

Using Perceptual Organization to Extract 3-D Structures

RAKESH MOHAN AND RAMAKANT NEVATIA, SENIOR MEMBER, IEEE

Abstract—We describe an approach to perceptual grouping for detecting and describing 3-D objects in complex images and illustrate it by applying it to the task of detecting and describing complex buildings in aerial images. We argue that representations of structural relationships in the arrangements of primitive image features, as detected by the perceptual organization process, are essential for analyzing complex imagery. We term these representations *collated features*. The choice of collated features is determined by the generic shape of the desired objects in the scene.

The detection process for collated features is more robust than the local operations for region segmentation and contour tracing. The important structural information encoded in collated features aids various visual tasks such as object segmentation, correspondence processes (as in stereo, for example) and shape description.

Our method initially detects all reasonable feature groupings. A *constraint satisfaction network* is then used to model the complex interactions between the collations and select the promising ones. Stereo matching is performed on the collations to obtain height information. This aids in further reasoning on the collated features and results in the 3-D description of the desired objects.

Index Terms—Aerial images, automated mapping, object segmentation, perceptual organization, shape representation, stereo.

I. INTRODUCTION

OUR goal is to detect and describe interesting three-dimensional structures in complex scenes. To do so, we need robust methods of obtaining good segmentation and of inferring the 3-D structure. The performance of such a system is likely to depend critically on the kinds of image features that are employed. The traditional approaches use edge contours or regions as their primary features. In edge-based methods, local edges are detected and then linked into contiguous curves. These curves typically do not give complete boundaries for complex objects and many curves that correspond to texture, surface marking, and noise are present. Many attempts have been made to connect these curve segments, using "contour tracing" methods, into meaningful objects. Such techniques have been successful for relatively simple scenes but fail in more complex environments. Region segmen-

tation techniques do give closed regions, by construction, but the regions often do not correspond to the objects in complex environments. The key problem with these segmentation techniques is their myopic nature; they operate locally on the image intensities and do not utilize global information.

One alternative to dealing with these fragmentation segmentations is to use "model-based" techniques; a survey can be found in [1]. Model-based techniques usually rely on *a priori* knowledge of the objects in the scene and predict the appearance of an object to the low-level descriptions that can be extracted from the fragmented segmentation. While impressive results can be obtained for restricted domains, we believe that this approach is too restrictive when applied directly to the very low-level image descriptions in complex scenes.

The alternative we pursue in this work is that of *perceptual organization* which consists of organizing the fragmented low-level descriptions into meaningful higher level descriptions, which are in turn used by the higher level reasoning processes. The human visual system is very good at detecting geometric relationships such as collinearity, parallelism, connectivity, and repetitive patterns in an otherwise randomly distributed set of image elements [2] and we can usually see shapes in arrangements of poor machine generated edge outputs from even very complex scenes. We believe that such capabilities must also be present in machine vision systems if they are to have generality and function in complex environments.

It is our view that the primary purpose of perceptual organization is to make salient the structural interrelationships between image features. We will use the term *collated features* for a group of features. Useful collated features identify those structural relationships that characterize objects of our visual domain and remain invariant with changing viewing conditions.

We believe that collated features are useful, if not essential, for the tasks of object detection and segmentation, shape description and object matching. We believe that such groupings are needed not only before higher level processes, such as object recognition can be performed, but that they also aid intermediate processes like stereo. Stereo is sometimes believed to give unambiguous data to help in the grouping process, but we show that sometimes the reverse may be more appropriate.

Perceptual organization has been studied extensively by investigators of psychology and computer vision [2]–[13],

Manuscript received December 30, 1987; revised February 29, 1989. Recommended for acceptances by C. Brown. This work was supported in part by the Defense Advanced Research Projects Agency under contract F33615-84-K-1404, monitored by the Air Force Wright Aeronautical Laboratories, DARPA order 3119, and in part by the DMA under Contract 800-85-C-0008.

The authors are with the Institute for Robotic and Intelligent Systems, Departments of Computer Science and Electrical Engineering, University of Southern California, Los Angeles, CA 90089.

IEEE Log Number 8929951.

yet, methods that work on real, complex scenes are lacking. We have chosen the task of detecting and describing complex buildings in aerial images as a test domain. Low-level segmentation in such images produces a large number of fragmented features and previous approaches that have used local approaches fail as the scenes get complex. This is also a domain where stereo processing using low-level features, such as lines, also fails as the needed global context is not available without some grouping first. We have developed a new methodology for detecting and utilizing collated features, in a system that we call *CANC* [14], [15], that gives excellent results in this difficult domain. Our system is, however, not limited to this domain as it does not rely on the knowledge that images are aerial, but only on the knowledge that the desired structures have certain *generic* shapes.

Choice of the appropriate collated features may depend on the problem domain and system goals. In this work, our principal interest is in detecting and describing objects of a certain *generic* shape. In this case, the collated feature choices can come from the expected shape itself, and structural decompositions of the shape. Collated features appropriate for the chosen domain are given in Section II. Once the desired collated features have been chosen, computing them still remains a major problem. We must deal with the computational complexity of the task and learn to distinguish between the large number of combinations that are possible. In our approach, we first compute all reasonable feature groupings, as described in Section III. Then, a process of mutual cooperation and competition, implemented as a constraint satisfaction network, selects the more promising collations as described in Section IV. We show how these collated features can be used for stereo matching in Section V, and for visual reasoning, object segmentation and description in Section VI. Our system has been tested on several examples and some results are shown in Section VII. We give the time complexity and run times for our system in Section VIII and present our conclusions in Section IX.

II. A VISUAL DOMAIN

We have chosen the task of detecting and describing complex buildings in suburban aerial images to test and demonstrate our methodology for detecting and utilizing collated features. However, our system uses no specific knowledge that it is looking at an aerial scene and we believe that the methodology has much broader applicability. We address this in Section IX, after we have described our method in detail.

The task of detecting and describing buildings in natural scenes is a difficult one and is best illustrated by an example. Fig. 1 shows a stereo pair of images of a building with wings of various heights in a suburban environment. The building is easy for humans to see, even without stereo, but it is in fact very difficult for current vision systems. Fig. 2 shows the line segments detected in the image-pair (Fig. 1) using the "Nevatia-Babu line finder" [16]. We are still able to see the roof structures of the

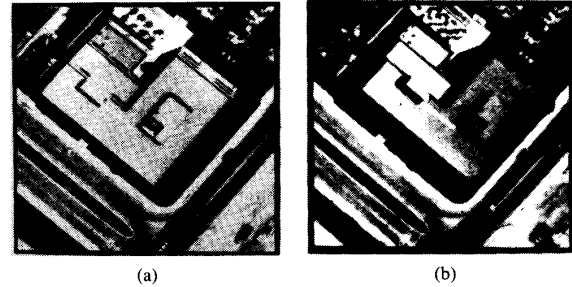


Fig. 1. Aerial image I (stereo pair): (a) right image, (b) left image.

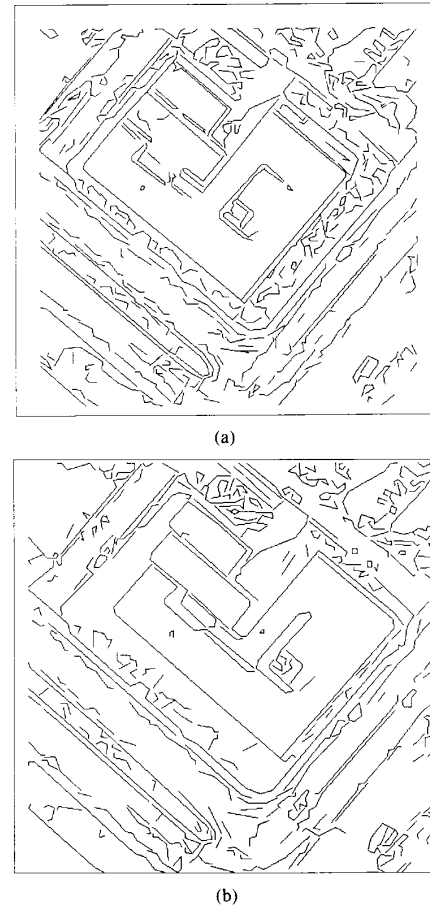


Fig. 2. Linear segments detected in Fig. 1: (a) right image, (b) left image.

buildings readily and easily, but the complexity of the task now becomes more apparent. The building boundary is fragmented, there are many gaps and missing segments. There are also many extraneous boundaries caused by other structures in the scene. While local techniques, such as "contour-tracing" have proved useful for simpler instances of such tasks [17], they are likely to fail for the scene of the complexity shown here.

This task is difficult for several reasons. The contrast between the roof of a building and surrounding structures

such as curbs, parking lots, and walkways can be low. Contrast between roofs of various wings, typically made of the same material, may be even lower. Low contrast alone is likely to cause low-level segmentation to be fragmented. In addition, small structures on the roof and objects, such as cars and trees, adjoining the sides will cause further fragmentation and give rise to “extraneous” boundaries. Roofs may also have patches on them caused by dirt or variation in material. Shadows and surface markings on the roof cause similar problems.

There are other characteristics of these images which may specifically cause problems for contour tracking type systems [17]–[19]. Roofs have raised borders which sometimes cast shadows on the roof. This results in multiple close parallel edges along the roof boundaries and often these edges are broken and disjoint. At roof corners and at junctions of two roofs, multiple lines meet leading to a number of corners making it difficult to choose a corner for tracking. Roofs cast shadows along their sides and often have objects on the ground near them like grass lots, trees, trucks, pathways etc., which lead to changes of contrast along the roof sides. Thus while tracking one can face reversal in edge direction. Often some structures both on the roof and on the ground are so near the roof that the border edges get merged with the edges of these objects, leading contour trackers off the roofs onto the ground or inside the roof. At junctions it is difficult to decide which path to take. Searching all paths at junctions leads to a combinatorial explosion of paths. It may be difficult to decide on the correct contours since contours may not close because of missing edge information, or more than one closed contours may be generated. Contours may merge roofs or roofs and parts of the ground. Figs. 2, 21, and 27 (see Section VII) illustrate some of these problems. Note the edges around the roofs of the buildings.

Stereo analysis is also difficult here. The roof tops have little texture and thus little context is available for the matching of their boundaries. In fact, the usual assumption that disparity changes smoothly, is violated in very many places, since the important boundaries largely represent depth discontinuities.

A. Previous Work

Much work has been done in computer vision on detecting buildings and other man-made structure in aerial images. While a wide variety of techniques have been applied towards this task, a systematic use of perceptual grouping has been lacking. Another interesting observation is that while man-made objects have rich geometric structure, little use of this structural information was made in the older systems.

In [20] a region segmenter is used and the relationships between such objects as roads and houses is used to improve detection. Contour tracing with some structural guidance as oriented corners and depth from shadow has been used in [17]–[19]. Fua and Hanson [21] segment the scene into regions, find edges lying on region boundaries,

and then see if there is evidence of geometric structure among these edges to classify the region as a man-made object. In the VISIONS system [22], region segmentation is the primary technique used and the regions are classified by their shape and spectral properties. SPAM [23] is a map-based system which uses region segmentation of aerial imagery.

Most of these systems work on simple scenes, for example rural scenes, where the building roof can be simply segmented (and even identified) from the surround on spectral properties. The buildings detected have simple shapes. Only a few systems compute and use depth information. None of the systems generate a description of the buildings at the level of shape descriptions of the different wings.

In the systems mentioned above, the generic feature extraction techniques used, namely region segmentation or contour tracing, are not suited for extracting particular shapes or organizations. Hough transform [24], [25] is a general mechanism for detecting groupings, but is practical only if exact shapes (rather than generic descriptions) to be detected are known. If the features being detected have simple geometric properties, it is more straightforward to use specific detection algorithms. For example, the MOSAIC system [26] uses oriented junctions to complete fragmented lines. This system also uses height information obtained from stereo and sophisticated geometrical reasoning to hypothesize likely wire frame models of the buildings. The complexity of this system, and its limited performance, are due to the use of simple features (lines and junctions) used to perform the detection, stereo matching, and reasoning. Recently, application of perceptual grouping to locate features indicating structure has been explored by Reynolds and Beveridge [27]. This system also employs specific routines to detect various geometric organizations indicative of structure. However, this system has limited use as the groupings found are sensitive to the layout of the scene rather than the object shapes, and consequently can not be used to either detect or describe any individual structures (like buildings) in the scene.

B. Choice of Collated Features

In our approach, we propose to compute features before higher level processes are applied. The choice of the collated features follows from the chosen task. We assume that roofs are the essential building structure we see, and that the roof shapes can be modeled as consisting of a combination of rectangles (note that in addition to rectangular buildings, this allows, for example, L, T, H, and E shaped buildings as well). We believe that this model applies to the vast majority of modern buildings. Such shapes can be successively decomposed into rectangles, U shape (a rectangle with one side missing), parallel lines, and straight lines. These then form the collated features that we seek. Besides the assumption that such collated features are useful, our system uses no other domain knowledge.

III. DETECTION OF COLLATED FEATURES

The detection of collated features works bottom-up from edges, grouping them step by step into more and more geometrically evolved shapes. Initially all reasonable groupings are considered as collated features; this set is then refined by identifying the *significant* collations as more global context from other collated features become available after the initial groupings are made.

The large collated features and their subsidiary collations exchange information in both directions. Simple collated features are used to form the more complex groupings which in turn help the formation of the simpler collated features by use of their more global view. This symbiotic relationship will become more evident in the following subsections.

A. Lines: Linear Structures

We use the "USC LINEAR" system, based on the "Nevatia-Babu line finder" [16] to detect linear segments in the scene. Due to poor contrast, many linear segments along the roof borders get fragmented. Near junctions or close presence of other strong features, these fragments get displaced from the straight line the border lies on, making simple collinearization useless.

A collection of parallel lines bunched along the same linear axis represents the presence of a *linear structure* at a higher granularity level than the edges (for example, the boundary of the roof as opposed to the individual lines belonging to its borders). We wish to group closely bunched parallel linear segments since they represent a linear structure of some object, like the border of a roof or the divider on a road.

To detect such groupings of edges, we "fold" the space around each segment (a segment here refers to a linear segment or a straight line fitted to the edges detected in the image) onto the segment repeatedly, like pleats in an accordion, collecting the segments from this space which lie parallel to it (see Fig. 3). This folding process is halted as soon as no new segments are located or when the threshold on the spread about the linear structure is exceeded (The width of each pleat is 0.151.¹ The spread threshold is 0.41.). Each of the groupings of closely bunched, overlapping, parallel line segments is represented by a single *line* (or linear feature) whose orientation is a length weighted average of the segments grouped and extent is between the maximum and minimum projections of the grouped segments onto this orientation. Fig. 4 shows the lines obtained from grouping the segments in Fig. 2.

We detect two types of corners between the lines, L and T-junctions. We currently do not investigate orthogonal trihedral vertices (OVT's) as few walls are visible, and those that appear highly foreshortened and have shadows, etc., near them making the OTV's difficult to detect accurately. T-junctions for urban aerial imagery do not have, in general, the usual interpretations of occlusion.

¹ 1 is the minimum length of a building side in the image.

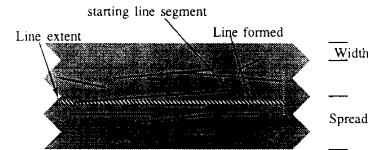


Fig. 3. Linearization.

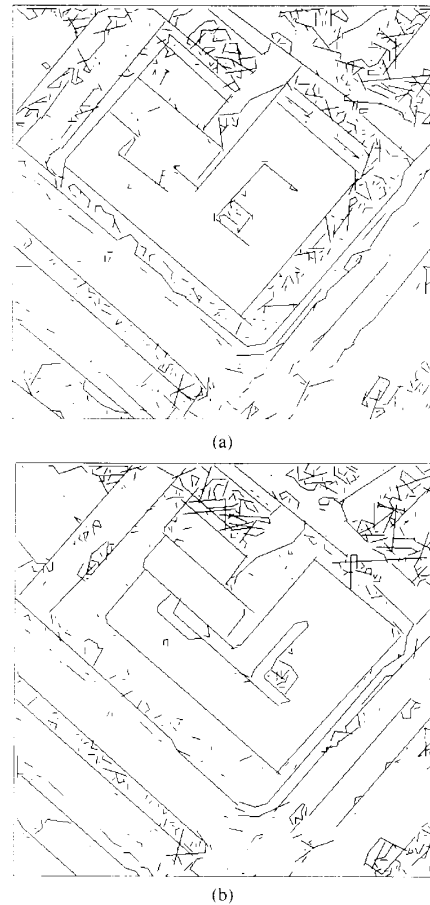


Fig. 4. Linear structures obtained from Fig. 2: (a) right image. (b) left image.

The buildings have wings which are aligned and nearby objects like roadways, etc. are also aligned to the building sides. In a top view, the sides of two different structures can create T-junctions in which the top line belongs to two different objects and is not occluding the stem. Therefore, the T-junctions are used to break the line belonging to the top of the T into sections. Finally, all the lines are extended to the corners detected at their ends.

B. Parallels

For each line obtained above, we find lines that are parallel (within ± 15 degrees) to it and have a sufficient overlap (at least 60 percent of the length of the shorter line) with it. Man-made structures in urban scenes like building-wings, roads and parking lots are organized in

regular grid-like patterns. These structures are all composed of parallel sides. As a consequence, for each significant line-structure detected in the scene, there is not one but numerous lines parallel to it. To reduce the number of parallels we impose the additional constraints that the length of the longer line of the parallel pair should be no longer than three times the length of the shorter one and if a line l_1 finds a parallel line l_2 on its (say) left side which completely overlaps it (l_1) then no lines parallel to l_1 on its left side lying further than l_2 are considered.

The formation of the *parallel* collated features in turn aids the formation of lines. The structure of two parallel lines strongly suggests a complete overlap between the two. If the original lines do not overlap completely, their extensions which will complete the overlap, are considered. These extension are alternate line groupings of the underlying linear segments. Often these new line collations will include new linear segments in the extension, which had not previously been included in the same linear structure as the information (in terms of proximity and collinearity) in the context of the lines alone was not sufficient to trigger the grouping. We also consider the formation of another parallel with the longer line contracted so that it just overlaps the shorter one (see Fig. 5). This is done because the line may be belong to a longer, occluding object, or we may have erroneously grouped segments from two close by structures while finding lines. One can see the new lines formed by extension by noting the lines in Fig. 4 which have been extended in Fig. 6 (which displays both the lines of the parallel groupings detected). However, the lines formed by contraction can not be made out as they are overlaid by their longer alternates.

C. U Structures

Each pair of parallel lines evolves into a set of parallels and the ends of their component lines are aligned (within ± 0.71). A set of parallel lines with aligned ends is a strong indication that there is possibly a line joining those ends to create a U shaped structure.

Thus the presence of a parallel with aligned ends triggers the formation of another collated feature, the U structure. The U collation, or rather the parallel with aligned ends, gives strong suggestion of a line joining the two ends (see Fig. 7). If an appropriate line joining the aligned ends does not already exist, a new linear collation is created. Note that due to the formation of the U collation, lines which were not combined into a single line on the relatively local structural basis of collinearity may now be grouped on the more global structural basis of the U-contour (the gap tolerated between line segments is now increased up to the width of the gap between the parallel lines and the angle tolerance for collinear lines increased by 50 percent). This new collation may incorporate any existing linear segments or may be "virtual"; again the perception of a complex structure, in this case a U, triggers the formation of a less evolved collated structure, the line. Fig. 8 shows the U-contours detected.

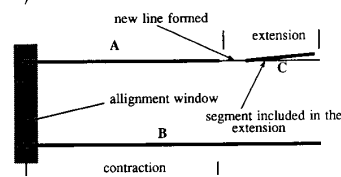
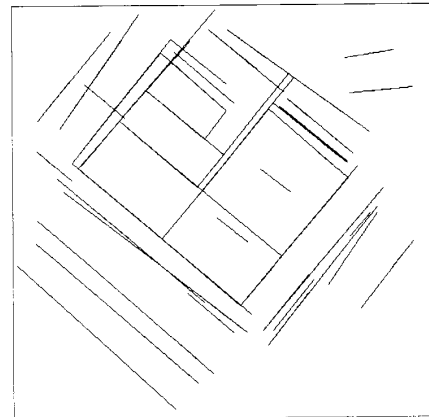
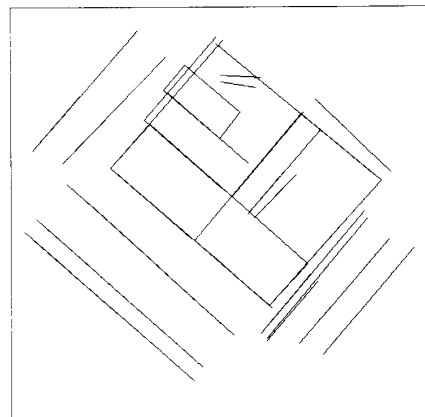


Fig. 5. Parallel collation.



(a)



(b)

Fig. 6. Parallels: (a) right image, (b) left image.

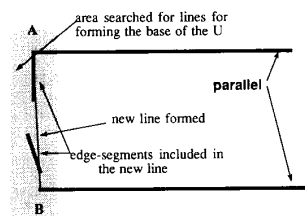


Fig. 7. U-contour collation.

D. Rectangles

Each parallel generates two U structures and the U's of a parallel taken together form a rectangle. Fig. 9 shows the rectangles formed from the U-contours in Fig. 8. The

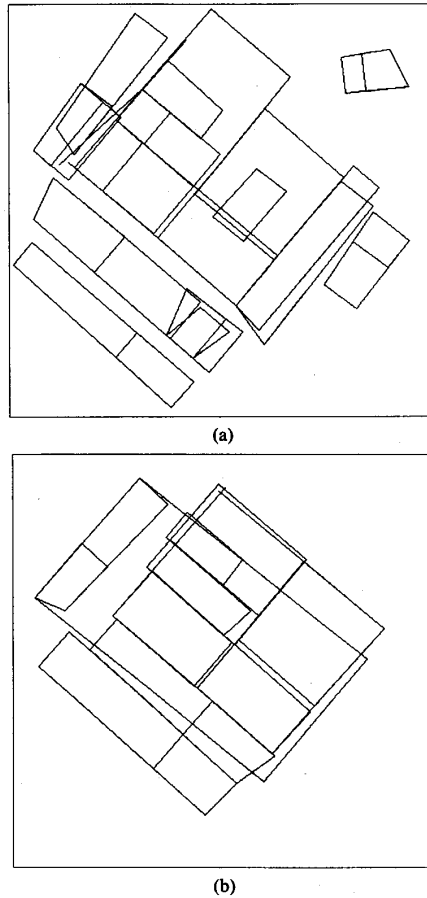


Fig. 8. U structures: (a) right image, (b) left image.

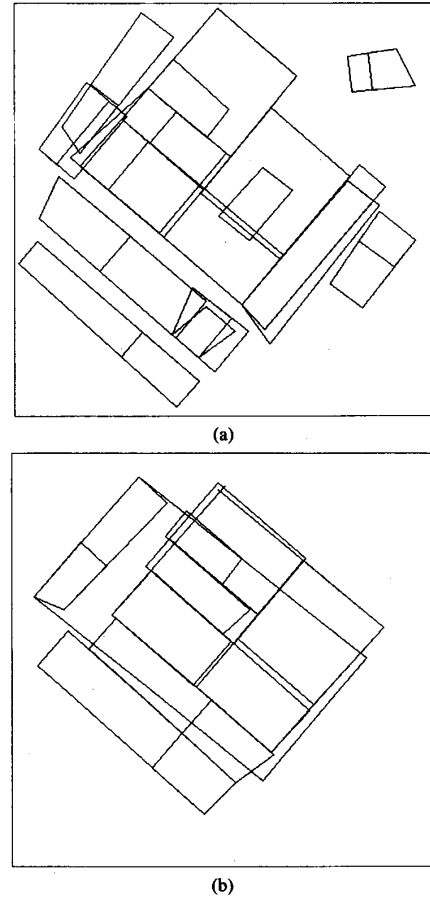


Fig. 9. Rectangles: (a) right image, (b) left image.

formation of a rectangle proposes the presence of presence of two other U-contours, orthogonal to the U-contours that formed the rectangle. If any of these U-contours have not already been formed, new U-contours are generated.

The thresholds used in the detection phase are robust; the same thresholds were used for all the images, including the ones displayed in this paper, that we processed.

IV. SELECTION OF COLLATED FEATURES

The detection of collated features, where all reasonable groupings among tokens resulted in the formation of collated features, is followed by selection where only the more suitable collated features are retained. The detection and selection processes could proceed simultaneously.

At each level of the collated features hierarchy, various collations are in contention as they provide alternative groupings of the underlying tokens. Also, some collations may have been formed on weak evidence; evidence that seems too weak when compared to that for other collated features at that level. A selection process has to choose "good" collations, i.e., those which have high probability of corresponding to individual object parts.

The "goodness" of a collated feature depends on how it compares to its alternatives in terms of the support it has from related collations at other levels, and the support or contradiction from its component primitive features and other related image features. A collated feature is not supported just by its component collations but also by the collations it itself is a component of. The later relationship is due to the fact that the percept of a larger structure strengthens that of a smaller component structure, for example as the percept of a U shape strengthens the percept of a line forming the base of the U. Thus a line and the U it belongs to are mutually supportive. In general terms, collated features which are linked by part-of relationships are mutually *supportive* and those that share component collations are mutually *competitive*.

The problem of selecting the best set of collations can be formulated as selecting the best set of hypotheses, given relationships of support and conflict among them. Consider:

A set of **Hypotheses** $H = \{h_i\}$.

A unary function **Value** $V(h_i) = V_i$, $0 \leq V_i \leq 1$, which assigns a confidence level to each hypothesis.

A binary function **Support** $S(h_i, h_j) = T_{ij} V_i V_j$, $T_{ij} >$

0 and its value depends on the support relationship between two hypotheses.

Similarly, a binary function **Conflict** $C(h_i, h_j) = T_{ij}V_iV_j$, $T_{ij} < 0$.

A unary function **Input** $I(h_i) = I_i$ which is a sum of the evidence or measurements for the hypothesis h_i .

Finally a function $E(h_i) = I_iV_i$, which measures the contribution to the total **evidence** from hypothesis h_i .

To choose the best consistent set of hypothesis that maximize evidence, we wish to assign confidence values to hypotheses such that: $\Sigma S + \Sigma C + \Sigma E$ (note the terms in C are negative) is maximized. Rewriting the above term we have $\Sigma \Sigma T_{ij}V_iV_j + \Sigma I_iV_i$. Our goal is to find the *optimal* feature groupings consistent with the known optical and geometrical constraints [28], [29]. Note that all the constraints must be *simultaneously* satisfied to reach global consistency across all levels of the hierarchy.

One parallel technique to solve this problem is relaxation where a cost function associated with the network is minimized. We wish to select the best *consistent* feature groupings, and reject the bad groupings. If we formulate the cost function such that the optimal solution corresponds to its global minima, then the problem of locating the best groupings reduces to that of optimizing the cost of the network given the constraints (defined by the relations) between the collated features and the observed image characteristics. Parallel optimization techniques such as simulated annealing [30], Hopfield networks [31], [32], Boltzman machines [29], [33], [34], probabilistic solutions [35] and connectionist methods [33], [36] have been proposed for such problems.

A. Constraint Satisfaction Networks

The collated features and the relationships of support and conflict among them naturally define a network with the collations serving as nodes and the relationships as arcs. We use a slightly modified version of Hopfield networks to implement this network as a constraint satisfaction network. Following the notation convention of Hopfield and Tank [37], [38] we describe the behavior of each node in the network by

$$du_i/dt = -u_i + \sum_{j=1}^N T_{ij}V_j + I_i - h_i \quad (1)$$

$$V_j = g(u_j) \quad (2)$$

$$h_i = \text{resting potential of node}_i \quad (3)$$

where N is the total number of nodes, T_{ij} is weight on link from node j to node i , I_i is the total input to node i , V_i is the output of node i , and u_i is the "membrane potential" of node i . The gain function g is sigmoidal and is defined as $\frac{1}{2}(1 + \tanh(u_i))$. The addition of h_i , the resting potential or bias, is useful in adjusting the sensitivity of a neuron by shifting its gain curve. For purposes of analysis of the network, the resting potential may be combined with the input.

When the network has symmetric connections, i.e., $T_{ij} = T_{ji}$, the network, where each element has the above equation of motion, converges to stable states. This property has also been shown for more general networks by Hummel and Zucker [39]. When the gain function g is high gain (width of the gain curve is narrow), the stable states of the N elements are the local minima of the following cost function with the outputs of the nodes at 0 or 1, [38].

$$E = -\frac{1}{2} \sum_i \sum_j T_{ij}V_iV_j - \sum_i V_iI_i. \quad (4)$$

Note that minimizing the above term is equivalent to maximizing the term given the previous subsection (after replacing T_{ij} by half its value in that term). The signs in the above cost function suggest that if we wish to select mutually supporting collations and reject mutually conflicting collated features, the weights T_{ij} between supporting hypotheses should be positive and that between conflicting hypotheses should be negative. Those optical and geometrical constraints, which are not expressed purely via the interrelationships between the interpretations should be fed as inputs I_i to the nodes. Again, the sign in (1) and (4) shows that supporting evidence should be included as positive input and contradicting evidence as negative input.

B. Construction of the Network

To construct the network, each of the collation detected is represented as a node or "neuron." The relationships between the collations define the links between the nodes. In the network in Fig. 10, nodes for collated features which *support* each other are connected via *positively* weighted links (thin lines) while mutually *conflicting* collations are linked via *negatively* weighted links (bold lines).

To ensure the selection of *perceptually significant* feature groupings in the scene, the choice of weights should reflect the perceptual importance placed on the optical and geometric constraints between the various collated features. The perceptual significance of a collated feature lies in its indication of actual object structure in the scene. For example, while any grouping of parallel lines [27] is indicative of some order in the scene, we are more interested in parallels that actually correspond to individual objects. Therefore, the parallels that have supporting structural evidence such as rectangles are more significant than those that do not.

Supporting links are between those collated features at *different* levels of the feature hierarchy which group the same underlying edges, i.e., collations that are connected by "part-of" relationships. For example, there is a supporting link between a parallel and each of the two lines that form the parallel. If the parallel is part of a rectangle, then there is a supporting link between the parallel and the rectangle. The relationship of support is also inherited, for example when a rectangle is formed from two U-contours, it also forms supporting links with the parts of

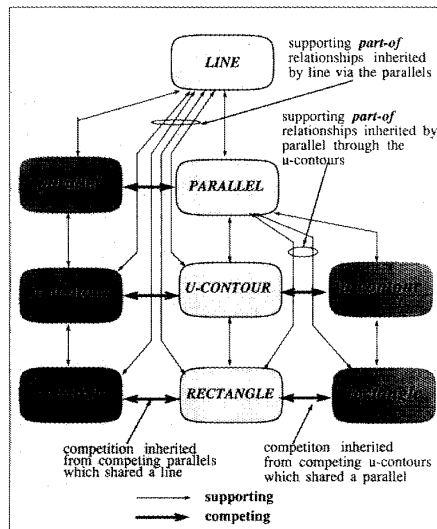
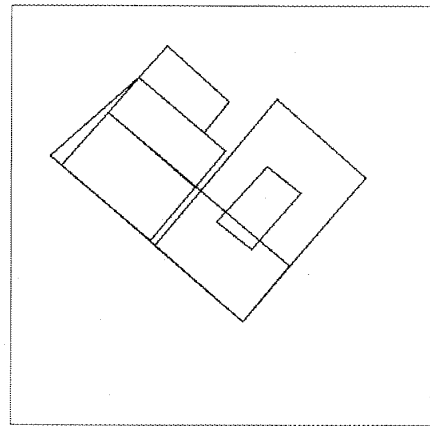


Fig. 10. Constraint satisfaction network.

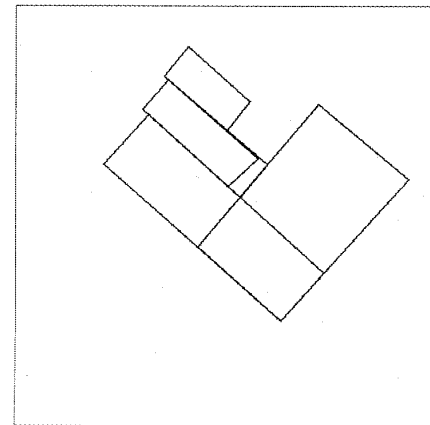
the U-contours, i.e., the parallels and the lines. The supporting links are generated as a part of the collation detection process; once a collation is detected, links of support between it and the collations it grouped are formed. For example, when a U-contour is detected, a node for it is generated and is linked to the nodes of the parallel and the line (base of the U) that were grouped to form it.

Conflicting links are formed between collated features at the *same* level of hierarchy, which are alternate groupings of the same edges. For example, when two lines of dissimilar length are found to be parallel, we form two parallel collations, one corresponding to the shorter line and the other to the longer line. These two parallel collations are in conflict and would be connected by a conflicting link. Relationships of conflict are also inherited. If two rectangles are formed from two conflicting parallels, then the two rectangles are also in conflict. The conflicting links are formed after all the collations are formed. The relationship of conflict is first found between collations at the lowest level of the hierarchy, i.e., lines, and then we progress up the hierarchy, with each level inheriting conflicts from its components in the lower levels.

The input to each node is a weighted sum of measurements on the collation represented by that node. The input to a node representing a *line* collation is the weighted sum of the percentage of its length actually covered by edges, the numbers of corners detected at its ends and the number of lines crossing through sections of it which are gaps ($I_{line} = \%edge - coverage + 0.4 \times \#corners - \#crossings$). The relevant measures on *parallels* are the amount of overlap between the lines, and the width of the parallels, on *U-contours* are the number of actual corners between the base and the parallel sides and the number of lines crossing the base, and on *rectangles* are the number of corners between its sides, and the "amount of texture" inside the rectangle (we count the number of lines lying inside a rectangle, which do not belong to any rectangle,



(a)



(b)

Fig. 11. Rectangles selected by CSN: (a) right image, (b) left image.

as a measure of the texture on the rectangle, with less "textured" rectangles being preferred).

C. Performance of the Network

The network is started with all nodes at rest, i.e., membrane potential and output at zero and the network is relaxed. The network converges in a few iterations and the nodes with high output (> 0.8) are selected. We have found the network to converge within ten iterations for all of our test scenes. While feature groupings at all levels of complexity get selected simultaneously, only the rectangles so selected have been displayed in Fig. 11.

The weights on the links range from -1.0 to 1.0 . We chose the weights in the proportion to the perceived importance of the source collation as supporting or conflicting evidence to the destination collation. The weights used in our system are as shown in Table I (the table is to be read as T_{ij} being the weight on the link from node j to node i). The input to each node is also represented as a real number. The evaluation of the input as a real number depends on the type of measurements made for each collation, and the relevance of the measurement to the collation. For example each corner contributes 0.7 to the input

U-M-I

Due to a lack of contrast between text and background, this page did not reproduce well.

TABLE I
INTERCONNECTION WEIGHTS FOR THE CSN

Weights	Line	Parallel	U-contour	Rectangle
Line	-	0.5	0.5	0.6
Parallel	0.6	-	0.3	0.8
U-contour	0.7	0.6	-	0.7
Rectangle	0.7	0.7	0.8	-

TABLE II
EFFECT ON SELECTION OF RECTANGLES BY CHANGE IN WEIGHTS

Change in weights	Number of rectangles selected	number of correct rectangles missed
0%	10	none
+100%	9	none
+200%	9	none
-50%	8	none
-75%	5	none

to the rectangle; indicating that the presence of a corner (a corner can be interpreted as two lines in the specific geometric relationship of orthogonality) is considered to be of the same order of significance as a parallel with full confidence level.

In our implementation, the weights on the links are not symmetric, so the convergence results for Hopfield networks cannot be used. However there is support that the networks can converge under nonsymmetric weights [40]. We have found our networks to converge on all our selection of weights within ten iterations.

In the beginning, some tuning of the weights, and the bias on the nodes, was done on one image so that the nodes that fired corresponded to the collations that we visually perceived as "good." After this initial setting of weights, no further tuning was done, the same set of weights (shown in Table I) have been used successfully for all the images we processed. We have found that the network is not sensitive to even large changes in the weights if the total amount of evidence ($\sum_{j=1}^N T_{ij} V_j + I_i$) arriving at the nodes does not change drastically. Table II shows the performance of the network in selecting rectangle collations, from those displayed in Fig. 9(b), against percentage change (from the values in Table I) in the weights. The last row of the table may indicate that when the weights are too low, some of the nodes, which may fire with higher weights, may not be able to rise above their threshold to fire. The network, in its present form, does not have the ability to "self calibrate," i.e., to automatically adjust the sensitivity of a node on the basis of the total amount of information arriving at it. The sensitivity of the nodes can be controlled by using a bias similar to the resting potential of neurons. By controlling the

TABLE III
TIME COMPLEXITY AND RUN-TIMES (IN SECONDS) FOR THE IMAGE DISPLAYED IN FIG. 1(b). THE LAST COLUMN GIVES THE ACTUAL NUMBERS FOR THIS IMAGE. THE I/O COLUMN STATES THE TIME REPORTED AS USED FOR I/O RELATED ACTIVITIES (THE FIGURES FOR RUN-TIMES INCLUDES THIS TIME).

Collation	Complexity	Run Time	I/O	Comments
Line	$O(ls)$	64.4	25.1	$s = 774, l = 12.3$
Parallel	$O(n^2)$	92.24	21.6	$n = 83$
U-Contour	(npu)	13.6	2.3	$p = 48, u = 35$
Rectangle	$O(np)$	0.8	0.6	
Network formation	$O(np^2)$	162.9	12.52	
Relaxation	$O(np^2)$	83.1	13.8	
Stereo	$O(r^2)$	7.2	2.1	$r = 10$ (left), 8 (right)
Structure	$O(r^3)$	1.9	0.0	analysis made on 4 rectangles

bias we control the amount of positive evidence required by a node to fire it.

This scheme, since it is based on competition between alternate collations for the same underlying edges, may lead to a situation where a collation, which is not comparable to the other selected collations of its hierarchy class in the scene (in terms of its related measurements or evidence), is selected solely because it is the only grouping of its component edges and has no competing collations. To avoid this situation, a "winner take all" [41] type of network is superimposed on the Hopfield-Tank network. In this network, each node has self excitation (+1) and competes (-1) with *all* the other collations of the same hierarchy class. While stability results for such "winner take all" type of networks are known [42], the stability results for the resulting "hybrid" network are not known.

It may be useful to note here that unlike many other applications of "neural-networks" to computer vision [33], each node represents a high-level feature. The features to be represented, and the relationships between them, have been selected by us. This, however, does not require the network to be made by hand; the nodes and their relationships get automatically formed as a byproduct of the detection process.

V. APPLICATIONS TO STEREO

Collated features are rich representations. They encode particular structural relationships at a particular scale of description. Matching collated features thus involves less ambiguity than edges as there are less possible alternatives and more information to judge a match. Also there are usually much less collated structures, at any given representation level than edges. The most probable role of collated features, and one that we employ here, is that correspondence of collated features provides a rough correspondence for their component primitive features, which can then be matched with less ambiguity. In a similar vein, recent stereo systems have shown improved performance by using more structure than individual edges [43]-[47].

For our vision system we use the rectangle collated features to aid stereo matching. For this visual domain, edge and segment based stereo matching algorithms displayed poor performances. The following factors indicate why stereo systems based on simple image features may not perform well in this domain.

- *Organized nature of the scene.* There are numerous parallel lines since the building-wings, roads, parking lots, etc., are all parallel. Roof borders and their shadows and road markings also give rise to close parallel lines. This leads to many ambiguous matches and it is difficult to resolve among the competing matches.

- *Absence of texture.* The buildings sides represent areas of high disparity change and there are insufficient markings on the roofs to support match-disparities at roof level while matches giving low disparities get favored due to the preponderance of features on the ground.

We choose to match the rectangles selected in each image of the stereo pair by the constraint satisfaction network. Finding a match between two rectangles corresponds to assigning a unique one-to-one correspondence between the sides of the rectangles. Two rectangles A , B match if their sides match in order, i.e., the leftmost side of A matches the leftmost side of B , and so on. Therefore, in Fig. 12, rectangles A and B match if a_1 matches b_1 , a_2 matches b_2 , a_3 matches b_3 and a_4 matches b_4 .

For two sides a_i and b_i to match, part of b_i has to lie within the epipolar window of a_i . The epipolar window of a_i is defined as the space bounded by the epipolar lines corresponding to the ends of a_i . Note that requiring only a part of b_i to lie in the epipolar window of a_i (instead of the stricter requirement of having the ends of b_i lie exactly on the epipolar lines of the ends of a_i) allows for occlusion of b_i . For our domain, we assume that the roofs are nearly parallel to the imaging plane. Therefore the two images of the side of a building in the two image of the stereo pair must appear parallel. Thus two sides a_i and b_i have to be nearly parallel to match.

The sides of a rectangle need not have equal disparities. If the four sides have different disparities, then some of the sides of the rectangle do not lie on the same roof as the other sides, but belong to occluding roofs. This allows for matches even when portions of a roof may be occluded by other roofs. However, if two side a_i and a_j of a rectangle form an L-junction, they belong to the same roof, whether the roof corresponds to the area enclosed by the rectangle, or an occluding roof, and their disparities must be equal.

The conditions for two rectangles A and B to match can be summarized as

$$\text{match}(A, B) \Leftrightarrow \left(\bigwedge_{i=1}^4 \text{match}(a_i, b_i) \right)$$

$$\text{match}(a_i, b_i) \Leftrightarrow \text{parallel}(a_i, b_i) \wedge \text{epipolar} \\ - \text{overlap}(a_i, b_i)$$

$$L - \text{junction}(a_i, a_j) \rightarrow (\text{disparity}(a_i) = \text{disparity}(a_j)).$$

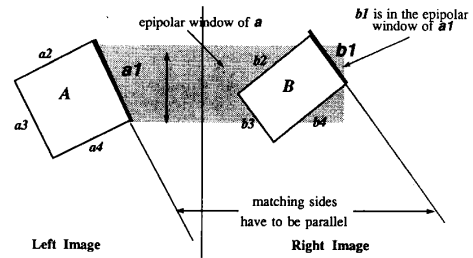


Fig. 12. Stereo matching of the rectangle collated feature.

The disparity assigned to the rectangle is the minimum disparity of its sides (assuming that disparity increases with increase in height above ground). This means that we assume that the side(s) with the least height belongs to the area enclosed by the rectangle and the other sides belong to occluding roofs. However, this assumption fails if the sides enclose a hole in the roof.

The choice of rectangle as the match primitive restricts the possible matches. Like other stereo matching systems we allow only matches falling within a disparity range reasonable for the stereo pair. To avoid mistaking rectangles corresponding to tennis courts, parking lots, and the like, the legal disparity range should start just above ground level. The other end of the interval should be high enough to encompass the tallest buildings in the scene. This estimate need not be exact, as possible wrong matches between rectangles usually result in disproportionate disparities. For our test cases we chose an ad hoc value which was more than twice the disparity of the tallest building in any of the test scenes.

The key problem with general stereo systems is the ambiguity in matching necessitating a mechanism to choose one among many competing matches for each match primitive. For this system we have found the constraints imposed by the structure of the collated feature sufficient to select unique matches for the primitives (rectangles). In the rare case of a rectangle finding more than one match, we choose the match with the least number of disparity differences between the sides, which is equivalent to preferring the least occluded interpretation.

Stereo serves as an important visual clue in selecting those collated features which have a very good chance of corresponding to actual object structures, in this case the roofs. Selection of the proper collated features is crucial for this domain as many other objects in the scene such as road segments, parking lots and sidewalks have rectangular structures. Furthermore, these objects are arranged in a regular grid like manner, and some collated features formed reflect the structure in the layout of the scene rather than that of specific objects. In general, objects in a scene are not organized in a regular fashion, and other sources of visual information such as stereo may not be required for aiding the selection process.

The rectangle collations which are components of the roof shapes have heights above the ground, and the disparities of their sides lie within ranges reasonable for the

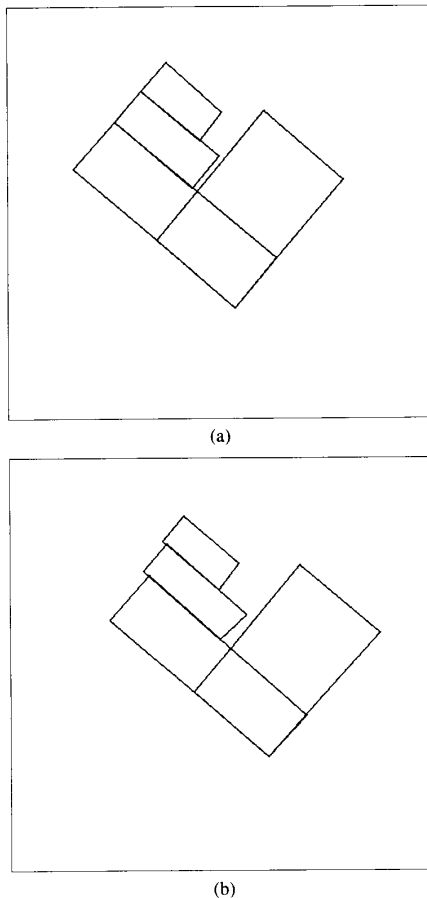


Fig. 13. Rectangles matched by stereo: (a) right image, (b) left image.

stereo pair. The rectangle collated features that meet this criteria are selected out from the rest. These have high probability of belonging to roofs or parts of roofs and further processing is performed on only these rectangles. Rectangle-groupings so selected are shown in Fig. 13.

A. Discussion

The present version of the system *CANC* has the drawback of using stereo to select among the existing collations but of not using it to check for missed collations. Given a rectangle in one image, which does not have a match in the other, the system does not go back to the grouping level, with relaxed constraints on the grouping (or the selection) to see if a matching rectangle could be detected. For example, in the scene in Fig. 1, the small roof inside the L-shaped roof is detected in the right image but not in the left image. Thus it is rejected, and does not appear in the final model of the building.

There is a loss of accuracy in the determination of the disparities as a result of the robustness in the detection of the matched primitives. The rectangles are collated features, and are thus primarily structural representations with low positional accuracy. The component lines of the

rectangle only represent the structure among the underlying edges, not their positions. For obtaining accurate disparity, matching of more precisely located features, namely the edges, is required. The lines of the rectangles are replaced by the linear segments they represented and these are then matched. There is little ambiguity in matching linear segments at this stage, since the linear segments of a line are matched only to the linear segments of the corresponding matching line found during matching the rectangles. While each line might represent multiple close parallel linear segments, we choose only the innermost edges for match. The rectangles represent areas bounded by the lines and so the inner edges are a sensible representation of the rectangle. This is a convenient approximation and may not be correct for all cases. We are currently working on using sensitive edge detectors on magnified portions of the image in small windows around the lines for precise detection and location of edges. We can consider even weak edges near the noise level of the image, since we have an idea of the direction of the edges, their geometry (straight lines), and an approximate idea of their location.

VI. APPLICATIONS TO SHAPE DESCRIPTION AND OBJECT EXTRACTION

The collations detected correspond to parts of objects in the scene. We have to combine the collated features into structures corresponding to the objects. The combination process automatically generates a shape description of the object in terms of the primitive shapes of the collations combined.

In the present case, we have identified strong rectangle collations which meet the height requirements of buildings in the scene. However, each rectangle collation may not correspond to a separate roof as a roof shape could be a combination of rectangles. To extract individual roofs in the scene (object extraction), we have to consider possible combinations of the rectangles into structures which correspond to roofs. As the shape of a roof is described as a combination of rectangles, this process, in addition to *segmentation* also provides *shape description*.

The combination of the rectangle collations is guided by reasoning based on the available 2-D and 3-D information. The visual reasoning carried out currently is primarily monocular, augmented by stereo as needed. The 2-D information is the geometrical relationships between the rectangles and the actual edges grouped by the rectangle collations. The 3-D information is obtained by stereo (see Section V). The combination process is rule-based; the set of rules governing the combination of the rectangles (and the resultant structures) is defined on the 2-D and 3-D relationships among the rectangles.

In contrast to previous uses of monocular analysis, we work with more organized structures than lines and junctions. Also T junctions, which are a key element in monocular analysis, cannot be utilized for this application domain because of the presence of false T junctions due to alignment. As with other phases of processing, the

reader will note that the organized nature of the primitives used for processing bring more information to the monocular analysis than is available with just edge and junction information.

The structural relationships being considered are those of *subsumption* or *inclusion*, *merger-compatibility*, *occlusion*, and *incompatibility*. Let the shapes be defined by their boundaries. Consider two structures A and B . The bounding contours of the structures correspond to groupings of the underlying intensity edges. Sections of the boundary thus correspond to grouped edge-contours, single edge-contours or are abstractions generated by the grouping process.

We find the intersections between the boundaries of A and B . These intersections divide the boundary of each structure into *contour-segments*. The contour segments of each structure are then assigned to one of three disjoint sets, one containing segments that lie *outside* the other structure, one containing segments that lie *inside* the other structure and one containing segments *shared* by the other structure. Since the positioning of the boundaries is approximate, allowances have to be made during the computation of these sets to account for these inaccuracies (for example close parallel and overlapping boundaries may be termed "shared").

O_{AB} : Set of contour segments of A outside B .

I_{AB} : Set of contour segments of A inside B .

S_{AB} : Set of contour segments shared by A and B , $S_{AB} \equiv S_{BA}$.

$Edg(X)$: Number of edges in the set X of contour segments.

Subsumption: If the outside-segment set of shape A is empty and the shared segment set nonempty and the edge-support for segments in the inside-segment set is poor or nonexistent, then we say that structure A is subsumed by structure B and can be removed (see Fig. 14).

$$(O_{AB} = \phi) \wedge (S_{AB} \neq \phi) \wedge (Edg(I_{AB}) < \tau) \\ \Rightarrow B \text{ subsumes } A.$$

The following relationships are only checked when subsumption is not present.

Occlusion: If the contour-segments of A inside B have strong edge support and those of B inside A have weak intensity edge support then we can assume that shape A occludes shape B (see Fig. 15). Note that this applies even if the rest of the contour-segments of A and B belong to the shared set or outside set.

$$(Edg(I_{AB}) > \tau) \wedge (Edg(I_{BA}) < \tau) \Rightarrow A \text{ occludes } B.$$

Merger-Compatibility: If the segments in the inside-segment and shared-segment set for both shapes A and B have poor edge support then we can conclude that A and B represent segmentation of one structure into two parts and can thus be merged into one structure (see Fig. 16). The merger operation is that of union. Note that an implicit assumption has been made that the outside-segment

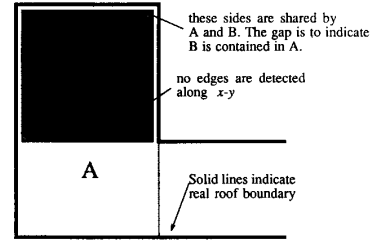


Fig. 14. Subsumption.

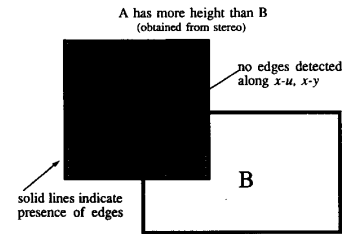


Fig. 15. Occlusion.

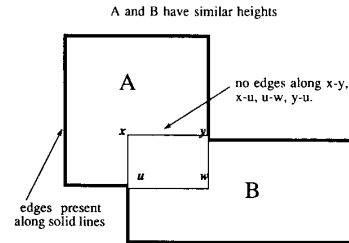


Fig. 16. Merger-compatibility.

set of A is nonempty since each shape has to have some edge support for it to be formed. The one combination of rectangles resulting from this process in Fig. 13(b) is shown in Fig. 17.

$$(Edg(I_{AB}) < \tau) \wedge (Edg(I_{BA}) < \tau) \wedge (Edg(S_{AB}) < \tau) \\ \Rightarrow A \cup B.$$

If on the other hand, the outside-segment set of A is empty and the shared-segment set of A has poor edge support then A and B are merged using the difference operation $B - A$.

$$(Edg(I_{AB}) > \tau) \wedge (I_{BA} = \phi) \wedge (O_{AB} = \phi) \\ \wedge (Edg(S_{AB}) < \tau) \Rightarrow B - A.$$

If stereo information is present, it should be checked that the heights of A and B are compatible since edge support could be lacking in shared-segments of adjoining objects of similar surface properties due to the absence of contrast.

Unrelated: If A and B have null inside-segment sets and null shared-segment sets then they are unrelated. If A and B have a nonempty shared segment set (and null inside-segment sets) but the shared segments have good edge

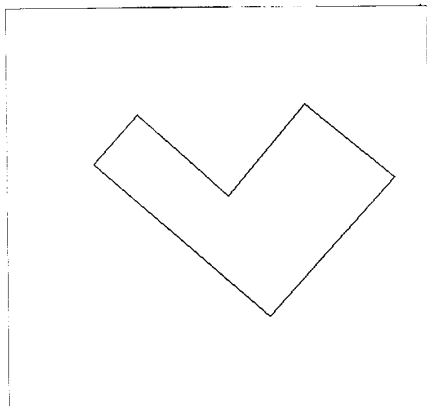


Fig. 17. Only possible combination of rectangles in Fig. 13(b).

support, then A and B are still unrelated (although adjoining).

$$(I_{AB} = \phi) \wedge (I_{BA} = \phi) \wedge ((S_{AB} = \phi) \vee (Edg(S_{AB}) > \tau)) \Rightarrow A \text{ and } B \text{ are unrelated.}$$

Incompatible: If A and B have nonempty inside-segment sets and the elements of the inside-segments of both A and B have strong edge support then at least one of A or B is a wrong structural grouping and must be deleted.

$$(I_{AB} \neq \phi) \wedge (I_{BA} \neq \phi) \wedge (Edg(I_{BA}) > \tau) \wedge (Edg(I_{AB}) > \tau) \Rightarrow A \text{ and } B \text{ are incompatible.}$$

The decision of which structure is erroneous is difficult to make in the context of just two structures, one could possibly retain the structure with more edge and corner support than the other. If A and/or B conflict with other structures then the one with the most conflicts can be deleted first, and so on. In the case of a tie, where we are left with a pair of mutually conflicting structures, other information such as stereo could be used to resolve conflicts.

Our current system reports on the conflicts but does not resolve them. In our test cases we had only one case of conflicting structures, and there the decision was easy to make (manually) as one of the structures was not only conflicting with a number of other structures but was also being "occluded" by a structure of lower height (as reported by stereo) than itself.

Starting from the rectangles selected from the previous stages, we perform the above analysis on all pairs of rectangles, first removing subsumed structures and then forming new structures on any possible mergers of rectangles. The process is recursively applied to the new structures along with the original structures from the previous step until no new structures are formed. During this combination, duplication of the structures is possible, but it is trivial to detect duplicate structures since they have exactly the same component rectangles. The roofs so obtained from the rectangles in Fig. 13 are shown in Fig. 18.

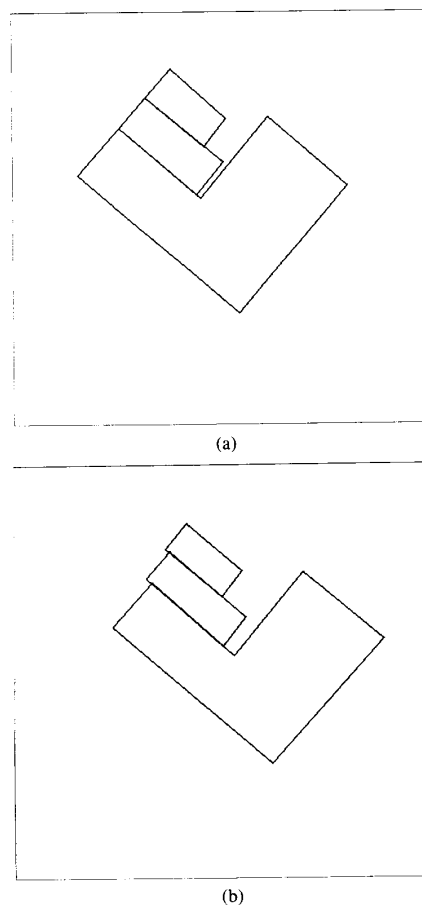


Fig. 18. Final combination of rectangles (corresponding to roofs): (a) right image, (b) left image.

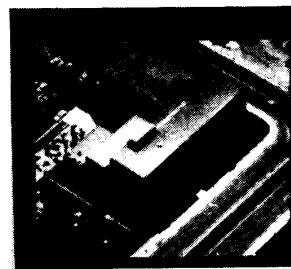


Fig. 19. Rendered view of 3-D model of the detected building.

The geometrical relationships among the shape primitives (rectangles) and their combinations form a graph which is a structural description of the objects in the scene in terms of the primitive. Structures in the graph which are not marked as subsumed, merged, or incompatible are selected as the top-level descriptions of the objects or object parts visible in the scene (roofs for our image domain).

The final structures are assigned heights from the disparity information previously obtained by stereo. The

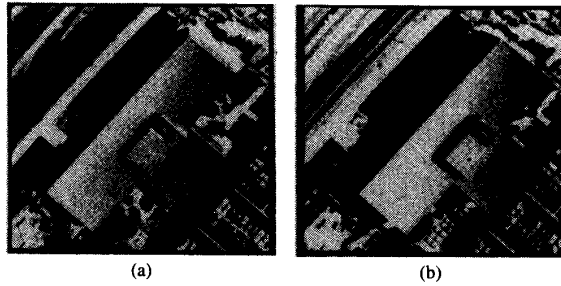


Fig. 20. Aerial image II (stereo pair): (a) right image, (b) left image.

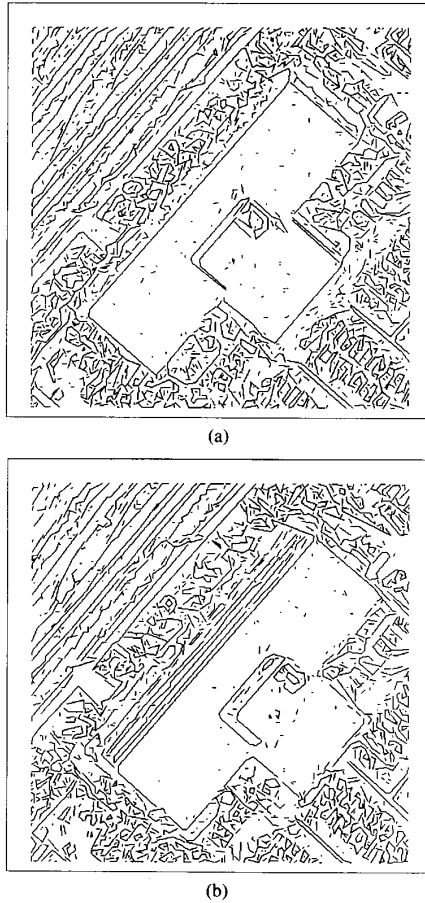


Fig. 21. Linear segments detected in aerial image II: (a) right image, (b) left image.

buildings are modeled by drawing walls straight down from the sides of the roofs to the plane below, be it of another roof or the ground. The resulting model is displayed in Fig. 19.

VII. MORE RESULTS

We show results² on two more examples to illustrate the range and robustness of our systems. Note that these

²Some intermediate results such as the parallel and U-contour collations detected, are not displayed.

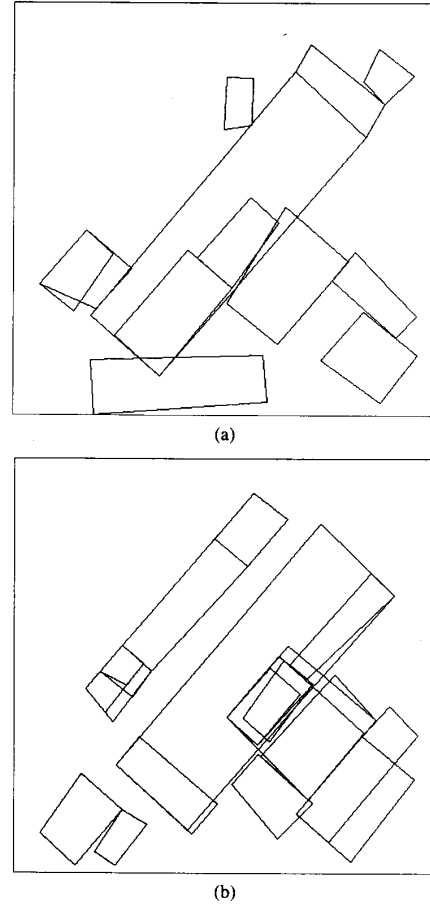


Fig. 22. Rectangles selected by CSN: (a) right image, (b) left image.

examples were processed without changing any of the parameters (thresholds or weights) of the process at various stages.

For the image pair in Fig. 20, the linear segments that were grouped are displayed in Fig. 21. Note the problems in the low-level segmentation for this scene. The edges detected around the boundary of the roof have numerous gaps and the edge contours along the roofs wander off into the ground nearby. Some of the edges on the roof boundaries are displaced and we also have multiple, nearby edges on the boundaries (note the right-hand side of the roof). For the structure on top of the roof, the edges for one side were not detected due to low contrast. Any vision system which just used image based segmentation, but did not use structural information, would not be able to recover that structure. If we compare Figs. 22 and 23, we see that many rectangles that were selected by the constraint satisfaction network as good collations, did not correspond to roofs (Fig. 24), and that stereo was able to correct all these mistakes. This illustrates that stereo matching is a strong source of information for selecting meaningful collated features. Thus, we are able to use the constraint satisfaction network primarily to select likely

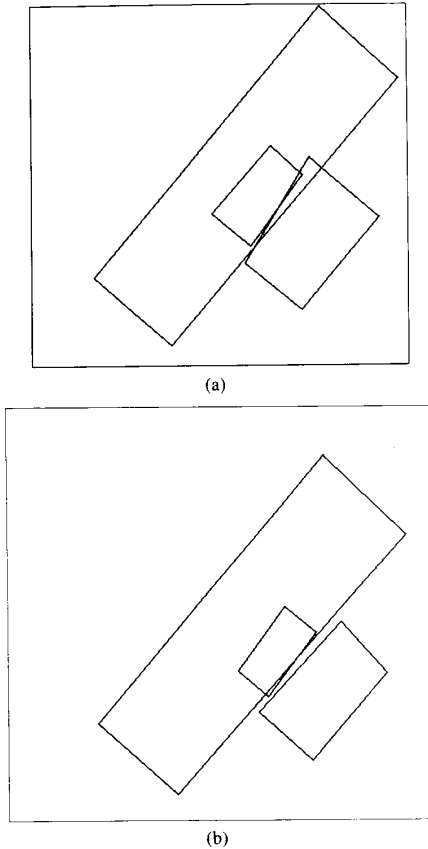


Fig. 23. Rectangles matched by stereo: (a) right image, (b) left image.

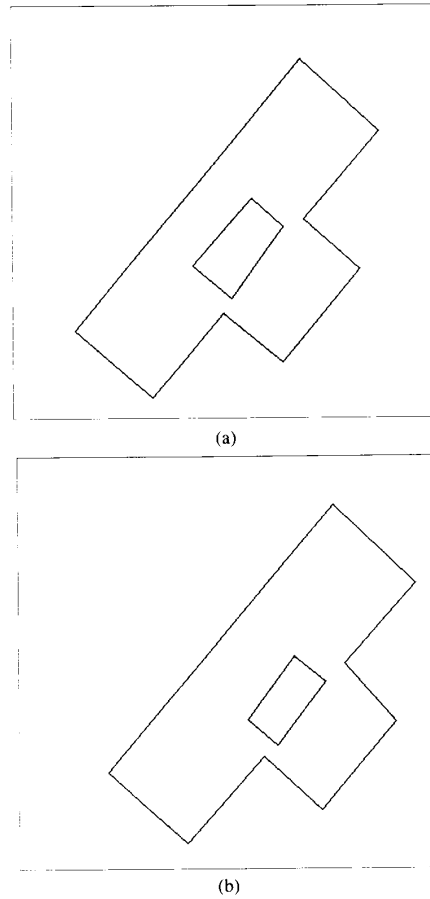


Fig. 24. Final combination of rectangles (corresponding to roofs): (a) right image, (b) left image.

collations, but are not dependent on it to identify exactly the right collations. (Fig. 25 shows a rendered, orthographic projection of the 3-D model our system generated of the building detected.)

In the third image pair (Fig. 26), the large number of linear segments detected (Fig. 27) shows another problem faced by systems based on low-level features. The number of edge-contours or linear segments typically detected in complex scenes is very large, and it is very time consuming to reason on all of them. For example, in a model matching scheme, which attempts to match object models directly to edges (or to simple related features such as straight lines, contours, or corners), rather than to more abstract descriptors, the numbers of matches to be considered would be prohibitively large. For this scene, Fig. 28 shows the rectangles selected by CSN, Fig. 29 displays the rectangles matched by stereo which also correspond to the roofs finally detected (as none of the rectangles merge). The final 3-D model is displayed in Fig. 30.

VIII. COST AND COMPLEXITY ANALYSIS

An informal analysis of the time complexity of the process is presented. Due to the large reduction in the number of the edge features in to collated features, the number of features is small in real terms, for operations such as

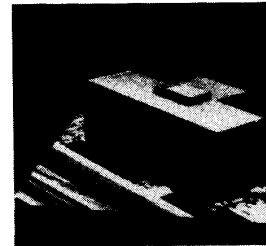


Fig. 25. Rendered view of the 3-D model of the building detected in aerial image II.

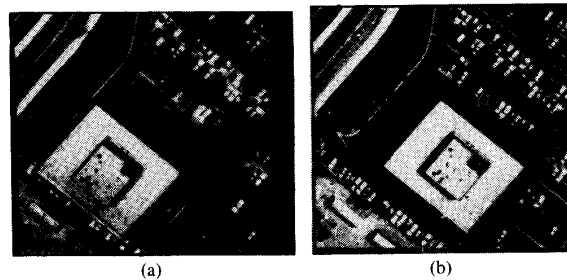


Fig. 26. Aerial image III (stereo pair): (a) right image, (b) left image.

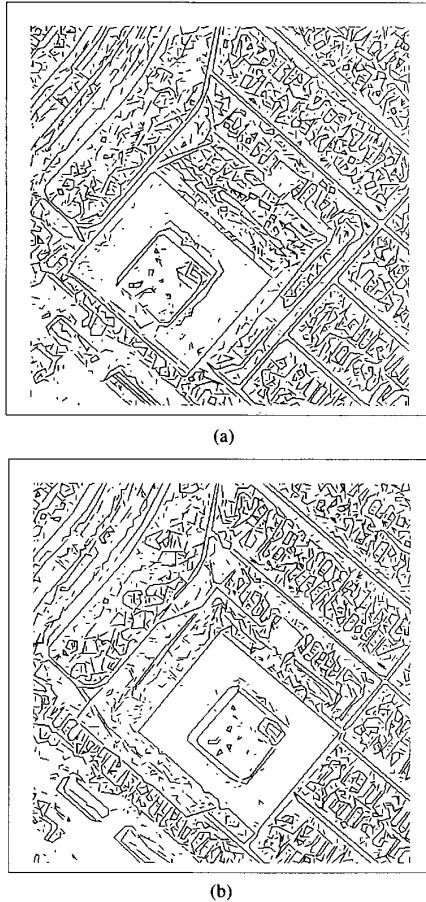


Fig. 27. Linear segments detected in aerial image III: (a) right image, (b) left image.

stereo and structural analysis. We have provided the number of features and processing time (on a Symbolics lisp machine) for the image in Fig. 1(b). This analysis is summarized in Table III.

A. Collated Features

- *Lines*: Let s be the number of line segments. Let l be the average length of each line. Each line searches a constant width (8 pixels each side) of area around it. Let m line segments be identified for each linear segment for grouping into a line. Since a line segment which is already a part of a linear grouping, does not itself look for linear groups, $O(s/m)$ groups are formed. For each group, the area search takes $O(l)$ time, and the average weighted line representation takes $O(m)$ time. Therefore, forming lines takes $O((sml)/m)$ or $O(ls)$ time, where $l \ll s$. We shall assume that finally n lines are formed from s linear segments, where $n \leq s$.

There were 774 linear segments detected in the image ($s = 774$) and after grouping 685 lines were formed ($n = 685$). The average length of these lines was 12.3. However, most of the lines were short, since we consider

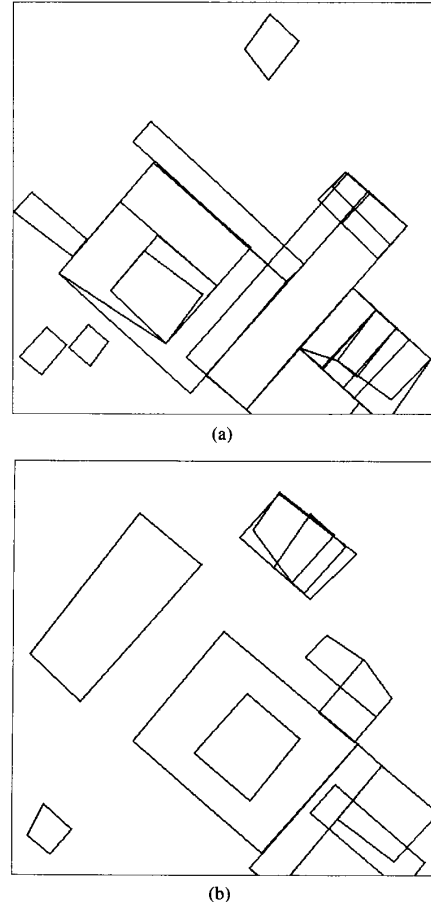


Fig. 28. Rectangles selected by CSN: (a) right image, (b) left image.

only lines of some minimum length (20 for these images), only 83 lines were considered so $n = 83$ for the rest of the processes [for the image in Fig. 1(b)]. The time taken for the grouping was 64.4 seconds of which 25.1 seconds were i/o related.

- *Parallels*: Each line looks at every other line to check for parallelism. Therefore, forming parallels takes at most $O(n^2)$ time.

Only 48 parallels are formed. This number is much smaller than 83^2 because the use of heuristics, such as overlap requirements, strongly limits the number of parallels formed. Finding the parallels took 92.24 seconds of which 21.6 seconds were i/o related.

- *U-Contours*: Let there be an average of p parallels for each line. Thus, there are at most np U-contours formed. For each U, in the worst case, a strip of area between the aligned ends has to be searched for the presence of lines. Let u be the average distance between the ends. Forming U-contours takes $O(np u)$ time.

Forty-four U-contours were formed in 13.6 seconds (2.3 seconds were i/o related).

- *Rectangles*: The two U-contours of a parallel form a rectangle. Each rectangle takes constant time to set up

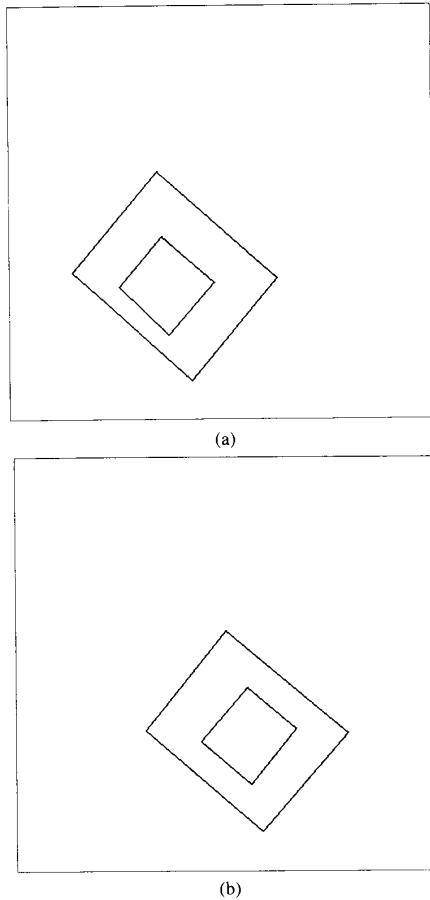


Fig. 29. Rectangles matched by stereo (none of the rectangles merge, the final shapes formed, corresponding to the roofs is the same): (a) right image, (b) left image.

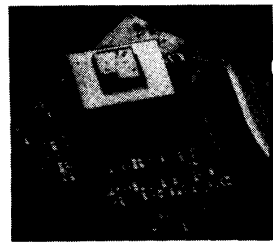


Fig. 30. Rendered view of the 3-D model of the building detected in aerial image III.

data structures, etc. Therefore processing rectangles takes $O(np)$ time.

Nineteen rectangles were formed, taking 0.8 seconds (0.6 seconds were i/o related).

Formation of collation takes $O(ls) + O(n^2) + O(np)$ time. l and u do not increase with n (given images of similar complexity and resolution) and $l \ll s$, $u \ll n$. p will increase with increase in n , but $p < n$ due to the use of heuristics for reducing the number of parallels formed. Therefore, the time complexity of forming

the collations, as a simple approximation, is no more than $O(n^2)$.

The total time to form the collations is 171 seconds (of which 49.6 seconds are i/o related).

B. Constraint Satisfaction Network

Each collation is fixed to a fixed number of collations by part-of relationships (for example, a rectangle is connected to at most four lines, two parallels and four U-contours). For each collation, there are at most p alternates.³ For each iteration, the time taken is, therefore, $O(p(n + np + np + np))$ or $O(np^2)$. The network is relaxed for a constant number of iterations.

Time taken to form the constraint satisfaction network, including the identification of competing collation for each collated feature formed, and computation of the value of input for each collation, was 162.9 seconds of which 12.52 seconds were i/o related. The time taken for 10 iterations of the network took 83.1 seconds of which 13.8 seconds were i/o related (a major part of this time is really taken up to do graphics indicating the progress of the relaxation process).

C. Stereo

Let there be $O(r)$ rectangles that get selected in each image by the CSN. The stereo takes at most $O(r^2)$ time.

Stereo matching took 7.2 seconds (2.1 seconds were i/o related).

D. Structural Analysis

In each image there are at most $O(r)$ rectangles which have suitable matches. Each rectangle's structural relationship is considered with the other rectangles. This takes $O(r^2)$ time. The structures formed by the combination number at most $O(r)$. They are combined again and again till no more combinations take place. Although the number of combination steps will take no more steps than the number of rectangles in the most complex roofs, in the worst case⁴ the number of combinations steps could be $O(r^3)$.

There were only 4 rectangles with valid stereo matches. Structural analysis on these took 1.9 seconds.

The total time complexity is $O(n^2) + O(np^2) + O(r^3)$.

To the above times, we could include for 40.8 seconds (of which 36.4 seconds were i/o related) to read in the linear segments detected by LINEAR, and setting up the data structures and graphic display. Thus processing time, as assigned to the left image of the stereo pair in Fig. 1, is 463.3 seconds (of which 98.1 seconds are i/o related). The processing times (and number of features) for the right image, and the other images displayed in this paper, were comparable.

³The alternatives for a line may be slightly higher, as some alternates may be constructed at the base of U-contours. However, the order of alternates is in the same range.

⁴Combined structures can share rectangles, therefore the combination step could take more than $O(\log r)$ steps.

IX. DISCUSSION AND CONCLUSIONS

We have proposed *collated features* as the representations computed by the process of perceptual organization applied to the primitive image elements. These collations represent structural relationships between the arrangement of their tokens. We have identified the structural relationships so represented, in terms of their significance for the shapes in our visual domain and the utility to other visual processes. Specifically, we have demonstrated that collated features are useful for stereo and the generation of shape descriptions and object segmentation.

Collated features are suited for performing mid- and high-level visual tasks such as the following.

- *Shape Description*: Object shapes are described in terms of component shapes. This decomposition of shape description follows a structural basis where individual shape components are "well formed" and have "simpler" individual descriptions than that of the combined shape. The collation of features identifies such individual structures that are useful in describing overall shapes. Collated features form small local structural descriptions. These can be combined to give global object descriptions, or be transformed to other descriptions more suitable for a particular visual process. As an example, roofs shapes are naturally decomposed into rectangles. The rectangle collated feature employed by us, gives local descriptions of the roofs, and this description provided by the rectangles is combined to generate a shape description of the roofs.

Other features such as regions or edge-contours have no associated shape description (we may be able to find a shape description for a region or a contour, but this is added effort, these features are not generated with any shape information directly associated to them), and are not helpful in providing a shape description of any object they are used to detect.

- *Active Vision*: By the identification of significant features, collated features can act as *attentional mechanisms* for detailed visual inspection, guiding the detection of features at interesting locations in greater detail. For example, after identifying roof boundaries, we may be able to use special or expensive edge detectors at these boundaries, rather than over the complete image.

- *Matching*: In such correspondence processes such as stereo, motion, and model matching, in recent systems, improved performances have been obtained by using more abstract features. This also, usually results in a significant reduction in the computational expense of matching [2], [43], [45]–[47]. Collated features are more structurally evolved representations than edges, thus there is much less ambiguity in matching collated features than is in matching primitive features. We have been able to show in our system the use of collations for stereo matching in a domain where traditional techniques performed poorly.

Even though we have presented examples from the aerial scene domain only, it should be clear that the system is not restricted to this domain. The system should work

equally well in other domains where the shapes are still composed of rectangular components; and this is a large set in man-made environments. Indoor scenes should, in fact, be easier to handle as they typically contain mostly man-made objects with regular shapes. We believe that our methodology can be easily extended to detect any other shape also, as long as the generic shape (or the set of shapes) is known in advance. Of course, we would need to write the routines to detect new collated features and construct a new instance of the constraint satisfaction network.

This approach, however, does not directly generalize to unknown shapes. Humans, of course, are able to perceive shapes in fragmented and noisy data *without* being told what shapes to expect (although the performance may degrade in comparison to the situation where we know what to expect). One may hypothesize that generality can be achieved by just having a set of specific shapes and that any shape is either one among the set or can be constructed from a combination of the shapes in the set. It may be that a not too large set consisting of common, regular geometrical shapes will suffice for a very large class of scenes in man-made environments. In current research we are pursuing an alternative approach [48], however. We seek a small set of collated features that have wide applicability. We believe that features with curvilinearity and symmetry properties are among this set. We expect to use methods very similar to those described here to choose among the alternative collations.

REFERENCES

- [1] T. O. Binford, "Survey of model based image analysis systems," *Int. J. Robotics Res.*, vol. 1, no. 1, 1982.
- [2] D. G. Lowe, *Perceptual Organization and Visual Recognition*. Hingham, MA: Academic, 1985.
- [3] D. G. Lowe and T. O. Binford, "Perceptual organization as a basis for visual recognition," in *Proc. AAAI-83*, Washington, DC, Aug. 1983.
- [4] A. P. Witkin and J. M. Tenenbaum, "On the role of structure in vision," in *Human and Machine Vision*, Beck, Hope, and Rosenfeld, Eds. New York: Academic, 1983, pp. 481–543.
- [5] S. E. Palmer, "The psychology of perceptual organization: A transformational approach," in *Human and Machine Vision*, Beck, Hope, and Rosenfeld, Eds. New York: Academic, 1983, pp. 269–339.
- [6] S. W. Zucker, "Computational and psychophysical experiments in grouping: Early orientation selection," in *Human and Machine Vision*, Beck, Hope, and Rosenfeld, Eds. New York: Academic, 1983, pp. 545–567.
- [7] R. E. Kelly, P. R. M. McConnell, and S. J. Mildnerberger, "The Gestalt photomapping system," *J. Photogram. Eng. Remote Sensing*, 1977.
- [8] A. Triesman, "Perceptual grouping and attention in visual search for features and objects," *J. Exp. Psychol.: Human Perception Perform.*, vol. 8, no. 2, pp. 194–214, 1982.
- [9] K. A. Stevens, "Computation of locally parallel structure," *Biol. Cybern.*, vol. 29, pp. 19–28, 1981.
- [10] D. Katz, *Gestalt Psychology: Its Nature and Significance*. New York: Ronald Press, 1950.
- [11] B. Julesz, "Figure and ground perception in briefly presented isodipole textures," in *Perceptual Organization*, Kubovy and Pomerantz, Eds. Hillsdale, NJ: Lawrence Erlbaum, 1981, pp. 27–54.
- [12] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.*, vol. 61, pp. 183–193, 1954.
- [13] M. A. Fischler and R. C. Bolles, "Perceptual organization and curve partitioning," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 100–105, Jan. 1986.

- [14] R. Mohan and R. Nevatia, "Perceptual grouping with applications to 3D shape extraction," in *Proc. IEEE Comput. Soc. Workshop Computer Vision*, Miami, FL, Dec. 1987.
- [15] —, "Perceptual grouping for the detection and description of structures in aerial images," in *Proc. DARPA Image Understanding Workshop*, Cambridge, MA, Apr. 1988; also available as USC-IRIS Tech. Rep. 225.
- [16] R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Comput. Vision, Graphics, Image Processing*, vol. 13, pp. 257-269, 1980.
- [17] A. Huertas and R. Nevatia, "Detecting buildings in aerial images," *Comput. Vision, Graphics, Image Processing*, vol. 41, no. 2, pp. 131-152, Feb. 1988.
- [18] —, "Detection of buildings in aerial images using shape and shadows," in *Proc. IJCAI*, Karlsruhe, W. Germany, Aug. 1983, pp. 1099-1103.
- [19] A. Huertas, R. Mohan, and R. Nevatia, "Detection of complex buildings in simple scenes," Inst. Robotics and Intelligent Systems, Univ. Southern California, Tech. Rep. IRIS 203, Sept. 1986.
- [20] T. Matsuyama and V. Hwang, "SIGMA: A framework for image understanding: Intergration of bottom-up and top-down analyses," in *Proc. IJCAI*, Los Angeles, CA, Aug. 1985.
- [21] P. Fua and A. J. Hanson, "Using generic geometric models for intelligent shape extraction," in *Proc. DARPA Image Understanding Workshop*, Los Angeles, CA, Feb. 1987.
- [22] A. Hanson and R. E. Riseman, *VISIONS: A Computer System for Interpreting Scenes*. New York: Academic, 1978.
- [23] D. M. McKeown, W. A. Harvey, and J. McDermott, "Rule-based interpretation of aerial imagery," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 570-585, 1985.
- [24] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [25] D. H. Ballard, "Form perception using transformation networks: Polyhedra," *Dep. Comput. Sci., Univ. Rochester, Tech. Rep. TR 148*, 1986.
- [26] M. Herman and T. Kanade, "The 3D MOSAIC scene understanding system: Incremental reconstruction of 3D scenes from complex images," in *Proc. DARPA Image Understanding Workshop*, 1984, pp. 137-148.
- [27] G. Reynolds and J. R. Beveridge, "Searching for geometric structure in images of natural scenes," in *Proc. DARPA Image Understanding Workshop*, Los Angeles, CA, Feb. 1987.
- [28] D. H. Ballard, G. E. Hinton, and T. J. Sejnowski, "Parallel visual computation," *Nature*, vol. 306, pp. 21-26, Nov. 1983.
- [29] S. E. Fahlman and G. E. Hinton, "Connectionist architectures for artificial intelligence," *Computer*, pp. 100-109, Jan. 1987.
- [30] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.
- [31] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.* vol. 52, pp. 141-152, 1985.
- [32] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: A model," *Science*, vol. 233, pp. 625-633, 1986.
- [33] D. E. Rumelhart, McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructures of Computing*. Cambridge, MA: M.I.T. Press, 1986.
- [34] S. E. Fahlman, G. E. Hinton, and T. J. Sejnowski, "Massively parallel architectures for AI: Net1, Thistle, and Boltzman machines," in *Proc. Nat. Conf. Artificial Intelligence*, AAAI, Menlo Park, CA. Los Altos, CA: Kaufman, 1983.
- [35] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721-741, Nov. 1984.
- [36] J. A. Feldman and D. H. Ballard, "Connectionist models and their properties," *Cognitive Sci.*, vol. 6, pp. 205-254, 1982.
- [37] J. J. Hopfield and D. W. Tank, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.*, vol. 79, pp. 2554-2558, Apr. 1982.
- [38] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci.*, vol. 81, pp. 3088-3092, May 1984.
- [39] R. A. Hummel and S. W. Zucker, "On the foundations of relaxation labeling process," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 267-287, May 1983.
- [40] G. A. Carpenter, M. A. Cohen, S. Grossberg, T. Kohonen, E. Oja, G. Palm, J. J. Hopfield, and D. W. Tank, "Technical comments: Computing with neural networks," *Science*, vol. 235, Mar. 1987.
- [41] M. A. Arbib, "Brain theory and cooperative computation," *Human Neurobiol.*, vol. 4, pp. 201-218, 1985.
- [42] S. Amari, "Competitive and cooperative aspects in dynamics of neural excitation and self-organization," in *Competition and Cooperation in Neural Nets*, Amari and Arbib, Eds. New York: Springer-Verlag, 1982.
- [43] R. Mohan, G. Medioni, and R. Nevatia, "Stereo error detection, correction, and evaluation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 113-120, Feb. 1989.
- [44] Y. Ohta and T. Kanade, "Stereo by intra and inter-scanline searching using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, Mar. 1983.
- [45] G. Medioni and R. Nevatia, "Segment-based stereo matching," *Comput. Graphics Image Processing*, vol. 31, pp. 2-18, 1985.
- [46] S. D. Cochran, "Steps towards accurate stereo correspondence," in *Proc. DARPA Image Understanding Workshop*, Los Angeles, CA, Feb. 1987, pp. 777-791.
- [47] H. S. Lim and T. O. Binford, "Stereo correspondence: A hierarchical approach," in *Proc. DARPA Image Understanding Workshop*, Los Angeles, CA, Feb. 1987, pp. 234-241.
- [48] R. Mohan and R. Nevatia, "Segmentation and description of scenes based on perceptual organization," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, San Diego, CA, June 1989.



Rakesh Mohan received the B.Tech. degree from Indian Institute of Technology, Kanpur, India, and the M.S. and Ph.D. degrees in 1989 from the University of Southern California, Los Angeles, CA, all in computer science.

He is currently with the Exploratory Computer Vision Group at the IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests include computer vision, artificial intelligence, robotics, and artificial neural systems.



Ramakant Nevatia (S'64-M'65-SM'86) received the B.S. degree from the University of Bombay, Bombay, India, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, all in electrical engineering.

He has been with the University of Southern California, Los Angeles, since 1975, where he is currently Professor of Electrical Engineering and Computer Science and Director of the Institute for Robotics and Intelligent Systems. He spent the academic year 1981-1982 at Stanford University as

a Visiting Professor. He has authored two books: *Machine Perception and Computer Analysis of 3-D Curved Objects*. His research interests include computer vision, artificial intelligence, and robotics.

Dr. Nevatia is a member of the Association for Computing Machinery and the American Association for Artificial Intelligence. He is an Associate Editor of the journals *Pattern Recognition* and *Computer Vision, Graphics, and Image Processing*, and the Technical Editor in the areas of robot vision and inspection system for the *IEEE JOURNAL OF ROBOTICS AND AUTOMATION*.