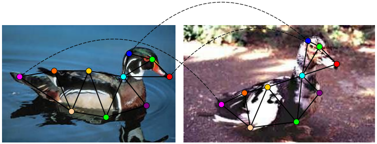
$$m : V^I \to V^J$$

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

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

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

Graph matching in computer vision



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

Graph matching

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

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

Graph matching problem between G^I and G^J

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

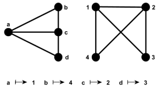
Depending on the required properties of a matching one distinguishes

- exact graph matching
- inexact graph matching

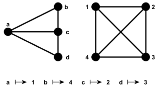
Exact graph matching I

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

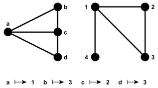
- mapping m is bijective $\rightarrow$ graph isomorphism (GI)



- mapping m is injective $\rightarrow$ graph monomorphism



- mapping m is total $\rightarrow$ graph homomorphism

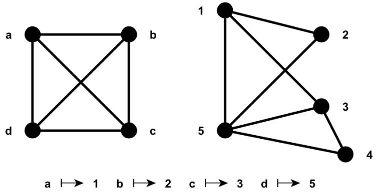


NP complete (except GI) [Garey and Johnson, 1979]

Exact graph matching II

Exact graph matching:

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



Inexact graph matching I

Introduce similarity measure between nodes/edges in the graphs

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

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

- second-order (edge) similarity $s_E(e_{ii'}, e_{jj'})$, $e_{ii'} \in E^I, e_{jj'} \in E^J$
- first-order (node) similarity $s_V(v_i, v_j)$, $v_i \in V^I, v_j \in V^J$
- assignment matrix $X \in \{0,1\}^{n_1 \times n_2}$, $X_{ij} = 1 \iff m(v_i) = v_j$
  vector form $x = \mathrm{vec}(X)$

Inexact graph matching II

The most common problem formulation:

Quadratic Assignment Problem (NP complete) [Burkard et al., 1998]

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

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

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

Inexact graph matching III

Solution techniques [Conte et al., 2004]

- discrete optimization
  - tree search [Bunke and Allermann, 1983, Shapiro and Haralick, 1981, Tsai and Fu, 1979, Wang et al., 1995]
  - simulated annealing [Hérault et al., 1990]
- continuous optimization
  - constraint relaxation [Gold and Rangarajan, 1996, Leordeanu et al., 2009, Lu et al., 2012, Vogelstein et al., 2011/14, Zaslavskiy et al., 2009]
  - spectral methods [Leordeanu and Hebert, 2005, Umeyam, 1988]
  - probabilistic frameworks [Armiti and Gertz, 2014, Kittler and Hancock, 1989, Luo and Hancock, 2001, Sanromà et al., 2012]
  - clustering [Carcassoni and Hancock, 2003, Cho et al., 2009, Qiu and Hancock, 2006, Lyzinski et al., 2011/14]

Drawback of the existing algorithms

- most of the algorithms are developed for matching relatively small graphs ($\sim$ 150 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

$$x^* = \mathrm{argmin}_X \|A^I - XA^J X^T\|^2 + \|D^I - XD^J\|_2^2$$

Aim of the master's thesis

- a novel framework that should help to extend the usability of existing graph matching algorithms to bigger graphs
- a variant of the well-known divide and conquer paradigm: subdividing initial problem into a set of smaller problems, which can be easily handled with existing algorithms
- iterative algorithm that allows to improve initial subdivision into 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 [Cho and Lee, 2012]
- 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 [Cho and Lee, 2012]
- sparse affinity matrix
- subdivide problem into a set of smaller subproblems $\leftarrow$

Complexity reduction

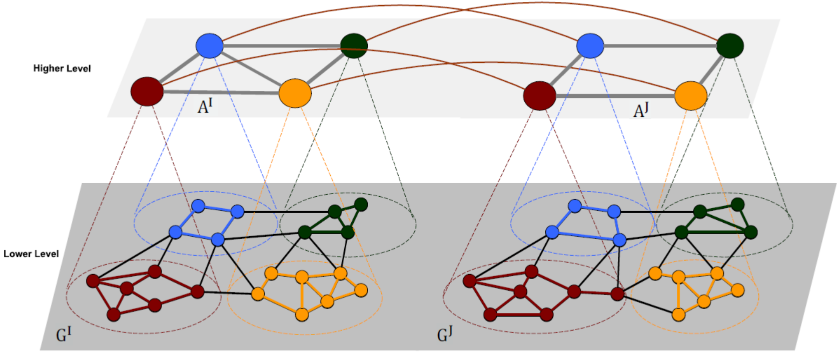
$$x^* = \arg\max x^T S x$$

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

- set of candidate correspondences [Cho and Lee, 2012]
- sparse affinity matrix
- subdivide problem into a set of smaller subproblems ← Similar works:
  - semisupervised case [Lyzinski et al., 2011/14]
  - another objective function [Lyzinski et al., 2011/14, Carcassoni and Hancock, 2003, Qiu and Hancock, 2006]
  - special kind of subproblem [Qiu and Hancock, 2006, Nie et al., 2015]

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: $G^l = (V^l, E^l, D^l) \rightarrow A^l = (V^{la}, E^{la}, U^{la})$
Equivalent: partitioning of $G^l \supset (G_1^l \cup \cdots \cup G_{|V^{la}|}^l)$
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

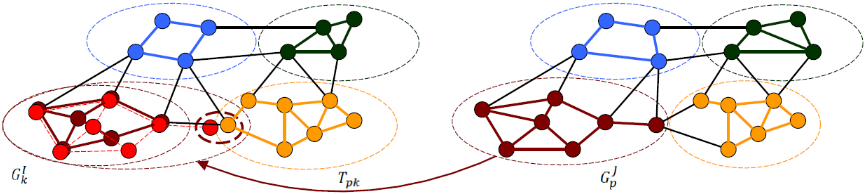
Goal: find correspondences between two anchor graphs
$A^I = (V^{Ia}, E^{Ia}, U^{Ia})$ and $A^J = (V^{Ja}, E^{Ja}, U^{Ja})$

- edge similarity: compare length of the edges beween anchors
- node similarity:
    - score of the matching of G_k^I and G_p^J
    - 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 [Cho et al., 2010])

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

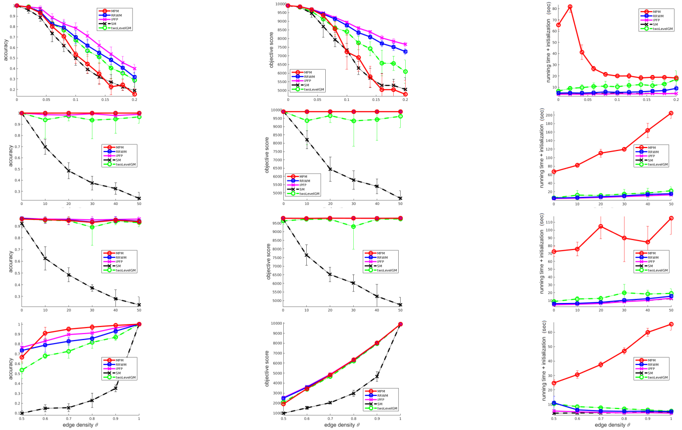


Evaluation

Two ways of evaluation have been used

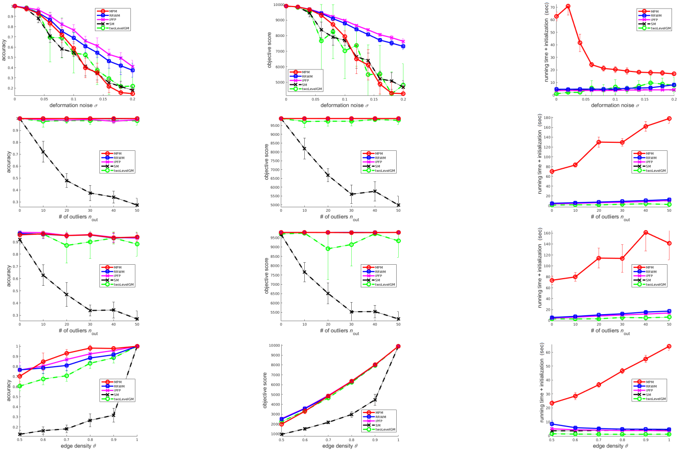
- synthetic data
- real data

Synthetic data I



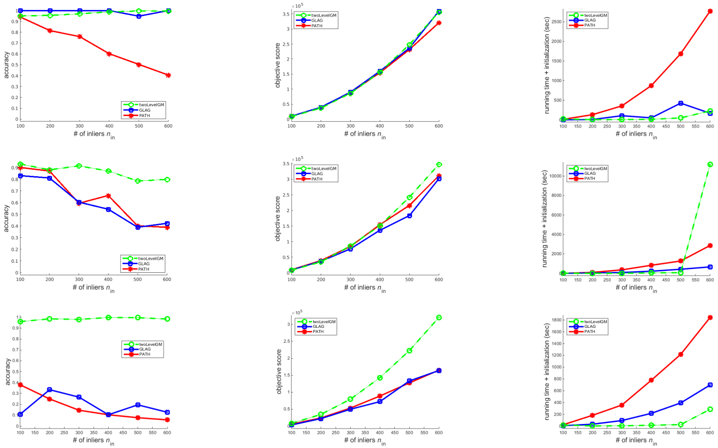
Performance of the 2LevelGM with non-attributed anchor graphs

Synthetic data II



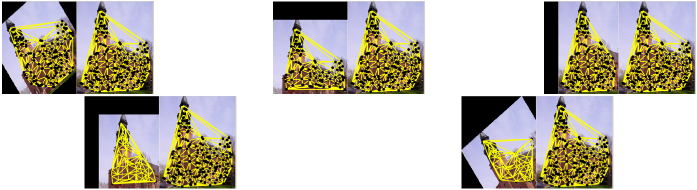
Performance of the 2LevelGM with attributed anchor graphs

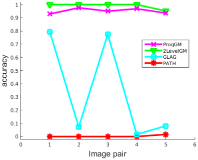
Synthetic data III

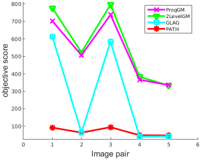


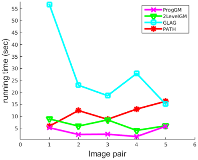
Comparison of 2LevelGM, GLAG and PATH on bigger graphs

Image affine transformation I



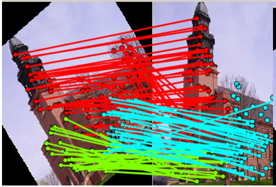




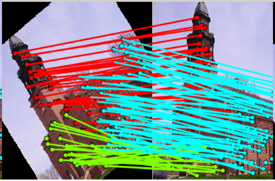


Evaluation of 2LevelGM on the synthetic image dataset

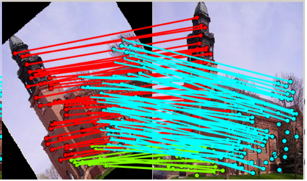
Image affine transformation II



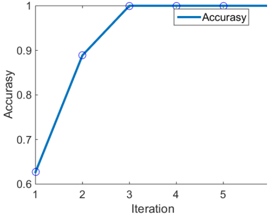
Iteration 1



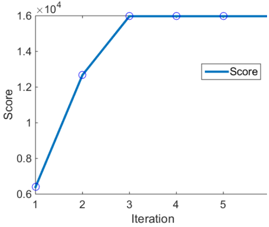
Iteration 2



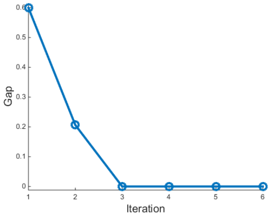
Iteration 3



Accuracy

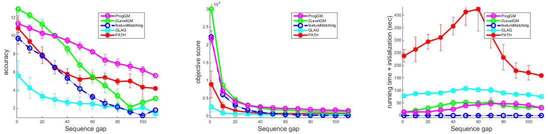


Matching score

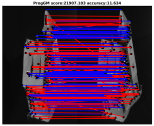


Gap between current and optimal solutions

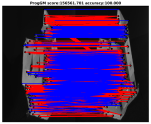
House data set



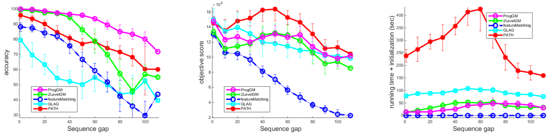
Evaluation of 2LevelGM on the CMU House sequence: not extrapolated solution



accuracy 11.63%



accuracy 100%



Evaluation of 2LevelGM on the CMU House sequence: extrapolated solution

Conclusions

The developed algorithm

- solves inexact graph matching problem
- is fast
- allows application of existing algorithms to bigger graphs
- shows very good results for (sub)graph isomorphism problems
- handles reasonably the affine deformations in graph structure

It has following properties:

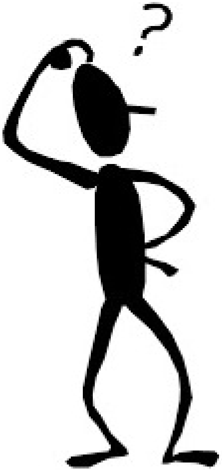
- it is sensible to non-affine deformations in graph structure
- it's complexity depends on the number of iterations and on the size of the anchor graphs and subgraphs
- anchor attributes for anchor graph matching are preferred

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

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. URL `http://doi.acm.org/10.1145/2618243.2618259`.

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

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

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

Minsu Cho and Kyoung Mu Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12. IEEE Computer Society, 2012.

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.

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

References III

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

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.

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

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

L. Hérault, 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.

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.

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

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

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.

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.

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.

References VI

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.

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

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.

References VII

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.

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.

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

References VIII

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.

Jason T.L. Wang, Kaizhong Zhang, and Gung-Wei Chirn. Algorithms for approximate graph matching. *Information Sciences*, 82(1-2):45–74, 1995. URL `http://www.sciencedirect.com/science/article/pii/002002559400057I`.

References IX

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.