A Unified Formulation of Invariant Point Pattern Matching

Tibério S. Caetano, Terry Caelli
National ICT Australia and Australian National University, Canberra ACT 0200, Australia

Abstract

We present a unified framework for modeling and solving invariant point pattern matching problems. Invariant features are encoded as potentials in a probabilistic graphical model. By using a specific kind of graph topology, different types of invariant matching models can be implemented via tree-width selection. Models with tree-widths 1, 2, 3 and 4 implement translation, similarity, affine and projective invariant point matching, respectively. The optimal match is then found by exploiting the Markov structure of the graph through the generalized distributive law in a dynamic programming setting. In the absence of noise in the point coordinates, the solutions found are optimal. Our early experiments suggest the approach is robust to outliers and moderate noise.

1 Introduction

Algorithms for point pattern matching in general may either be explicitly designed to handle a given class of transformations or not. For example, on the one hand there are techniques specifically designed for matching under rigid motions [2], similarity transformations [6] or affine transformations [3]. On the other hand there are structural graph matching techniques, which do not explicitly encode the transformation, but rather abstract point sets as graphs [4] where the graph topologies are used in attempting to preserve structural properties of point sets under "smooth" transformations. We are not aware of the existence of a framework that unifies both formulations to point set graph matching which is also able to explicitly handle different classes of transformations.

The aim of this paper is to show how a single graph matching framework can be used to model explicitly a wide variety of transformations in point pattern matching. We abstract one of the point sets as being nodes of a graph and the other as being a collection of possible labellings for these nodes. Nodes of the graph are seen as random variables and the labels are seen as possible realizations. We then create a probability density function on the set of possible matchings, and aim to find its mode (MAP estimate), what in this formulation corresponds to the most likely assignment. The key aspect in our formulation is that we are able to design densities (i) whose MAP estimates can be found in polynomial time and (ii) which are capable of encoding invariances w.r.t. a range of possible transformations. This is achieved by noting that a certain type of graph is adequate not only for the encoding of high order features but also for exact inference via dynamic programming.

2 The Model

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. Let X_i be a random variable indexed by the i^{th} vertex of the graph, x_i a generic realization of X_i, X the collection of N discrete random variables $X = \{X_1, \cdots, X_N\}$ corresponding to the N vertices and x the joint realization of X, $x = \{x_1, \cdots, x_N\}$. Then, an *undirected probabilistic graphical model*, or Markov random field (MRF), is the family of densities $p(x)$ whose conditional independency[1] assumptions among the variables X_i are those implied by the graph G.

We model the first point pattern as an undirected probabilistic graphical model, where each point is a random variable corresponding to a node in the graph. We model the second point pattern as the set of possible realizations for each random variable. As a result, a joint realization of all the random variables becomes a match between the first and the second point patterns. The optimal solution is then the map that maximizes the joint probability distribution, i.e. the Maximum a Posteriori (MAP) estimate of the random variables.

According to the Hammersley-Clifford theorem [5], the joint probability distribution of the resulting MRF will be factorized over the maximal cliques[2] of the graph:

$$p(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(x_c), \tag{1}$$

[1] Two sets of random variables, A and B, are conditionally independent on a third set, C, if $P(AB|C) = P(A|C)P(B|C)$. In the graph, this is read as: there is no path from A to B that doesn't pass through C.

[2] A clique is a fully connected subgraph and a maximal clique is a clique which is not a proper subset of another clique.

0-7695-2521-0/06/$20.00 (c) 2006 IEEE

which can be written as

$$p(x) = \frac{1}{Z} \exp[-U(x)], \tag{2}$$

where $U(x) = \sum_{c \in \mathcal{C}} U_c(x_c)$, with $U_c(x_c) = -\log \psi_c(x_c)$. $\mathcal{C}$ is the set of maximal cliques and c is a particular maximal clique. Since maximizing $p(x)$ (finding the MAP solution) is the same as minimizing the cost or energy $U(x)$, this connection is useful in order to reveal which type of MRF is associated with some cost function.

Our approach here consists in showing how a proper choice of energy functions can lead to a MRF which can be solved in polynomial time but nevertheless represents faithfully the problem of point matching under invariant transformations.

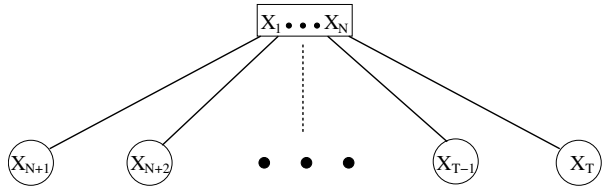

Figure 1. The general topology of the proposed models. By varying N from 1 to 4 we implement translation, similarity, affine and projective point matching.

Figure 1 shows the general form of the Graphical Model that we propose. The "box" in the upper layer is a compact representation for a set of N nodes that are *fully connected*. Each node in the bottom layer is then connected to (an only to) each of these upper nodes. Note that this induces a graph with $T - N$ maximal cliques, one for each node in the bottom layer. According to the Hammersley-Clifford theorem, the joint probability distribution over this graph can be written as

$$p(x) = \frac{1}{Z} \prod_{i=1}^{T-N} \psi_i(x_1, \ldots, x_N, x_{N+i}), \tag{3}$$

so its maximizer is the minimizer of the energy function

$$U(x) = \sum_{i=1}^{T-N} U_i(x_1, \ldots, x_N, x_{N+i}), \tag{4}$$

where $U_i(\cdot) = -\log \psi_i(\cdot)$. Each term of the resulting energy function has $N + 1$ arguments. The *key* thing to observe here is that there are always N arguments common to all terms U_i:$x_1, \ldots, x_N$.

The basic idea that we now introduce is the following: we will design the cost function U_i of the i^{th} maximal clique in such a way that it is minimized when all the "sub" cost functions defined over the $N + 1$ subsets of order N in this clique "agree". The exact meaning of "agreement" will be defined in the next section, but for now we can say that the joint realization of each of these $N + 1$ subsets in a clique will specify a certain *geometric transformation*, and "agreement" will mean the equality of all these transformations within a single maximal clique.

The reason for doing so is simple to understand: since one of the $N + 1$ subcliques of size N in every maximal clique is actually *common* to all maximal cliques (namely, the subclique $x_1, \ldots, x_N$), the equality of subclique transformations in *every* maximal clique will imply the equality of transformations across the whole network, what will guarantee that a single final transformation is found. We can then say that, in this model, "local agreement" (agreement within a clique) implies "global agreement" (agreement across the whole network). Our model is a *chordal* (or *triangulated*) graph with a *tree-width* equal to N—the number of nodes in the upper layer. In the following we will see how, by varying the tree-width of the graph, we can implement different types of point set matching invariant to different types of transformations.

3 Invariances

Here we show in detail how the energy functions U_i must be designed so that different types of invariant transformations can be handled in the point matching problem.

3.1 Translation Invariance

In the plane, a single point correspondence uniquely determines a translation. That is easy to see, since the translation model for a single correspondence $(y, z) \mapsto (y', z')$ is

$$\begin{pmatrix} y' \\ z' \end{pmatrix} = \begin{pmatrix} y \\ z \end{pmatrix} + \begin{pmatrix} \overline{y} \\ \overline{z} \end{pmatrix}, \tag{5}$$

what gives two equations and two unknowns, then specifying uniquely the transformation vector $(\overline{y}\ \overline{z})^T$.

Now, consider Figure 1 for $N = 1$. In this case the graph is simply a tree with node X_1 connected to all other nodes in a "fan" manner, and the maximal cliques of the graph are the pairs $X_1 X_i$ for $2 \leq i \leq T$. Recall that each node in the model corresponds to a point in the template and each possible realization for a node corresponds to a point in the scene. As a consequence, each single realization of a node is a correspondence between a point in the template and another in the scene. Given these facts, we notice that a particular realization of X_1 uniquely specifies a translation. The same happens for every other X_i. Now, the only thing we need to do is to specify cost functions $U_i(x_1, x_i)$ that are minimized when (and only when) the transformations

defined by $(y_1, z_1) \mapsto (y'_1, z'_1)$ and $(y_i, z_i) \mapsto (y'_i, z'_i)$ are *the same*. In other words, we need a measure of discrepancy between the vectors $(\overline{y}_1\ \overline{z}_1)^T$ and $(\overline{y}_i\ \overline{z}_i)^T$. If this discrepancy is zero, it follows that both transformations "agree". The point here is that, since this is applied to every maximal clique *and* there is a common variable to all maximal cliques (X_1), the transformations that minimize every one of the U_i's will be actually one and the same.

3.2 Similarity Invariance

Exactly the same framework can be applied to cope with similarity transformations by setting $N = 2$ in the model of Figure 1. In this case, the maximal cliques are $X_1X_2X_i$, and we need to check the agreement of transformations defined by the pairs X_1X_2, X_1X_i and X_2X_i. A similarity transformation is uniquely determined given *two* point correspondences, as can be easily seen from its defining equations,

$$\begin{pmatrix} y' \\ z' \end{pmatrix} = \alpha \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} + \begin{pmatrix} \overline{y} \\ \overline{z} \end{pmatrix}, \quad (6)$$

which involves a vector of 4 parameters: $(\alpha\ \theta\ \overline{y}\ \overline{z})^T$. Since there are only two equations per point correspondence, we need two point correspondences to uniquely specify such a transformation. In this way, the pairs X_1X_2, X_1X_i and X_2X_i will define each a valid similarity transformation. Our task is then to design U_i such that it is minimized if the 3 transformations are the same. Since X_1X_2 is common to all maximal cliques, the similarity transformation that maximizes the agreement in every maximal clique will be necessarily the same.

3.3 Affine Invariance

An affine invariant point matching model is obtained if we set $N = 3$ in Figure 1. In the case of an affine transformation in the plane,

$$\begin{pmatrix} y' \\ z' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} + \begin{pmatrix} \overline{y} \\ \overline{z} \end{pmatrix}, \quad (7)$$

there are 6 parameters and 2 equations. In order to uniquely specify such a transformation, we then need 3 point correspondences. Each maximal clique in the graph has size 4 in this case, and contains 4 possible subcliques of size 3. One of these subcliques ($X_1X_2X_3$) is common to all maximal cliques. The associated energy U_i for maximal clique i should then be an agreement measure over the 4 possible affine transformations defined by the 4 subcliques of size 3. In an analogous manner to the previous cases, transitivity is established by subclique $X_1X_2X_3$ and the transformations that minimize every U_i are guaranteed to be equal.

It is easy to verify that, by defining $N = 4$, one obtains a projective-invariant point matching model, since 4 correspondences are required to uniquely determine a projective transformation in the plane.

4 Inference

So far we have seen how to create the energy functions over the maximal cliques of the graph, but not how to optimize the entire energy function, $U(x) = \sum_i U_i(x_1, \ldots, x_N, x_i)$. This is equivalent to maximizing $p(x)$ (finding its mode, or the MAP solution). This can be done in a systematic way by taking advantage of the Markov properties of the graph through the use of the generalized distributive law in a dynamic programming scheme [1]. Let's assume we want to find the optimal assignment for one of the X_i's in the bottom layer of the model. From the Hammersley-Clifford factorization of Eq. 3, max-marginalization over the other variables becomes

$$p(x_j) \quad = \quad \max_{\widetilde{x_j}} \frac{1}{Z} \prod_{i=1}^{T-N} \psi_i(x_1, \ldots, x_N, x_{N+i}), \quad (8)$$

where $\max_{\widetilde{x_j}}$ means "maximize over variables other than x_j". In the following we take the argmax and use the distributive law to push the maximizations as far as they can go into the products ($\square$ denotes the variables in the "box" of Figure 1: $x_1, \ldots, x_N$):

$$\begin{aligned} \arg\max_{x_j} p(x_j) &= \arg\max_{x_j} \max_{\widetilde{x_j}} \prod_{i=1}^{T-N} \psi_i(\square, x_{N+i}) \\ &= \arg\max_{x_j} \max_{\square} \max_{x_{N+1}} \psi_1(\square, x_{N+1}) \ldots \\ &\ldots \max_{x_{j-1}} \psi_{j-1-N}(\square, x_{j-1}) \max_{x_{j+1}} \psi_{j+1-N}(\square, x_{j+1}) \ldots \\ &\ldots \max_{x_T} \psi_{T-N}(\square, x_T). \end{aligned} \quad (9)$$

Note that the last maximization, $\max_{x_T} \psi_{T-N}(\square, x_T)$, will be only a function of $\square$, so it can be pulled out from the inner maximizations and be put in from of the outer maximization $\max_{\square}$. But this can then be done recursively from x_T downwards to x_{N+1}, and we obtain

$$\arg\max_{x_j} p(x_j) = \arg\max_{x_j} \max_{\square} \prod_{i=1:T, i\neq j} m_i(\square). \quad (10)$$

where the factors $m_i(\square)$ can be seen as "messages" coming from the leaves to the box (see Figure 1). Since the box involves just a few variables, this brute force maximization can be performed efficiently. This is done for every X_i—by reusing messages—, and the mode of the resulting marginal

distribution is then set as being the correct assignment for X_i. The overall computational complexity of this algorithm is $O(TS^{N+1})$ (S is the number of possible outcomes for X_i—the number of points in the scene). The algorithm thus gets more expensive as we move from translation to projective invariance—but it's always of polynomial complexity on the number of points.

5 Experiments

We present some early experiments with both synthetic and real data. In both, the energy U_i of a clique was set as being the sum of pairwise Euclidean distances between the parameter vectors of all the $N+1$ transformations induced in the clique of size $N+1$. For the synthetic experiments, we generated 100 sets of ∼30 points and applied random transformations in order to generate the second point set. This was done for translation, similarity and affine transformations. We also introduced outliers (10) in the neighborhood of the second point set, an noise in the point coordinates. Figure 2 shows a typical result of our algorithm, for affine invariance. The overall performance is shown in Table 1, where percentages are over the total of 3000 individual matches. Note that the algorithm has perfect performance for zero noise (in particular it is insensitive to the presence of outliers). Note also that as noise increases the performance of transformations with more degrees of freedom tend to be poorer (see the axes in Fig. 2 for a reference to the σ values).

	Noise level (σ of Gaussian noise)				
	0	0.01	0.02	0.03	0.04
Translation	100%	96%	92%	87%	81%
Similarity	100%	93%	86%	80%	70%
Affine	100%	89%	81%	72%	61%

Table 1. Fraction of correct correspondences for different noise levels and different transformation assumptions.

For the experiments with real image data, we used the CMU house database, which is available at http://vasc.ri.cmu.edu/idb/html/motion/house/. It contains 111 frames of a video sequence of a toy house. We extracted landmark points (∼30) from each image and matched pairs separated by 10, 20, 30, 40 and 50 frames (thus forming 101, 91, ..., 61 image pais), under assumptions of translation, similarity and affine invariance. Table 2 summarises the results obtained, where percentages are over all 101×30, ..., 61×30 individual matches. It is clear that for large baseline the affine assumption is more appropriate, whereas for low baselines similarity and translation become adequate, as expected.

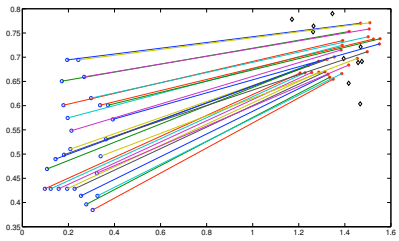

Figure 2. A pattern correctly matched to affine-transformed version (note outliers).

	Baseline				
	10	20	30	40	50
Translation	97%	88%	77%	67%	52%
Similarity	97%	92%	87%	79%	65%
Affine	95%	93%	90%	84%	77%

Table 2. Fraction of correct correspondences for different baselines between frames and different transformation assumptions.

6 Conclusion

This paper introduced a unified approach for solving invariant point pattern matching problems. The matching problem is formulated as one of finding the optimal labeling in a probabilistic graphical model. By changing the tree-width of the underlying graph, we obtain models which are able to solve point matching under a number of different invariance assumptions. Inference is performed exactly by dynamic programming. In the absence of noise the results are perfect, and our early experiments suggest the algorithm is robust to outliers as well as to moderate noise.

References

[1] S. M. Aji and R. J. McEliece. The generalized distributive law. *IEEE Trans. Inform. Theory*, 46(2):325–343, 2000.

[2] T. S. Caetano, T. Caelli, D. Schuurmans, and D. A. C. Barone. Graphical models and point pattern matching. *IEEE Trans. on PAMI*, 2006. [to appear].

[3] M. Costa, R. Haralick, and L. Shapiro. Optimal affine-invariant point matching. In *10th International Conference on Pattern Recognition*, volume 1, pages 233 – 236, 1990.

[4] E. Hancock and R. C. Wilson. Graph-based methods for vision: A yorkist manifesto. *SSPR & SPR 2002, LNCS*, 2396:31–46, 2002.

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

[6] S. Z. Li. Matching: Invariant to translations, rotations and scale changes. *Pattern Recognition*, 25(6):583–594, 1992.

0-7695-2521-0/06/$20.00 (c) 2006 IEEE