Graph Matching Using
a Direct Classification of Node Attendance *

Fred DePiero and Mohan Trivedi
Computer Vision and Robotics Research Laboratory
Electrical and Computer Engineering Department
The University of Tennessee, Knoxville, TN 37996-2100
depiero@falcon.engr.utk.edu
trivedi@falcon.engr.utk.edu

Steve Serbin
Mathematics Department
University of Tennessee, Knoxville, TN 37996-2100
serbin@sugarbowl.engr.utk.edu

Abstract

An algorithm has been developed that finds isomorphisms between both graphs and subgraphs. The development is introduced in the object recognition problem domain. The method isolates matching subgraphs, finds a node-to-node mapping and reorders nodes thus permitting a direct comparison to be made between the resultant graphs. The algorithm is of polynomial order. It yields approximate results, maintaining a performance level for subgraph isomorphisms at or above 95% under a wide variety of conditions and with varying levels of noise. The performance on the full size comparisons associated with graph isomorphisms has been found to be 100/100, also under a variety of conditions. Performance metrics, methods of testing and results are presented.

KEYWORDS: Direct Classification, Graph Isomorphism, Subgraph Isomorphism, Graph Matching, Object Recognition.

1 Introduction

Object recognition is fundamentally a problem of subgraph isomorphism in that a model describes objects in their entirety; in contrast to an observed object where all features are not typically seen in a single view [1]. Current techniques using range data for shape-based recognition of cluttered scenes are typically quite time consuming and can have undesirable tradeoffs between speed and accuracy.

[0]*This research is sponsored in part by a grant awarded by the Office of Technology Development, U.S. Department of Energy as part of the Environmental Restoration and Waste Management Program.

Techniques are also sometimes tailored for the bin picking problem which focuses on a cluttered scene involving just a few different kinds of objects [2]. We pursue active vision research [3] topics in the Computer Vision and Robotics Research Laboratory (CVRR), as well as real time ranging techniques involving Structured Light [4] and STG Stereo [5], for example. Given these interests, we desire a rapid recognition technique to help facilitate these efforts. Approximate results are acceptable from this perspective as a tradeoff for speed. A more rapid technique will also aide in the processing of larger databases of known objects.

Our approach is to evaluate evidence describing the likelihood of a node's predicted attendance in another graph. The evidence is based on measures that are local to each node. A global verification step completes the process. In this way, the algorithm performs a direct classification of node attendance (DCA). The presence of a node in the other graph is viewed in an isomorphic sense, i.e., there is some node-to-node mapping under which the matched nodes appear as identical members of their graphs. A node's attendance in the other graph is rated on a scale 0.0 to 1.0. The evidence that describes each node characterizes its local structural properties and any node and edge properties that a graph may possess. After the best matching pairs of nodes are identified, the nodes of one graph are reordered to allow a side by side verification of the graphs' similarity.

Attention has been focused on testing DCA under challenging, realistic conditions. Extensive experiments have been run on artificial data sets that were generated with a relatively low distinction in the local character of each node. The simulation tool developed to test DCA serves as a means to carefully investigate performance under varying conditions.

1.1 Goals for DCA Algorithm Development

The goals for finding subisomorphisms were driven by the object recognition application. First and foremost, it was desired to have an efficient algorithm. Results which are approximate - either in terms of accuracy or the size of extracted matches - were acceptable, rather than more lengthy and potentially more complete analyses. This is consistent with trends in active vision [6][7]. We prefer a rapid examination of a scene followed by active exploration. Exploration can yield more information about a scene and can result in the availability of new viewpoints for observations. Achieving a rapid analysis of a scene is commensurate with this goal of active exploration.

In object recognition it is also very helpful to provide more than just a simple yes or no answer to the isomorphism question. A node-to-node mapping between the scene and database graphs is a required final result. The mapping allows the adjacency matrices and other properties of the two graphs to be compared directly.

Some object recognition algorithms depend on node and edge properties having a very "information rich" character. Requiring close matches based on these "rich" properties greatly reduces the number of mappings that must be considered. Too great a reliance on such characterizations can create problems due to occlusions and other noise sources. Our goal is to find an algorithm that relies on the dynamic range of these properties as little as possible. This will provide an increased noise tolerance and broader applicability.

1.2 Review of Related Studies in Machine Vision Community

Because of the importance of object recognition in machine vision, many types of techniques have been pursued [8] [9]. Some researchers confront the recognition problem directly from a standpoint of graph isomorphisms [10] [11] [12] although this approach is less common than other methods. These other techniques can be divided into three categories: maximal clique-based, relaxation labeling and tree search-based approaches [1] [13]. We begin with a discussion of maximal clique-based techniques.

1.2.1 Maximal Clique-Based Approaches

This type of technique builds an association graph which describes all possible compatible mappings between two graphs [14] [15]. Compatibility can be based on node and edge properties and can include a variety of geometrical and topological relationships. The association graph is then searched for a maximal clique [16]. This clique represents the largest possible compatible mapping between the two graphs. The general problem of finding a maximal clique is NP-complete [17]. This can result in extremely long analysis times as the problem size increases. Exact methods exist for finding maximal cliques that are recursive and use depth-first search [18]. Here a tree that can lead to all possible cliques is searched. Pruning occurs when a branch in the search is found that cannot lead to a clique.

More approximate methods for finding maximal cliques have also been explored, via a Markov Random Field (MRF), for example [19]. In this particular work, the low level acquisition and processing of the sensor were to form likelihoods of observing a given object feature. These likelihoods are used as a compatibility measure in the association graph. The MRF determines memberships in cliques. The MRF operates on each edge on the association graph by either including or excluding it from a clique. Mutual consistency of neighboring clique entries guides the MRF convergence.

The association graph itself can be troublesome in these approaches. Given the graphs G^1,G^2 with N^1,N^2 nodes, respectively, the association graph A_g contains a node for each compatible pair of nodes in G^1 and G^2. This necessitates that a threshold be applied to some application-specific norm that measures the distance between the node and edge properties of G^1 and G^2. The threshold is a truncation of data, the effects of which cannot be recovered from (other than by repeating with a different threshold). The threshold can eliminate any possibility of including a given pair of nodes in the final clique. Hence it is somewhat of a devastating operation since it is performed first, during the formation of A_g.

1.2.2 Relaxation Labeling Techniques

Widespread effort has been focused on the method of relaxation labeling [20] [21]. This approach describes the compatibility of a given labeling using continuous values, rather than a discrete assignment. Compatibility is based on the expected occurence of a given label and on the consistency of neighboring labels. A cost function is defined that describes the support for a given labeling arrangement. The labeling assignment is iteratively optimized to find a local maximum of the mean compatibility of all assignments. The local nature of the optimization can be a limitation, as it introduces a strong dependence on a good initial guess.

This approach can be implemented in a manner that avoids the construction of an association graph. This is an advantage over clique-based approaches. It is approximate, however, in that the final labeling assignments may not be unambiguous and may not correspond to a maximal clique [1].

1.2.3 Tree Search Techniques

These types of approaches match scene features to database features, starting with the most similar elements [22] [8]. Node and edge properties are typically employed here, as well as local comparisons of node connectivity. This is an incremental process that improves the estimate of an object's pose with each step down the search tree. Kalman filters and other methods can be used to refine the estimate of the object's pose [1].

Typically a small set of local features will be matched to the model at each step down the search tree. This local feature set (LFS) is formed such that the locations of its members completely determine the pose between the the model and scene. A complete specification of the object's transform is advantageous because it allows the expected location of the next LFS to be computed without ambiguity. This allows incorrect object models or improper object-to-scene matches to be rejected at relatively high levels in the search tree. However, this does place a constraint on the selection of features when forming the LFSs, in that the features must be grouped during the matching process. This grouping may not always yield the optimal ordering for recognition purposes.

1.3 Review of Related Studies from Mathematics Community

A graph G: (N,Np,E,Ep) is defined to have the typical nodes (N) and edges (E) as well as both node properties (Np) and edge properties (Ep). Let G^1 and G^2 be two graphs with N^1 and N^2 nodes, respectively. G^1 and G^2 are said to be isomorphic if there exists a mapping which is both one-to-one and onto that associates the nodes of G^1 to G^2. The mapping also must maintain all adjacencies so that any pair of adjacent nodes in G^1 are also adjacent in G^2 under the mapping [16]. For object recognition it is also required that both node and edge properties be maintained by the mapping. The similarity of these properties must be determined by norms or other measures that are application dependent. DCA finds a match between a given graph (a "scene") and a set of known graphs (a "database"). The reported match is found using the above norms and DCA's measure of topological similarity.

Subgraph isomorphism is a condition of isomorphism that exists between two subgraphs. Object recognition is fundamentally a problem of subgraph isomorphism [1] in that a database describes its objects in their entirety; in contrast to an observed object where all features are not typically seen in a single view. Real scenes also include clutter resulting in a scene graph that contains extra nodes and edges, beyond a simple subset of database graphs.

The class of difficulty for the isomorphism problem is not known [16]. A brute force approach requires effort O(N!) for a graph of N nodes. This is reduced somewhat in cases where automorphisms of a given graph make certain sets of nodes interchangeable without affecting the graph's topology [23]. Note that in the object recognition scenario this notion of automorphism must be extended to include the similarity of both node and edge properties.

The subgraph isomorphism problem is proven to be NP-complete [16]. This class of problems require a worst-case amount of effort that is of exponential order $O(a^N)$ and are considered to be intractable for many applications. The number of possible solutions is reduced by any automorphisms that may exist in the subgraphs. Efficient isomorphism algorithms do exist for certain special types of graphs [17] [24] [25] [26] [27] [28]. Unfortunately these won't apply well to object recognition problems because of the restricted nature of the graph topologies that are addressed therein.

Efficient solutions have also been developed in the random graph community [29] [30] [23]. The latter techniques are able to handle nearly all possible random graphs. The small fraction of graphs which must be rejected tends to zero as N increases in each of the methods. A problem limiting the application of these techniques to object recognition is the assumption of a purely random graph structure. Two methods are typically used to create graphs, Models A and B. Model A uses a given probability (often 0.5) to determine the existence of an edge in each entry of the adjacency matrix [23]. Model B starts with a given number of edges and places these randomly between nodes. If polygonal objects are assumed, for example, then neither Model A or B is the best choice. In this case the mean number of sides of the polygonal surfaces will determine the mean number of edges that are incident to each node. This implies that the number of edges per node in a database graph should be roughly constant - it should not be a function of the number of nodes - as is the case with Models A and B. Graphs of this type are known as strongly-regular. This is a significant factor because these types of graphs produce the most challenging inputs to isomorphism routines [24] [31].

The techniques of [29] [30] use Model A. In [29], for example, it is observed that random graphs of sufficient size will tend to have a subset of nodes that each have a unique degree. These nodes are used to form a "foundation" for their node classification scheme. The remaining nodes are characterized by their connectivity into this "foundation". Hence, this style of approach will have limited application to object recognition because of the Model A assumption and the requirement for a subset of nodes to have unique degrees.

The above techniques focus strictly on full-sized isomorphism problems, not on the subgraph isomorphism case, which is combinatorially much worse. A subisomorphism algorithm presented by [31] is more efficient than a complete enumeration and is noteworthy because it does not take advantage of the additional constraints that node and edge properties offer. It operates on graphs based on a connectivity analysis alone using a tree search approach with judicious pruning. Unfortunately [31] does not estimate the effort required for the algorithm's subisomorphism version. They abandoned their approach at the level of $(N_s,N_d) \approx (10,15)$ describing this size of problem as being "uncomfortably large".

1.4 Object Representation

It is believed that the DCA approach to finding graph subisomorphisms can have applicability beyond object recognition as in [32], but here, the algorithm will be presented in this context. To aide this presentation, a specific style of object representation will be used in the discussion. The representation assumed herein will be a 2 1/2-D relational surface patch model [12]. In general these graphs have surface shape parameters associated with their nodes. Edges are used to describe the relative orientation between adjacent surfaces. See Fig. 1 for an example of this type of representation.

In general there are tradeoffs between the overall size of an object's graph and the fineness of the representation. An example of a finer representation than above is a winged-edge graph [33]. These include nodes for each surface, surface-edge and corner. Edges in the graph would have to be redefined accordingly. This has the advantage of providing a more redundant description of an object and hence, improving noise tolerance. However this type of representation will also produce larger graphs which could be computationally prohibitive. These types of tradeoffs are being examined using randomly generated test cases.

2 Direct Classification of Node Attendance: Algorithm and Performance Metrics

The novelty and practical utility of the direct-classification of node attendance (DCA) approach is derived from an integration of both application-specific data and a topological description as a means for node consistency checks. If the graphs G^1 and G^2 are under scrutiny and node n_i^1 of G^1 and n_k^2 of G^2 are being compared, then the application-specific data includes an examination of the node properties of n_i^1 and n_k^2, the properties of those edges incident to n_i^1 and n_k^2, as well as the similarity of node properties for all nodes that are adjacent to n_i^1 and n_k^2. See Fig. 2.

The connectivity signature describes the local topology of the graph. The extent of the signature is variable and DCA's performance has been studied as function of this parameter. If too restricted, not enough connectivity information is included for any benefit. If too broad, then problems occur because of mismatches in the connectivity signatures of the scene and database graphs that are due to the absence of unobserved nodes in the scene graph.

The extent of the connectivity signature must be tuned for an application. It is largely determined by

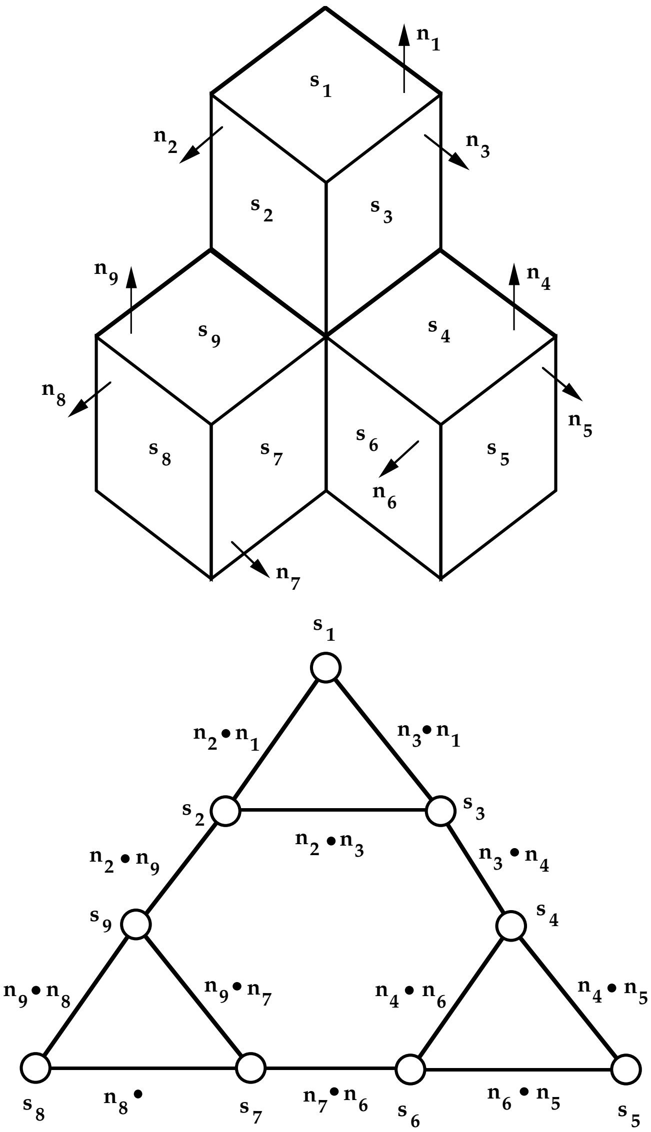

Figure 1: *Example of the 2 1/2-D relational surface patch representation. Scene on the left contains a cube with 3 visible surfaces,* s_i. *Node properties on the right are used to describe these surfaces. Edge properties describe the relative orientations of the surface normals using a dot product.*

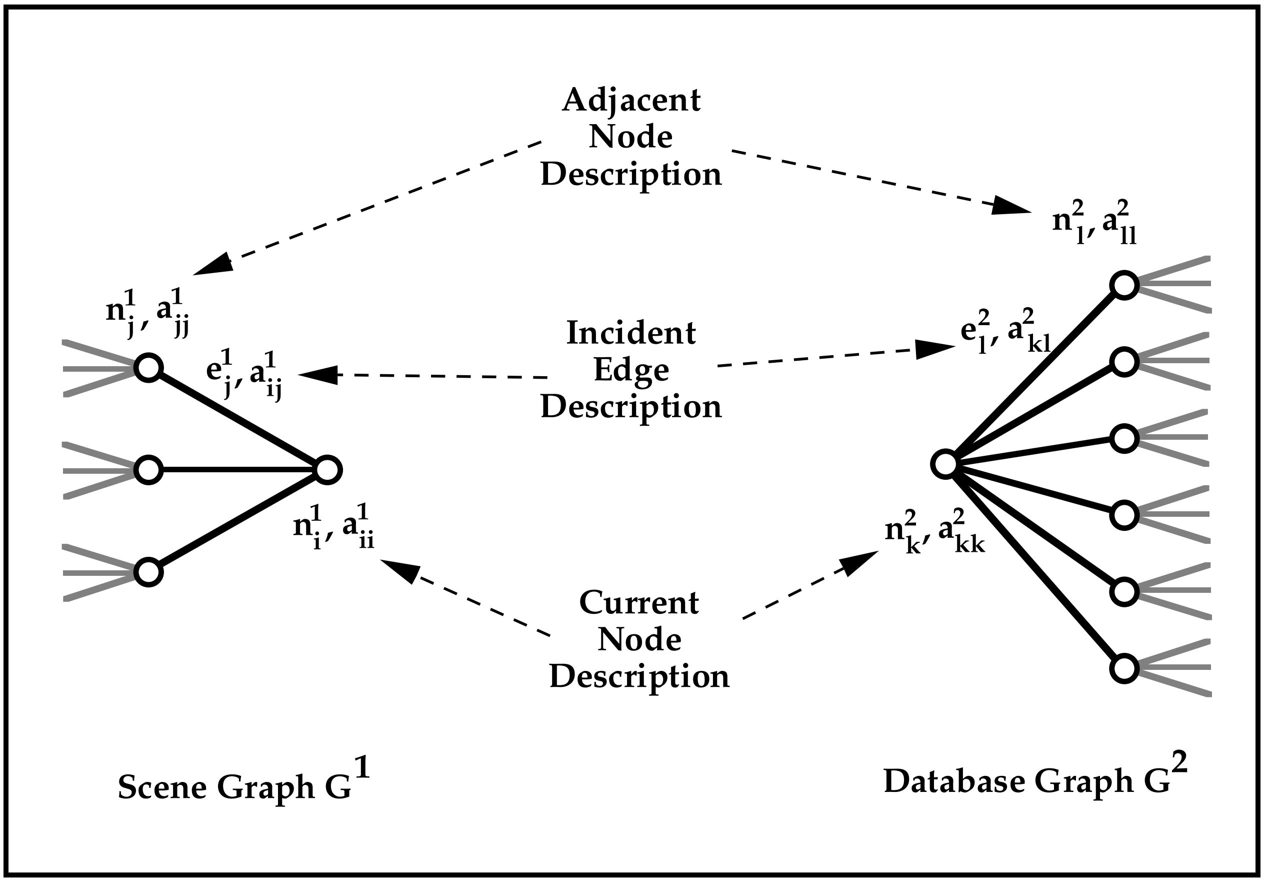

Figure 2: *Graph data used when forming the local characterization of nodes.*

the expected observability of objects that is associated with a given sensor. The connectivity signature is computed by forming A^m for $0 < m < V_{max}$, where A is the adjacency matrix. An element ${a_{ij}}^m$ of A^m, describes the number of paths from node i to node j of length m [16]. V_{max} determines the extent of the signature. These powers of A are used to form a data structure called the "A^m cube" which consists of layers of the matrices A^m. A^1 forms the lowest layer, followed by A^2, A^3, and so on, up to $A^{V_{max}}$. After each A^m layer is formed, its values are normalized by dividing through by the smallest nonzero element in the layer. In this way an element ${a_{ij}}^m$ describes the relative degree of connectivity from node i to j, when compared against all other paths of length m.

The connectivity signature for n_i^1 and n_j^1, $i \neq j$, is a vertical column in the A^m cube above each element of the original adjacency matrix (${a_{ij}}^1$), denoted by

$$\mathbf{a}_{ij} = \left| \begin{array}{c} {a_{ij}}^{V_{max}} \\ \vdots \\ {a_{ij}}^2 \\ {a_{ij}}^1 \end{array} \right| . \tag{1}$$

In this case n_i^1 and n_j^1 were distinct. A connectivity signature has also been defined for a single node. This describes the closed paths of various lengths that include n_i^1. These are denoted by $\mathbf{a}_{ii}$.

The general approach of DCA is to compare nodes by combining local comparisons of node and edge properties with the $\mathbf{a}_{ij}$ signatures. All node pairs n_i^1 and n_j^2 are compared in this fashion to form a total of N^1N^2 attendance ratings. Peak attendance values are identified. Currently this is done by a simple approach that passes over the attendance ratings to find a peak, records this node-to-node mapping, and repeats until a mapping is found for all node pairs. Matching subgraphs of the scene and database entry are formed, each of size $K \leq min(N^1, N^2)$. The nodes of the scene subgraph are reordered using the attendance peaks so that n_1^1 and n_1^2 are associated with a peak, as are n_2^1 and n_2^2 and so on.

At this stage all the properties of n_i^1 and n_i^2 can be directly compared and a verification step is performed. Thresholds are applied to the difference in node and edge properties and to differences in the adjacency matrix. All subgraph nodes are examined in this manner and the worst matching node is removed. This process is repeated (at most K times) until either a null graph exists or until all properties match suitably well. Any remaining nodes with zero degree are then removed. Note that if this is compared to a hypothesis-and-test approach to object recognition, then here only a single hypothesis and a single test are performed for each database entry.

Good performance of the algorithm was achieved with $V_{max} \approx N\rho$, where $N = min(N^1, N^2)$, $\rho = 0.25$. Values of V_{max} in this range are necessary due to the limited observability of an object that is possible from a given viewpoint. V_{max} is related to $N\rho$ because path lengths used in the connectivity signature need to be limited in order to provide the best possible match between the database and expected scene conditions. Note that when forming connectivity signatures for the database, V_{max}

could be set to N. However, if this same value of V_{max} were used for the scene, then the resulting signatures would be dissimilar because the subgraph in the scene won't contain all the paths in the database entry. For this reason the value of V_{max} must be limited when the database is analyzed so as to match the expected scene conditions as best as possible. When DCA is applied to an isomorphism problem, the V_{max} parameter can vary over a broader range with little effect on performance.

2.1 DCA Algorithm

Subgraph Formation Stage
Compare each pair of nodes in each graph
1)For each node of n_i^1 of G^1:
  2)For each node of n_k^2 of G^2:
    Compare edges incident to current nodes
    3)For each edge e_j^1 incident to n_i^1:
      4)For each edge e_l^2 incident to n_k^2:
        4a)Compare connectivity of a_{ij}^1 with a_{kl}^2
        4b)Compare adjacent nodes' connectivity of a_{jj}^1 with a_{ll}^2
        4c)Compare edge properties of e_j^1 with e_l^2
        4d)Compare adjacent nodes' properties of n_j^1 with n_l^2
        4e)Combine results of (4a) to (4d) via IOP
    3a)Save comparison of best matching edges and adjacent nodes
  Compare current nodes
  2a)Compare connectivity of a_{ii}^1 with a_{kk}^2
  2b)Compare node properties of n_i^1 with n_k^2
  Find similarity of current nodes
  2c)Combine (3a) (2a) and (2b) to form attendance rating of n_i^1 to n_k^2

Form matching subgraphs
5) Find peaks of attendance ratings to define mapping from all n_i^1 to n_k^2
6) Form subgraphs g^1 and g^2 using node pairs n_i^1 and n_k^2 with attendance ratings above threshold T
7) Use node-to-node mapping to reorder nodes in g^1 such that n_1^1 maps to n_1^2, n_2^1 maps to n_2^2, etc

Subgraph Verification Stage
8) Eliminate nodes with poorly matching Np
9) Eliminate nodes with poorly matching Ep
10) Eliminate nodes with poorly matching A^1

In step (6) the subgraphs g^1 and g^2 of G^1 and G^2 respectively, are formed by applying the threshold T to the attendance ratings in order to find "significant" node pairs. A value of 0.5 was used for T in all the tests reported herein.

In steps (4c), (4d) and (2b) node and edge properties are compared. These methods of comparison are application-dependent. In these tests, node and edge properties were all scalars. A pair of scalar properties (a,b) were compared using the ad hoc relationship

$$1.0/(1.0 + |a - b|) \tag{2}$$

where || denotes absolute value. Connectivity signatures are compared in (4a), (4b) and (2a). Two signatures (c,d) are compared in a similar means using

$$\frac{1.0}{V_{max}} \sum_{v=1}^{V_{max}} 1.0/(1.0 + |c^v - d^v|). \tag{3}$$

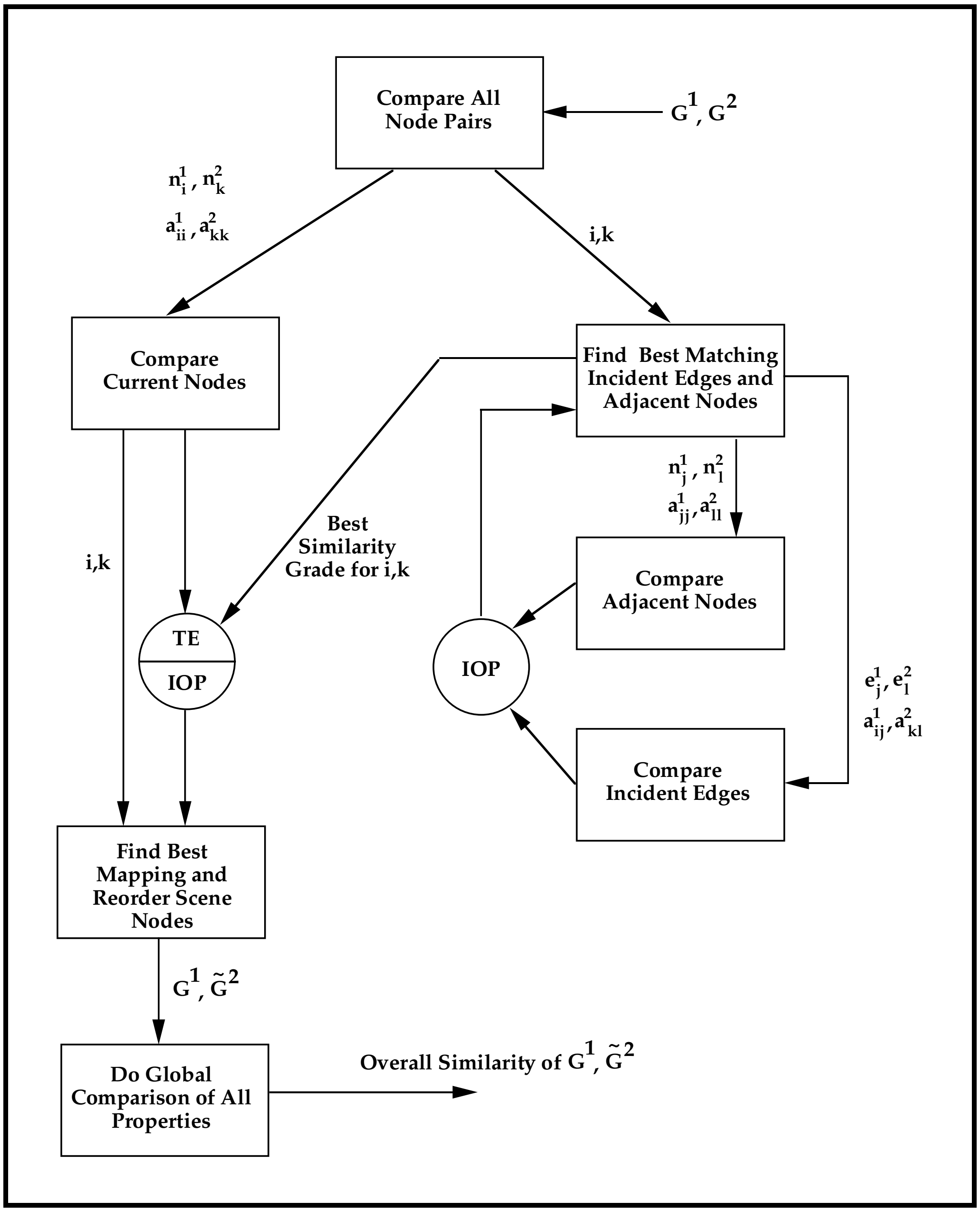

Figure 3: *Steps involved in node classification. Unlabeled arrows describe data flows of attendance ratings. Circles indicate operations where attendance ratings are combined via either the Theory of Evidence (TE), or by an Independent Opinion Pole (IOP).*

The selection of the methods used to combine evidence in (4e) and (2c) is critical to achieving good performance with DCA. See Fig. 3. No optimal means for combining evidence is claimed to have been found. Experiments have been performed using various methods, such as the Independent Opinion Pole (IOP), Linear Opinion Pole, Harmonic Opinion Pole [34] and Theory of Evidence (TE) [35]. Combinations of these techniques have also been attempted. Bayesian classifiers [36] were used to combine the results of multiple techniques, however this did not appreciably impact final results.

When analyzing subgraphs, the method that appeared to work best used TE

$$s_{ds} = 1.0 - (1.0 - s_1)(1.0 - s_2) \tag{4}$$

where s_1 and s_2 are attendance ratings and s_{ds} is a combined value. When analyzing full-sized graphs in an isomorphism problem, IOP

$$s_{iop} = s_1 s_2 \tag{5}$$

worked well. This mode switch was required at step (2c) when combining the evidence associated with neighboring nodes with the current nodes. The two modes of operation are indicated in Fig. 3 by the split circle. The unlabeled arrows in the figure represent data flows of attendance ratings. The circles indicate an operation taking place that combines attendance values.

2.2 Performance Metrics for DCA

Several performance measures have been established for DCA that allow test conditions to be evaluated and parameters to be adjusted. Fig. 4 shows a histogram of attendance ratings for nodes that were members of present and absent subgraphs. Dual data sets are plotted with pairs of light and dark bars on the same scale. This "dual histogram" has left (darker) entries that describe the attendance of nodes belonging to database graphs that were absent from the scene. The right (lighter) entries describe the attendance of nodes that belonged to database graphs that were present in a scene. Nodes of these two categories are referred to as "present" and "absent" nodes.

As seen in the histogram, absent nodes tend to rank lower and present nodes rank higher on the scale. The segmentation of nodes is not perfect in the case of subisomorphisms. Note the subset of absent nodes that rank high on the scale. It is believed that this is somewhat unavoidable because these entries represent single node subisomorphisms, which seem to occur relatively often. The verification step that completes DCA processing eliminates these nodes from the final subgraph by comparing the reordered adjacency matricies. This enforces a global connectivity check which is above and beyond the requirement of matching local properties.

Examining the dual histograms is not always a certain indication of DCA's ability to recognize objects. It can give a rough indication. In full-size isomorphism problems, the segmentation is very good - see Fig. 5 - and recognition success rates have been benchmarked at 100 out of 100 cases. Also note that this test case had a dynamic range of *zero* for the node and edge properties, and was run on

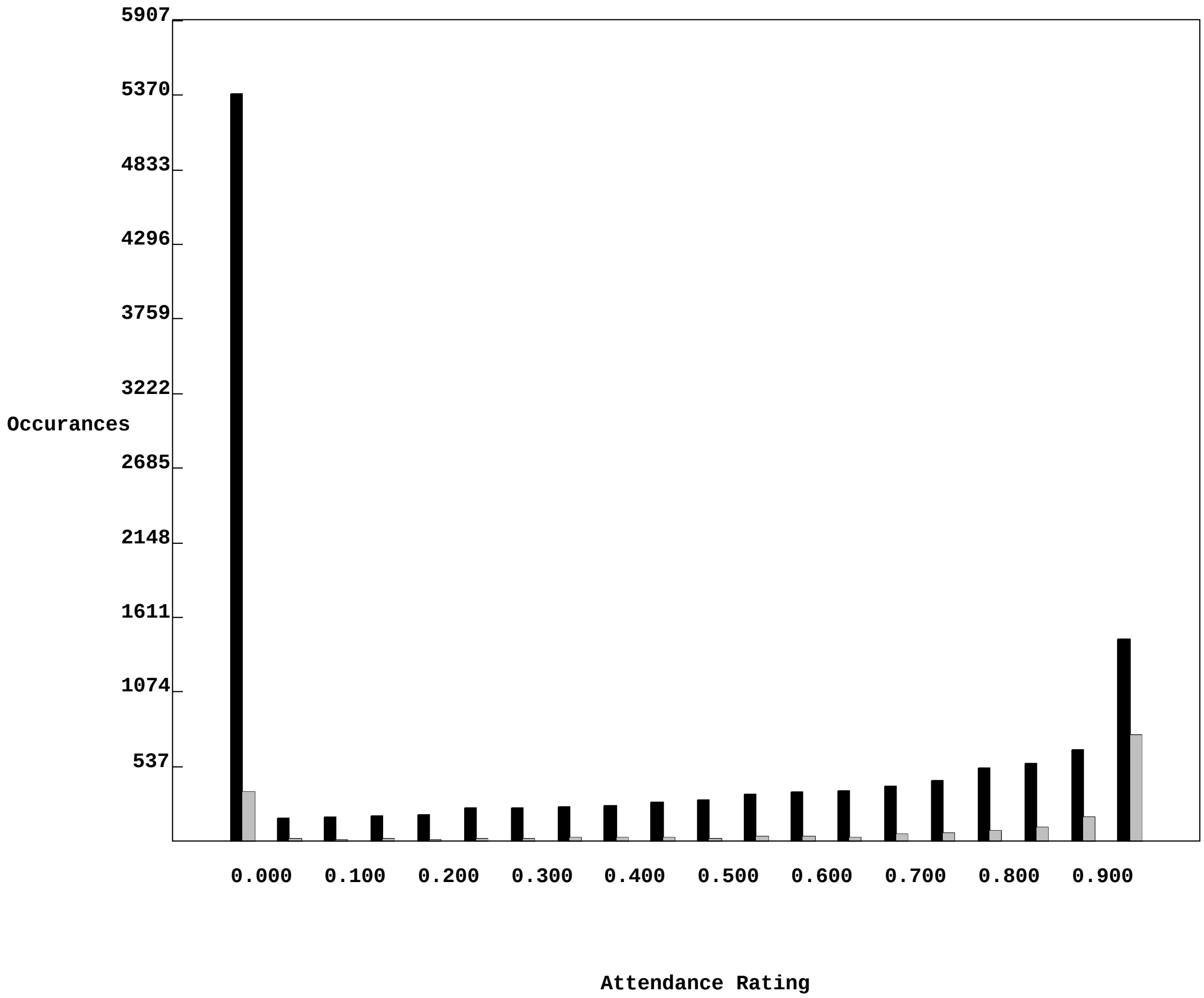

Figure 4: *Dual histogram of attendance ratings recorded in a subisomorphism problem. Entries for present nodes are given by the lighter (right) bars. Entries associated with absent nodes are indicated with the darker (left) bars. Present and absent nodes are segmented using the attendance rating. Dichotimization is sufficient given such a distribution, provided the global verification step is performed to complete DCA processing.*

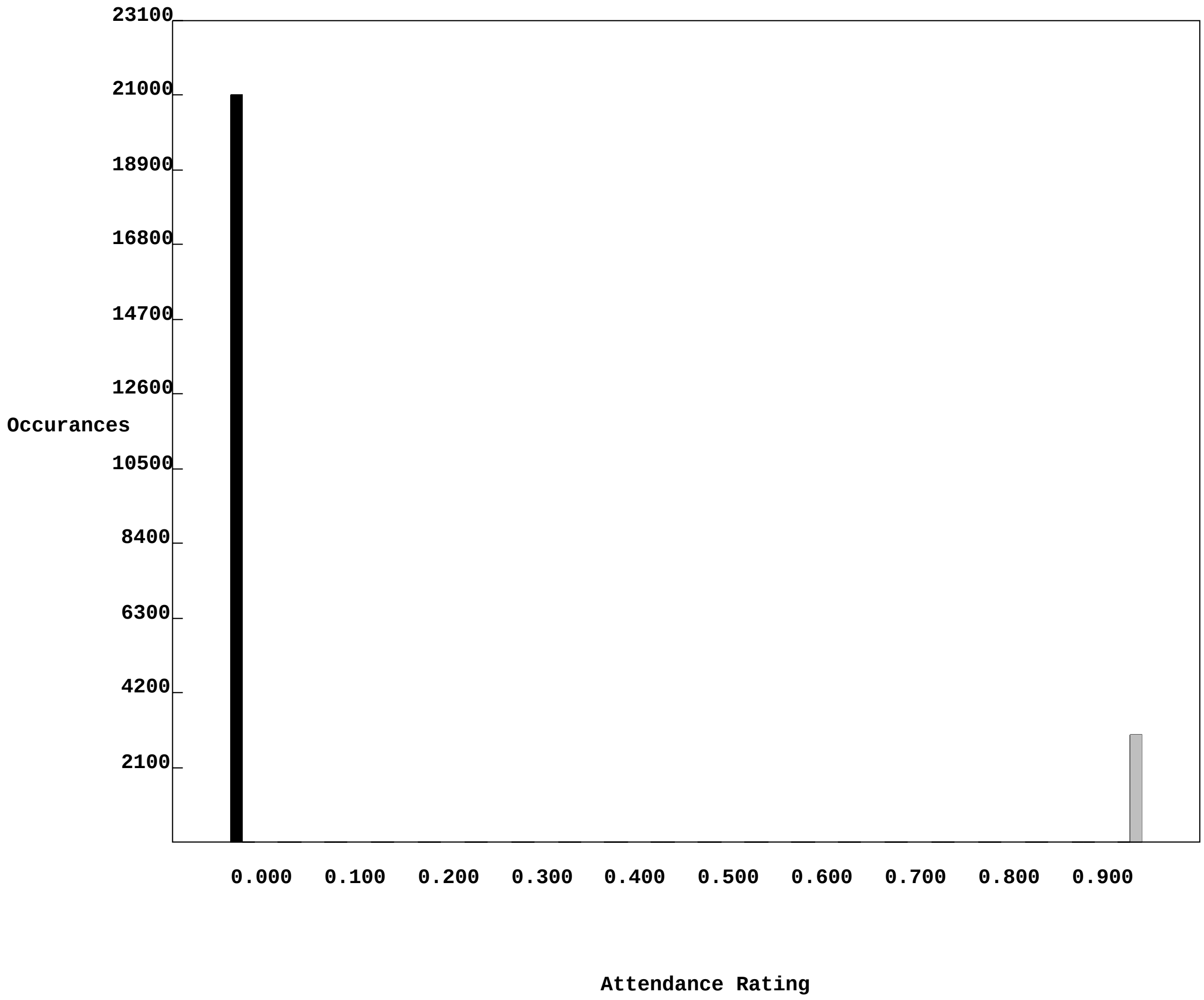

Figure 5: *Dual histogram of the attendance ratings in an isomorphism problem. Note the dichotimization is significantly improved over the subisomorphism case. These data were recorded from a test with node and edge properties having a* **zero** *dynamic range and with strongly-regular graphs.*

strongly-regular graphs. The matching in these tests was achieved solely via the connectivity signatures.

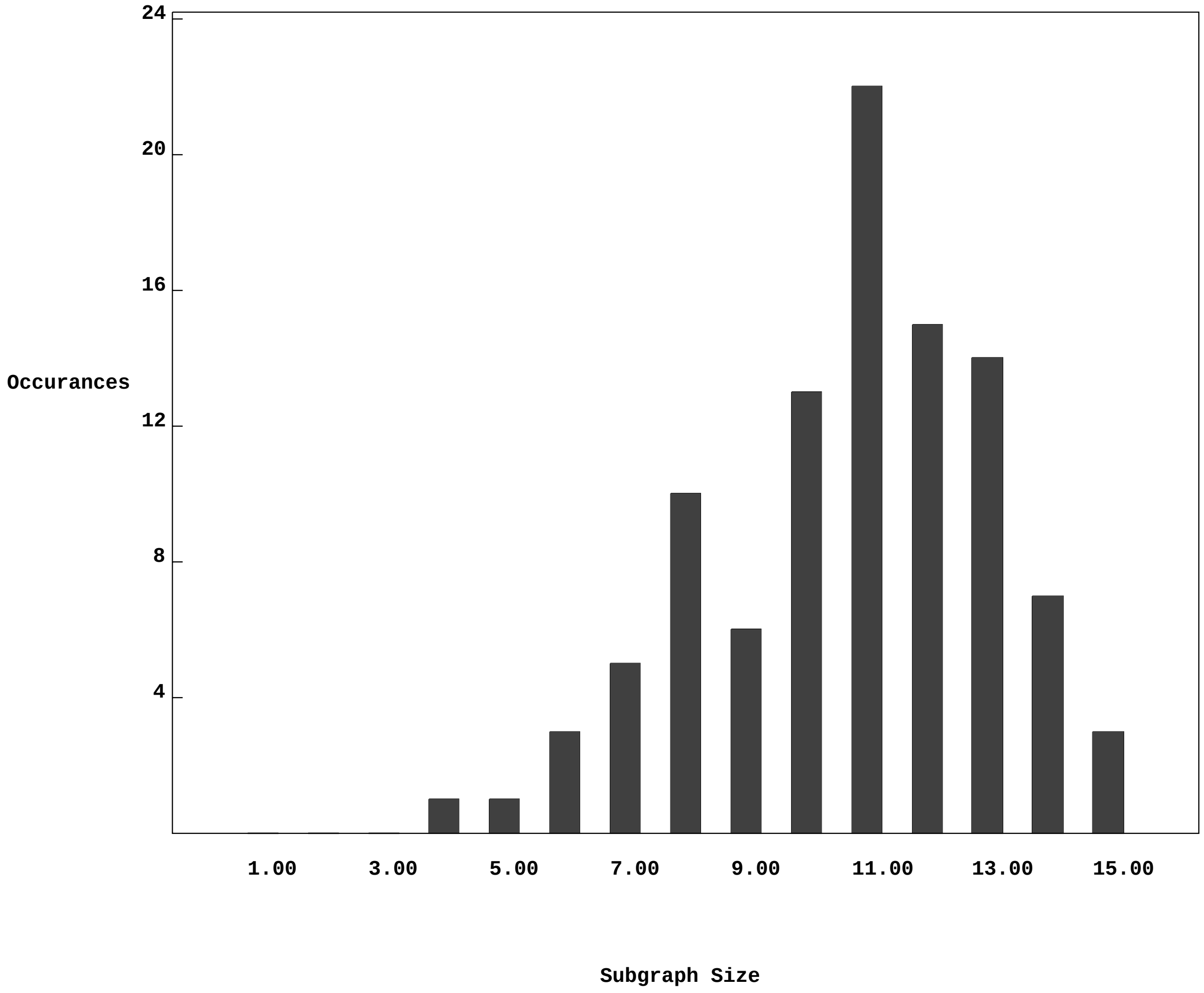

Figure 6: *Histogram of subgraph size. This test involved database graphs with 30 nodes and a single scene graph having 15 nodes. This subisomorphism problem had a mean* $\pm$ *standard deviation of* 10.7 ± 2.3 *nodes.*

Another benchmark of DCA's performance is the size of matched subgraphs. Fig. 6 shows a histogram of these sizes on tests involving database graphs with 30 nodes and a single scene graph having 15 nodes. These 100 trials had a mean subgraph size of 10.7 and standard deviation of 2.3 nodes. A metric is needed to determine the best match between the scene contents and database entries. Subgraph size has been weighted by a norm measuring the closeness of node and edge properties for this final quality measure.

During testing each node of each graph in the database was assigned a unique name tag for use as a final check of correctness. The percentage of correct name tags was also tallied as a performance measure. Unfortunately random subgraph automorphisms do exist in this type of testing, and the name tags flag automorphisms as an error, despite perfectly matching topologies and all other properties. Fig. 7 shows a histogram of the fraction of correct labels in the case described above. The mean $\pm$ standard deviation of this metric in the test case cited above was $98 \pm 5\%$.

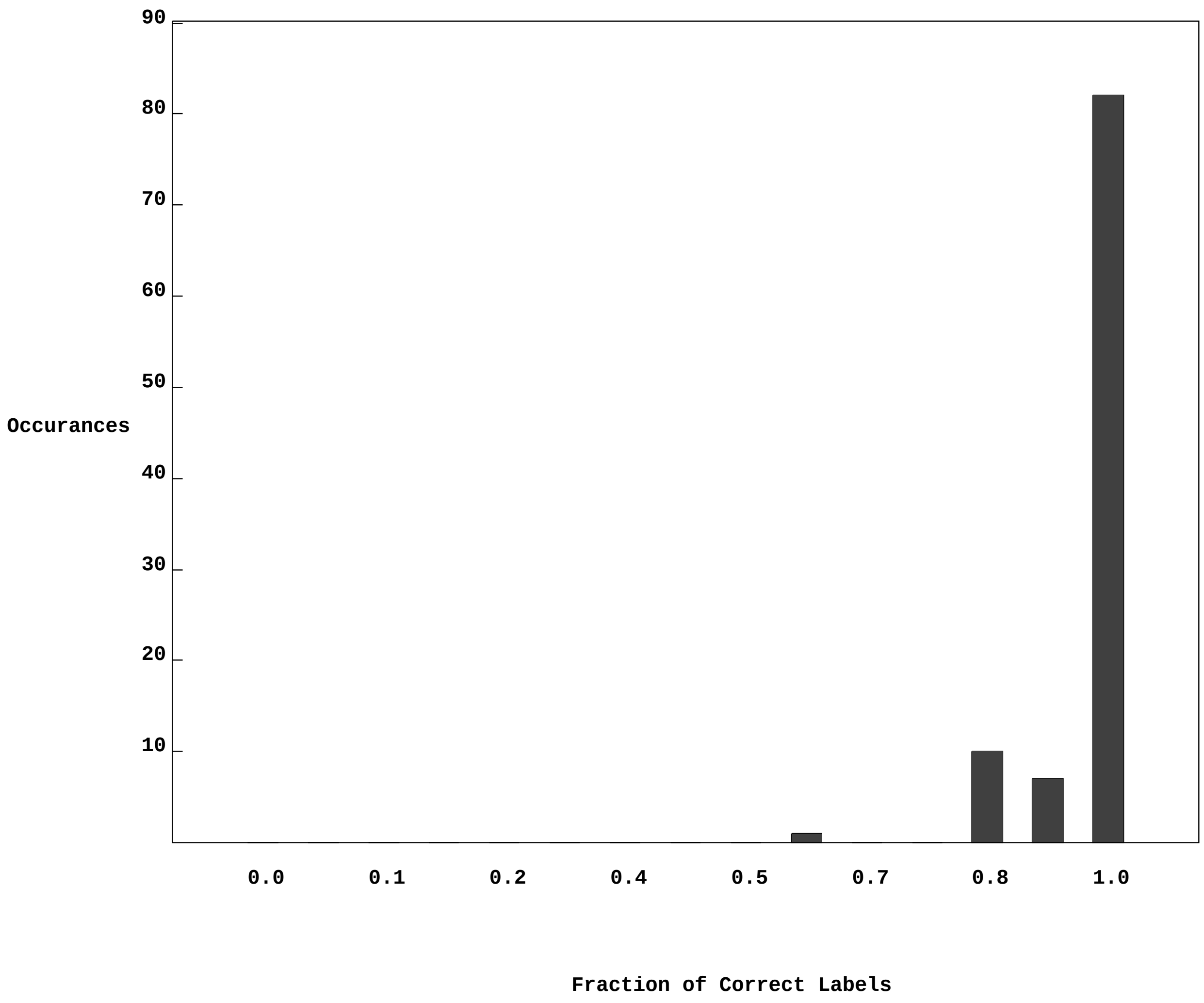

Figure 7: *Fraction of correct labels. This test involved database graphs with 30 nodes and a single scene graph having 15 nodes. The mean* $\pm$ *standard deviation of correct labels was* $98 \pm 5\%$.

2.3 Computational and Memory Requirements for DCA

The effort required for each database examination is O(N^3) for graphs with N nodes. Loops (1) and (2) in the algorithm have an upper limit of N_s and N_d. Loops (3) and (4) have a limit determined by the shape complexity of objects, which is not a function of N_d because of the strongly-regular graphs. Hence, the limit on loops (3) and (4) is determined by the mean degree of the nodes. Comparing the connectivity signatures requires effort O(V_{max}). Overall this yields an O(N_s N_d V_{max}) which is O(N^3).

The A^m Cube must also be generated for each scene. This is an O(N^4) operation, but is only done once, prior to the database examinations. DCA requires O(N^3) memory for the A^m Cubes.

2.4 Pose Determination

Once DCA has isolated and reordered the best matching subgraphs of a scene and database, an object's pose can be determined. This portion of the DCA approach has not been implemented to date. However, the pose determination step will be a straight forward process employing established techniques. Given two matching subgraphs, the coordinates of physical features in the scene and database can be used to find the needed transformation. This can be accomplished via a closed-form solution [37]. If some nodes have been improperly mapped during the matching process then these errors will be revealed by examining the residual error of the transformed data points. Removing these outliers and refitting the transform will serve as a final verification step.

Once the transform of a core subgraph is found, it should be possible to gather additional scene nodes into the segmented object. This could be accomplished either by traversing the database entry and examining the scene for consistent nodes, or by a search outward from the scene's subgraph looking for new scene nodes that can be included in the object. By whatever means chosen to generate new scene nodes for consideration, the new nodes will have to agree with the object's topology and the established transformation. This agglomerative process will be similar to the latter stages of a tree search technique. DCA has the advantage of providing a good starting point for the search. This can be seen in the results documenting the final subgraph size.

2.5 Justification of DCA

An ideal solution to the graph isomorphism problem would be to find a node classification scheme that uniquely characterizes all nodes in any case. No such classification is known to exist for this general problem [23] [24] [16]. Of course, if the dynamic range of node or edge properties becomes large enough then this classification becomes trivial. In the more general case it is necessary to form a classification scheme that involves a node's topological relationship to the rest of the graph. The added challenge of finding a general classification for subgraphs can be appreciated by considering that by definition, subgraphs are missing portions of their structure. Hence any topologically-oriented metric will tend to be affected during subgraph formation. The extent of the connectivity signature is limited to help mitigate this tendency.

The connectivity measure has the effect of extending the characterization of nodes and edges. This increases the richness of the local description by using information that is independent of node and edge properties. Note that the connectivity signature is not unique. This is the underlying factor in the approximate nature of the results obtained by DCA. Future efforts will include a theoretical performance evaluation based on randomly generated, strongly-regular graphs. Simulations will then be run for comparison against the theoretical limits. Herein only this intuitive justification is presented.

The mode switch between IOP and TE in step (2c) of the algorithm can be appreciated by considering the fundamental difference between the nature of the matching that occurs in subgraphs vs. full-sized graphs. In a full-size isomorphism problem, all nodes need to be matched somewhere. This situation is more consistent with Bayesian reasoning in that good local matches imply that a node-to-node mapping is correct, and poor local matches imply the mapping is not correct. IOP is a Bayesian means to combine evidence. In the subgraph case, poor local matches don't necessarily imply that a node should be mapped elsewhere - it may mean that the node should not be included in the matching subgraph at all. Poor matches in subgraphs do not provide any evidence to help establish the disposition of a node's attendance. This is consistent with the reasoning in the Theory of Evidence [35]. It is this fundamental difference in the conclusions which can be drawn from the attendance rating that causes the mode switch in DCA.

3 Object Recognition Using DCA: Testing Methodology

Test parameters have been designed to reflect conditions in an object recognition scenario as closely as possible, while still permitting the use of randomly generated cases that could be run in large numbers. This section discusses how the test conditions were derived and presents the testing procedure in a step-by-step manner.

3.1 Relation of Test Conditions to an Object Recognition Scenario

One test parameter is the fraction of a database graph that is observable in a scene. Consider a rectangular solid for example, most viewpoints reveal 1/2 of its surfaces. Certain views can restrict the fraction down to 1/6. Hence most tests were performed with randomly generated subgraphs that were half the size of their original database versions.

The amount of interconnectivity in a graph was an important test parameter. As discussed above, strongly-regular graphs are the most appropriate for object representation. All tests were performed with this type of graph. Note that a graph which has precisely the same number of edges incident to each node can not exist in all arbitrary cases [16]. Hence an algorithm was written to generate graphs that are strongly-regular (or almost regular) by simply adding edges randomly between any nodes that are below a given degree. This generated graphs rapidly and maintained the near regular condition.

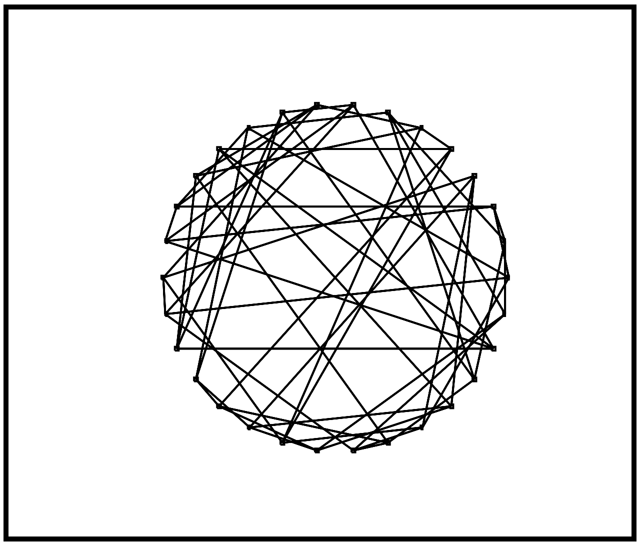

Figure 8: *An example of a randomly generated database entry. This graph is strongly-regular, each node has degree* ≈ 4.

The graph in Fig. 8 was generated using this method.

Another important parameter that affects the applicability of the test results is the dynamic range of node and edge properties. In some applications node properties can be made quite rich involving various shape parameters, texture or color, and so forth [8]. This sort of node property would likely be a vector quantity. Only scalar integer quantities were considered here. This was in an attempt to keep the tests challenging. In most tests a dynamic range of 4 was used for both node and edge properties. This can be thought of as a set of objects having surfaces with 4 distinct shapes, for example. This includes all possible colorations, textures, shapes and sizes. The dynamic range of edge properties can be envisioned as 4 discrete relative orientations between adjacent surfaces. In a 2-D application involving the matching of features between two images [38], edge properties could be used to describe the North, South, East or West relative locations. These values were selected with the assumption that most machine vision applications would have properties with at least this much dynamic range.

The noise level added to each node and edge property during the simulated sensing process was also varied. Flat noise with zero mean was added to each node and edge property. A noise peak of 0.25 was typically used. This corresponded to a +/-12.5% peak variation away from the ideal values stored in the database.

3.2 Testing Procedure

The individual steps in the testing process are described in Fig. 9. Each trial consists of 100 cases, beginning with the generation of a database of objects. A number of database entries were randomly selected to be scene contributors. A subgraph was formed for each of these by randomly dropping nodes. The node ordering was then randomly perturbed and a scene constructed using all contributors.

Topological noise was added by placing random interconnecting edges between the scene contributors and by adding spurious nodes. DCA then compared the scene against each database entry. The best matching database entry was retained as the final selection.

Fig. 10 shows a scene constructed from a single object's subgraph. Four additional noise nodes were added to the scene to represent the effect of background clutter. The topological noise can be seen in the 4 edges that connect the central nodes to the 4 peripheral (noise) nodes. Fig. 11 has two contributing subgraphs, 8 outlying nodes, and 4 interconnecting edges between the database contributors. In both of these scenes, subgraphs had 1/2 the number of nodes of their original database versions.

4 Experimental Verification: Results and Implications

Extensive simulations of the recognition process have been completed and results follow. These tests have been performed in an abstract graphical domain. Hence, the physical location and orientation of object features were not part of the testing process. For this reason tests on pose determination have not been completed. All tests were run on a Silicon Graphics Indy2 workstation that had a 100 MHz R4000 RISC processor with an R4010 floating point unit. Tests were run at normal priorities and in a multi-user environment.

Tables are used to present the test conditions and performance measures below. An explanation of the table headings follows.

- "D/S Number of Graphs" in the first column designates to the number of graphs in the database and scene, respectively.

- "D/S Size of Graphs" gives the size of graphs in the database followed by the size of each subgraph contributor to the scene.

- "Excess N/E in Scene" are the number of extra nodes added to each scene and number of extra interconnecting edges added between each subgraph contributor.

- The "Rate Hz" is the average rate at which each database entry was examined in Hertz. Note that some preliminary processing was necessary for the scene graph. This was only performed once for each scene. For graphs of these sizes this required less than one second and is not included in these figures.

- "% Correct Selection" is the percentage of trials in which correct database contributors were selected. All tests were performed with 100 trials.

- "Subgraph Size" gives the mean $\pm$ standard deviation of the size of the extracted subgraphs. These subgraphs were reduced in size until all properties and topologies of the scene and database matched within specified tolerances.

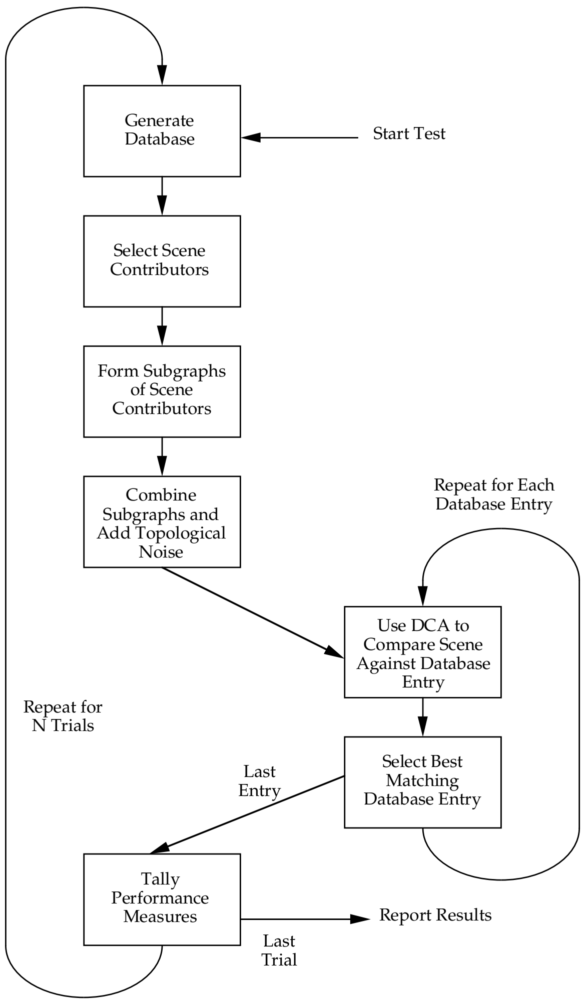

Figure 9: *DCA testing procedure. Typically 100 trials were run for each set of test conditions. Each trial involved the random generation of databases and scenes. Both topological and property noise were then added. Three performance metrics were used for benchmarking.*

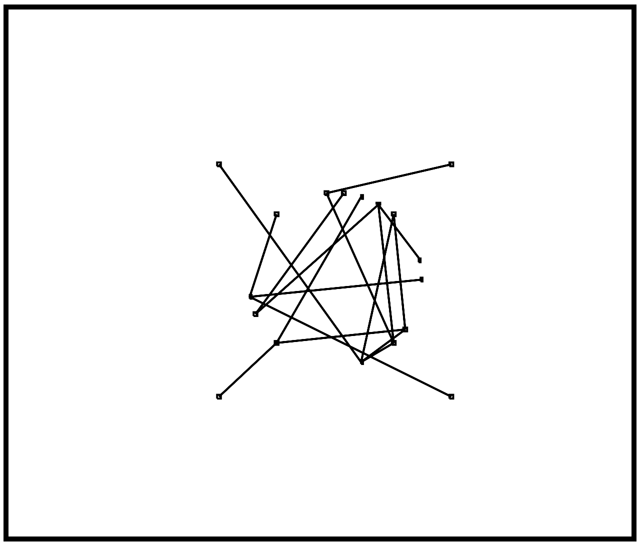

Figure 10: *A scene graph generated from randomly selected database contributors. This scene is composed of a single subgraph with some added topological noise that can be seen in the 4 edges that connect the central nodes to the 4 peripheral nodes.*

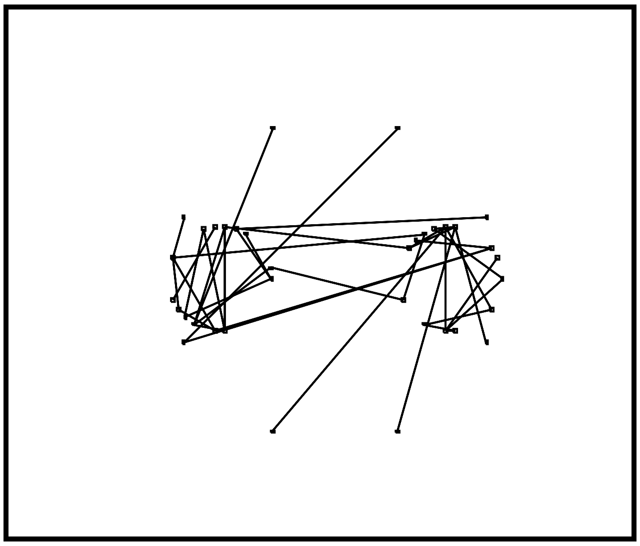

Figure 11: *A scene graph formed from 2 database contributors. Excess nodes and edges are present.*

Table 1: *Summary of test results when matching full-size graphs. The graphs in these tests had no node or edge properties, no property noise and no topological noise. All graphs were strongly-regular with degree ≈ 4. Performance was maintained at nearly 100%, provided the graph size was above ≈ 10.*

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	10/10	0/0	0/0, .0/.0	34.3	97	10.0 ±0.0	75.6 ± 26.1
8/1	16/16	0/0	0/0, .0/.0	9.5	100	16.0 ± 0.0	99.5 ± 3.0
8/1	20/20	0/0	0/0, .0/.0	2.6	100	20.0 ± 0.0	99.8 ± 1.4
8/1	24/24	0/0	0/0, .0/.0	2.7	100	24.0 ± 0.0	99.7 ± 1.6
8/1	30/30	0/0	0/0, .0/.0	1.3	100	30.0 ± 0.0	100.0 ± 0.0
8/1	40/40	0/0	0/0, .0/.0	0.4	100	40.0 ± 0.0	99.9 ± 0.7

- "% Correct Tag" is the percentage of correct name tags present in the matched subgraphs. The name tags uniquely identify each node of each graph in the database. This performance measure is also given in mean ± standard deviation form.

4.1 Tests Using DCA to Find Full-Sized Isomorphisms

DCA's yielded the best performance on full-sized isomorphism problems. To emphasize this a series of tests were run having a zero dynamic range for node and edge properties. Here, only the connectivity signatures were available as a means to determine node attendance. Table 1 show a correct database selection rate of 100/100 and a full sized matching subgraph in all tests. The percentage of correct name tags was not totally perfect. This is due in part to automorphisms which are more prevalent under these test conditions because the lack of node and edge properties makes nodes less distinctive. In all tests $V_{max} = N_s/2.0$ and IOP was used to combine evidence. The number of nodes ranged from 10 to 40 in these tests.

4.2 Tests Using DCA to Find Subisomorphisms

DCA's performance on subisomorphisms has been tested under a variety of conditions. These are grouped into two categories for clarity. The first examines size-related parameters and the second looks at effects associated with node and edge properties and with noise.

4.2.1 Problem Size Effects

Graph size and the number of graphs are important factors affecting performance. In general graphs need to be sufficiently large before the connectivity signature attains a sufficient character to help distinguish nodes. This can be seen in Tables 2 and 3, as well as in the tests on full-sized isomorphisms above. The fraction of nodes present in a scene subgraph was varied in Table 2. Changes in this fraction correspond to varying occlusion levels or to varying sensor capabilities.

All tests were run with 8 database graphs and 1 subgraph in each scene. The size of graphs in the database was fixed at 30 nodes. In all cases V_{max} was set using the ceiling function $V_{max} = ceil(N_s/4.0)$.

Table 2: *Summary of test results examining effect of subgraph size. These results indicate that at least half of the graph should be retained given these size and noise conditions.*

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	30/10	4/4	4/4, .25/.25	11.9	64	4.3 ± 1.9	91.4 ± 18.3
8/1	30/15	4/4	4/4, .25/.25	6.8	99	11.2 ± 2.3	96.9 ± 5.6
8/1	30/20	4/4	4/4, .25/.25	4.6	100	17.7 ± 1.6	98.9 ± 3.1
8/1	30/25	4/4	4/4, .25/.25	2.9	100	22.7 ± 1.4	99.9 ± 1.1

TE was used to combine evidence describing the node attendance. These results indicate that at least half of the graph should be retained under these conditions.

The effect of graph size is examined in Table 3. This effect is important because of its implications to object representation and on overall computational effort. Given these test conditions, it appears that a winged-edge graph may be appropriate for applications that involve relatively simple objects. In this representation a cube has 26 nodes. V_{max} and TE were used as above.

When a large number of graphs are present in the database the recognition process becomes more challenging because of the additional potential matching subgraphs that are available. Results in Table 4 indicate that most performance measures remain fairly constant except for percentage of correct database selections which dropped slightly to 90% when 64 database entries are present. These tests begin to give a feeling for the discriminatory ability of DCA. An application requiring many dozens of database objects may need to have an increased dynamic range for iits node and edge properties, than the ranges used here.

The optimal choice for the V_{max} parameter is related to size, so these tests have been included here. V_{max} parameter determines the extent of the connectivity signature. As seen in Table 5, performance peaks in the range $4 \leq V_{max} \leq 6$.

The best choice of V_{max} and other parameters will be largely application-dependent. The relative importance of subgraph size vs. correct database selection, vs. correct node mappings will vary between applications. In all other tests reported herein $V_{max} = ceil(N_s/4.0)$.

Another size-related issue is the number of subgraph contributors to each scene. In Table 6 performance metrics remain reasonably consistent, except for the percentage of correct name tags. This had a rather serious drop off, down to $\approx 75\%$ for scenes with 4 subgraphs. It should be possible to mitigate some of the effects of this degredation during the pose determination stage. At this point the residual error of each node's physical location can be computed and outliers can then be removed.

4.2.2 Evaluation of Dynamic Range and Noise Effects on DCA Performance

The dynamic range of node and edge properties has a very direct impact on performance because of its ability to improve the local characterization of nodes. Various combinations of node and edge ranges are presented in Table 7. Albeit, this is a tiny sampling of the span of all possible cases, it appears that

Table 3: *Summary of test results examining effect of graph size. Perfomance degrades as the graph size decreases due to the reduced information present in the connectivity signature.*

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	20/10	4/4	4/4, .25/.25	15.0	95	7.2 ± 1.7	95.3 ± 12.0
8/1	24/12	4/4	4/4, .25/.25	11.7	98	9.0 ± 2.0	94.5 ± 10.0
8/1	36/18	4/4	4/4, .25/.25	4.2	99	13.8 ± 2.6	97.7 ± 5.0
8/1	40/20	4/4	4/4, .25/.25	3.3	99	14.7 ± 2.8	96.0 ± 6.5

Table 4: *Summary of test results examining effect of the number of database elements. Most performance measures remain fairly constant except for the percentage of correct database selections.*

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
4/1	30/15	4/4	4/4, .25/.25	7.0	99	10.9 ± 2.3	96.4 ± 6.4
8/1	30/15	4/4	4/4, .25/.25	6.2	99	11.2 ± 2.3	96.9 ± 5.6
16/1	30/15	4/4	4/4, .25/.25	6.8	96	11.4 ± 2.0	95.7 ± 7.2
32/1	30/15	4/4	4/4, .25/.25	7.2	93	11.3 ± 2.1	96.1 ± 7.3
64/1	30/15	4/4	4/4, .25/.25	7.3	90	11.8 ± 2.0	97.5 ± 6.4

Table 5: *Summary of test results examining effect of V_{max}. Values for the V_{max} parameter are given in the second column. A value of 1 is trivial in the sense that no additional information is included in the connectivity signatures beyond that of the original Adjacency matrix. The need for the connectivity signature can be seen by the jump in correct database selections that accompanys nontrivial V_{max} extents. In all other testing $V_{max} = ceil(N_s/4.0)$.*

D/S Number of Graphs	D/S Size and V_{max}	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	30/15,1	4/4	4/4, .25/.25	8.9	8	2.0 ± 0.0	56.2 ± 46.4
8/1	30/15,2	4/4	4/4, .25/.25	8.1	96	11.6 ± 2.6	97.6 ± 6.1
8/1	30/15,3	4/4	4/4, .25/.25	7.3	96	11.8 ± 2.4	95.8 ± 8.3
8/1	30/15,4	4/4	4/4, .25/.25	7.0	99	11.2 ± 2.3	96.9 ± 5.6
8/1	30/15,5	4/4	4/4, .25/.25	6.4	100	10.7 ± 2.3	98.0 ± 4.9
8/1	30/15,6	4/4	4/4, .25/.25	6.0	100	9.0 ± 2.6	98.6 ± 6.7
8/1	30/15,7	4/4	4/4, .25/.25	5.6	99	7.0 ± 2.6	99.5 ± 3.0
8/1	30/15,8	4/4	4/4, .25/.25	5.1	89	4.7 ± 2.1	99.8 ± 1.5

Table 6: *Summary of results examining effect of the number of subgraphs present in scene. The percentage of correct name tags suffered a significant drop off here. These incorrectly matched nodes would have to be identified and removed during the pose determination process.*

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	30/15	4/4	4/4, .25/.25	6.8	99	11.2 ± 2.3	96.9 ± 5.6
8/2	30/15	4/4	4/4, .25/.25	3.1	98	10.1 ± 1.9	87.1 ± 12.5
8/3	30/15	4/4	4/4, .25/.25	1.8	94	10.0 ± 2.0	78.2 ± 16.6
8/4	30/15	4/4	4/4, .25/.25	1.2	98	9.9 ± 1.7	74.4 ± 16.8

Table 7: *Summary of test results examining dynamic range of node and edge properties. Results indicate the importance of the dynamic range of edges.*

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/2	30/15	4/4	4/2, .25/.25	3.0	78	10.2 ± 1.9	75.6 ± 18.6
8/2	30/15	4/4	4/4, .25/.25	2.5	98	10.1 ± 1.9	87.1 ± 12.5
8/2	30/15	4/4	8/0, .25/.0	3.3	91	18.7 ± 2.4	46.5 ± 14.6
8/2	30/15	4/4	8/2, .25/.25	3.3	97	11.1 ± 1.8	87.2 ± 13.1
8/2	30/15	4/4	8/4, .25/.25	3.3	99	11.5 ± 1.9	95.2 ± 9.0
8/2	30/15	4/4	16/0, .25/.0	3.3	94	15.5 ± 2.2	70.8 ± 12.6
8/2	30/15	4/4	16/2, .25/.25	3.2	100	12.2 ± 1.6	94.6 ± 7.9
8/2	30/15	4/4	16/4, .25/.25	3.2	100	12.2 ± 1.4	98.1 ± 4.5
8/2	30/15	4/4	16/8, .25/.25	3.1	100	12.2 ± 1.3	98.7 ± 4.0

Table 8: *Summary of test results examining noise in node and edge properties. Results are fairly consistent across the span of noise intensities, except in the extreme cases.*

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	30/15	4/4	4/4, .0/.0	7.0	16	2.8 ± 1.0	57.9 ± 41.9
8/1	30/15	4/4	4/4, .05/.05	6.8	95	7.3 ± 2.6	97.9 ± 6.4
8/1	30/15	4/4	4/4, .10/.10	6.8	94	11.9 ± 2.3	96.8 ± 5.3
8/1	30/15	4/4	4/4, .15/.15	6.8	95	11.9 ± 2.2	96.4 ± 5.8
8/1	30/15	4/4	4/4, .20/.20	6.9	99	11.6 ± 2.3	96.8 ± 5.6
8/1	30/15	4/4	4/4, .25/.25	7.0	99	11.2 ± 2.3	96.9 ± 5.6
8/1	30/15	4/4	4/4, .30/.30	7.0	100	10.6 ± 2.5	96.9 ± 6.7
8/1	30/15	4/4	4/4, .35/.35	6.9	99	9.9 ± 2.7	97.1 ± 7.1
8/1	30/15	4/4	4/4, .40/.40	6.9	97	9.4 ± 2.8	96.6 ± 8.4
8/1	30/15	4/4	4/4, .45/.45	6.7	97	8.3 ± 2.9	97.7 ± 6.0

the range of edge properties needs to be ≥ 4 until the range of node properties becomes ≥ 16. Note the poor results when the edge range drops to 0. Here also, the subgraph size increases above the true size of the contributor. This is consistent with the poor percentage of label matches in these cases.

The results of tests adding noise to node and edge properties were interesting. A gradual decline in performance can be seen in Table 8 as the noise took on larger values, as expected. The low noise cases were suprising. Note the severe drop off in performance with zero noise. In applications with very low property noise, a weighting scheme may have to be introduced to reduce the effect of property comparisons when forming the attendance ratings.

The effects of topological noise shown in Table 9 were not as severe as the cases presented in Table 6 where additional scene graphs were added. Increasing the number of scene graphs has a much greater influence on the scene's topology than the noise introduced here.

Table 9: *Summary of tests examining addition of topological noise. The topological disturbances associated with multiple scene graphs is more severe than the noise in these tests.*

D/S Number of Graphs	D/S Size of Graphs	Excess N/E in Scene	Range, Noise N/E, N/E	Rate Hz	% Correct Selections	Subgraph Size Mean ± Std.Dev.	% Correct Tag Mean ± Std.Dev.
8/1	30/15	0/0	4/4, .25/.25	9.0	98	9.5 ± 2.7	97.8 ± 6.9
8/1	30/15	2/2	4/4, .25/.25	8.3	98	10.5 ± 2.3	97.6 ± 6.3
8/1	30/15	4/4	4/4, .25/.25	7.2	99	11.2 ± 2.3	96.9 ± 5.6
8/1	30/15	6/6	4/4, .25/.25	6.7	98	11.3 ± 2.2	95.8 ± 9.5

5 Concluding Remarks

The objective of the research reported in this paper was to develop a subgraph isomorphism algorithm that will work well in object recognition applications. Current techniques for shape-based recognition are typically quite time consuming and can have undesirable tradeoffs in speed vs. accuracy. DCA can meet the computational bounds associated with an active sensing paradigm.

The DCA approach evaluates evidence describing the likelihood of a node's attendance in another graph. The evidence is based on node and edge properties and on a local connectivity signature. Peak attendance values are identified to form a node-to-node mapping and a global verification step completes the process. After the best matching pairs of nodes are identified, the nodes of one graph are reordered to allow a side by side verification of the graphs' similarity.

Attention has been focused on testing DCA under challenging conditions. Test cases included both topological and feature noise. DCA produced good results in over 95% of test cases and under a wide variety of conditions. These results indicate that DCA has potential in a variety of applications, particularly due to the low reliance on the dynamic range of node and edge properties.

The research reported in this paper is part of a larger effort being pursued in the CVRR laboratory directed towards realization of an integrated Machine Vision system that will also include range data acquisition, surface modeling and graph formation. The Structured Light testbed in our laboratory generates range data at frame rates. The system uses a laser line projector that produces a plane of light. A CCD camera captures images of the laser plane as it intersects objects in the scene. Objects are moved under the sensor using a conveyer belt. This eliminates the need for any moving optical components. An encoder monitors the motion of the conveyer belt. Our current work focus is on rapid methods of range segmentation and surface fitting. This will serve as the front end process to generate input data for DCA.

In addition to the Structured Light-based system, we intend to explore use of the DCA algorithm with other 3-D scene characterization activities pursued in our laboratory [5][39][40]. Good performance of DCA in cases with a dynamic range of 4 for edge properties suggests suitability with applications involving North-South-East-West relative positions, such as stereo matching [38], for example.

6 Acknowledgements

The authors would like to thank Scott M. Thayer for his technical inputs and constructive criticism.

References

[1] O. D. Faugeras. *Three-Dimensional Computer Vision, A Geometric Viewpoint.* MIT Press, London, 1993.

[2] C. H. Chen and A. C. Kak. A robot vision system for recognizing 3-d objects in low-order polynomial time. *IEEE Trans. on System, Man, and Cybernetics*, 19(6):1535–1563, 1989.

[3] D. H. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.

[4] G. Stockman and G. Hu. Sensing 3-d surfaces and edges using a projected grid. In *Proceedings Vision 1986 Conf.*, Detroit, MI, June 3-5 1986. SME.

[5] A. K. Dalmia and M. M. Trivedi. Real-time depth extraction using image streams acquired by a single camera. *Computer Vision and Image Understanding*. To appear 1995.

[6] H. P. Gadagkar and M. M. Trivedi. An integrated system for active exploration using contact and non-contact sensors. In *Proceedings of The IEEE IROS '92 Conference*, Raleigh, NC, July 1992. IEEE.

[7] K. Ikeuchi and M. Hebert. Task Oriented Vision. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2187–2194, Raleigh, NC, 1992.

[8] M. M. Trivedi and C. Chen. Sensor-driven intelligent robotics. *Advances in Computers*, 32:105–148, 1991.

[9] R. T. Chin and C. R. Dyer. Model-based recognition in robot vision. *ACM Computing Surveys*, 18(1):67–108, 1986.

[10] S. Watanabe, editor. *Frontiers of Pattern Recognition*, chapter Some Techniques for Recognizing Structure in Pictures, pages 1–29. Academic Press, New York, 1972.

[11] T-J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-d objects using surface descriptions. *IEEE Trans. on Pattern Anal. Machine Intell.*, 11(11):1140–1157, 1989.

[12] M. Oshima and Y. Shirai. Object recognition using three dimensional information. *IEEE Trans. on Pattern Anal. Machine Intell.*, 5(4):353–361, 1991.

[13] R. Nevatia. *Machine Perception*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[14] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone. A versatile computer-controlled assembly system. *IJCAI*, pages 298–307, 1973.

[15] C. R. Bidlack and M. M. Trivedi. Geometric model based object recognition and localization robotic manipulation tasks. In *Applications of Artificial Intelligence IX Conf.*, pages 270–280, Orlando, April 1991. SPIE.

[16] L. R. Foulds. *Graph Theory Applications*. Springer-Verlag, New York, 1992.

[17] A. V. Aho, J. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison Wesley, Reading, MA, 1974.

[18] J. Bron and H. Kerbosch. Algorithm 457 - finding all cliques in an undirected graph. *Comm. ACM*, 16(575):218–231, 1973.

[19] M. D. Wheeler and K. Ikeuchi. Sensor modeling, markov random fields and robust localization for recognizing partially occluded objects. In *Proceedings of The Image Understanding Workshop*, Wash. D. C.,, April 1993. DARPA.

[20] R. A. Hummel and S. W. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. on Pattern Anal. Machine Intell.*, (5):267–287, 1983.

[21] W-Y. Kim and A. C. Kak. 3-d object recognition using bipartite matching embedded in discrete relaxation. *IEEE Trans. on Pattern Anal. Machine Intell.*, 13(3):224–251, 1991.

[22] N. Ayache and O. D. Faugeras. (hyper): A new approach for the recognition and positioning of two-dimensional objects. *IEEE Trans. on Pattern Anal. Machine Intell.*, 8(1):44–54, 1986.

[23] E. M. Palmer. *Graphical Evolution*. John Wiley and Sons, New York, 1985.

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

[25] R. C. Read. Algorithms in graph theory. *Applications of Graph Theory*, (1), 1979.

[26] J. E. Hopcroft and R. E. Tarjan. A v2 algorithm for determining isomorphism of planar graphs. *Information Processing Letter*, (1), 1971.

[27] J. E. Hopcroft. An n log n algorithm for isomorphism of planar triply connected graphs. Technical Report STAN-CS-71-192, Stanford Computer Science, University of Toronto, Canada, 1971.

[28] D. G. Corneil. Graph isomorphism. Technical Report 18, Dept. of Comp. Sci., 1970.

[29] L. Babai and L. Kucera. Canonical labeling of graphs in linear average time. In *20th Annual IEEE Symposium on Foundations of Comp. Sci.*, pages 39–46, Puerto Rico, 1979. IEEE.

[30] R. M. Karp. *Relations between combinatorics and other parts of mathematics*, chapter Probabalistic Analysis of a canonical numbering algorithm for graphs, pages 365–378. Amer. Math. Soc., Providence, RI, 1979.

[31] J. R. Ullmann. An algorithm for subgraph isomorphism. *Journal ACM*, 23(1):31–42, 1976.

[32] R. J. Wilson and L. W. Beineke, editors. *Applications of Graph Theory,*, chapter Chemical Applications of Graph Theory, pages 328–363. Number 1. Academic Press, London, 1979.

[33] B. Baumgart. A polyhedral representation for computer vision. In *National Computer Conference*, pages 589–596. AFIPS, 1975.

[34] J. O. Berger. *Statistical Decision Theory and Bayesian Analyses*. Springer-Verlag, New York, 1985.

[35] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, 1976.

[36] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.

[37] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of Optical Society of America*, 4(4):629–642, 1987.

[38] S. B. Marapane and M. M. Trivedi. Multi-primitive hierarchical (MPH) stereo analysis. *IEEE Trans. on Pattern Anal. Machine Intell.*, 16:227–240, March 1994.

[39] S. B. Marapane and M. M. Trivedi. Experiments in active vision with real and virtual robot heads. *International Journal of Applied Intelligence, Special issue on Sensor Fusion*, 5(3):237–250, July 1995.

[40] F. W. DePiero and R. L. Kress. Design and in situ calibration of a structured light sensor. In *Proceedings of Intl. Conf. on Intelligent Robotics and Systems*, pages 513–518, Pittsburgh, PA, August 5-9 1995. IEEE/RSJ.