Faster Graphical Models for Point-Pattern Matching

Tibério S. Caetano and Julian J. McAuley*

Statistical Machine Learning Program, NICTA,

and the Research School of Information Sciences and Engineering, ANU

Abstract

It has been shown that isometric matching problems can be solved exactly in polynomial time, by means of a Junction-Tree with small maximal clique size. Recently, an iterative algorithm was presented which converges to the same solution an order of magnitude faster. Here, we build on both of these ideas to produce an algorithm with the same asymptotic running time as the iterative solution, but which requires only a *single* iteration of belief propagation. Thus our algorithm is much faster in practice, while maintaining similar error rates.

Keywords: Point-pattern matching; graphical models; isometric matching.

1 Introduction

A fundamental problem in Computer Vision and Pattern Recognition is to find a correspondence between two sets of image features. This has many applications, including reconstruction of the spatial geometry of a 3D scene (such as Hartley & Zisserman (2004)), feature-based object recognition (such as Felzenszwalb & Huttenlocher (2005)), and people detection in images (such as Caetano et al. (2007)).

We investigate a particular instance of the matching problem called *near-isometric point pattern matching*. In this setting, one assumes that the two feature sets to be matched differ only by isometric transformations, but allowing for point jitter and outliers. Therefore, the two sets may have different sizes. We introduce a method for obtaining the best match in the above setting which turns out to be much faster than the best competitor, while maintaining the same accuracy guarantees. Our method is directly inspired by the early work on probabilistic graphical models for matching using Junction Trees, first introduced in Caelli & Caetano (2005).

---

*tiberio.caetano@nicta.com.au, julian.mcauley@nicta.com.au

2 Background

The problem of point pattern matching consists of finding an instance of a 'template' graph (say, $\mathcal{S}$) in a 'target' scene (say, $\mathcal{U}$).[1] In this paper we will assume that the instance ($\mathcal{S}'$) of $\mathcal{S}$ in $\mathcal{U}$ is simply a near-isometric transformation of $\mathcal{S}$ (i.e. an isometric transformation, subject to some noise). In other words, we are attempting to find a mapping $\hat{y} : \mathcal{S} \to \mathcal{U}$ such that the distances between pairs of points in $\mathcal{S}$ are preserved in $\mathcal{S}'$. That is,

$$\hat{y} = \underset{y}{\operatorname{argmax}} \sum_{i,j;i \neq j} -|d(s_i, s_j) - d(y(s_i), y(s_j))|^2 \qquad (1)$$

($d(a, b)$ is simply the Euclidean distance between a and b). Unfortunately, this is an instance of the *quadratic assignment* problem, which is NP-hard in general (see Anstreicher (2003)).

It was shown however in Caetano et al. (2006) that under isometric transformations, it is not necessary to include dependencies between *all* pairs of nodes, but rather it is sufficient that the dependency structure constitutes a *globally rigid* graph (see Connelly (2005)). In such a graph, the lengths of the present edges uniquely determine the lengths of the absent edges (and thus the absent edges can be ignored – see Caetano et al. (2006) for details). Importantly, it is shown in Caetano & Caelli (2006) that there exist globally rigid graphs with small tree-width (see figure 2), meaning that the problem of (eq. 1) can be solved efficiently by means of exact probabilistic inference in a graphical model with small maximal clique size.

2.1 Preliminaries

Being more formal, we start by reviewing the general modeling methodology in the line of work of Caelli & Caetano (2005), Caetano et al. (2006), Caetano & Caelli (2006), McAuley et al. (2008), which is also explored in this paper. One represents the problem of finding a correspondence between two sets of point features as a problem of finding the most likely realisation of random variables in a Markov Random Field (i.e. an undirected probabilistic graphical model – see Bishop (2006), Pearl (1988)). Each point in $\mathcal{S}$ is represented by a random variable, which is graphically depicted by a circle in the probabilistic graphical model (see figures 1 and 2). Each random variable has a state space consisting of $|\mathcal{U}|$ possible realisations, i.e. the number of point features in the *scene* pattern. Indeed, in this model the fact that the random variable s_i has realisation u_j has precisely the semantics that the point s_i in $\mathcal{S}$ is matched to the point u_j in $\mathcal{U}$. Therefore, an entire joint realisation of the graphical model consisting of all the variables $\{s_1, \ldots, s_{|\mathcal{S}|}\}$ represents an entire match between the point patterns $\mathcal{S}$ and $\mathcal{U}$. In a Markov Random Field, *edges* between nodes represent the pattern of conditional independence statements assumed in the model. More precisely, if $n(i)$ is the set of nodes connected to node i by an edge, then the model assumes

---

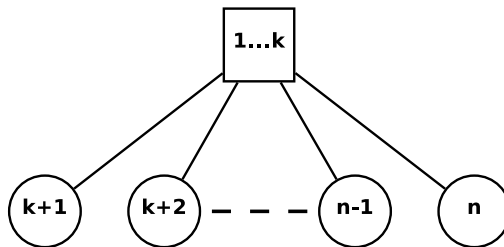[1]Strictly, we are talking about *embeddings* of graphs in the plane.

Figure 1: A k-tree graph. The square denotes an entire clique of k nodes. Note that the set of nodes $\{1 \ldots k, i\}$, for $k < i \le n$, forms a maximal clique in this diagram.

$p(y_i | y_{\text{rest}}) = p(y_i | y_{n(i)})$. Under this model, the probability of a match y is, by the Hammersley-Clifford theorem, decomposed over the maximal cliques of the underlying graph:

$$p(Y = y) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(y_c), \tag{2}$$

where $\mathcal{C}$ is the set of maximal cliques c in the graph and $\psi_c(y_c)$ are arbitrary non-negative real-valued functions, called *potential functions*, which intuitively encode the 'goodness' of the partial match y_c for the variables in the clique c. In other words, the Hammersley-Clifford theorem gives a clear algebraic form for the joint distribution that satisfies the conditional independence statements assumed in the model. Therefore, in order to find the most likely joint realisation in such a model (which corresponds to the best point matching), one needs to find y that maximises an expression which factorises over the variables in the cliques of the graph, as shown in (eq. 2).

2.2   Models for Matching

In Caetano & Caelli (2006), it is shown that by using a k-tree as a graphical model (figure 1), one can solve matching problems under translations, similarity, affine, and projective transformations (for $k = 1, 2, 3, 4$ respectively), due to the number of variables involved in each type of transformation. This is done by encoding in each maximal clique a potential function $\psi_c(y_c)$ that enforces agreement of transformations of points in that clique.[2] Of interest to us is the fact (shown in Caetano et al. (2006)) that a 2-tree can be used to solve matching problems under rotation and translation, whereas *isometric* matching problems (i.e. rotations, translations, and reflections) can be solved using a 3-tree (due to its global rigidity). Furthermore, it is shown that inference in a k-tree can be solved in $O(|\mathcal{S}||\mathcal{U}|^{k+1})$ time by means of the Junction-Tree algorithm.

---

[2]For example, for affine-invariant matches, $k = 3$ and the joint assignment in the 4-clique $\{1, 2, 3, i\}$ needs to be such that the joint realisations in each of the 4 triangles within the clique induce the same affine transformation.

In McAuley et al. (2008), the authors present another graph which is also globally rigid, but whose maximal clique size is smaller (three nodes instead of four). However, the clique-graph of their model no longer forms a tree, meaning that standard Junction-Tree algorithms cannot be applied. Instead, they show that Loopy Belief Propagation in such a graph will *converge* to the optimal solution, though not necessarily in a single iteration. Thus the asymptotic complexity is decreased from $O(|\mathcal{S}||\mathcal{U}|^4)$ to $O(|\mathcal{S}||\mathcal{U}|^3)$. Their experiments reveal that in practice, the performance increase is very close to $|\mathcal{U}|$, despite having to run the algorithm for several iterations.

3 Our Model

As mentioned, a 2-tree can be used to solve matching problems subject to translations and rotations. To achieve this, we enforce not only that the mapping $\hat{y}$ preserves the distances between nodes, but also that it preserves orientations between triangles. That is, for a 2-tree graph (say, $\mathcal{G}$), the problem of maximising (2) becomes

$$\hat{y} = \underset{y}{\operatorname{argmax}} \underbrace{\prod_{i,j\in\mathcal{G}} \exp(-|d(s_i,s_j) - d(y(s_i),y(s_j))|_2^2)}_{\text{edges}} \times$$

$$\underbrace{\prod_{i=3}^{|\mathcal{S}|} I\left(\operatorname{sign}(\det(\begin{bmatrix} s_1 - s_i \\ s_2 - s_i \end{bmatrix})) = \operatorname{sign}(\det(\begin{bmatrix} y(s_1) - y(s_i) \\ y(s_2) - y(s_i) \end{bmatrix}))\right)}_{\text{triangles}} \quad (3)$$

(the sign of the determinant determines the orientation of the triangle; $I(\cdot)$ is an indicator function; note that this is equivalent to a max-sum formulation in which assignments not obeying the above equality are given a potential of $-\infty$).

The idea behind our method stems from the realisation that such a model can also be used to handle reflections, simply by running it twice: during the first iteration, we insist that the orientations are preserved by the mapping (as in (eq. 3)), whereas during the second iteration, we insist that the orientations are *different*. We then simply choose the $\hat{y}$ which has the lower cost amongst both iterations.

To continue this idea, we can augment our 2-tree to include a boolean variable encoding both of these possibilities. The graphical model we use is shown in figure 2 (bottom). Note that we do not actually use a 2-tree, but our graph maintains the same properties (we choose this topology for easier comparison to the method of McAuley et al. (2008)). Notice that although the maximal clique size of our model is increased from three to four, the asymptotic running time is not changed by this addition – it is still $O(|\mathcal{U}||\mathcal{S}|^3)$, and only a single iteration is required. This is because the number of states of the new boolean variable is (obviously) two (independent of $|\mathcal{U}|$ or $|\mathcal{S}|$)

To be precise, we add the variable R to our model, with domain $\{0, 1\}$. Our maximisation problem now becomes

$$\hat{y}, \hat{r} = \operatorname*{argmax}_{y,r} \prod_{i,j \in \mathcal{G}}^{|\mathcal{S}|} \exp(-|d(s_i, s_j) - d(y(s_i), y(s_j))|_2^2) \times$$

$$\prod_{i=1}^{|\mathcal{S}|-2} \left[(1-r)I\left(\mathrm{sd}(s_i, s_{i+1}, s_{i+2}) = \mathrm{sd}(y(s_i), y(s_{i+1}), y(s_{i+2})) \right) \right.$$

$$\left. + rI\left(\mathrm{sd}(s_i, s_{i+1}, s_{i+2}) \neq \mathrm{sd}(y(s_i), y(s_{i+1}), y(s_{i+2})) \right) \right] \quad (4)$$

(where $\mathrm{sd}(a, b, c)$ is simply the sign of the determinant $\mathrm{sign}(\det(\begin{bmatrix} a - c \\ b - c \end{bmatrix}))$; $\hat{r}$ now recovers whether the template scene is reflected ($\hat{r} = 1$) or not ($\hat{r} = 0$) in the target).

Although this model will correctly solve isometric matching problems, it will be somewhat sensitive to noise, as a small error in a single point can change the orientation of a triangle (if it contains an angle very close to $0°$ or $180°$). Hence, rather than imposing an infinite penalty upon triangles with non-matching orientations, we suggest a 'soft-error' model which imposes a penalty of $P|\varphi|^2$, where φ is the smallest angle by which we could rotate a point to give the two triangles the same orientation, and the same angle (see figure 3 for examples).[3] The constant P is determined experimentally (we found $P = 100$ to work well in practice).

4   Experiments

We implemented the proposed model, as well as those from Caetano et al. (2006) and McAuley et al. (2008) (with topologies shown in figure 2) in C++ using the Blitz++ array library.[4] OpenMP was used to parallelise the initialisation of our model, which simply consists of computing the potential for each possible assignment, and is done independently in each clique. Experiments were run on a Quad-Core, 2.66Ghz machine. When using the method of McAuley et al. (2008) we ran 20 iterations of belief propagation (in McAuley et al. (2008), the authors use a convergence test after each iteration, and stop the algorithm at convergence; we have chosen not to do so, as running this test appears to take almost as long as propagation itself).

4.1   House Data

For our first experiment, we consider the CMU 'house' sequence,[5] a sequence of 111 frames of a toy house, with the same 30 points labelled in each frame. This

---

[3]In the max-product setting, we multiply the potential by $\exp(-P|\varphi|^2)$.
[4]Blitz++ array library: `http://www.oonumerics.org/blitz`
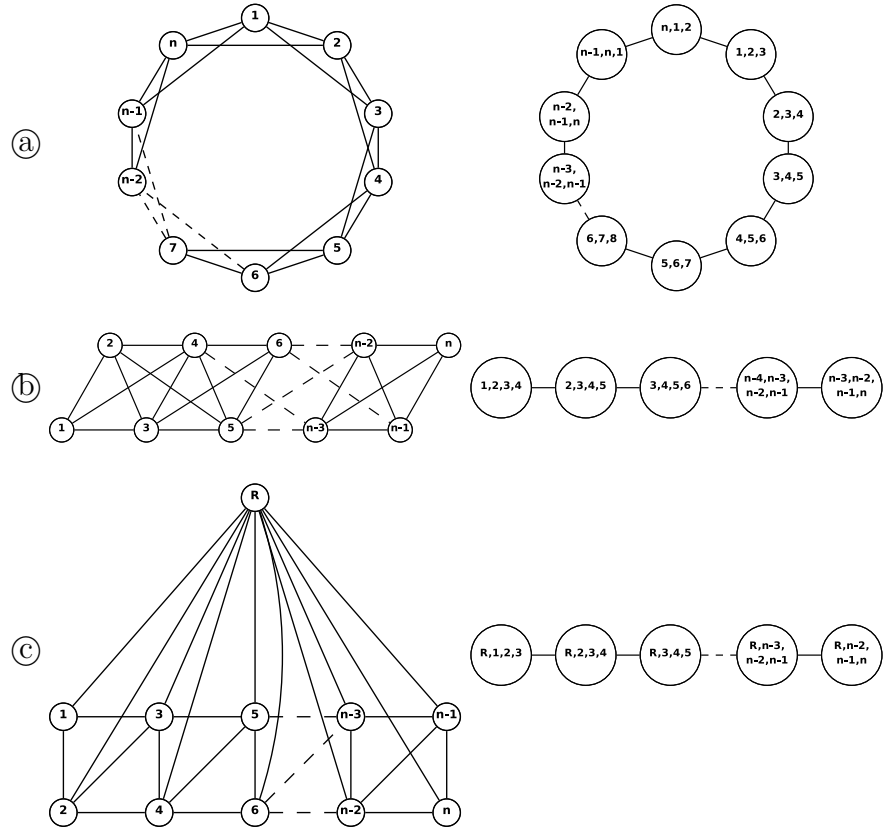[5]CMU 'house' sequence: `http://vasc.ri.cmu.edu/idb/html/motion/house/index.html`

Figure 2: Graphical models (left), and their clique-graphs (right). (a) – The model of McAuley et al. (2008). (b) – The model of Caetano & Caelli (2006). (c) – The proposed model.
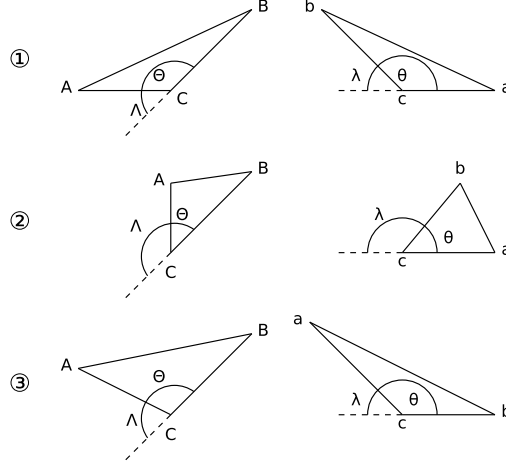
Figure 3: Three examples mapping the triangle (ABC) to (abc). ① – the orientations are different; the penalty is the squared sum of the smaller angles, $(\Lambda + \lambda)^2$. ② – the orientations are different; the penalty is $(\Theta + \theta)^2$. ③ – the orientations are the same; there is no penalty.

experiment appeared in both Caetano et al. (2006) and McAuley et al. (2008), and is presented here for comparison.

In figure 4 (top), we compare the performance of our model to those of Caetano et al. (2006) and McAuley et al. (2008) as the baseline (separation between frames) increases (the problem becomes increasingly difficult for higher baselines, as the house is gradually subject to more significant non-isometric transformations).

It is interesting to note that when imposing a hard-error (i.e. when strictly enforcing that orientations are consistent), our model appears to perform *worse* than those of Caetano et al. (2006) and McAuley et al. (2008) for low baselines, but much better for high baselines. As a possible explanation, we may observe poor performance with a low baseline due to our model's sensitivity to noise; the better performance for high baselines may simply be explained by the fact that our potential function implicitly enforces that mappings are injective within cliques, whereas the potentials of Caetano et al. (2006) and McAuley et al. (2008) do not. Alternately, when using a soft error, we observe almost identical performance to the other methods.

The bottom of figure 4 compares the running time of our method against those of Caetano et al. (2006) and McAuley et al. (2008). In McAuley et al. (2008), the authors reported an improvement of almost precisely $|\mathcal{U}|$ over the method of Caetano et al. (2006), despite the fact that their algorithm requires several iterations to converge. The explanation for this appears to be that the running time of their implementation is dominated by the initialisation stage
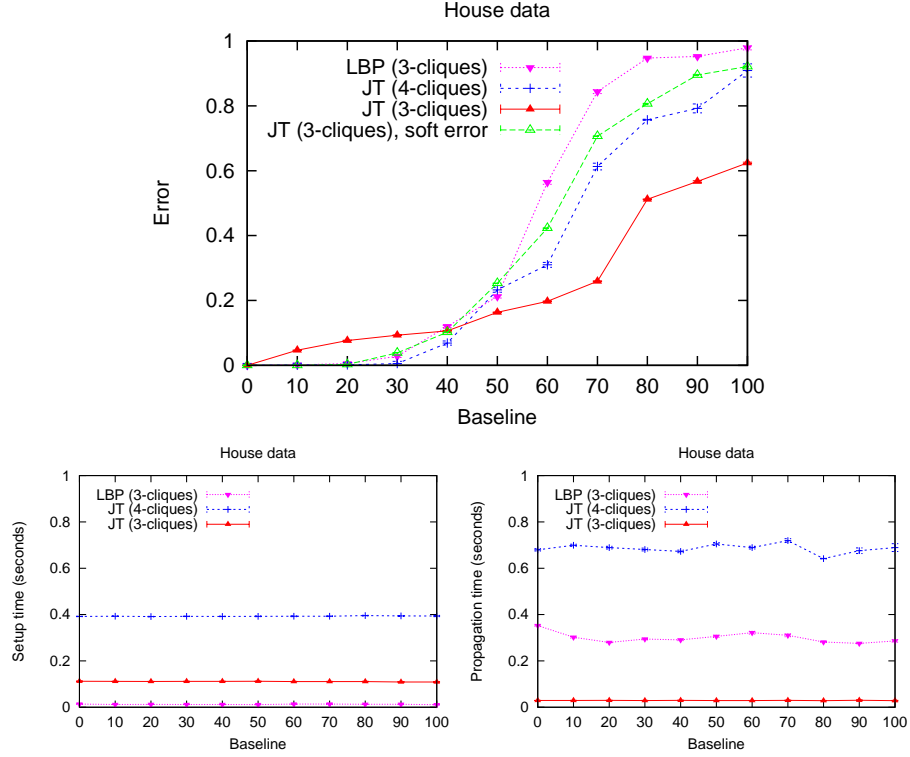
Figure 4: Performance on the 'house' dataset. Top: Error as the baseline (separation between frames) increases. Our model (JT, 3-cliques) is compared to those of Caetano & Caelli (2006) (JT, 4-cliques) and McAuley et al. (2008) (LBP). Our model with hard error appears to exhibit the best performance for higher baselines, despite performing slightly worse for lower baselines (each of the other models appears to perform equally). Bottom: Time taken to initialise the model (left), and to perform belief propagation (right). For each baseline, the average performance on all examples is shown, with error bars indicating standard error.
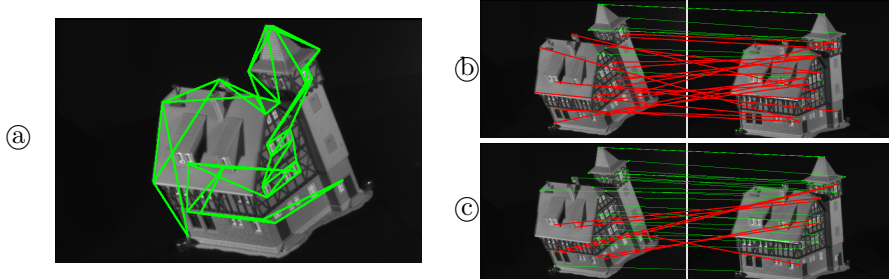
Figure 5: ⓐ – The topology of our graphical model. ⓑ – An example matching, using the model of McAuley et al. (2008) (9/30 points correct, mismatches are shown in red). ⓒ – An example matching, using the proposed method (23/30 points correct, using hard-error).

(i.e. computing the clique potentials), whereas message passing is done using efficient array libraries. We have attempted to address this issue by parallelising the initialisation stage, meaning that we are able to do initialisation much faster than propagation. As expected, our initialisation is slower than that of McAuley et al. (2008), due to the fact that we must compute a potential not only for each triangle, but also for each orientation. However, this is more than made up for during propagation, wherein our algorithm achieves a marked improvement.

Figure 5 shows the topology of our model, as well as an example of matching using our model and that of McAuley et al. (2008).

4.2 Synthetic Data

For our second experiment, we use a silhouette image from the Mythological Creatures 2D database (from Bronstein, Bronstein & Kimmel (2007), Bronstein, Bronstein, Bruckstein & Kimmel (2007)).[6] We randomly selected 30 points on the silhouette, and randomly perturbed their x and y-coordinates uniformly between $-\epsilon/2$ and $\epsilon/2$ pixels. For each value of ϵ (from 0 to 20), 20 such perturbations were created, resulting in $20 \times 19 = 380$ pairs for matching. The results of matching in this setting are shown in figure 6. The performance of all four models is similar for this dataset (and the running times are similar to those of the previous experiment).

Figure 7 shows an example of matching using this dataset. Since the order of points in our graphical model is chosen at random for this experiment, the topology of our model appears to have much less 'structure' than in figure 5.
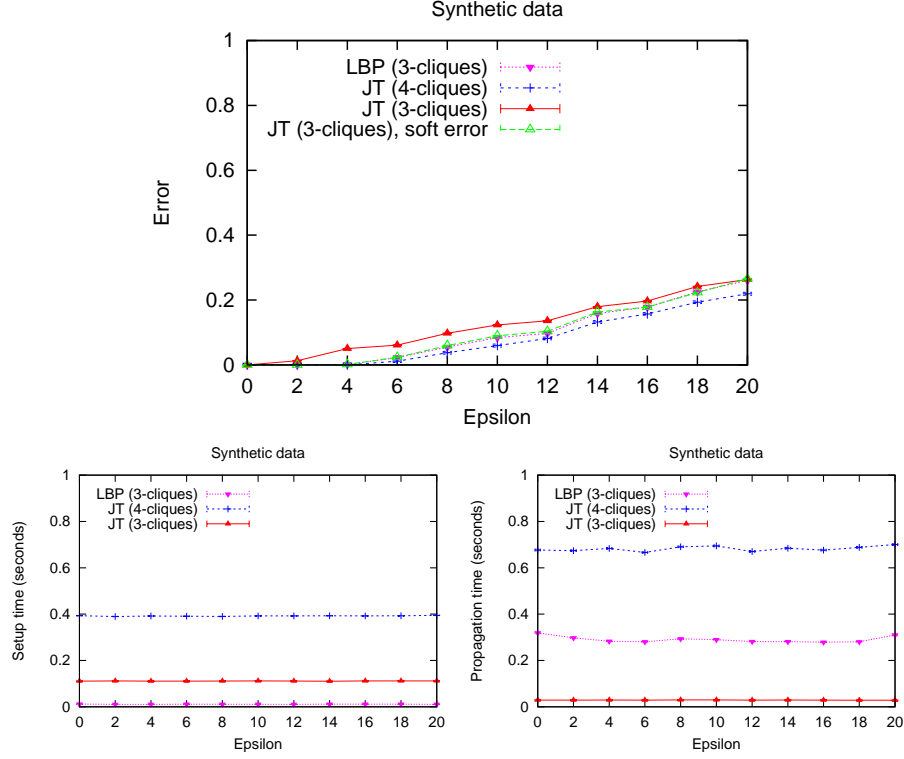
---

[6]Mythological Creatures 2D database: `http://tosca.cs.technion.ac.il`

Figure 6: Performance on the synthetic dataset. Top: Error as ϵ (epsilon) increases. Our model (JT, 3-cliques) is compared to those of Caetano & Caelli (2006) (JT, 4-cliques) and McAuley et al. (2008) (LBP). In this case, using a hard-error appears to result in slightly worse performance, whereas the other models appear to perform equally. Bottom: Time taken to initialise the model (left), and to perform belief propagation (right). For each value of ϵ, the average performance on all examples is shown, with error bars indicating standard error.
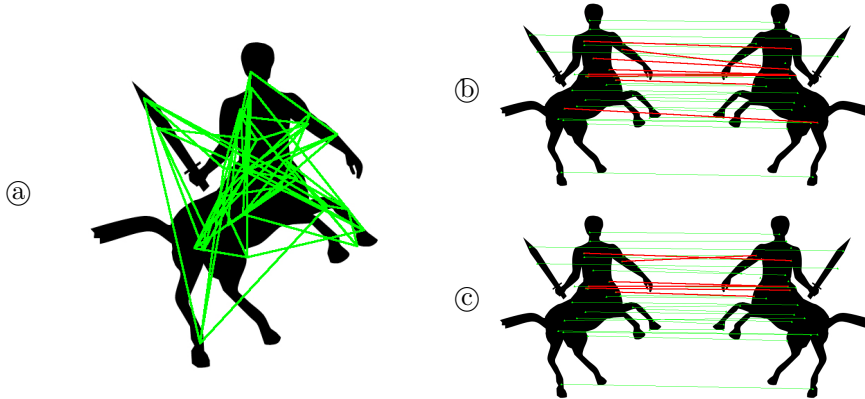
Figure 7: (a) – The topology of our graphical model. In contrast with the topology shown in figure 5, the nodes are ordered randomly. (b) – An example matching, using the model of McAuley et al. (2008) (22/30 points correct, mismatches are shown in red). (c) – An example matching, using the proposed method (24/30 points correct, using hard-error).

5   Conclusion

We have presented a model for performing near-isometric point pattern matching which is faster than its nearest competitors, while maintaining the same accuracy guarantees. In comparison to the approach of Caetano & Caelli (2006), whose running time is $O(|\mathcal{S}||\mathcal{U}|^4)$ (where $|\mathcal{S}|$ and $|\mathcal{U}|$ are the sizes of the template and target scenes, respectively), and the approach of McAuley et al. (2008), whose running time is $O(|\mathcal{S}||\mathcal{U}|^3)$ *per iteration*, the running time of our method is $O(|\mathcal{S}||\mathcal{U}|^3)$, but requires only a single iteration to converge. This result is achieved by noting that a 2-tree can be used to perform matching up to rotation and translation, if we either enforce or prohibit that the template scene is reflected in the target. We then augment this model to include a boolean variable, allowing for both of these possibilities.

References

Anstreicher, K. (2003). Recent advances in the solution of quadratic assignment problems, *Mathematical Programming* **97**, 27–42.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, Springer.

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

Bronstein, A. M., Bronstein, M. M. and Kimmel, R. (2007). *Numerical Geometry of Non-rigid Shapes*, Springer.

Caelli, T. and Caetano, T. S. (2005). Graphical models for graph matching: Approximate models and optimal algorithms, *Pattern Recognition Letters* **26**(3), 339–346.

Caetano, T. S. and Caelli, T. (2006). A unified formulation of invariant point pattern matching, in *International Conference on Pattern Recognition*, pp. 121–124.

Caetano, T. S., Caelli, T., Schuurmans, D. and Barone, D. A. C. (2006). Graphical models and point pattern matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(10), 1646–1663.

Caetano, T. S., Cheng, L., Le, Q. V. and Smola, A. J. (2007). Learning graph matching, in *International Conference on Computer Vision*, pp. 1–8.

Connelly, R. (2005). Generic global rigidity, *Discrete & Computational Geometry* **33**(4), 549–563.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition, *International Journal of Computer Vision* **61**, 2005.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*, Cambridge University Press.

McAuley, J. J., Caetano, T. S. and Barbosa, M. S. (2008). Graph rigidity, cyclic belief propagation and point pattern matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(11), 2047–2054.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*, Morgan-Kaufman.