

# Structural Graph Matching With Polynomial Bounds On Memory and on Worst-Case Effort

Fred W. DePiero

CalPoly State University, San Luis Obispo, CA, USA, fdepiero@calpoly.edu

## Abstract

*A new method of structural graph matching is introduced and compared against an existing method and against the maximum common subgraph. The method is approximate with polynomial bounds on both memory and on the worst-case compute effort. Methods work on arbitrary types of undirected graphs, and tests with strongly regular graphs are included. No node or edge colors are required for matching; the common subgraph is extracted based in structural comparisons only. Monte Carlo test trials included up to 100% additional (noise) nodes that were introduced to both input graphs. Results are possible that recover 100% of the original number of input nodes, and that are within 10% of the number of nodes in the maximum common subgraph. Over 2000 test trials are reported.*

## 1. Introduction

In this paper we address the problem of finding the maximum common subgraph via methods targeting practical, 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.

The reason for setting these goals is to develop a method 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 with real-time range image 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 accomplished in a non-iterative fashion.

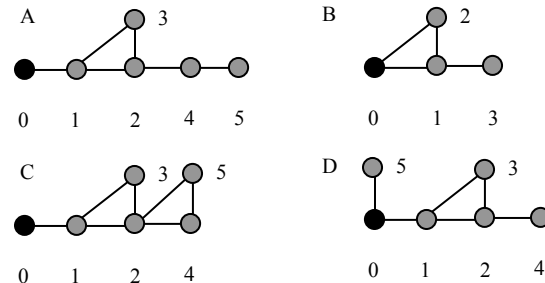
Established methods for graph matching may be categorized as either exact or approximate. As the problem of finding a maximum common subgraph is known to be NP-complete, exact methods inevitably have an exponential worst-case compute effort. Recently published approximate methods include [10] [12] [15] [9]

[7]. The technique in [10] is optimized for large databases of objects that may contain similar subgraph structures. The method is efficient during recognition, but does require preprocessing time to construct a recognition library. It also uses attributed graphs. Most reported methods not only rely on graph attributes but are also iterative, making them less desirable for real-time systems. For example in methods based on relaxation labeling comparisons of node and edge colors are needed to establish an initial guess for the node mapping, before iterations begin [8]. More recent work in this area uses the color comparisons initially and during iterations [2]. Expectation-maximization is another method that has been used recently to iteratively adjust mapping probabilities [9]. Some other methods also have exponential memory requirements [15], which may be problematic in applications.

Earlier work in graph matching included methods that provided exact results, but that required exponential worst-case execution times [14]. Other methods matched whole graphs, but not subgraphs, such as [11].

## 2. Comparing Graph Structure Dynamically

Two approximate methods are compared in this paper, one using ‘Basis Graphs’ (‘BG’, a new approach) and one using the ‘LeRP’ algorithm, which is based on length-r paths [5].



**Figure 1. Basis graphs A-D. Root nodes are darker. The order of nodes used during placement is indicated.**

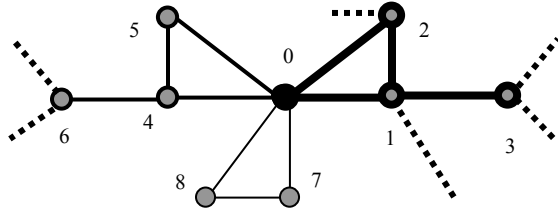
A feature that distinguishes the BG and LeRP methods from other techniques has to do with the size of the neighborhood used to compare local graph structure. In our techniques the size of the neighborhood varies dynamically – the more similar the structure, the larger the neighborhood. We refer to the size of the neighborhood as the ‘horizon’. Hence our techniques have a dynamic horizon.

Wilson and Hancock describe using a ‘superclique’ neighborhood in [15]. This is a good counter example of a method that uses a static horizon. The local neighborhood always consists of a central node and its adjacent nodes. Earlier methods for scene labeling incorporated unary or binary constraints only, thus using static horizons with even lesser extent. Methods that employ a limited horizon for an initial comparison of structure must somehow expand or combine the local measures in order to then approximate the maximum common subgraph. This is accomplished in various ways, for example by making soft assignments and then iterating [7], via MAP probabilities and hill climbing [3], or via MAP & EM [9].

### 3. Approach Using Basis Graphs

Local structural comparisons are computed using basis graphs. Specifically, the basis graphs are employed to form a local neighborhood, which has an invariant node order.

The basis graphs used herein were relatively small (4 to 6 nodes) compared to the graphs being matched that had up to 65 nodes. See Figure 1. Basis graphs have a designated root node and do not contain any structural symmetry (automorphism).



**Figure 2.** Three instances of a basis graph are located relative to a common root node ( $n_i$ , label 0). The order of placement of the basis graphs is indicated by bolder and lighter edges. Resulting node order for the neighborhood is indicated. Additional edges outside the neighborhood are dashed.

To form an invariant L-neighborhood, a basis graph,  $B$ , is rooted at node  $n_i$  and all possible placements within  $G1$  are enumerated from this root position. A histogram  $H1[n_i][n_x][k]$  is incremented when node  $k$  of  $B$  coincides with node  $n_x$  in  $G1$  during the placement operation. After histogramming, non-overlapping instances of the basis graph,  $B$ , are laid on top of  $G1$  in a depth-first manner, rooted at  $n_i$ . The location of the  $k^{\text{th}}$  node of  $B$  corresponds to the largest  $H1[n_i][*][k]$  value. The local ordering for  $L1$  is then given by the order of nodes encountered during the  $B$  overlay operation. See Figure 2. The  $L1$  neighborhood is the induced subgraph associated with the nodes touched during the overlay operation. Instances of  $B$  in  $G1$  may be partial versions, (this can occur due to constraints of the  $G1$  graph structure). Ties in the  $H1$  histogram halt placement. The  $L2$  neighborhoods in  $G2$  are setup in a

similar fashion. Using this histogramming method, basis graphs  $B$  are located in the ‘most common’ location within  $G1$ .

The ordered L-neighborhoods are used to find the probability of node-to-node mappings,  $P(n_i, n_j)$ . This is found via the complement of the edit distance, computed using the adjacency matrix of each  $L1, L2$  neighborhood. Because the nodes are ordered within  $L1$  and  $L2$ , cyclic permutations of the L-neighborhood are not necessary, as with [12]. When two neighborhoods contain a different number of nodes, the adjacency matrix for the smaller one is padded with zeros.

Mapping probabilities are refined via a fixed number of iterations of continuous relaxation. One iteration consists of updating  $P(n_i, n_j)$  for each  $n_i, n_j$  node pair. Starting with  $n_i, n_j$ , a candidate mapping  $M12$  is formed in a best-first fashion. The probabilities of each node pair contained within the  $M12$  mapping is combined via Dempster-Shaffer and the results used to update  $P(n_i, n_j)$ . After a fixed number of relaxation iterations ( $R$ ), then final mapping is found best-first.

Additional notes on BG approach:

- Initial mapping probabilities also adjusted by similarity in degree.
- Multiple basis graphs employed, each in turn. Largest common subgraph reported as output.
- Nodes in common subgraph with zero degree are dropped from output.

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.

### 4. Compute Effort & Memory Requirements

The compute effort and memory requirements for each stage of the algorithm are given in Table 1. This assumes an  $N$ -node input, and a  $V$ -node basis.

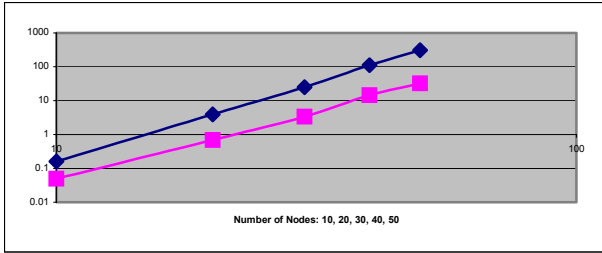
	Processing Step	Effort	Memory
1a	Histogramming	$O(N^V)$	$VN^2$
1b	Placement	$O(N^2)$	$VN^2$
2	Neighborhood Comparisons	$O(N^2)$	$N^2$
3	Mapping	$O(RN^2)$	$VN^2$

**Table 1.** Order of computational effort and memory.

Larger input graphs may necessitate larger basis graphs, however this increase may be mitigated in some applications. For example, with inputs having widespread noise, basis graphs should be kept small to reduce the impact of noise on descriptions of local structure. Also, tests demonstrate the best results for inputs with little or no noise, implying the choice of basis is less critical in

these cases. Hence some situations - having either high or low noise content - do not necessarily require larger basis for satisfactory performance. The set of basis graphs used for testing herein was held constant.

Figure 3 shows the mean compute times for the BG approach. Standard deviations ranged from 4%-6%. When using LeRP the mean compute times were all under 0.1 seconds. Timing was benchmarked on a 1.6GHz PC.



**Figure 3. Duration of processing in seconds for Basis Graph approach. Tests associated with the lower curve included node and edge integer-valued coloring, and used fewer relaxation iterations (5). Integer coloring ranged 0-3. Polynomial compute effort is apparent in the log-log plot, as expected.**

## 5. Testing Method

A Monte Carlo-style analysis was performed to benchmark the final size of the common subgraph. Comparisons with the maximum common subgraph are also reported, computed via the edit distance.

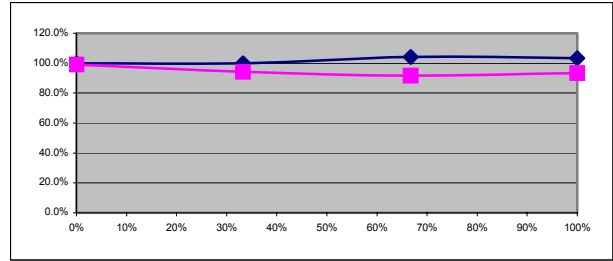
Two different types of random graphs were used for inputs: Model A and strongly regular. Using Model A [13] is analogous to flipping a weighted coin to determine the existence of an edge. The strongly regular graphs were generated by randomly choosing pairs of nodes that each had a degree below a given target value. Strongly regular graphs were used because these are notoriously difficult [16] particularly for techniques that partition nodes by degree [11]. A test trial began by generating graphs G1 and G2 identically, randomizing node order, and then randomly adding nodes and edges to each input graph. The same number of noise nodes was added to G1 and G2, except as noted.

Although input graphs were randomly generated, one restriction was employed: nodes of zero degree were not permitted. Nodes with zero degree were eliminated because their lack of connected structure makes mapping ambiguous.

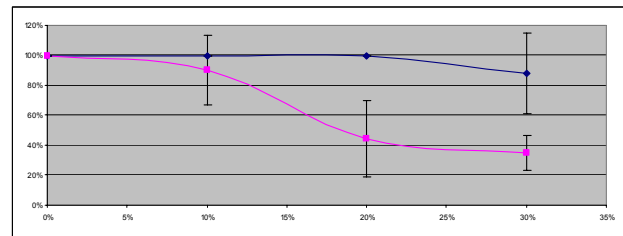
## 6. Testing Results

Figure 4 describes the size of the common subgraph computed. Tests included graphs generated via Model A (edge probability ranging 0.15, 0.2, 0.3) and strongly regular graphs (degree ranging 2, 3, 4). The size the initial graph was fixed at 12 nodes. The number of noise nodes

varied: 0%, 33%, 67% and 100% of the initial size. Noise nodes were added to both input graphs. Figure 4 shows the minimum and maximum value of the mean found when varying the type of input graph (6 styles, described above). Standard deviations were typically 8%-12%, 1200 tests are summarized. Results above 100% of the initial size are possible because noise nodes are added to each input graph.



**Figure 4. Size of common subgraph for BG approach, with varying amounts of noise. Range of the mean number of nodes in output subgraphs is shown, for varying types of input graphs. Results indicate that it is possible to attain a common subgraph that is the same size as the input, with up to 100% added noise. LeRP performed more poorly, maintaining a minimum of ~70% of the nominal size, in these trials.**



**Figure 5. Size of common subgraph for varying amounts of noise. Results of Basis Graph are shown in upper curve & LeRP in the lower curve. Noise added to one graph only, 50 nodes nominal. Model A style input graphs (0.2). The error bars indicate one standard deviation. The BG approach clearly out performs LeRP.**

Tests associated with Figure 5 used Model A (0.2) type graphs with 50 nodes, nominal. Noise was added to one input only. Comparing to results presented in [18], the basis graph approach appears to yield better mapping correspondence.

The edit distance was found via the absolute sum of differences in the adjacency matrix of the output common subgraph versus the maximum common subgraph. (Differences were summed over the upper triangle). All permutations were enumerated to find the proper (lowest) edit distance. These tests were more limited in number (120 trials) as the edit distance was found via exhaustive means. Six different graph types were used (as in the tests

associated with Figure 4. Graphs had 10 nodes nominally, with 100% noise nodes added to one input only.

Comparisons against the maximum common subgraph demonstrate a clear superiority of the basis graph technique versus LeRP. Results may be achieved within 10%, typically, of the number of nodes in the maximum common subgraph.

	BG	LeRP
<b>Edit Distance</b>	$0.6 \pm 1.5$	$4.9 \pm 2.1$
<b>Size Difference</b>	$2\% \pm 7\%$	$25\% \pm 13\%$

**Table 2. Edit distance between computed subgraphs versus the maximum common subgraph. The mean difference between the number of nodes of the maximum common subgraph is also reported. BG is superior to LeRP in these tests.**

## 7. Dissemination of Software

See the author's web page [6] to download. The software is free, for non-commercial, non-profit, non-military, non-defense purposes.

## 8. Conclusion & On-Going Studies

The basis graph technique incorporates a dynamic comparison horizon, as does LeRP. The dynamic horizon is achieved by using varying sizes of basis graphs. Varying bases, result in variable-sized neighborhoods that are used for local comparisons. The dynamic nature of the neighborhood allows more local structure to be included in comparisons of noise-free portions of input graphs. This yields better local comparisons.

Global constraints are introduced into the matching process via a fixed number of iterations of continuous relaxation. Using the BG approach to establish a good estimate for initial probabilities reduces the tendency of convergence to non-global minima. This is desirable for deterministic approaches to estimating the maximum common subgraph.

The tests described herein demonstrate an ability to find a common subgraph, while not requiring any node or edge colors. This is advantageous compared to other methods that have decreased performance with a reduced dynamic range of coloring.

In future studies we are interested in formally estimating the a-priori probabilities of a node-to-node mapping, given the complement of the edit distance comparing two local neighborhoods. The a-priori PDFs relating structure to mapping probability may vary with the general style of input graph (Model A...) and on the style of input noise (widespread or concentrated). Properties of an optimum set of basis graphs are also under study.

We have interest in pattern matching applications with graphs that include a probabilistic description of structure.

These probabilities describe how likely a given node and edge is present, and would be incorporated into the matching process. We are interested in estimating these probabilities via a clustering analysis, made possible with matching techniques such as BG.

As suggested by a conference reviewer, it may be possible to improve efficiency by using a method similar to Tarjan [17] to reduce effort while histogramming.

The author would like to thank and acknowledge reviews and consultations with John K. Carlin and Leonard D. Myers.

## 9. 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/](http://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] R. Tarjan, et.al., Time Bounds for Selection, CS Dept., Stanford, Tech. Report STAN-CS-73-349 (1973).
- [18] T. Caelli and S. Kosinov, An eigenspace projection clustering method for inexact graph matching, *IEEE Trans. PAMI*, 26 (4) (2004) 515-519.