Graph Rigidity, Cyclic Belief Propagation, and Point Pattern Matching

Julian J. McAuley, Tibério S. Caetano, and
Marconi S. Barbosa

Abstract—A recent paper [1] proposed a provably optimal polynomial time method for performing near-isometric point pattern matching by means of exact probabilistic inference in a chordal graphical model. Its fundamental result is that the chordal graph in question is shown to be *globally rigid*, implying that exact inference provides the *same* matching solution as exact inference in a complete graphical model. This implies that the algorithm is optimal when there is no noise in the point patterns. In this paper, we present a new graph that is also globally rigid but has an advantage over the graph proposed in [1]: Its maximal clique size is smaller, rendering inference significantly more efficient. However, this graph is not chordal, and thus, standard Junction Tree algorithms cannot be directly applied. Nevertheless, we show that loopy belief propagation in such a graph converges to the optimal solution. This allows us to retain the optimality guarantee in the noiseless case, while substantially reducing both memory requirements and processing time. Our experimental results show that the accuracy of the proposed solution is indistinguishable from that in [1] when there is noise in the point patterns.

Index Terms—Point pattern matching, graph matching, graphical models, belief propagation, global rigidity, chordal graphs.

◆

1 INTRODUCTION

POINT pattern matching is a fundamental problem in pattern recognition and has been modeled in several different forms depending on the demands of the application domain in which it is required [2], [3], [4], [5]. A classic formulation that is realistic in many practical scenarios is that of near-isometric point pattern matching, in which we are given both a "template" ($\mathcal{T}$) and a "scene" ($\mathcal{S}$) point pattern, and it is assumed that $\mathcal{S}$ contains an instance of $\mathcal{T}$ (say, $\mathcal{T}'$) apart from an isometric transformation and possibly some small jitter in the point coordinates. The goal is to identify $\mathcal{T}'$ in $\mathcal{S}$ and find which points in $\mathcal{T}$ correspond to which points in $\mathcal{T}'$.

Recently, a method was introduced that solves this problem efficiently by means of exact belief propagation in a certain graphical model [1].[1] The approach is appealing because it is optimal not only in that it consists of exact inference in a graph with small maximal clique size (= 4 for matching in $\mathbb{R}^2$) but also in that the *graph itself* is optimal. There it is shown that the maximum a posteriori (MAP) solution in the sparse and tractable graphical model where inference is performed is actually the *same* MAP solution that would be obtained if a fully connected model (which

1. See also [6] for further treatment of shape matching in the Bayesian setting; see [7] and [8] for further treatment using tree-structured models.

• *The authors are with the Statistical Machine Learning group, NICTA, Locked Bag 8001 Canberra ACT 2601 Australia, and the Research School of Information Sciences and Engineering, Australian National University, ACT 0200.*
E-mail: {julian.mcauley, Tiberio.Caetano}@nicta.com.au, marconi@ifsc.usp.br.

is intractable) could be used. This is due to the so-called *global rigidity* of the chordal graph in question: When the graph is embedded in the plane, the lengths of its edges uniquely determine the lengths of the absent edges (i.e., the edges of the graph complement) [9]. The computational complexity of the optimal point pattern matching algorithm is then shown to be $O(nm^4)$ (both in terms of processing time and memory requirements), where n is the number of points in the template point pattern and m is the number of points in the scene point pattern (usually with $m > n$ in applications). This reflects precisely the computational complexity of the Junction Tree algorithm in a chordal graph with $O(n)$ nodes, $O(m)$ states per node, and maximal cliques of size 4. The authors present experiments which give evidence that the method substantially improves on well-known matching techniques, including Graduated Assignment [10].

In this paper, we show how the same optimality proof can be obtained with an algorithm that runs in $O(nm^3)$ time per iteration. In addition, memory requirements are precisely decreased by a factor of m. We are able to achieve this by identifying a new graph, which is globally rigid but has a *smaller* maximal clique size: three. The main problem we face is that our graph is *not* chordal, so in order to enforce the running intersection property for applying the Junction Tree algorithm, the graph should first be triangulated; this would not be interesting in our case, since the resulting triangulated graph would have larger maximal clique size. Instead, we show that belief propagation in this graph *converges to the optimal solution*, although not necessarily in a single iteration. In practice, we find that convergence occurs after a small number of iterations, thus improving the runtime by an order of magnitude. We compare the performance of our model to that in [1] with synthetic and real point sets derived from images and show that in fact comparable accuracy is obtained while substantial speedups are observed.

2 BACKGROUND

We consider point matching problems in $\mathbb{R}^2$. The problem we study is that of near-isometric point pattern matching (as defined above), i.e., one assumes that a near-isometric instance ($\mathcal{T}'$) of the template ($\mathcal{T}$) is somewhere "hidden" in the scene ($\mathcal{S}$). By "near-isometric," it is meant that the relative distances of points in $\mathcal{T}$ are approximately preserved in $\mathcal{T}'$. For simplicity of exposition, we assume that $\mathcal{T}$, $\mathcal{T}'$, and $\mathcal{S}$ are ordered sets (their elements are indexed). Our aim is to find a map $x : \mathcal{T} \mapsto \mathcal{S}$ with image $\mathcal{T}'$ that best preserves the relative distances of the points in $\mathcal{T}$ and $\mathcal{T}'$, i.e.,

$$x^* = \underset{x}{\operatorname{argmin}} \|D(\mathcal{T}) - D(x(\mathcal{T}))\|_2^2, \quad (1)$$

where $D(\mathcal{T})$ is the matrix whose (i,j)th entry is the euclidean distance between points indexed by i and j in $\mathcal{T}$ and $\| \cdot \|_2$ is the Frobenius norm. Note that finding x^* is inherently a combinatorial optimization problem since $\mathcal{T}'$ is itself a subset of $\mathcal{S}$, the scene point pattern. In [1], a generic point in $\mathcal{T}$ is modeled as a random variable (X_i) and a generic point in $\mathcal{S}$ is modeled as a possible realization of the random variable (x_i). As a result, a joint realization of all the random variables corresponds to a match between the template and the scene point patterns. A graphical model (see [11] and [12]) is then defined on this set of random variables whose edges are set according to the topology of a so-called 3-tree graph (any 3-tree that spans $\mathcal{T}$). A 3-tree is a graph obtained by starting with the complete graph on three vertices, K_3, and then adding new vertices that are connected only to those
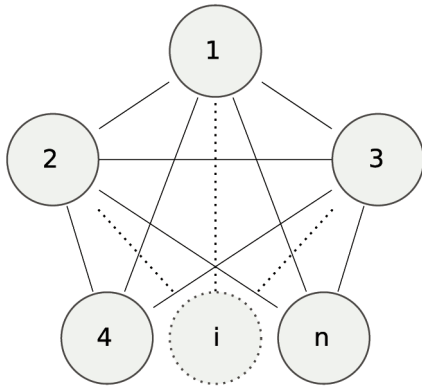
Fig. 1. An example of a 3-tree. Each node corresponds to a point in the template set.

Fig. 2. The general form of the graph we consider, with n nodes.

same three vertices.[2] Fig. 1 shows an example of a 3-tree; it should be noted that the points in the template set correspond to the *nodes* in this graph, whereas the points in the scene correspond to their *assignments*. The reasons claimed in [1] for introducing 3-trees as a graph topology for the probabilistic graphical model are that 1) 3-trees are globally rigid in the plane and 2) 3-trees are chordal[3] graphs. This implies 1) that the 3-tree model is "optimal" (in a way that will be made clear in Section 3 in the context of the new graph we propose) and 2) that 3-trees have a Junction Tree with fixed maximal clique size $(= 4)$; as a result, it is possible to perform exact inference in polynomial time [1].

Potential functions are defined on pairs of neighboring nodes and are large if the difference between the distance of neighboring nodes in the template and the distance between the nodes they map to in the scene is small (and small if this difference is large). This favors isometric matchings. More precisely,

$$\psi_{ij}(X_i = x_i, X_j = x_j) = f\big(d(X_i, X_j) - d(x_i, x_j)\big), \qquad (2)$$

where $f(\cdot)$ is typically some unimodal function peaked at zero (e.g., a zero-mean Gaussian function) and $d(\cdot, \cdot)$ is the euclidean distance between the corresponding points (for simplicity of notation, we do not disambiguate between random variables and template points or realizations and scene points). For the case of *exact* matching, i.e., when there exists an x^* such that the minimal value in (1) is zero, then $f(\cdot) = \delta(\cdot)$ (where $\delta(\cdot)$ is just the indicator function $1_{\{0\}}(\cdot)$). The potential function of a maximal clique (Ψ) is then simply defined as the product of the potential functions over its six $(= C_2^4)$ edges (which will be maximal when every factor is maximal). It should be noted that the potential function of each edge is included in no more than *one* of the cliques containing that edge.

For the case of exact matching (i.e., no jitter), it is shown in [1] that running the Junction Tree algorithm on the 3-tree graphical model with $f(\cdot) = \delta(\cdot)$ will actually find a MAP assignment which coincides with x^*, i.e., such that $\|D(\mathcal{T}) - D(x^*(\mathcal{T}))\|_2^2 = 0$. This is due to the "graph rigidity" result, which tells us that equality of the lengths of the edges in the 3-tree and the edges induced by the matching in $\mathcal{T}'$ is sufficient to ensure the equality of the lengths of all pairs of points in $\mathcal{T}$ and $\mathcal{T}'$. This will be made technically

precise in Section 3, when we prove an analogous result for another graph.[4]

3 AN IMPROVED GRAPH

Here, we introduce another globally rigid graph that has the advantage of having a *smaller maximal clique size*. Although the graph is *not* chordal, we will show that exact inference *is* tractable and that we will indeed benefit from the decrease in the maximal clique size. As a result, we will be able to obtain optimality guarantees like those from [1].

Our graph is constructed using Algorithm 1.

Algorithm 1. Graph generation for $\mathcal{G}$.
  1 Create a cycle graph by traversing all the nodes in $\mathcal{T}$ (in any order)
  2 Connect all nodes whose distance in this cycle graph is two (i.e., connect each node to its neighbor's neighbor)

This algorithm will produce a graph like the one shown in Fig. 2. We will denote by $\mathcal{G}$ the set of graphs that can be generated by Algorithm 1. $G = (V, E)$ will denote a generic graph in $\mathcal{G}$.

In order to present our results, we need to start with the definition of a globally rigid graph.

Definition 1. *A planar graph embedding G is said to be globally rigid in $\mathbb{R}^2$ if the lengths of the edges uniquely determine the lengths of the edges of the graph complement of G.*

Therefore, our statements are really about *graph embeddings* in $\mathbb{R}^2$, but for simplicity of presentation, we will simply refer to these embeddings as "graphs."

This means that there are no degrees of freedom for the absent edges in the graph: They must all have specified and fixed lengths. To proceed, we need a simple definition and some simple technical lemmas.

Definition 2. *A set of points is said to be in a general position in $\mathbb{R}^2$ if no three points lie in a straight line.*

Lemma 3. *Given a set of points in general position in $\mathbb{R}^2$, if the distances from a point P to two other fixed points are determined, then P can be in precisely two different positions.*

Proof. Consider two circles, each centered at one of the two reference points with radii equal to the given distances to point P. These circles intersect at precisely two points (since the three points are not collinear). This proves the statement.   □

---

2. Technically, connecting new vertices to the three nodes of the original K_3 graph is not required: It suffices to connect new vertices to any existent 3-clique.

3. A chordal graph is one in which every cycle of length greater than three has a chord. A chord of a cycle is an edge not belonging to the cycle but which connects two nodes in the cycle (i.e., a "shortcut" in a cycle).
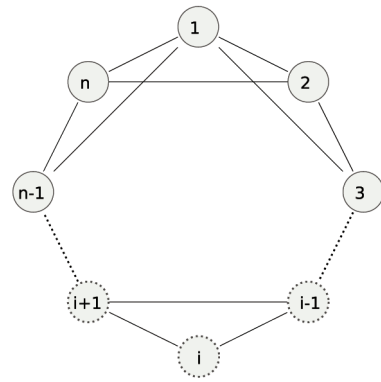
4. It is worth noting that a 2-tree would be sufficient if we only wished to allow for translations and rotations, and a 1-tree would be sufficient if we only wished to allow for translations [13]. Since we wish to handle all isometries, these models are not further considered.

The following lemma follows directly from Lemma 1 in [1] and is stated without proof:

Lemma 4. *Given a set of points in general position in $\mathbb{R}^2$, if the distances from a point P to three other fixed points are determined, then the position of P is uniquely determined.*

We can now present a proposition.

Proposition 5. *Any graph $G \in \mathcal{G}$ arising from Algorithm 1 is globally rigid in the plane if the nodes are in general position in the plane.*

Proof. Define a reference frame S, where points 1, 2, and n have specific coordinates (we say that the points are "determined"). We will show that all points then have determined positions in S and therefore have determined relative distances, which by definition implies that the graph is globally rigid.

We proceed by contradiction: Assume there exists at least one undetermined point in the graph. Then, we must have an undetermined point i such that $i - 1$ and $i - 2$ are determined (since points 1 and 2 are determined). By virtue of Lemma 4, points $i + 1$ and $i + 2$ must then be also undetermined (otherwise, point i would have determined distances from three determined points and as a result would be determined).

Let us now assume that *only* points i, $i + 1$, and $i + 2$ are undetermined. Then, the only possible realizations for points i, $i + 1$, and $i + 2$ are their reflections with respect to the straight line that passes through points $i - 1$ and $i + 3$, since these are the only possible realizations that maintain the rigidity of the triangles $(i - 1, i, i + 1)$, $(i, i + 1, i + 2)$, and $(i + 1, i + 2, i + 3)$, since $i - 1$ and $i + 3$ are assumed fixed. However, since $i + 4$ and $i - 2$ are also fixed by assumption, this would break the rigidity of triangles $(i + 2, i + 3, i + 4)$ and $(i, i - 1, i - 2)$. Therefore, $i + 3$ cannot be determined. This can then be considered as the base case in an induction argument that goes as follows: Assume *only* $i, \ldots, i + p$ are undetermined. Then, by reflecting these points over the line that joins $i - 1$ and $i + p + 1$ (which are fixed by assumption), we obtain the only other possible realization consistent with the rigidity of the triangles who have all their vertices in $i - 1, \ldots, i + p + 1$. However, this realization is inconsistent with the rigidity of triangles $(I + p, i + p + 1, i + p + 2)$ and $(i, i - 1, i - 2)$; therefore, $i + p + 1$ must not be determined and, by induction, any point j such that $j > i + 2$ must not be determined, which contradicts the assumption that n is determined. As a result, the assumption that there is at least one undetermined point in the graph is false. This implies that the graph has all points determined in S, and therefore, all relative distances are determined, and by definition, the graph is globally rigid. This proves the statement. □

Although we have shown that graphs $G \in \mathcal{G}$ are globally rigid, notice that they are *not* chordal. For the graph in Fig. 2, the cycles $(1, 3, 5, \ldots, n - 1, 1)$ and $(2, 4, 6, \ldots, n, 2)$ have no chord. Moreover, triangulating this graph in order to make it chordal will necessarily increase (to at least 4) the maximal clique size (which is not sufficient for our purposes since we arrive at the case of [1]).

Instead, consider the clique graph formed by $G \in \mathcal{G}$. If there are n nodes, the clique graph will have cliques $(1, 2, 3), (2, 3, 4), \ldots, (n - 2, n - 1, n), (n - 1, n, 1), (n, 1, 2)$. This clique graph forms a cycle, which is depicted in Fig. 3.[5]

---

5. Note that if we connected *every* clique whose nodes intersected, the clique graph would have the same topology as the graph in Fig. 2; here, we have only formed enough connections so that the intersection of any two cliques is shared by the cliques on at least one path between them (similar to the running intersection property for Junction Trees).
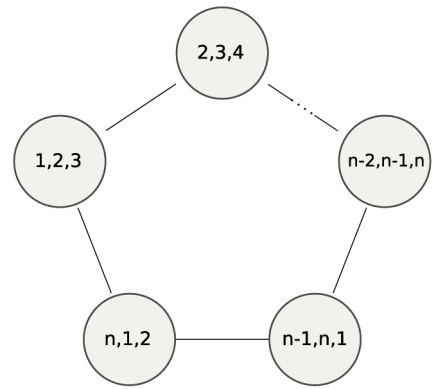

Fig. 3. The clique graph obtained from the graph in Fig. 2.

We now draw on results first obtained by Weiss [14] and confirmed elsewhere [15], [16]. There it is shown that, for graphical models with a single cycle, belief propagation converges to the optimal MAP assignment, although the computed marginals may be incorrect. Note that, for our purposes, this is precisely what is needed: We are after the most likely joint realization of the set of random variables, which corresponds to the best match between the template and the scene point patterns. Max-product belief propagation [17] in a cycle graph like the one shown in Fig. 3 amounts to computing the following messages iteratively:

$$m_{i \mapsto i+1}(U_i \cap U_{i+1}) = \max_{U_i \backslash U_{i+1}} \Psi(U_i) m_{i-1 \mapsto i}(U_i \cap U_{i-1}), \qquad (3)$$

where U_i is the set of singleton variables in clique node i, $\Psi(U_i)$ is the potential function for clique node i, and $m_{i \mapsto i+1}$ is the message that passed from clique node i to clique node $i + 1$. Upon reaching the convergence monitoring threshold, the optimal assignment for singleton variable j in clique node i is then computed by $\mathrm{argmax}_{U_i \backslash j} \Psi(U_i) m_{i-1 \mapsto i}(U_i \cap U_{i-1}) m_{i+1 \mapsto i}(U_i \cap U_{i+1})$.

Unfortunately, the above result is only shown in [14] when the *graph itself* forms a cycle, whereas we only have that the *clique graph* forms a cycle. However, it is possible to show that the result still holds in our case by considering a new graphical model in which the *cliques themselves* form the nodes, whose cliques are now just the edges in the clique graph. The result in [14] can now be used to prove that belief propagation in *this* graph converges to the optimal MAP assignment, which (by appropriately choosing potential functions for the new graph) implies that belief propagation should converge to the optimal solution in the *original* clique graph also.

To demonstrate this, we need not only to show that belief propagation in the new model converges to the optimal assignment, but also that belief propagation in the new model is *equivalent* to belief propagation in the original model.

Proposition 6. *The original clique graph (Fig. 3) can be transformed into a model containing only pairwise potentials, whose optimal MAP assignment is the same as the original model's.*

Proof. Consider a "node" $C_1 = (X_1, X_2, X_3)$ (in the original clique graph) whose neighbors share exactly two of its nodes (for instance, $C_2 = (X_2, X_3, X_4)$). Where the domain for each node in the original clique graph was simply $\{1, 2, \ldots, |\mathcal{S}|\}$, the domain for each "node" in our new graph simply becomes $\{1, 2, \ldots, |\mathcal{S}|\}^3$.

In this setting, it is no longer possible to ensure that the assignment chosen for each "node" is consistent with the assignment to its neighbor, that is, for an assignment (x_1, x_2, x_3)

to C_1 and (x'_2, x'_3, x_4) to C_2, we cannot guarantee that $x_2 = x'_2$ or $x_3 = x'_3$. Instead, we will simply define the potential functions on this new graph in such a way that the optimal MAP assignment implicitly ensures this equality. Specifically, we shall define the potential functions as follows: For two cliques $C_I = (I_1, I_2, I_3)$ and $C_J = (J_1, J_2, J_3)$ in the original graph (which share two nodes, say, (I_2, I_3) and (J_1, J_2)), define the *pairwise* potential for the clique ($\Psi'_{I,J}$) in the new graph as

$$\Psi'_{I,J}(i_{(123)}, j_{(123)}) = \begin{cases} \Psi_I(i_1, i_2, i_3), & \text{if } (i_2, i_3) = (j_1, j_2), \\ \rho, & \text{otherwise,} \end{cases} \quad (4)$$

where Ψ_I is simply the clique potential for the Ith clique in the *original* graph; $i_{(123)} \in \text{domain}(I_1) \times \text{domain}(I_2) \times \text{domain}(I_3)$ (sim for $j_{(123)}$). That is, we are setting the pairwise potential to simply be the original potential of *one* of the cliques if the assignments are compatible and ρ otherwise. If we were able to set $\rho = 0$, we would guarantee that the optimal MAP assignment was exactly the optimal MAP assignment in the original graph—however, this is not possible, since the result of [15] only holds when the potential functions have finite dynamic range. Hence, we must simply choose ρ sufficiently small so that the optimal MAP assignment cannot possibly contain an incompatible match—it is clear that this is possible, for example, $\rho = \left(\prod_C \max_{\mathbf{x}_C} \Psi_C(\mathbf{x}_C)\right)^{-1}$ will do (if the potentials are scaled to be at least one).

The result of [14] now implies that belief propagation in *this* graph will converge to the optimal MAP assignment, which we have shown is equal to the optimal MAP assignment in the *original* graph.[6]   □

Proposition 7. *The messages passed in the new model are equivalent to the messages passed in the original model, except for repetition along one axis.*

Proof. We use induction on the number of iterations. First, we must show that the outgoing messages are the same during the first iteration (during which the incoming messages are not included). We will denote by $m^i_{(X_1,X_2,X_3) \mapsto (X_2,X_3,X_4)}$ the message from (X_1, X_2, X_3) to (X_2, X_3, X_4) during the ith iteration:

$$m^1_{(X_1,X_2,X_3) \mapsto (X_2,X_3,X_4)}(x_2, x_3) = \max_{X_1} \Psi_{(X_1,X_2,X_3)}(x_1, x_2, x_3), \quad (5)$$

$$\begin{aligned} m^1_{(X_{(123)},X_{(234)}) \mapsto (X_{(234)},X_{(345)})}&(x_{(234)}) \\ &= \max_{X_{(123)}} \Psi'_{X_{(123)},X_{(234)}}(x_{(123)}, x_{(234)}) \\ &= 1 \times \max_{X_1} \Psi_{(X_1,X_2,X_3)}(x_1, x_2, x_3) \\ &= m^1_{(X_1,X_2,X_3) \mapsto (X_2,X_3,X_4)}(x_2, x_3). \end{aligned} \quad (6)$$

This result only holds due to the fact that ρ will never be chosen when maximizing along any axis. We now have that the messages are equal during the first iteration (the only difference being that the message for the new model is repeated along one axis).[7] Next, suppose that during the $(n-1)$st iteration, the messages (for both models) are equal to $\kappa(x_1, x_2)$. Then, for the nth iteration, we have

$$\begin{aligned} m^n_{(X_1,X_2,X_3) \mapsto (X_2,X_3,X_4)}&(x_2, x_3) \\ &= \max_{X_1} \{\Psi_{(X_1,X_2,X_3)}(x_1, x_2, x_3)\kappa(x_1, x_2)\}, \end{aligned} \quad (7)$$

$$\begin{aligned} m^n_{(X_{(123)},X_{(234)}) \mapsto (X_{(234)},X_{(345)})}&(x_{(234)}) \\ &= \max_{X_{(123)}} \{\Psi'_{X_{(123)},X_{(234)}}(x_{(123)}, x_{(234)})\kappa(x_1, x_2)\} \\ &= 1 \times \max_{X_1} \{\Psi_{(X_1,X_2,X_3)}(x_1, x_2, x_3)\kappa(x_1, x_2)\} \\ &= m^n_{(X_1,X_2,X_3) \mapsto (X_2,X_3,X_4)}(x_2, x_3). \end{aligned} \quad (8)$$

Hence, the two message passing schemes are equivalent by induction.   □

We can now state our main result.

Theorem 8. *Let $G \in \mathcal{G}$ be a graph generated according to the procedure described in Algorithm 1. Assume that there is a perfect isometric instance of $\mathcal{T}$ within the scene point pattern $\mathcal{S}$.[8] Then, the MAP assignment x^* obtained by running belief propagation over the clique graph derived from G is such that $\|D(\mathcal{T}) - D(x^*(\mathcal{T}))\|_2^2 = 0$.*

Proof. For the exact matching case, we simply set $f(\cdot) = \delta(\cdot)$ in (2). Now, for a graph $G \in \mathcal{G}$ given by Algorithm 1, the clique graph will simply be a cycle, as shown in Fig. 3, and following Propositions 6 and 7, as well as the already mentioned result in [14], belief propagation will find the correct MAP assignment x^*, i.e.,

$$\begin{aligned} x^* &= \arg\max_x P_G(X = x) \\ &= \arg\max_x \prod_{i,j:(i,j)\in E} \delta\big(d(X_i, X_j) - d(x_i, x_j)\big), \end{aligned} \quad (9)$$

where P_G is the probability distribution for the graphical model induced by the graph G. Now, we need to show that x^* also maximizes the criterion which ensures isometry, i.e., we need to show that the above implies that

$$\begin{aligned} x^* &= \arg\max_x P_{\text{complete}}(X = x) \\ &= \arg\max_x \prod_{i,j} \delta\big(d(X_i, X_j) - d(x_i, x_j)\big), \end{aligned} \quad (10)$$

where P_{complete} is the probability distribution of the graphical model induced by the complete graph. Note that x^* must be such that the lengths of the edges in E are precisely equal to the lengths of the edges in $E_{\mathcal{T}'}$ (i.e., the edges induced in $\mathcal{S}$ from E by the map $X = x^*$). By the global rigidity of G, the lengths of $\bar{E}$ must then be also precisely equal to the lengths of $\bar{E}_{\mathcal{T}'}$. This implies that $\prod_{i,j:(i,j)\in\bar{E}} \delta(d(X_i, X_j) - d(x_i^*, x_j^*)) = 1$. Since (10) can be expanded as

$$x^* = \arg\max_x \left\{ \prod_{i,j:(i,j)\in E} \delta\big(d(X_i, X_j) - d(x_i, x_j)\big) \right. $$
$$\left. \prod_{i,j:(i,j)\in\bar{E}} \delta\big(d(X_i, X_j) - d(x_i, x_j)\big) \right\}, \quad (11)$$

it becomes clear that x^* will also maximize (10). This proves the statement.   □

---

6. To reiterate, the fact that the dynamic range is finite ensures that belief propagation converges to a unique fixed point (for graphs with a single loop) [15]; from this, the results in [14] and [16] guarantee optimality of the MAP assignment.

7. To be completely precise, the message for the new model is actually a function of only a *single* variable—$X_{(234)}$. By "repeated along one axis," we mean that for any given $(x_2, x_3, x_4) \in \text{domain}(X_2) \times \text{domain}(X_3) \times \text{domain}(X_4)$, the message at this point is independent of x_4, which therefore has no effect when maximizing.

8. In addition, we require that there is only a *single* isometric instance of $\mathcal{T}$ in $\mathcal{S}$ in order to ensure that the optimal MAP assignment is unique.

4 EXPERIMENTS

We have set up a series of experiments comparing the proposed model to that of [1]. Here, we compare graphs of the type shown in Fig. 2 to graphs of the type shown in Fig. 1.

The parameters used in our experiments are given as follows:

ϵ. This parameter controls the noise level used in our model. Here, we apply Gaussian noise to each of the points in $\mathcal{T}$ (with standard deviation ϵ in each axis). We have run our experiments on a range of noise levels between 0 and $4/256$ (where the original points in $\mathcal{T}$ are chosen randomly between 0 and 1). Note that this is the same as the setting used in [1].

Potential functions $\psi_{ij}(X_i = x_i, X_j = x_j) = f(d(X_i, X_j) - d(x_i, x_j))$. As in [1], we use a Gaussian function, i.e., $\exp\left(\frac{(d(X_i,X_j)-d(x_i,x_j))^2}{2\sigma^2}\right)$. The parameter σ is fixed beforehand as $\sigma = 0.4$ for the synthetic data and $\sigma = 150$ for the real-world data (as is done in [1]). While this potential function does *not* enforce that the chosen mapping is injective, it ensures that noninjective mappings are discouraged by having a low potential.

Dynamic range. As mentioned in Section 2, the potential function $\Psi(\mathbf{x})$ is simply the product of $\psi_{ij}(X_i = x_i, X_j = x_j)$ for all edges (i, j) in $\mathbf{x}$ (here, each maximal clique $\mathbf{x}$ contains three edges). The *dynamic range* of a function is simply defined as its maximum value divided by its minimum value (i.e., $\frac{\max_{\mathbf{x}} \Psi(\mathbf{x})}{\min_{\mathbf{x}} \Psi(\mathbf{x})}$). In order to prove convergence of our model, it is necessary that the dynamic range of our potential function is finite [15]. Therefore, rather than using $\Psi(\mathbf{x})$ directly, we use $\Psi'(\mathbf{x}) = (1/d) + (1 - 1/d)\Psi(\mathbf{x})$. This ensures that the dynamic range of our model is no larger than d and that $\Psi' \to \Psi$ as $d \to \infty$. In practice, we found that varying this parameter did not have a significant effect on convergence time. Hence, we simply fixed a large finite value ($d = 1,000$) throughout.

MSE-cutoff. In order to determine the point at which belief propagation has converged, we compute the marginal distribution of every clique and compare it to the marginal distribution after the previous iteration. Belief propagation is terminated when this mean-squared error is less than a certain cutoff value for every clique in the graph. When choosing the mode of the marginal distributions after convergence, if two values differ by less than the square root of this cutoff, both of them are considered as possible MAP estimates (although this was rarely an issue when the cutoff was sufficiently small). We found that as $|\mathcal{S}|$ increased, the mean squared error between iterations tended to be smaller and therefore that smaller cutoff values should be used in these instances. Indeed, although the number of viable matches increases as $|\mathcal{S}|$ increases, the distributions increase in sparsity at an even faster rate—hence, the distributions tend to be *less peaked* on the average and changes are likely to have less effect on the mean squared error. Hence, we decreased the cutoff values by a factor of 10 when $|\mathcal{S}| \geq 30$.[9]

The clique graph in which messages are passed by our belief propagation algorithms is exactly that shown in Fig. 3. It is worth noting, however, that we also tried running belief propagation using a clique graph in which messages were passed between *all* intersecting cliques; we found that this made no difference to the performance of the algorithm[10] and we have therefore restricted our experiments to the clique graph in Fig. 3 with respect to its optimality guarantees.

---

9. Note that this is not a parameter in [1], in which only a single iteration is ever required.

10. Apart from one slight difference, including the additional edges appears to provide convergence in fewer iterations. However, since the number of messages being passed is doubled, the overall runtime for both clique graphs was ultimately similar.
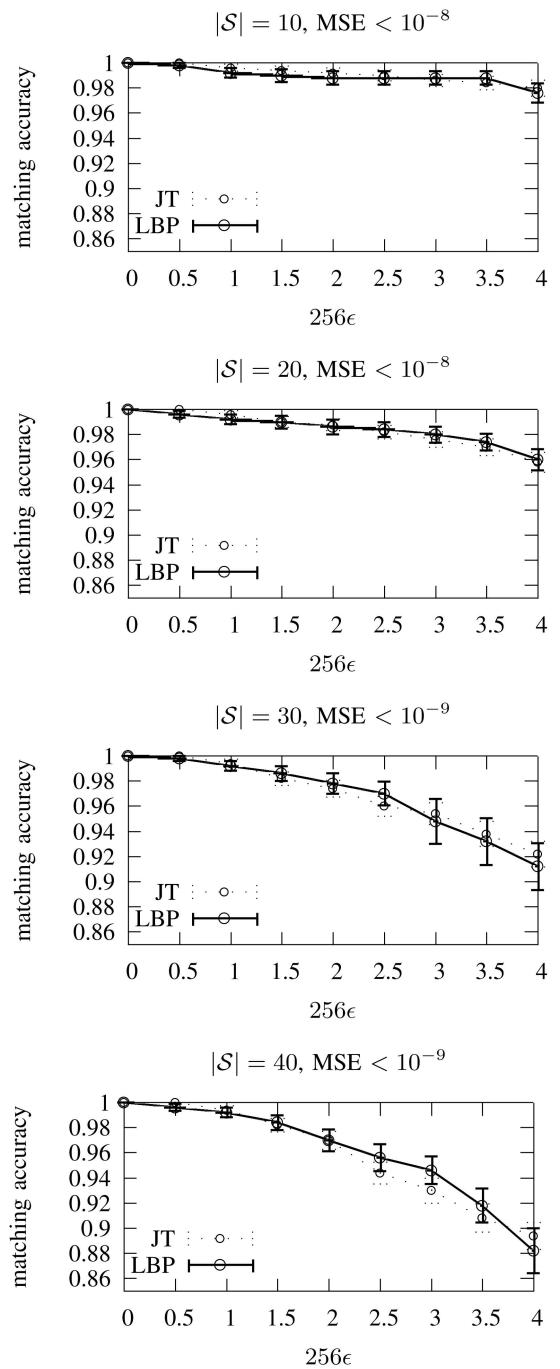

Fig. 4. Matching accuracy (proportion of correct matches) of our model against that in [1]. The performance of our model is statistically indistinguishable from that in [1] for all noise levels. The error bars indicate the average and standard error of 50 experiments.

For the sake of runtime comparison, we implemented the proposed model as well as that of [1] using the *Elefant*[11] belief propagation libraries in Python. However, to ensure that the results presented are consistent with those of [1], we simply used code that the authors provided when reporting the matching accuracy of their model. Our implementation computes messages (3) in a random order for all sites during each iteration.

Fig. 4 compares the matching accuracy (proportion of correct matches) of our model with that in [1] for $|\mathcal{S}| = 10, 20, 30,$ and 40
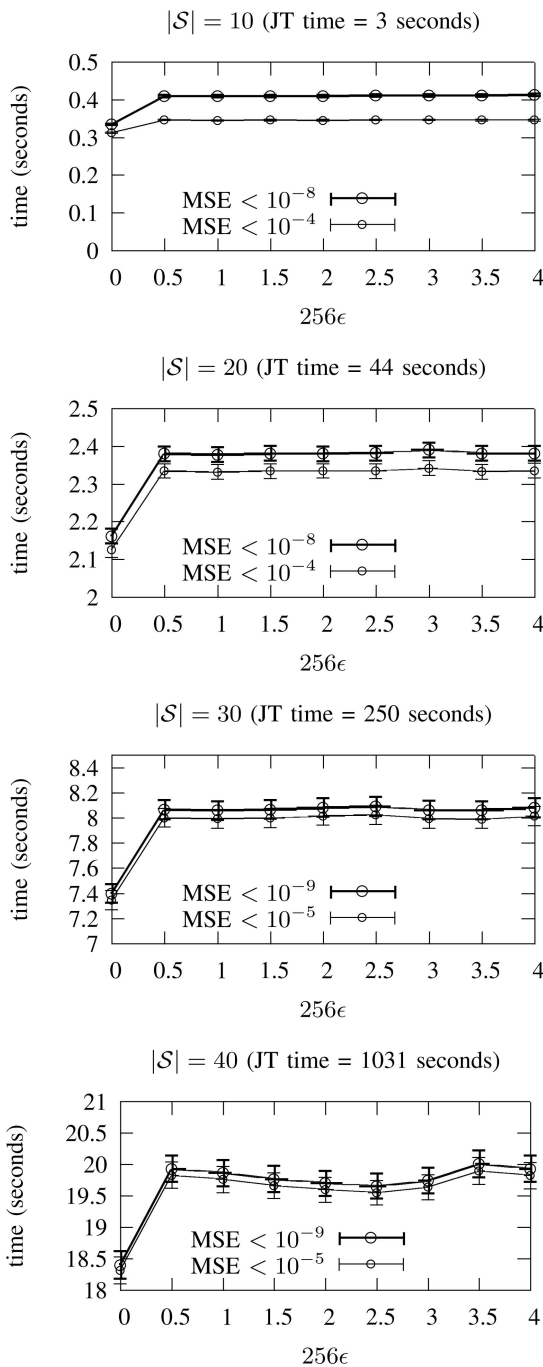
---

11. http://elefant.developer.nicta.com.au/.

Fig. 5. Running-time of our model as the jitter varies, for different MSE-cutoffs. Speed-ups are almost exactly one order of magnitude.
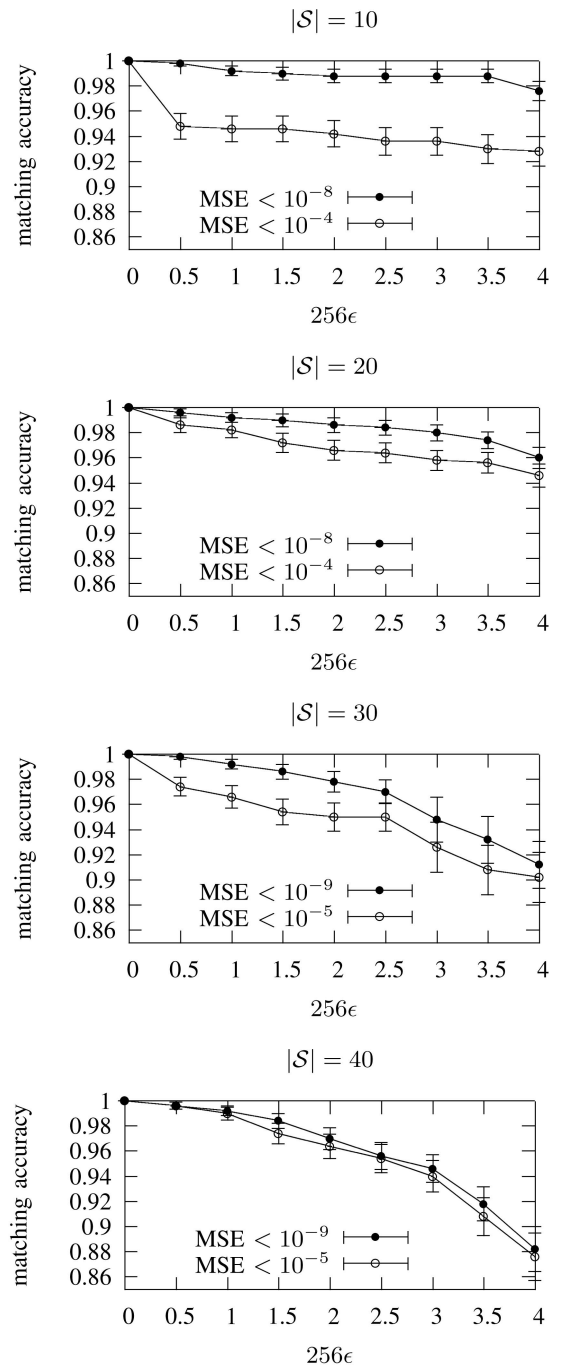

Fig. 6. Matching accuracy of our model as the MSE-cutoff varies. This figure suggests that the higher cutoff value should be sufficient when matching larger point sets.

(here, we fix $|\mathcal{T}| = 10$). The performance of our algorithm is indistinguishable from that of the Junction Tree algorithm.

Figs. 5 and 6 show the runtime and matching accuracy (respectively) of our model, as we vary the mean-squared error cutoff. Obviously, it is necessary to use a sufficiently low cutoff in order to ensure that our model has converged, but choosing too small a value may adversely effect its runtime. We found that the mean-squared error varied largely during the first few iterations, and we therefore enforced a minimum number of iterations (here, we chose at least 5) in order to ensure that belief propagation was not terminated prematurely. Fig. 5 reveals that the runtime is not significantly altered when increasing the MSE-cutoff—revealing that the model has almost always reached the lower cutoff value after five iterations (in which case, we should expect a speedup of

precisely $|\mathcal{S}|/5$). Furthermore, decreasing the MSE-cutoff does not significantly improve the matching accuracy for larger point sets (Fig. 6), so choosing the lower cutoff does little harm if runtime is a major concern. Alternately, the Junction Tree model (which only requires a single iteration), took (for $S = 10$ to 40) 3, 44, 250, and 1,031 seconds, respectively. These models differ only in the topology of the network (see Section 3) and the size of the messages being passed; our method easily achieves an order of magnitude improvement for large networks.[12]

12. In fact, the speedup appears to be *more* than an order of magnitude for the large graphs, which is likely a side effect of the large memory requirements of the Junction Tree algorithm.
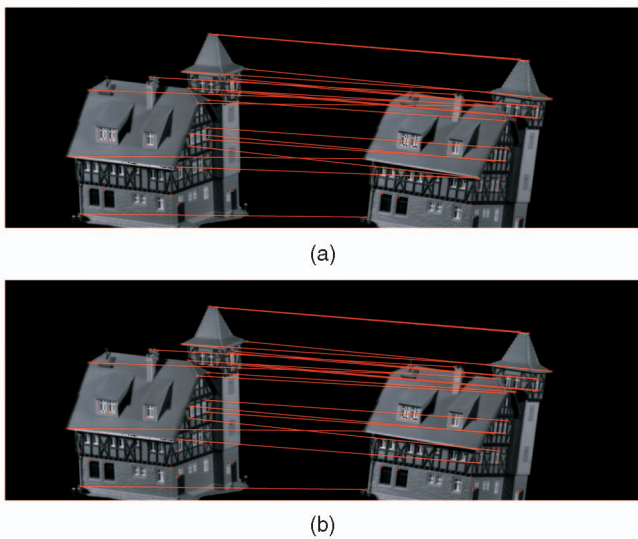
Fig. 7. (a) Points matched using the Junction Tree algorithm (the points in the left frame were matched to the corresponding points in the right frame); 16 points are correctly matched by this algorithm. (b) Points matched using our algorithm; 17 points are correctly matched.

Finally, we present matching results using data from the CMU house sequence.[13] In this data set, 30 points corresponding to certain features of the house are available over 111 frames. Fig. 7 shows the 71st and the last (111th) frames from this data set. Overlayed on these images are the 30 significant points, together with the matches generated by the Junction Tree algorithm and our own (matching the first 20 points); in this instance, the Junction Tree algorithm correctly matched 16 points and ours matched 17. Fig. 8 shows how accurately points between frames are matched as the baseline (separation between frames) varies. We also vary the number of points in the template set ($|\mathcal{T}|$) from 15 to 30. Our model seems to outperform the Junction Tree model for small baselines, whereas, for large baselines and larger point sets, the Junction Tree model seems to be the best. It is however difficult to draw conclusions from both models in these cases, since they are designed for the near-isometric case, which is violated for larger baselines.

5   CONCLUSIONS

We have shown that the near-isometric point pattern matching problem can be solved much more efficiently than what is currently reported as the state of the art, while maintaining the same optimality guarantees for the noiseless case and comparable accuracy for the noisy case. This was achieved by identifying a new type of graph with the same global rigidity property of previous graphs, in which exact inference is far more efficient. Although exact inference is not *directly* possible by means of the Junction Tree algorithm since the graph is not chordal, what we managed to show is that loopy belief propagation in such a graph does converge to the optimal solution in a sufficiently small number of iterations. In the end, the advantage of the smaller clique size of our model dominates the disadvantage caused by the need for more than a single iteration.
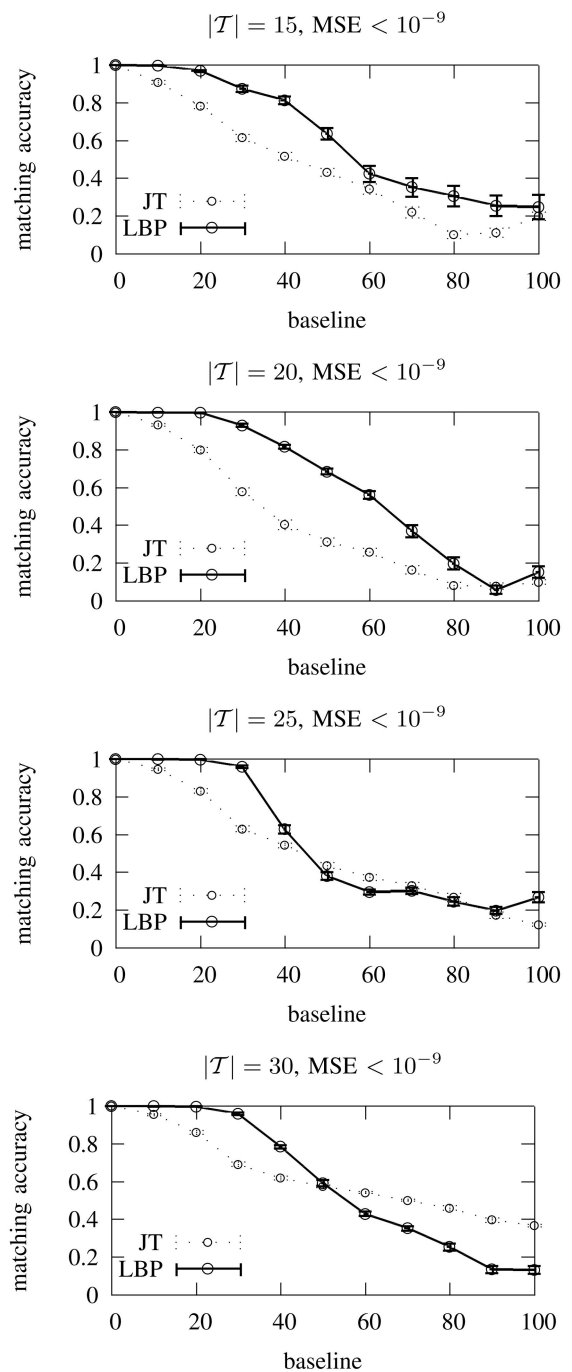
13. http://vasc.ri.cmu.edu/idb/html/motion/house/index.html.

Fig. 8. Matching accuracy of our model using the "house" data set, as the baseline (separation between frames) varies.

REFERENCES

[1] T.S. Caetano, T. Caelli, D. Schuurmans, and D.A.C. Barone, "Graphical Models and Point Pattern Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 10, pp. 1646-1663, Oct. 2006.
[2] Y. Amit and A. Kong, "Graphical Templates for Model Registration," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 3, pp. 225-236, Mar. 1996.
[3] M. Carcassoni and E.R. Hancock, "Point Pattern Matching with Robust Spectral Correspondence," *Computer Vision and Pattern Recognition,* vol. 1, p. 1649, 2000.
[4] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 509-522, Apr. 2002.
[5] A. Robles-Kelly and E.R. Hancock, "Point Pattern Matching via Spectral Geometry," *Syntactical and Structural Pattern Recognition, Statistical Pattern Recognition,* pp. 459-467, 2006.

[6]	A. Rangarajan, J. Coughlan, and A.L. Yuille, "A Bayesian Network Framework for Relational Shape Matching," *Proc. Int'l Conf. Computer Vision (ICCV '03)*, p. 671, 2003.

[7]	P.F. Felzenszwalb, "Representation and Detection of Deformable Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 208-220, Feb. 2005.

[8]	D. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial Priors for Part-Based Recognition Using Statistical Models," *Proc. Int'l Conf. Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 10-17, 2005.

[9]	R. Connelly, "Generic Global Rigidity," *Discrete and Computational Geometry*, vol. 33, no. 4, pp. 549-563, 2005.

[10]	S. Gold and A. Rangarajan, "A Graduated Assignment Algorithm for Graph Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 377-388, Apr. 1996.

[11]	S.L. Lauritzen, *Graphical Models (Oxford Statistical Science Series)*. Oxford Univ. Press, July 1996.

[12]	C.M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, Aug. 2006.

[13]	T.S. Caetano and T. Caelli, "A Unified Formulation of Invariant Point Pattern Matching," *Proc. Int'l Conf. Pattern Recognition (ICPR '06)*, pp. 121-124, 2006.

[14]	Y. Weiss, "Correctness of Local Probability Propagation in Graphical Models with Loops," *Neural Computation*, vol. 12, pp. 1-41, 2000.

[15]	A.T. Ihler, J.W. Fisher, and A.S. Willsky, "Message Errors in Belief Propagation," *Advances in Neural Information Processing Systems*, pp. 609-616, 2005.

[16]	Y. Weiss and W.T. Freeman, "On the Optimality of Solutions of the Max-Product Belief-Propagation Algorithm in Arbitrary Graphs," *IEEE Trans. Information Theory*, vol. 47, 2001.

[17]	J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Generalized Belief Propagation," *Advances in Neural Information Processing Systems*, pp. 689-695, 2000.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.