GOAL-DIRECTED CONTROL OF LOW-LEVEL PROCESSES FOR IMAGE INTERPRETATION

Charles A. Kohl
Allen R. Hanson
Edward M. Riseman

COINS Technical Report 87-31

April 1987

This work was supported in part by the Air Force Office of Scientific Research under grant AFOSR-86-0021 and the National Science Foundation under grant DCR-8318776.

GOAL–DIRECTED CONTROL OF LOW–LEVEL PROCESSES FOR IMAGE INTERPRETATION

Charles A. Kohl Allen R. Hanson
Edward M. Riseman

Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003

March 30, 1987

ABSTRACT

The task of image interpretation is viewed as a coordinated process in which high–level interpretation processes and low–level segmentation processes interact through what is known as the *intermediate–level* of processing. Control processes at this intermediate–level respond to requests from the interpretation processes which are expressed in terms of *goals*. The presentation of a request for low–level processing, which is represented by the posting of a goal, results in the creation of an intermediate–level control process which is an instantiation of a control structure known as a *schema*. The set of schemas represented at the intermediate level define the types of goals which may be processed at this level.

The instantiations of the schemas utilize knowledge of the image domain to translate the goals into appropriate low level process specifications which include the identification of the image features, algorithms, sensitivity settings, or rules to

[1]This work was supported in part by the Air Force Office of Scientific Research Grant AFOSR-86-0021 and the National Science Foundation Grant DCR-8318776

be used by the process to accomplish the desired task. These low-level process specifications are then passed to a low level process controller which directs the execution of the process and then returns the results to the intermediate-level control process.

Using this control paradigm, high-level interpretation processes gain the capability to create or refine segmentation data according to predefined goals and/or current hypotheses regarding the content of the image, and thereby improve the quality of the overall interpretation.

1. THE INTERMEDIATE LEVEL OF IMAGE INTERPRETATION

Image interpretation is a complex process through which a numeric array, representing a digitized visual scene, can be analyzed to provide a semantic description of the scene content. One subgoal of the interpretation process is **Image segmentation**, the low-level process by which the digitized image is abstracted into a set of primitive elements which may be used by interpretation processes as the basis for the construction of an abstract symbolic model of the original scene. According to many commonly accepted views of interpretation, image segmentation is simply the first stage of the interpretation process. We take the view, however, that segmentation is a process which does not exist in isolation, but rather is an integral part of the overall image interpretation process.

This concept of the interdependence between the segmentation and interpretation processes has developed slowly over the years as researchers have become aware of the difficulty of each of the phases of processing. Researchers such as Marr ([20]), Brooks and Binford ([3]), Thompson ([34]), Reynolds ([29]), Hwang ([15]) and Hanson and Riseman ([30]) have all stressed the need for interaction between high and low level processes.

To date, however, little has been done to address this problem, and thus there are no commonly accepted protocols for the interaction. If we are to adequately deal with interaction between high and low-level processes, there are three separate issues which must be resolved. First, there must be a common knowledge representation which is applicable to all levels of the interpretation process; intercommunication between processes at different levels is difficult or impossible if data objects and hypotheses are not consistent across the (somewhat artificial) boundaries between levels. Second, there is the necessity for a single unified control mechanism which can coordinate and schedule the activities of high and low-level processing tasks. Finally, there is the requirement for an explicit representation of the knowledge necessary for the effective application of low-level processes (i.e. the selection of the appropriate algorithms, sensitivity settings, and image features). For example, we require a mechanism which can encode the knowledge which indicates that certain image features and algorithms are generally useful when segmenting textured image areas, while others are more useful when segmenting smooth image areas.

In this report, we outline the design of a system which has been developed within the VISIONS Image Understanding System ([17]) to deal with these three issues and accomplish the integration of the the high and low levels of the image understanding process. The system is designed to provide the mechanism by which high-level requests for data may activate a set of appropriate low-level processes which may be capable of producing the desired data. The specification of the data request is in the form of a *goal*, a data structure which defines the nature of the image abstractions desired by the requesting process. Attributes stored within the goal data structure, known as *constraints*, express additional information which may be useful in the specification of the most appropriate low-level processes. The goal constraints express the desired characteristics of the data to be produced, and may be represented in either semantic or syntactic terms. The semantic constraints are defined in terms of semantic labels (e.g. *"segment a specific portion of the image to separate tree and sky"*), while the syntactic constraints are expressed in terms of measurable image features (e.g. *"produce regions which exhibit homogeneous texture measures"*). Through the use of the information expressed in these constraints, the system is able to select the most appropriate image features, algorithms, sensitivity settings, rules, etc., for the specification of the low level task.

In VISIONS, the interface between the high and low levels of the interpretation process is called the *intermediate* level; data at this level is represented by *tokens*.

Tokens may represent any abstraction of the raw image data which is in spatial registration with the actual data and/or groups of these abstractions which have been formed by the application of grouping operations ([35,2,28]). Thus, intermediate level tokens may represent any of the normal types of segmentation output, such a regions, lines, or surfaces. The intermediate level is distinguished from the low level in that there is no explicit representation of pixels, and from the high level by the requirement that tokens are in spatial registration with the image data.

The GOLDIE (Goal Directed Intermediate-Level Exectutive) system ([16]) has been developed as a mechanism to provide this intermediate-level control of low-level processing. The basic control structure of GOLDIE is the *schema*, a declarative specification of *control* strategies which may be used by the system to satisfy a specific goal. Schemas provide a flexible and extensible control structure which is used within VISIONS to direct both the high and intermediate processing levels ([36,27,11]), and they provide a natural interface between processes at each of these levels. GOLDIE extends this concept of schema-directed control to the low-level processes.

2. THE NEED FOR GOAL-DIRECTED CONTROL

Low-level segmentation processing is in many ways more of an art than a science. Despite an enormous number of excellent techniques and algorithms (e.g. see [31,14,1,8,10]), no general methods have been found which perform adequately across a wide variety of images. Methods such as region-growing ([12]), histogram cluster labeling ([22]), thresholding ([18]), zero-crossings ([13,20]), and edge and line detection ([5]) have all been found to be effective within restricted domains, but rarely demonstrate the robustness necessary for generalized image interpretation.

The failure of general segmentation techniques to provide "good" segmentations can be traced to a variety of causes. In many cases the image data itself is complex because of the physical situation from which the image was derived and/or the nature of the scene. Heavy textures, unconstrained lighting, view angle, surface reflectivities, markings, and curvature all conspire to produce ambiguous image

Figure 1: Intensity Image of Outdoor Scene

data. This results in under– or over–merged regions in the region segmentations and missing, fragmented, or incorrectly merged lines in the edge/line segmentations. The type of error often depends upon which feature(s) is being used to produce the segmentation and whether the algorithm has a global or local view of the image data during processing. The type of error produced quite often varies spatially within a single image, depending on the type of data in each locality of the image.

To take a specific example, the intensity surface of an image (Figure 1) may contain a great deal of high frequency information which typically will lead to significant *fragmentation* in the segmentation such as shown in Figure 2. Here we see that the region segmentation, which was produced through a zero–crossing algorithm ([20]), has produced a large number of small regions across the textured projection of the tree. While these regions do represent image areas of local homogeneity, in most cases they are of little semantic interest. If our overall interpretation goal is to identify the gross objects in the scene, a segmentation such as that shown in Figure 3, which is produced through a one–dimensional histogram clustering algorithm, would be much more useful in the identification of the trees in the image.

In the case of the smoothly varying or flat surfaces of this image, however, the opposite is true. Here, the slow gradients of the intensity surface result in

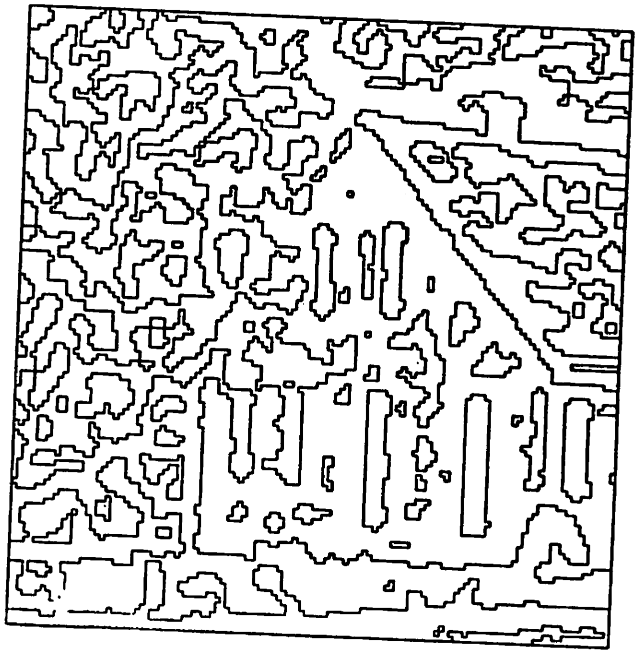

Figure 2: Zero–Crossing Segmentation

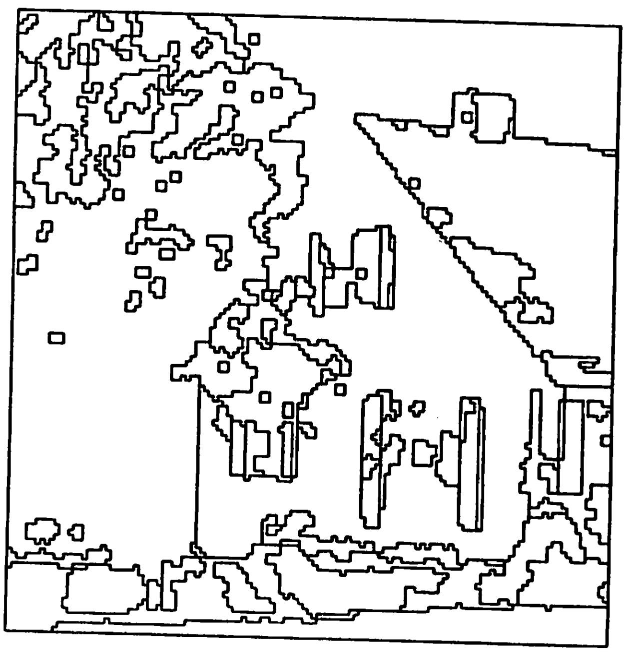

Figure 3: Histogram Clustering Segmentation

missing object boundaries in the segmentation produced by the clustering algorithm, resulting in the class of segmentation error known as over-merging. An example of this class of error is the lack of a boundary between the sky and house wall in Figure 3.

Existing image interpretation systems attempt to compensate for these two types of error within the high-level interpretation process. Undermerging errors are typically corrected with grouping processes which are used to merge adjacent regions with similar labels ([37,33]) . Overmerging errors are typically bypassed implicitly through the use of a segmentation which has such a fine resolution that it can be assumed that any object boundary which does exist in the image has a corresponding region boundary in the image ([9,19,25]).

However, since both types of errors may occur simultaneously in different areas of any given segmentation, it is our assertion that error correction properly belongs within the domain of the low-level segmentation processes themselves. As an interpretation process discovers problems with a segmentation, the segmentation itself should be redefined to resolve those problems. This redefinition of the segmentation may require the application of a different segmentation algorithm, or of the same algorithm with different sensitivity settings, over the portion of the image which presents the difficulties for the interpretation process.

Another factor leading to variability in low-level segmentation processing has to do with the *choice of image features* over which the segmentation processes will run. In typical RGB images, a large number of computed image features can be defined, some of which are known to be effective in segmentation processing ([24,25,32,21]). However, in a large number of segmentation algorithms, only Intensity (the average of R, G, and B) is used. In other algorithms, the selection of of features is typically made in an *ad hoc* fashion, where the general experience of the person designing the segmentation algorithm is brought into play. If, however, such knowledge and experience were represented explicitly in the system, it would be possible to perform this selection automatically in response to the overall goals of the interpretation process.

Finally, there is the problem of *evaluating* different region/line segmentations.

Despite some efforts to quantify the evaluation of segmentation processing ([23,26]), it has been our experience that no low-level evaluation measure restricted to making measurements on the segmentation can provide a useful comparative metric. Rather, the quality of a segmentation can be measured only with respect to the goals of an interpretation process which is applied to the segmentation data. Different criteria must be used to evaluate the quality of segmentation results on highly textured image areas than on smoothly varying image areas.

In view of this variability in the selection of different segmentation methodologies and the difficulties of evaluating the results of various techniques, it becomes apparent that some form of intelligent intermediate-level control is necessary. We believe that the most effective form for this control is a mechanism which can select and apply the most appropriate segmentation process for each distinct area of the image. Through the use of an initial region segmentation of the image, these distinct image areas may be coarsely distinguished, and then through an iterative process of resegmentation and merging (using appropriate algorithms and rules), the areas may be more precisely identified. In other words, our methodology is to form a hierarchical set of segmentation plans in which the results of early processing are combined with the intermediate results of the interpretation process and used to guide and constrain the later processing.

The knowledge required under this control paradigm provides the rules and information by which the goal constraints may be used to determine the utility of the various low-level processes available to the system. In the specification of these low-level processes, the control process is able to choose among a wide array of possible intermediate and low-level tools which include:

- a set of parameterized segmentation algorithms,
- a set of low-level image analysis and enhancement routines,
- a set of measurable image features,
- processes for accessing and modifying the tokens and their associated attributes in the intermediate-level database, and
- a formalized specification of segmentation goals.

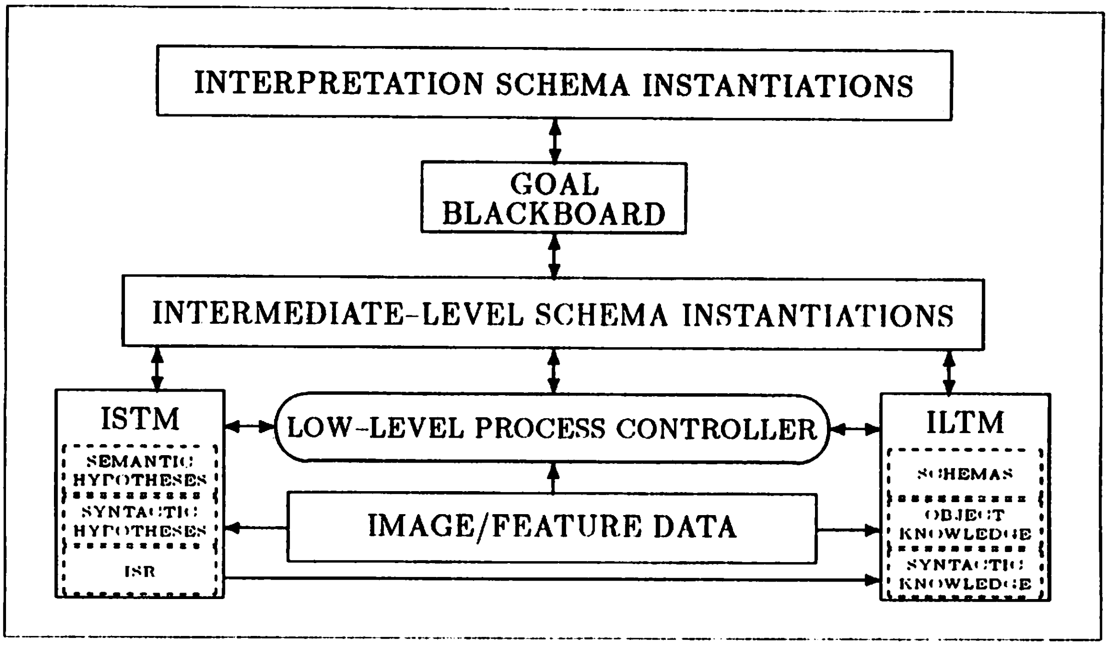

Figure 4: Overview of the GOLDIE System

In the next section, we demonstrate the mechanisms by which GOLDIE represents knowledge of these processes and tools, and how it uses the knowledge to control low-level processing.

3. INTERMEDIATE-LEVEL KNOWLEDGE AND CONTROL

The GOLDIE system, outlined in Figure 4, may be described in terms of five major functional components: the low-level process controller, the data structures for intermediate-level tokens and hypotheses (ISTM), the representation of explicit intermediate-level knowledge (ILTM), the representation of the control state of the system (Goal Blackboard and Schema Instantiations), and the actual control process which manages the system (represented in the figure by the arcs between the other modules). In this section, we will present a brief description of each of these components, and then demonstrate the mechanisms by which the components are combined to produce a working system.

All of the actual image processing that takes place within this system is managed by the *low-level process controller*. The controller maintains a consistent protocol for the activation of low-level processing tasks, permitting the schema representation of low-level processing to be expressed in a manner independent of the actual procedural implementation of these tasks. Thus the addition, deletion, or modification of low-level processing tasks may be accomplished with minimal modification of the schemas which control the tasks. Since many low-level tasks produce intermediate-level data in the form of tokens, the controller is also responsible for the maintenance of a data structure which encodes the relations between tokens and the processes by which they were created. This data structure provides the data necessary for backtracking if the schemas decide at any point that specific segmentation processing should be "undone". Thus if several different attempts were made to segment an area of the image, but only one of these was found to be acceptable, this mechanism allows the deletion of the tokens produced by the unacceptable segmentation processes.

The dynamic *data structures* of the GOLDIE system are the sets of tokens and hypotheses about tokens which are stored in ISTM (Intermediate-level Short Term Memory). The tokens themselves are defined through the ISR (Intermediate Symbolic Representation), a combination database and semantic network in which tokens are defined as graph nodes with associated bitmaps which represent the spatial extent of the token in the image. Attributes of the tokens, such as their compactness measure or mean feature value, are computed on demand by the ISR and then stored in the database for efficient retrieval by any other process which may require the data. Relations between tokens, such as boundary contrast between two adjacent region tokens for a given image feature, are represented as arcs which have values indicating the nature of the relation. Tokens may be directly accessed by name or associatively accessed through the specification of constraints on their attributes.

ILTM (Intermediate-level Long Term Memory) encodes the image-independent *intermediate-level knowledge* structures of GOLDIE which includes schemas, object domain knowledge, and general knowledge of image syntax. These structures form the core of the system in that they contain the knowledge which permits the intermediate–level schemas to utilize low–level processes to achieve high–level goals without explicit high–level control.

As was stated earlier, the schemas of the ILTM are the modular representation of the set of strategies which may be used in an attempt to satisfy the processing requests we call goals. Each schema is designed to serve one particular type of goal, and thus there is a one to one correspondence between the goals that can be recognized by the GOLDIE system and the schemas of the ILTM. The strategies of the schemas specify the various low or intermediate–level tasks through which the goal may potentially be satisfied. The execution of the strategies will typically involve a sequence of operations which may include the posting of subgoals, the execution of image processing tasks through the low–level process controller, and the construction of hypotheses based on data represented in the ISTM. Message–passing constructs allow the executing strategies to communicate and synchronize their activities.

Another knowledge structure of the ILTM is the object domain knowledge which encodes information about the appearance of objects in an image which can be useful in the specification of segmentation tasks. This knowledge is represented as a hierarchical set of graph nodes corresponding to specific semantic objects or classes. Associated with each node is the set of features which have been experimentally shown to best discriminate that object from all other objects in a training set of images. Each of these nodes also contains heuristic information in the form of a syntactic (i.e. feature based) characterization of the object (e.g. textured, smooth, gradient), the segmentation resolution at which the object may be best extracted (high, medium, or low), and a set of rules which can be used by the intermediate–level system to provide crude hypotheses about the possible semantic identity of a particular region token.

ILTM also contains the system's knowledge of image syntax. This knowledge encodes the heuristic information which permits the evaluation of a token, without any use of semantic information, to create a hypothesis as to whether further processing is required. This knowledge includes the set of types of regions (textured, smooth, gradient) or lines (low-contrast, high-contrast, horizontal, long, etc.), and the set of rules to be used to syntactically evaluate each of these types of token. The results of the application of these rules are used to create syntactic hypotheses about the tokens which indicate the degree to which the image feature data supports the existence of the token (e.g. a set of rules which are to be used to determine whether a region token which is categorized as textured should be resegmented).

Syntactic knowledge also has to do with the utility of specific image features and algorithms in the creation of tokens with specific characteristics. For example, knowledge structures in the ILTM encode experimentally derived information that indicates that certain features and algorithms are generally useful when segmenting a textured region, while others are useful when segmenting smooth or gradient regions.

The actual control of processing in GOLDIE is managed through another semantic network which represents the *control state* of the system. Goals and schema instantiations are represented as nodes of this network, and arcs represent the relations between these entities ([36,7,6]). The goal blackboard is a section of the network in which goal nodes, representing requests for intermediate-level processing, are created by the schema instance that requires the results of that processing. Any constraints on the request are expressed as attributes of the goal node. By using an attribute–value list to express these constraints (Figure 5), the schema instantiation which is posting the goal can express any information which may possibly be of use to the responding schema instance. The schema instance which responds to a goal will extract the value of any named attribute which it can utilize in processing.

The *control process* of the GOLDIE system continually monitors this goal blackboard for the existence of new goal nodes. As a new goal is observed, the control process creates a new instance of the corresponding schema. The node for this schema instance is linked to the goal node by an arc which represents a contract by the instance to attempt to satisfy the goal. The communication between invoking and invoked schema instances is achieved through a mailbox on the goal node. When the invoked schema instance has obtained results which may satisfy the goal, these results are placed in the mailbox of the goal. The invoking schema process is

Region-Segmentation
'((*region-token* .Region-000042)
(*objective* . object-hypothesis-conflict)
(*evaluation-constraint* . nil)
(*object-labels* . (tree grass))

Figure 5: Goal Node Representation

then able to examine the results and either accept them as satisfying the goal, or request that additional strategies be used in an attempt to produce more acceptable data.

As a way of minimizing the number of unacceptable results, many schemas are capable of using a specific type of constraint known as the evaluation constraint. This constraint is expressed as a function–value pair which is used to evaluate results prior to their being returned to the invoking schema instance. If the value returned from the application of the function to a potential set of results does not exceed the threshold value expressed in the constraint, the invoked schema instance will automatically continue processing until a set of results is found which can meet this constraint. This mechanism makes it possible to tune the schema processing to highly specific goals. For example, an evaluation constraint could be created such that the region segmentation schema would return only segmentations which contained regions of a certain shape or color, or both.

A typical scenario illustrating the interaction between GOLDIE and the high-level interpretation system could involve a situation where a high-level interpretation process had formed two conflicting hypotheses regarding the semantic label to be assigned to a particular region token. Under the assumption that the region token actually represented an area that contained two different objects, the interpretation schema instance would post a goal (Figure 5) for region segmentation. The constraints on the goal would indicate the region token of interest, and would additionally indicate that the reason for the goal posting was that there was an object hypothesis conflict for the two specified object labels.

In response to this goal posting, the system would activate an instance of the region segmentation schema. This schema instance would select image features and algorithms appropriate for the discrimination of the two objects and then initiate a segmentation process which would hopefully split the original region into two new regions, each representing one of the two objects. Tokens corresponding to these two regions would then be returned to the interpretation schema instance for evaluation. If the interpretation schema instance were satisfied with the results, the goal and region segmentation schema instance would be deleted. Otherwise, the schema instance would continue processing until an acceptable segmentation were found or until the strategies of the region segmentation schema instance were exhausted.

4. INTERMEDIATE–LEVEL SCHEMAS

The set of intermediate–level schemas which have been implemented in the GOLDIE system are shown in Table 1 and Figure 6. Although the figure implies a hierarchy of schemas in the system, this hierarchy is designed only to demonstrate the relationship between the initialization schema and the other schemas of the system. We will use this hierarchy as the basis for our discussion, although it is important to recognize that the actual interaction between the schemas is more complex than indicated by Figure 6.

4.1 The Initialization schema

At the highest level of the hierarchy is the **initialization** schema. This schema is typically intended to be invoked at system startup to provide the initial set of image tokens for the interpretation process. Since the interpretation schemas of the high–level system are designed to operate across sets of tokens representing image abstractions, no interpretation processing may take place until we have an initial segmentation produced by the initialization schema.

The intent of this schema is to produce the "best possible" segmentation of a region token without any assistance from interpretation processes. If used to provide the initial segmentation data, the only constraints on the satisfaction of the

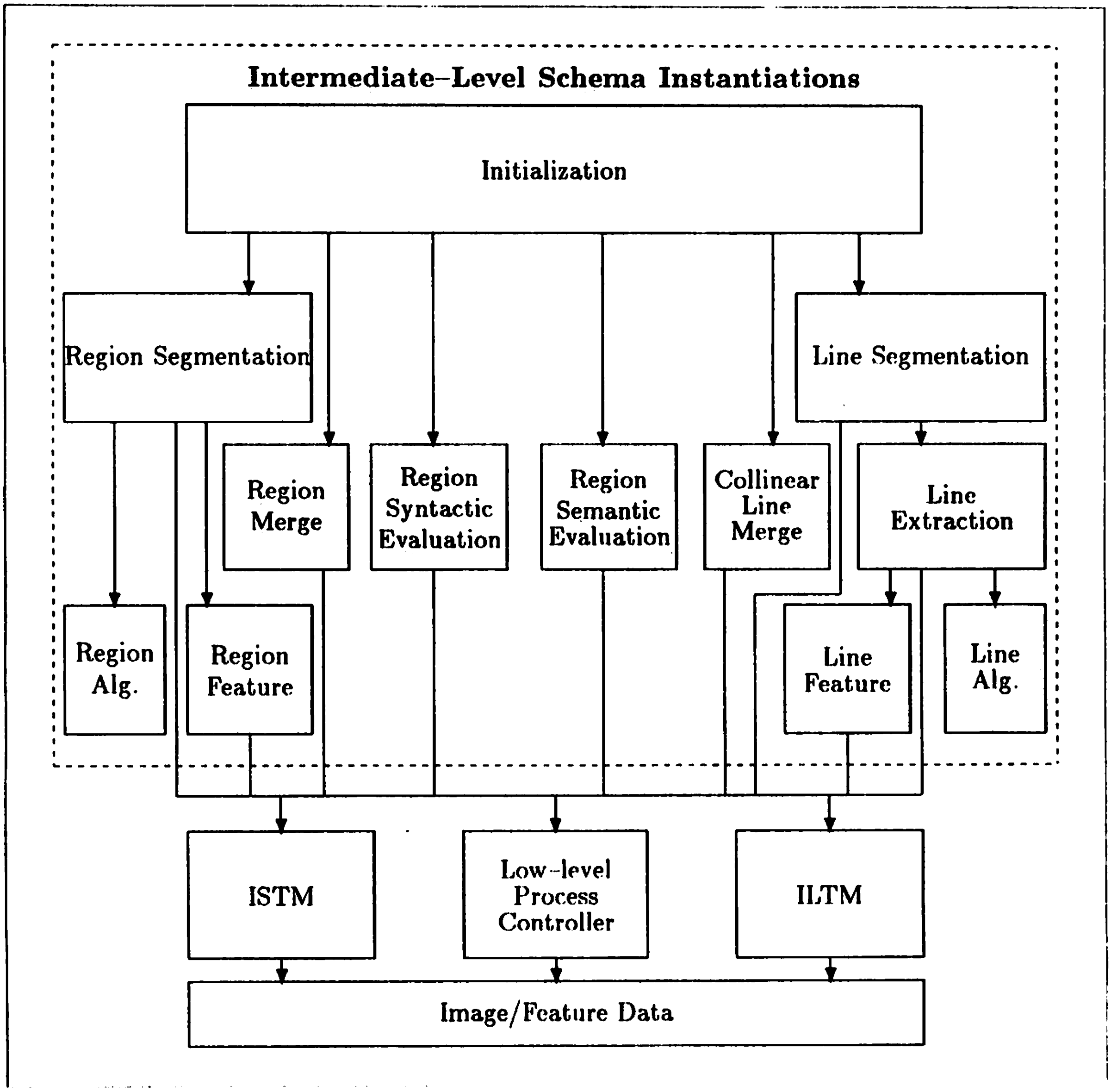

Figure 6: Schemas of the GOLDIE System

Schemas of the GOLDIE System	
Schema Name	Action
initialization	Obtain the "best" segmentation of the specified region token (possibly the whole image) with respect to (possibly initial) specified constraints
region segmentation	Segment a region token
region-algorithm	Select an algorithm for region segmentation
region-feature	Select an image feature for region segmentation
region merge	Merge adjacent region tokens which meet criteria selected as a result of the constraints
syntactic region evaluation	Evaluate region tokens for resegmentation or merging with respect to specified constraints
semantic region evaluation	Provide crude hypotheses for the semantic identity of the specified region tokens
line segmentation	Segment region tokens using evidence from a (set of) line token(s)
line extraction	Extract line tokens from image data
line-algorithm	Select an algorithm for line extraction
line-feature	Select an image feature for line extraction
collinear line merge	Merge adjacent line tokens which meet the criteria selected as a result of the constraints

Table 1: Intermediate-Level Schema Descriptions

initialization goal are the *a priori* constraints of the overall system. For example, if we were implementing a system whose primary goal was to identify different types of trees in outdoor scenes, the *a priori* constraints on the goal for initialization would be "*emphasize regions with texture characteristics*" and "*emphasize tree objects*". In this way, the initial segmentation which is presented to the interpretation system has already been at least partially tuned to the overall goals of the interpretation process.

The initialization schema directly or indirectly makes use of all of the other intermediate-level schemas of the system in this attempt to provide the "best" data-directed segmentation, and is the most complex schema present in the GOLDIE system. Although the schema contains only a single strategy, the execution of this strategy involves region segmentation, region merging, line segmentation, region evaluation, and recursive invocation to achieve the desired result.

The initialization schema requires a region token as input to define the area over which the segmentation is to take place. In the case where the schema is being used to initialize the system, the bitmap for this region token is defined as the entire image. The first step of the strategy is the posting of a goal for region segmentation on this region token. Unless overridden by its own goal constraints, the initialization schema instance will specify an evaluation constraint on the region segmentation goal which makes use of the syntactic evaluation hypotheses created by an instance of the syntactic region evaluation schema. The function associated with this evaluation constraint requires that the majority of the region tokens produced are hypothesized to be "good". In this context, "goodness" would indicate that no further segmentation was necessary, and thus the function computes a value based in part on the resegmentation hypotheses associated with each of the region tokens in the segmentation. Obviously, if this were the only factor, the best segmentation would be one in which each pixel were represented as an individual region token, and thus other factors such as the number of region tokens are also used in this function. The specification of this constraint is an attempt to balance the cost of further segmentation against the cost of region merging to produce the highest quality segmentation at the lowest computational cost. Figure 7 shows the segmentation selected by the region segmentation schema according to this evaluation constraint

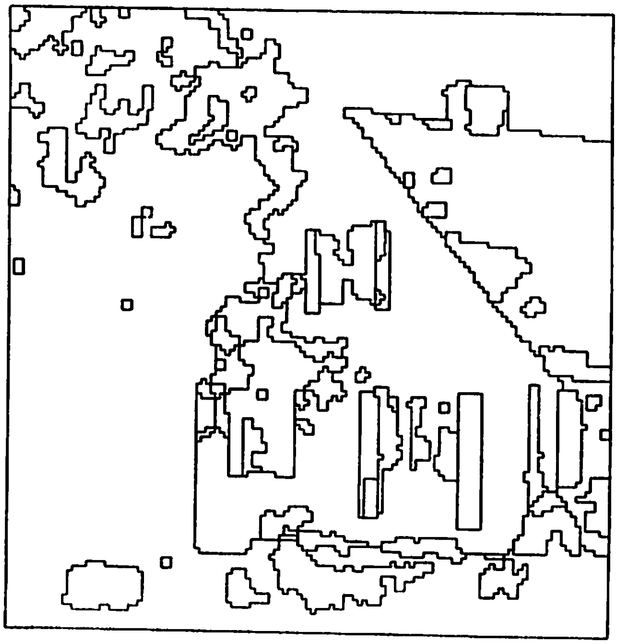

Figure 7: Top-level Segmentation Produced by Initialization Schema

for the image from Figure 1.

Once such a segmentation has been obtained, it is typically the case that despite good overall quality, the segmentation data may still contain many obvious (to an observer) errors which can be corrected at the intermediate level. One such category of error is the possible absence of significant long lines which are obvious in the raw data. Thus the next step in the initialization strategy is to post a line segmentation goal for the insertion of all long, high contrast lines. In this process, which is somewhat similar to the process employed by Nazif ([23]), a set of line tokens are extracted from the raw data by an instance of the line extraction schema and the long, high contrast lines (Figure 8) are used to split any existing region tokens that they intersect (Figure 9).

The execution of this stage of the strategy attempts to assure that all significant discontinuities in the raw image data are represented by the boundaries of region tokens in the segmentation, but fails to assure that all diffuse object boundaries (i.e. those represented by slow spatial changes in feature values) will be represented. Therefore, long line insertion is followed by the posting of a goal for the syntactic evaluation of the region tokens. Through the execution of the syntactic region evaluation schema, it is expected that when a given region token spans such a diffuse boundary, the evaluation rules used by the schema will produce a strong

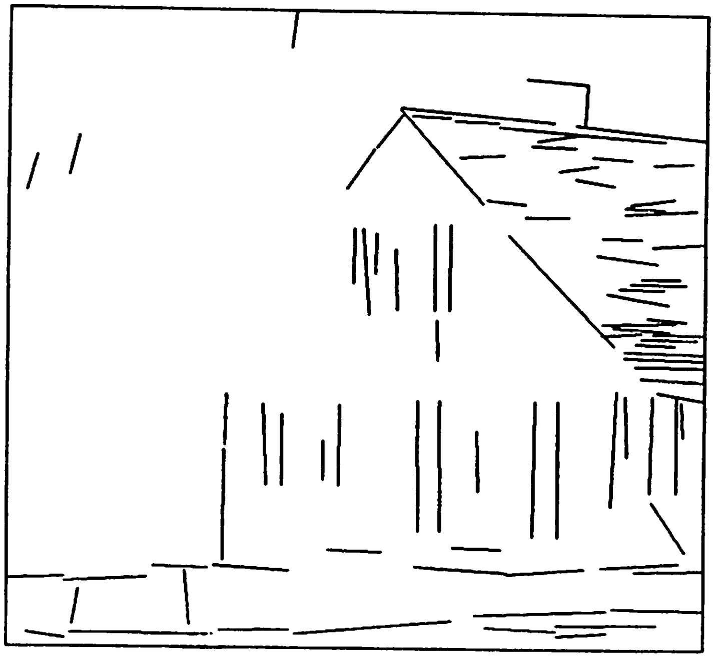

Figure 8: Long High Contrast Lines

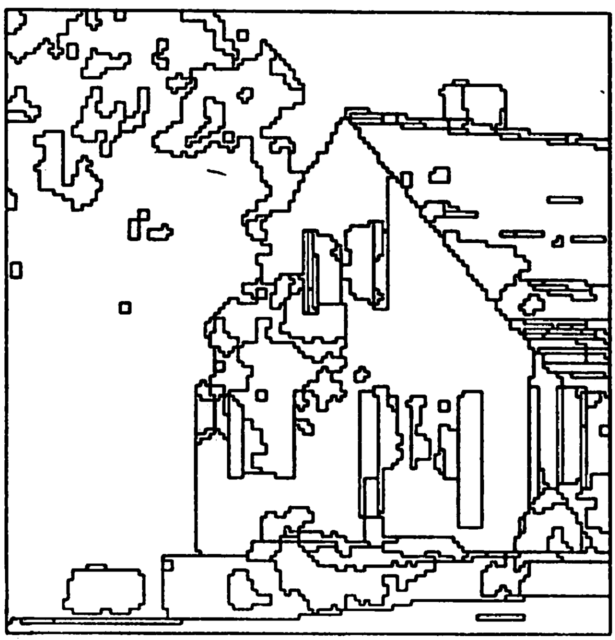

Figure 9: Insertion of Long High Contrast Lines

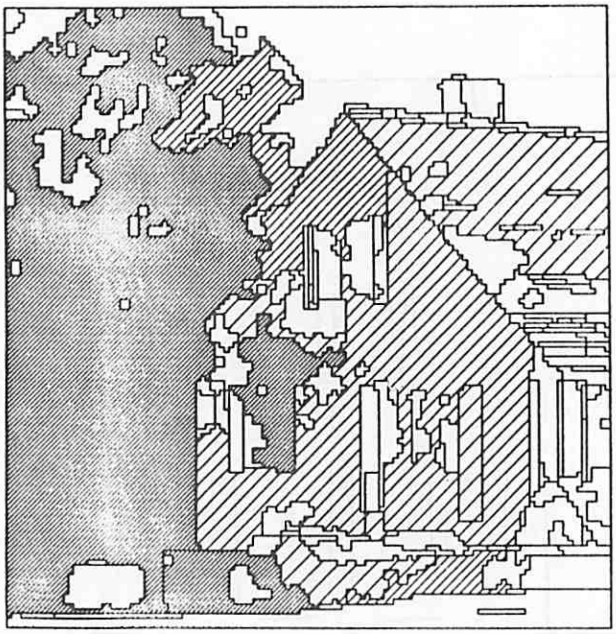

Figure 10: Regions with Strong Resegmentation Hypotheses

resegmentation hypothesis. Figure 10 indicates, with cross-hatching, those regions for which the resegmentation hypotheses are strong enough to indicate a need for resegmentation. The stronger the hypothesis, the more dense the cross-hatching.

Using the goal mechanism of the system to recursively invoke the initialization schema over the set of region tokens for which such a hypothesis exists, the segmentation may be refined in the image areas which contain these diffuse boundaries. Since the image data is progressively more localized as the system proceeds with this recursive invocation, more use may be made of local context to focus on the correct algorithms and features for segmentation. For example, by using semantic hypotheses produced by an instance of the semantic evaluation schema, the instances of the initialization schema are able constrain the instances of the region segmentation schema which they invoke. Thus, if an initialization schema were invoked over a region token which spanned the boundary between bush and tree (as indicated in Figure 11), the constraining knowledge that both semantic objects might be present in the region could help the associated region segmentation schema instance to select features and algorithms which could best discriminate these two objects. Figure 12 demonstrates the nature of this process.

In this case, the overmerged region on the left of Figure 10 is resegmented using the hypotheses for the existence of both bush and tree in the region. Based on

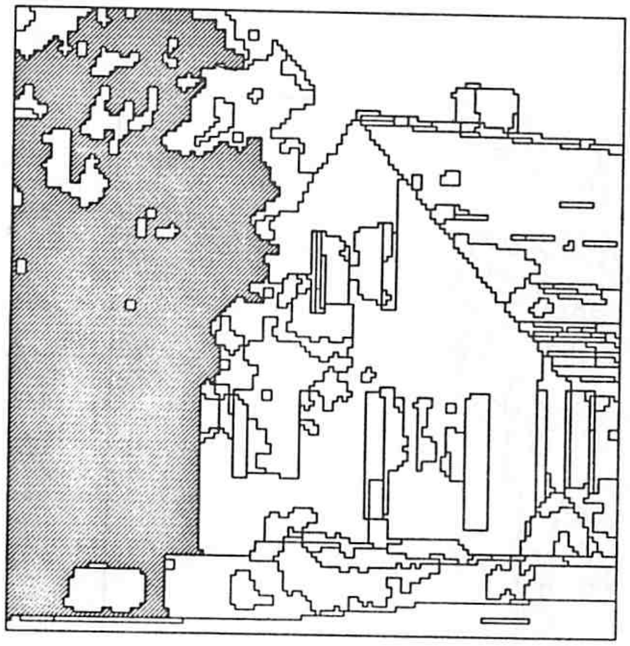

Figure 11: Tree/Bush Region Which Has Strong Resegmentation Hypothesis

this knowledge, the image feature which is selected for the segmentation task is a rotation of RGB space which maximally distinguishes the mean RGB of bush objects from that of tree objects (according to data which has been extracted from a training set and stored in ILTM). With respect to the original image in Figure 1, it can be seen that the segmentation in Figure 12 has distinguished tree from bush and has captured all of the highlight/shadow boundaries in the tree. However, the segmentation is far from ideal in that the two different types of bush in the bottom left have not been distinguished, and also in the fact that many of the tree highlight/shadow boundaries are quite local and semantically unimportant. But since the area defined by the original tree/bush region (Figure 11) is being processed by a complete recursive instance of the initialization schema, the remaining tasks of the initialization strategy, including long, high contrast line insertion, potential recursive invocation of the initialization schema, and region merging (see below) are used to complete the processing on this area. The final result which is returned by this instance of the initialization schema (Figure 13) identifies all major semantic boundaries in the area without a great deal of fragmentation. Note that several of the small regions near the top of the image which appear to be artifacts of fragmentation were not defined as being part of the resegmentation region, and

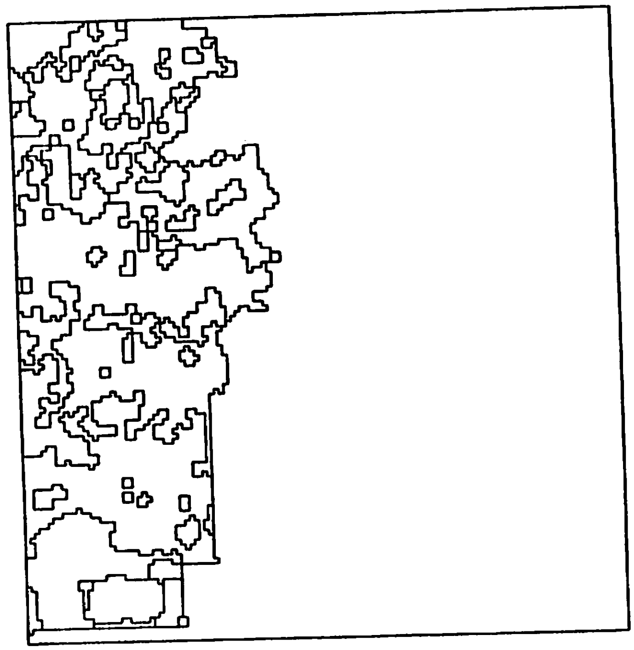

Figure 12: Resegmentation of Tree/Bush Region

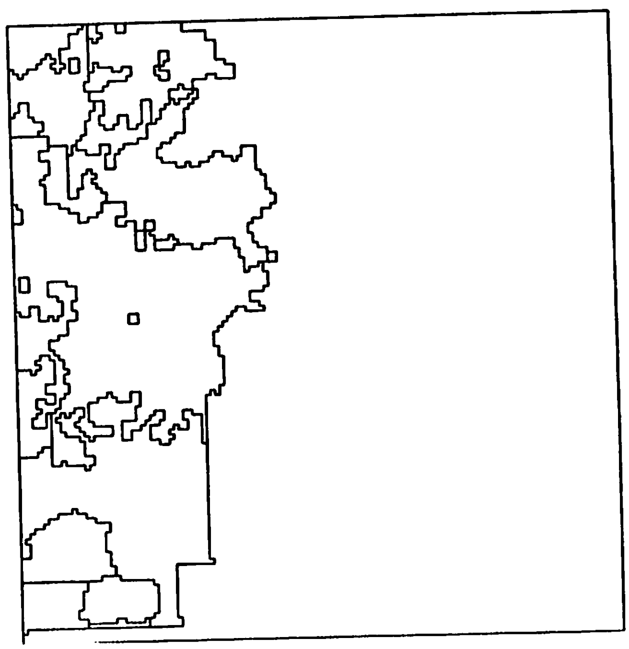

Figure 13: Final Result from Resegmentation of Tree/Bush Region

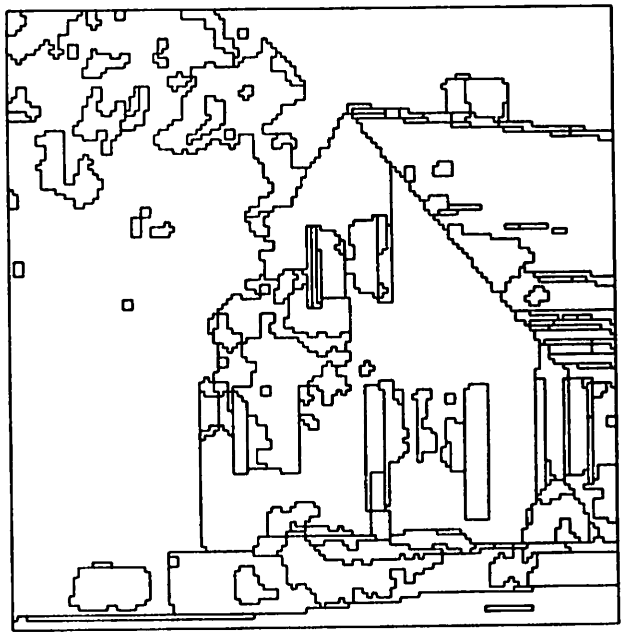

Figure 14: Initialization Schema Segmentation Prior to Final Merge

thus may not be merged by this particular instance of the initialization schema.

At the point that all recursive invocations of the initialization schema have completed, and all region tokens in the ISR have acceptably low resegmentation hypotheses, experience has shown that the updated segmentation typically will still display some degree of overfragmentation (Figure 14). The region merge schema is therefore invoked by the initialization schema to merge all adjacent regions which exhibit reasonable similarity under the constraints specified.

Since the constraints specified in the original instance of the initialization schema are passed down into any goals posted by that schema instance, this entire process occurs within the original context. Thus, even though a complex chain of recursive invocations of the initialization schema has occurred, the top level constraints are observed at all levels of processing.

The final segmentation results produced by the initialization schema (with no *a priori* constraints) on the image from Figure 1 are shown in Figures 15 and 16. This segmentation appears to be qualitatively more useful than either of the two earlier segmentations which were produced by segmentation algorithms that were applied globally and uniformly across the image. The types of errors which do remain (e.g. overfragmentation near the gutter of the house, and overmerging in the windows) are generally the result of ambiguous image data and would require

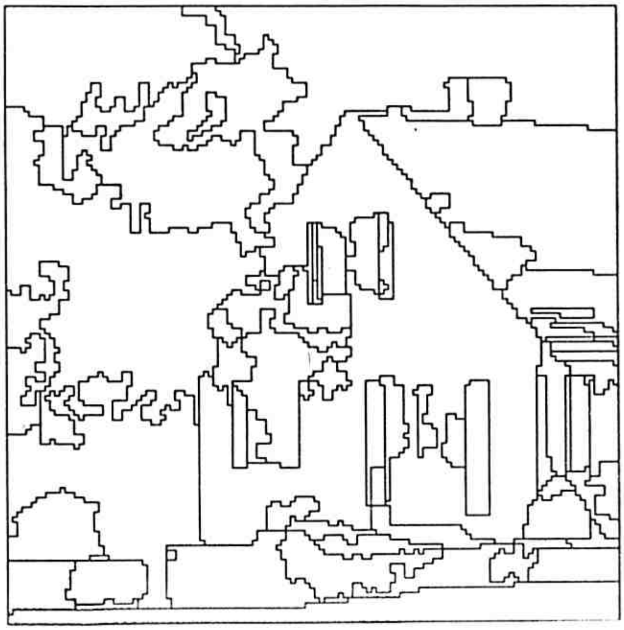

Figure 15: Segmentation from Initialization Schema

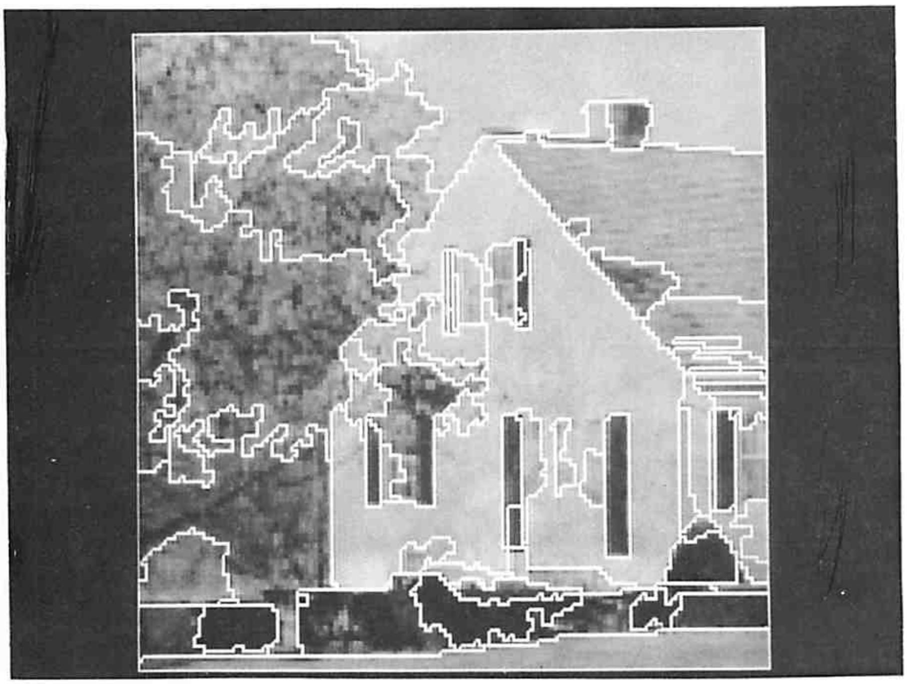

Figure 16: Image with Overlaid Segmentation from Initialization Schema

high-level interpretation processing to resolve the ambiguity or direct the additional reorganization of the intermediate-level tokens.

4.2 The region schemas

The intent of the **region segmentation** schema is to produce a high quality segmentation over the portion of the image specified by a region token. If explicit constraints are known, such as *"concentrate on regions of the image which have the potential for being interpreted as tree"*, the schema is able to use strategies which constrain the segmentation processes such that the tokens produced should meet the expectations of the invoking higher level schema. In the case of this particular goal constraint, the schema would make use of syntactic knowledge about trees, such as the fact that they are present in textured areas of the image or that they exhibit certain color characteristics, to select the low-level segmentation task which would best discriminate tree from all other objects present in the image.

The region segmentation schema is designed to work according to a hypothesize-and-test paradigm. The schema hypothesizes appropriate processing parameters such as image features, segmentation algorithms, and sensitivity settings, and then uses these parameters to control the application of specific low-level image processing tasks through the low-level process controller. The test portion of the hypothesize-and-test paradigm is then implemented by the use of the evaluation constraint associated with the goal. If the test fails, the schema instance continues to hypothesize new features, algorithms, and settings until the test succeeds. If no segmentation is found which meets the evaluation constraint, the segmentation which came closest to meeting the constraint is returned. Figures 17 and 18 show two segmentations for the tree/grass region (Figure 11) which were rejected according to this mechanism prior to the acceptance of the segmentation which was shown in Figure 12. With respect to the evaluation constraint described above, the segmentation in Figure 17 received a low score due to a need for significant resegmentation, and the low evaluation score for the segmentation in Figure 18 was due to overfragmentation. Note, however, that given a different evaluation constraint, either of these segmentations could have been chosen over that of Figure 12.

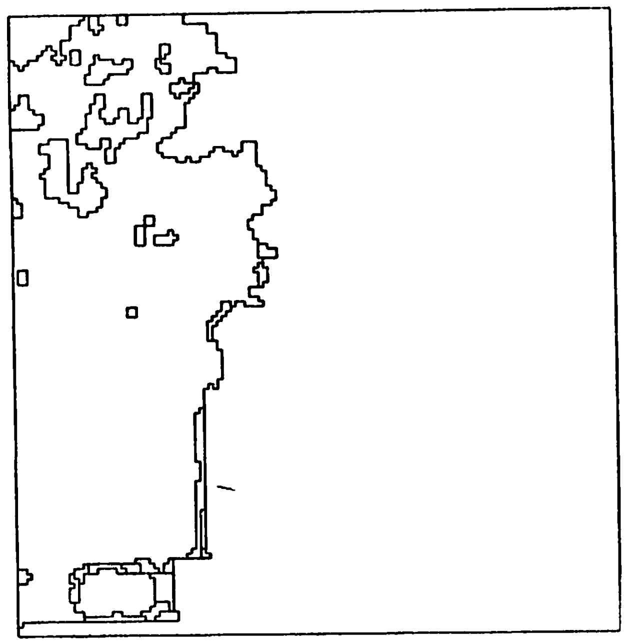

Figure 17: Region Segmentation Which Was Rejected By Region Segmentation Schema

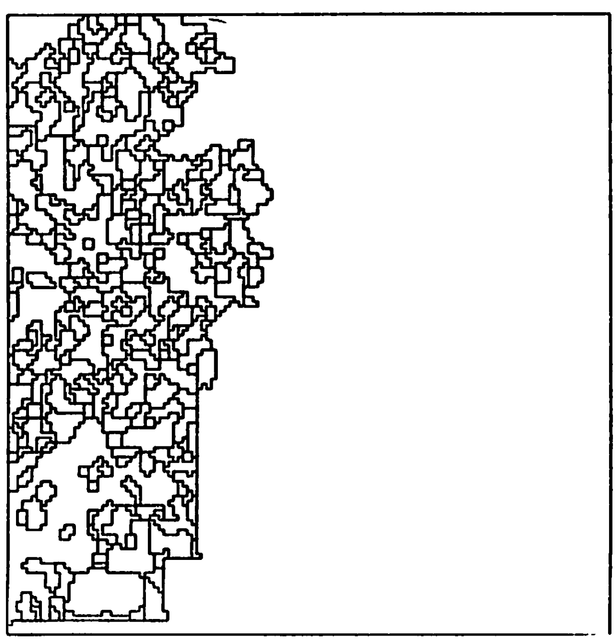

Figure 18: Region Segmentation Which Was Rejected By Region Segmentation Schema

Instances of the region segmentation schema make use of region-feature and region–algorithm schemas to establish hypotheses for the parameters involved in the specification of low–level segmentation tasks. Within the **region–feature** schema, initial hypotheses for the selection of an appropriate image feature are made on the basis of the specified constraints. Thus, if the constraints for an instance of this schema specified an interest in tree and sky, the schema would use the knowledge in the ILTM to hypothesize a set of features which would best discriminate regions representing these two objects. The region–feature schema instance would then refine these initial hypotheses by evaluating the actual frequency distribution of each of these features over the area of interest. Based on this evaluation, the hypotheses would be ranked and returned to the invoking schema instance in order of hypothesized utility. If none of this initial set of hypothesized image features were acceptable to the invoking schema, additional less restrictive strategies would then be employed by the region-feature schema instance to propose additional segmentation features.

If, however, no constraints were specified on the goal, the schema instance initially hypothesizes a set of default features which are known to typically produce good results across the particular image domain. It would be this set of features which was evaluated and returned in order of hypothesized utility. Thus we see that, as is the case with most of the schemas in this system, the more specific the information provided on the goal for the region segmentation schema instance, the more specific the hypotheses.

The GOLDIE system currently utilizes a set of 43 image features such as red, green, blue, intensity, local deviation of intensity, hue, etc., all of which are computed from the original RGB. Given the mechanisms by which these features are managed, we believe that it would be a straightforward extension to include other types of spatially distributed data such as infrared or range data.

The **region–algorithm** schema functions in much the same way as the region-feature schema, but in this case the schema instance does not actually initiate any low level processes. This schema contains a set of strategies for proposing segmentation algorithms and sensitivity settings which are based on the nature of the particular image feature and the other constraints. As currently implemented, this schema is able to select between algorithms for global one-dimensional histogram clustering, localized one–dimensional histogram clustering, zero-crossings, thresholding, and global two–dimensional histogram clustering, each of which can be used with a variety of sensitivity settings. Given a particular image feature, or set of image features, an instance of this schema is able to hypothesize the algorithm and sensitivity setting which is most likely to produce region tokens with the desired characteristics.

The schema first proposes the most specific algorithm that it can hypothesize under the constraints, and if this is not accepted, it can then propose progressively more general algorithms. For example, if an instance of this schema were invoked with the constraints that the region–feature was Deviation (a textural feature which tends to smooth over object boundaries) and the object label of interest was tree, the schema instance would first propose the thresholding algorithm with low sensitivity. This algorithm would be expected to distinguish high–texture areas (tree) from low–texture regions without introducing excessive fragmentation in either area. Additionally, the thresholding algorithm would be expected to place the boundary between region tokens at the midpoint of the image feature gradient of the boundary, hopefully restoring the blurred object boundary. If this algorithm was found to be unacceptable, the schema instance would then propose a low sensitivity zero–crossing algorithm. If this were also rejected, a final hypothesis of low-sensitivity one-dimensional histogram clustering would be proposed.

Once the region segmentation schema has produced an acceptable set of region tokens, other region schemas may be used to evaluate or modify these tokens. The **syntactic region evaluation** schema is concerned with the creation of ISTM hypotheses about region tokens which indicate a belief in the "goodness" of a particular region structure (i.e. the degree to which the existence of the region token is supported by the underlying image feature data). Based on assumptions expressed in goal constraints, such as *"we are interested in regions which exhibit slow intensity gradients"*, the schema selects various subsets of the evaluation functions which are stored in ILTM in order to determine whether or not the region token should be merged or resegmented. The results of these evaluation functions are combined, and stored in ISTM as hypotheses about the region tokens. The hypothesis values may then be accessed by other schemas of the system which will take the appropriate actions.

The schema for the **semantic region evaluation** utilizes sets of rules which evaluate the characteristics of regions, such as short line density, color, etc. to assign a plausible semantic label to a region (e.g. sky or tree). These rules, which are stored in the ILTM, are not as precise or complete as those used by a complete interpretation system ([2,30]), but provide valuable information for the intermediate-level schemas in the case that no overriding hypotheses have been specified as goal constraints.

The **region merge** schema is used by the system to reduce potential overfragmentation. By using goal constraints to select a subset of the evaluation functions stored in ILTM, instances of this schema are able to evaluate the desirability of a merge of a set of adjacent regions, and, if necessary, direct the actual merge process. The schema may either be applied in a general fashion over the entire image, evaluating all region tokens in the ISR for merge potential, or it may be invoked to directly merge a set of adjacent region tokens which some high-level process has deemed to be similar. Again using our example of a system with the *a priori* constraint to discriminate tree and sky, an instance of this schema would select two sets of merge criteria. In textured areas of the image, regions would be merged if they were adjacent, at least one of the tokens had a strong hypothesis for merge potential, and they both exhibited similar textural and hue characteristics. In non-textured areas of the image, the merge criteria would still make use of adjacency and merge hypothesis, but the regions would also have to demonstrate similar low values of intensity deviation as well as demonstrating co planar intensity surface fits.

4.3 The line schemas

The **line segmentation** schema is designed to either directly insert lines into a region representation and thereby redefine the region mapping, or to control a region segmentation process which is designed to produce region boundaries which show a high degree of overlap with a set of specified lines. The former option is utilized in the case that there is strong belief in the existence of the lines, and when the specifications of the lines are known with a high degree of accuracy. Thus, given a set of lines which had been produced through the line extraction schema, and were therefore known to be present in the raw image data, the lines could be directly inserted into the region representation. A different rationale for this process would be a situation in which the position of a mobile robot was known with respect to a road, but the boundaries of the road were not immediately apparent in the robot's image data. In this case, the road boundaries could be determined from an internal map and then be placed directly into the segmentation data to aid in the interpretation of the remainder of the image.

This type of behavior by the schema was demonstrated in Figure 9, where the insertion of long lines restored several important boundaries which had been missed in the original region segmentation from Figure 7. Since the lines are used to split all regions with which they have significant intersection, several artificial boundaries have also were introduced by this process. However, as was shown in the final result (Figure 15), the region merging process was able to remove most boundaries which did not have a basis in the underlying data.

If a high-level interpretation schema concerned with shape were to make a hypothesis that a line might possibly exist in an area of the image, this type of direct insertion would not be desirable (i.e. the hypothesis might be incorrect). In this case, the characteristics of image feature distributions across the hypothesized line would be examined in an attempt to find a feature which could be used by a region segmentation process to produce a region mapping in which the line was matched by region boundaries.

The line tokens which are utilized by these schemas are either created as hypothesized lines by an interpretation schema or are extracted from the data by an instance of the **line extraction** schema, using one of several different straight line extraction algorithms ([4,35]). The schemas for **line–algorithm** and **line–feature** produce the necessary hypotheses for an instance of the line extraction schema, and these two schemas function in a fashion similar to their analogues for region segmentation. Figure 19 shows the set of lines produced by the line extraction schema,

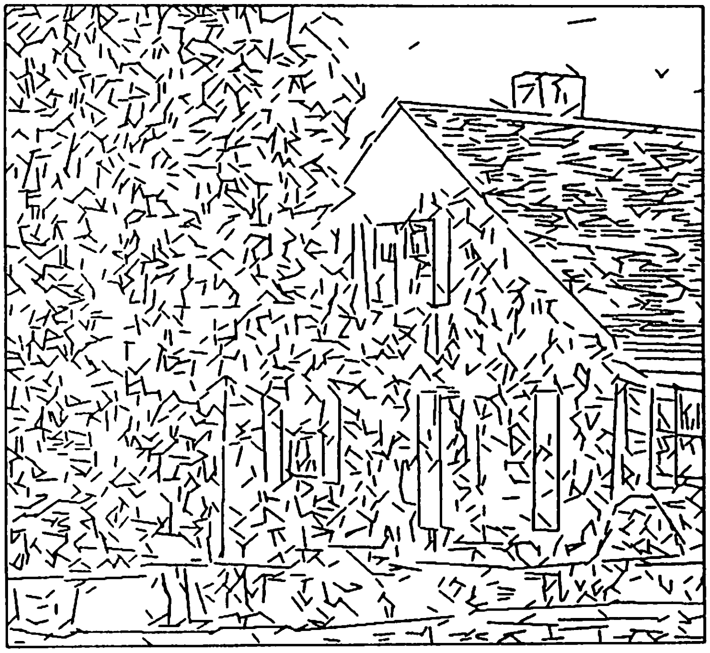

Figure 19: Lines Extracted by Line Extraction Schema

from which the set of long high contrast lines in Figure 8 were selected.

The low-level extraction processes controlled by instances of this schema may be constrained through goal constraints to restrict the output according to location, line orientation, line length, or contrast across the line. Thus a high-level interpretation process dealing with object shape which needed to find a particular line to complete a rectangle could post a goal to find a straight line in the image data which is similar to the desired line. On the other hand, a schema process which was interested in the short line density over an area of the image would invoke this schema with the constraint to find all possible lines. The set of line tokens would then be filtered on length, and perhaps contrast, to compute the density measure.

The **collinear line merge** schema is utilized in the case that a desired line has not been found intact by the line extraction processes. The low-level process controlled by this schema examines the set of existing line tokens which have been produced by the line extraction process to determine if suitable line tokens may be merged to produce the desired line ([35]). Figure 20 shows the long lines created by the application of the collinear line merge schema to the lines of Figure 8.

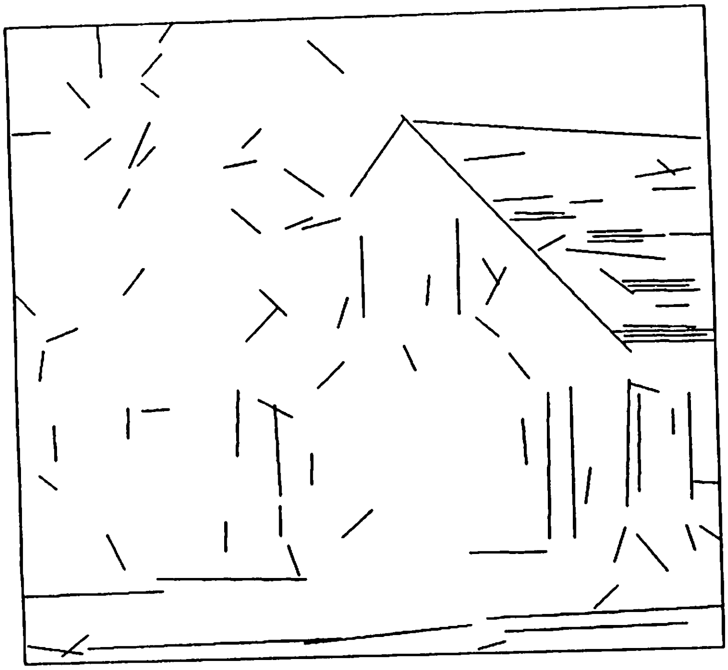

Figure 20: Collinear Line Grouping

5. CONCLUSION

The set of schemas we have described is currently implemented within the GOLDIE system at the University of Massachusetts, and research is proceeding on the interface between this system and the high-level interpretation schemas being developed by other researchers in the VISIONS group. As currently implemented, the GOLDIE system provides an incomplete, yet powerful mechanism for the control of low-level image processing. Future development of the system to add schemas for additional low-level processing tasks, and to improve the computational efficiency, will produce a system which dramatically affects the overall image interpretation process.

There are four major aspects of GOLDIE which contribute to the utility of the system. Foremost is the fact that the system is goal-driven. This paradigm provides a coherent mechanism for top-down control in which low-level processing may be tuned to meet the expectations of higher level processes which are expressed through goal constraints. Second, the system makes use of a unified data representation which allows processes at any level of the interpretation hierarchy to create, access, or modify the data which represents the current state of interpretation processing. The third aspect is that the system contains an explicit representation of knowledge about low-level processes and the image domain which provides a flexibility permitting extension to additional processes or image domains. Finally, the GOLDIE system is capable of operating within a hypothesize–and–test protocol, exploring a variety of potential solutions to high–level requests rather than operating in a strictly deterministic manner.

References

[1] Dana H. Ballard and Christopher M. Brown. *Computer Vision*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1982.

[2] R. Belknap, E. Riseman, and A. Hanson. The information fusion problem and rule-based hypotheses applied to complex aggregations of image events. In *Proceedings of IEEE–CVPR Conference*, pages 227–234, June 1986.

[3] Rodney A. Brooks and Thomas O. Binford. Interpretive vision and restriction graphs. In *Proceedings of The First Annual National Conference on Artificial Intelligence*, pages 21–27, August 1980.

[4] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. In *Proceedings Seventh International Conference on Pattern Recognition*, pages 482–485, Montreal, July 1984.

[5] John F. Canny. *Finding Edges and Lines in Images*. Technical Report 720, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1983.

[6] B. Draper, R. Collins, J. Brolio, J. Griffith, A. Hanson, and E. Riseman. Tools and experiments in knowledge-based interpretation of road scenes. In *Proceedings: Image Understanding Workshop*, 1987. To be published.

[7] B. Draper, A. Hanson, and E. Riseman. *A Software Environment for High Level Vision*. Technical Report, Computer and Information Science Department, University of Massachusetts at Amherst, 1986. In preparation.

[8] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, NY, 1973.

[9] Joey Griffith, Allen Hanson, and Edward Riseman. A rule based region merging system. In Preparation.

[10] Allen R. Hanson and Edward M. Riseman, editors. *Computer Vision Systems*. Academic Press Inc., New York, NY, 1978.

[11] Allen R. Hanson and Edward M. Riseman. VISIONS image understanding system - 1986. In Christopher M. Brown, editor, *Advances in Computer Vision*, Erlbaum Assoc., 1987. To be published.

[12] Allen R. Hanson, Edward M. Riseman, and Paul R. Nagin. *Region-growing in Textured Outdoor Scenes.* Technical Report 75C-3, Computer and Information Science University of Massachusetts at Amherst, Amherst Massachusetts 01003, 1982.

[13] Robert M. Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Trans on Pattern Analysis and Machine Intelligence*, PAMI-6(1):58–68, January 1984.

[14] Robert M. Haralick and Linda G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–132, January 1985.

[15] Vincent Shang-Shouq Hwang, Larry S. Davis, and Takashi Matsuyama. *Hypothesis Integration in Image Understanding Systems.* Technical Report TR-1137, Center for Automation Research, University of Maryland, College Park, Maryland 20742, June 1985.

[16] Charles A. Kohl. *Goal Directed Image Segmentation.* PhD thesis, Computer and Information Science Department, University of Massachusetts, Amherst, Massachusetts 01003, 1987. In Preparation.

[17] R. R. Kohler and A. R. Hanson. The VISIONS image operating system. In *Proceedings Sixth International Conference on Pattern Recognition*, pages 71–74, Munich, October 1982.

[18] Ralf Kohler. A segmentation system based on thresholding. *Computer Graphics and Image Processing*, (15):319–338, 1981.

[19] Martin D. Levine and Samir I. Shaheen. A modular computer vision system for picture segmentation and interpretation. *Pattern Analysis and Machine Intelligence*, PAMI-3(5):540–554, September 1981.

[20] David Marr. *Vision.* W. H. Freeman, San Francisco, CA, 1982.

[21] M. Nagao and T. Matsuyama. *A Structural Analysis of Complex Aerial Photographs.* Plenum, New York, NY, 1980.

[22] Paul A. Nagin. *Studies in Image Segmentation Algorithms Based on Histogram Clustering and Relaxation.* Technical Report 79-15, Computer and Information Science University of Massachusetts at Amherst, Amherst Massachusetts 01003, 1979.

[23] Ahmed M. Nazif. *A Rule-Based Expert System for Image Segmentation.* PhD thesis, Electrical Engineering Department, McGill University, Montreal, Canada, March 1983.

[24] Ronald B. Ohlander. *Analysis of Natural Scenes.* PhD thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213, April 1975.

[25] Yu-ichi Ohta. *Knowledge-based Interpretation of Outdoor Natural Color Scenes.* Pitman Advanced Publishing Program, Boston, MA, 1985.

[26] Yu-ichi Ohta. *A Region-Oriented Image-Analysis System by Computer.* PhD thesis, Kyoto University Department of Computer Science, 1980.

[27] Cesare C. Parma, Allen R. Hanson, and Edward M. Riseman. *Experiments in Schema-Driven Interpretation of A Natural Scene.* Technical Report 80-10, Computer and Information Science Department - University of Massachusetts at Amherst, Amherst, Massachusetts 01003, April 198.

[28] George Reynolds and J. Ross Beveridge. Searching for geometric structure in images of natural scenes. In *Proceedings: Image Understanding Workshop*, 1987. To be published.

[29] George Reynolds, Nancy Irwin, Allen Hanson, and Edward Riseman. Hierarchical knowledge-directed object extraction using a combined region and line representation. In *IEEE 1984 Proceedings Of The Workshop On Computer Vision Representation And Control*, pages 238–247, IEEE, 1984.

[30] Edward M. Riseman and Allen R. Hanson. *A Methodology for the Development of General Knowledge-Based Vision Systems.* Technical Report 86-27, Computer and Information Science University of Massachusetts at Amherst, Amherst Massachusetts 01003, 1986.

[31] A. Rosenfeld and A. Kak. *Digital Picture Processing.* Academic Press, New York, NY, 1976.

[32] Bert Shaw. *Some Remarks on the Use of Color for Machine Vision.* Technical Report 83-31, Computer and Information Science Department - University of Massachusetts at Amherst, Amherst, Massachusetts 01003, September 1983.

[33] J. M. Tenenbaum and H. G. Barrow. *Experiments in Interpretation-Guided Segmentation.* Technical Report 123, SRI International, 333 Ravenswood Ave., Melno Park, CA 94025, March 1976.

[34] William B. Thompson and Albert Yonas. What should be computed in low level vision. In *Proceedings of The First Annual National Conference on Artificial Intelligence*, pages 7–10, August 1980.

[35] Richard Weiss and Michael Boldt. Geometric grouping applied to straight lines. In *Proceedings of IEEE-CVPR Conference*, pages 489–495, June 1986.

[36] Terry E. Weymouth. *Using Object Descriptions in a Schema Network for Machine Vision.* PhD thesis, University of Massachusetts, Amherst, Massachusetts 01003, 1986.

[37] Y. Yakimovsky. *Scene Analysis Using a Semantic Base for Region Growing.* PhD thesis, Stanford University, Palo Alto, CA, 1973.