International Journal of Information Technology, Vol. 12 No. 2 2006

A Dynamic Generating Graphical Model for Point-Sets Matching

Xuan ZHAO, Shengjin WANG, and Xiaoqing DING

State Key Laboratory of Intelligent Technology and Systems,
Department of Electronic Engineering, Tsinghua University
{zhaoxuan,wsj, dxq}@ocrserv.ee.tsinghua.edu.cn

Abstract

This paper presents a new dynamic generating graphical model for point-sets matching. The existing algorithms on graphical models proved to be quite robust to noise but are susceptible to the effect of outliers. We discuss the separator's influences on point-sets matching in inference on graphical models theoretically and find that the separator which consists of outliers will break the message-passing, which will directly lead to the breaking down of the existing methods. Due to the conclusion above, in order to minimize the outliers in separator we propose a new algorithm in generating a graphical model and the corresponding Junction Tree for point-sets matching. The experimental results show that the proposed algorithm is significantly more stable and possesses higher accuracy on point-sets matching, which can overcome the limitation of sensitivity on outliers in the existing graphical models.

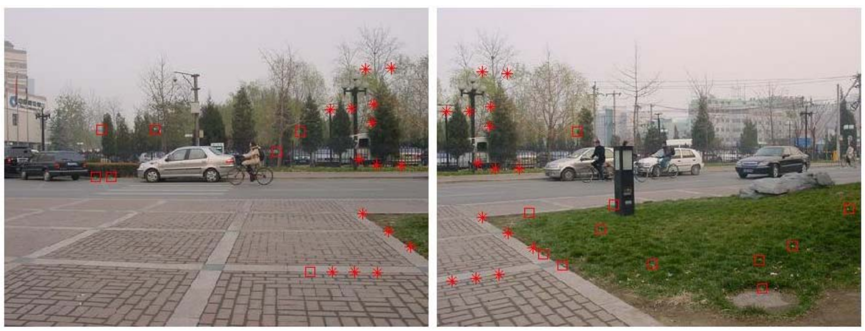
1 Introduction

Point-sets matching is a fundamental method for graph matching in many applications, such as stereo matching, panoramic mosaic, and object recognition.

The point-sets matching consists in finding correspondences between two point sets. Many researchers make their attempt to obtain a higher accuracy result and reduce the complexity of the computation [1-6]. In recent work, Caetano [1] have presented a probabilistic graphical model for point-sets matching, which is assured to be optimal in the Maximum a Posteriori sense and has polynomial time dependency on the point set sizes. Comparison [7] has been made between this method and standard probabilistic relaxation labeling (PRL) [6] using different forms of point metrics and under different levels of additive noise, which shows that Caetano's method is more effective than PRL.

The constraint to Caetano's model is that the mapping must be a total function: every point in the data graph must map to one point in the target graph, that is, no outliers are allowed. But in many applications such as panoramic mosaic, the constraint can not be assured, say, there are more or less outliers existing in the real world. The outliers are the feature points in either graph that have no counterparts in the other one, while "signal" points in this paper are the feature points which are not outliers. Fig. 1 shows two graphs to be mapped(Fig.1(a) is defined as data graph, while Fig.1(b) is defined as target graph), with feature points detected by Harris corner detector. In these two graphs, "signal" points are denoted by *, and outliers are denoted by □. Theoretically, the point matching between the two graphs can not be solved using Caetano's model. The experimental results in section 4 show that the performance of Caetano's algorithm gets worse when outliers increase in data graph.

The contribution of our paper is that we have discussed the reason of why Caetano's algorithm soon breaks down when there are outliers in data graph, and draw the conclusion that the selection of the separators in the graphical models will influence strongly the result of the mapping. A new algorithm is brought forward to generate graphical models which can make sure an optimal selection of the separators. The results demonstrate that the proposed technique outperforms Caetano's model.



(a) (b)

Fig. 1. "signal" points are marked by * in both graphs, they correspond to each other between two graphs. Outliers are marked by□.The outliers do not have corresponding points in the other graph.

2 Previous Work

2.1 Theory of Graphical Models and Junction Tree

A graphical model (Fig. 2(a)) is a graph endowed with joint probability distribution. The nodes in a graphical model represent random variables while the edges represent dependence between the nodes.

A clique (denoted by ellipse in Fig. 2) is a maximal subgraph with every pair of vertices joined. A Junction Tree (Fig. 2(b)) is a graph where the nodes correspond to the cliques such that the *running intersection property* is satisfied. This property states that if a variable is contained in two cliques, then it is contained in every clique along the path connecting. A separator (denoted by rectangle in Fig. 2) contains the intersection of the cliques to which they are linked. Inference refers to the problem of calculating the conditional probability distribution of a subset of the nodes in a graph given another subset of the nodes [8,9].

2.2 Graphical Model in Polynomial Time

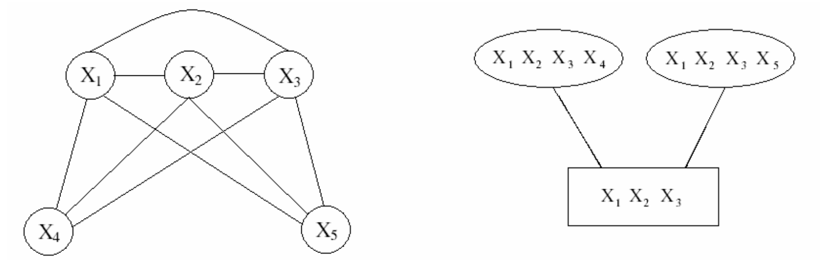
Inference on graphical model is a method of global optimality and is optimal in maximum a posteriori sense, but is limited in application for its N-P hard complexity. In 2004 Caetano proposed an optimal probabilistic graphical model for point-sets matching [1], which can perform exact inference in polynomial time. The algorithm defines a hidden Markov Random Field on the data graph. A single node in graphical model which corresponds to a feature point in the data graph is considered as a discrete random variable which can assume any point in the target graph. A Junction Tree can be obtained from the graphical model, both nodes and separators of which are endowed with "clique potentials" (which can be obtained from equation (1) in reference [1]). Then, Hugin algorithm [8] can be run to accomplish exact inference in graphical models. The algorithm assures that the resulting potential in each clique of the Junction Tree is equal to the (global) maximum a posteriori probability distribution of the set of enclosed singleton nodes. The corresponding one in target graph to which each point in data graph should be assigned can be obtained by observing the index that maximize out the other points of the clique in which it is contained[1].

3 Proposed Algorithm

3.1 The Separators' Influence in Junction Tree Inference

The Hugin algorithm works in two steps: initialization and message-passing. In the first step, the "clique potentials" are generated. In the second step, information is transferred between each clique. For all the cliques to be consistent with each other, we only need to ensure local consistency between neighboring cliques.

Suppose that we have two adjacent cliques A (consists of nodes x_1,x_2, x_3,x_4) and B(consists of nodes x_1,x_2 x_3,x_5) in a Junction Tree through a separator set S(x_1,x_2 x_3), as showed in Fig. 2. Assume that nodes x_1, x_2, x_3 are outliers, while x_4, x_5 are "signal" points. The message-passing operation of Junction Tree algorithm is an exchange of information between A and B with S serving as a conduit for the flow of information. If every point of separator is outlier, message can not pass from clique A to B through S. A brief proof is given bellow.



(a) (b)

Fig. 2 A simple graphical model (a) and the corresponding Junction Tree (b) with outliers and “signal” points.

Proof:

An outlier is independent on the other points on the Markov Random Field. That is: $x_1 \perp x_4$, $x_2 \perp x_4$, $x_3 \perp x_4$. Suppose that we observe the evidence $x_4 = e$, the update equation in Hugin algorithm is:

$$\Phi_B^* = \frac{P(x_1,x_2,x_3,x_5)}{P(x_1,x_2,x_3)} \max_{x_4} P(x_1,x_2,x_3,x_4=e)$$

$$= \frac{P(x_1,x_2,x_3,x_5)}{P(x_1,x_2,x_3)} \max_{x_4} [P(x_1,x_2,x_3 \mid x_4=e) \times P(x_4=e)]$$

$$= \frac{P(x_1,x_2,x_3,x_5)}{P(x_1,x_2,x_3)} \max_{x_4} [P(x_1,x_2,x_3) \times P(x_4=e)]$$

$$= \frac{P(x_1,x_2,x_3,x_5)}{P(x_1,x_2,x_3)} P(x_1,x_2,x_3) \times \max_{x_4} P(x_4=e)$$

$$= P(x_1,x_2,x_3,x_5)$$

$$= \Phi_B$$

The equation $\Phi_B^* = \Phi_B$ means that no message of x_4 is passed to x_5. So we can reach the conclusion that no message can pass through a separator without “signal” points. It can also easily to get the deduction that message can only pass the separator partially when not all of points in the separator are outliers.

When unluckily three outliers are chosen to form the common separator of Junction Tree model proposed by Caetano, nearly no information can be delivered between the “signal” points. The inference will reach the worst result.

Since the separators play an important role in the Junction Tree inference, they should not be selected randomly. Now methods are explored to minimize the outliers in the separators.

3.2 Generating Junction Tree Model Automatically Based on an Optimal Selection of the Separators

Given the case that we do not know which points in data graph are the "signal" points, a pre-mapping is required to find three points with a high probability to be the "signal" points. For the fraction of correct correspondence is not important here, 3-tree Junction Tree clique (max members of cliques is 3) is advised to reduce the running time. The complexity is $O(D^3T)$(D is the number of points in data graph, and T is the number of points in target graph). We present a new algorithm which is named *initializing* algorithm to select an optimal separator. The detailed process is as follows:

Initializing algorithm**:**

Step1: Use the Junction tree in Fig. 3 to run the inference from data graph to target graph. The result will be a total matching of point sets in data graph.

Step2: Use the Junction tree in Fig. 3 to run the inference from target graph to data graph. The result will be a total matching of point sets in target graph.

Step3: Find three pairs of points (D_i, T_j), with D_i matches to T_j in the result of step1 and T_j matches to D_i in result of step2. The D_i of the three pairs will form the first separator (D_i and T_i are feature point of data graph and target graph, respectively).

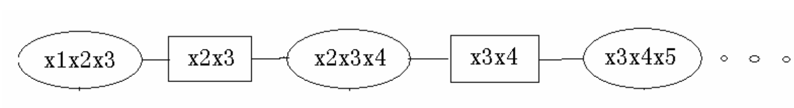


Fig. 3. A 3-tree model of Junction Tree

If we use the separator obtained in *initializing* algorithm to be the common separator in Caetano's model, it will significantly perform better than randomly selecting 3 points as the separator. But there still be some cases that not all members of the separator are "signal" points. In order to reduce the probability that all the separators are "bad" ones, using dynamic separators instead of constant one is a better choice. Here we propose an *updating* algorithm to generate a Junction Tree model with the dynamic separators. The generated Junction Tree will be in the chain shape, as showed in Fig. 4. The detailed process is as follows:

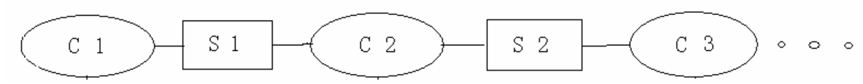


Fig. 4. Clique is denoted as C and separator is denoted as S.

Updating algorithm**:**

Step1: Randomly add one point to a separator to form the first clique. The separator consists of 3 selected points.

Step2: Compute the maximum of the joint probability distribution of every three point of the first clique, with the result of four numbers. The three joint points corresponding to the maximum of the four numbers will get a higher probability to be the "signal" points. Use the resulting 3 joint points as the separator between first clique and the second clique, and randomly select a point from the rest of the data graph to form the second clique with the separator.

Step3: Repeat step2 until all the points of data graph is contained in the Junction Tree.

Updating algorithm leads to higher probability of the message-passing. Both the *initializing* and *updating* processes will contribute to the performance of our algorithm. The two algorithms can be combined by using the separator obtained from *initializing* algorithm as the separator in step1 of *updating* algorithm. Experiments show that the combined algorithm outperforms each single algorithm.

3.3 Backward Mapping

Another problem brought by the outliers is that there may exist more than one point in data graph mapping to the same point in target graph, as shown in Fig.6 and Fig.8. Backward mapping is used to solve this problem. After the forward mapping (in section 3.2), we define a Markov Random Field on target graph and then do a point mapping from target graph to data graph. To pick out the outliers and reduce the complexity, the node set in the graphical model is only a subset of the original point set of the target graph, which can find a corresponding point in the data graph in the forward mapping. When D_i to T_i is the matching result of forward mapping and T_i to D_i is the matching result of backward mapping, the pair (D_i,T_i) is considered the final mapping result, where D_i and T_i is the feature point of data graph and target graph, respectively.

4 Experiments and Results

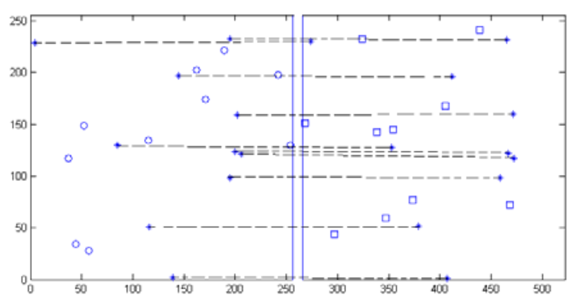
We carried out three experiments. In the first experiment, we generate both data graph and target graph in images of size 256*256. 10 pairs of "signal" points are added to both the graphs. The noise in each pair of "signal" points consisted in adding independent random numbers drawn from a normal distribution with zero mean and standard deviation 2 to both x and y coordinates. Then, we add 10 outliers to target graph and a varying of 1~10 outliers are added to data graph. The 10 pairs of "signal" points are linked by broken lines, as showed in Fig. 5. Fig.6 shows a mapping result using combination algorithm of *initializing* and *updating*, in which 10 pairs of "signal" points are rightly mapped. The mapping result after process of backward mapping is shown in Fig.7, in which one-to-one mapping is assured.

The second experiment is to compare our algorithms with Caetano's algorithm. The condition is the same as in the first experiment. Randomicity of the data graph and target graph will influence the fraction of the correct correspondence. To get rid of this, for every generated data graph and target graph, all algorithms are applied. The same similarity function and parameter σ (introduced in [1]) is used to make sure the impartiality of the comparison.

The fraction of correct assignment was calculated based on 100 trials. The obtained performances are shown in Fig. 8.

Figure 8 show that, the performance of Caetano's algorithm will degrade with the addition of outliers in data graph. Both the *initializing* algorithm and the *updating* algorithm will reach a better performance. The combination of them is significantly robust. Running time for our algorithm is 94s which is a little more than Caetano's algorithm (84s) in the same condition. Both of them run in $O(D^4T)$.

Our final experiment focuses on real world data. Fig.9 shows the correspondence matches obtained by combination algorithm with two panorama photos. The feature points shown in Fig.9 are subset of original Harris corner points, which are obtained by pre-matching using local gray information. Fig.10 shows the matching results with additive backward mapping.



(a) **(b)**

Fig.5. Data graph and Target graph when the number of outliers is set to 10. Points marked by *，o,□ are the point-sets to be matched, in which Points marked by o are outliers.

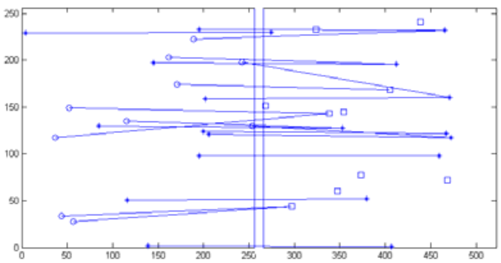


Fig. 6. Matching result of fig.5

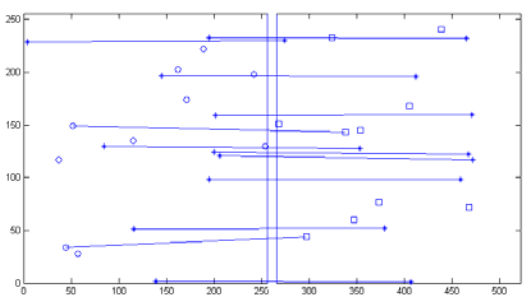


Fig.7. Matching result after running algorithm of inter-matching

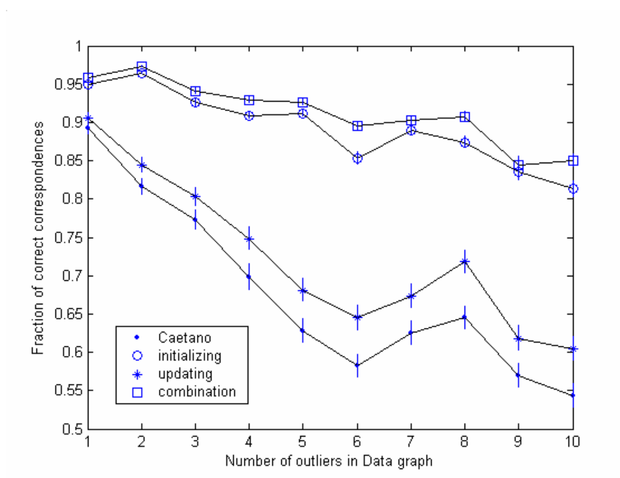


Fig. 8. Performances of Caetano’s algorithm and our algorithms. std is set to 2 pixels.

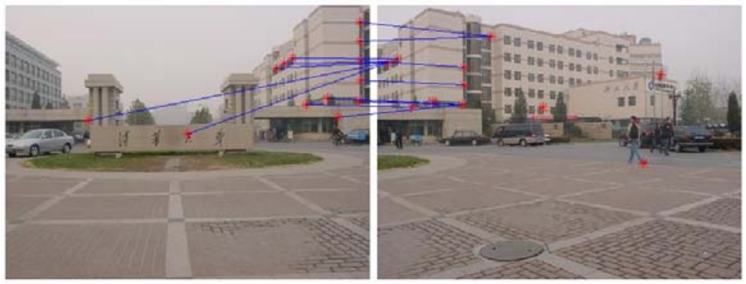


Fig.9. Matching of real world data

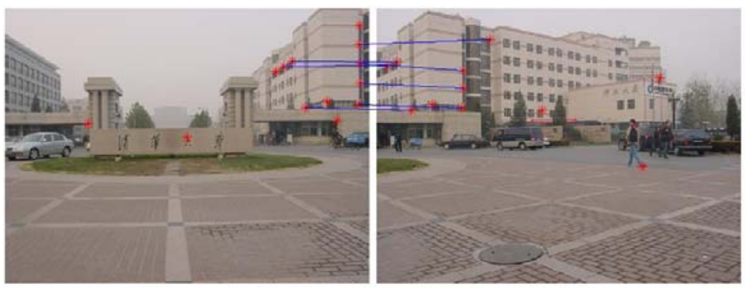


Fig.10. Matching of real world data after running algorithm of inter-matching, note that information of local gray is not used here.

5 Conclusion

In this paper, we have discussed how the separator influences the message-passing in the Junction Tree inference. Two ways are considered to improve the existing algorithm to select optimal separators. The combination of *initializing* algorithm and *updating* algorithm gives out the performances which is least influenced by the outliers in data graph. In order to solve the problem brought by the outliers that there may exist more than one point in data graph mapping to the same point in target graph, we have presented a backward mapping process after the forward mapping. The experimental results show that our algorithm is significantly more stable and possesses higher accuracy on point-sets matching than the existing algorithm on graphical models on both synthetic and real world data.

References

1. Caetano, T.S, Caelli, T., Barone, D.A.C.: An Optimal Probabilistic Graphical Model for Point Set Matching. Proceedings lecture notes in computer science 3138: 162-170, 2004
2. Carcassoni, M., Hancock, E.R.: Spectral correspondence for point pattern matching. Pattern Recognition 36 (2003) 193–204
3. Luo, B., Hancock, E.R.: Structural Graph Matching using the EM Algorithm and Singular Value Decomposition. IEEE Trans. PAMI 23 n.10 (2001) 1120–1136
4. Akutsu, T., Kanaya, K., Ohyama, A., Fujiyama, A.: Point matching under nonuniform distortions. Discrete Applied Mathematics 127 (2003) 5–21
5. Carcassoni, M., Hancock, E.R.: Spectral correspondence for point pattern matching. Pattern Recognition 36 (2003) 193-204
6. Christmas, W.J, Kittler, J., Petrou, J.: Structural matching in computer vision using probabilistic relaxation. IEEE Trans. PAMI, 17(8): 749-764, 1994
7. Caetano, T.S, Caelli, T., Barone, D.A.C.:A comparison of Junction Tree and Relaxation Algorithms for point matching using different distance metrics.Proceedings of the 17th International Conference on Pattern Recognition(ICPR'04) IEEE
8. Lauritzen, S.L.: Graphical Models. Oxford University Press, New York, NY, 1996
9. Jordan, Irwin, M.: Learning in Graphical Models. The MIT Press, Cambridge, Massachusetts, London, England, 1999

Xuan ZHAO (1983-) received his BS degree in Electronic Engineering from Tsinghua University, China, in 2004. He is currently a postgraduate student at State Key Laboratory of Intelligent Technology and Systems, Department of Electronic Engineering in Tsinghua University.

His reserch interests are computer vision, graphical models and network storage.