Unified Graph Matching in Euclidean Spaces

Julian J. McAuley
NICTA/ANU
Canberra, ACT 0200
julian.mcauley@nicta.com.au

Teófilo de Campos*
University of Surrey
Guildford, GU2 7XH, UK
t.decampos@st-annes.oxon.org

Tibério S. Caetano
NICTA/ANU
Canberra, ACT 0200
tiberio.caetano@nicta.com.au

Abstract

Graph matching is a classical problem in pattern recognition with many applications, particularly when the graphs are embedded in Euclidean spaces, as is often the case for computer vision. There are several variants of the matching problem, concerned with isometries, isomorphisms, homeomorphisms, and node attributes; different approaches exist for each variant. We show how structured estimation methods from machine learning can be used to combine such variants into a single version of graph matching. In this paradigm, the extent to which our datasets reveal isometries, isomorphisms, homeomorphisms, and other properties is automatically accounted for in the learning process so that any such specific qualification of graph matching loses meaning. We present experiments with real computer vision data showing the leverage of this unified formulation.

1. Introduction

Graphs are typically used as high-level models for images, meaning that identifying a correspondence between their nodes (popularly called a 'matching') is a fundamental problem in computer vision. Applications range from object and character recognition [1, 2], to 3-D scene reconstruction [3]. In this paper we are strictly concerned with graphs embedded in Euclidean spaces, i.e., graphs whose nodes encode point coordinates.

Many types of graphs have been considered in the graph matching literature, depending on how one models a given matching problem. Perhaps the simplest setting arises when the graphs are represented as sets of nodes and edges and the goal is to find an *isomorphism* between them [4] (or more generally, a *subgraph* isomorphism, to allow for outlying nodes). Also of interest is the *homeomorphism* problem, which allows for edges to be 'split' by additional nodes, so long as the topology is preserved [5]. For graphs embedded in Euclidean spaces, one is additionally interested in the notion of *isometry*: two graphs are isometric if the corresponding distances between nodes are preserved [6]. Figure 1 illustrates the aspects of isometry, isomorphism and homeomorphism. In addition to these notions, the concept of *attributed* nodes and edges gives rise to the notion of attributed graph matching, i.e., one is interested in comparing specific abstract features encoded by nodes and/or edges of the graph (e.g. nodes can encode colors of image regions, edges can encode adjacency of objects, etc.).

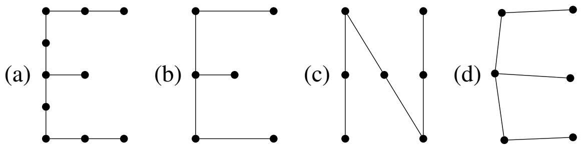


Figure 1. Examples of the transformations typically observed in computer vision scenarios. Isometric: $b \subseteq a$, $b \subseteq c$. Isomorphic: $b \subseteq d$, $c \subseteq a$, $d \subseteq b$. Homeomorphic: $b \subseteq a$, $b \subseteq d$, $c \subseteq a$, $d \subseteq a$, $d \subseteq b$. The notation $x \subseteq y$ allows for 'outliers' in y.

In the literature, these different types of formulations for graph matching usually come with different types of models and algorithms. The purpose of the present paper is to combine these hard notions of different types of graph matching problem into a single concept. Specifically, we focus on combining the properties of isometry, isomorphism, homeomorphism and node attributes into a single model which can not only be solved efficiently and exactly for graphs embedded in Euclidean spaces but is also provably optimal in the absence of noise. We further compare our unified model with recently proposed ones, which make specific assumptions about the type of matching problem at hand, and we find that ours is able to better leverage the fact that real data does not come exactly as isometries, homeomorphisms or isomorphisms: it is simply real data.

1.1. Related Work

We shall be interested in comparing the results of our model with some well-known models that make specific as-

[*] Julian McAuley and Teófilo de Campos were at Xerox Research Centre Europe at the time of this work.

sumptions about the type of graph matching problem being solved. The first are methods based exclusively on *node attributes*. These methods do not consider the relational aspects between nodes; such approaches typically consist of minimizing a first-order objective function, i.e., points in the template should be mapped to similar points in the target, but relationships between points are not considered. Examples include 'shape-context' matching [2], 'inner distance' matching [7], and the 'longest common subsequence' algorithm [8].

Other approaches consider the *distances* between nodes in the graphs, in order to solve the problem of *isometric matching* [6].

Finally, by expressing the matching problem using a quadratic assignment objective, it is trivial to model topological constraints, though the resulting optimization problem is in general NP-complete. As such, a great body of work has been focused on producing efficient and accurate approximations. Examples include 'spectral methods' [9, 10], 'probabilistic approaches' [11, 12], and 'graduated assignment' [13].

2. The Model

2.1. Isometry

Suppose we have a 'template' graph $\mathcal{G}$ with nodes and edges (V, E), which we wish to identify in a 'target' graph $\mathcal{G}' = (V', E')$ (we allow that $|V'| \geq |V|$, so that there may be outliers in G'). The isometric matching problem consists of finding a mapping $g : V \to V'$ which preserves the distances between pairs of nodes in $\mathcal{G}$, i.e.,

$$g = \operatorname*{argmin}_{f:V \to V'} \sum_{(p_1,p_2) \in V^2} \left| d(p_1, p_2) - d(f(p_1), f(p_2)) \right|, \qquad (1)$$

where $d(\cdot, \cdot)$ is our distance metric.[1] Note that the 'topologies' of our graphs, defined by E and E', are ignored by this objective function.

It is shown in [14, 15] that in the case of exact isometric matching, one need not consider *all* edges in V^2, but only a subset of edges that constitute a 'globally rigid' graph: by definition, preserving the distances of the edges in such a subgraph implies that the distances between all edges in the *complete* graph will be preserved. Thus, for a globally rigid subgraph $\mathcal{R} = (V, E^{\mathcal{R}})$ of $\mathcal{G}$, we need to solve

$$g = \operatorname*{argmin}_{f:V \to V'} \sum_{(p_1,p_2) \in E^{\mathcal{R}}} \left| d(p_1, p_2) - d(f(p_1), f(p_2)) \right|. \qquad (2)$$

If the graph $\mathcal{R}$ has small maximal cliques, (eq. 2) can be modelled as inference in a tractable graphical model (whose nodes and assignments correspond to points in V and V' respectively); [14] reports running times and memory requirements of $O(|V||V'|^{n+1})$, where n is the number of dimensions. We use a similar topology which replaces the 'ring' structure from [14] with a 'junction-tree', which has the advantage that only a single iteration is required for exact inference [15]. The topology of this graph is shown in Figure 2 (for $n = 2$), and it is rigid by Theorem 2.1.

Theorem 2.1. *The graph in Figure 2 is globally rigid when embedded in* $\mathbb{R}^2$.

Proof. The claim is trivially true for $k \leq 3$ since all edges are included. For $k > 3$, assume that the positions of $V_1 \ldots V_k$ are determined. If the distances from V_{k-1} and V_k to V_{k+1} are known, then V_{k+1} may occupy precisely two positions. The two positions result in an opposite sign for $\det\begin{bmatrix} V_{k-1} - V_{k+1} \\ V_k - V_{k+1} \end{bmatrix}$ (note that the points V_k are 2-D coordinates, so that this is a 2×2 matrix). If the correct sign is determined by the variable M, then the position of V_{k+1} is uniquely determined. The proof follows by induction. □

Theorem 2.1 generalizes to $\mathbb{R}^n$ by increasing the clique size of the graph to $n + 1$.

2.2. Isomorphism/Homeomorphism

To model isomorphisms and homeomorphisms, the optimization problem we would like to solve becomes

$$g = \operatorname*{argmin}_{f:V \to V'} \sum_{(p_1,p_2) \in E^{\mathcal{R}}} \left| d(p_1, p_2) - d(f(p_1), f(p_2)) \right| + \sum_{(p_1,p_2) \in E} \left| h(p_1, p_2) - h(f(p_1), f(p_2)) \right|. \qquad (3)$$

Note that in the second term, the function $h(\cdot, \cdot)$ is used. If we want to solve the subgraph *isomorphism* problem, h simply indicates the presence or absence of an edge in E or E' (i.e., $h(f(p_1), f(p_2)) = I_{E'}((f(p_1), f(p_2))))$. Alternately, in the case of the subgraph *homeomorphism* problem, h is just the distance d in $\mathcal{G}$, but becomes the *shortest-path distance* between $f(p_1)$ and $f(p_2)$ in $\mathcal{G}'$. We observe a value of zero in (eq. 3) exactly when the mapping is isometric (by Theorem 2.1), and isomorphic/homeomorphic (as every topological constraint has been included).

This problem can also be solved using a tractable graphical model, so long as E is a subset of $E^{\mathcal{R}}$. To this end, we make the straightforward observation that we can augment $\mathcal{R}$ by 'copying' some of its nodes, as in Figure 3 (producing $\mathcal{G}^\sharp = (V^\sharp, E^{\mathcal{R}^\sharp})$). The resulting graph remains globally rigid, and obtains the correct solution due to the following Theorem:

Theorem 2.2. *If additional replicates of some nodes are added to the model, each of the replicates will map to*

[1] For the case of *exact* matching, we could use the indicator function $1 - I_{\{d(p_1,p_2)\}}(d(f(p_1), f(p_2)))$; we instead use the difference to allow for some 'noise' in the point coordinates.

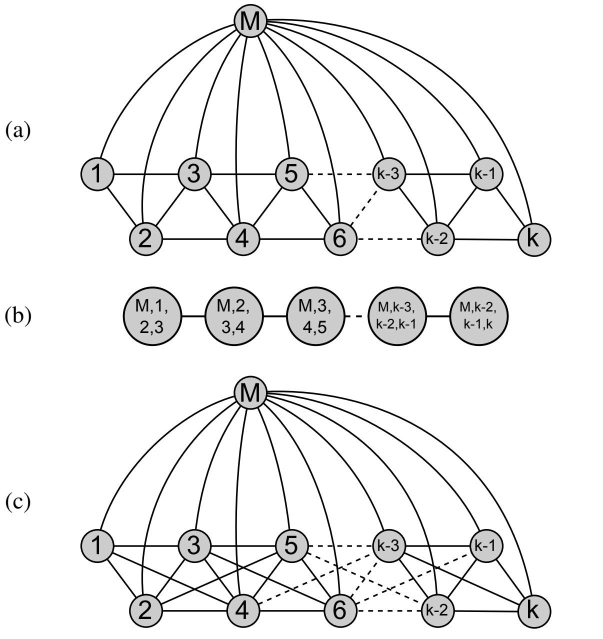


Figure 2. (a) The graphical model used in our experiments (top), and (b) its clique-graph. M is a boolean variable indicating whether or not the template pattern appears reflected in the target. Although the graph has maximal cliques of size four, the running time is cubic in $|V'|$ (for the case of 2-D isometric matching), since the node M is boolean. The model in (a) handles isometric transformations, however we could obtain scale invariance (for example) by inserting an additional constraint, as shown in (c); the size of the maximal cliques grows with the number of parameters in the transformation we wish to handle.

the same solution (assuming that an isometric isomorphism/homeomorphism exists).

Proof. Assume the image of $V_1 \ldots V_k$ corresponds to an isometric isomorphism/homeomorphism under the mapping f (and thus we can determine M). This mapping will have zero cost according to our potential function. Suppose V_{k+1} is a copy of a previous node V_j. Given the values of V_{k-1}, V_k, and M, we know from Theorem 2.1 that there is *at most one* value for $f(V_{k+1})$ which incurs zero cost. Since $f(V_{k+1}) = f(V_j)$ incurs zero cost, the minimum cost solution for the model $M, V_1 \ldots V_{k+1}$ must have $f(V_{k+1}) = f(V_j)$. Again, the proof follows by induction (the 'base case' is trivial, since there are no replicates). □

In order to add all of the edges in E, we will increase the size of $V^\sharp$ to $O(|V| + |E|)$.[2] At this point we simply solve

$$g = \operatorname*{argmin}_{f:V\to V'} \sum_{(p_1,p_2)\in E^{\mathcal{R}^\sharp}} \left|d(p_1,p_2) - d(f(p_1),f(p_2))\right| + \left|h(p_1,p_2) - h(f(p_1),f(p_2))\right|, \quad (4)$$

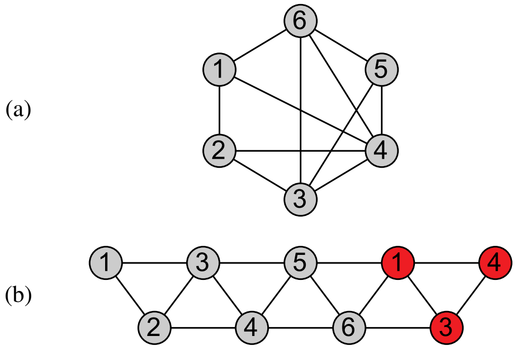


Figure 3. The topology of this graph (top) is not captured by the graphical model from Figure 2. By repeating some of the nodes in the graphical model (highlighted in red), we can include all of the topological constraints; the augmented model will still recover the correct solution (the variable M is suppressed for readability).

which can be done in $O((|V| + |E|)|V'|^{n+1})$ time and space. Assuming the topology E is sparse, this will typically be no worse than $O(|V||V'|^{n+1})$ in practice.

It is worth briefly mentioning that this model can be applied under transformations besides isometries, for instance to the problem of subgraph isomorphism/homeomorphism under perspective projection or affine transformation; this idea is demonstrated in Figure 2. Following the analysis of [16], the number of free parameters in the transformation determines the maximal clique size required in our model. Our experiments are concerned with isometries in 2-D images, which require maximal cliques of size three, resulting in an asymptotic complexity of $O((|V| + |E|)|V'|^3)$. Essentially, we are making a trade-off between the problem classes we wish to handle exactly, versus the computational complexity we can afford. Our third-order model is 'optimal' in the case of isometric transformations, and we shall show in Section 4 that it provides accurate results even as our model assumptions become violated (e.g. under affine transformations and high noise). The low tree-width of our model allows inference to be done more quickly than existing quadratic-assignment solvers, a result that could not be obtained with a more complex model.

3. Parametrization and Structured Learning

We have shown that in the case of *exact* matching, it is sufficient to attribute only the edges, using the two 'features' in (eq. 4). Of course, in the case of *near*-exact matching, we may achieve much better performance if we incorporate first-order features such as Shape-Context or SIFT [2, 17]. Since our graphical model contains 3-cliques, we can also include third-order features, ensuring (for example) preservation of angles and similarity of triangles.

In the most general setting, we can parametrize $\mathcal{G}^\sharp$ as

[2] Finding the most efficient construction of this graph is a difficult problem, however finding a construction no larger than $|V| + 2|E|$ is trivial.

follows:

$$g = \operatorname*{argmin}_{f:V\to V'} \sum_{p_1\in\mathcal{G}^\sharp} \left\langle \underbrace{\Phi^1(p_1, f(p_1))}_{\text{node features}}, \theta^{\text{nodes}} \right\rangle + \sum_{(p_1,p_2)\in\mathcal{G}^\sharp} \left\langle \underbrace{\Phi^2(p_1, p_2, f(p_1), f(p_2))}_{\text{edge features}}, \theta^{\text{edges}} \right\rangle + \sum_{(p_1,p_2,p_3)\in G^\sharp} \left\langle \underbrace{\Phi^3(p_1, p_2, p_3, f(p_1), f(p_2), f(p_3))}_{\text{triangle features}}, \theta^{\text{tri}} \right\rangle . \quad (5)$$

In order to apply *Structured Learning* [18], we have two requirements: the model must be parametrized linearly (which is satisfied by $\Theta = \left(\theta^{\text{nodes}}; \theta^{\text{edges}}; \theta^{\text{tri}}\right)$), and the so-called 'column-generation' procedure must be tractable. To satisfy the latter requirement, we specify a loss function $\Delta(\hat{g}, g)$ (the cost of choosing the mapping $\hat{g}$ when the correct mapping is g), which decomposes over the nodes in our model; the simplest example is the (normalized) Hamming loss:

$$\Delta(\hat{g}, g) = 1 - \frac{1}{|V'|} \sum_{p\in V'} I_{\{g(p)\}}(\hat{g}(p)). \quad (6)$$

In the majority of our experiments, nodes are attributed using shape-context features. Edges are attributed using distances (the left-hand-side of (eq. 4)), and topological features (the right-hand-side (eq. 4)); we use *both* the indicator (for isomorphisms) *and* the shortest-path distance (for homeomorphisms) – thus our model is in effect *learning* the extent to which the mapping is isomorphic or homeomorphic. 3-cliques are parametrized by preservation of angles, triangle similarity, and an 'occlusion' feature indicating whether two or more nodes in a clique map to the same point.

The goal of structured learning is merely to find the parameter vector Θ which minimizes our regularized risk:

$$\hat{\Theta} = \operatorname*{argmin}_{\Theta} \underbrace{\frac{1}{N} \sum_{i=1}^{N} \Delta(\hat{g}^i, g^i)}_{\text{empirical risk}} + \underbrace{\frac{\lambda}{2} \|\Theta\|_2^2}_{L_2 \text{ regularizer}} , \quad (7)$$

where $g^1 \dots g^N$ is our training set, and λ is a regularization constant. The 'Bundle Methods for Risk Minimization' (BMRM) solver of [19] is used to minimize (a convex upper bound on) this risk. See [18] for more details.

4. Experiments

We replicate several existing graph-matching experiments [20] and [6] for which topological information was provided, and add new experiments on real and synthetic point sets. The techniques that we shall compare can be summarized as follows:

Linear Assignment (from [20]) can be optimized exactly, but only includes node attributes (i.e., no topological information can be included).

Quadratic Assignment (from [20]) can include topological information, but optimization can only be done approximately (due to NP-completeness).

Isometric Matching (from [6]) can include *some* topological information, and can be optimized exactly;[3] is provably optimal for the isometric point-pattern matching problem.

Unified Matching (the proposed approach) can include *all* topological information, and can be optimized exactly; is provably optimal for the isometric isomorphism/homeomorphism problem (thus unifying the above approaches).

4.1. Proof of Concept

For our first experiment, we created a simple point-pattern which includes topological constraints (shown in Figure 5(a), center). In the target scene, several copies of this pattern were randomly (isometrically) transformed, and noise was applied to their topologies (while ensuring that exactly one isomorphic instance existed in the target scene).[4] This represents a problem that cannot be solved by first-order approaches, as there would be many spurious copies of the template shape if the topological information were ignored. Finally, uniform noise was applied to the point coordinates in the target scene.

Note that in this experiment, we use only those features described in Section 2.2 (i.e., there are no node features). Thus we confirm our claim that the problem can be solved using only these features.

In Figure 5(a), we compare our model to quadratic assignment. Although both methods solve the problem close to exactly when there is only a single copy of the template scene in the target ('no outliers'), we found that quadratic assignment was no better than random as soon as outliers were introduced (i.e., $\Delta \simeq 1$; each outlying graph contributes 19 nodes to the scene). Alternately, the proposed method is able to identify the correct instance even in the case of several outliers, and is able to deal with very large graphs due to its modest computational complexity. Note that our model is provably optimal only in the case of zero noise, though it appears to perform quite accurately under reasonable noise levels.

[3] Rather, it is shown in [14] to *converge* to the optimal solution. If the algorithm is not run until convergence, the results may be suboptimal.

[4] This was done only to simplify the experiment – it is of course possible for our method to find *every* isomorphic instance if more than one exists.

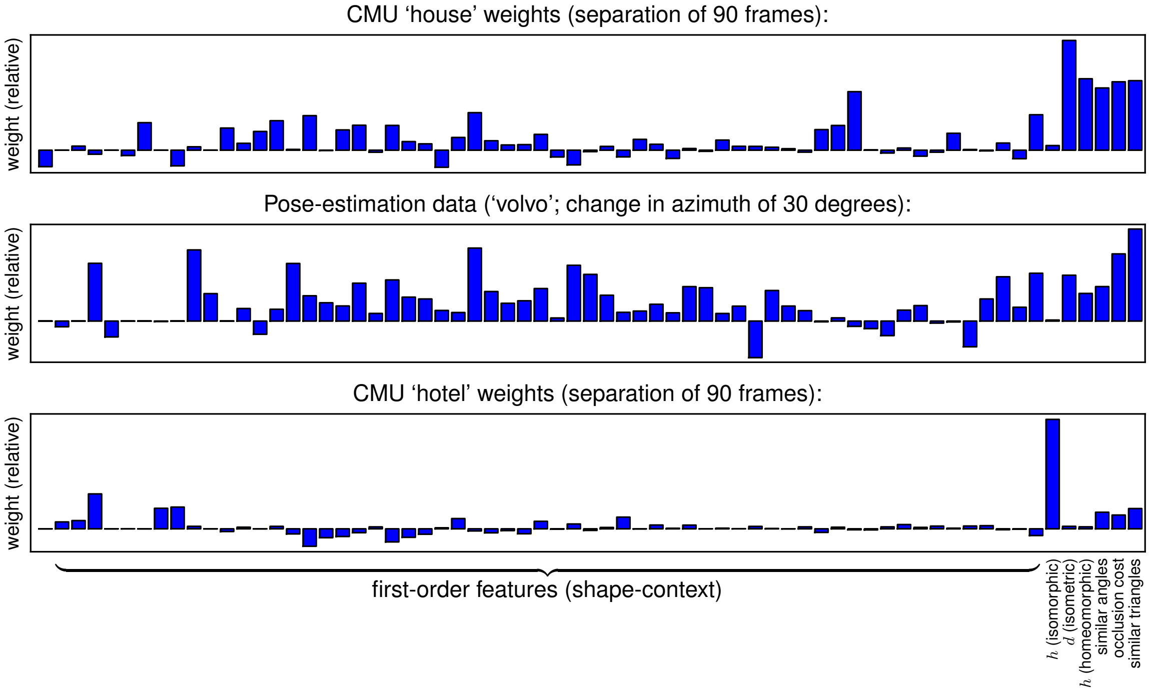


Figure 4. The weight vectors (Θ) learned from some of our experiments. Note that in the 'house' experiment the weight for the 'homeomorphic' feature is very high, whereas the weight for the 'isomorphic' feature is very low; the opposite effect is observed for the 'hotel' experiment. In effect, our algorithm has learned whether isomorphisms or homeomorphisms better capture these datasets.

4.2. Synthetic Data

In [20], the performance of linear and quadratic assignment (with learning) is reported on a series of silhouette images from [21]. The point sets in question are subject to increasing amounts of various distortions (shear, rotation, noise). In Figure 6(a) we report the performance from [20] against our method on the 'shear' dataset; the 'noise' dataset is not shown, as it is similar to our own synthetic dataset from the previous experiment. The 'rotation' dataset is also not shown, as our method will provably achieve zero error on the entire dataset (due to its rotational invariance). The topological structure of the graph was simply determined using the outline of the silhouette (see Figure 6(a)).

The 'balanced graph-matching' algorithm of [10] (which to the best of our knowledge is the state-of-the-art non-learning approach) is also reported for comparison, for those experiments where results were available from [20].

For each of our experiments involving learning, we split our data into training, validiation, and test sets, each of which contains one third of the data. Learning is performed on the *training* set for different values of the hyperparameter $\lambda \in \{0.001, 0.01, 0.1, 1, 10\}$. We always report the performance on the *test* set, for the value of λ that gave the best performance on the *validation* set.

Our learning approach is fully-supervised, i.e., we assume that the correspondences are fully labeled. We find that even a small number of such correspondences can be enough to obtain a substantial improvement over non-learning. Alternately, in cases where fully supervised data cannot be obtained, we note that unsupervised instances of this problem are likely to require the fully supervised version as a component [22].

4.3. 3-D Models from Different Viewpoints

In this experiment, our sequences represent different views of 3-D models. We consider several sequences: the CMU 'house' and 'hotel' sequences (used in [6,20]), as well as two new sequences from [24], which are advantageous in that their changes in rotation/azimuth are known.

Weight vectors from three of our experiments are shown in Figure 4. Unlike the previous experiments, the 'topology' of each point-pattern was *automatically generated* for each image: by forming a spanning tree (so that the graph is connected), and by connecting K-nearest-neighbors (examples are shown in Figure 6). Consequently, there is substantial noise in the topological features. Nevertheless, we observe high weights for the isomorphic and homeomorphic features in these datasets, indicating that these features are useful. Consequently, the unified model achieves better performance when the separation between frames is large. Note that in cases where the transformations are non-isometric, the model simply *learns* not to use the isometric features.

For the CMU 'house' sequence, we also present a timing plot (Figure 5(b), right). To improve the running time of our model, we used the same approach as [6], where for each node we only consider the P matches with the lowest linear cost, thus reducing the running time to $O((|V| + |E|)P^3)$. This parameter trades off accuracy against running time, though we found that even for small values of P, the impact on accuracy was minor; in this experiment we used $P = 15$, where $|V'| = 30$. The running time of balanced graph-matching [10] is also shown, though this is based on a Matlab implementation, while the others were written in C++. Our implementation is available online [25].

4.4. Video Sequence

In this experiment, we used the Harris-Hessian detector of [23] to identify keypoints in the 'claire' video sequence from [25], and extracted SIFT features for each keypoint [17]. Next, we selected 15 of these keypoints in each frame corresponding to important points-of-interest (such as the eyes/ears). Our goal is exactly the same as in the previous experiments, though we are now dealing with a very large number of outliers (typically, several hundred keypoints were identified), as well as occlusions in both the template and the target scenes (since the keypoint detector may not capture some points-of-interest). Performance on this dataset is shown in Figure 5(c); note that we do not use the Hamming loss in this experiment, but rather the endpoint loss (i.e., the average distance from the 'correct' keypoint) – this is a more sensible choice in the presence of many outliers. Quadratic assignment was not used, as it was prohibitively expensive on graphs of this size.

5. Conclusion

In this paper, we have presented a unified formulation for graph matching in Euclidean spaces, which accounts for isometries, isomorphisms, homeomorphisms, and node attributes. By means of structured estimation, the model is able to automatically learn the extent to which each of these properties are present in the training data instead of relying on *a priori* assumptions about the specific 'type' of graph matching problem under consideration. Inference in this model is efficient and exact since it comprises a junction tree of small tree-width. This method is provably optimal when there is no noise in the point coordinates and attributes, while we have shown that it remains highly accurate under reasonable noise levels.

Acknowledgements

The research leading to these results has received funding from the EC FP7 grant 216529 (PinView), as well as EPSRC/UK grant EP/F069626/1, ACASVA. NICTA is funded by the Australian Government's *Backing Australia's Ability* initiative, and the Australian Research Council's *ICT Centre of Excellence* program.

References

[1] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[2] J. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on PAMI*, 24(4):509–522, 2002.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[4] J. Ullman. An algorithm for subgraph isomorphism. *Journal of the ACM*, 23(1):31–42, 1976.

[5] A. S. LaPaugh and R. L. Rivest. The subgraph homeomorphism problem. In *ACM Symposium on Theory of Computing*, 1978.

[6] J. J. McAuley, T. S. Caetano, and A. J. Smola. Robust near-isometric matching via structured learning of graphical models. *NIPS*, 2008.

[7] H. Ling and D. Jacobs. Shape classification using the inner-distance. *IEEE Trans. on PAMI*, 29(2):286–299, 2007.

[8] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[9] L. Shapiro and J. Brady. Feature-based correspondence - an eigenvector approach. *Image and Vision Computing*, 10:283–288, 1992.

[10] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *NIPS*, 2006.

[11] R. C. Wilson and E. R. Hancock. Structural matching by discrete relaxation. *IEEE Trans. on PAMI*, 19(6):634–648, 1997.

[12] J. V. Kittler and E. R. Hancock. Combining evidence in probabilistic relaxation. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 3:29–51, 1989.

[13] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. PAMI*, 18(4):377–388, 1996.

[14] J. J. McAuley, T. S. Caetano, and M. S. Barbosa. Graph rigidity, cyclic belief propagation and point pattern matching. *IEEE Trans. on PAMI*, 30(11):2047–2054, 2008.

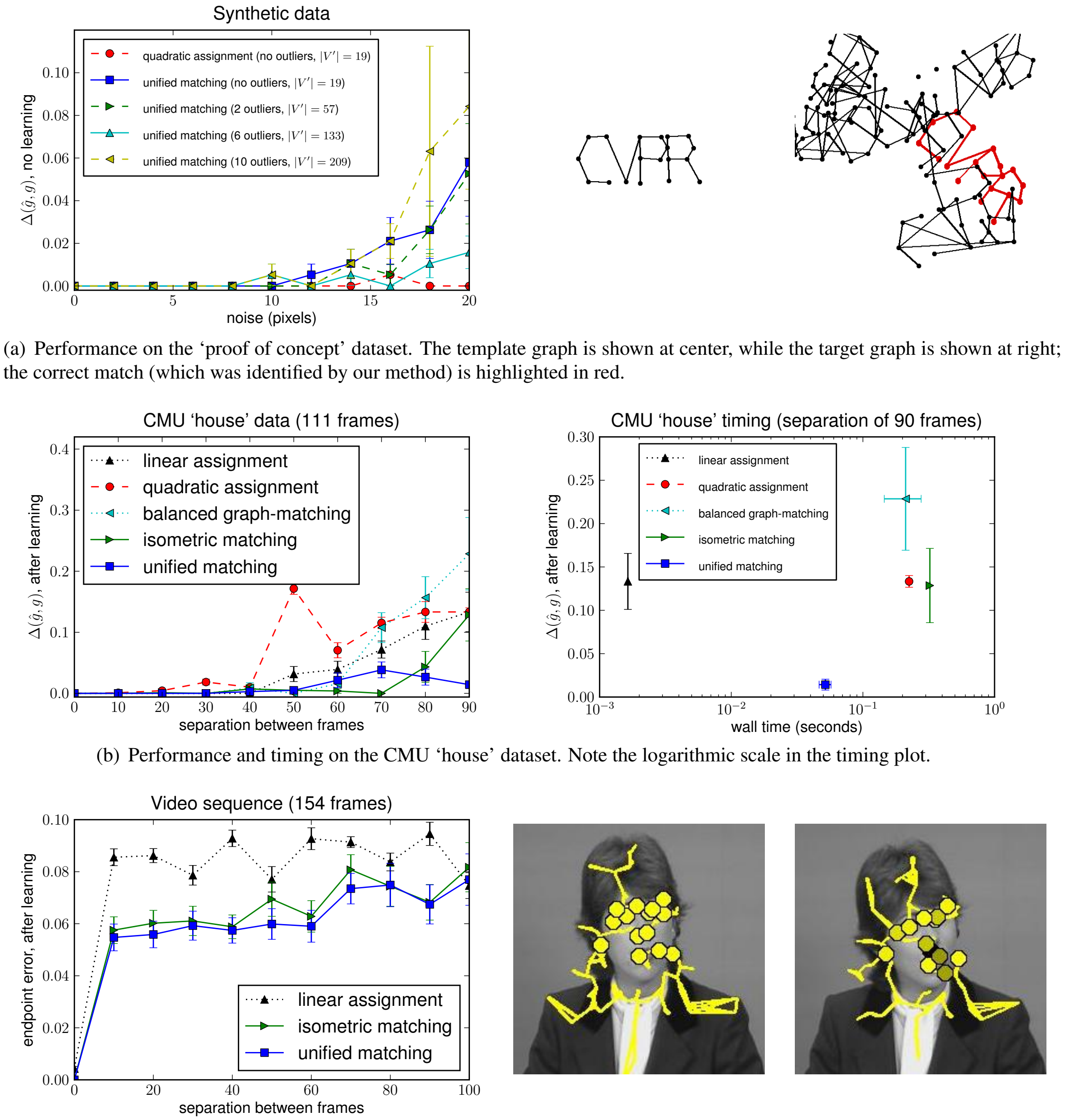
[15] T. S. Caetano and J. J. McAuley. Faster graphical models for point-pattern matching. *Spatial Vision*, 2009.

[16] T. S. Caetano and T. Caelli. A unified formulation of invariant point pattern matching. In *ICPR*, 2006.

[17] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[18] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[19] C. H. Teo, Q. Le, A. J. Smola, and S. V. N. Vishwanathan. A scalable modular convex solver for regularized risk minimization. In *KDD*, 2007.



(a) Performance on the 'proof of concept' dataset. The template graph is shown at center, while the target graph is shown at right; the correct match (which was identified by our method) is highlighted in red.

(b) Performance and timing on the CMU 'house' dataset. Note the logarithmic scale in the timing plot.

(c) Performance on SAMPL 'claire' dataset. The template graph is shown at center (yellow nodes indicate points-of-interest), while its mapping in the target is shown at right. Intensity corresponds to error: yellow indicates a correct match, while black indicates an error of 40 pixels.

Figure 5. Performance, timing, and example matches. Note that the *x*-axis of each plot essentially captures the 'difficulty' of the matching task: as the separation between frames increases, the transformations become less and less rigid. Images have been cropped for visibility.

[20] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *IEEE Trans. on PAMI*, 31(6):1048–1058, 2009.

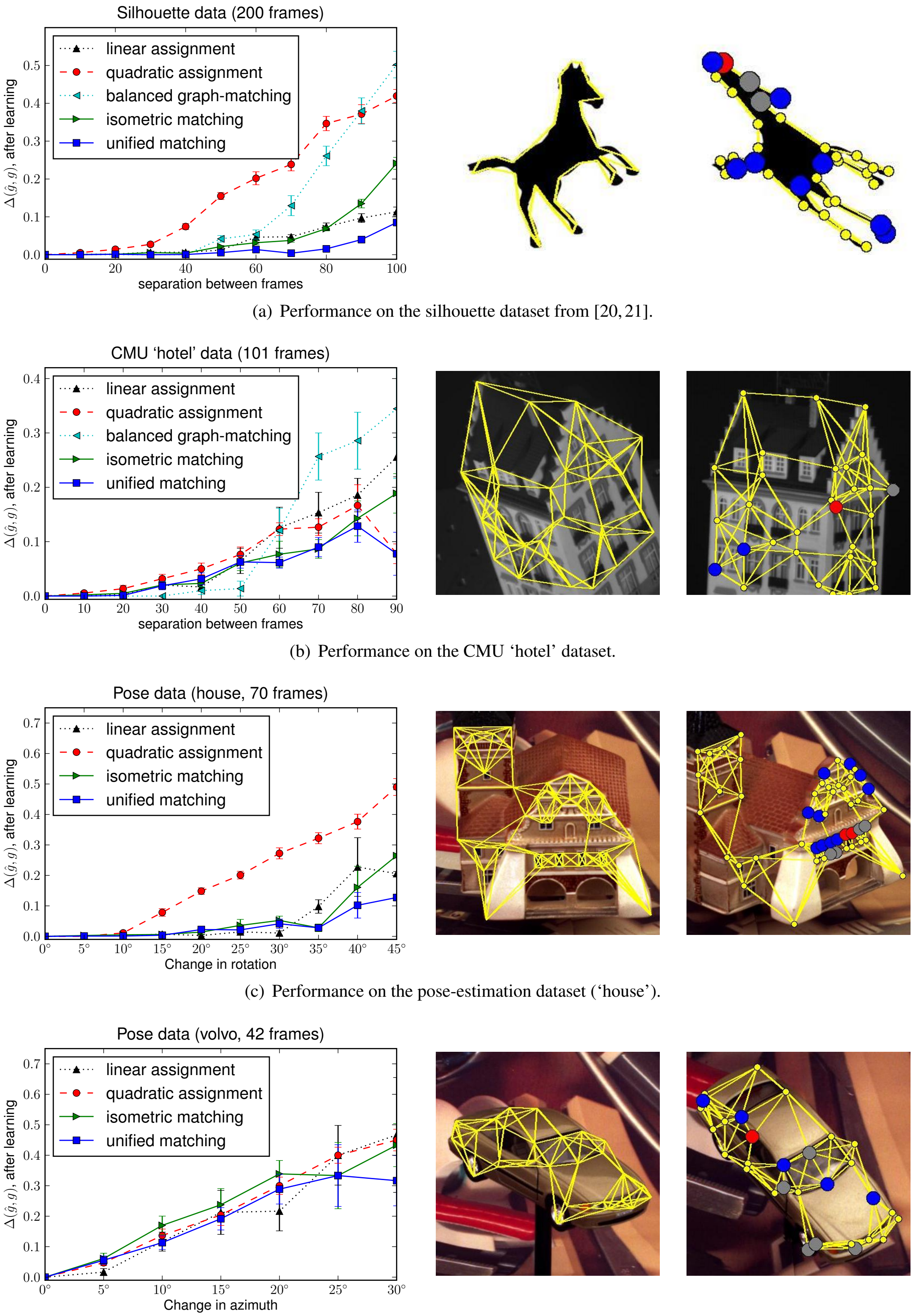
[21] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel. Analysis of two-dimensional non-rigid shapes. *International Journal of Computer Vision*, 2007.

[22] C. J. Yu and T. Joachims. Learning structural SVMs with latent variables. In *ICML*, 2009.

[23] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR*, 2006.

[24] F. Vikstén, P. Forssén, B. Johansson, and A. Moe. Comparison of local image descriptors for full 6 degree-of-freedom pose estimation. In *ICRA*, 2009.

[25] Implementation of our method:
`http://users.cecs.anu.edu.au/~julianm/`
Dataset sources:
`http://sampl.ece.ohio-state.edu/database.htm`
`http://tosca.cs.technion.ac.il`
`http://vasc.ri.cmu.edu/idb/html/motion/`
`http://www.cvl.isy.liu.se/research/objrec/posedb/`.



(a) Performance on the silhouette dataset from [20, 21].

(b) Performance on the CMU 'hotel' dataset.

(c) Performance on the pose-estimation dataset ('house').

(d) Performance on the pose-estimation dataset ('volvo').

Figure 6. Additional results (see Figure 5). Template graphs are shown at center with their topologies. Target graphs are shown at right, comparing accuracy with and without topological features. The color of each node indicates whether the correct match was identified by both methods (yellow), neither method (gray), only *without* topological features (red), or only *with* topological features (blue).