Structural Matching Via Optimal Basis Graphs

Fred W. DePiero
CalPoly State University, San Luis Obispo, CA, USA, fdepiero@calpoly.edu
John K. Carlin
Agilent Technologies, Colorado Springs, CO, USA, john_carlin@agilent.com

Abstract

The 'basis graph' approach to structural matching uses a fixed set of small (4 node) graphs to characterize local structure. We compute mapping probabilities by first finding the probability of a basis graph being an induced subgraph of the input graph. The similarity of these probabilities is used to compare nodes of the input graphs. The method permits common subgraphs to be identified without the use of any node or edge coloring. We report on an improved, simpler, version of the algorithm, which has also been optimized. Performance is compared with the LeRP method, which is based on length-r paths. Both methods are approximate with polynomial bounds on both memory and on the worst-case compute effort. These methods work on arbitrary types of undirected graphs, and tests with strongly regular graphs are included. Monte Carlo test trials (3000+) included up to 100% additional (noise) nodes.

1. Introduction

In this paper we address the problem of finding the maximum common subgraph via methods appropriate for real-time measurement systems. Our approach has polynomial bounds on memory and on worst-case compute effort. Graph matching is accomplished solely via comparisons of structure; without node or edge attributes. No assumptions on graph structure (planar, for example) are made herein. Our methods do ensure a one-to-one mapping between nodes in the two input graphs, and ensure the resultant common subgraph is a valid subgraph. However the method is approximate (inexact), so no guarantee of a maximum number of common nodes is asserted.

We target a method for graph matching with broad applicability. Of particular interest are real-time applications where an approximation to the maximal common subgraph is acceptable, provided it can be found deterministically. For example in range data registration, having fewer nodes than the maximum common subgraph is tolerable, but lengthy computations are not [4]. Use of graph matching in this application permits the steps of determining correspondence and pose to be separated and computed in a deterministic fashion.

In general, noise in sensor data impairs the ability to compare graph attributes. This effectively lowers the dynamic range of coloring. Our method is optimized to handle little or no coloring, and hence can perform well in applications with noisy sensor data. In this paper we restrict coloring to integer values with a range of 0 (no coloring) and 2 (two distinct colors). Also to demonstrate the capabilities of our technique we include test trials with strongly regular input graphs (a challenging style [16]).

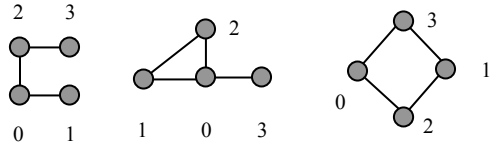

Figure 1. Optimal 4-node basis graphs used to characterize the local structure of input graphs. Note the varied structure (loops of length 0, 3 and 4).

Established methods for graph matching may be categorized as either exact or approximate. As the problem of finding a maximum common subgraph is know to be NP-complete, exact methods inevitably have an exponential worst-case compute effort. Recently published approximate methods include [10] [12] [15] [9] [7] [19-22]. The technique in [10] by Messmer, for example, is optimized for large databases of objects that may contain similar subgraph structures. Subgraphs that are common among a suite of known input graphs are identified in a preprocessing step to create a library. These commonly occurring subgraphs are then identified in input data graphs. This method is efficient during recognition, but does require preprocessing time to build the recognition library. It also uses attributed graphs.

The 'basis graph' ('BG') approach [17] is approximate and is compared to another such method 'LeRP', which is based on length-r paths [5]. Basis graphs are small graphs (4 nodes) that are used to describe local structure in an input graph. See Figure 1.

2. Comparing Graph Structure Dynamically

A feature that distinguishes the BG and LeRP methods from other techniques has to do with the size of the neighborhood used to compare local graph structure. We refer to the size of this neighborhood as the 'horizon'. Approaches employing larger horizons may be able to describe local structure better. This is particularly so when comparing neighborhoods that are structurally consistent. In our techniques the size of the neighborhood varies dynamically – the more similar the structure, the larger the neighborhood.

While a larger horizon may be preferable in general, global constraints are still needed to reliably approximate the maximum common subgraph. This may be accomplished in a variety of ways, for example by making soft assignments and iterating [7], by relaxation [2][8][15], via MAP probabilities and hill climbing [3], or via MAP & EM [9]. We use a relaxation method.

In this paper we seek to find an optimum set of basis graphs capable of describing structure for a wide range of input graphs (strongly regular, banded, with and without coloring, etc.). Our use of a set of small graphs is similar in concept to Messmer [10]. However, here we are striving to find an optimum basis useful for any input graphs; rather than a basis optimized for a given set of inputs.

The degree of variation in structure of our basis graphs, of Wilson & Hancock's supercliques [15] and of Messmer's dictionary is another possible taxonomy to compare approaches. In our studies, a more varied structure performs better. (See Figure 1).

3. Basis Graph Algorithm

Basis graphs are used to describe local structure. We approximate a probability density function $p_1[ni][nx][b]$ which describes how likely a subgraph of an input graph G_1, will match the structure of a particular basis graph. The PDF, p_1, gives the probability of having matching (local) structure when the basis graph has its first node associated with node n_i of G_1, and its b^{th} node associated with n_x (of G_1). Similarly, we find $p_2[n_j][n_y][b]$ for the other input graph G_2. Estimates of a mapping probability $P[n_i][n_j]$ are determined by finding the closest matching pairs of p_1 and p_2 values. An a-priori model related the p_1-p_2 differences to the probability of n_i being mapped to n_j in the common subgraph. A-priori probabilities of the best-case matching pairs of p_1, p_2 are combined via Dempster-Shafer [19] to estimate $P[n_i][n_j]$. A fixed number of relaxation iterations are then used to refine $P[n_i][n_j]$ and to determine the final mapping.

More specifically, we estimate $p_1[n_i][n_x][b]$ via a histogram $h_1[n_i][n_x][b]$. The histogram h_1 counts the number of occurrences of a selected basis graph, B (V nodes), exactly matching a subgraph of the input G_1. All permutations of k nodes of G_1 are examined. The k nodes of B and of G_1 are compared via their adjacency matrices, which must match exactly for h_1 to be incremented. We vary k from 2 to V. When a match occurs with the first node of the basis associated with node n_i of G_1 and node b of the basis associated with node n_x of G_1, then we increment $h_1[n_i][n_x][*]$. The PDF estimate, p_1, is found by normalizing h_1, and p_2 is found similarly. The p_1, p_2 PDFs describe local structure only and could be computed in parallel.

Next we determine initial estimates of the mapping probabilities $P[n_i][n_j]$. For each pair of nodes n_i, n_j in G_1, G_2, respectively, we search for $p_1[n_i][*][*]$ and $p_2[n_j][*][*]$ values which are similar. For n_i, n_j, we find

$$\mathbf{MIN}\{\, p_1[n_i][n_x][b] \;-\; p_2[n_j][n_y][b] \,\}$$

by searching over all n_x, n_y, b entries. We require p_1 and p_2 values to be nonzero. At most one selection is made for each n_x, n_y value. We define

$$p_w = \mathbf{N_b}(\, p_1[n_i][n_x][b] - p_2[n_j][n_y][b] \,)$$

as the w^{th} minimum difference found for a particular value of b, where $\mathbf{N_b}$ () is a probability computed via an a-priori Gaussian model. The mapping probability

$$P[n_i][n_j] = \mathbf{DS}\{\, p_w \,\} \text{ for all w,}$$

where **DS**{} is the Dempster-Schafer rule to combine evidence [19]. A separate Gaussian model $\mathbf{N_b}()$ is prepared for each node, b, of each basis graph.

The a-priori models describe the likelihood of a given p_1-p_2 difference occurring for nodes n_i, n_j that should be associated in the common subgraph. These models describe the variations under given structural noise conditions.

The matching algorithm may readily be expanded to include comparisons of graph color or other attributes. These restrict potential matches, improving performance in terms of both speed and the size of the common subgraph. Compared to [17], this version of the BG algorithm is simpler, includes the a-priori PDFs, and an optimal choice of basis graphs. FYI, we consider the (dynamic) neighborhood for node n_i of G_1 to consist of all nodes n_x, where $p_1[n_i][n_x][*]$ is non zero.

4. Finding the Optimal Set of Basis Graphs

We seek results that are as broadly applicable as possible. Hence we select a relatively wide range of conditions for our input graphs during the optimization process. We generate four different styles of random graphs: Model A [13], strongly regular, and 'banded' with and without color. Parameters associated with these styles of inputs are also varied, as is the structural noise level. Generation via Model A is analogous to flipping a weighted coin to determine the existence of an edge. Edge probabilities varied (0.2, 0.3). Strongly regular graphs

were included to provide a challenging test case [16] and the target degree varied (4, 6). Banded graphs have a band limited adjacency matrix [17] [5]. Banded graphs are a useful approximation to both natural and man-made structures (e.g. chemical molecules and VLSI circuits). This is a minimum bandwidth over all node orderings. Input graphs had 16 or 32 nodes, nominally. Noise levels varied from 0%, 50% and 100% additional nodes. Noise edges were randomly added A limited amount of coloring was also introduced into the optimization, to help broaden the applicability of our results.

When the BG algorithm operates with multiple basis graphs, it simply uses each one in turn and selects the final result that is the largest common subgraph. Hence examining various combinations of individual basis graphs provides an assessment for the performance of a team. We run Monte Carlo test trials under varying conditions for individual basis graphs, store the results, and then examine team membership.

In finding an optimal basis set, we examined basis graphs with 4 nodes. (Chosen via data from prior work [17], based on an accuracy versus speed tradeoff). With these small basis graphs, the optimum set ('team') could be assessed via enumeration. We examined teams of size T, 1 to 5. Larger teams perform better of course. Finally, we selected a team size by identifying a point of diminishing returns. See Table 1.

Table 1. Performance for varying team sizes.

Mean # Nodes	88%	92%	94%	95%	96%
Team Size, T	1	2	3	4	5

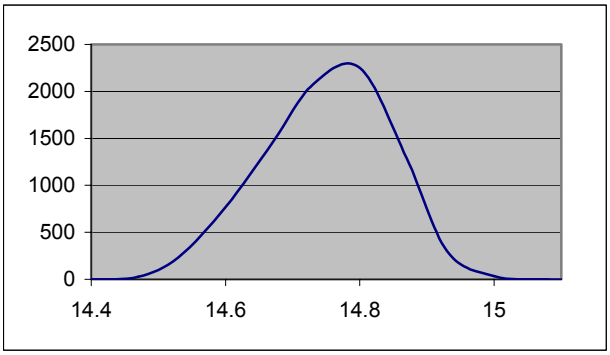

Figure 2. Histogram of number of teams versus mean size of common subgraph. Histogram includes all possible teams with three 4-node basis graphs.

Figure 2 shows a typical histogram of team performance. This gives the number of teams vs. the mean number of nodes in the common subgraph (varying from 14.5 to 15.1). Teams had 3 basis graph members.

Results from Figure 2 indicate that there are many possible teams that yield near optimal performance. We feel the stability of our choice for an optimum basis set is an important consideration for reporting a more general result. To best identify an optimal team (or team characteristic) we selected basis graphs that often appeared in the better-performing teams (with varying size). The three most commonly appearing are shown in Figure 1. We intend this approach to improve stability for our selection, rather than reporting the very best team.

Table 1 shows the performance of teams of basis graphs, of varying size (T). The 'mean # nodes' refers to the average number of nodes in extracted common subgraphs, relative to the number of nominal nodes in the input graph before adding noise nodes. Based on these results, we select teams with T=3 basis graphs, which appears to be the point of diminishing returns.

Note the optimal basis graphs shown in Figure 1 possess fundamentally different structures, in terms of the number and size of loops. From an intuitive standpoint, we believe this variation in structure is important for achieving better performance

Table 2. Performance benchmarks. Inputs with 'Color-2' had integer-valued colors for node & edge attributes (dynamic range=2). Inputs 'N=32' had 32 nodes

Type of Input	**Noise**	**BG**	**LeRP**
Model A (0.2)	0%	99 ± 4 %	100 ± 2 %
Model A (0.3)	0%	100 ± 0 %	100 ± 0 %
Model A (0.2)	50%	85 ± 16 %	63 ± 11 %
Model A (0.3)	50%	94 ± 13 %	55 ± 9 %
Model A (0.2)	100%	77 ± 12 %	63 ± 9 %
Model A (0.3)	100%	75 ± 14 %	56 ± 6 %
Regular (3)	0%	100 ± 1 %	89 ± 13 %
Regular (4)	0%	100 ± 0 %	99 ± 4 %
Regular (3)	50%	78 ± 12 %	68 ± 8 %
Regular (4)	50%	86 ± 15 %	61 ± 8 %
Regular (3)	100%	74 ± 8 %	69 ± 8 %
Regular (4)	100%	75 ± 10 %	64 ± 6 %
Banded (4)	0%	99 ± 3 %	99 ± 4 %
Banded (6)	0%	100 ± 1 %	99 ± 6 %
Banded (4)	50%	96 ± 8 %	94 ± 9 %
Banded (6)	50%	95 ± 9 %	90 ± 11 %
Banded (4)	100%	96 ± 7 %	95 ± 8 %
Banded (6)	100%	93 ± 10 %	88 ± 13 %
Banded (4) Color-2	0%	100 ± 0 %	99 ± 4 %
Banded (6) Color-2	0%	100 ± 0 %	97 ± 6 %
Banded (4) Color-2	50%	100 ± 0 %	94 ± 8 %
Banded (6) Color-2	50%	100 ± 0 %	86 ± 12 %
Banded (4) Color-2	100%	100 ± 0 %	95 ± 8 %
Banded (6) Color-2	100%	100 ± 0 %	91 ± 11 %
Banded (4) (N 32)	0%	96 ± 10 %	97 ± 4 %
Banded (6) (N 32)	0%	93 ± 13 %	98 ± 3 %
Banded (4) (N 32)	50%	98 ± 5 %	96 ± 7 %
Banded (6) (N 32)	50%	92 ± 13 %	95 ± 5 %
Banded (4) (N 32)	100%	91 ± 11 %	97 ± 4 %
Banded (6) (N 32)	100%	90 ± 14 %	95 ± 5 %

5. Test Results Using the Optimal Basis

Monte Carlo-style testing was used to benchmark performance in terms of the mean size of the common subgraph. A test trial began by generating graphs G_1 and G_2 identically, randomizing node order, and then randomly adding nodes and edges to the G_2 input. Although input graphs were randomly generated, the trials employed were restricted to cases of connected graphs.

The mean size of the common subgraph is reported in Table 2. Tests included graphs generated via Model A (edge probability 0.2, 0.3) and strongly regular graphs (degree 3, 4). The size the initial graph was fixed at 16 nodes, except as noted (N=32). The number of noise nodes added to G_2 varied: 0%, 50% and 100% of the initial size. Results show that BG performs better than, or similar to LeRP. LeRP may perform slightly better for larger graphs (it can form larger neighborhoods). Note performance of BG with very modest color is ideal in these trials (3000 reported, total).

6. Compute Effort & Memory Requirements

The compute effort and memory requirements for each stage of the algorithm are given in Table 3. This assumes an N-node input, and a V-node basis. The number of relaxation iterations, R, was fixed for all reported tests. T is the number basis graphs used in teams.

As the compute effort increases with TN^V the size of the basis graph is an important issue. We compared performance with somewhat larger teams with smaller bases vs. smaller teams with larger bases and chose V=4. The parameter V remains fixed in all our reported trials.

Table 3. Order of computational effort and memory.

	Processing Step	Effort	Memory
1	Find p_1, p_2	$O(TN^V)$	$O(VN^2)$
2	Find $P[n_i][n_i]$	$O(TVN^2)$	$O(N^2)$
3	Relaxation	$O(TRN^2)$	$O(VN^2)$

7. Conclusion & On-Going Studies

Our technique for graph matching is deterministic and does not relay on any node or edge attributes (coloring). The basis graph technique incorporates a dynamic comparison horizon, as does LeRP. The dynamic nature of the neighborhood allows more local structure to be included in comparisons of noise-free portions of input graphs, benefiting local comparisons.

We are interested in possibilities for an HDL-based hardware implementation. We are also studying a variation on the described method to estimate the p1 probabilities via random comparisons of the basis graphs versus input graphs. Our software is available [6].

8. References

[1] M. Carcassoni and E.R. Hancock, Correspondence Matching with Modal Clusters, *IEEE Trans. PAMI*, 25 (12) (2003) 1609-1614.

[2] W J Christmas, J Kittler and M Petrou, Probabilistic feature-labelling schemes: modelling compatibility coefficient distributions. *Image and Vision Comp*, 14 (1996) 617-625.

[3] A.D.J. Cross, E.R. Hancock, Graph matching with a dual-step EM algorithm, *IEEE Trans. PAMI*, 20 (11) (1998) 1236.

[4] F. W. DePiero, "Deterministic Surface Registration at 10Hz Based on Landmark Graphs With Prediction," 14th British Machine Vision Conf. (*BMVC2003*), Norwich, UK, Sept, 2003.

[5] F. W. DePiero and D.W. Krout, LeRP: An algorithm using length-r paths to determine subgraph isomorphism, *Pattern Rec Journal,* 24 (1) (2003) 33-46.

[6] F. W. DePiero., "Home Page", Software for Graph Matching, www.ee.calpoly.edu/~fdepiero/ (August, 2004).

[7] S. Gold, A Rangarajan, A graduated assignment algorithm for graph matching, *IEEE Trans. PAMI*, 18 (4) (1996) 377-388.

[8] J. Kittler, E. R. Hancock, Combining Evidence in Probabilistic Relaxation, *Intl. Journal of Pattern Recognition and Artificial Intelligence*, 3 (1989) 29-51.

[9] B. Luo and E.R. Hancock, Structural graph matching using the EM algorithm and singular value decomposition, *IEEE Trans. PAMI*, 23 (10) (2001) 1106-1119.

[10] B.T. Messmer, H. Bunke, A new algorithm for error-tolerant subgraph isomorphism detection, *IEEE Trans. PAMI*, 20 (5) (1998) 493-504.

[11] B. McKay. Practical Graph Isomorphism, *Congressus Numerantium*, 30 (1981) 45-87.

[12] R. Myers, R.C. Wilson, E.R. Hancock, Bayesian graph edit distance, *IEEE Trans. PAMI*, 22 (6) (1997) 628-635.

[13] E. M. Palmer, Graphical Evolution – An Introduction to the Theory of Random Graphs, Wiley-Interscience, 1985.

[14] A. Sanfeliu, K.S. Fu, A distance measure between attributed relational graphs for pattern recognition, *IEEE Trans. Systems, Man and Cybernetics*, 13 (1983) 353-363.

[15] R.C. Wilson, E.R. Hancock, Structural matching by discrete relaxation, *IEEE Trans. PAMI*, 19 (6) (1997) 634-648.

[16] R. C. Read and D. G. Corneil, The graph isomorphism disease, Journal of Graph Theory, 1 (1) 339-363 (1977).

[17] F. W. DePiero, Structural Graph Matching with Polynomial Bounds on Memory and on Worst-Case Effort, (ICPR2004) Cambridge, UK, August 23-26, 2004.

[18] T. Caelli and S. Kosinov, An eigenspace projection clustering method for inexact graph matching, *IEEE Trans. PAMI*, 26 (4) (2004) 515-519.

[19] A. Robles-Kelly and E.R. Hancock, Graph Edit Distance from Spectral Seriation, *IEEE T. PAMI*, 27 (3) (2005) 365-378.

[20] M. Gori, M. Maggini and L. Sarti, Exact and Approximate Graph Matching Using Random Walks, *IEEE T. PAMI*, 27 (7) (2005) 1100-1111.

[21] R.C. Wilson, E.R. Hancock and B. Luo, Pattern Vectors from Algebraic Graph Theory, *IEEE T. PAMI*, 27 (7) (2005) 1112-1124.

[22] L. Cordella, P. Foggia, C. Sansome and M. Vento, A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs, *IEEE T. PAMI*, 26 (10) (2004) 1367-1372.