The Information Fusion Problem: Forming Token Aggregations Across Multiple Representations

Edward Riseman, Allen Hanson,
and Robert Belknap

COINS Technical Report 87-48[1]

Abstract

A constraint-based approach to uniformly combining information from multiple representations and sources of sensory data is described. The approach is important to research in intermediate grouping, knowledge-based model matching, and information fusion. The techniques presented extend the capabilities of an earlier system that applied constraints to attributes of single types of extracted image events called tokens. Relational measures are defined between symbolic tokens so that sets of tokens across representations can be selected and grouped on the basis of constraint functions applied to these relational measures.

Since typical low-level representations involve hundreds or thousands of tokens in each representation, even binary relational measures can involve very large numbers of token pairs. Control strategies for ordering and filtering tokens, based upon constraints on token attributes and token relationships, can be formed to reduce the computation involved in producing token aggregations. The system is demonstrated using region and line data and an associated set of relational measures. The approach can be naturally extended to include tokens extracted from motion, stereo, and range data.

---

[1] This work has been supported by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. N00014-82-K-0464, Air Force Office of Scientific Research under contract F49620-83-C-0099, and the National Science Foundation under contract DCR-8500332.

Contents

1 Introduction

A major problem confronting vision systems which use multiple sensors, or which generate multiple low-level descriptions from image data, is the coherent and consistent integration of information contained in the multiple representations. Most vision systems utilize only one type of sensory data (e.g., intensity or color) and only one type of low-level process producing a single type of extracted image event that we call a symbolic *token* (e.g., regions of a region segmentation). However, after many years of computer vision research [HAN78a,b,HAN87,RIS87] it is clear that such systems are fundamentally limited by their restricted and unreliable view of the image data, and consequently their performance must suffer by the degree to which the image descriptions fail to support the system's goals.

More recently, the availability of multiple sensors has increased and multiple algorithms have become available for processing each type of sensor data. Consequently, the need is becoming more acute for computer vision systems to fuse information that is extracted by different types of low-level vision algorithms. Depth maps are being obtained directly from laser range data, and indirectly from motion and stereo algorithms that are applied to pairs and sequences of images, respectively. It has also become evident that each low-level process extracts only partial descriptions, and that there is a great deal of redundancy which can be profitably exploited across these descriptions. Thus, maximum reliability can only be achieved through processes that can integrate information represented in widely varying forms.

To be somewhat more specific, consider the interpretation of a road scene; the formation of a 'road' hypothesis should not be based on any single type of extracted image event (e.g. regions), but rather on an aggregation of multiple types of events (e.g. lines, regions, and surfaces) that have specific relationships to each other and which contribute to the support of the road structure. For example, one might like to find a homogeneous region of an expected intensity and color, bounded by two converging straight lines, and approximately covered by a horizontal planar surface.

Of course the reader should not be misled into an oversimplified view of the problem; there are extremely difficult low-level issues to be dealt with, such as instability of segmentation algorithms that leads to unpredictable fragmentation of lines, regions, and surfaces, and inconsistencies between the elements extracted in these representations. These are problems that are implicit in the nature of the problem of integrating unreliable information and will be true of all approaches, not just the one presented here. Our view is that to fully integrate multiple representations there will need to be complex grouping strategies that utilize the techniques presented here as part of a knowledge-directed interpretation process [BOL87,DRA87a,b,HAN87,REY87b].

Efforts to integrate multiple processes, or sources of low-level information, that produce tokens of different types may be loosely categorized into four broad classes. The most tightly coupled of these categories involves communication of intermediate data between the processes/representations so that the extraction of descriptive elements is directly integrated at the algorithm level [KOH83, MIL79]. The resulting description involves only a single type of low-level element, for example a region segmentation which incorporates edge and line information into the region formation process. A second approach is to allow each individual process to run to completion, resulting in multiple image descriptions and types of tokens, and then to manipulate the output of each algorithm into consistency, for example by splitting and merging, so that again a single description is formed [NAS83, KOH83]. A variation occurs when one process is allowed to run to completion; the results are then used to constrain a second process. The third approach involves using the separate descriptions to independently generate fairly high-level hypotheses such as objects [BAJ76, MCK84], or intermediate groups of image abstractions [REY84,REY87a], or 3D environmental depth [SHA86]; in these areas information fusion takes place at either the intermediate level of representation or at the semantic level after some degree of interpretation has occurred or some combination of both. The final approach makes use of high-level knowledge to guide the fusion process, resulting in an object-specific grouping of the low-level descriptions [BAL76, WEY86].

In this paper, we take the view that information fusion can be accomplished during later stages of the interpretation process, rather than at the earliest time that the image events are extracted (e.g. by attempting to directly integrate region and line algorithms). We also believe this will avoid some of the severe ambiguity problems of performing interpretation on the independent representations prior to information fusion. Our approach to fusion will be illustrated here by extending a constraint-based object hypothesis system [RIS87] to operate over multiple token types, in this case regions and lines extracted from the image data. When surface elements extracted from depth maps produced by motion and stereo algorithms are available, they can be aggregated with regions and lines. The techniques could also be easily extended to include fusion of information from textured areas, corners, volumes, and generally any other token abstracted from the sensory data.

2 Background

2.1 The Intermediate Symbolic Representation (ISR)

It is generally accepted that a computer vision system must perform a variety of transformations of the data during the interpretation process. One of the key abstractions is the transformation of pixels, or more generally arrays of sensory data, into image events which can be named and accessed by their properties. We refer to the symbolic representation of an extracted image event as a *token*; attributes are associated with tokens, and tokens participate in relationships with other tokens. It should also be noted that we will allow new tokens to be formed by a grouping of a set of tokens. The resultant representation is called the "intermediate symbolic representation" (ISR) [SOU86]. This representation serves as the communication interface between the low-level descriptive processes and the high-level interpretive processes in the VISIONS system [HAN87]

At a coarse conceptual level, the VISIONS system is organized into three levels of processing: low, intermediate, and high. Currently, the low-level, or segmentation, processes output a symbolic representation of the data in the form of regions and lines. Attributes, such as color, texture, location, size, shape, and orientation, are then calculated for each region or line. All of this information is then organized into the ISR which allows flexible access to the attributes of a token as well as the relations between the tokens. Interpretation processes use knowledge of the objects in the domain to control a set of intermediate-level processes for generation of initial object hypotheses and reorganizing the low-level output. In general, the only requirement for placing a new type of low-level token into the intermediate representation is that each primitive element of that data type must have a symbolic name (e.g., region-240, surface-38, corner-46) and a non-empty set of attribute-value pairs. It is the values of the token attributes, and as we shall present here the relational information between tokens, that provide the basis for initial interpretation processes.

In the next two subsections we will very briefly outline the low-level segmentation algorithms, and the current version of the constraint-based system for generating hypotheses from single types of intermediate elements. Succeeding sections will present the extensions of this system to operate over multiple types of intermediate level tokens.

2.2 Segmentation Algorithms

The construction of the region and edge representation makes use of two low-level algorithms: a local histogram-based region segmentation algorithm [BEV87, NAG82, KOH83] and a straight line extraction algorithm [BUR86]. The region segmentation algorithm involves the detection of clusters in a one-dimensional feature histogram, associating labels with the clusters, mapping the labels onto the image pixels, and then forming regions of connected pixels with the same label. In general, the process of global histogram labeling causes many errors to occur; hence an additional key aspect of this algorithm is the localization of the histogramming and peak selection processes to subimages, with some coordination of peak selection between adjacent subimages. Regions are formed from the local histogram labels via a connected components algorithm, and then the subimage regions are merged across the artificial subimage boundaries by means of a set of merging rules. These rules are applied to adjacent regions and take into account local feature similarity along the common boundary, as well as global feature similarity across the regions, and other global properties such as relative size of the regions, length of their common boundary, etc. An example region segmentation of the image in Figure 1a is shown in Figure 1b.

The line algorithm is directed toward the extraction of linear features in intensity images. It provides a representation of intensity variations by segmenting the intensity array into connected subsets of pixels which have similar gradient orientation. These pixels then serve as the *line-support-set*[2] for the extraction of a line. A straight line is fit to the pixel line-support-set and attributes of the associated line are then extracted. Thus, both regions and lines have a common pixel-based representation, which allows simple intersection operations to provide the basis of the relational measures between line and region tokens that will be defined later in the paper. An example set of extracted lines is shown in Figure 1c.

2.3 Other Related Research

There have been a few attempts to integrate results from multiple low-level processes operating on one or more sensory sources [HAN78b, KOH83, NAS83]. In the past, efforts to combine multiple processes operating on visual data have typically involved the integration of line and region data, which are the two most common types of low-level algorithms employed. More recently, there has been an increasing number of efforts to combine range and vision data [ARK87,BES85,SHA86]. Shafer and Thorpe [SHA86] have developed a blackboard system for the CMU NAVLAB mobile vehicle. Here range data and visual data are independently processed and combined during the interpretation process. On the other hand, Agarwal [NAN87] combined the processing of infrared and visible light images with computational models of the image generation process to improve the processing beyond either alone.

Kohler [KOH83] proposed solutions to the information fusion problem at the segmentation level. The first method allowed each process, a line algorithm and a region algorithm, to segment the image independently. Tokens resulting from each process were then combined in one of two ways: 1) all line boundaries were projected onto the region representation producing a merged (and possibly more fragmented region representation), or 2) only those boundaries with support from all segmentations are accepted, and the resulting boundary segments are connected. The second method allows each segmentation process to influence the other segmentation processes via communication of intermediate results during their execution. Unfortunately, Kohler did not

---

[2] The term *line-support-region* from [BUR86] has been renamed as *line-support-set* of pixels, with the understanding that they are a contiguous set of pixels, in order to avoid confusion between *regions* and *line-support-regions*.

demonstrate that the combined results provide significant improvement beyond the best individual segmentation process operating alone.

Nasif and Levine [NAS83] also address the problem of integrating multiple sources of information at the segmentation level. Their system uses a rule-based expert system to segment natural scenes based on a region and line representation. Production rules are used to split and merge regions and lines based on similarity and differences of spectral attributes and binary relations between regions and lines. An additional set of rules is used to group regions and lines into focus-of-attention areas which are used to guide the application of segmentation processes. Our position is that the number of rule applications that need to be executed in low-level vision make the use of production systems unacceptably inefficient, although the constraints we define can be cast in this framework if one so desires.

Bajcsy and Tavakoli [BAJ76] developed a system for recognizing roads in aerial images. In their system a world model of roads is used to extract road segments as elongated regions. These segments are then grouped based on orientation and proximity to form road hypotheses.

Mckeown, Harvey, and McDermott [MCK84,85,86] perform map-guided interpretation of aerial images by applying prototype rules to regions. These rules build initial class or sub-class hypotheses for each region. Regions of a common class or sub-class are then grouped and the prototype rules are reapplied to the groups. This process continues until acceptable functional areas are defined for the image.

Reynolds, Irwin, Hanson, and Riseman [REY84] use region and line information to guide processing in a hierarchical interpretation of very large aerial images. Lines which bound regions in a coarse-level segmentation are used to locate regions with rectangular outlines. The parameters of long lines with high contrast are mapped through a Hough space to find a peak of common orientation. The lines which correspond to this peak and regions with rectangular outlines are used to select interesting areas of the image, which are then subjected to finer levels of segmentation and interpretation. This work was performed in our research environment and initiated the more general framework that is presented here.

3 Constraints on Tokens of a Single Type

3.1 Constraint Functions on Token Attributes

A simple type of knowledge source for generating hypotheses of object class labels for particular regions has been under development in the VISIONS environment for some time [WIL81, WEY83, RIS84, BEL86, HAN87a, REY87a]. The mechanism for generating object hypotheses involves the definition and application of constraint functions to the attributes of tokens; in previous work each constraint function on a single attribute was referred to as a simple "rule". In turn, compound constraints were defined as combinations of the output of a set of simple constraint functions. However, we are avoiding the term "rule" here so that there is no confusion with the IF-THEN rules of production systems. A constraint function can be viewed as a mapping of a feature value into a weighted "vote" or "score" for an object on the basis of that feature. The combined response of these constraints serve as a focus of attention mechanism for other, more complex knowledge-based processes [DRA87a,b,HAN87a,WEY86]. The region features include color, texture, shape, size, image location, and relative location to other objects. More recently, the approach has been extended to lines, with attributes including length, orientation, contrast, width, etc. In many cases, it is possible to define constraints which provide evidence, in the Dempster-Shafer sense, for and against the semantically relevant concepts representing the domain knowledge [REY85,REY87a].

While no single constraint on the features of a single token is totally reliable, the combined evidence from many such constraints often imply the correct interpretation of a token, or at least result in the correct hypothesis being among the top candidates. Rather than viewing the application of the constraint set as a classification process in the pattern recognition sense, the output of a constraint set applied to regions or lines can be used as a rank-ordered set of unreliable hypotheses in an AI focus-of-attention process [HAN87b]. Consequently, all hypotheses must be verified at some point in the interpretation process.

Let us now more formally specify a constraint function which hereafter will often be referred to simply as a constaint. A simple constraint function is a function F applied to the k^{th} attribute (or feature) of the j^{th} token of type T. Thus, $F(T_{jk})$ would specify the application of the constraint function F to the token value T_{jk}. For example, if the k^{th} attribute of line tokens (type L) is length, then L_{jk} will be the length of the j^{th} line token, and $F(L_{jk})$ would be the response of the constraint function when applied to the line length of token j. A variety of forms for the function F have been employed, with no appreciable difference in the results. The first was an extended real-valued piecewise-linear function F

$$F(T_{jk}) \in \{[0,1] \cup VETO\} \tag{1}$$

specified by six points in the feature range $\{\theta_i, i = 1, \ldots 6\}$ [WEY83, WEY86, RIS87]. Here F was defined as shown in Figure 2 as a central range $[\theta_3, \theta_4]$ of maximum response of 1, surrounded by linearly ramped functions from 0 to 1 defined by $[\theta_2, \theta_3]$ and $[\theta_4, \theta_5]$, with surrounding zero intervals and $F(T_{jk}) =$ VETO for $T_{jk} \leq \theta_1$ and $T_{jk} \geq \theta_6$. Note that any subset of the intervals $[\theta_i, \theta_{1+1}]$ for $i = 1, \ldots, 5$ can be null. The simplicity of this approach is that F in this form could be compactly stored as a 6-tuple, or sets of 6-tuples.

Compound constraints are a hierarchical collection of simple constraint functions and other compound constraints with an arithmetic or logical combination function for collecting the individual responses into a single response. Thus, the combination function C is also an extended real-valued function on a vector of constraint functions:

$$C[F_1(T_{j_1k_1}), \ldots F_M(T_{j_mk_m})] \in \{[0,1] \cup VETO\} \tag{2}$$

In addition to the highly structured piecewise-linear form described above, alternative variations have been used with additive and multiplicative combination functions. For additive combination, an arbitrary piecewise linear functions with either $\{[0,1] \cup VETO\}$ or a $[-1,1]$ range have been employed. The function has also been generalized to (a discrete sampling of) a continuous function of the class-conditional Bayesian likelihood. For multiplicative combination, the range $[0, \infty]$ has been used, with a response of 1 considered neutral (e.g. no information) and a response of 0 corresponding to a veto. Finally, we have even employed functions defined via a linguistic interface with terms for ranges of Very-Low, Low, Medium, High, and Very-High of a scalar-valued attribute. For discussions of these variations, see [RIS84,RIS87,HAN87a,KIT86,WEY86].

In some experiments, the top-level compound constraint for an object was structured as a combination of five other compound constraints (as shown in Figure 3) to represent color, texture, size, shape, and location constraints, each of which was composed of a set of simple constraints [RIS87]. The responses of these separate component constraints were combined using a weighted average function, although any mathematical combination of the scores can be defined. Note that the weights in the combination function were only expected to provide a coarse range of importance, using 3-5 integer values to approximate values of "low", "medium" and "high".

3.2 Relational Similarity Constraints on Tokens of the Same Type

When dealing with unreliable segmentation processes, forming aggregations of tokens is usually necessary since the set of tokens need to be grouped and reorganized in order to match an object model [DRA87b,HAN87a,HAN87b,REY87b]. The basis of this grouping usually involves not only the attributes of tokens, but also the relations between tokens. The constraints described in the previous section are unary, since they accept a single token attribute as input and return a value that can be viewed as a confidence or rating for the hypothesized object. The highest ranked of these hypotheses can serve as a partial (and probably errorful) interpretation of the original image. However, it is clear that constraints on relationships between tokens are also fundamental to object recognition. They can be handled in much the same way by defining binary relational constraints on pairs of tokens. In this case, the response of a constraint function specifies the degree to which the constraint on a relation is satisfied. Binary constraints can begin to capture some of the contextual expectations that the developing interpretation is expected to satisfy. For example, the similarity of a given region's intensity can be compared with the intensities of all other regions via a simple difference measure.

Let us consider as an example a line token set with an implicit relational measure of simple absolute difference applied to the attribute of orientation. Given a specific line token (e.g. L-435), then all other tokens can be rank-ordered via a constraint function applied to the feature difference with the designated token. Figure 4 shows two different constraints F_1 and F_2 for processing line tokens relative to a given line token. F_1 gives a maximum response of 1 for all tokens that are within 10° of L-435, a linear decreasing response from 10° to 30° and a veto response beyond 45°. The effect of applying F_1 is to rank equally all lines that are very similar in orientation to that of L-435; beyond 10° they are ordered based on their relative orientation. Lines whose relative orientation is greater than 45° away from that of L-435 are excluded. The constraint embodied in F_2 is the selection of all line tokens that are within 5° of being orthogonal to L-435. In the case of constraint F_1, the relational measure for those lines not vetoed is mapped into a response which can be used to coarsely rank order the line pairs, while F_2 is used to filter the subset of lines. In effect constraint F_2 defines a relation on approximately orthogonal pairs of lines that have L-435 as a member; the relation is defined to have a value of True for all pairs where $F_2(L_{435,k}; L_{j,k}) = 1$. Either of the subsets resulting from application of these constraints could be followed by a token attribute constraint for ordering or filtering the remaining lines on other attributes such as location, contrast, length, etc. in absolute terms or relative to L-435.

The constraints on relational measures that we have been describing operate on tokens of the same type (e.g., either regions or lines), and result in a grouping of tokens into hierarchical and more abstract descriptions. Prior to the work reported here, constraints on region token pairs [RIS87] were used to form aggregations of regions with similar properties. Concurrently with the work reported here, constraints on relational measures have been used to define line relations and applied to form colinear aggregations [BOL87] and rectilinear geometric aggregations of lines [REY87b]. One should note in that this latter effort, the grouping could be based on a set of token attributes or relations or both.

In general, regions can be grouped on similarity of multiple attributes, such as color, texture, and adjacency. They can also be grouped on the degree to which subsets of tokens in a token set satisfy some constraint, such as shape on an n-ary relational measure (e.g. rectangularity) in order to define an n-ary relation. Relations that could be used to group lines include similarity of orientation, proximity of endpoints, overlap, parallelness, etc. [BOL87,WEI86]. Pairs or sets of parallel lines could be grouped, as could pairs or sets of lines whose endpoints fall in the same local neighborhood. Each of these groups could then be stored as an entity with its own set of attributes. Of course the benefit of these processes is that matching larger aggregate line structures to a model should be much more reliable than matching based on the attributes of a single line [REY87b,WEY86].

4 Integrating Representations via Construction of Token Aggregates

The fundamental problem that is being addressed in this paper is the integration of multiple low-level representations into the interpretation process. While the approach presented here offers only one type of information fusion mechanism and deals with only some of the most general levels of the information fusion problem, there are several important advantages. First, it offers an entirely modular and natural method for incorporating additional processes and representations as a vision system undergoes incremental development; in particular, existing low-level representations do not have to be modified in any way. Secondly, the integration is accomplished at the intermediate grouping and/or interpretation levels through constraints which relate entities in the independent representations. Since there is no direct interaction of the processes, the mistakes of one low-level process will not affect the output of the other low-level processes. If a sufficient body of consistent information exists in several representations, then low-level mistakes in a given representation may be detected and either ignored or corrected, as opposed to integrating partially erroneous data in some form, such as a least-squares optimization process. Third, this approach is an extension of an approach applied to token attributes that has already proven to be somewhat effective on very complex natural scenes [RIS84,HAN86,HAN87a]. Finally, the techniques can be used as part of general intermediate grouping processes. The grouping can be viewed to be knowledge-directed (e.g. via a model of an object), or it could be viewed as a data-directed token aggregation process whose goal is to extract interesting structures of a priori importance [REY87b].

4.1 The Formation of Aggregations of Tokens

One step in the abstraction process towards semantic identification is the formation of aggregations of tokens of different types into new tokens. Sets of constraint functions on token attributes have already been introduced. Certainly a set of tokens that exhibit some common attribute could be a useful grouping, e.g. a set of lines of horizontal orientation. Thus, aggregations can be constructed as a subset of tokens that satisfy a set of unary constraints on attributes of a single token type.

Relational constraints are defined as a real-valued function on a relational measure defined over a token set. A *relational measure* $M(T_{j_1k_1} \ldots T_{j_mk_m})$ is a function of the attributes of multiple tokens, possibly of different types. As we have already discussed, for tokens of the same type there is an implicit binary relational measure in that the same scalar-valued attribute of any two tokens can be compared by their similarity or difference. For example, given a specific region, similarity relational measures be used to compute the distance between feature centroids as well as the difference in mean intensities between the given region and all other regions. Once such a scalar relational measure exists, a constraint function can be applied to the relational measure to produce a response that represents the degree to which the relational constraint is satisfied.

In order to compare tokens of different types a relational measure must be defined between each pair of token types. In this paper, only binary measures between region and line tokens are developed. The constraint functions on relational measures can then be applied to sets of tokens across the multiple representations in order to group tokens into aggregations. For example, relational measures between line tokens and region tokens can be defined, such as the degree of intersection between tokens; a constraint on this relational measure could then select, for each region, all lines that are sufficiently interior to the region. The reader should note that when a relational constraint is used to filter the token tuples, the resulting tuples define a relation on the token sets.

4.2 Region-Line Relational Measures

In this paper we present a specific set of relational measures to provide a computational method of relating regions and lines. Relational constraints on these relational measures will then be used to implement the following relations between regions and lines:

- BOUNDING lines – those lines associated with a region boundary;
- INTERIOR lines – those lines interior to a region; and
- OTHER lines – those lines which intersect a region, but are neither bounding nor interior to the region.

The relational measures chosen are based on the intersection of sets of pixels. Only lines which intersect a given region are of interest since the line-support-sets of pixels that we employ can be expected to overlap the regions that they bound or are interior to. Thus, we naturally have INTERSECTION (defined in the usual way on pixel subsets) as a relation which can be used as a filtering constraint in the sense of selecting a subset of tokens, as opposed to ordering the tokens via the constraint response. In the following discussion, however, three other relational measures will be used: "interior-line-percentage", "region-perimeter-percentage", and "line-boundary-percentage", as shown in Figure 5. Some alternative approaches to line-region relational measures are briefly discussed in the next section to deal with situations where lines have support sets that do not intersect a region, but rather are in close proximity.

The first relational measure, "interior-line-percentage", is the ratio of line area interior to the region to total line area of the line-support-set. The interior line area is simply the number of pixels in the intersection of the region and the line-support-set; an example is shown in Figure 6. The interior-line-percentage measure discriminates lines that are entirely INTERIOR from BOUNDING lines, whose line-support set will lie partially outside the region (refer to Table I). An INTERIOR line will have a value of 100% for this relational measure, indicating that the line-support-set is completely contained by the region. An ideal BOUNDING line with a symmetric line-support-set of pixels lying exactly on the region boundary would have half its pixels in the region and a value of 50% for its interior-line-percentage. However, this measure will not discriminate BOUNDING lines from "OTHER" lines that have a portion of the line inside the region, but little on the boundary. As shown in Table I both BOUNDING lines and OTHER lines which intersect the region will have some value between 0% and 100% for interior-line-percentage; this indicates that the line-support-set crosses the region boundary and has at least one pixel in common with of the region. All lines that do not intersect the region will have a value of zero for this measure.

The other two relational measures can be used to discriminate BOUNDING lines from INTERIOR lines. The natural duality between regions and their boundary lines can be exploited in a straightforward manner to indicate how much of a region boundary or a line is covered by the other. "Region-perimeter-percentage" measures the fraction of a region boundary made up of one line and is defined to be the ratio of the intersection of the region perimeter pixels and the line-support-set to the length of the region perimeter (see Figure 7). "Line-boundary-percentage" measures the fraction of a line contributing to the region boundary and is defined to be the ratio of the intersection of the line-support-set and the region perimeter to the total line length in pixels (see Figure 8). Ideally, a line which lies approximately on a region boundary will have a high value of line-length-percentage since the region boundary will cover much of the line. The same is true of region-perimeter-percentage although a single line will be expected to cover a smaller portion of the entire region boundary. As shown in Table I these two measures distinguish BOUNDING lines from INTERIOR and OTHER lines. Figure 9 shows the lines which intersect a region and the values for the three relational measures for some of the lines.

4.3 Alternative Region-Line Relational Measures

The measures described in the last section may be unreliable for several reasons. If a region boundary is irregular where the line intersects it, the region boundary length counted in terms of pixels could be elevated and the measures will be inaccurate. In addition the measure will not distinguish very well between lines which are parallel to the region boundary and lines which pass through the boundary at a shallow angle. Also, lines might lie along and near the region boundary but not intersect it, and thereby fail to produce useful intersection relational information in these cases.

In this section we briefly present a few alternative methods of computing region-line relational measures. These alternatives address some of the problems listed in the previous sections, including inaccurate placement of line-support-sets relative to region boundaries, the lack of line-to-region boundary angle measures, and intersection as the only basis of the relational measures.

Line-support-sets as we have extracted them are based solely on collections of pixels with common gradient direction. The algorithm to fit a line to these regions is not affected by small amounts of noise in the line-support-set and usually computes an appropriate line [BUR86]. However, the line-region relational measures presented are based on the line-support-set, rather than the extracted line, and may produce inaccurate results. A solution to this problem is to use something other than the line-support-set to determine relations and compute associated features. The following alternatives also can be used with a line produced by any algorithm, i.e. those that do not produce a line-support-set of pixels directly.

One approach is to use only those pixels through which the extracted line passes (as opposed to the entire set of pixels in the line-support-set) to compute the relational measures presented above. This removes the possibly large set of low-magnitude gradient pixels in the extracted line support set from participating in the relational measure. This would increase the accuracy of interior/perimeter discrimination, but would lead to problems when a line lies parallel and nearby to the region boundary, but offset, and therefore does not overlap it. Since different low-level algorithms often produce tokens with some spatial inconsistency, the region-perimeter-percentage and line-length-percentage features could be rendered useless.

A variation of this approach is to define an artificial line-support-set which uniformly surrounds a line. The width of the artificial line-support-set could be varied according to the sharpness of the edge, (i.e. the gradient magnitude and width). This might retain the good properties of line-support-sets while eliminating the problems of ambiguous and inaccurate placements of lines. This method has been implemented and appears to produce improved line-region relation accuracy.

A third alternative is to use the idea of chamfering [BAR78] to form distances between region boundaries and lines. This algorithm can be naturally thought of as implemented on a cellular array machine. A wave of spreading activation on the array can be implemented to measure distance by starting all cells on the line (or region boundary) with a value of 0 and propagating the field to adjacent 8-connected neighbors while incrementing their count by 1. In t steps of propagation, cells that are a distance t away will be reached with a count of t. The cells on the receiving region boundary (or line) will have their distance determined by the earliest marker (i.e., lowest value) or by some function of the values resident in the cells corresponding to the boundary or line. Thus, distance is obtained naturally and various other geometric relations, such as intersection, parallelness, and so forth can also be measured. This approach captures many of the measures we have just presented, but does not demand the actual intersection of the two types of elements in order to extract useful information. However, without parallel hardware its computational cost is significantly larger and could be prohibitive.

4.4 Relational Constraints

Relational constraints are used as the final step in the formation of aggregations from multiple representations. The relational measures presented in the preceding sections provide the basis for defining these relational constraints.

A relational constraint function for lines and regions can be specified for each relational scalar measure that has been defined (in the same manner that token attribute constraints are defined). Thus, a simple constraint can be specified for each of interior-line-percentage, line-length-percentage, and region-perimeter-percentage measures; note that any of these simple constraints may be omitted. A combination function can be defined for combining the output of the set of simple constraints into a compound relational constraint.

The form of the function that combines the simple constraints is not critical, and in this paper we will use the same simple piecewise-linear function with a range of $\{[0,1] \cup VETO\}$ described earlier. The VETO range(s) serves only as a first filter for selecting or removing candidates for processing, in the sense that the vetoed tokens do not satisfy the constraint. This does in fact, define a relation over the token sets, but the remaining non-vetoed tokens in the relation still have the graded response from the constraint function, which can be used for ranking or further filtering of token pairs to produce a more restricted subset of tokens.

4.5 Controlling the Formation of Token Aggregations

The aggregation of tokens via relational constraints must, of course, contend with the combinatorics of the large number of image tokens whose relationships must be examined. The concept of *focus-of-attention* becomes important when one considers that the representations being used typically involve 2,000 to 10,000 lines and 200 to 1,000 regions. Thus, there are potentially 400,000 to 10 million line-region pairs which could be related, and these numbers become much larger as additional representations, multiple sensory sources, finer image resolutions, or larger images are considered.

The order and manner in which attribute constraints and relational constraints are applied are the basis of the *control strategy* for the construction of token aggregations. Tokens of different types can be aggregated in many ways and one seeks to avoid the combinatorics of computing relations between large token sets. There are two obvious ways in which the constraint functions can be used in control. A constraint on the cross product of the token sets can be used either to ***rank order*** the set of token tuples via the response of the constraint function, or to *filter* (i.e. select) a subset of the token tuples for further processing. Of course the responses of the constraint functions that are used for ordering could also be used for filtering by specifying filtering criteria such as thresholds.

The techniques of rank-ordering and filtering will prove very useful as focus-of-attention mechanisms for efficiently controlling the overall grouping process, since it is usually necessary to have one or more filtering stages in order to reduce the combinatorics of the process. The relative order in which the token attributes and token relations are used to select subsets will also affect the structure of the token aggregation and the computation necessary to achieve it. Tokens of each type could be separately processed, or alternatively a particular token type might be used as the primary grouping element to which tokens of other types could be aggregated. Constraints on token attributes can be applied before or after relational constraints are used to begin to aggregate tokens of the same or different types.

Figure 10a depicts one way of attaining aggregations of region and line tokens by using constraints on both token attributes and token relations. In this case a set of region attribute constraints and line attribute constraints are applied independently. The resulting token sets can be rank-ordered based upon the response of the constraint function, and here they are further reduced by filtering each ordered set into smaller candidate sets R' and L' on the basis of the constraint function response. At this point relational measures are extracted, for each region-line pair in the Cartesian product $R'X\ L'$ and then relational constraints are applied. Here again, one can filter the rank-ordering to produce the token aggregations that sufficiently satisfy the relational constraint function. The important point to note in the example sequence is that the lines and regions are each filtered independently, and the structuring of the aggregate would take place <u>after</u> the token sets of each type were reduced in number. Thus, the relational measures are only applied to the much smaller number of tokens in $R'X\ L'$.

Let us contrast the control strategy above with an alternative mechanism for constructing region-line aggregates by using one of the token types as the "primary" grouping element, and then integrating information from the other token type in a secondary manner. Figure 10b depicts a modification of the previous control strategy whereby a subset of region tokens R' is first extracted via region attribute constraints. Then the INTERSECTION relation is used to select the subset of lines $L' \subseteq L$ which intersect R' (note that the INTERSECTION constraint is a binary-valued constraint). Then line attribute constraints are applied to this reduced subset of lines to further reduce the line subset , after which the relational constraints are applied in the manner previously described. This technique would be much more efficient than the previous strategy when the size of R$'$ is small, so that fewer lines L$'$ are selected for which computation must be expended to produce L$''$, and possibly fewer token pairs in S$'$ are produced for which relational measures must be computed. In our image domains and with our line algorithm, the large number of lines can be greatly reduced, making this second technique desirable in many cases.

5 Results

This section presents the results of forming aggregations via token attribute and token relational constraints applied to suburban house scenes and road scenes. The examples chosen for this paper involve aggregations of tokens that serve as texture measures and aggregations of tokens with specific shape properties.

A simple texture measure of line density can be computed by counting the number of lines within a region and normalizing by the region size. This is accomplished by forming aggregations of regions and their interior lines. A filtering constraint uses the interior-line-percentage relational measure to select only those lines which are completely (or mostly) interior to the region. A filter is defined to group into an aggregation those lines associated with each region which sufficiently satisfy both relational and attribute constraints. The density of interior lines in each aggregation is then computed as an attribute of this new token and mapped to a score for the region (which can also be thought of as a score for the region-line aggregation).

Figure 11a shows an example of extracting interior lines for regions in a house scene, and then computing the interior line density of these regions as a texture measure (see Figure 11b). Some objects, notably the roof of a house, are characterized by short horizontal lines (due to the shingles) interior to the region. By adding the additional attribute constraint of horizontal orientation on the interior lines, the previous result can be extended to focus attention on the house roof as shown in Figure 11 c,d. Additional constraints on line length and line contrast can be defined to extract only short, horizontal interior lines to the degree that these characteristics of the expected texture element are known.

The roof region could be obtained or verified in another way. The line-boundary-percentage relational measure could be used to select lines which lie to a great extent on the boundary of the region (see Figure 12a). A line attribute rule could then be defined to favor long lines (see Figure 12b). The lines which received high scores from both the line boundary rule and the line length rule (i.e., long boundary lines) could then be grouped to form a region-boundary aggregation. At that point parallel relations, rectangle or parallelogram structures could be identified.

Another simple shape measure can be computed by determining if a region is bounded by a pair of long vertical lines. As Figure 13 shows, the process is useful for extracting telephone poles in road scenes. The filtering relational constraint for this measure uses line-boundary-percentage to select only those lines which lie on a region boundary; an attribute constraint then selects the long vertical lines. Relational measures can be defined to form aggregations of pairs of parallel overlapping lines from the long vertical boundary lines.

A variety of more complex 2D shapes can be matched to lines by extracting lines that bound regions. The techniques presented here are only a part of more complex grouping and model matching procedures that are being developed in other research [HAN87a,REY87].

Figure 14 shows a set of rectangles extracted from a house scene. In this case, the set of lines intersecting a region were filtered to extract the set of bounding lines. These were further filtered on the basis of co-parallel, collinear, and endpoint coincidence relations. Pairs of 'adjacent' lines were then filtered on the basis of constraints on their relative orientation in order to form corner hypotheses and the resulting set of lines were matched to a rectangle model. As the figure indicates partial matches to the rectangle model are allowed. The matches could be further restricted if one is seeking dark shutters by using a constraint on the intensity of regions, although this was not done in this example.

There are sometimes serious problems with constructing aggregations through relational measure directly computed from initial token representations. If the desired primary token is fragmented, whether it be a region or a line, then the expected relational responses might be distorted significantly because some of the expected token attributes, token relations, and features of the extracted aggregation may be significantly changed. One must balance the unreliability of extracting useful primary tokens by the computational savings achieved by focussing upon the subset of secondary tokens that satisfies some relational constraints with respect to the filtered primary set. To the degree that these problems occur, many stages of hierarchical token aggregation may be necessary with more complex strategies for applying relational measures.

Let us consider a specific real example, shown in Figure 15, where both representations (e.g. regions and lines) would have some difficulty in directly providing the basis for aggregations of tokens from the other representations. The roof region in Figure 1 is fragmented into many smaller regions (Figure 15a). In this case no region will get full benefit of the bounding roof lines, since each region is only bordered by a subset of the roof lines; Figure 15b depicts the lines intersecting the largest roof region. An alternate grouping strategy would be to extract a set of roof lines first (which may be a difficult task in itself), then the set of regions that are bounded by these lines could be used in some manner to aggregate regions. The line information if properly filtered shows the outline of the roof fairly clearly (Figure 15c), but the initial set of lines in that spatial vicinity provide many possible aggregations. The process of grouping lines into meaningful geometric structures is a non-trivial problem and is a focus of continuing work on grouping and knowledge-directed processing in our research group [BOL87,HAN87a,HAN87b,REY87,WEI86,WEY86].

6 Future Work

An obvious and important extension to the existing system is to add additional representations obtained from other low-level processes. The types of information that could be added include surface segmentations, and 2D and 3D motion and depth token attributes. Each segmentation or low-level process would create a set of tokens with associated attributes which could be added to the intermediate-level representation. These new tokens could then be used in the same way as regions and lines are used now. Each new token type would require the definition of relational measures between tokens of different types. There would be no other major modifications to the system.

There were a number of extensions outlined in the paper for improved relational measures based on the formation of artificial line-support-sets of pixels, and the use of chamfering between lines and regions. These modifications are aimed at increasing the accuracy and utility of relational measures and rules. The necessity of developing such techniques will depend upon empirical investigation in various application domains.

The ideas presented here can be used to build hierarchical aggregations; for example, aggregates of lines could be formed by grouping colinear sets of lines into new longer line tokens [e.g.,BOL87,REY87b,WEI86]. By treating each aggregation as a new token, attributes could then be computed for each and the constraints applied recursively.

Finally, each object that must be recognized could be defined by a separate model and control strategy for aggregating the different token types. In addition, there is no simple solution to the problem of dealing with fragmented token sets or whether to use a particular token type as the primary basis of organizing token aggregations. The effectiveness of different aggregation methods is undoubtedly data-dependent and varies with the task domain and may also be object-dependent (e.g. short lines in tree texture, versus long lines on road boundaries). Thus, multiple alternate grouping strategies will probably be required in order to extract and utilize information across multiple representations in a generally robust and efficient manner. In the VISIONS system [DRA87a,b,HAN87a] the knowledge-based schema system provides flexible mechanisms for defining and applying control strategies. The mechanisms described in this paper are only meant to serve as the first stage of this organizing process.

7 Conclusion

The use of constraints on relational measures between tokens of the same and different types is a uniform, straightforward way of combining information from multiple low-level processes. The techniques developed in this paper allow information fusion to take place during the interpretation process as intermediate tokens are aggregated via object-dependent constraints on token attributes and token relations.

The approach overlaps issues and techniques in the areas of grouping and model matching. To the degree that tokens in one of the several representations do not exhibit the characteristics that provide the basis for directly extracting the desired structures, more complex perceptual organizing processes and knowledge-based strategies will be required. While the concepts of relational measures and relational constraints can still be the basis of these strategies, many stages of hierarchical aggregations may be required. In such cases efficient control strategies will become a major issue.

References

[ARK87] R. Arkin, "Towards Cosmopolitan Robots: Intelligent Navigation in extended Man-Made Environments", Ph.D. dissertation, Computer and Information Science Department, University of Massachusetts at Amherst, expected in August 1987.

[BAJ76] R. Bajcsy and M. Tavakoli, "Computer Recognition of Roads from Satellite Pictures," *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-6** (September 1976), pp. 623–637.

[BAL76] D. H. Ballard, *Hierarchical Detection of Tumors in Chest Radiographs*, Basel: Birkhäuser-Verlag (ISR-16), January 1976.

[BAR78] H. G. Barrow, J. M. Tennenbaum, R. C. Bolles, and H. C. Wolf, "Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching," *Proc. Fifth International Conference on Artificial Intelligence* (August 1977), pp. 659–663.

[BEL86] R. L. Belknap, E. M. Riseman, and A. R. Hanson, "The Information Fusion Problem and Rule-Based Hypotheses Applied to Complex Aggregates of Image Events," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Fl., June 1986.

[BES85] P.J. Besl, and R.C. Jain, "Range Image Understanding", in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 9-13, 1985, New York, pp. 430-451.

[BEV87] R. Beveridge, A. Hanson, and E. Riseman, "Segmenting Images Using Localized Histograms and Region Merging", COINS Technical Report, Computer & Information Science Department, University of Massachusetts at Amherst, in preparation.

[BOL87] M. Boldt and R. Weiss, "Token-Based Image Abstraction", COINS Technical Report, Computer & Information Science Department, University of Massachusetts at Amherst, July 1987.

[BUR86] J. B. Burns, A. R. Hanson, and E. M. Riseman, "Extracting Straight Lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 425–455, July 1986.

[DRA87a] B.A. Draper, R.T. Collins, J. Brolio, J. Griffith, A. Hanson, and E. Riseman, "Tools and Experiments in the Knowledge-Directed Interpretation of Road Scenes", *Proc. of the DARPA Image Understanding Workshop*, Los Angeles, CA, February 1987, pp. 178-193. Also COINS Technical Report 87-05, Computer & Information Science Department, University of Massachusetts at Amherst, January 1987.

[DRA87b] B.A. Draper, J. Brolio, R. Collins, A. Hanson, and E. Riseman, "The Schema System", COINS Technical Report, Computer & Information Science Department, University of Massachusetts at Amherst, in preparation.

[HAN78a] A. R. Hanson and E. M. Riseman, "VISIONS: A Computer System for Interpreting Scenes," *Computer Vision Systems* (A. Hanson and E. Riseman, eds.) (1978), pp. 303–333, Academic Press.

[HAN78b] A. R. Hanson and E. M. Riseman (Eds.), *Computer Vision Systems*, New York, Academic Press, 1978.

[HAN87a] A. R. Hanson and E. M. Riseman, "The VISIONS Image Understanding System - 1986", in *Advances in Computer Vision*, (Chris Brown, Ed.), to be published by Erlbaum Press. Also COINS Technical Report 86-62, Computer & Information Science Department, University of Massachusetts at Amherst, December 1986.

[HAN87b] A.R. Hanson and E.M. Riseman, "From Image Measurements to Object Hypotheses", COINS Technical Report, University of Massachusetts at Amherst, in preparation.

[KIT86] L. Kitchen, R. Weiss and J. Tuttle, "Identification of Human Faces Using Data-Driven Segmentation, Rule-Based hypothesis Formation, and Interative Model-Based Hypothesis Verification", COINS Technical Report 86-53, Computer & Information Science Department, University of Massachusetts at Amherst, October 1986.

[KOH83] R. R. Kohler. "Integrating Non-Semantic Knowledge into Image Segmentation Processes," Ph.D. Thesis, University of Massachusetts at Amherst, September 1983; also COINS Technical Report 84-04.

[MCK84] D. M. McKeown, W. A. Harvey and J. McDermott, "Rule Based Interpretation of Aerial Imagery," Dept. of Computer Science, Carnegie-Mellon University, (September 1984).

[MCK85] D. M. McKeown, W.A. Harvey and J. McDermott, "Rule Based Interpretation of Aerial Imagery", *IEEE PAMI*, Vol. PAMI-7, No. 5, September 1985, pp. 570-585.

[MCK86] D. M. McKeown, C.A. McVay and B.D. Lucas, "Stereo Verification in Aerial Image Analysis", *Optical Engineering*, Vol. 25, No. 3, March 1986, pp. 333-346, Also available as Technical Report CMU-CS-85-139.

[MIL79] D. L. Milgram, "Region Extraction Using Convergent Evidence," *Computer Graphics and Image Processing* 11, 1979, pp. 1–12.

[NAG79] P. A. Nagin, "Studies in Image Segmentation Algorithms Based on Histogram Clustering and Relaxation," COINS Technical Report 79-15, University of Massachusetts at Amherst, September 1979.

[NAG82] P. A. Nagin, A. R. Hanson, and E. M. Riseman, "Studies in Global and Local Histogram-Guided Relaxation Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **3** (May 1982), pp. 263-277.

[NAN87] N. Nandhakumar and J.K. Aggarwal, "Multisensor Integration–Experiments in Integrating Thermal and Visual Sensors", *Proc. IEEE ICCV*, London, England, June 8-11, 1987, pp. 83-92.

[NAS83] A. M. Nasif and M. D. Levine, "Low Level Segmentation: An Expert System," Technical Report TR-83-4 (April 1983), Electrical Engineering, McGill University.

[REY84] G. Reynolds, N. Irwin, A. R. Hanson, E. M. Riseman, "Hierarchical Knowledge-Directed Object Extraction Using a Combined Region and Line Representation," *IEEE Proceedings of the Workshop on Computer Vision: Representation and Control* (1984). pp. 238–247.

[REY85] G. Reynolds, D. Strahman, and N. Lehrer, "Converting Feature Values to Evidence", *Proc. of the DARPA Image Understanding Workshop*, Miami Beach, FL, 1985.

[REY87a] G. Reynolds, N. Lehrer and J. Griffith, "A Method for Initial Hypothesis Formation in Image Understanding", *Proc. of the DARPA Image Understanding Workshop*, Los Angeles, CA, February 1987. Also COINS Technical Report 87-04, Computer & Information Science Department, University of Massachusetts at Amherst, January 1987.

[REY87b] G. Reynolds and J. Ross Beveridge "Geometric Line Organization Using Spatial Relations and A Connected Components Algorithm", COINS Technical Report, Computer & Information Science Department, University of Massachusetts at Amherst, in preparation, 1987.

[RIS84] E. M. Riseman and A. R. Hanson, "A Methodology for the Development of General Knowledge-Based Vision Systems," *IEEE Proc. of the Workshop on Computer Vision: Representation and Control* (1984), pp. 159–170.

[RIS87] E. M. Riseman and A. R. Hanson, "A Methodology for the Development of General Knowledge-Based Vision Systems", in *Visions, Brain, and Cooperative Computation*, (M. Arbib and A. Hanson, Eds), MIT Press, 1987, pp. 285-328. Also COINS Technical Report 86-27, University of Massachusetts at Amherst, July 1986.

[SOU86] R. Southwick, "An Intermediate Representation of Image Feature Data," Master's Thesis, Computer and Information Science Department, University of Massachusetts at Amherst, February 1986.

[SHA86] S.A. Shafer, A. Stentz, and C.E. Thorpe, "An Architecture for Sensor Fusion in a Mobile Robot", *International Conference on Robotics and Automation*, San Francisco, CA, 1986, pp. 2202-2011.

[WEI86] R. Weiss and M. Boldt, "Geometric Grouping Applied to Straight Lines", *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami, FL, June 22-26, 1986, pp. 489-495.

[WEY83] T. Weymouth, J. Griffith, A. R. Hanson, and E. M. Riseman, "Rule Based Strategies for Image Interpretation," *Proc. AAAI-83* (August 1983).

[WEY86] T. E. Weymouth, "Using Object Descriptions in a Schema Network for Machine Vision," Ph.D. Thesis, University of Massachusetts at Amherst (April 1986). Also COINS Technical Report 86-24, Computer & Information Science Department, University of Massachusetts at Amherst, May 1986.

[WIL81] T. Williams, "Computer Interpretation of a Dynamic Image from a Moving Vehicle," Ph.D. Thesis and COINS Technical Report 81-22, University of Massachusetts at Amherst (May 1981).

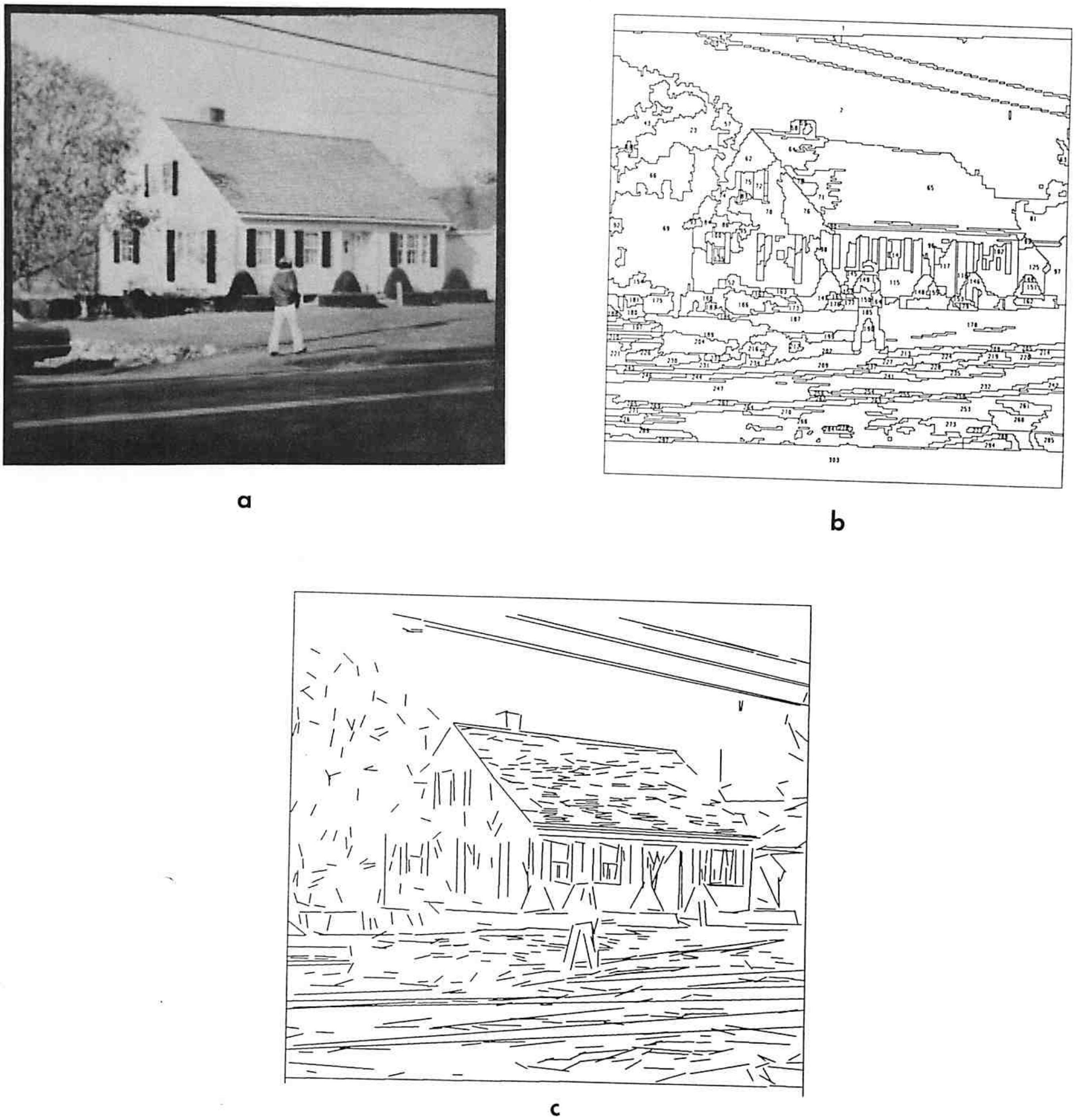


Figure 1. Segmentation Results. (a) a black and white rendering of the original color image; (b) regions produced by a segmentation system using localized histograms followed by region merging; (c) straight lines produced by an algorithm which uses similarity of gradient orientation as the primary organizational feature.

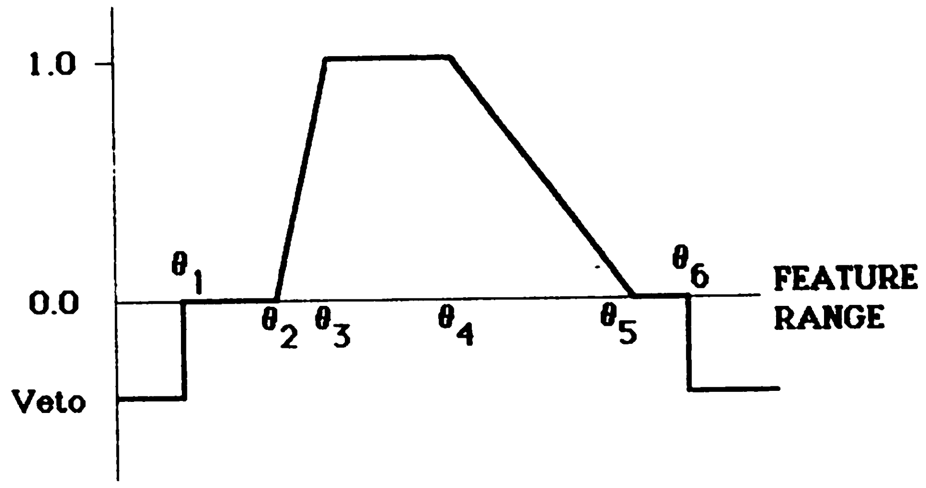


Figure 2. Structure of a simple constraint function as a piecewise linear function F mapping an image feature measurement into support for an object label hypothesis. It is specified via 6 points $\{\theta_i, i = 1, \ldots 6\}$.

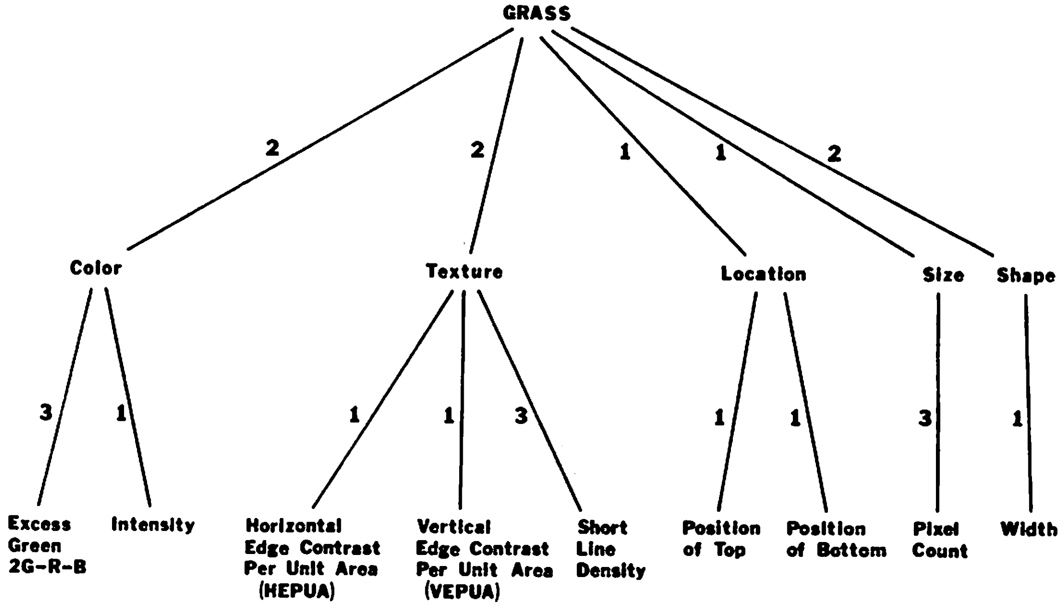


Figure 3: The structure of a compound constraint for grass showing the five component constraints defined on region attributes.

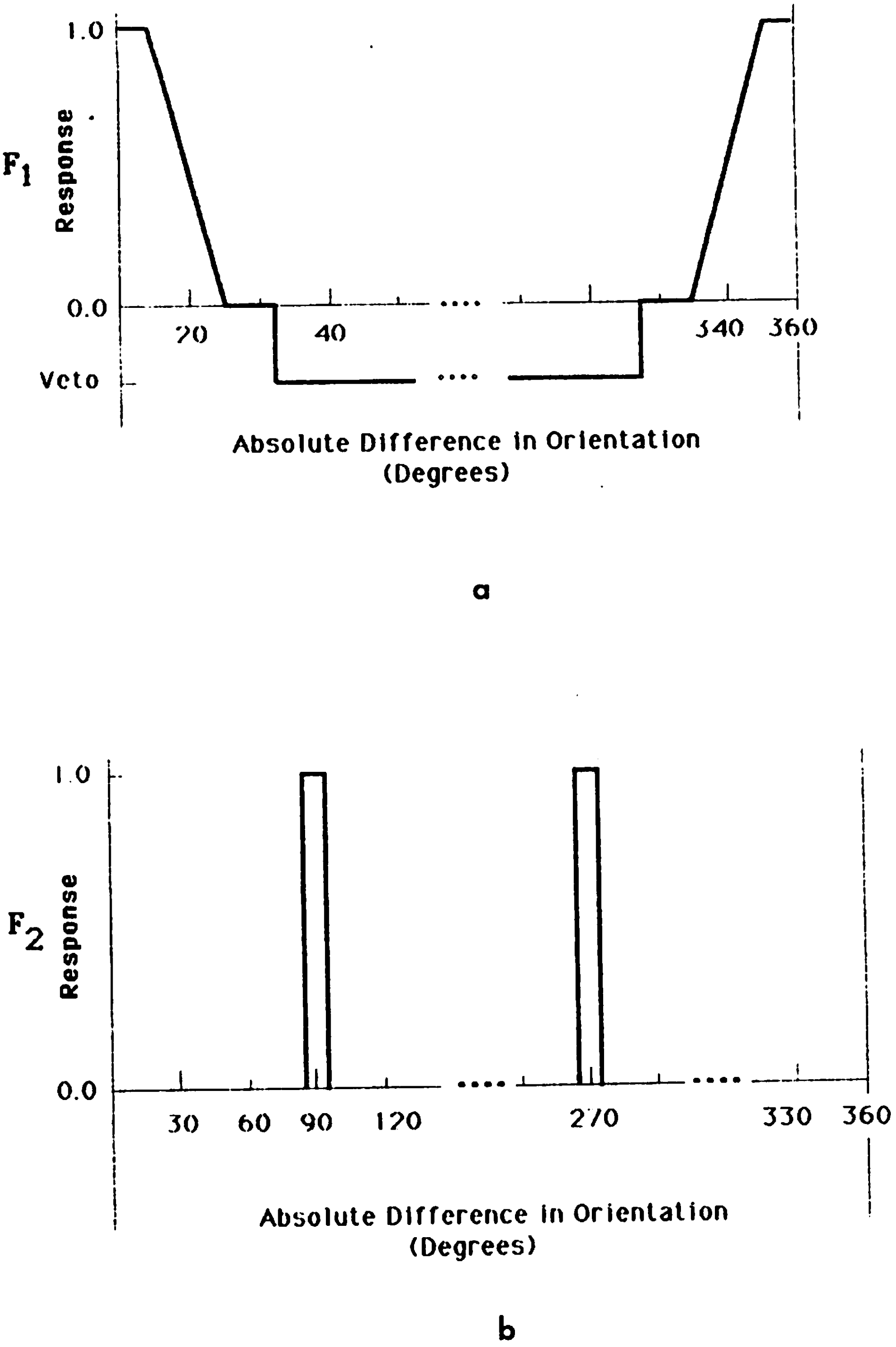


Figure 4. Example Constraint Functions on Relational Measures. The structure of two constraint functions for relating one line token to another is shown. The constraint is applied to the difference in orientation of two line tokens. (a) F_1 equally ranks all lines whose orientation is within ±10° of the given line; the response falls off as a function of the difference in orientation. (b) F_2 equally ranks all line tokens that are within ±5° of being orthogonal to the given token.

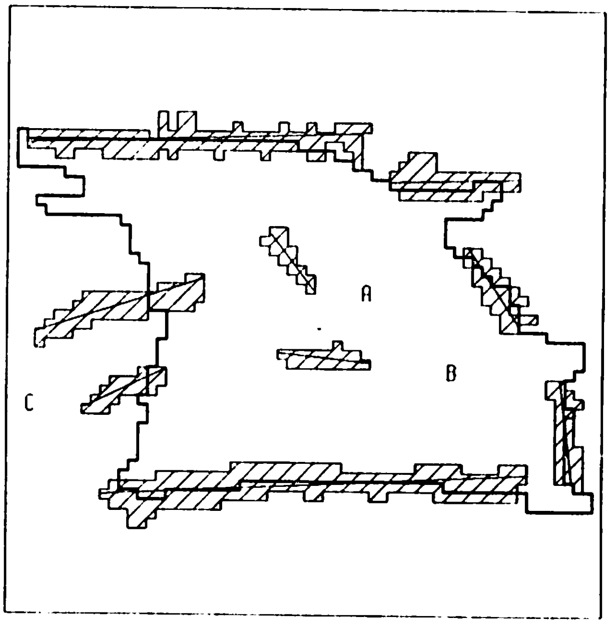


Figure 5. An intensity region with line-support-sets superimposed on the region; (note that line-support-sets are shaded). Examples of the three line categories are labeled: A - interior, B - boundary, C - Other.

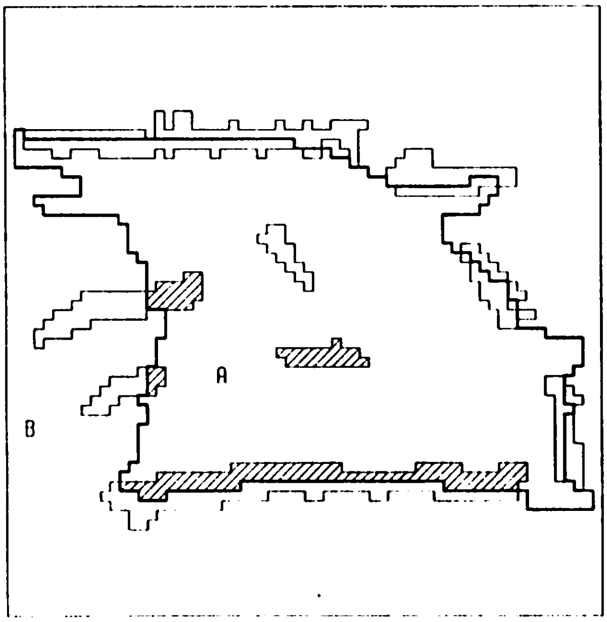


Figure 6. Example of Interior-Line-Percentage as a Relational Measure. For each line this measure is computed by dividing the number of pixels in the line-support-set which intersect the region (A) by the total number of pixels in the line-support-set (A + B).

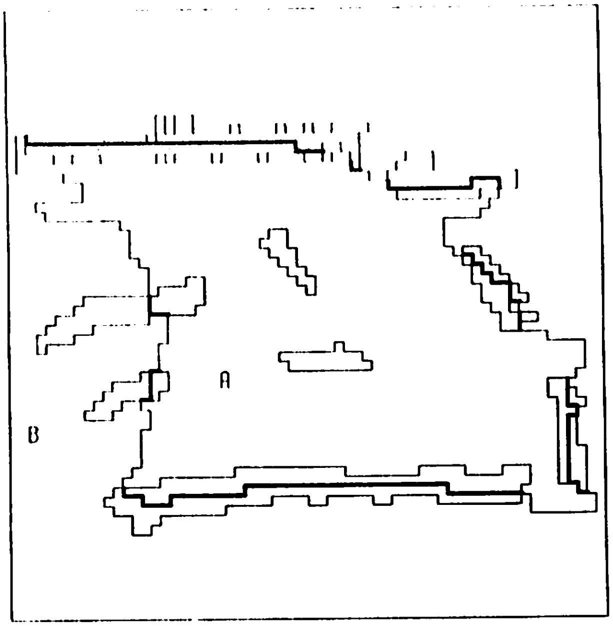


Figure 7. Example of Region-Perimeter-Percentage as a Relational Measure. It is computed by dividing the length of the portion of the region boundary covered by a line-support-set (A) by the length of the entire region boundary (B).

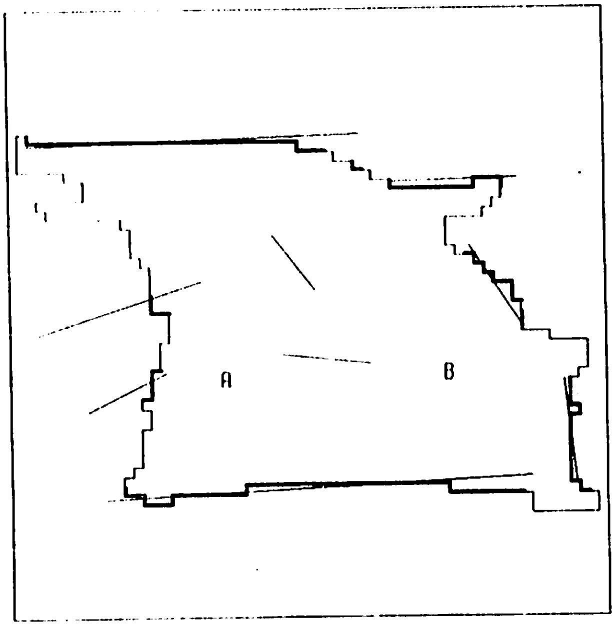
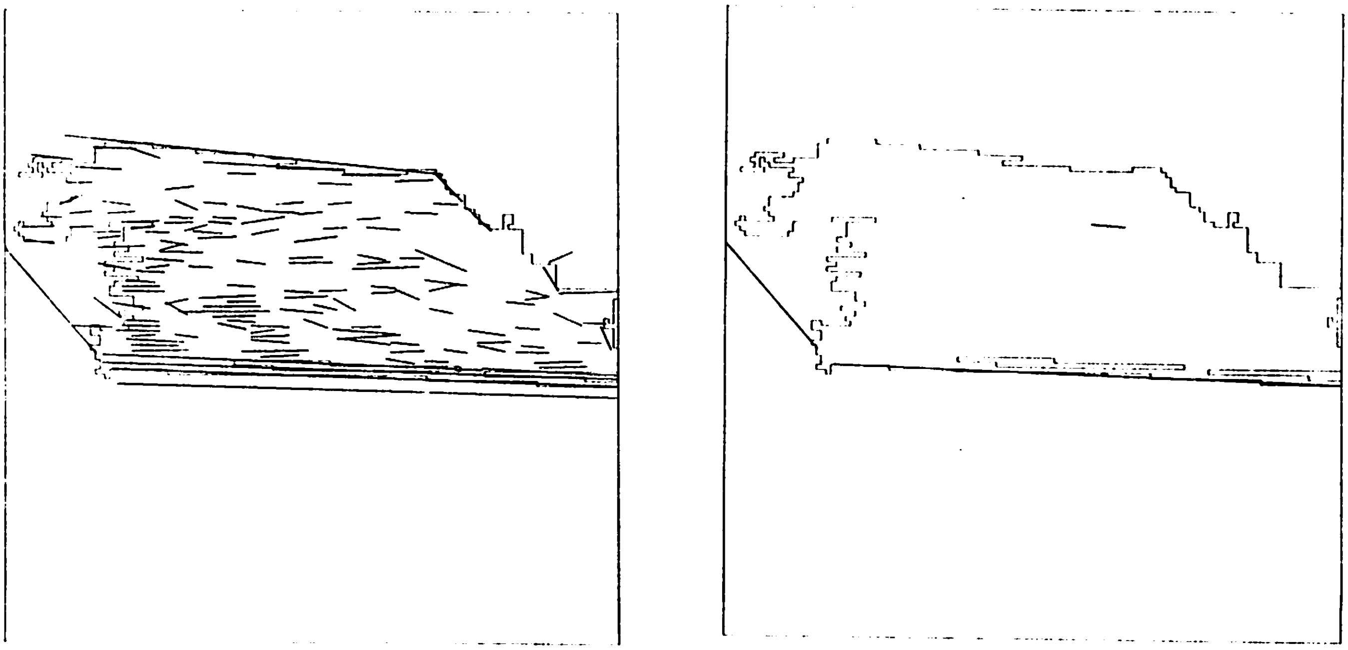


Figure 8. Example of Line-Boundary-Percentage as a Relational Measure. The feature is computed by dividing the length of the portion of the region boundary covered by a line-support-set (A) by the length of the line (B).



a

b

line	interior%	perimeter%	length%
A	100	0	0
B	7	1	13
C	48	20	100

c

Figure 9. Example Values of Relational Measures. (a) All the lines which intersect the main roof region; refer to Fig. 1b for a clearer view of the region being used. (b) the three lines being considered; (c) The values of the three relational measures for the three lines.

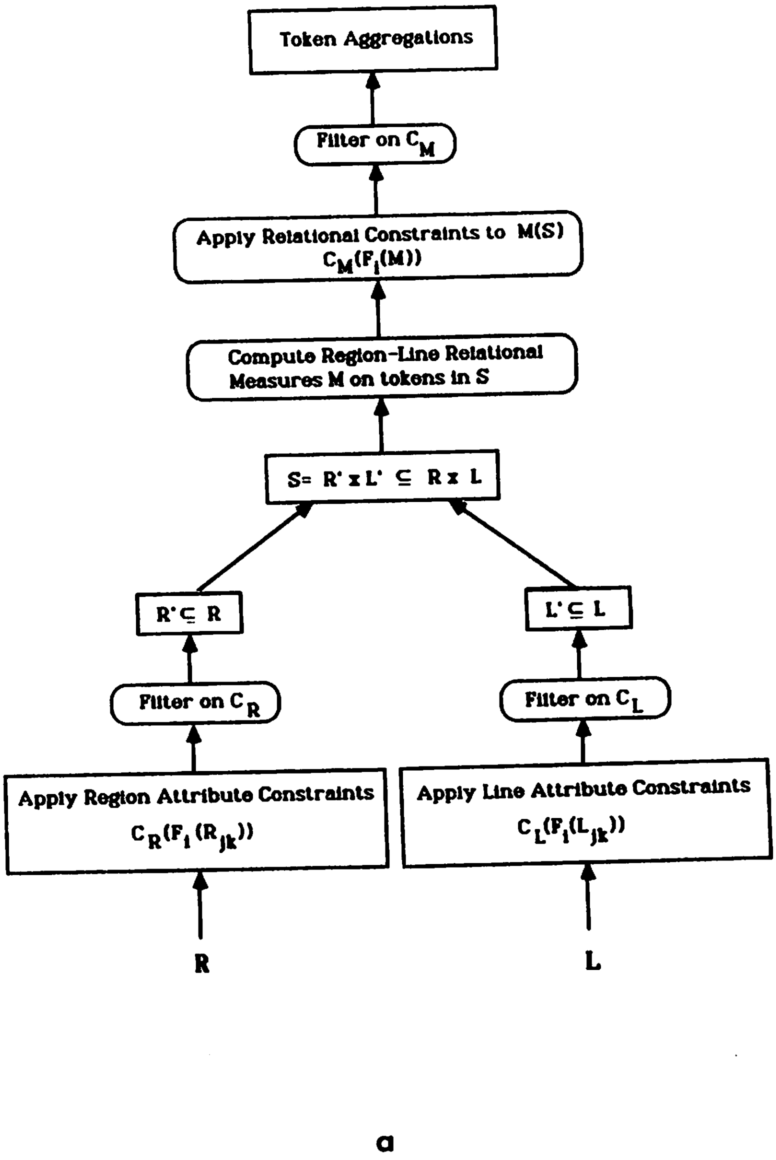


Figure 10. Examples of Control Strategies in Applying Constraints to Produce Aggregations. (a) Constraints on region and line attributes (C_R and C_L, respectively) are independently applied and then relational measures M and relational constraints C_M are applied to the filtered subsets of region and line tokens. (b) Constraints on regions are applied to produce a filtered subset of regions, which are then used to select a subset of lines via the intersection relations; line attribute constraints on the remaining subset of line tokens are used to further filter it before line-region relational information is employed.

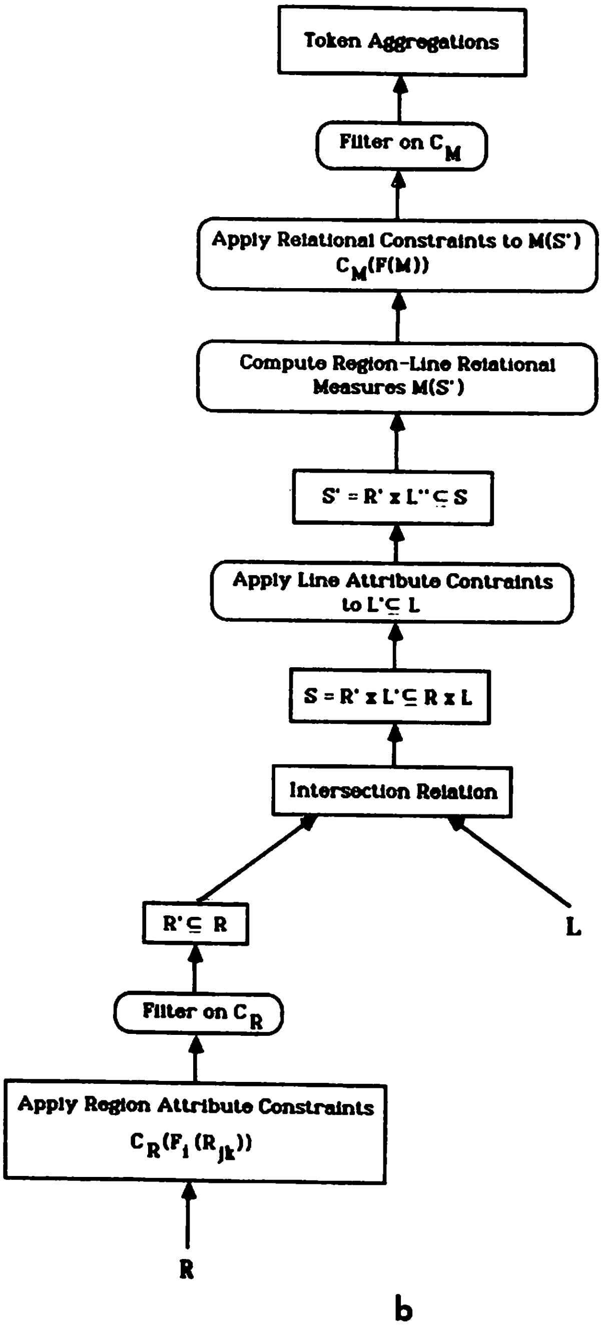


Figure 10,continued

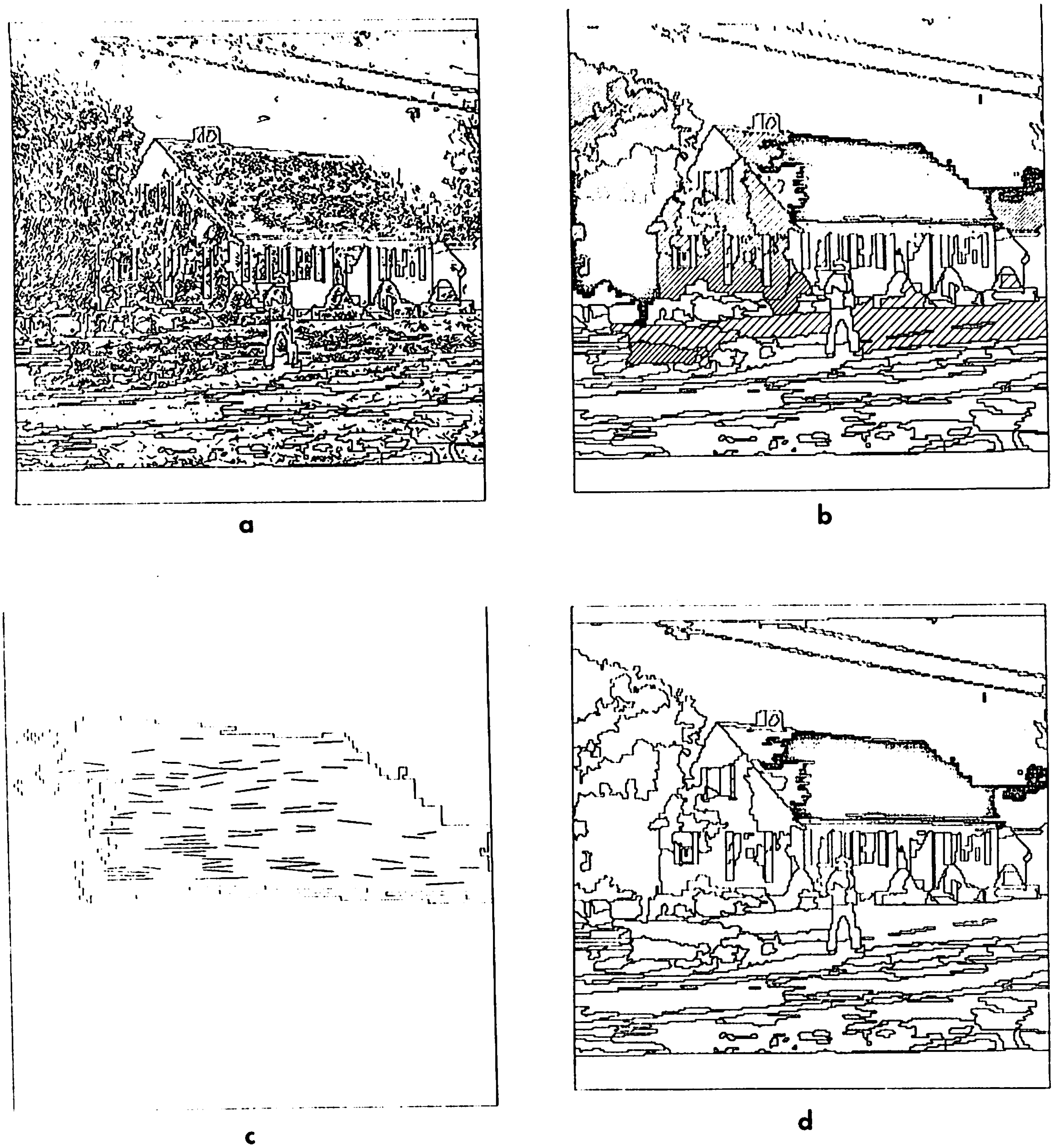


Figure 11. Example of Texture Measure for Extracting House Roof. A simple texture measure computed by a relational constraint based upon the density of lines within a region. (a) Lines which received a high score on the relational measure of interior-line-percentage (i.e. INTERIOR lines); (b) The density of interior lines for each region represented by the density of shading; (c) Horizontal INTERIOR lines for the roof region; (d) Density of horizontal interior lines.

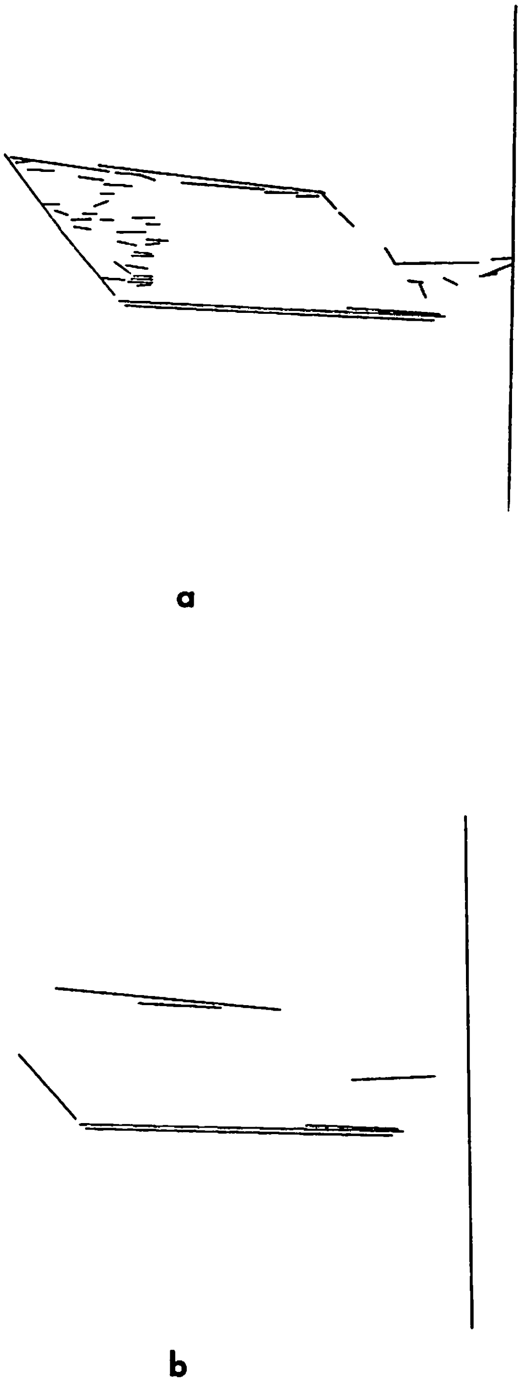
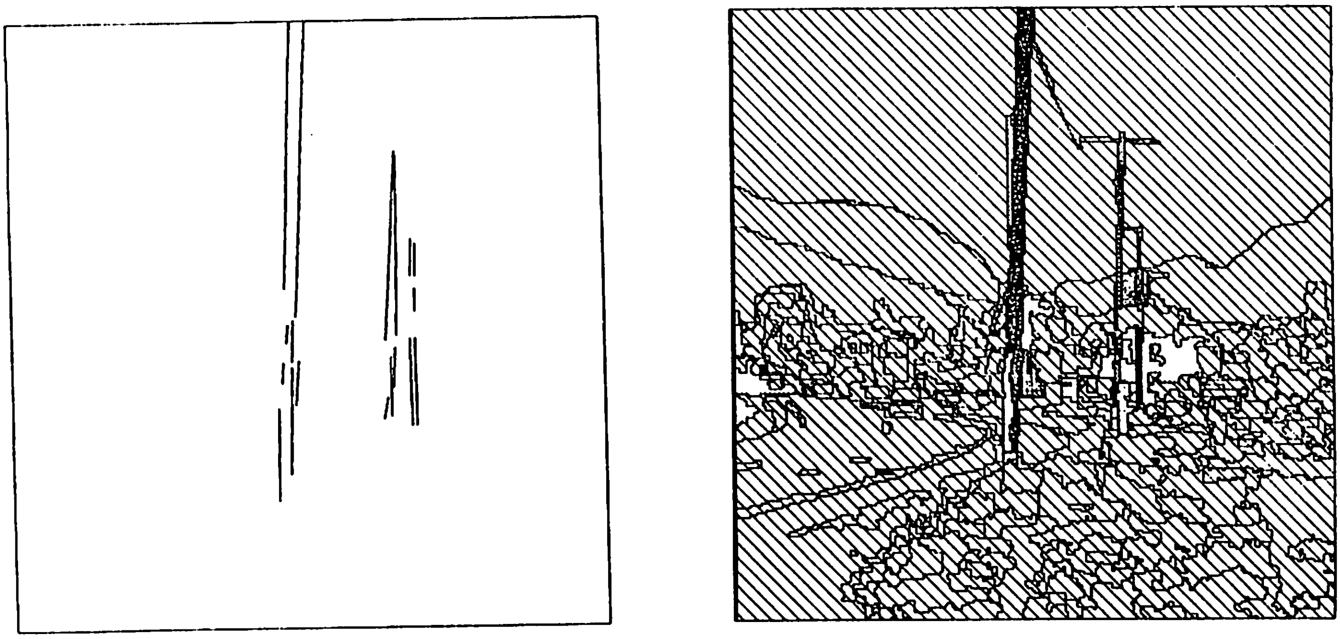


Figure 12. **Extracting Roof via Long Bounding Lines.** Given a possible roof region the long bounding lines can be filtered to find a roof shape. (a) Lines bounding the hypothesized roof region, and (b) Long bounding lines which can be the basis of forming an approximate parallelogram shape.



a

b

Figure 13. Example of Extracting Telephone Poles via Vertical Bounding Lines. (a) The line pairs formed by the line constraint of bounding vertical lines. (b) The relational measures are mapped back to the regions; hatched regions have no long vertical bounding lines and are vetoed.

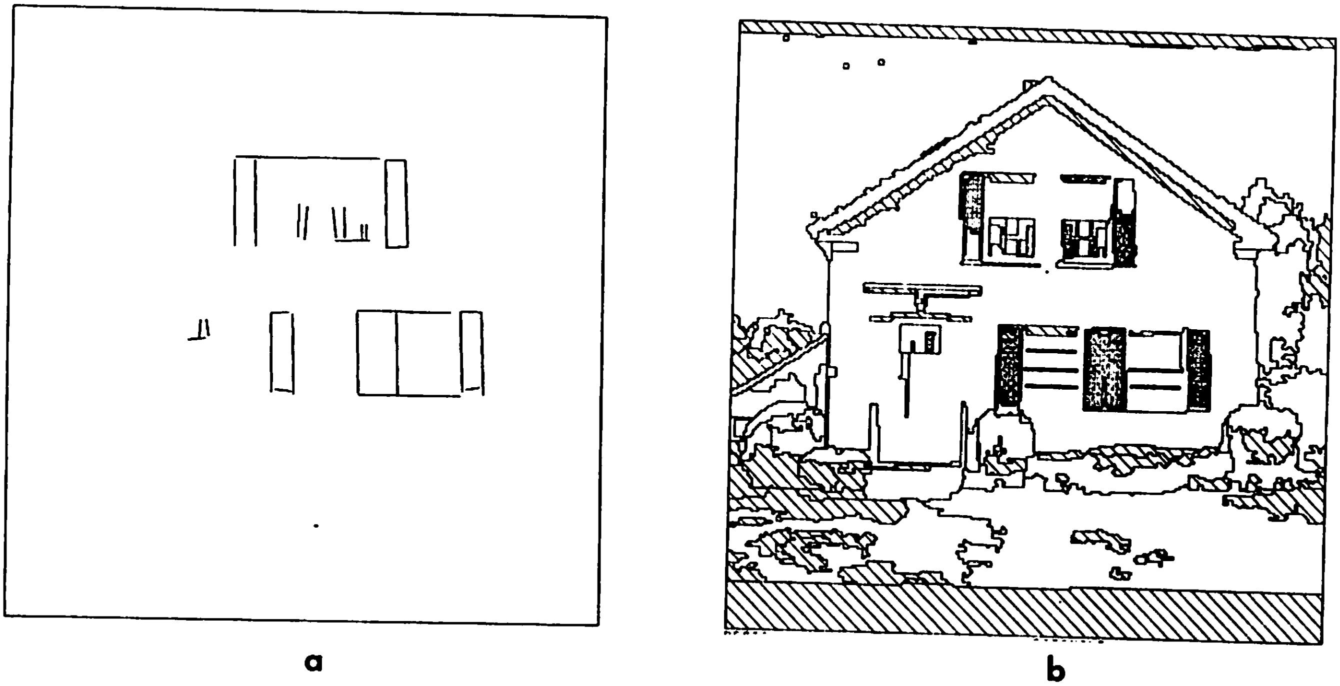


Figure 14. Extracting Rectangular Window and Shutter Hypotheses. (a) Horizontal and vertical bounding lines; (b) The regions produced by a relational constraint on the extracted bounding lines.

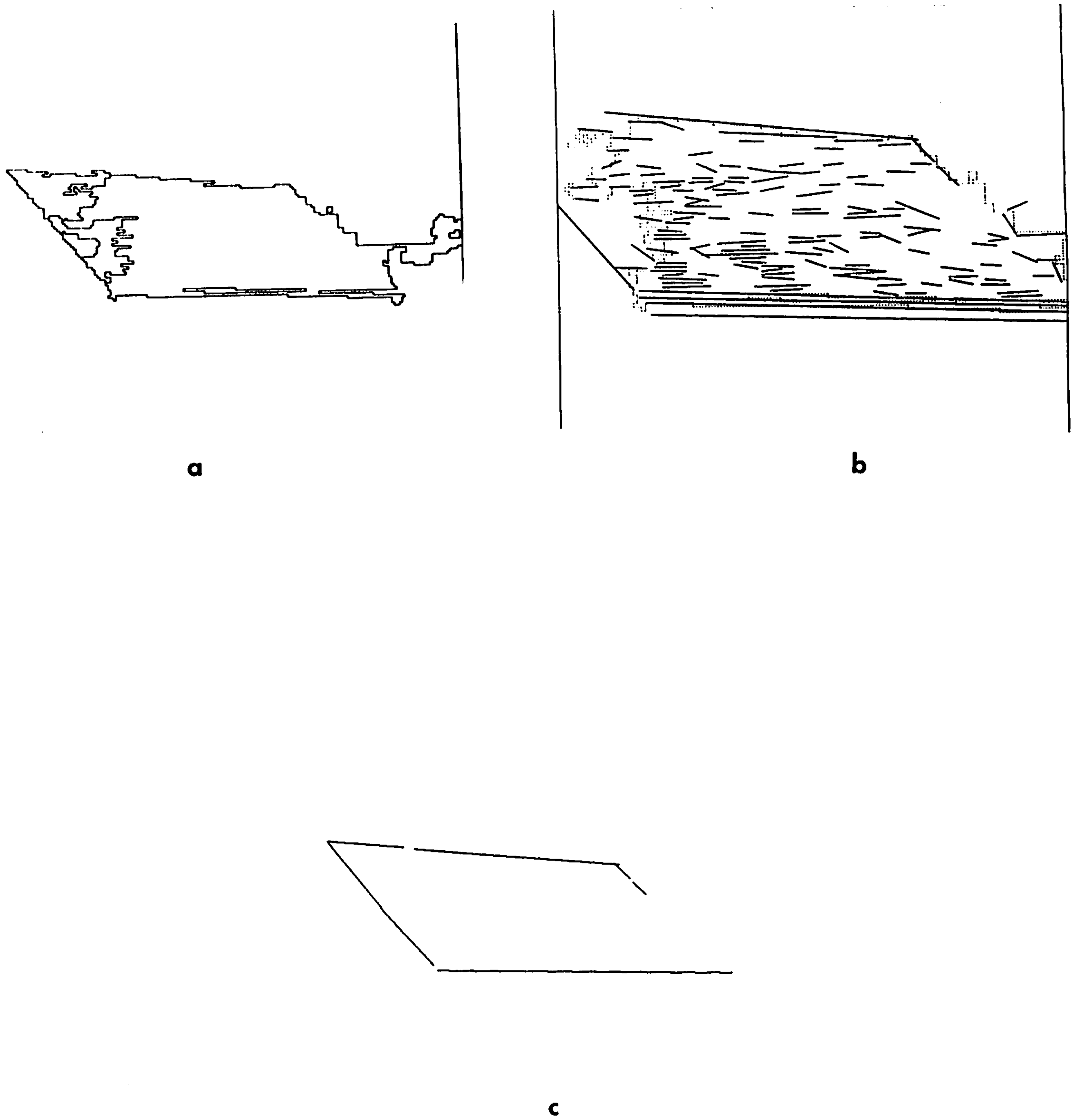


Figure 15. Example of difficulties in aggregating tokens. (a) The roof is fragmented in the region representation; (b) The lines intersecting the largest roof region only capture a portion of the relevant line information; note the two key missing lines at the upper left corner of the roof. (c) A subset of the full set of lines which, if they could somehow be selected, would provide the appropriate line aggregation to group the roof regions, and allow the roof outline to be completed in a straightforward manner [WEY86].

Type of Relational Measure	Value of Relational Measure		
	High	Medium	Low
Interior Line Percent	Interior	Bounding, Other	Other
Region Perimeter Percent	-----	Bounding	Interior, Other
Line Boundary Percent	Bounding	Bounding	Interior, Other

TABLE I