Organization of Relational Models for Scene Analysis

LINDA G. SHAPIRO, SENIOR MEMBER, IEEE, AND ROBERT M. HARALICK, SENIOR MEMBER, IEEE

Abstract—Relational models are commonly used in scene analysis systems. Most such systems are experimental and deal with only a small number of models. Unknown objects to be analyzed are usually sequentially compared to each model. In this paper, we present some ideas for organizing a large database of relational models. We define a simple relational distance measure, prove it is a metric, and using this measure, describe two organizational/access methods: clustering and binary search trees. We illustrate these methods with a set of randomly generated graphs.

Index Terms—Database organization, graphs, relational distance, relational matching, relational models, structural pattern recognition.

I. INTRODUCTION

ANALYSIS of scenes containing complex objects requires knowledge about the structures of the objects being recognized. Relational models, which describe an object in terms of its parts and their interrelationships, have proved to be very useful in the recognition process. In this paper, we discuss the problem of organizing a database of relational models for use in a scene analysis system.

A *relational model* or *relational description* of an entity is a set of relations describing the entity. Relations can perform such functions as listing the global attributes of an entity, listing the parts, and describing relationships among the parts such as connection or parallelness. Relational models have been used in scene analysis for a number of years. The grammars used in syntactic pattern recognition (see Fu [5]) were among the first of the relational models used in picture processing. Graph models and matching procedures were described by Barrow, Ambler, and Burstall [3]. The generalized cylinder models of Agin and Binford [1], Nevatia and Binford [7], and Marr and Nishihara [6] are all relational models. Some other recent examples include the two-dimensional shape models of Shapiro [11], the attributed relational graphs of Tsai and Fu [16], the spatial data structures of Shapiro and Haralick [12], and the relational structures of Cheng and Huang [4].

One important use of relational models in scene analysis is to help identify an unknown object that has been extracted from a scene. A structural description of the unknown object can be constructed and compared to a relational model to determine how similar they are. The matching process is related to the process of finding graph isomorphisms, and its complexity is exponential in the number of model parts. In a real-life application such as robot vision, there is a large number of models in the database. Thus, an unordered linear organization of the database where the unknown object is compared to each model one at a time is highly undesirable. One solution is to compare in parallel, with a processor for each model. A second possibility is to organize the models for fast access based on the unknown object being considered. The purpose of this paper is to investigate this second approach.

Most of the experimental systems that have employed relational models have had so few models that organization was not necessary. However, a few researchers have recognized that the problem exists. Nevatia [7] used an indexing scheme to access models likely to match an unknown object. Each model had a three bit code describing each of its distinguished pieces. Encoded are: 1) connectively (one end or both), 2) type (long or wide), and 3) conical (true or false). Objects with the same code are grouped together, and the correct group is found before full relational matching is attempted. Marr and Nishihara [6] also advocated the use of indexing. They distinguished between indexing clues that can be used before there is a guess at the three-dimensional configuration of the model (for example, connectivity and some length comparisons) and those that cannot. Schneier [10] utilizes a structure called the "graph of models" where common primitives and relations are shared across models and within models. In his matching process, primitives and relation schemata index all models in which they occur, and models index all primitives and relation schemata within them. Thus, there is never any need to match against every one of a library of stored models. Our approach differs from Schneier's in that we are trying to group together entire models that are similar rather than common relationships.

Much of computer science has dealt with the organization of data. Some of the most common organizations and corresponding access functions include: 1) unordered lists with linear search, 2) ordered lists with binary search, 3) tree structures with binary or multiway search, 4) hash tables with hashing functions, 5) hierarchic indexing schemes, 6) relational structures, and 7) clustering. We have already indicated that approach 1) is unacceptable. Approach 2) requires an ordering function which, at present, seems incompatible with the concept of a relational model. Similarly, approaches 4) and 5) require unique keys for each object, and also seem unpromising at present. Approach 6) (the idea of relations that define the relationships of models to each other) is interesting and will be saved for future work. In this paper, we will investigate 3) and 7): search trees and clustering. In Section

II, we discuss some relational distance measures to be used to determine object similarity. Section III describes the clustering approach, and Section IV describes a binary tree approach to grouping.

II. RELATIONAL DISTANCE MEASURES

To simplify our discussion of relational distance, we will work with the simplest relational models—graphs. Each model will consist of a single directed graph, and all graphs will have the same number of nodes. For a generalization to arbitrary N-ary relations with arbitrary domains and for experiments with a more general distance measure, see Shapiro *et al.* [14]. The distance measure of [14] is briefly defined in Section II-B of this paper. Because we wish to state and prove the properties of our distance measure, we will make use of some mathematical notation to make things precise.

A. A Simple Distance Measure

Let G be a graph having $|S|$ nodes. We will represent such a graph by a vector of $|S|^2$ components representing an ordering of the $|S|^2$ possible arcs of G. A component of the vector will have value 1 when the corresponding arc is present in G and 0 otherwise. Henceforth, we will refer to both the graph G and the vector representing it as G. We will also use the characteristic function χ_G defined by

$$\chi_G(s, s') = \begin{cases} 1 & \text{if } (s, s') \in G \\ 0 & \text{otherwise} \end{cases} \quad \text{when convenient.}$$

For any permutation f of S, we define $f(G)$ by

$$f(G) = \{(f(s), f(s')) \mid (s, s') \in G\}.$$

G_i and G_j are *isomorphic* if there exists a permutation f of S such that $f(G_i) = G_j$. We can now define a *distance measure* $D(G_i, G_j)$ for a pair of graphs (G_i, G_j) by

$$D(G_i, G_j) = \min_f \left\| f(G_i) - G_j \right\|$$

where $\| \ \|$ represents any norm and f is a permutation of S. For distance threshold d, we say that G_i and G_j are *related* if $D(G_i, G_j) \leq d$. The fact that D is a metric is proved in Lemma 1.

Lemma 1: The distance measure D is a metric.

Proof:

a) We first show that $D(G_i, G_j) = 0$ if and only if G_i is isomorphic to G_j.

If G_i is isomorphic to G_j and f is the isomorphism, then $f(G_i) = G_j$ and $\left\| f(G_i) - G_j \right\| = 0$. Conversely, $D(G_i, G_j) = 0$ implies that for some f, $\left\| f(G_i) - G_j \right\| = 0$, which is only true if $f(G_i) = G_j$, making f an isomorphism.

b) Next, we show that $D(G_i, G_j) = D(G_j, G_i)$ for every pair of graphs (G_i, G_j).

$$\min_f \left\| f(G_i) - G_j \right\|$$

$$= \min_f \left\| f^{-1}(f(G_i) - G_j) \right\| \quad \text{since } f^{-1} \text{ is a permutation}$$

$$= \min_f \left\| f^{-1}(G_j) - G_i \right\|$$

$$= \min_{f^{-1}} \left\| f^{-1}(G_j) - G_i \right\|.$$

c) Finally, we show that $D(G_i, G_j) \leq D(G_i, G_k) + D(G_k, G_j)$ for any three graphs G_i, G_j, and G_k.

$$D(G_i, G_j) = \min_f \left\| f(G_i) - G_j \right\|$$

$$= \min_f \left\| f(G_i) - G_j + g(G_k) - g(G_k) \right\|$$

for arbitrary permutation g and, in particular, for that g that minimizes $\left\| g(G_k) - G_j \right\|$. Thus,

$$D(G_i, G_j) \leq \min_f \left\| f(G_i) - g(G_k) \right\| + \left\| g(G_k) - G_j \right\|$$

$$= \min_f \left\| f(G_i) - g(G_k) \right\| + \min_h \left\| h(G_k) - G_j \right\|$$

$$= \min_{g^{-1}f} \left\| g^{-1}f(G_i) - G_k \right\| + \min_h \left\| h(G_k) - (G_j) \right\|$$

$$= D(G_i, G_k) + D(G_k, G_j). \qquad \text{Q.E.D.}$$

B. Other Relational Distance Measures

In [14], we defined a more robust distance measure that was a generalization of the measure of Section II-A. The more general measure is defined as follows.

Let $D_1 = \{R_1, \cdots, R_K\}$ and $D_2 = \{S_1, \cdots, S_K\}$ be two relational models. For any N-ary relation $R \subseteq A^N$ and association $f \subseteq A \times B$, we define the *composition* $R \circ f$ by

$$R \circ f = \{(b_1, \cdots, b_N) \in B^N \mid \text{ there exists } (a_1, \cdots, a_N) \in R$$

$$\text{with } (a_n, b_n) \in f, n = 1, \cdots, N\}.$$

The *structural error* of an association $f \subseteq A \times B$ with respect to N-ary relations $R \subseteq A^N$ and $S \subseteq B^N$ is given by

$$E_s(f) = \left| R \circ f - S \right| + \left| S \circ f^{-1} - R \right|.$$

The structural error is a measure of tuples found in R, but not in S or in S, but not in R.

The *completeness error* of an association $f \subseteq A \times B$ with respect to N-ary relations $R \subseteq A^N$ and $S \subseteq B^N$ is given by

$$E_c(f) = \left| S - R \circ f \right| + \left| R - S \circ f^{-1} \right|.$$

The completeness error is a measure of tuples in S that no tuples in R map to and vice versa. When $|A| \neq |B|$, the completeness error reflects relationships among elements of A that have no corresponding elements in B, and the reverse situation.

The combined error is then given by

$$E_{R,S}(f) = c_1 E_s(f) + c_2 E_c(f).$$

The total error of f with respect to descriptions D_1 and D_2 is given by

$$E(f) = \sum_{k=1,K} E_{R_k, S_k}(f).$$

The relational distance between D_1 and D_2 is given by

$$GD(D_1, D_2) = \min_f E(f).$$

In our earlier experiments with the distance measure GD on relational models of three-dimensional objects, we set c_1, the

weight of the structural error, to 4, and c_2, the weight of the completeness error, to 1. The resulting measure was not a metric since it failed to satisfy the triangle inequality. The reason for its failure was the unequal weights.

Sanfeliu [9] has developed a distance measure between relational attributed graphs to be used in the classification of muscle tissue patterns. The measure, which he uses to compare an input graph to a reference graph, is based on the cost of node recognition plus the cost of the operations necessary to transform the input graph to the reference graph. The measure is defined by

$$\min_{\text{all configurations}} \{w_{nr}c_{nr} + w_{ni}c_{ni} + w_{nd}c_{nd} + w_{bi}c_{bi} + w_{bd}c_{bd}\}$$

where w_{nr}, w_{ni}, w_{nd}, w_{bi}, and w_{bd} are weights summing to 1 and c_{nr}, c_{ni}, c_{nd}, c_{bi}, and c_{bd} are the costs of node recognition, node insertion, node deletion, branch insertion, and branch deletion, respectively. In comparison, the general distance measure GD is a bidirectional measure which takes into account the last four of these costs, but makes no allowance for node recognition. Sanfeliu's distance measure is not a metric.

All the relational distance measures must find that mapping from the nodes of one graph to the nodes of a second that minimizes some error function. This is, unfortunately, an NP-complete problem. We have proposed look-ahead operators to help speed up the search [15], and are currently studying several hardware solutions.

III. CLUSTERING AS A METHOD OF GROUPING

In the related area of document storage and retrieval, Salton [8] suggested a clustered file organization in which documents with somewhat similar content descriptions are grouped into clusters. Each cluster is identified by a representative cluster profile which is a weighted vector of terms derived from the term vectors representing the documents in the cluster. A user wishing to locate documents of a certain type submits a query vector of terms that these documents should contain. The query vector is compared to the representative vectors for each cluster and to the individual document vectors in those clusters deemed similar enough. Thus, only a small percentage of the documents in the system need be examined for a given query.

Translating this idea to the domain of relational models, similar models would be grouped into clusters and a cluster representative would be chosen. As unknown object would be compared to the cluster representatives and then to the models in those clusters deemed similar enough.

There are several requirements for carrying out such a procedure. First, a distance measure or similarity measure is needed. Second, a suitable clustering algorithm must be selected. Third, a method of constructing cluster representatives and a way of comparing such a representative to an object model must be defined. Finally, a method of choosing the parameters that decide if an object is close enough to a cluster or to a model must be determined. We have already discussed several distance measures. Using the simple distance measure of Section II-A, we will discuss the other related problems.

A. Defining Cluster Representatives

Suppose we have a set of graphs $\{G_1, \cdots, G_K\}$. We can form a graph relation R_d by

$$R_d = \{(G_i, G_j) | D(G_i, G_j) \leqslant d\}.$$

Clusters of the relation R_d give us a way of grouping the original graphs. Suppose we construct a set of such clusters $\{C_1, \cdots, C_J\}$ using some arbitrary clustering algorithm. We wish to define a representative for each cluster.

For given graph G_i and cluster C, define the total distance of G_i with respect to C by

$$T(G_i, C) = \sum_{G \in C} D(G_i, G).$$

In each cluster C, there is at least one "best graph" $G_b \in C$ where $T(G_b, C) = \min T(G, C)$. If there is more than one such graph, the best graph G_b can be arbitrarily selected from among them. G_b is certainly one representative of cluster C.

Another method of constructing the representative is to count the frequencies of each possible arc and make these frequencies part of the representative. In particular, for each $G \in C$, there is a best permutation mapping $f_G : S \to S$ that was used to calculate $D(G_b, G)$. We can use these f_G's to translate each graph in cluster C so that they can all be considered from the same frame of reference. Let C' be the transformed cluster

$$C' = \{f_G(G) | G \in C\}.$$

Then a representative A of C can be defined by

$$A = \sum_{G \in C'} G / |C'|.$$

Thus, A is an "average" graph, and the components of A are real numbers between 0 and 1. Since A can be represented by a vector, our simple definition of the distance between two graphs also applies to a graph and a cluster representative. That is, if G is a graph and A a cluster representative, then the distance between A and G is given by

$$D(G, A) = \min_f \|f(G) - A\|.$$

We can similarly calculate the distance between two cluster representatives defined in this manner. We can also define a "median" graph M by

$$\chi_M(s, s') = \begin{cases} 1 & \text{if } \sum_{G \in C'} \chi_G(s, s') \geqslant \frac{1}{2}|C'| \\ 0 & \text{if } \sum_{G \in C'} \chi_G(s, s') < \frac{1}{2}|C'|. \end{cases}$$

Again, the median graph can be represented by a vector, and so the distance from a graph G to the median $D(G, M)$ and the distance between two such medians $D(M, M')$ are both well defined.

Now we have defined a best graph G_b, an average graph A, and a median graph M as possible cluster representatives. We need some ways of evaluating the suitability of these representatives. The strongest condition we can ask for is that the system never make any mistakes when identifying unknown objects. In particular, if G is an unknown graph, but is known

to be (isomorphic to) some graph(s) in the database, then the following should hold.

(+) For each cluster C and its representative R_C, there exists a threshold T_C satisfying

1) if $G \in C$, then $D(G, R_C) \leqslant T_C$, and
2) if $G \notin C$, then $D(G, R_C) > T_C$.

Condition (+) is dependent on both the choice of clustering algorithm and the choice of representative. We are not aware of any standard clustering algorithm that operates on a distance matrix and produces clusters and representatives that satisfy (+). In Section III-B, we will discuss some results obtained using a hybrid method that first constructs preliminary clusters, then constructs representatives, and finally tries to satisfy (+). In the remainder of this section, we will look at some weaker conditions and show how they can be satisfied by our average and median graphs.

Let $C = \{G_1, G_2, \cdots, G_N\}$ be a cluster of graphs. Let $G_b \in C$ be the best graph, let A be the average graph, and let M be the median graph. One measure of the utility of representative R is the distance sum

$$(++) \quad \sum_{n=1, N} D(G_n, R)^P = \sum_n \min_{f_n} \|f_n(G_n) - R\|^P$$

for some positive integer P. Unfortunately (++) does not suggest a construction for R, and there is an exponentially explosive number of possibilities to be searched. However, we can find a reasonable upper bound for (++) and determine the best R with respect to the upper bound.

Lemma 2: When $\| \ \|$ is the two-norm, $\sum_n D(G_n, R)^2$ is bounded by an expression that is minimal when $R = A$.

Proof: We know that G_b was selected to be that member of C which, along with its corresponding $f_{G_1}, f_{G_2}, \cdots, f_{G_N}$, *minimizes*

$$\sum_n \|f_{G_n}(G_n) - G_b\|.$$

If R is the best representative of C, then clearly

$$\sum_n \min_{f_n} \|f_n(G_n) - R\|^2 \leqslant \sum_n \|f_{G_n}(G_n) - R\|^2.$$

Consider the two-norm. Then

$$\sum_{n=1}^{N} \|f_{G_n}(G_n) - R\|_2^2$$

$$= \sum_{n=1}^{N} \left(\sum_{k=1}^{|S|^2} (f_{G_n}(G_{n,k}) - R_k)^2 \right)$$

where $G_{n,k}$ is the kth component of vector G_n.

$$= \sum_{k=1}^{|S|^2} \left(\sum_{n=1}^{N} (f_{G_n}(G_{n,k}) - R_k)^2 \right).$$

Let $y_n = f_{G_n}(G_{n,k})$. We know that $\sum_{n=1}^{N} (y_n - R_k)^2$ is minimized when $R_k = 1/N \sum_{n=1}^{N} y_n$. But, if $R_k = 1/N \sum_{n=1}^{N} \cdot f_{G_n}(G_{n,k})$, then $R = A$ by definition of the average graph!

Q.E.D.

Lemma 3: When $\| \ \|$ is the one-norm, $\sum_n D(G_n, R)$ is bounded by an expression that is minimal when $R = M$.

Proof: Again we note that for best representative R,

$$\sum_n \min_{f_n} \|f_n(G_n) - R\| \leqslant \sum_n \|f_{G_n}(G_n) - R\|.$$

For the one-norm,

$$\sum_{n=1}^{N} \|f_{G_n}(G_n) - R\|$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{|S|^2} |f_{G_n}(G_{n,k}) - R_k|$$

$$= \sum_{k=1}^{|S|^2} \left(\sum_{n=1}^{N} |f_{G_n}(G_{n,k}) - R_k| \right).$$

Again, let $y_n = f_{G_n}(G_{n,k})$. Then $\sum_{n=1}^{N} |y_n - R_k|$ is minimized when $R_k = \text{median} \{y_n\}$. But, if $R_k = \text{median} \{f_{G_n}(G_{n,k})\}$, then $R = M$ by definition of the median graph!

B. A Hybrid Clustering Scheme and Some Examples

We considered several classical clustering algorithms for use in this study. Algorithms that require the objects being clustered to be points in n-space were omitted from consideration since in this problem, our objects are, at best, graphs, and we know only their pairwise distances, not their locations in n-space. Clique finding algorithms were discarded because for any distance threshold d and corresponding graph relation R_d, there would be many small cliques, and the access process would not be fast enough. Single-link methods, where the clusters are the connected components of R_d, were avoided because the resultant clusters could have very few interconnections among their members. Hierarchic methods seemed reasonable, but tended not to include the notion of a representative. The isodata method, which iteratively moves objects around in order to minimize the sums of the squared distances between objects and their cluster centroids, comes closest in spirit, but is not guaranteed to satisfy (+).

In [13], we presented a graph-theoretic clustering algorithm for clustering the points of a shape into near-convex pieces. The graph-theoretic algorithm allows the specification of parameters that determine: 1) how compact or close knit a cluster must be, 2) how much two clusters must intersect before they can be merged, and 3) the minimum cluster size. These are all properties that we would like to specify for our clusters of models. Since the clusters produced by the graph-theoretic method do not satisfy (+), we developed the following hybrid method.

1) Select a distance threshold d.
2) Apply the graph-theoretic clustering algorithm to R_d.
3) For each cluster C
   a) compute the representative R_C
   b) compute

$$\text{maxdistin} = \max_{G \in C} D(G, R_C) \quad \text{and}$$

$$\text{mindistout} = \min_{G \notin C} D(G, R_C).$$

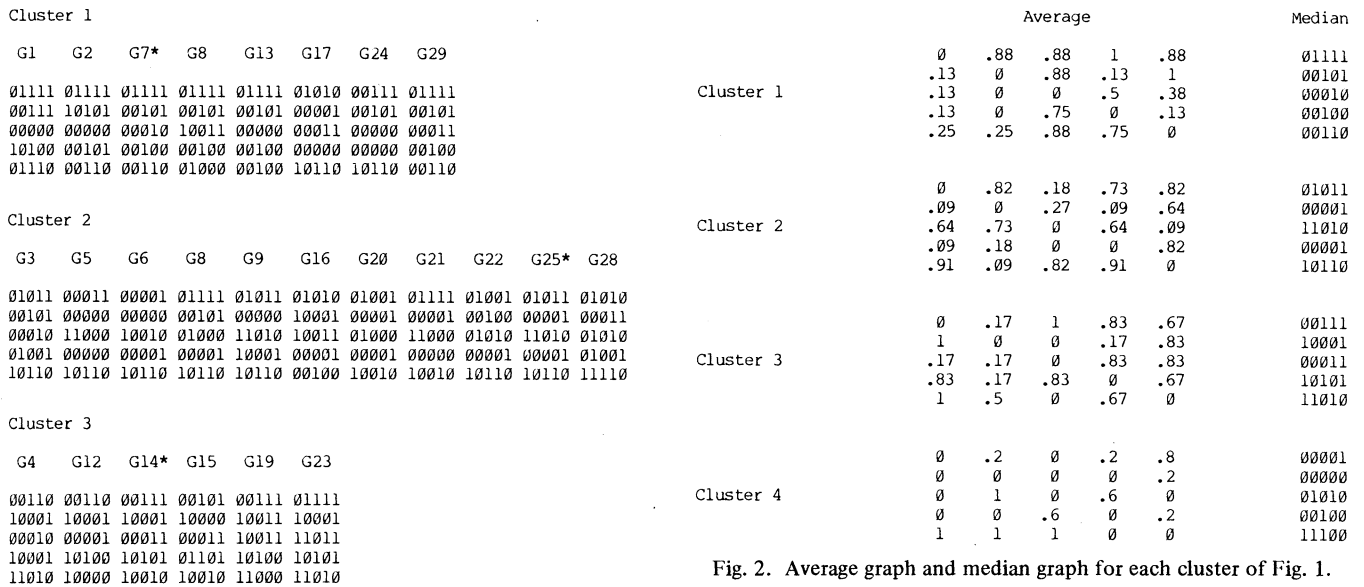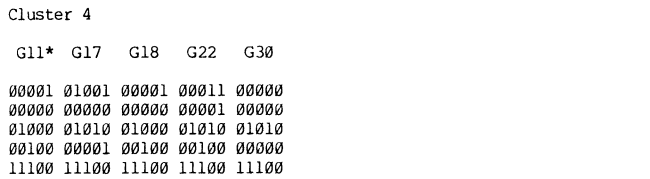   c) if maxdistin < mindistout, leave C alone
   d) otherwise construct

Cluster 1

G1 G2 G7* G8 G13 G17 G24 G29

```
01111 01111 01111 01111 01111 01010 00111 01111
00111 10101 00101 00101 00101 00001 00101 00101
00000 00000 00010 10011 00000 00011 00000 00011
10100 00101 00100 00100 00100 00000 00000 00100
01110 00110 00110 01000 00100 10110 10110 00110
```

Cluster 2

G3 G5 G6 G8 G9 G16 G20 G21 G22 G25* G28

```
01011 00011 00001 01111 01011 01010 01001 01111 01001 01011 01010
00101 00000 00000 00101 00000 10001 00001 00001 00100 00001 00011
00010 11000 10010 01000 11010 01000 01000 11000 01010 11010 01010
01001 00000 00001 00001 10001 00001 00001 00000 00001 00001 01001
10110 10110 10110 10110 10110 00100 10010 10010 10110 10110 11110
```

Cluster 3

G4 G12 G14* G15 G19 G23

```
00110 00110 00111 00101 00111 01111
10001 10001 10001 10000 10011 10001
00010 00001 00011 00011 10011 11011
10001 10100 10101 01101 10100 10101
11010 10000 10010 10010 11000 11010
```

Cluster 4

G11* G17 G18 G22 G30

```
00001 01001 00001 00011 00000
00000 00000 00000 00001 00000
01000 01010 01000 01010 01010
00100 00001 00100 00100 00000
11100 11100 11100 11100 11100
```

Fig. 1. A set of initial clusters of 30 graphs using a graph-theoretic clustering algorithm. The best graph in each cluster is indicated with an asterisk (*), and each graph in a given cluster has been translated according to its mapping with the best graph. The graphs are shown in adjacency matrix form rather than pictorially to allow the reader to easily compare the arcs present in one graph to the arcs present in another.

$$S = \{G \mid \text{mindistout} \leqslant D(G, R_C) \leqslant \text{maxdistin}\} \quad \text{and}$$

$$T = \{D(G, R_C) \mid G \in S\}$$

e) for each threshold $t \in T$, define C_t by

$$C_t = \{G \mid D(G, R_C) \leqslant t\}$$

f) let T_c be the smallest $t \in T$ that minimizes

$$|\#C - \#C_t|$$

g) output the new cluster

$$C' = \{G \mid D(G, R_C) \leqslant T_C\}.$$

Essentially, the hybrid method uses the graph-theoretic clustering algorithm to obtain initial clusters. If a cluster satisfies (+), it is left alone; otherwise, objects are moved in and out depending on their distance to the representative where the distance threshold is chosen to minimize change in cluster size. Although the clusters may change, the representatives are *not* recomputed, since recomputing them can cause (+) to not be satisfied.

Example: We generated 30 random graphs on five nodes to experiment with. The distance threshold d was selected so that between $\frac{1}{4}$ and $\frac{1}{3}$ of the possible edges in the graph relation R_d were present. With the minimum compactness threshold set to 0.9 and the minimum overlap required for merging set at 0.6, four clusters were obtained. The clusters are shown in Fig. 1, with the best graph indicated by an

	Average				Median
	Ø .88 .88 1 .88				01111
	.13 Ø .88 .13 1				00101
Cluster 1	.13 Ø Ø .5 .38				00010
	.13 Ø .75 Ø .13				00100
	.25 .25 .88 .75 Ø				00110
	Ø .82 .18 .73 .82				01011
	.09 Ø .27 .09 .64				00001
Cluster 2	.64 .73 Ø .64 .09				11010
	.09 .18 Ø Ø .82				00001
	.91 .09 .82 .91 Ø				10110
	Ø .17 1 .83 .67				00111
	1 Ø Ø .17 .83				10001
Cluster 3	.17 .17 Ø .83 .83				00011
	.83 .17 .83 Ø .67				10101
	1 .5 Ø .67 Ø				11010
	Ø .2 Ø .2 .8				00001
	Ø Ø Ø Ø .2				00000
Cluster 4	Ø 1 Ø .6 Ø				01010
	Ø Ø .6 Ø .2				00100
	1 1 1 Ø Ø				11100

Fig. 2. Average graph and median graph for each cluster of Fig. 1.

Representative = Best Graph

Size Change	Cluster
Ø (2 out, 2 in)	1) 1, 2, 7, 13, 24, 27, 28, 29
3 (1 out, 4 in)	2) 3, 5, 6, 8, 9, 10, 17, 20, 21, 22, 25, 26, 28, 29
1 (0 out, 1 in)	3) 4, 12, 14, 15, 19, 23, 24
2 (1 out, 3 in)	4) 6, 11, 16, 18, 21, 22, 30

Representative = Average Graph

Size Change	Cluster
1 (2 out, 3 in)	1) 1, 2, 5, 7, 13, 24, 28, 29, 30
0 (2 out, 2 in)	2) 3, 5, 6, 8, 9, 10, 17, 20, 21, 22, 25
0 (1 out, 1 in)	3) 4, 12, 14, 19, 23, 25
0 (0 out, 0 in)	4) 11, 17, 18, 22, 30
1	5) 15
1	6) 16
1	7) 26
1	8) 27

Representative = Median Graph

Size Change	Cluster
Ø (2 out, 2 in)	1) 1, 2, 7, 13, 24, 27, 28, 29
3 (1 out, 4 in)	2) 3, 5, 6, 8, 9, 10, 17, 20, 21, 22, 25, 26, 28, 29
3 (2 out, 5 in)	3) 3, 4, 8, 9, 10, 14, 19, 23, 25
Ø (1 out, 1 in)	4) 11, 16, 18, 22, 30
1	5) 12
1	6) 15

Fig. 3. Final clusters and change in cluster sizes for each type of representative.

asterisk (*), and each graph represented by an adjacency matrix and translated according to its mapping with the best graph. Fig. 2 gives the average graph and median graph for each cluster. Fig. 3 shows the final clusters and change in cluster sizes for each of the three types of representatives.

It is interesting to note that in this experiment, although the clusters have changed some in composition, those graphs that were the best graphs of the original four clusters are still the best graphs in the new clusters obtained: 1) when the representative was the best graph, 2) when the representative was the average graph, and 3) in three out of four cases when the representative was the median graph. Although the average graph and median graph are almost always going to change as the clusters change, the best graph is much more stable. In our experiments where the representative graph was the best graph, the representative remained the best graph in 72 percent of the cases.

IV. Binary Search Trees as a Method of Grouping

Tree structured tables are commonly used in both internal and external storage to organize data objects that have numeric or alphabetic keys. The simplest version, a binary search tree, consists of a binary tree with the following properties.

1) If the key of an unknown object matches the key of the root, the object is in the root of the tree.

2) If the key of the unknown object is less than the key of the root, the object is in the left subtree (if it is in the tree at all).

3) If the key of the unknown object is greater than the key of the root, the object is in the right subtree (if it is in the tree at all).

4) The left and right subtrees are both binary search trees.

If a binary search tree is balanced (the left and right subtrees differ in depth by no more than one level and both are balanced trees themselves), then a tree containing n objects will take $O(\log_2 n)$ comparisons to search. There are known algorithms for balancing binary search trees [2].

Organizing relational models into binary search trees must be handled differently since there are no associated keys. Instead, given a database of models, we can define a method for splitting them into two groups at each node of the tree. The idea is, given a set of graphs, to choose two graphs A and B so that some of the other graphs are closer to A and the rest are closer to B. Of course, we want to choose A and B in an intelligent manner.

Let S be a set of graphs, $A \in S$, and $B \in S$. Let

$$P_A = \{G \in S \mid D(G,A) \leqslant D(G,B)\}$$

and

$$P_B = \{G \in S \mid D(G,B) < D(G,A)\}.$$

To choose A and B intelligently, we would like to minimize the sum of the distances of graphs in P_A to A plus the sum of the distances of graphs in P_B to B. We have

$$\min_{A,B} \left\{ \sum_{G \in P_A} D(G,A) + \sum_{G \in P_B} D(G,B) \right\}$$

$$= \min_{A,B} \left\{ \sum_{G \mid D(G,A) \leqslant D(G,B)} D(G,A) \right.$$

$$\left. + \sum_{G \mid D(G,B) < D(G,A)} D(G,B) \right\}$$

$$= \min_{A,B} \left\{ \sum_{G \in S} \min \{D(G,A), D(G,B)\} \right\}.$$

This last quantity is easily computable and gives rise to the following tree creation algorithm.

Step 0) Construct a root node.

Step 1) Given set S, choose graphs A and B, graph-number $(A) <$ graph-number (B), to minimize

$$\sum_{G \in S} \min \{D(G,A), D(G,B)\}.$$

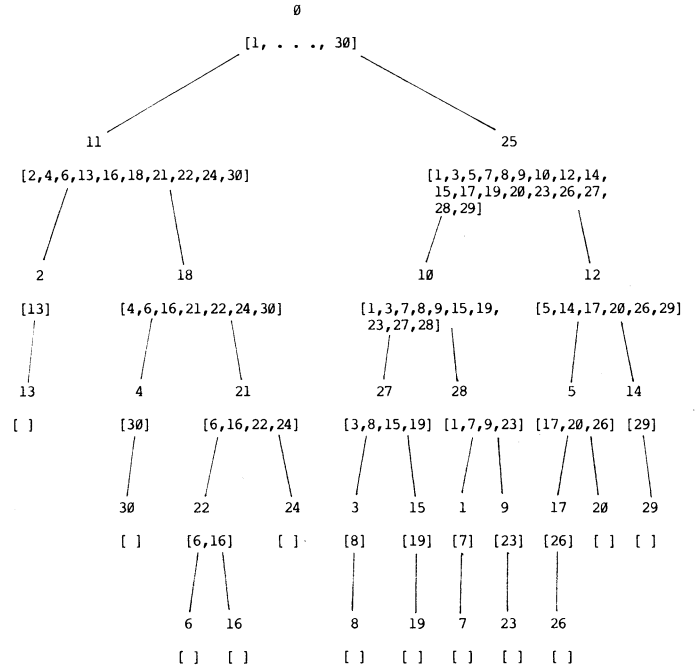Make A the left subtree of the root and B the right subtree.

Fig. 4. Binary search tree obtained for 30 random graphs.

Step 2) Construct P_A and P_B.

Step 3) If P_A is large enough to split, recursively execute Steps 1), 2), and 3) for root A and set $S = P_A$. If P_B is large enough to split, recursively execute Steps 1), 2), and 3) for root B and set $S = P_B$.

Fig. 4 shows the tree obtained for the 30 random graphs used in the experiments of Section III. The tree is not entirely balanced, but does have $5 = [\log_2 30]$ levels of comparison. To identify an unknown graph as either isomorphic to a graph in the tree or not in the tree, the unknown graph is compared to the graphs at the left child of the root and the right child of the root. If it is isomorphic to either, the procedure terminates. Otherwise, it is called recursively for the left and right subtrees. Thus, if the tree is balanced, there are $2 \log_2 n$ comparisons for n graphs. We have not investigated any possible balancing algorithms, but we do note that for relational models that are expected to fall into natural clusters, the trees should be balanced or close to balanced.

What if we wish to determine that graph in the tree that is (or the top K graphs in the tree that are) most similar to an unknown graph? Consider, for example, a database consisting of the first ten random graphs. The tree creation algorithm chooses graphs 2 and 9 as A and B with the left subtree containing 1, 2, 4, 7, and 8 and the right subtree containing 3, 5, 6, 9, and 10. Now suppose that graph 11 is the unknown graph. Of the first ten graphs, graph 11 is closest to graph 6. But if compared only to graphs 2 and 9, graph 11 is closer to graph 2. Thus, the procedure would choose the left branch of the tree and never encounter graph 6. It is clear from this example that the binary tree search may not perform perfectly for inexact matches. In these cases, we must accept approximate answers. In situations where an isomorphism or exact match can be expected, the search gives the exact correct solution.

V. Conclusions

We have discussed two methods of organization for relational models, both based on a relational distance measure, which we have shown is a metric. The clustering method groups models into clusters of similar ones, and represents each cluster by a cluster representative. An unknown object is compared to the cluster representatives and then to the models in those clusters deemed similar enough. We defined three possible cluster representatives: the best graph, the average graph, and the median graph. Although we have not shown any of the three to be superior, our theoretical results show that the sum of the distances of the graphs in a cluster to the representative is bounded by a quantity that is minimized by the average graph using the two-norm and the median graph using the one-norm. Our experimental results, using a hybrid clustering method, showed that the best graph is a more stable representative than the other two; as the composition of the clusters change, the best graph remains constant longer.

Our hybrid scheme satisfies the criteria that for each cluster C, there is a threshold T_C satisfying
1) if $G \in C$, then $D(G, R_C) \leqslant T_C$, and
2) if $G \notin C$, then $D(G, R_C) > T_C$.

Thus, for exact matching, an unknown object that is actually in a particular cluster will always "match" the representative of that cluster, and will not match the representatives of any clusters that it is not in. For inexact matching, if an unknown object U is not identical to any model in the database, but is closest to some model M in some cluster C with representative R_C, then since D is a metric, we have

$$D(U, R_C) \leqslant D(U, M) + D(M, R_C).$$

Since $D(M, R_C) \leqslant T_C$, if $D(U, M) \leqslant \in$ for some small real number $\in$, then $D(U, R_C) \leqslant T_C + \in$. In practice, we can choose such an $\in$ and increase each threshold by $\in$ so that any object that is within distance $\in$ of a stored model will match the representative of its cluster.

As far as savings on matching time, the results depend on the number of final clusters and degree of similarity of clusters. If m models are grouped into n clusters of m/n models each, then finding the closest model to an unknown object would require $n + cm/n$ matches if the unknown object is judged similar enough to c cluster representatives. When $c = 1$, then the number of matches is minimized for $n = m^{1/2}$ clusters. In this case, exactly $2m^{1/2}$ matches take place. For example, if there are 400 models grouped into 20 clusters of 20 objects each, and if an unknown object is judged similar to only one cluster representative, then 40 matches would take place instead of the 400 required with no grouping. If, however, the unknown object is judged similar to 10 cluster representatives, then 220 matches would be required. Thus, it is important to be able to cluster the models into approximately as many groups as the square root of the number of models, with approximately the same number of models in each group, *and* for the clusters (and therefore the representatives) to be sufficiently different in order for the scheme to work optimally.

The second organization is a binary search tree whose nodes are relational models. For exact matching, an unknown object is compared to the left and right children of the root. If it is identical to either, the process stops. If not, it continues recursively down the subtree to whose root model it is most similar. If the tree is balanced and contains m objects, then at most $2 * \log_2 m$ matches are required. Thus, for 400 models, a maximum of 18 comparisons would be made. For inexact matching, the procedure is not guaranteed to find that model that is closest to an unknown object, and therefore, a clustered organization might be preferred.

Many of the results of this paper are based on the distance measure being a metric. In the simple distance measure used for our grouping experiments, the mapping was single valued, one-one, and onto. Our definition of the general distance measure GD allowed the mapping to be an arbitrary relation, and assigned possibly different weights to the structural and completeness portions of the error function. When these weights are different, the distance measure may not be a metric. When the mapping is not one-one and single valued, the distance measure may not be a metric. We have seen counterexamples in both cases. This brings up the question of how useful are those restricted distance measures that are metrics.

One useful measure that is a metric is the one obtained by setting c_1, the weight on the structural error, to 1; setting c_2, the weight on the completeness error, to 0; and requiring the mapping to be single values and one-one. This gives us

$$E_1(f) = \sum_{k=1}^{K} \left| R_k \circ f - S_k \right| + \left| S_k \circ f^{-1} - R_k \right|.$$

The distance measure based on this error function E_1 is useful in making preliminary selections of relational models at a rough top level of image analysis. Other techniques using weights on parts and more precise information can be invoked after the ruling-out process. A version of this error function

$$E_2(f) = \sum_{k=1}^{K} \left| R_k \circ f - S_k \right|$$

was used successfully in many of our experiments on two-dimensional shape matching [11]. Note also that the Euclidean distance measure, used commonly in pattern recognition, is also an unweighted measure.

This paper represents our preliminary work in this area. Neither method presented here has been shown to be superior to all others. We expect our future research to further study these and other organizations and access methods which will be needed for any practical computer vision system, and to refine the concept of the cluster representative.

REFERENCES

[1] G. J. Agin and T. O. Binford, "Computer description of curved objects," in *Proc. 3rd Int. Joint Conf. Artificial Intell.*, Stanford Univ., Stanford, CA, 1973, pp. 629–635.

[2] J. L. Baer and B. Schwab, "A comparison of tree-balancing algorithms," *Commun. Ass. Comput. Mach.*, vol. 20, pp. 322–330, May 1977.

[3] H. G. Barrow, A. P. Ambler, and R. M. Burstall, "Some techniques for recognizing structure in pictures," in *Frontiers of Pattern Recognition*, S. Watanabe, Ed. New York: Academic, 1972, pp. 1–29.

[4] J. K. Cheng and T. S. Huang, "Image recognition by matching relational structures," in *Proc. IEEE Conf. Pattern Recognition and Image Processing*, Dallas, TX, Aug. 1981, pp. 542–547.

[5] K. S. Fu, *Syntactic Methods in Pattern Recognition*. New York: Academic, 1974.

[6] D. Marr and H. K. Nishihara, "Spatial disposition of axes in a generalized cylinder representation of objects that do not encompass the viewer," M.I.T. AI Lab., Memo. 341, Dec. 1975.

[7] R. Nevatia and T. O. Binford, "Description and recognition of curved objects," *Artificial Intell.*, vol. 8, pp. 77–90, 1977.

[8] G. Salton, *The SMART System–Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[9] A. Sanfeliu, "An application of a distance measure between graphs to the analysis of muscle tissue patterns," School Elec. Eng., Purdue Univ., W. Lafayette, IN, TR-EE 81-15, June 1981.

[10] M. Schneier, "A compact relational representation," presented at the Workshop on the Representation of Three-Dimensional Objects, R. Bajcsy, Director, Univ. Pennsylvania, Philadelphia, May 1979.

[11] L. G. Shapiro, "A structural model of shape," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 111–126, Mar. 1980.

[12] L. G. Shapiro and R. M. Haralick, "A general spatial data structure," in *Proc. IEEE Conf. Pattern Recognition and Image Processing*, June 1978, pp. 238–249.

[13] ——, "Decomposition of two-dimensional shapes by graph-theoretic clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 10–20, Jan. 1979.

[14] L. G. Shapiro, J. D. Moriarty, R. M. Haralick, and P. G. Mulgaonkar, "Matching three-dimensional objects using a relational paradigm," Dep. Comput. Sci., Virginia Polytechnic Inst., Blacksburg, Tech. Rep. C5800140-R, Dec. 1980.

[15] L. G. Shapiro and R. M. Haralick, "Structural descriptions and inexact matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 504–519, Sept. 1981.

[16] W. H. Tsai and K. S. Fu, "Error correcting isomorphisms of attributed relational graphs for pattern analysis," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, Dec. 1979.
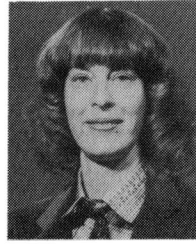
Linda G. Shapiro (S'74–M'74–SM'81) was born in Chicago, IL, in 1949. She received the B.S. degree in mathematics from the University of Illinois, Urbana, in 1970, and the M.S. and Ph.D. degrees in computer science from the University of Iowa, Iowa City, in 1972 and 1974, respectively.

She was an Assistant Professor of Computer Science at Kansas State University, Manhattan, from 1974 to 1978. She is currently an Associate Professor at Virginia Polytechnic Institute and State University, Blacksburg. Her research interests include scene analysis, pattern recognition, spatial information systems, computer graphics, and data structures. She has completed an undergraduate textbook on data structures with R. Baron.

Dr. Shapiro is a senior member of the IEEE Computer Society, and a member of the Association for Computing Machinery, the Pattern Recognition Society, and the American Association for Artificial Intelligence. She is currently Co-Editor of the *IEEE Technical Committee on Machine Intelligence and Pattern Analysis Newsletter*.

Robert M. Haralick (S'62–M'69–SM'76) received the Ph.D. degree from the University of Kansas, Lawrence, in 1969.

He is presently a Professor in the Departments of Electrical Engineering and Computer Science and Director of the Spatial Data Analysis Laboratory, Virginia Polytechnic Institute and State University, Blacksburg. He has done research in pattern recognition, multi-image processing, remote sensing, texture analysis, image data compression, clustering, artificial intelligence, and general system theory.

Dr. Haralick is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, *Pattern Recognition*, and *Computer Graphics and Image Processing*. He is on the Editorial Board of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and is the Chairman of IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence. He is the Computer Vision and Image Processing Editor of *Communications of the ACM*. He is also a member of Association for Computing Machinery, the Pattern Recognition Society, and the Society for General System Research.