Matching Complex Images to Multiple 3D Objects using View Description Networks

J. Brian Burns and Edward M. Riseman *

COINS Technical Report 91-87

November 15, 1991

Abstract

This paper addresses the problem of effectively matching a single 2D image of a potentially cluttered scene to a library containing multiple polyhedral objects. In our approach to recognition, the search for matches to 3D objects from a multiple object library is optimized by generating descriptions of the projections of the objects from expected views, organizing all the descriptions into a single network representation, and then, during the *recognition phase*, finding matches between the resulting *view description network* and the input image. Our design for the recognition phase process is presented and demonstrated on images containing multiple objects and outdoor scenes. The efficiency of the system and the general approach are discussed.

*This research was supported by the Defense Advanced Research Projects Agency (via TACOM) under contract DAAE07-91-C-R035.

1. Introduction

Our research objective is a system capable of recognizing modelled polyhedra from 2D images of cluttered scenes. The system is presented with a library containing multiple objects and an image of a scene that may contain any combination of the objects in arbitrary positions with respect to the viewer. The goal of the system is to find the correct 3D matches to all objects in the scene for which there is sufficient evidence in the image, where a *3D match* is an assignment of model features to image features and the estimated coordinate transformation between model and camera (pose) that best aligns the assigned features. The recognition system must be designed to find the correct 3D matches in an efficient manner. A 3D match is completed and verified by the potentially costly process of searching for image data that comprise sufficient evidence for the match [13]. In addition, the number of possible 3D matches can be very large; for the images and objects studied in this paper, there are tens of billions. Therefore, it is essential for the efficiency of the system that the number of 3D matches selected for completion be minimized.

In our approach, the search for matches to 3D objects from a multiple object library is optimized by generating descriptions of the projections of the objects from expected views, organizing all the descriptions into a single network representation, and finding matches between the resulting *view description network* and the input image. The contributions of our research have been in the automatic compilation of network descriptions of object *views* [6, 8] and the analysis of the usefulness of 2D features for 3D object discrimination under view variation [7, 8].

In the latter work, a study of invariants with respect to view position is presented. The recognition of 3D objects from a single, unknown view can be facilitated by the use of image features that are invariant; our study concerned invariants that are functions of projected 3D point sets. View invariants exist for special classes of objects (3D point sets), such as those constrained to planes, and these functions have been used in recognition research [18, 19]. However, we have shown that there does not exist a function that is view-invariant for *arbitrary* 3D point sets of size n, for any n, and recognition systems for the unrestricted problem cannot be expected to be as effective as the special-case schemes. This analysis and conclusion is consistent with subsequent work on affine invariants [9]. Since view invariants cannot be guaranteed to exist or be useful for any given recognition problem (i.e., collection of 3D objects), it is imperative that systems be developed that are capable of utilizing view-varying features as well. For example, features such as relative orientation and length of projected line segments in arbitrary positions can be useful for the discrimination of many objects, from most of their views [3, 8]. Because features crucial for recognition may vary with view, their probability distributions may be non-trivial. Systems developed to use these features must effectively model the distributions and apply probabilistic means of evaluating the matches [2, 3, 8].

In this paper, we present the *recognition phase* of our matching system; this phase involves the search for matches between a compiled *view description network* and the input image. Issues include the effective evaluation of partial matches generated during the search and the control of the search process. Our design for the recognition phase process is discussed and demonstrated on images containing multiple objects and outdoor scenes. The efficiency of the system and the general approach are discussed.

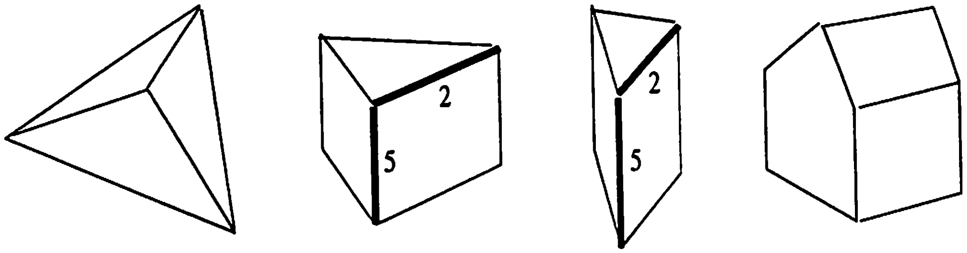


Figure 1: Objects used to demonstrate the system.

2. The description network approach to object recognition

Objects in a library, such as in Fig. 1, will typically have a variety of similarities and differences which must be considered while optimizing the selection of the correct matches. In general, objects can be differentiated using combinations of two types of features: metric and structural. A *metric* feature is any function that can be defined for a set of projected elements in the image; examples of this are the *affine coordinates* [18], which are view invariant for planar objects under weak perspective projection. Clearly, this type of feature can be useful for discrimination; however, it is important to note that non-invariant (view varying) metric features should also be considered for discrimination purposes. For example, the two triangular prisms in Fig. 1 *cannot* be discriminated by affine coordinates or other view invariants, since they are identical up to an affine transformation (they are affine equivalent). Yet, these two prisms *do* have different proportions. This differentiating property can be measured in terms of the length ratio of the bold line segments labelled 2 and 5, which is actually capable of distinguishing the two prisms for the great majority of their views

[7, 8]. In addition, some view-varying features are potentially more tolerant of image noise than other features: the angle between two line segments depends only on the estimated orientation of the detected lines, not on the exact position of the four endpoints, as required by affine coordinates.

Structural features represent how elements that make up an object, such as straight line segments, are connected or otherwise organized. For example, in Fig. 1, some objects have line segments assembled into triangles and others into pentagons or parallelograms, and these assemblages themselves are connected into larger structures. Clearly, the identification or detection of these structural features can help in the discrimination of the objects. The identification of structural features in the image is important in another way: matched structural descriptions provide a context in which metric features can be measured and meaningfully used for discrimination. For example, the length ratios considered useful in Fig. 1 are meaningless if we do not know *which* pair of image segments are to be measured. There may be an enormous number of ordered pairs of line segments in a cluttered image, and each pair produces its own length ratio. The image segment pair whose ratio provides the discriminating length-to-width proportion must be identified, and this can be done by first matching a structural description.

The importance of structural features for discrimination must be stressed, for their use has a fundamental effect on the design of a multi-object recognition system. Since identifying structural features in the image means searching for matches to non-trivial structural descriptions, an important part of the object discrimination process is an optimized search for these matches. Organizing the structural descriptions is an important step towards optimizing this search, an approach stressed in this study and related work [1, 4, 10].

In our design, object information is organized into *description networks* where parts or geometric aspects shared by objects are explicitly represented as nodes in a network, with direct or indirect links to all the objects characterized by them. Objects are then recognized via the network by a process referred to here as *recursive indexing.* In this strategy, indexing of object information takes place in stages; each indexing step identifies important substructures (parts) in the image which are in turn used as structural features to index more complex descriptions, until descriptions to specific objects are indexed and successfully matched.

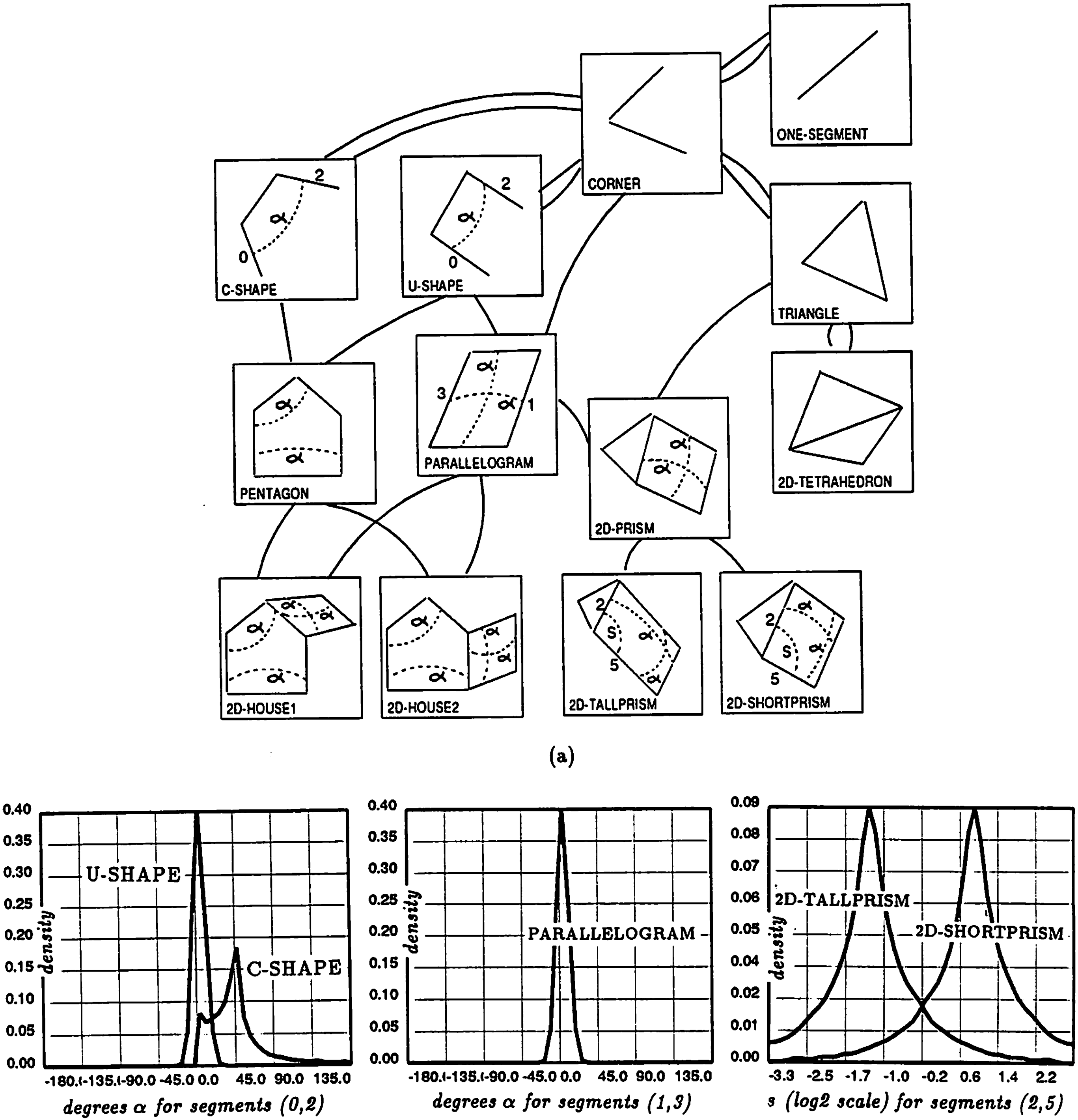
This approach to recognition is in contrast to two other important recognition strategies that have recently seen increased development: the single-level, *geometric hashing* methods [9, 18, 19, 22, 24], and the use of *interpretation* trees [12, 15, 23].

Lamdan *et al* [19] have demonstrated by simulation studies that the voting (indexing) step of their geometric hashing algorithm produces a significant improvement over exhaustive testing by 3D pose determination, even in the presence of noise. However, recursive indexing could be a further improvement over the unstructured, or single-level, hashing methods. In single-level systems, simple image feature combinations are used to directly index the objects in the library. In [22], experiments using real image data were performed to estimate the time complexity with respect to object library size for a single-level system. As indicated by these experiments, the object-specificity of the features greatly affects the number of objects retrieved, and the simple combinations of features used in their system were not sufficient to avoid a saturation effect (where the number of objects retrieved grows at least linearly with the size of the object library). The specificity problem may become more manageable using a recursive, multilevel procedure, primarily because combinations of structural parts identified in earlier steps could provide the system with a rich set of composite features for indexing at the next step. This is consistent with the analysis of object indexing complexity made by Clemens and Jacobs [9], in which they observe that indexing is of very limited benefit unless a process is incorporated that can extract a relatively small number of interesting feature sets that each contain a large number of features. They recommend a feature *grouping* process. A recursive indexing design that incorporates perceptual organization heuristics can also achieve this behavior; such a system is presented in Section 4.

It is also important to distinguish the recursive indexing of descriptions in a network from the use of interpretation trees. An interpretation tree search when clutter is present in the image may be prohibitively inefficient [12]. The recursive indexing strategy supported by networks, however, has a fundamentally different behavior that may mean much greater efficiency. Instead of methodically searching a large portion of a tree for possible interpretations, the evidence from matches to multiple parts and the *convergent* structure of the description network are used together to provide a potentially focused search for the correct interpretation. The experiments in cluttered images presented in this paper provide some demonstration of this.

3. View description networks

View description is an important approach to recognizing 3D objects in 2D images. In this method, prior to recognition, descriptions of the projections of each object from distinct



(a)

(b)

Figure 2: View description network for objects in Fig. 1. (a) Each 2D model node in the network is represented in the figure by a line-segment example that satisfies the 2D model. Dashed lines connecting line segments indicate metric features that are functions of the pair of specified line segments. (b) The feature probability densities given the different 2D models discriminated by them. Each density function is stored in the appropriate 2D model node and is inherited by all of its successors in the network.

views are generated; then, during recognition, these 2D expectations are matched to the image. A view description may be valid for the object's projection over a range of views, as long as there is a description associated with each view and the descriptions of each object are distinct from those of other objects. This method is useful since it does not depend on reliable 3D scene reconstruction. It also facilitates the use of image information sufficient for recognition [3, 7] and the efficient matching of *multiple* 3D objects to cluttered 2D images through the organization of the view descriptions into networks. Similar approaches can be found in [2, 10, 11], and also in [15], though the system of Ikeuchi is not for single intensity images.

In our system, a view description is a relational graph that represents the discriminating features for some object and range of view. Each distinct element in the graph represents a line segment, and the relations (arcs) are features defined over the associated line segments. The relation stores the feature type, such as line segment length ratio s or relative orientation α, and the probability densities of the feature given the associated pair of object line segments and a uniform sampling across the range of view.

The view descriptions are organized into a network, where each terminal node corresponds to an object-specific view description. The view descriptions are recursively built up from smaller, simpler relational graphs associated with intermediate nodes in the network via *combination* and *specialization* links. We will refer to the information stored in any node as a *2D model*. Fig. 2 shows a network automatically constructed to recognize the objects in Fig. 1. The combination link specifies how a set of part descriptions are combined into more complex, object-specific ones; currently, combinations are formed of pairs of parts.

For example, in Fig. 2(a), the model 2D-PRISM is represented as a combination of 2D models TRIANGLE and PARALLELOGRAM, which are isomorphic to two of its subgraphs. Specialization links between a 2D model node and its network successors specify the addition of new relations to the 2D model; in other words, new features (see Fig. 2a, dashed labelled arcs) and their probability density functions given the object associated with each successor node (see Fig. 2b). For example, the 2D-PRISM model is associated with two objects (*short-prism* and *tall-prism*), and each is assigned a successor containing a new relation: the feature s for element pair $(2,5)$ and its density function for the relevant object. (New relations can also be added during combination, see Fig. 2b.) Discrimination of the objects based on the stored probability information is discussed below in Section 4.2.

It is important to note that, even though our compilation process generates view descriptions for rigid 3D objects, the view description network and the recognition phase of the system are based on relational graphs. They can thus be readily used to represent and match information about *non-rigid* objects. This adaptability to non-rigid domains is a key property of our system.

4. Matching images to view description networks

Given a view description network, the goal of the recognition phase of the system is to identify objects in images by efficiently searching for matches between image line segments and object view descriptions (2D model nodes) stored in the network. It was argued above that *recursive indexing* is an effective way to recognize using a description network. This approach is best understood as recursive 2D match extension followed by 3D match com-

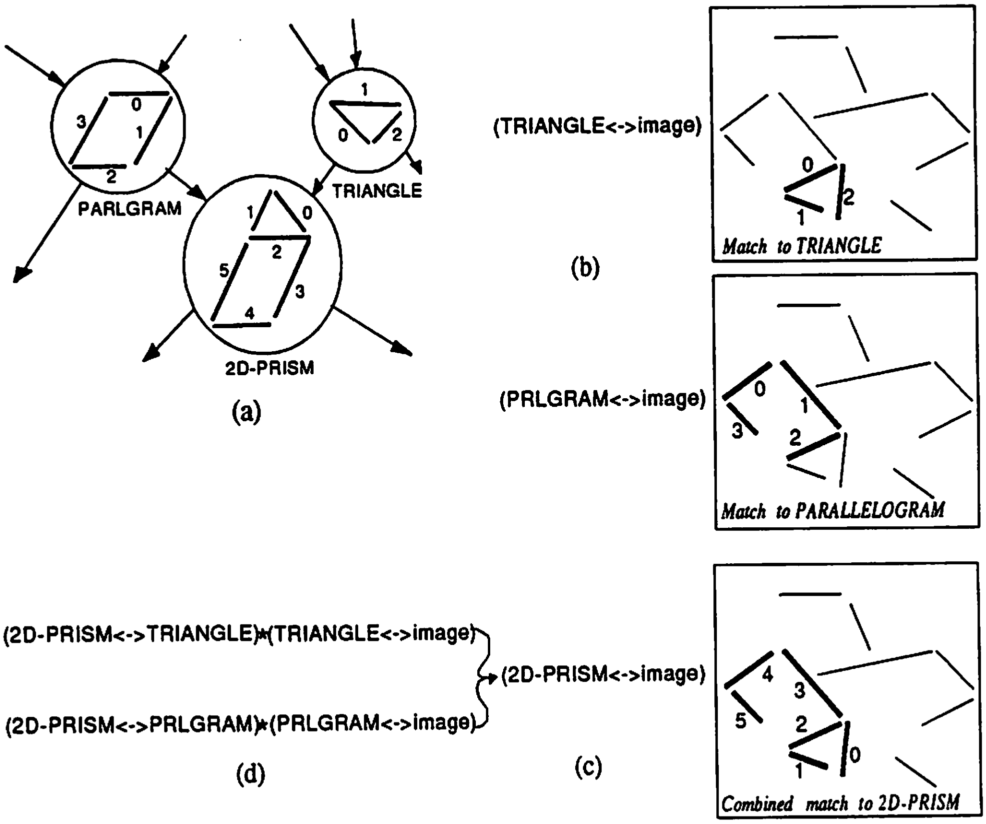


Figure 3: An example of match extension. (a) Relevant portion of Fig. 2. (b) Matches to TRIANGLE and PARALLELOGRAM. (c) A match to their common successor 2D-PRISM, created by (d) composing each predecessor node match with the model-to-model maps specified in the network and then merging the two resulting maps.

pletion and verification. *Match extension* is the creation of a match to a 2D model (a *2D match*) from matches to its predecessors in the network, where a 2D match is the assignment of the line elements in the 2D model to line segments detected in the image. For example, given the network in Figs. 2(a) and 3(a), matches to TRIANGLE and PARALLELOGRAM (Fig. 3b) can be combined into a match to 2D-PRISM (Fig. 3c).

The design of the recognition system follows naturally from the recursive nature of matching to view description networks. The process is initialized by detecting line segments in the image and generating promising matches to the initial, simple 2D model nodes in the network. The system then searches for correct matches to the more complex 2D models, and eventually to the 3D models, by iteratively executing the following three steps:

1. *Extend* or *verify* the selected 2D matches, depending on the type of match.
   (a) Matches to 2D model nodes in the terminal portions of the network are associated with 3D objects; when selected, the system attempts to verify them by computing the 3D match.
   (b) Otherwise, the system attempts to extend the match to a more complex 2D model node match.
2. *Evaluate* and *incorporate* the resulting 3D or 2D matches into the current state of the system.
   (a) All new *2D* matches are added to the pool of matches that *may* be extended or verified in future cycles.
   (b) If a *3D* match is verified, it is output from the system, and competing 2D model matches are eliminated. A match is competing if it assigns the same image segments.
3. *Select* the best 2D matches for extension and verification in the next cycle.

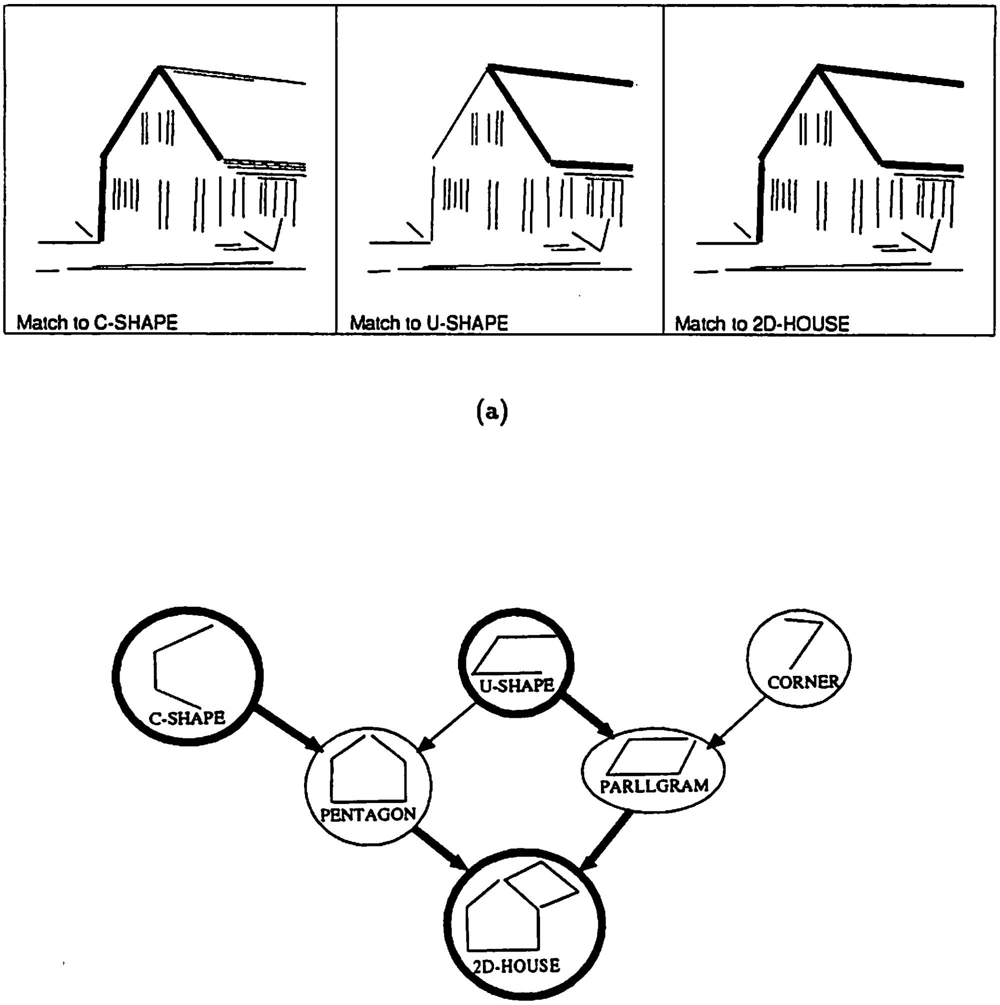
In the experiments, the process was made to terminate on discovery of all the correct 3D matches. The 3D matches were not required to be complete, but the matched image segments had to be close to the estimated projected segments in position and extent.

The organization of the system into the above three steps naturally structures the discussion into three parts: 2D match *extension*, *evaluation* and *selection*. Verification by 3D match completion is also clearly important, but is outside the scope of this paper; further discussion of match completion and pose analysis can be found in [13, 14, 16, 17]. An implementation of the 3D pose algorithm of Kumar [16, 17] was used in the demonstrations presented in this paper.

4.1 Match extension

As discussed above, the matching of descriptions organized into networks assumes the form of recursive extension. Given a pair of existing 2D matches, their extension has two steps: retrieve 2D models to which they can be extended, and then, compute the line segment assignments for the new, extended match as in Fig. 3.

Complications can occur when key parts of an object's projection, or important relationships between the parts, are poorly represented in the image. The network contains an idealized description of the object projections, with ideal parts and relations between these parts. If the actual representation of a part in the image is poor, then a simple, step-by-step, recursive extension could be difficult and the extension process must be made more adaptable. The matching of a house image in Fig. 4(a) provides an example of this. (See Fig. 7 for the digital image.) Given the network in Figs. 2(a) and 4(b), a 2D-HOUSE model is made up of two parts, PENTAGON and PARALLELOGRAM, which are in turn made up of the simpler models, U-SHAPE, C-SHAPE and CORNER. The extension of the



(a)

(b)

Figure 4: An example of extension to an *indirect* successor in the network of Fig. 2; see text. (a) Matches to C-SHAPE, U-SHAPE and 2D-HOUSE. (b) A portion of the network, with relevant nodes and links in bold.

U-SHAPE match in Fig. 4(a) to a match of its *direct* successor, PARALLELOGRAM, via combination with CORNER is not possible since the image frame excludes the part of the projection associated with the desired CORNER match. Without the evidence associated with this CORNER match, there is no way to know whether the U-SHAPE match should be interpreted as part of a PARALLELOGRAM or PENTAGON match, both being successors of U-SHAPE (see Fig. 4b). However, a viable, unambiguous extension to the *indirect* successor 2D-HOUSE is possible by combining the U-SHAPE match with the available C-SHAPE match shown in Fig. 4(a). The importance of indirect predecessor extension for matching can be appreciated in the application of the recognition system to cluttered and corrupted images as demonstrated later in this paper. To support indirect predecessor extension, our system is designed to index 2D models given pairs of matches to their fragments, that is, their indirect predecessors. During the compilation phase of the system, 2D models and potentially useful pairs of their indirect predecessors are hashed into an *extension table* for rapid indexing during the recognition phase. In addition, to facilitate the extension operation, the compilation phase process also pre-computes and stores the model line segment mappings between the 2D models and useful indirect predecessors.

4.2 The evaluation of matches to 2D model nodes

Once a 2D match has been generated, its priority for extension or 3D verification needs to be determined for effective control. An important factor in assessing this priority is the probability that the match is correct, which is estimated in two steps. First, all of the metric and structural features represented in the 2D model are measured; then, the values of the

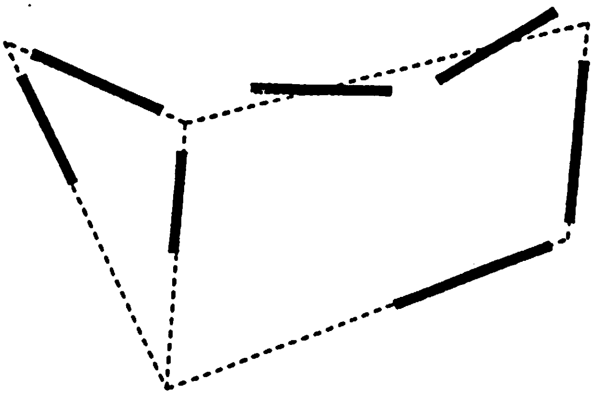


Figure 5: An example of line segment reconstruction during 2D model matching. Bold lines: line segments detected in an image of a tall prism. Dashed lines: reconstruction of the projection given a 2D-PRISM match.

measured features and the conditional density functions stored in the models are used to estimate the posterior probability of the match.

The metric and structural features used in the match evaluation are measured with respect to an approximate reconstruction of the object's projection consistent with the match. Fig. 5 shows an example of this. The structural features in the 2D model specify how the matched line segments should be connected together. The reconstruction enforces the specified segment connectivity, while minimizing the error between the detected fragments and reconstructed lines [8]. Since the errors reflect the degree of actual connectivity of the detected line segments, they are used as measurements (indications) of the structural features for match evaluation. The *metric* features, such as line segment length ratio s or relative orientation α, are measured with respect to the reconstructed lines.

A match is an assignment of 2D model lines to segments in the image, and this assignment is one possible interpretation of the image segments among many. These alternative interpretations are classes, and the problem of estimating the probability that a given match is correct can be treated as a problem in Bayesian classification:

$$P(\omega_1 \mid \vec{f}) = \frac{p(\vec{f} \mid \omega_1)P(\omega_1)}{\sum_{j=1}^{n} p(\vec{f} \mid \omega_j)P(\omega_j)}$$

where ω_1 is the class associated with the 2D model match being evaluated, the $\omega_j, j > 1$ are alternative interpretation classes, and $\vec{f}$ is the vector of measured features. This formulation is useful and straight-forward; however, it requires the determination of the alternative interpretation classes $\omega_j, j > 1$, and, for all classes $\omega_j, j \geq 1$, the class priors $P(\omega_j)$ and the conditional density functions $p(\vec{f} \mid \omega_j)$.

For a given match, there can be a very large number of alternative interpretations of the same set of image line segments; however, these interpretations can be usefully organized into a small number of classes. For each alternative class, the actual match being evaluated is considered *incorrect*, and in our formulation, each alternative interpretation class is associated with a different type of matching mistake that could have produced the incorrect match. The match being evaluated is generated by extending a pair of other matches, say M_1 and M_2, and this implies the following classes of alternative interpretations (matching mistakes): (1) M_1 and M_2 are both correct but the given extension is not (i.e., some *other* extension is); (2) M_1 and M_2 are correct but they cannot be combined into *any* valid extension; (3) M_1 is itself incorrect; (4) M_2 is incorrect; and (5) *both* M_1 and M_2 are in correct. These distinct classes have different priors and class conditional densities for the features.

The priors assigned to each class are derived from view analysis and estimates of match error rates. Assuming a uniform distribution for views, the prior probabilities for the given 2D match and classes of type (1) are a function of the visibility of the different 2D models; i.e., the fraction of views for which they are valid. The alternative interpretation classes associated with the other matching mistakes (2-5) are assigned priors that reflect the expected rate at which these mistakes occur. These error rates have not been rigorously estimated; however, this does not seem to have caused serious problems in the experiments.

Estimating the probability that a 2D match is correct also requires the conditional density functions for the features given each of the interpretation classes defined above. For all the classes considered, the features are assumed to be independent. As discussed in Section 3, the density functions for the *metric* features given a correct match to the 2D model are represented with the model (see Fig. 2b). The *structural* features are measured in terms of the reconstruction fit errors. Given that the 2D model match is correct, the error in the fit is strictly a function of the inaccuracies of the detected image segments, and the associated *detection error* density function is assumed to be a Gaussian with zero mean. When the match is *incorrect*, structural features (fit errors) are largely due to the incorrect model line assignments, and the associated *assignment error* density function is a much wider Gaussian. For some of the alternative interpretation classes defined above, not all of the match is considered incorrect, and thus not all of the associated reconstruction fit errors are modelled as assignment errors. For example, in class (3), the part of the match extended from M_1 is considered incorrect but not the part from M_2. The probability densities of the fit errors for these two different parts of the match thus reflect assignment and detection errors respectively. A more complete treatment can be found in [8].

4.3 Control: Selecting 2D matches for extension and verification

As discussed above, the overall form of the matching system is that of a search for correct 3D matches by iterative 2D match extension and verification. Therefore, the system's behavior is controlled by the selection of matches to extend or verify during each iteration. While verification is the transformation of a *single* 2D match into its 3D counterpart, match extension involves the combination of *pairs* of matches. It is computationally prohibitive to select them by evaluating and assigning a priority to every pair of existing matches. Thus, for each iteration, the selection proceeds in two steps: (1) select a set of *individual* matches, each with a high likelihood of leading to the correct interpretation, and (2) for each selected match that is to be extended, retrieve *partner* matches for combination.

Given the above process, it is clear that not all of the possible combinations of a match are attempted when it is selected for extension. Even if the combinations tried seem the most promising, the correct one may have been excluded. It is thus desirable to be able to *re-select* a match in another iteration, and retrieve a new set of partners to combine with it.

4.3.1 Selecting individual matches to extend or verify.

The set of individual matches selected during each iteration should satisfy some combination of the following two criteria: the matches must be promising candidates for extension or verification, and the selected set must be distributed about the image in a way that is advantageous for recursive matching to view description networks.

The first factor, selecting candidates for extension or verification with high likelihoods of leading to the correct 3D interpretation, is clearly a function of the probability that the

2D match itself is correct. However, for those matches selected for extension, this likelihood is also a function of whether or not the match can be successfully combined with another match. From the above discussion, it is clear that a match may have already been selected in an earlier iteration, and thus some of the most promising combinations with it will have been attempted already. It seems reasonable to assume that the chance that the correct combination has *yet* to be generated goes down each time a match is re-selected for extension. The priority assigned to a match is a product of the probability that it is correct and the probability that it can yet be successfully combined.

The second factor, matching an image to an object represented by a view description network, is best satisfied if the match extension activity is distributed over the projection of the scene object. In this way, matches to different parts of the projection will have a greater chance of being available for important combinations at roughly the same time. The following is the method used for distributing the selected matches: (a) initialize the matching system by generating primitive 2D model node matches in different portions of the image; and (b) for subsequent iterations, select the highest priority match in the neighborhood of each of the matches last extended. In our system, the *neighborhood* of a match includes itself and all matches related to it through extension.

4.3.2 Selecting match combination partners

Once a match is selected for extension, the system searches for other matches that provide promising combinations. Each time a match is re-selected for extension, a new set is sought for combination, in order of most promising sets first. A pair of matches is a

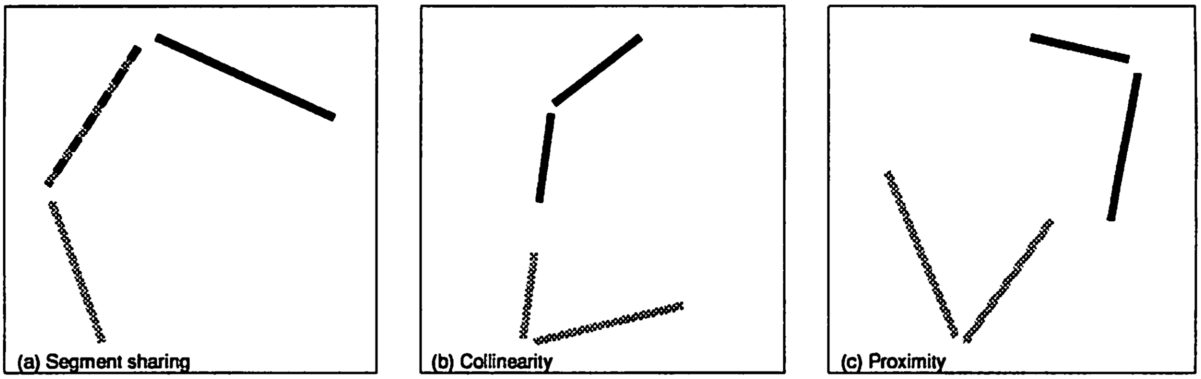


Figure 6: Different pairs of CORNER matches demonstrating three perceptual organization factors important for assessing the potential usefulness of a match pair as a combination. For each example, line segments associated with one match are shown in black, those with the other, in gray, and those with both, in alternating black/gray. The factors exemplified are (a) connectedness (segment sharing), (b) good continuity (collinearity), and (c) proximity.

promising combination if the resulting extension to a new 2D match has a high probability of being correct. In a cluttered image, it is easy to select a pair of matches whose individual probabilities are high, but their *combination* has a low probability of being correct; thus, it is important to be able to rank candidate pairs based on how they combine. An important set of heuristics for ranking combinations in order of their probability of being correct can be found in psychological studies of perceptual organization [21]. Generally, image features exhibit compelling perceptual organization if they appear to the viewer as parts of the same object. Perceptual organization heuristics have already been successfully incorporated into object recognition systems [9, 20, 24], and the effect of perceptually organizing, or *grouping*, image data on the time complexity of object indexing has been experimentally demonstrated by Clemens and Jacobs [9]. The contribution of our research has been the incorporation of perceptual organization processes into a *recursive* indexing design.

The three perceptual organization factors exemplified in Fig. 6, connectedness, good continuity, and proximity, are important for assessing match combinations and have been incorporated into our recognition system. *Connectedness* is exhibited when two matches share the same image line segment. If these two matches are correct, there is a high probability that they are of the same object's projection. Two matches exhibit *good continuity* if a pair of image line segments, one from each match, are close and approximately collinear. Given good continuity, the visual organization is very compelling, though not with the strength of connectedness. Finally, two matches are considered proximate if their associated image line segments are close. Proximate matches do appear more likely to be parts of the same object than more distant pairs, but this factor is clearly not as compelling as the others.

The combinations of a given match are generated in order of the compellingness of the grouping heuristic involved: all combinations exhibiting connectedness are of highest priority, then those with good continuity, and for subsequent passes, the rest are ordered by proximity. Given this scheme, the extensions of a match are attempted in roughly best-first order. In addition, the retrieval of the desired match combinations, given each of the factors, can be made efficient by suitable image and match data base organization, as has been implemented in our system.

5. Matching experiments

Our matching system has been applied to the recognition of objects in real, digital images. For each image, the same initialization procedure was followed. First, lines were detected [5] and filtered by length (> 10 pixels) and intensity contrast (> 5 gray levels). Next, all

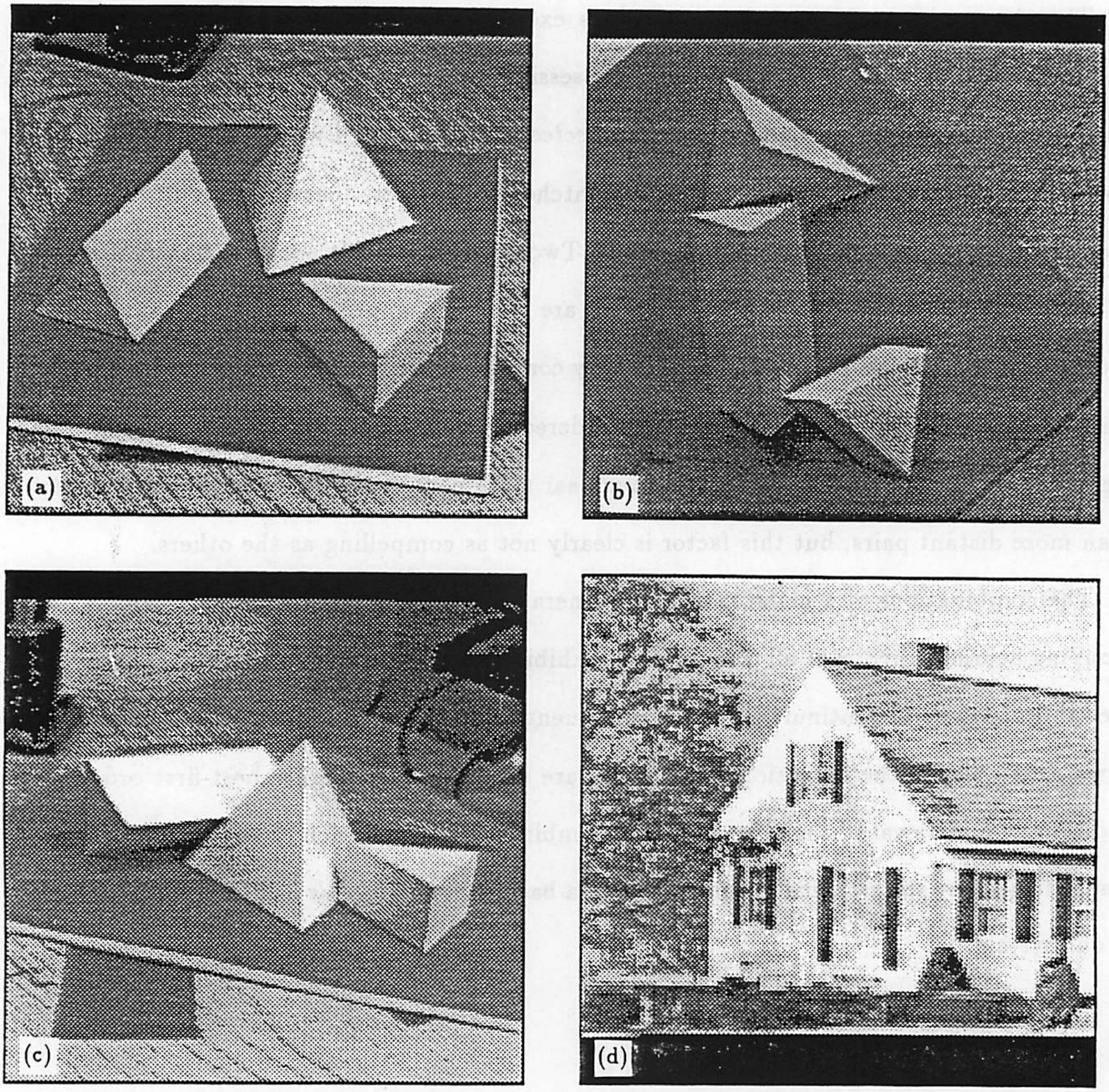


Figure 7: Images of scenes containing multiple objects and an outdoor scene. Images: (a) ***separated-objects***, (b) ***top-scene***, (c) ***side-scene*** and (d) ***outdoor-scene***

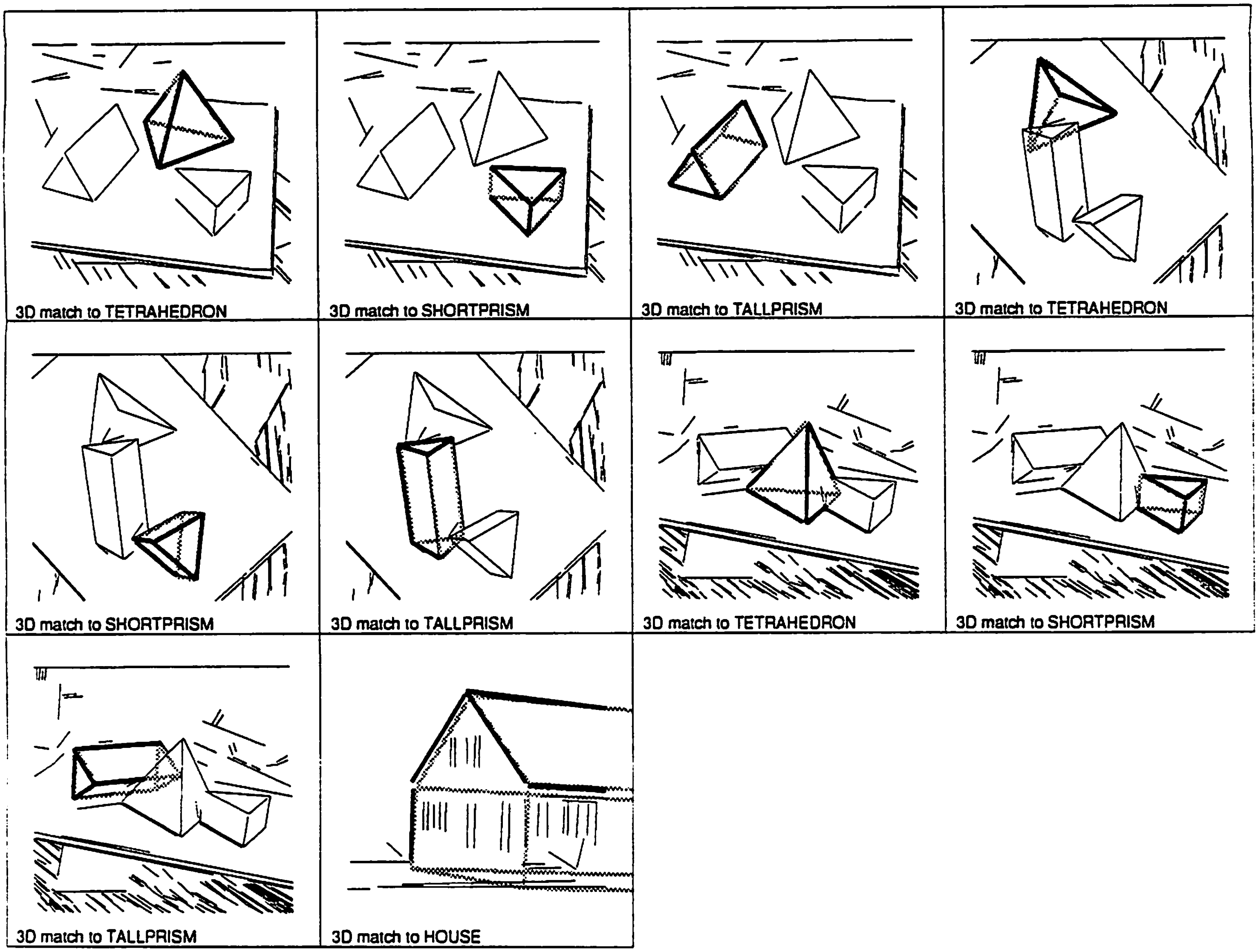
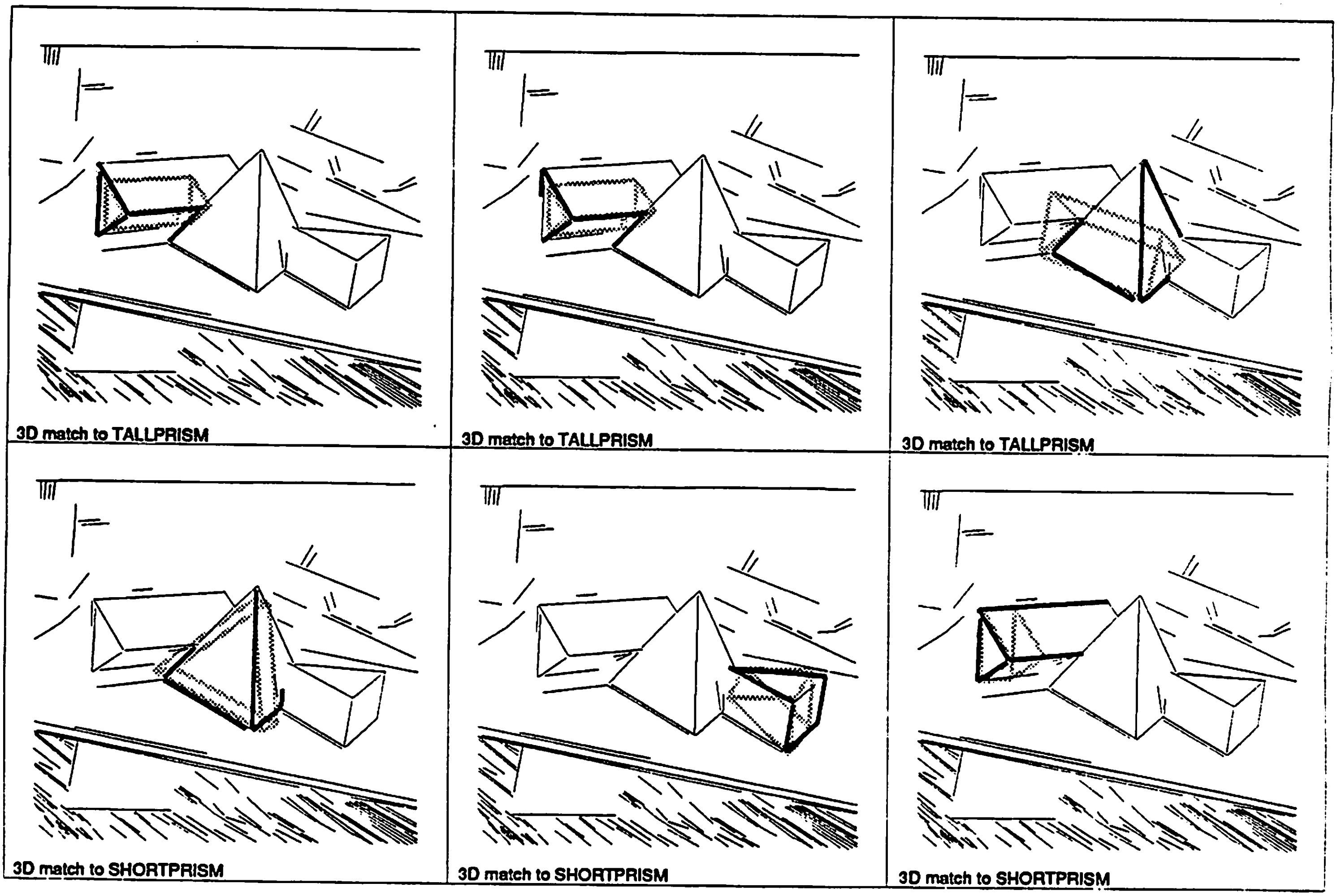


Figure 8: The resulting correct 3D matches for the images. The line segment assignments of each match was generated by the network matching process and are indicated by the thick black lines. The thick gray lines are projections of the object from the 3D pose estimated by the algorithm of Kumar, given the segment assignments shown.



3D match to TALLPRISM

3D match to TALLPRISM

3D match to TALLPRISM

3D match to SHORTPRISM

3D match to SHORTPRISM

3D match to SHORTPRISM

Figure 9: The incorrect 3D matches generated by the system and rejected by the verifier. All of the incorrect 3D matches are from the image *side-scene*; the matches are represented as in the previous figure.

matches to CORNER with low gap error (< 12.5% of line length) and low overshoot of the intersection (< 3.5%) were found. Finally, redundant CORNER matches due to redundantly represented object edges were removed by the following filter: if a line segment is matched to CORNER multiple times, and the other segments in these multiple CORNER matches are next to and parallel with each other, then select only the best one.

The first three images in Fig. 7 are of scenes containing multiple objects. These were matched to a network for the three objects in the scenes; it was identical to the one shown in Fig. 2, without the house view descriptions. The system was also applied to the last image in Fig. 7 of an outdoor house scene. The system searched for matches between this image and all four objects shown in Fig. 7, using the view description network shown in Fig. 2. All of the correct 3D matches were found for each image; Fig. 8 shows the original hypothesized 3D matches as thick black lines and the resulting projection given the estimated 3D pose in thick gray lines. The system also hypothesized and attempted to verify some incorrect 3D matches, shown in Fig. 9. Note that there were no incorrect 3D matches generated for the images *separated-objects*, *top-scene* and *outdoor-scene*, and only six for *side-scene*. The first four of the incorrect matches were due to false line-segment junctions and accidental parallels in projection (or shadows), which produced erroneous small 2D matches of high probability. In spite of such complications in the image, the matching system as a whole seems effective.

Table 1 shows some useful statistics for each matching trial and the average across all of them (last column). For the images and objects studied here, a confident 3D match typically requires the assignment of five model line segments, making the possible 3D matches per

Statistics	Images				
	Separate-objects	Top-scene	Side-scene	Outdoor-scene	Average
object library size	3	3	3	4	3.25
image segments	63	77	142	41	80.8
3D matches possible	2.5×10^{9}	7.1×10^{9}	1.6×10^{11}	3.6×10^{8}	4.3×10^{10}
3D matches generated	3	3	9	1	4.0
3D matches per correct match	1	1	3	1	1.6
number of match iterations	4	4	5	4	4.25
ave height of network	4.7	4.7	4.7	4.8	4.73
iterations per network level	.86	.86	1.07	.83	.91
3D and 2D matches generated	230	264	379	137	252.5
ave # line segs per match	2.6	2.5	2.3	1.9	2.37

Table 1: Statistics for matching experiments. Each column reports the statistics for a different image, except for the last column which is the average over all runs. The statistics are explained in the text.

image number in the tens of billions. In spite of this, the number of 3D matches actually hypothesized is very small, averaging 1.6 per correct 3D match. Another illuminating statistic is the number of iterations that the system runs before finding all of the correct matches. Including the initialization and final verification steps, the average number of iterations is 4.25. In relation to the average height of the view description network, this is a good result. The average height of the network roughly represents the number of steps in a network-directed construction of the 3D match, starting from an unmatched set of image line segments. For the network used in this study, the average height is 4.7, which is higher than the actual average number of iterations used by the system to generate the correct 3D matches. In part, this reflects the fact that the system sometimes performed match extensions to *indirect* successors in the network and thus avoided some of the construction steps. It also reflects the utility of the match combination approach in general. As argued in Section 2, the evidence from matches to multiple parts and the convergent structure of the description network are used together to provide a potentially focused search for the correct interpretation. The results presented here help demonstrate this potential.

The total number of 3D and 2D matches generated also seems reasonable. This is especially true when one considers the number of line segments in the image and the average size of the matches (number of assigned model lines). In the trials presented here, matches to the two simplest 2D models, LINE-SEGMENT and CORNER, make up the majority of the total, and another large portion of the total is made up of matches to 2D models that are almost as simple: the three-segment matches to U-SHAPE, C-SHAPE and TRIANGLE. For the images tested, the system consistently converges on the correct interpretation without generating many 2D or 3D matches of size greater than three segments – even though there can be many initial matches to the smaller 2D models.

6. Summary and conclusions

In our approach to recognition, the search for matches to a multiple 3D object library is optimized by matching *view description networks* that are automatically compiled from the library. The contribution of the research described in this paper is the development of effective strategies for the network matching process. The experiments show that a recognition system based on view description networks is capable finding the correct matches to 3D objects in complex images with a potentially high level of efficiency.

It is important to note that, even though our compilation process generates view descriptions for rigid 3D objects, the view description network and the recognition phase of the system are based on relational graphs. They can thus be readily used to represent and match information about *non-rigid* objects. This adaptability to non-rigid domains is a key property of our system.

REFERENCES

[1] Biederman, I. (1985) "Human Image Understanding: Recent Research and a Theory", CVGIP 32, p29-73

[2] Binford, T.O., T.S. Levitt, W.B. Mann, "Baysian Inference in Model-Based Machine Vision", Proc. AAAI Uncert. Work., 1987.

[3] Ben-Aire J. (1990) "The Probabilistic Peaking Effect of Viewed Angles and Distances with Application to 3D Object Recognition", IEEE PAMI 12.8, p760-774

[4] Brooks, R.A., "Symbolic Reasoning among 3D Models and 2D Images", AI, vol. 17, pp. 285-348, 1981.

[5] Boldt, M., R. Weiss and E. Riseman (1989) "Token-based Extraction of Straight Lines", IEEE SMC v.19.6. pp. 1581-1594.

[6] Burns, J.B. and L.J. Kitchen (1987) "Recognition in 2D images of 3D Objects from Large Model Bases using Prediction Hierarchies", Proc. IJCAI-10.

[7] Burns, J.B., R. Weiss and E. Riseman (1990) "View-variation of point-set and line-segment features", DARPA IUW, Also to appear in IEEE PAMI (1992)

[8] Burns, J.B. (1991) "Matching 2D Images to Multiple 3D Objects Using View Description Networks", PhD. Thesis, UMass Amherst.

[9] Clemens, D. and D. Jacobs (1991) "Model group indexing for recognition", Proc. IEEE CVPR, pp. 4-9.

[10] Draper, B., J. Brolio, B. Collins, A. Hanson, and E. Riseman (1989) "The Schema System", IJCV, v.2.3, p.209-50.

[11] Gigus, Z. (1990) "Computing the Aspect Graph for Line Drawings of Polyhedral Objects" IEEE PAMI, v.12.2, pp. 113-122.

[12] Grimson, W.E.L. (1991) *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, Cambridge, MA.

[13] Heller, A. and J. Stenstrom (1989) "Verification of recognition and alignment hypothesis by means of edge verification statistics", Proc. DARPA IUW, pp. 957-966.

[14] Jacobs, D. (1991) "Optimal matching of planar models in 3D scenes ", IEEE CVPR, pp. 269-275.

[15] Ikeuchi, K. (1987) "Precompiling a Geometrical Model into an Interpretation Tree for Object Recognition in Bin-picking Tasks", DARPA IUW, pp. 321-339.

[16] Kumar, R. and A. Hanson (1989) "Robust Estimation of Camera Location and Orientation from Noisy Data Having Outliers", IEEE Work. on Interp. of 3D Scenes, pp. 52-60

[17] Kumar, R. and A. Hanson (1990) "Sensitivity of the Pose Refinement Problem to Accurate Estimation of Camera Parameters", IEEE ICCV, pp. 365-369.

[18] Lamdan, Y. and H. Wolfson (1988) "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme", Proc. IEEE ICCV, pp. 238-249.

[19] Lamdan, Y. and H. Wolfson (1991) "On the error analysis of 'geometric hashing' ", Proc. IEEE CVPR, pp. 22-27.

[20] Lowe, D. (1985) *Perceptual Organization and Visual Recognition.* Kluwer Academic, The Netherlands.

[21] Rock, I. (1985) *The logic of perception*, MIT press, Cambridge MA.

[22] Stein, F. and G. Medioni (1990) "Efficient 2D Object Recognition", Proc. ICPR, pp. 13-17.

[23] Swain, M (1988) "Object recognition from a large database using a decision tree", Proc. DARPA IUW, pp.690-696.

[24] Wayner, P. (1991) "Effeciently using invariant theory for model-based recognition", IEEE CVPR, pp. 473-478.