# An Optimal Probabilistic Graphical Model for Point Set Matching

by

Tibério S. Caetano
Terry Caelli
Dante A. C. Barone

Technical Report TR 04-03
February 2004

DEPARTMENT OF COMPUTING SCIENCE
University of Alberta
Edmonton, Alberta, Canada

# An Optimal Probabilistic Graphical Model for Point Set Matching

Tibério S. Caetano[†‡],Terry Caelli[†] and Dante A. C. Barone[‡]

†Department of Computing Science
University of Alberta
Edmonton, AB, Canada, T6G 2E8

‡Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brazil, CP 15064

March 10, 2004

## Abstract

*We present a probabilistic graphical model for point set matching. By using a result about the redundancy of the pairwise distances in a point set, we represent the binary relations over a simple triangulated graph that retains the same informational content as the complete graph. The maximal clique size of this resultant graph is independent of the point set sizes, what enables us to perform exact inference in polynomial time with a Junction Tree algorithm. The resulting technique is optimal in the Maximum a Posteriori sense. Experiments show that the algorithm significantly outperforms standard probabilistic relaxation labeling.*

## 1 Introduction

The Point Matching Problem is a fundamental one in structural Pattern Recognition, having many applications ranging from stereo matching techniques [1] to the analysis of electrophoresis images [2].

Many algorithms are available in the literature, both for exact and inexact matching. For exact matching, optimal polynomial time algorithms exist, most of them based on ingenious combinations of sorting and searching, inclusive for multidimensional point matching [3]. For inexact matching, there is a variety of proposed techniques, which are generally significantly different from exact techniques, since the need for a similarity measure usually requires that the problem be posed as an optimization one instead of (or in addition to) deterministic search [4].

This paper presents a principled approach for point set matching which is both applicable to exact and inexact problems. Moreover, it is assured to be optimal in the Maximum a Posteriori sense and has polynomial time dependency on the point set sizes. This is possible due to the proposed Markov Random Field (MRF) formulation, which poses point set matching as an exact inference problem that can be effectively solved by Junction Tree methods [5]. The polynomial time performance is obtained from a key observation which exploits concepts of the rigidity of straight line graph embeddings [6, 7]. We also show via a series of experiments that the proposed technique, when applied to inexact problems, presents extremely robust performance under augmentation of the point set sizes. In addition, experiments indicate that the proposed approach is significantly more robust than standard probabilistic relaxation labeling both under varying point set sizes and varying noise levels.

## 2 The Problem

We consider the problem in $\mathbb{R}^d$, $d \geq 2$, of finding the subset of an $S$-sized point set (the *codomain pattern*) that best matches another point set (the *domain pattern*) having $T$ points, where $T \leq S$. There may or may not exist distortions due to noise, but if there are, we assume no prior knowledge of the type of noise. We restrict the matching to be invariant up to isometries. In this work, the scale of both point sets is assumed to be the same. The only constraint enforced in the mapping is that it must be a total function: every point in the domain pattern must map to one point in the codomain pattern (but the opposite may not hold).

## 3 Theoretical Foundation

This section presents the fundamental result that enabled us to formulate the point set matching problem as one of optimal inference in a MRF, while keeping the overall complexity of the algorithm polynomial.

### 3.1 Global Rigidity: Basic Definitions

We start by presenting some basic definitions of the global rigidity of graphs [6]. A *configuration* is a finite set of $n$ labeled points, $\mathbf{p} = (p_1, \cdots, p_n)$, such that each $p_i \in \mathbb{R}^d$. A *framework* in $\mathbb{R}^d$ consists of a straight line embedding of a graph $G$ with $n$ vertices with configuration $\mathbf{p} = (p_1, \cdots, p_n)$, and is denoted by $G(\mathbf{p})$. In this representation the lengths of the edges correspond to the Euclidean distances between the corresponding vertices. A configuration in *general position* (or *general configuration*) in $\mathbb{R}^d$ is such that no ($d$+1) points lie in the same ($d$-1)-dimensional hyperplane. In $\mathbb{R}^2$, this means that no 3 points are collinear.

Two frameworks $G(\mathbf{p})$ and $G(\mathbf{q})$ are said to be *equivalent*, denoted by $G(\mathbf{p}) \equiv G(\mathbf{q})$, if when $\{i, j\}$ is an edge of $G$, then $||p_i - p_j|| = ||q_i - q_j||$, where $||.||$ is the Euclidean norm. It is said that a configuration $\mathbf{p} = (p_1, \cdots, p_n)$ is *congruent* to $\mathbf{q} = (q_1, \cdots, q_n)$, and are denoted by $\mathbf{p} \equiv \mathbf{q}$, if, for all $\{i, j\} \in \{1, \cdots, n\}$, $||p_i - p_j|| = ||q_i - q_j||$. This is equivalent to saying that congruent configurations are those related by an *isometry*, or a transformation that preserves distances. A framework $G(\mathbf{p})$ is called *globally rigid* in $\mathbb{R}^d$ if $G(\mathbf{p}) \equiv G(\mathbf{q})$ implies $\mathbf{p} \equiv \mathbf{q}$. In other words, a framework is globally rigid when the specification of the edge lengths uniquely specifies the remaining pairwise distances between vertices that are not joined by an edge. In the following we present a key fact about the global rigidity of a special kind of framework, which turns out to allow for the development of an effective technique for point matching.

### 3.2 Global Rigidity of $k$-trees

In order to present the basic result that allows us to develop a model for optimal point matching, we start by reviewing some basic definitions from graph theory [8]. In what follows a complete graph with $n$ vertices is denoted as $K_n$. We recall that a *framework* is a straight line embedding of a graph.

**Definition 1 ($k$-clique)** *A* k-clique *of a graph is a complete subgraph with k vertices.*

**Definition 2 ($k$-tree, base $k$-clique)** *A* k-tree *is a graph that arises from $K_k$ by zero or more iterations of adding a new vertex adjacent to each vertex of a k-clique in an older graph and nonadjacent to the remaining vertices. The k-cliques adjacent to the new vertices are called* base k-cliques*.*

Figure 1 shows the process of creating a $k$-tree, where $k = 3$. We start with a $K_3$ graph. Then we add new vertices by connecting them to 3 vertices of any existing base 3-clique. Note that all intermediate graphs generated in this way are themselves legitimate 3-trees.

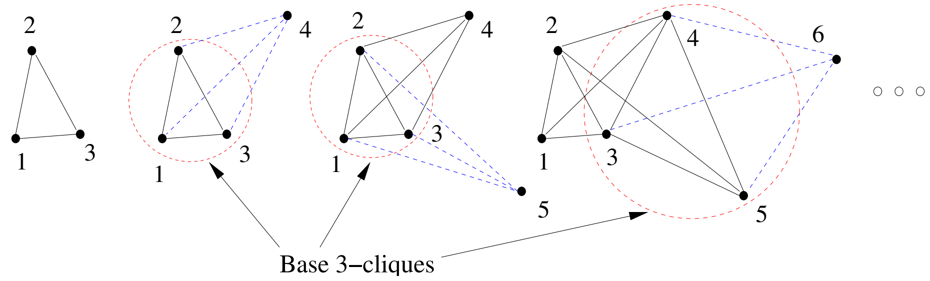


Figure 1: The process of constructing 3-trees

From these definitions and those of the global rigidity of frameworks, we can infer the following result:

**Proposition 1** *Any $k$-tree framework having each of its base $k$-cliques in general position in $\mathbb{R}^{k-1}$ is globally rigid in $\mathbb{R}^{k-1}$.*

This follows from the fact that $n+1$ hyper-spheres in $\mathbb{R}^n$ which do not lie in a $n-1$ vector subspace intersect in at most one point. If all points are in general position (not coplanar), then any $k$-tree is globally rigid. But, if not all subsets of points are in general position, the only globally rigid $k$-trees will be those where all the base $k$-cliques are themselves in general position.

The direct implication of the result is that the $k$-tree framework has *exactly* the same informational content than a fully connected framework (since the absent edges have uniquely determined lengths).

In the next section we show how, by taking advantage of this result, we can formulate a MRF model that has precisely the structure of a $k$-tree, and where exact probabilistic inference is feasible in polynomial time.

# 4 The Probabilistic Graphical Model

A direct consequence of the definition of a $k$-tree is that the size of its maximal clique is at most $k+1$, being precisely $k+1$ if the number of vertices is greater than $k$ and being $k$ if the number of vertices equals $k$.

This observation is what impelled us to propose a probabilistic formulation based on Graphical Models [9], which involves algorithms for optimal inference in probabilistic networks with exponential complexity on the size of the maximal clique of the underlying graph. Since the size of the maximal clique is fixed in $k+1$, the dependency on the number of points is only polynomial, as will be shown. As a result, we obtain a polynomial time procedure for optimal matching. The description of the model and the optimization procedure follow.

## 4.1 The Model

Here we present a probabilistic graphical model for point set matching. The cardinalities of the domain and codomain pattern sets are denoted, respectively, by $T$ and $S$. Each point in the domain is associated with a vertex of a graph $G_t$, and each point in the codomain is associated with a vertex of a graph $G_s$. The relative distance between a pair $\{i, j\}$ of points in a pattern is seen as an *edge attribute* of the edge that connects vertices $i$ and $j$ in the respective graph. In this formulation, point pattern matching turns out to be an attributed graph matching problem.

The model formulation consists, initially, in defining each of the $T$ vertices in $G_t$ as a random variable (*r.v.*) that can assume $S$ possible values (discrete states), corresponding to the vertices in $G_s$. The core of a Markov clique condition lies in the compatibilities between joint matches of two *r.v.*'s $x_i$ and $x_j$ in $G_t$ to values $\theta_k$ and $\theta_l$ in $G_s$ and is defined by

$$\psi_{ij;kl} = p(x_i = \theta_k | x_j = \theta_l)$$

or, in matrix form, for each pair $\{i, j\}$ in $G_t$, we define the *potential*

$$\psi_{ij} = \psi(x_i, x_j) \equiv p(x_i|x_j) = \frac{1}{Z} \begin{pmatrix} \mathcal{S}(y_t^{ij}, y_s^{11}) & \dots & \mathcal{S}(y_t^{ij}, y_s^{1S}) \\ \vdots & \ddots & \vdots \\ \mathcal{S}(y_t^{ij}, y_s^{S1}) & \dots & \mathcal{S}(y_t^{ij}, y_s^{SS}) \end{pmatrix} \tag{1}$$

where $y_a^{bc}$ denotes the edge attribute between vertices $b$ and $c$ in graph $G_a$. $Z$ is a normalization constant that equals the sum of all elements in the matrix, in order to keep $\psi_{ij}$ compatible with a probability distribution. $\mathcal{S}$ is a similarity function that measures the compatibility of the two arguments. Several options are available for $\mathcal{S}$ [10], and the specific function $\mathcal{S}$ to be used is not a central issue of this work. Here we choose the univariate Gaussian function:

$$\mathcal{S}(y_t^{ij}, y_s^{kl}) = \exp\left(-\frac{1}{2\sigma^2}||y_t^{ij} - y_s^{kl}||^2\right). \tag{2}$$

As a result we can now define a Markov Random Field (MRF) graphical model over the domain graph $G_t$ where nodes in the model correspond to the vertices in $G_t$ whose states are defined by discrete random variables given by the set of vertices in $G_s$. The local cliques are defined by the connections between the *r.v.'s* of the $k$-tree topology and their conditional dependencies (Markov condition) are defined via Eq.(1).

Since we have shown that considering only this subset of edges is equivalent to considering all the edges, this MRF actually represents a complete model of the probabilistic interactions in the graph. Figure 2 shows an example of a particular 3-tree MRF. The result of section 3 implies that this model is equivalent to a complete model for matching tasks in $\mathbb{R}^2$, where our experiments will take place. Each connection represents the interaction between the corresponding random variables, which is given by the associated "potential" $\psi_{ij}$.

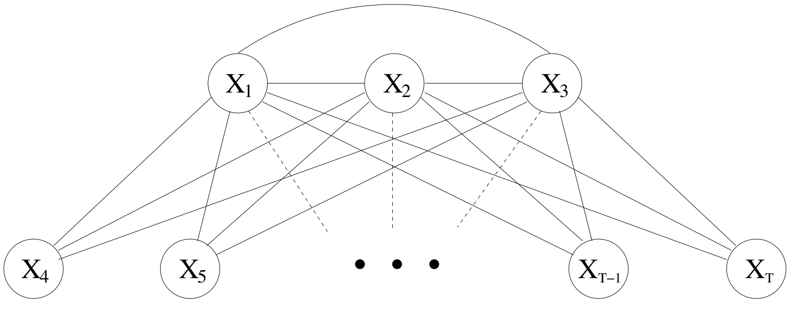


Figure 2: A possible $k$-tree model for $k = 3$ (3-tree). Other 3-trees are possible, as long as their base 3-cliques correspond to non-collinear points.

Now we need an optimization procedure to infer the most likely realization of the set of random variables, which is precisely the solution to the point set matching problem.

### 4.2 Optimization

Inference in MRFs typically capitalizes on the Gibbs' distribution to employ simulated annealing to derive the assignment [11]. However, with the above results we can use the Junction Tree (JT) framework, which provides a set of deterministic algorithms for *exact* inference in arbitrary graphical models [5, 9]. A JT of a graph is another graph where the nodes correspond to the maximal cliques of the former graph such that the *Junction Tree property* is satisfied. This property states that all the nodes in the path between any two nodes in the JT must contain the intersection of these two nodes. It is known that the condition for the existence of a JT is that the graph must be chordal [9]. A $k$-tree is a chordal graph, and this allows us to use the JT framework to perform optimization over the model.

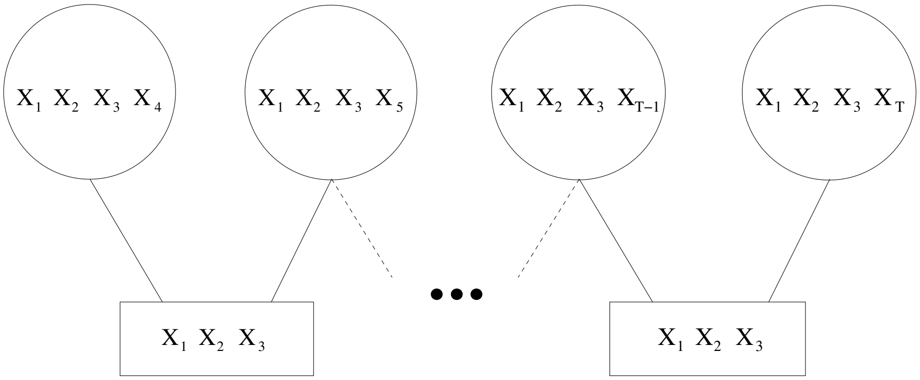


Figure 3: The Junction Tree obtained from the model in Figure 2.

Figure 3 shows a JT obtained from the model in Figure 2. The nodes of the JT are denoted by circles in which are listed the nodes of the original graph that correspond to the respective maximal cliques. The rectangles are the so-called *separators*, that contain the intersection of the nodes to which they are linked. Both the nodes and the separators are endowed with "clique potentials", and the optimization process consists in updating these potentials, as explained below.

In this paper we applied the HUGIN algorithm [9], an instance of the JT framework, to accomplish exact inference in the 3-tree model shown in Figure 2. The complexity of JT using HUGIN in our $k$-tree model is $O(S^{d+2}T)$, where $d$ ($d > 1$, $d = k - 1$) is the dimension of the Euclidean space. As a result, the complexity on $S$ and $T$ is polynomial. For the model and experiments presented here (in $\mathbb{R}^2$), we have $O(S^4T)$. The HUGIN algorithm essentially works in two steps: initialization and message-passing. During the initialization, the clique potential of each separator ($\Phi$) is set to unity and the clique potential of each node ($\Psi$) is introduced. These clique potentials are assembled as an element-by-element product of the pairwise potentials (see Eq. 1) in the respective clique. For example, for the 3-tree model, $\Psi(x_i, x_j, x_k, x_l) = \psi(x_i, x_j).\psi(x_i, x_k).\psi(x_i, x_l)$.

The second step is the message-passing scheme, which involves a transfer of information between two nodes $V$ and $W$ [5]. This operation is defined by the following equations:

$$\Phi_S^* = \max_{V \setminus S} \Psi_V \qquad\qquad \Psi_W^* = \frac{\Phi_S^*}{\Phi_S} \Psi_W$$

where we used standard notation for the current and updated (*) versions of the separator potentials ($\Phi$) and the clique potentials ($\Psi$). The first equation is a maximization over all sub-configurations in $\Psi_V$ that do not involve the singleton nodes which are common to $\Phi_S$ and $\Psi_V$. The second is simply a normalization step necessary to keep $\Psi_W$ consistent with the updated version of $\Phi_S$ (division and multiplication are performed element-by-element). The above potential update rules must respect the following protocol: a node $V$ can only send a message to a node $W$ when it has already received messages from all its other neighbors. If this protocol is respected and the equations are applied until all clique nodes have been updated, the algorithm assures that the resulting potential in each node and separator of the JT is equal to the (global) maximum a posteriori probability distribution of the set of enclosed singleton nodes [5]. In our particular case, we need the maximum probability for each singleton, what can be obtained by maximizing out the remaining 3 singletons in each of the nodes. The indexes for which the final potentials are maximum are considered the vertices in $G_s$ to which the corresponding vertices in $G_t$ must be assigned.

## 5 Experiments and Results

We have carried out two experiments. In both of them, we compare our technique (denoted simply as JT) with probabilistic relaxation labeling (PRL) [12]. We have implemented the standard algorithm for PRL presented in [13], for the same reasons than those explained in [14]. PRL does not guarantee global optimal assignments like, under the right conditions and enough iterations, simulated annealing does [11]. However, it is representative of the class of methods that locally update evidence, in parallel, for assignment in terms of the compatibilities between the label of each node and those within its neighborhood. The algorithm runs in $O(S^3T^2)$.

In both experiments, we generated codomain patterns in images of size 256x256 with $S$ points at random (but not coincident) positions. Then, we extracted randomly a subset of $T$ of these points to build the domain pattern.

In the first experiment, we assume that there is a small amount of noise (position jitter) in the codomain pattern, and vary the size of the codomain pattern (in the absence of noise - when the domain pattern is exactly related via an isometry to a subset of the codomain pattern - our method always gives perfect results). A set of 8 increasingly complex matching tasks were carried out, where patterns of size ($T$,$S$) = $\{(10,15)$, $(10,20)$, $(10,25)$, $(10,30)$, $(10,35)$, $(10,40)$, $(10,45)$, $(10,50)\}$ were matched using both JT and PRL. For each of these matching tasks, the fraction of correct assignment was calculated based on 1000 trials. The obtained performances are shown in Figure 4(a).

In the second experiment, the sizes of both graphs are kept constant, but the degree of noise in the codomain pattern is increased. We used $T = 10$ and $S = 20$, and the noise consisted in adding independent random numbers drawn from a normal distribution with zero mean and varying standard deviation (denoted as *std* in the figures) to both the $x$ and $y$ coordinates of each point from the codomain pattern. The standard deviation was progressively set to $\{1,2,3,4,5,6,7,8,9,10\}$ pixels. The final performance is presented in Figure 4(b). Each point in this Figure also represents the result over 1000 trials. The only parameter in the model is $\sigma$ (see Eq. 2), and it was estimated so that the minimal possible value for $\mathcal{S}(y_t^{ij}, y_s^{kl})$ was $10^{-12}$. This is done to prevent underflow and also to guarantee that the similarity function $\mathcal{S}$ will effectively behave monotonically with the observed value of $|y_t^{ij} - y_s^{kl}|$.

## 6 Discussion

Figure 4(a) shows that, for a fixed amount of noise, the augmentation of the codomain pattern size damages severely the performance of PRL, whereas that of JT remains significantly robust. This is a very important result, since scalability is an important factor in real applications such as stereo matching and image registration. The sensitivity of PRL to matching problems involving many elements has already been reported [14].

In Figure 4(b), we observe that, for fixed sizes of the domain and codomain patterns, the increasing of additive noise still keeps JT preferable for all the experimented values of noise. It is reasonable to expect that both techniques will perform similarly for extremely severe perturbations, when the performance cannot exceed significantly that of pure choice.

# 7 Conclusion

This work has presented a novel technique for both exact and inexact point pattern matching in $\mathbb{R}^{k-1}$ ($k \geq 3$), which runs in polynomial time and is optimal in the Maximum a Posteriori sense. By representing a point pattern with the correspondent relative pairwise distances between them, we showed that a subset of these distances is sufficient for uniquely determining the remaining ones. From this result, a special class of graph emerges, a $k$-tree, which has the same representational power as the full graph, but has a maximal clique limited to size $k+1$. By exploiting the Markovian properties of this simple graph structure which has a fixed maximal clique size, we developed a probabilistic graphical model where optimal inference is realizable in polynomial time. The proposed technique presents perfect results in the absence of noise, and is much more robust than standard probabilistic relaxation labeling to varying point set sizes when under noise. The technique is also robust under augmentation of additive noise, where it clearly outperforms standard probabilistic relaxation labeling.

# Acknowledgements

We are grateful to NSERC/Canada and CAPES/Brazil for financial support.

# References

[1] D. Reimann and H. Haken, "Stereo vision by self-organization," *Biological Cybernetics*, vol. 71, no. 1, pp. 17–26, 1994.

[2] T. Akutsu, K. Kanaya, A. Ohyama, and A. Fujiyama, "Point matching under non-uniform distortions," *Discrete Applied Mathematics*, vol. 127, pp. 5–21, 2003.

[3] P. J. de Rezende and D. T. Lee, "Point set pattern matching in d-dimensions," *Algorithmica*, vol. 13, pp. 387–404, 1995.

[4] B. Luo and E. R. Hancock, "Structural graph matching using the em algorithm and singular value," *IEEE Trans. PAMI*, vol. 23, no. 10, pp. 1120–1136, 2001.

[5] M. I. Jordan, *An Introduction to Probabilistic Graphical Models*, in preparation.

[6] R. Connelly, *Generic Global Rigidity*, preprint.

[7] B. Jackson and T. Jordán, *Connected Rigidity Matroids and Unique Realizations of Graphs*, citeseer.nj.nec.com/jackson03connected.html.

[8] D. B. West, *Introduction to Graph Theory, 2nd Edition.* Upper Saddle River, NJ: Prentice Hall, 2001.

[9] S. L. Lauritzen, *Graphical Models.* New York, NY: Oxford University Press, 1996.

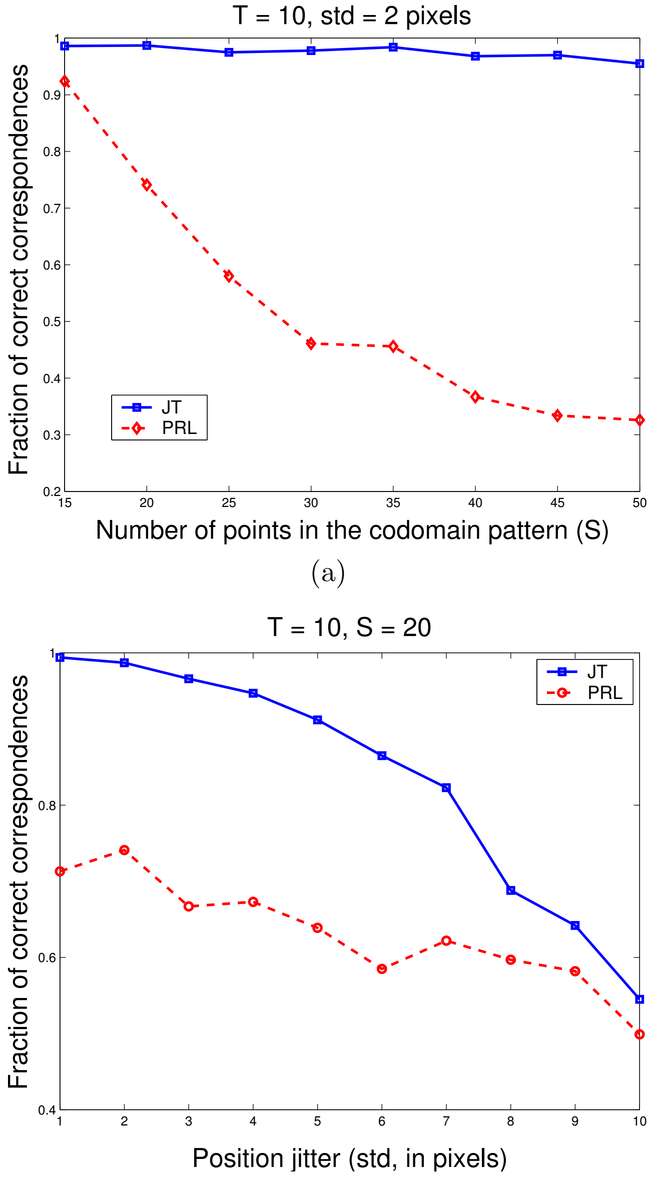
[10] M. Carcassoni and E. R. Hancock, "Spectral correspondence for point pattern matching," *Pattern Recognition*, vol. 36, pp. 193–204, 2003.

[11] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. PAMI*, vol. 6, no. 6, pp. 721–741, 1984.

[12] W. J. Christmas, J. Kittler, and M. Petrou, "Structural matching in computer vision using probabilistic relaxation," *IEEE Trans. PAMI*, vol. 17, no. 8, pp. 749–764, 1994.

[13] A. Rosenfeld and A. C. Kak, *Digital Picture Processing, Vol. 1.* New York, NY: Academic Press, 1982.

[14] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Trans. PAMI*, vol. 18, no. 4, pp. 377–388, 1996.



(a)

(b)

Figure 4: (a) Performances of JT and PRL when the size of the codomain pattern ($S$) is increased; $T = 10$, std (noise jitter) = 2 pixels. (b) Performances of JT and PRL when the position jitter (std) in the codomain pattern is increased; $T = 10$, $S = 20$.