Approximating the problem, not the solution: an alternative view of point set matching

Tibério S. Caetano and Terry Caelli

National ICT Australia, Canberra ACT0200, Australia

Abstract. This work discusses the issue of approximation in point set matching problems. In general, one may have two classes of approximations when tackling a matching problem: a representational approximation, which involves a simplified and suboptimal modeling for the original problem, and algorithmic approximation, which consists in using suboptimal algorithms to infer the assignment. Matching techniques in general have relied on the second approach: to keep a complete model of the original problem and use suboptimal techniques to solve it. In this paper, we show how a technique based on using exact inference in simple graphical models, which is an instance of the first class, can significantly outperform instances of techniques from the second class. We give theoretical insights of why this happens, and experimentally compare our approach with the well-known Shapiro and Brady and Christmas et al. methods, which are exemplars of the second class. We perform experiments with synthetic and real-world data sets, which reveal a significant accuracy improvement of the proposed technique both under point position jitter and size increasing of the point sets. The main conclusion is that techniques based on optimal algorithms with appropriate suboptimal representations may lead to better results than their counterparts which consist in using optimal representations, but approximate algorithms.

1 Introduction

The point set matching problem consists in finding correspondences between two point sets, which may be one-to-one or also many-to-one [1, 2], and arises in a variety of real-world vision tasks such as stereo vision, registration, model-based object recognition and the like. In any real vision problem we are faced with *inexact* point set matching, or matching under structural corruption, which is known to be an NP-hard problem [3]. As a result, one must rely on approximate techniques do derive the "best" assignment in some suboptimal sense. Several approaches for solving matching problems, and in particular the inexact point set matching problem, have been proposed along the years [4]. Major representatives are *spectral* methods [5, 6] and *relaxation labeling* methods [7, 8]. These families of techniques consist in encoding all the available information into some complete representation of the problem and subsequently using approximate algorithms to derive the assignment. Several limitations have been reported with respect

to spectral methods when structural corruption is present and with respect to relaxation methods when matching large point sets [1, 6, 9, 10].

This paper shows how a shift in point of view can lead to an alternative method that overcomes many of the limitations existent in some spectral and relaxation methods. Essentially, instead of using an optimal representation and an approximate algorithm, we do the opposite. We model the structure of points by a sparse representation that deliberately disregards a particular set of relational information that is actually available. In other words, we *approximate the problem*. The reason for that becomes clear in the next step: by taking advantage of this sparsity, we are able to apply an *optimal algorithm* to derive the best assignment. As a result, we advocate in favor of approximating the problem instead of the algorithm for solution.

For performance evaluation, we conducted experiments with synthetic and real world data sets, where we compared the proposed approach with traditional versions of spectral and relaxation methods, namely the Shapiro and Brady spectral method [5] and the Christmas et al. relaxation method [8]. Results indicate that the accuracy of the results obtained with the proposed technique significantly exceeds that obtained by the alternative techniques, either under structural corruption by point position jitter or under augmentation of the point set sizes.

2 Problem Definition

We consider the problem in, $\mathbb{R}^2$, 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, so we do not consider scaling. 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 The Model

The basic idea of the modeling strategy is to consider an undirected probabilistic graphical model (a Markov random field) where the nodes are points in the domain pattern and their possible realizations are points in the codomain pattern. In order to fully specify a graphical model, it is necessary to define (i) the potential functions and (ii) the connectivity of the model [11]. We start by formally specifying the model and introducing the potential functions.

3.1 Potential functions

The cardinalities of the domain and codomain pattern sets are denoted, respectively, by T and S. Each point d_i in the domain is associated with a vertex of

a graph G_d, and each point c_k in the codomain is associated with a vertex of a graph G_c. The relative Euclidean distance between a pair $\{d_{i_1}, d_{i_2}\}$ of points in the domain pattern is denoted as $y^d_{i_1 i_2}$. Analogously for the codomain pattern, we have that $y^c_{k_1 k_2}$ is the distance between points c_{k_1} and c_{k_2}. These distances are seen as *edge weights*. In this formulation, point pattern matching turns out to be a weighted graph matching problem.

The model formulation consists, initially, in defining each of the T vertices in G_d as a random variable that can assume S possible values (discrete states), corresponding to the vertices in G_c. Note that in this formulation the solution to the problem (the best match) corresponds to finding the most likely (the best) realization of the set of random variables.

Figure (1) illustrates a pairwise map and a possible measure which is relevant in order to construct the potential functions ($|y^d_{i_1 i_2} - y^c_{k_1 k_2}|$).

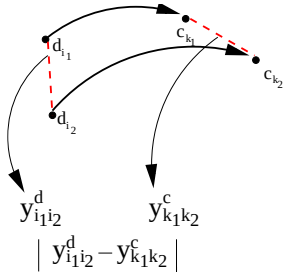

Fig. 1. An example of a pairwise mapping. An appropriate potential function should penalize more severely mappings for which $|y^d_{i_1 i_2} - y^c_{k_1 k_2}|$ is higher

Since each node in the domain graph can map to S different nodes in the codomain graph, each pair of nodes can map to S^2 different pairs in the codomain graph. Figure (2) illustrates the kernel structure of our model: a pairwise clique, where each random variable represents a point in the domain graph which in turn can assume a set of S possible realizations (which themselves correspond to points in the codomain graph).

Fig. 2. The kernel structure of the graphical model

The sample space for this clique has S^2 elements, corresponding to all possible combinations that a pair of points in the domain graph can map to in the codomain graph. A potential function is a function that associates to each

element of the sample space a positive real number. In our case, the only requirement that the potential function must obey is that its value must be as higher as more similar are the distances of the mapped edges, as illustrated in Figure (1).

Formally, we can specify the potential function by

$$\psi_{ij;kl} = p(X_i = x_k | X_j = x_l), \tag{1}$$

or, in matrix form, for each pair $\{X_i, X_j\}$ in G_d, we define

$$\psi_{ij} = \psi_{ij}(X_i, X_j) = \frac{1}{Z} \begin{pmatrix} \mathcal{S}(y_{ij}^d, y_{11}^c) & \dots & \mathcal{S}(y_{ij}^d, y_{1S}^c) \\ \vdots & \ddots & \vdots \\ \mathcal{S}(y_{ij}^d, y_{S1}^c) & \dots & \mathcal{S}(y_{ij}^d, y_{SS}^c) \end{pmatrix}, \tag{2}$$

where y_{bc}^a denotes the edge weight between vertices with indexes 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 ψ_{ij} compatible with a probability distribution. $\mathcal{S}$ is a similarity function that measures the compatibility of the two arguments. We use here the Gaussian function,

$$\mathcal{S}(y_{ij}^d, y_{kl}^c) = \exp\left(-\frac{1}{2\sigma^2}|y_{ij}^d - y_{kl}^c|^2\right). \tag{3}$$

This "proximity measure" is needed in order to model the uncertainty due to the presence of noise. Obviously, its maximal value must be reached when there is no noise ($y_{ij}^d = y_{kl}^c$).

Having specified the potential functions, it remains to be determined the connectivity of the graphical model: which nodes will be neighbors in the model?

3.2 Connectivity

We have derived elsewhere [2, 4] a particular connectivity of the graphical model (a sparse graph) which has a very unique property: it is provably *optimal* in the particular case of exact matching. By optimal we mean the fact that the optimization problems over this particular sparse graph and over the fully connected graph are one and the same. This is important, because the optimization problem in the sparse graph can be solved in polynomial time, whereas the one over the complete model is NP-complete [4]. This equivalence holds strictly only for exact matching, and it is still an open issue to determine theoretically how close to optimal is the solution for inexact matching problems. Nevertheless, there is experimental evidence that for small to moderate noise the results are impressively good [4].

Here we use this particular subgraph as the connectivity pattern for the Markov random field under consideration. The graph topology is depicted in Figure (3).

The resulting graph is technically a 3-tree [12], which is a graph that arises by adding new nodes that are connected to precisely 3 existent nodes (these 3

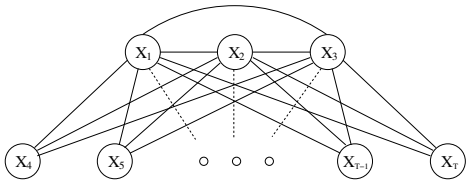

Fig. 3. A 3-tree graphical model. Each of the T nodes is a random variable representing a point in the domain graph. Each variable can assume S possible realizations, corresponding to points in the codomain graph

nodes must form a clique, i.e. every pair must be connected). In the example in Figure (3), X_1, X_2 and X_3 are the 3 "reference" nodes to which the additional $T-3$ nodes are connected.

Given the graph connectivity and the pairwise potential functions, the model is defined. The last step then consists in inferring what is the most likely joint realization of all the random variables for the given connectivity and set of potentials of the model. This is precisely the MAP inference problem in this model, whose solution represents the best assignment in the point set matching task and is described in what follows.

3.3 MAP Computation

The Junction Tree framework provides a set of deterministic algorithms for *exact* inference in arbitrary graphical models [11, 13]. Here we use an algorithm from this framework in order to find the optimal MAP estimate for the model in Figure (3). A Junction Tree of a graph is another graph where (i) the nodes correspond to the maximal cliques of the former graph (a maximal clique is a clique which is not a proper subset of another clique) and (ii) the *running intersection property* is satisfied. This property states that all the nodes in the path between any two nodes in the Junction Tree must contain the intersection of these two nodes. It is known that the condition for the existence of a Junction Tree is that the graph must be *chordal*, or *triangulated* [13]. A chordal graph is a graph with no chordless cycles[1]. The 3-tree is a chordal graph, and this allows us to use the Junction Tree framework to calculate the MAP estimate of the random variables in the model.

Figure (4) shows a Junction Tree obtained from the model in Figure (3). The nodes of the Junction Tree 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.

[1] A chord in a cycle is an edge between two non-consecutive nodes in the cycle.

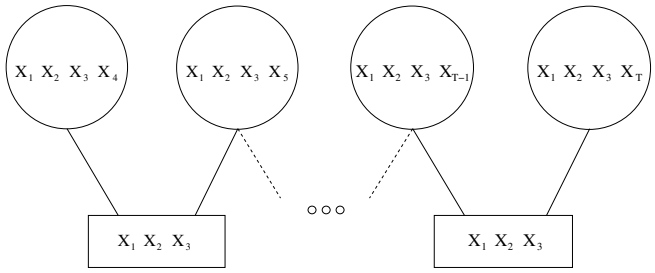

Fig. 4. The Junction Tree obtained from the model in Figure (3)

In this paper we applied the Hugin algorithm [13], an instance of the Junction Tree framework, to accomplish exact inference in the 3-tree model shown in Figure (3). The complexity of Junction Tree using Hugin in our 3-tree model is $O(S^4T)$. As a result, the complexity on S and T is polynomial. The Hugin algorithm essentially works in two steps: initialization and message-passing. During the initialization, the clique potential of each separator (Φ) is set to unity and the clique potential of each node (Ψ) is introduced (see subsection 3.1). These last clique potentials are assembled as an element-by-element product of the pairwise potentials (see Eq. 2) 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 in a systematic way until every pair of nodes in the Junction Tree has participated in the process [11]. 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 (Φ) and the clique potentials (Ψ). The first equation is a maximization over all sub-configurations in Ψ_V that do not involve the singleton nodes which are common to Φ_S and Ψ_V. The second is simply a normalization step necessary to keep Ψ_W consistent with the updated version of Φ_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 Junction Tree is proportional to the (global) maximum a posteriori probability distribution of the set of enclosed singleton nodes [11]. The constant of proportionality is guaranteed to be the same for every node, what implies that the mode of the local potentials will correspond to the MAP estimate. 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_c to which the corresponding vertices in G_d must be assigned.

4 Experiments and Results

We have carried out two sets of experiments, one with synthetic point sets and another with real-world data. In both of them, we compare our technique (denoted simply as JT) with the probabilistic relaxation labeling version described in [8], denoted as PRL, as well as with the spectral method of Shapiro and Brady [5], denoted as SB. These two methods encode all the pairwise distances in their model representation, whereas our method only encodes those distances that correspond to the 3-tree topology. On the other hand, our approach uses a non-iterative algorithm which is optimal, whereas the other two are based on approximate and heuristic algorithms. Results show how these different approximation principles affect accuracy in point set matching.

4.1 Synthetic data

In the experiments with synthetic data, we generated random points according to a bivariate uniform distribution in the interval $x = [0, 1]$, $y = [0, 1]$. We carried out two experiments: one with graphs of equal sizes and another with graphs with different sizes (the SB method is not suited for different graph sizes). In the first experiment, with equal sized graphs, we used graphs of sizes (10,10), (20,20) and (30,30) nodes. Then, for each of these 3 instances, we perturbed the codomain pattern with progressive levels of noise in the position of the nodes (white Gaussian noise with varying standard deviation). This setting allows us to have an idea of the relative performances both under the augmentation of the graph sizes and under progressive structural corruption of the patterns. Figure (5)-left shows the obtained curves under these experimental conditions. Each point in a curve is the average over 100 trials.

The graphs show that differential noise increasing has a very different impact on SB when compared to JT and PRL. JT and PRL performances are much less affected than that of SB for a same amount of incremental noise. Also, it is possible to note that scaling up the sizes of the graphs, for small levels of noise ($std = 0 - 1$), practically does not affect the performance of JT, whereas the performances of PRL and SB are significantly affected. This may suggest that the proposed technique may be a serious alternative to these other approaches in circumstances where the noise involved is not high. For larger, but still moderate amount of noise ($std = 1 - 4$), the proposed technique still dominates the others. It is clear however that PRL has similar or possibly superior performance for extremely high levels of noise. This is probably due to the fact that under severe noise the 3-tree approximation becomes poor. However, note that severe stochastic perturbation is not a common issue in point set matching problems. Usually, in real applications like stereo matching, shape matching or registration,

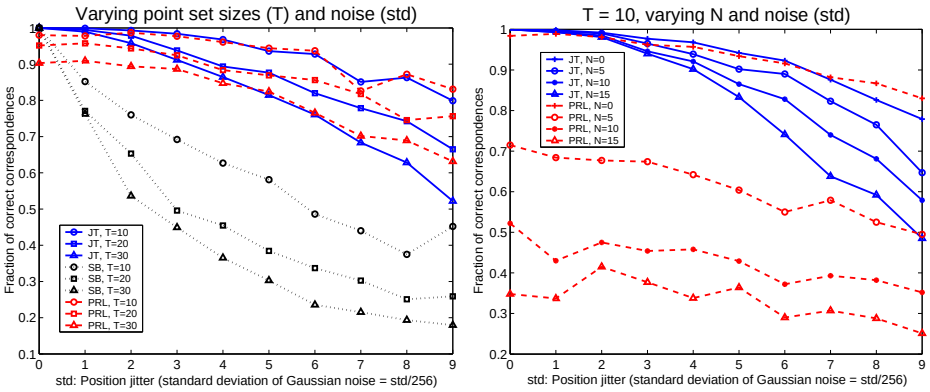

Fig. 5. Left: performances of JT, PRL and SB when the noise (position jitter) increases, for various sizes of the graphs ($T = S$ in this experiment). Right: Performances of JT and PRL when the noise (position jitter) increases, for fixed size of the domain graph (T) and various sizes of the codomain graph ($T + N = S$)

the point sets differ essentially by some isometric, affine or projective transformation (possibly together with a non-linear deformation), but the stochastic perturbation in the position of the points is itself small or at most moderate [4].

In the second experiment with synthetic point sets, we kept constant the size of the domain point set (10 nodes) and varied the size of the codomain point set (from 10 to 25 nodes in steps of 5). The introduction of noise by position jitter was analogous to the equal-size graphs experiment described above. In this experiment, we only compared JT with PRL, since SB is not suited for graphs with different sizes. Figure (5)-right shows the graphs corresponding to this experiment. Each point in a curve corresponds to the average over 100 trials. In this experiment, it is clear that PRL does not perform well when the sizes of the point sets are significantly different. It is also possible to notice the fact that for larger graphs the proposed technique has a very superior performance for small levels of noise.

4.2 Real-world data

In the real-world experiments, we performed comparisons of the algorithms using the CMU house sequence, as done in [1]. Figure (6) shows the images used in the experiments.

In total, 29 landmark points were manually marked in each of the images. Then we run the three algorithms (JT, PRL and SB) in several pairs of them, in a systematic way, described as follows. First we matched pairs that are consecutive in Figure (6) (1-11, 11-21, 21-31, 31-41, 41-51). Then we matched pairs separated by a single image (1-21, 11-31, 21-41, 31-51), by two images (1-31, 11-41, 21-51), by three images (1-41, 11-51) and finally by four images (1-51). For each of

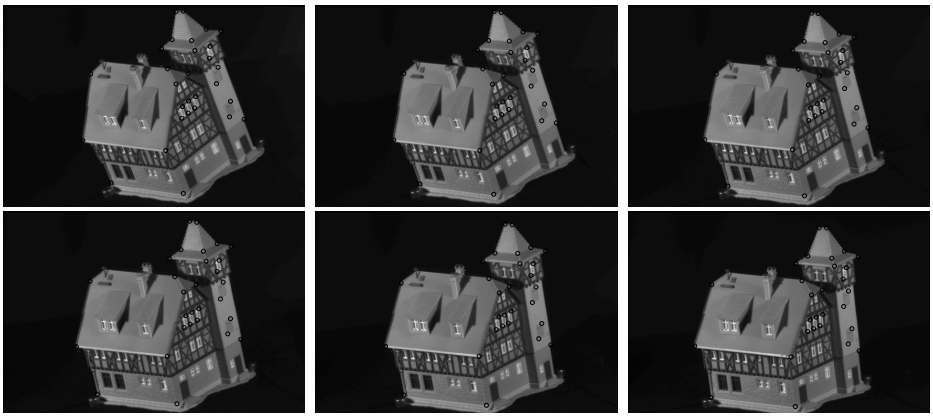

Fig. 6. Images from the CMU house data set (from left to right and top to bottom: images 1, 11, 21, 31, 41 and 51)

Table 1. Matching results (correct correspondences out of 29)

Image pair	1 - 11	11 - 21	21 - 31	31 - 41	41 - 51	1 - 21	11 - 31	21 - 41	31 - 51
JT	29	29	29	29	29	28	28	29	29
PRL	29	29	29	29	29	28	28	28	28
SB	19	16	18	23	23	17	16	15	14

Image pair	1 - 31	11 - 41	21 - 51	1 - 41	11 - 51	1 - 51
JT	28	28	29	27	25	25
PRL	28	26	27	26	26	25
SB	21	14	14	11	13	7

the experiments, the amount of correct correspondences for each technique was recorded and is shown in Table 1.

5 Discussion

A general understanding of the results is possible if we pay attention to a few key observations. PRL is an heuristic iterative optimization procedure that for problems with big search spaces may not converge in a feasible amount of iterations (even when it does, it reaches only a local optimum). In fact, in all experiments we have used 200 iterations for PRL, what is already considered to be a very large number compared to its use in many applications [8]. The SB method is non-iterative and also effective in the noiseless case, but it is clear from the experiments that the eigen-structure of the point proximity matrix does not provide robust features for matching under (even very small) noise. On the other hand, the proposed technique is non-iterative and always finds the optimal solution for a model that itself is optimal in the limit of zero noise. Thus it is reasonable to

expect that for small levels of noise the solution will be "close" to the optimal solution (although the precise meaning of "close" is not yet clear, as already mentioned), what is strongly suggested by the virtually perfect performance in the range $std = [0, 1]$. Current efforts are being dedicated to obtain theoretical results on how the 3-tree approximation deteriorates as noise increases.

6 Conclusion

In this work, we have investigated how different sources of approximation affects the performance of point set matching methods. Usual approaches to point set matching, such as spectral and relaxation-based methods, encode all the available information in the model representation, but rely on approximate algorithms for deriving the assignment. Our method, in contrast, consists in approximating the problem itself such that the resulting representation is suitable for the use of optimal algorithms for finding the match. Our method consists in modeling the relational features of a point set in a Markov random field framework where the underlying graph structure is sparse and allows for optimal MAP computation in polynomial time. Experiments were performed both with synthetic and real-world data sets, which indicate that the proposed "approximate model - optimal algorithm" approach is a serious alternative to other "optimal model - approximate algorithm" approaches.

References

1. Wang, H., Hancock, E. R.: A kernel view of spectral point pattern matching. Workshop on S+SSPR (2004). LNCS 3138, 361–369
2. Caetano, T. S., Caelli, T., Barone, D. A. C.: An optimal probabilistic graphical model for point set matching. Workshop on S+SSPR (2004). LNCS 3138, 162–170
3. Akutsu, T., Kanaya, K., Ohyama, A., Fujiyama, A.: Point matching under non-uniform distortions. Discrete Applied Mathematics **127** (2003) 5–21
4. Caetano, T. S.: Graphical models and point set matching. Ph.D. Thesis, Universidade Federal do Rio Grande do Sul, Brazil (2004)
5. Shapiro, L., Brady, J. M.: Feature-based Correspondence - An Eigenvector Approach. Image and Vision Computing **10** (1992) 283–288
6. Carcassoni, M., Hancock, E. R.: Spectral correspondence for point pattern matching. Pattern Recognition **36** (2003) 193–204
7. Rosenfeld, A., Kak, A. C.: Digital Picture Processing, Vol. 1. Academic Press, New York, NY (1982)
8. Christmas, W. J., Kittler, J., Petrou, M.: Structural Matching in Computer Vision Using Probabilistic Relaxation. IEEE Trans. PAMI **17** n. 8 (1995) 749–764
9. Caelli, T., Kosinov, S.: An eigenspace projection clustering method for inexact graph matching. IEEE Trans. PAMI **26** n. 4 (2004) 515–519
10. Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. IEEE Trans. PAMI **18** n. 4 (1996) 377–388
11. Jordan, M. I.: An Introduction to Probabilistic Graphical Models. In preparation
12. West, D. B.: Introduction to Graph Theory, 2nd Edition. Upper Saddle River, NJ. Prentice Hall (2001)
13. Lauritzen, S. L.: Graphical Models. Oxford University Press, New York, NY, (1996)