S17-63
98
188137

A Database/Knowledge Structure for a Robotics Vision System

D.W. Dearholt and N.N. Gonzales
New Mexico State University
Las Cruces, NM 88003

N X 73657

ABSTRACT

Desirable properties of robotics vision database systems are given, and structures which possess properties appropriate for some aspects of such database systems are examined. Included in the structures discussed is a family of networks in which link membership is determined by measures of proximity between pairs of the entities stored in the database. This type of network is shown to have properties which guarantee that the search for a matching feature vector is monotonic. That is, the database can be searched with no backtracking, if there is a feature vector in the database which matches the feature vector of the external entity which is to be identified. The construction of the database is discussed, and the search procedure is presented. A section on the support provided by the database for description of the decision-making processes and the search path is also included.

I. Introduction

Several structures have been proposed which have properties desirable for use in a robotics vision database system. Some of these structures are examined in this paper, including a new family of networks in which link membership is determined by measures of proximity between pairs of entities represented by nodes in the database and the triangle inequality.

Suitable domains for the database structures considered here are those for which the entities to be stored are describable by a few feature vectors, e.g. color, or shape using Fourier descriptors. We consider the following to be desirable properties for such a database:

1. The database system should support efficient search, so that the feature vector which provides the best match to some external entity can be found quickly;

2. The structure should support classification, so that external entities can be named and higher levels of abstraction are supported;

3. The structure should support a modest level of self-description, so that entities along a search path provide information about the template-matching and classification decisions being made;

4. The neighbors of an entity should reflect consistency with respect to class, so that entities within a given mode of a class should be stored in a manner that reflects the associations and enhances retrieval;

5. Learning, considered to be the addition of entities to the database system, should be done in such a way that the previous properties are preserved.

While this list is too demanding to be well satisfied by any structure known to the authors, it is informative to explore the limitations of the various structures. The paradigm of preprocessing the data (entities to be represented in the database) so that search is facilitated is an important concept, selected by Dobkin and Lipton [1], Bentley and Friedman [2], and Bentley and Maurer [3]. Dobkin and Lipton [1] extended binary search to multidimensional search problems, and could efficiently respond to queries which included the nearest-neighbor problem. Bentley and his colleagues specialized in range searching queries, in which it is desired to find all entries in the database in which each component of the feature vectors is within some given range.

The k-d trees and range trees discussed by Bentley and Friedman [2] are interesting structures, designed specifically for range queries. Range queries are, of course, extremely useful for a wide variety of applications, and can be used as a sort of de facto classification scheme. Other papers presenting results of studies using k-d trees or range trees are Bentley and Maurer [3] and Chang and Fu [4]. These methods suffer the limitations imposed by any hierarchical scheme in terms of descriptive power and classification strategies, since abstractions are necessarily limited to those representable by hierarchies. Category information is, in fact, not a strong point of these methods, since only hierarchical neighbors of matching entities are readily available.

Kalvin, et al. [5], have developed a technique for pattern recognition based on geometric descriptions of the boundaries of objects. This technique is designed specifically for identification of overlapping and partially occluded objects. Attributes used for matching are derived from geometric features of segments of the boundaries of a set of objects. The search procedure is based on geometric hashing of objects in 5-space, where the coordinates are attribute values obtained during preprocessing of the data. A set of candidate matches (models) is selected on the basis of frequency of inclusion in the hypercubes of 5-space in which the unknown's attributes place it. A match rate is computed for each candidate match (model) based on the ratio of the number of match points for the model and the number of possible match points for a particular unknown. The database organization provides for efficient search for the specific application for which it was intended; however, no categorical information is provided and varying levels of abstraction are not supported.

Pathfinder networks share some attributes and objectives with memory-based reasoning, as discussed in Stanfill and Waltz [6]. Both paradigms make use of feature values to compute distance or dissimilarity functions, search memory for best match(es), and classify entities; and both paradigms share the philosophy of classifying entities by direct reference to memory. Pathfinder networks, however, are organized algorithmically so that associations are explicit, which results in categories being evident in the link structure. The search procedure described in Section III of this paper guarantees that search is monotonic (i.e., there is no backtracking) if there is a match in the database, so that search is not exhaustive, as it is in the scheme used for the Connection Machine described in Stanfill and Waltz [6]. Furthermore, the memory-based reasoning paradigm does not support descriptions of the search path and the decisions made which contribute to classification.

"Description" is used here to mean that the salient feature values of entities along the search path, the search path itself, the reasons for selecting the search path, and the neighborhood of the goal node, are available for a summary of the entire process. Section IV of this paper is devoted to a discussion of descriptive processes supported by the database developed here.

The organization of the entities in the database is based on a network model of semantic memory in humans. This model is called Pathfinder, and the properties of Pathfinder networks (PFNETs), previously called link-weighted networks (LWNs), are described in Dearholt, Schvaneveldt, and Durso [7] and in Schvaneveldt, Dearholt, and Durso [8]. Earlier work on databases intended for vision systems is described in Dearholt, Gonzales, Ellington, and Phillips [9], and in Dearholt, Gonzales, and Kirpekar [10]. These database schemas also used Pathfinder networks. Motivations for the database described in the latter paper and extended in this paper include (1) increased efficiency in the search process by eliminating backtracking; (2) the efficient determination whether or not a given entity is represented in the database; (3) organization of the database so that similar entities are clustered together; (4) provision for category-level information by means of the clustering inherent in Pathfinder networks; and (5) support of description of the search and classification processes.

The efficiency in the search process is accomplished at the time the network is generated by establishing links which provide a path between any two nodes so that the relative distance between nodes, as the path is traversed, is monotonically decreasing. Then, if a feature vector representing an external entity is presented to the database as a query, the corresponding node can be found rapidly from any node in the database. Heuristics to improve the initial node of the search can further improve the search efficiency. The determination that a given entity is not in the database follows from the procedure to be described in Section III. The clustering of similar entities and the resultant category-level information is a feature of the PFNETs to be described in Section II. These features provide a basis for the support of the description processes to be discussed in Section IV.

II. The Generation of a Pathfinder Network Database

Because PFNETs provide for clustering of similar entities, they seem to be a good paradigm for a database organization; indeed, their original purpose in modeling the semantic memory of humans provides for a database of concepts. Thus it seemed natural to extend PFNETs to feature-based applications in which each entity is described by a feature vector. Vision systems used for pattern recognition and image analysis are well served by such databases, and the properties of PFNETs support search, classification, and description, as mentioned previously. Our first effort in this direction was the database for insect identification (Dearholt, et al., [9]), in which PFNETs were used to organize the database. PFNET(∞, n-1) was used because it is the PFNET with fewest links, but there was no very effective search procedure associated with this PFNET organization. Our second effort (Dearholt, et al., [10]) justified and described the construction of the PFNETs which guarantee monotonic search. The purposes of this paper are to list the desirable properties of a vision database for robotics, and to describe the results of our work with Pathfinder networks relevant to these properties. It should thus be regarded as a progress report of our project, written for the purposes of the workshop.

The database schema we will present relies on the Pathfinder network model as a means of organizing the entities in the database. Development of this model (previously called link-weighted networks) has been ongoing for the past six years. Pathfinder yields network structures (PFNETs) for a set of entities, given estimates or measures of the pair-wise distances between the entities. The original purpose of Pathfinder models was to model human semantic memory, so that the estimates of distances were typically estimates of similarity. For the database schema discussed here, however, the entities are each represented by a feature vector, and distances between pairs of entities are presented to the system as a weight matrix. If the weight matrix is symmetric, then the PFNETs derived from it are undirected, whereas an asymmetric weight matrix yields directed PFNETs.

A maximally connected network contains a link between every pair of nodes, so that each weight in the weight matrix is represented by a link. Such a network contains all the original information in the data, but it provides very little information about the structure underlying the data. Pathfinder includes only the links necessary to preserve geodetic paths, thus facilitating analysis and interpretation. Two parameters are required for the complete definition of a PFNET for a particular weight matrix. These are the r-metric and the q-parameter. The r-metric is the value of the Minkowski parameter which is used to compute the distance between nodes in the network which are not directly linked. That is, the weights along the path used to compute distance are individually taken to the r power, these values are summed, and the rth root of the resulting sum is the distance. In general,

$$d(N_i,N_j)=[\sum_{k=1}^{p} w_k{}^r]^{1/r}$$

where the w_k are the weights along the path between N_i and N_j. The r-metric parameter may take on values from 1 to ∞. The q-parameter determines the maximum number of links in paths considered to connect two nodes. For example, for $q = 2$, paths having more than two links are not considered in the preservation of minimum-distance (geodetic) paths in the PFNET.

PFNETs possess properties of inclusion which vary as values of the r-metric and q-parameter change (Dearholt, et al., [7]). Briefly, for a particular weight matrix, PFNET2 is a spanning subgraph of PFNET1 if and only if the r-metric used for PFNET1 is less than or equal to the r-metric used for PFNET2, provided that the q-parameter is held constant. In addition, for a particular weight matrix, PFNET2 is a spanning subgraph of PFNET1 if and only if the value of the q-parameter used for PFNET1 is less than or equal to the value of the q-parameter used for PFNET2, provided that the r-metric is held constant. The PFNET generated with $r = \infty$ and q = the-number-of-nodes-less-one always has the minimum number of links and is the union of all minimum cost spanning trees of an undirected PFNET.

As a PFNET is constructed, precedence is given to small weight values, because they represent the strongest associations. During each stage of development of an undirected PFNET, the complete set of nodes is partitioned into connected subgraphs, called node sublists. When a link is added which joins nodes in different sublists, the two sublists are merged to form a single node sublist. Links in an undirected PFNET are labeled according to the basis for their inclusion in the PFNET. The four types of link labels are PRIMARY, SECONDARY_A, SECONDARY_B, and TERTIARY. A PRIMARY link provides the only path between a node sublist containing a single node and some other node sublist. A SECONDARY link joins two sublists which are not connected, and in which there are either alternate paths to terminator nodes, or the node size of both node sublists exceeds one. SECONDARY_A links are in all minimum cost spanning trees. SECONDARY_B links are in only some minimum cost spanning trees, as they provide alternate paths of the same length between two nodes. A TERTIARY link joins nodes within a single node sublist. TERTIARY links are not in any minimum cost spanning tree. The link-labeling rule yields important structural information, and the potential use of link labels in the descriptive processes will be discussed in Section IV.

Investigation of transformations on the values of the weight matrix has yielded two results of importance relating to the structure of PFNETs. A multiplicative transformation applied to the elements of a weight matrix preserves link structure in the PFNET for any values of r or q. A monotonic transformation applied to data in a weight matrix preserves the structure of the PFNET only for $r = \infty$.

The construction of the database for vision applications presumes a set of entities and some procedure to derive feature vectors to represent these entities. Typically, the entities to be represented by nodes in the database are examined to obtain salient features. Each class of feature values is presented as a vector; e.g., a color descriptor could include intensity values obtained from red, green, and blue filtered images. Similarly, shape descriptors might consist of a vector of Fourier coefficients. Difference measures for this paper are obtained using the $L1$ norm (the computation is the same as the Minkowski distance for $r = 1$) for each pair of entities, to obtain the weight matrix for input to Pathfinder.

The Pathfinder model preserves all geodetic (minimum cost) paths having no more links than the value of the q parameter, and leads to clustering based upon similarity of nodes. Pairs of nodes which are not directly linked in a PFNET are likely to be in different categories or subcategories. PFNETs provide a means of scaling data similar in some respects to clustering methods (e.g., Shepard and Arabie, [11]) and to multidimensional scaling (Kruskal, [12]), but the links in PFNETs provide information not directly available in clustering or in multidimensional scaling. Another network scaling scheme is NETSCAL (Hutchinson, [13]), but unfortunately Hutchinson did not consider triangle inequalities of dimension greater than two.

The domain assumed for the database consists of those problems in which each entity in the database is represented by a vector having d feature values, and corresponding features have their feature values in corresponding locations in the feature vectors. In addition, we assume that the features of the entities are such that taking the difference of corresponding feature values (as a part of applying the $L1$ norm) is appropriate. To begin the process of generating a PFNET, it is necessary to compute a scalar weight matrix W For the purposes of this paper, we will compute the scalar weight values of W using the $L1$ norm. For the Lm norm,

$$w_{ij} = [\sum_{k=1}^{d} |w_{i_k} - w_{j_k}|^m]^{1/m}$$

The $L1$ norm sums the magnitudes of the differences between corresponding components for the feature vectors being compared. If the $L2$ norm were used, it would be the Euclidean metric. For a discussion of distance measures suitable for data such as this, see Tversky and Krantz [14]. This article also justifies the Lm norms (the Minkowski metrics) as being the only metrics which possess both intradimensional subtractivity and interdimensional additivity, a feature that seems as important for vision databases as for cognitive modeling.

Although the r-metric parameter used with PFNETS can vary from 1 through ∞, for the purposes of the databases described in this paper, ∞ will be used. The consequence of this is that the distance between nodes not directly linked is the value of the largest weight along the path connecting the nodes. This is sometimes called the *dominant* metric. The databases in this paper use $q = 2$. The notation used for a PFNET in which feature vectors are used to compute W is PFNET(Lm, r, q), with the parameters in parentheses corresponding to the parameters discussed above.

A primary purpose of the databases constructed as PFNETs is to support effective search, so that notation for search paths is helpful. Search paths begin at some initial node, follow links established in the construction of the PFNET, and end at a node. Since there is at most one link between any two nodes, we will denote a search path from node N_i to node N_k (passing through N_j) as

$$P(N_i, N_j, \ldots N_k)$$

N_i is said to be a *predecessor* of N_j because it precedes N_j in the search path. One way of viewing the network is to think of the entities N_i as points in d-dimensional space, establishing links according to the link membership rule of PFNETs, and traversing these links according to the search procedure to be described.

Another concept of importance is the *lune* of two nodes. The lune is discussed in Toussaint [15] in his definition and discussion of relative neighborhood graphs (RNGs). Lunes are also discussed in Lee [16] and in Katajainen and Nevalainen [17]. The lune of two nodes (points) N_i and N_j will be denoted by lune(N_i, N_j), and is defined as the set of points in which each point has a distance (we'll use the $L1$ norm, rather than the $L2$ norm as in the original work on RNGs) from both N_i and N_j less than the distance between N_i and N_j. In the weight matrix W, these internode distances using the $L1$ norm are already computed. Using $L1$, lune(N_i, N_j) is a rectilinear figure of dimension d. If $L2$ were used, lune(N_i, N_j) would be the set of points in the intersection of two hyperspheres of dimension d.

A notable difference between RNGs and PFNETs is in the assumptions usually made about the input spaces. For the RNG, two-dimensional space is normally assumed for the input data, but no such constraint is necessary for PFNETs. If the input spaces are two-dimensional, however, then the link membership of an RNG using $L2$ is such that the RNG is a spanning subgraph of PFNET($L2$, $r=2$, $q=2$). In the RNG, two nodes N_i and N_j are linked directly if and only if there is no other node in the lune(N_i, N_j). Euclidean distance is used in the 2-space in which the entities are customarily represented as vectors for the RNG, so that the lunes are intersections of circles.

Although PFNET($L2$, $r=2$, $q=2$) is satisfactory for a database organization, in terms of search for a matching entity in the database, the use of PFNET($L1$, ∞, 2) is preferable because the latter requires less computation in both the construction of the network and in search. Using $L1$, and assuming two dimensional input space, the link membership for an RNG is the same as the link membership of PFNET($L1$, $r=\infty$, $q=2$). For either, the link membership rule is

l_{ij} is in PFNET($L1$, ∞, 2) if and only if

$$w_{ij} \leq \min [\max [w_{ik}, w_{kj}]]$$

over all two-link paths between N_i and N_j.

This definition can be viewed as providing a link membership rule for one of a family of PFNETs, or as an extension of the RNG to a new application making use of the $L1$ metric.

The reason this PFNET is efficient in searching for a matching entity in the database is that the link placement is such that backtracking is never needed, provided that there is a matching node in the database.

III. Monotonic Search of a Pathfinder Database

Efficient, monotonic search, in which there is always a link to direct the search path(s) toward a node matching the external entity to be identified, is one of the attributes of a database organized as a Pathfinder network. Justification of monotonic search of a PFNET($L1$, ∞, 2) database follows. Consider that a set of entities N_i has been established with their corresponding feature vectors, and that the scalar weight matrix W has been computed using the $L1$ norm. Suppose that the PFNET($L1$, ∞, 2) has been constructed, and that E_x is an external entity represented by a feature vector compatible with the feature vectors of the N_i in the database; it is desired to find the N_k which provides the closest match to E_x within the database. Further suppose that the initial node (the node where search is initiated) is chosen to be N_i, and that node N_k in the database is a match for E_x.

That is, we assume for this discussion that the feature vectors for E_x and N_k are identical. The goal is to find a path between N_i and N_k, applying the match criterion at each node along the search path, until it is determined that E_x does indeed match N_k. An appealing argument can be made through the use of the lunes defined by the nodes along the search path. Consider lune(N_i, N_k) --if there is no other node in this lune, then N_i and N_k are linked directly, and a one-link search path connects the initial node with the goal node. Alternatively, if there is another node in lune(N_i, N_k), then N_i and N_k are not directly linked. But each node in the lune(N_i, N_k) is closer to N_k than N_i is to N_k, so that in progressing to any node in the interior of the lune, we diminish the distance to N_k. The node N_j closest to N_i will be linked to N_i, and the search path can proceed to N_j. The search path can, however, proceed to any node in lune(N_i, N_k) which is linked to N_i, and it is usually advantageous to go to the node which diminishes the distance to the goal node the most. Suppose the search progresses to node N_j. Here, the process is repeated with lune(N_j, N_k), and every node in this lune is closer to N_k than is N_j, so again the distance to the goal diminishes. In this fashion, the goal node is reached using only distance measurements between E_x and the nodes which are candidate successors for nodes on the search path, since we assume that E_x and N_k have the same feature vectors. That is, at node N_j, the difference between E_x and nodes linked to N_j is taken, and the difference which is smallest determines the next node in the search path. Thus the link structure guarantees that no backtracking is ever necessary if the entity E_x is in the database. If the distance from some node in the search path to E_x does not reach zero and cannot be diminished, then this indicates that there is no node which exactly matches E_x in the database.

A matching criterion based upon network properties is under development, although some aspects of a match criterion are necessarily dependent upon the problem domain. Throughout this paper, we presume that the matching criterion requires the goal node and E_x to match much more closely than E_x matches any other node in the database. Refinement of the theory of matching criteria is an area we are continuing to investigate.

There are four aspects of the search process for a database as described above, although the fourth is not always needed. These are:

(1) The selection of an initial node from which to begin the search, usually by means of some heuristic.

(2) The selection of a path from the initial node to the best matching node in the database.

(3) The application of the match criterion to each node along the path through the database, to determine whether any node on the path is a satisfactory match for the feature vector representing the entity to be identified.

(4) The determination of nearest neighbors of the node most nearly matching the external entity. Pathfinder networks support this search for nearest neighbors, because the link structure preserves geodetic (minimum) distances throughout the network.

The selection of an initial node can be accomplished by means of an index on some of the most salient features, so that, most of the time, the initial node is in the same class as the entity to be identified. Search efficiency is enhanced, of course, if the search is begun in the proper category. But the property of PFNET($L1$,∞,2) of guaranteeing that from each node, the distance to every other node in the database can be diminished by traversing some link assures that the choice of the initial node does not affect the convergence to the goal node if the latter is in the database. This is important because it is not alway possible to begin the search at a node in the same category as the goal node.

The search procedure consists of the following steps, to be done at each node in the search path, from the initial node to the node at which the decision regarding a match can be made:

(1) The match criterion is applied to the present node (starting with the initial node) to determine whether or not the present node is a satisfactory match with the entity E_x. If the match is satisfactory, then halt.

(2) The distance $d(N_i,E_x)$ between each node N_i adjacent (linked) to the present node and the entity E_x is computed using the $L1$ metric.

(3) The node N_i which decreases the distance $d(N_i,E_x)$ the most is selected as the next node in the search path. If it is not possible to decrease the distance to the goal node (represented by E_x) then there is no matching node in the database. Otherwise, return to step one.

As an example, consider the set of nodes

$$\begin{array}{ll} N_1 = (2, 1) & N_5 = (9, 4) \\ N_2 = (4, 1) & N_6 = (9, 6) \\ N_3 = (5, 4) & N_7 = (7, 8) \\ N_4 = (3, 5) & N_8 = (10, 8) \end{array}$$

The weight matrix for this set of feature vectors, computed using the $L1$ norm, is

$$W = \begin{bmatrix} 0 & 2 & 6 & 5 & 10 & 12 & 12 & 15 \\ & 0 & 4 & 5 & 8 & 10 & 10 & 13 \\ & & 0 & 3 & 4 & 6 & 6 & 9 \\ & & & 0 & 7 & 7 & 7 & 10 \\ & & & & 0 & 2 & 6 & 5 \\ & & & & & 0 & 4 & 3 \\ & & & & & & 0 & 3 \\ & & & & & & & 0 \end{bmatrix}$$

The PFNET($L1$,∞,2) is constructed using W, and is shown in Figure 1.

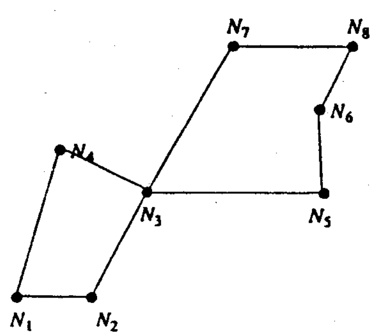

Figure 1

If the search is started at $N_1 = (2, 1)$, with $E_x = (10, 8) = N_8$, then the search path is $P(N_1, N_4, N_3, N_7, N_8)$, and the search procedure halts at N_8 with a match. Links are followed at each step, and the distance to the goal node decreases monotonically at each node.

The match criterion necessarily has some aspects which are domain dependent, but could be as simple as requiring that the distance between the goal node (providing the presumed best match) and E_x be less than some threshold value computed from the smallest distances between nodes in the database. Refinements could include the addition of an element of context in the sense that each category may have somewhat differing variability associated with satisfac- tory matching. The search process outlined above does not guarantee that the path with fewest links will be found, but it does guarantee that a path to a matching node will be found in which the distance decreases monotonically along the search path. If there is no node in the database which is an exact match to E_x, this is determined when the distance from a node N_j (on the search path) to E_x is larger than the distance from the preceding node in the search path to E_x. In this case, using the $L1$ norm, then it is not possible to get a better match to E_x than the predecessor of N_j; if that is not a satisfactory match, then E_x is considered not to be in the database. Formal proof of this is forthcoming in Dearholt [18].

IV. Description of Decisions and Neighborhoods

For an intelligent system, it is desirable to have support for the description of decisions made during the search process. This information can be very useful for communicating with the system in an attempt to understand not only the classification decision for a particular entity, but the properties of the neighborhood surrounding the entity in the network. The latter information can, of course, be used in some of the more sophisticated classification algorithms. Because of the clustering properties of PFNETs, and the directness of the search process associated with PFNET($L1$, ∞, 2), there is substantial information available regarding the classification results. The categories of nodes along the search path, and values of some of their most salient features, are the principal pieces of information used for our descriptive processes. We will focus mainly on four issues:

(1) The node where search is initiated,

(2) The search path,

(3) Link labels,

(4) The classification decision.

The selection of the initial node for the search procedure is a very significant decision. Although the PFNET($L1$, ∞, 2) guarantees convergence between any pair of nodes, the search time can be lessened substantially in a large database by judicious selection of the initial node. Beginning at a node which is in the same cluster as the goal node is a desirable objective; but the solution of this problem would imply that the classification problem for the entities in the domain is also solved. For many domains, the selection of a few key features which often lead to correct classification can be used to provide a sort of indexing into the network, so that the initial node could be the node having highest degree in the category indicated by the feature values in the set of key features. These key feature values, and the heuristic selection of an initial node based on them, are thus a part of the descriptive process at the beginning of the search.

As the search begins from the initial node (selected by some heuristic), at each node N_i in the search path some decision is made based upon the distance between the external entity and the nodes linked to N_i. The most suitable strategy, as discussed in Section III, is to select the node which most decr ses the distance to the goal node, although there is no guarantee that the goal node will be reached in the fewest steps by using this strategy. The values of the feature vectors of the nodes which are candidate successors to N_i are available, and for the N_j which is the successor node to N_i, the feature value(s) which are most responsible for diminishing the distance to the goal are available for use in description. Together, these feature values are an indication of progress made toward the goal entity, since backtracking is never necessary. Furthermore, if there are multiple search paths (previously defined as the paths leading to the goal node), properties of all of the search paths can provide important information to the descriptive process.

The link labels on the links traversed also have some significance, as pointed out in Section II. PRIMARY and SECONDARY_B links are typically indicative that the search is progressing within a category, while the traversal of a TERTIARY link in a PFNET($L1$, ∞, 2) seems to indicate that the search path has progressed to a new category or subcategory. The traversal of a SECONDARY_A link also usually indicates entry into a new category. Proofs of these observations are difficult and not yet available, partly because the definition of "category" or "subcategory" is difficult. We are continuing to investigate the information provided by link labels, however, and that information is available for description also.

The last part of the search involves classification of the external entity, if that is possible. The choices of nodes available at the last step of the search provide information regarding the class and the centrality of the entity, so that some level of confidence in the decision of category could be assigned. That is, if the match with the goal node were borderline, and the node were near another category, then the confidence in the classification should not be very great. But if the goal node matched quite well and were also in the "center" of a category, then the confidence of the classification should be high. The Pathfinder paradigm supports a local search for nearest neighbors, so that this information can be used in either the classification decision or in the description of the neighborhood surrounding the goal node. The search for neighboring entities can be viewed as search by spreading activation, which would leave the goal node and travel to the neighboring nodes, so that their feature values and classification is available. Thus the centrality of the goal node, and its relationships to other nodes and categories, can be readily determined.

V. Summary and Conclusions

The construction of a database for vision applications using Pathfinder networks (PFNETs) was described, and it was shown that the search procedure associated with this database organization is monotonic (a distance measure steadily decreases at each node throughout the search path), provided there is a node in the database which matches the external entity. Relationships with the relative neighborhood graph (RNG) were discussed, and the search procedure was described. The database organization and search procedure provide a basis for descriptive processes of the decisions made along the entire search path, from the initial node to the goal node (or to the point where it is decided that there is no match in the database). Description based on these feature values and changes in feature values along the search path(s) and in the neighborhood of the goal node is expected to be useful in enhancing communication between the vision or robotics system and humans working with the system.

Research is continuing on some of the open questions encountered thus far. The match criterion used to determine whether the external entity matches some node in the search path "satisfactorily" is an important problem, and it has some domain-dependent properties and some characteristics which can be determined from a graph-theoretic perspective. The precise role of the PRIMARY, SECONDARY, and TERTIARY links in PFNET($L1$, ∞, 2) is also being studied, particularly how these link labels relate to category structure of the network. The descriptive processes are under investigation from the perspective of graph theory, although there seems to be a domain-dependent aspect also.

VI. Acknowledgements

The research described in this document was performed at New Mexico State University and was supported in part by National Science Foundation Grant IST-8506706 and by the Computing Research Laboratory, New Mexico State University.

VII. References

[1] Dobkin, D. and R.J. Lipton, "Multidimensional Searching Problems", SIAM Journal of Computing, Vol. 5, No. 2, pp. 181-186, June 1976.

[2] J.L. Bentley and J.H. Friedman, "Data Structures for Range Searching", Computing Surveys, Vol. 11, No. 4, December 1979.

[3] J.L. Bentley and H.A. Maurer, "Efficient Worst-Case Data Structures for Range Searching", Acta Informatica 13, pp. 155-168, 1980.

[4] J.M. Chang and K.S. Fu, "Extended k-d Tree Database Organization: A Dynamic Multiattribute Clustering Method", IEEE Transactions on Software Engineering, Vol. SE-7, No. 3, May 1981.

[5] A. Kalvin, E. Schonberg, J. Schwartz, and M. Sharir, "Two Dimensional Model Based Boundary Matching Using Footprints", New York University, Dept. of Computer Science, Courant Institute of Mathematical Sciences, Technical Report No. 162, Robotics Report No. 44, May 1985.

[6] C. Stanfill and D. Waltz, "Toward Memory-based Reasoning", Communications of the ACM, Vol. 29, Number 12, pp. 1213-1228, December 1986.

[7] D.W. Dearholt, R.W. Schvaneveldt, and F.T. Durso, "Properties of Networks Derived from Proximities", Computing Research Laboratory Memorandum 14, Computing Research Laboratory, New Mexico State University. MCCS-

85-14, June 1985.
[8] R.S. Schvaneveldt, D.W. Dearholt, and F.T. Durso, "Networks from Proximity Data", to appear in Computers and Mathematics with Applications, 1987.
[9] D.W. Dearholt, N. Gonzales, J. Ellington, and K. Phillips, "A Specialized Database for Classification Using Template Matching", Proc. of the Nineteenth Annual Asilomar Conference, pp. 494-496, Asilomar, California, November 1985.
[10] D.W. Dearholt, N.N. Gonzales, and H. Kirpekar, "A Vision Database Designed for Search, Classification, and Description", Proc. of the Twentieth Annual Asilomar Conference, Asilomar, California, November 1986.
[11] R.N. Shepard and P. Arabie, "Additive Clustering: Representation of Similarities as Combinations of Discrete Overlapping Properties", Psychological Review, 86, 87-123, 1979.
[12] J.B. Kruskal, "Multidimensional Scaling and Other Methods for Discovering Structure", in Enslein, Ralston, and Wilf (Eds.), Statistical Methods for Digital Computers. New York: Wiley, 1977.
[13] J.W. Hutchinson, "Network Representations of Psychological Relations", unpublished Ph.D. dissertation, Stanford University, 1981.
[14] A. Tversky and D. H. Krantz, "The Dimensional Representation and the Metric Structure of Similarity Data", Journal of Mathematical Psychology 7, pp. 572-596, 1970.
[15] G.T. Toussaint, "The Relative Neighbourhood Graph of a Finite Planar Set", Pattern Recognition, Vol. 12, pp. 261-268, 1980.
[16] D.T. Lee, "Relative Neighborhood Graphs in the L1-metric", Pattern Recognition, Vol. 18, No. 5, pp. 327-332, 1985.
[17] J. Katajainen and O. Nevalainen, "Computing Relative Neighbourhood Graphs in the Plane", Pattern Recognition, Vol. 19, No. 3, pp. 221-228, 1986.
[18] D.W. Dearholt, "Associative Network Databases Constructed for Efficient Search", in preparation.