# SUMMARY OF PROGRESS IN IMAGE UNDERSTANDING AT THE UNIVERSITY OF MASSACHUSETTS

Edward M. Riseman and Allen R. Hanson

COINS Technical Report 85-48

November 1985

## ABSTRACT

This research summary documents recent activity across a range of computer vision research at the University of Massachusetts. The work is divided into several areas: motion analysis, knowledge-based processing, low-level and intermediate-level processing, and parallel vision architectures.

This research was supported by the following Grants: Defense Advanced Research Projects Agency under contract number N00014-82-K-0464, Army under contract number DACA76-85-C-0008, National Science Foundation under contract number DCR-8318776, National Science Foundation under contract number DCR-8500332, Defense Mapping Agency under contract number 800-85-C-0012, and Air Force Office of Scientific Research under contract number 86-0021.

## TABLE OF CONTENTS

# I. MOTION ANALYSIS

Our research in motion analysis continues to broaden with research in several theoretical and experimental areas.

## I.1. EFFECTIVENESS IN RECOVERING TRANSLATIONAL MOTION PARAMETERS

We have continued the analysis of algorithms for constrained sensor motion [LAW84]. In particular we are evaluating the robustness, accuracy, and efficiency of the algorithm for recovering translational motion parameters [PAV85]. Here the global search for the focus-of-expansion (FOE) requires the computation of the sum of errors (e.g., via correlation) associated with the displacement of a set of feature points in two or more frames. A sparse sampling of the possible location of the FOE provides a global error function whose minimum localizes the FOE, and thus the direction of motion.

The accuracy and robustness of the algorithm is a function of the number of points that are matched for contributions to the error function, which of course must be traded off against the amount of computation that can be tolerated for real-time motion analysis. Thus far, our experiments on simulated environments imply that there is a wide range of situations for which the motion parameters can be approximately recovered at relatively modest computational expense. Specifically, when the angle between the image plane and the direction of translational motion is less than 60 degrees, then between 4 and 16 points which are widely spaced in the image are sufficient to recover the approximate motion of the sensor. A smaller number of points (4–8 points) is necessary when the camera is oriented approximately in the direction of motion, (0-15 degrees) and a larger number of

points (8–16) when the camera orientation is at a modest angle (15-45 degrees) with respect to translation. When the angle between camera orientation and translation is large (60-90 degrees) there appears to be a flat error surface around the correct direction of motion, leaving a wide range of ambiguity no matter how many feature points are employed. This result is not surprising in that it states, for example, that when a camera is pointing out the driver's side window, accurate determination of the motion of a vehicle moving down the road is not possible.

Similar analyses for other cases of constrained sensor motion, including pure rotation, and planar motion in a known plane, remain for future work. We believe that they will exhibit similar levels of robustness and computational requirements.

## I.2. INHERENT AMBIGUITY IN MOTION ANALYSIS OF NOISY FLOW FIELDS

In the cases where the sensor motion is unconstrained and/or there are independently moving objects in the environment, the algorithms for direct recovery of motion parameters and environmental structure are not applicable. Therefore, we turn to the usual method of motion analysis which is decomposed into two phases: computation of an optical flow field and interpretation of this field. In the present discussion, the term "optical flow field" refers to a "velocity field", composed of vectors describing the instantaneous velocity of image elements. The computation of reliable flow fields is the subject of work presented in Section I.3. The second phase, which is the general interpretation of flow fields, was the subject of previous work by Adiv [ADI85a,b].

The work discussed in this section mathematically examines the robustness of algorithms for interpreting general motion from flow fields. The analysis focusses on ambigu-

ities that are inherent in the sense that they are true of all algorithms, and can only be resolved if constraining assumptions or other sources of visual information are employed.

Two problems which may arise due to the presence of noise in the flow field have been examined. Since noise in flow fields must be expected almost always to be present, we believe this analysis is relevant to all real situations of motion interpretation.

The first ambiguity is in recovering the motion parameters from a noisy flow field generated by a rigid motion. Motion parameters of the sensor or a rigidly moving object may be extremely difficult to estimate because there may exist a large set of significantly incorrect solutions which induce flow fields similar to the correct one. We found that if the field of view corresponding to the region containing the interpreted flow field is small, and the depth variation and translation magnitude are small relative to the distance of the object from the camera, then the determination of the 3-D motion and structure can be expected to be very sensitive to noise and, in the presence of a realistic level of noise, practically impossible. We experimentally found that there is also a relation between the location of the FOE and the degree of ambiguity.

The second ambiguity is in the decomposition of the flow field into sets of vectors corresponding to indepedently moving objects. The rigidity assumption [ULL79] has been found to be inappropriate for noisy flow fields; that is, the consistency of a set of flow vectors with the same motion parameters, up to the estimated noise level, does not reasonably guarantee that they are really induced by one rigid motion. Two independently moving objects may induce optical flows which are compatible with the same motion parameters and hence, there is no way to refute the hypothesis that these flows are generated by one rigid object. As an alternative to the usual rigidity assumption, it is assumed in

[ADI85a,b] that a connected set of flow vectors, which is consistent with a rigid motion of a *planar* surface, is induced by a rigid motion. This assumption is weaker than the standard assumption in the sense that it can only be applied in more restricted situations and, therefore, it is more likely to be correct.

The results of the ambiguity analysis can be used when the effectiveness of motion algorithms is evaluated for real-world tasks. They can help to decide which algorithm to choose, and in what situations this algorithm can be expected to be effective. Recovering motion and structure of independently moving objects may be particularly difficult, as was demonstrated by the flat error surfaces obtained for such objects in the second and fifth experiments in [ADI85b]. In general, ambiguity in recovering 3-D motion and structure of independently moving objects can be expected, since the effective field of view and the ratio of the depth variation to the distance between the object and the camera are usually small. Even in ambiguous situations, constraints and parameters might be extracted. Integration of such partial information over a time sequence of flow fields may, eventually, resolve the ambiguity and result in a unique interpretation.

## I.3. RELIABLE COMPUTATION OF OPTIC FLOW: A SMOOTHNESS CONSTRAINT AND A CONFIDENCE MEASURE

Although our hierarchical correlation algorithm [GLA83] for the computation of dense displacement fields has proved to be an efficient and reliable technique, there are still a number of situations where the algorithm makes mistakes. These situations arise in areas of images without significant intensity variations and at occlusion or motion boundaries. Our previous work [ANA84] attempted to identify such situations through the use of a confidence measure which indicated the reliability of a match vector. Our current work

attempts to improve matches with low confidence based on neighbouring matches with higher confidences, by means of a relaxation process.

The confidence measure that was described in [ANA84] is a scalar value between 0 and 1 that indicated the reliability of the displacement vector at a pixel in the image. One such value was provided for each pixel. This measure was derived by studying the properties of the error-surface obtained during the process of computing the displacement at a pixel. However, the image displacement vector is a two-dimensional quantity. Hence, it is appropriate to have a two-dimensional confidence measure associated with the displacement vector.

In our previous work [ANA84], we observed that the error-surface allowed us to distinguish between situations where we had completely reliable information information regarding the displacement vector (i.e., at high curvature points along image contours), where we had partial information (i.e., at edge locations where only the displacement perpendicular to the edge can be reliably measured), and situations where we had no reliable information (at homogeneous intensity areas of the image). The new confidence measure is a vector quantity which uses these distinctions.

Our current work consists of two steps. The first is the computation of these vector-valued confidence measures and the second is the smoothing process which corrects unreliable displacement vectors based on their reliable neighbours.

1. The new confidence measure is best described as a two-dimensional vector. It is convenient to describe the vector in terms of two orthogonal basis vectors $e_{max}$ and $e_{min}$, which vary from pixel to pixel in an image. The displacement vector $D$ can be decomposed in terms of its components along these basis vectors and confidence

measures $c_{max}$ and $c_{min}$ are associated with these components. The basis vectors and the confidence measures can be easily understood by their behaviour at a high curvature point, an edge point and a point in a homogeneous area of the image.

At a high-curvature point both $c_{max}$ and $c_{min}$ will be high, indicating that all the components of the displacement vector are highly reliable. In this case the exact directions of $e_{max}$ and $e_{min}$ are not crucial, and will depend on the precise shape of the contour. At an edge point $c_{max}$ will be high and $c_{min}$ low, and $e_{max}$ and $e_{min}$ will respectively be perpendicular and parallel to the edge. At a homogeneous area both the confidences will be low, and the directions of the basis vectors will depend on the details of the image intensity variations at that point.

Finally, the new confidence measures are also based on the shape of the correlation error surface. The details of their computation are described in [ANA85]. It is worthwhile to note that these are no longer bound to the range between 0 and 1. The formulation of the smoothness constraint described below requires that these values be allowed to vary between 0 and $\infty$.

2. The process of improving unreliable match estimates based on its neighbours is formulated as a smoothness constraint on the displacement vector field. The smoothness constraint consists of two errors $E_{smooth}$ and $E_{approx}$, whose sum is minimized.

$E_{smooth}$ measures the spatial variation of the displacement field - i.e., the smoother the variation, the smaller is the error. One example of such a constraint can be found in the work of Horn and Schunck [HOR81]. $E_{approx}$ measures the deviation of the smooth

displacement field from the initial field provided by the matching process.

$$E_{approx} = \sum c_{max}((U - D) \cdot e_{max})^2 + c_{min}((U - D) \cdot e_{min})^2$$

where U is the smoothed displacement vector and D is the initial vector at a pixel provided by the matching process. The definition of this error makes it clear that the low confidence estimates are allowed to vary more than the high confidence estimates. Hence, the smoothing process modifies the initial displacement values at locations of low confidence measures more than those at the locations of high confidence measures.

The smoothness constraint translates into a minimization problem. We solve this problem using the finite-element method, because this method permits the inclusion of known discontinuities in the displacement field. The application of this method leads to a local relaxation algorithm, which iteratively updates the displacement vector field.

Our future work will consist of developing techniques for locating the displacement discontinuities, gaining a greater understanding of the confidence measures (in particular how to normalize them), and possible improvements to the smoothness error.

## I.4. REFINEMENT AND PREDICTION OF IMAGE DYNAMICS AND ENVIRONMENTAL DEPTH MAPS OVER MULTIPLE FRAMES

To a large extent research in the interpretation of motion has focussed on the recovery of the motion parameters of a sensor moving through a static environment, and more generally the relative motion between a sensor and a visible object. Under ideal conditions, once these motion parameters are known, a depth map can be recovered from two frames if the displacement (flow) field is exact.

In previous sections of this review, we have discussed various reasons why displacement fields are not perfect. Even with perfect information about sensor motion, displacement vectors from translational motion are a function of the depth of the surface element. Any ambiguity or error in displacements along linear paths emanating radially from the FOE leads to ambiguity in the depth of that surface element. There are several sources of such ambiguity including multiple minima in the matching process for computing displacements, noise affecting the match location, and finally the resolution in the matching process along that radial path. Consequently, we are viewing the matching process as a dynamic refinement of depth over multiple frames.

The work that we discuss here is a first step in the exploration of several issues involved in the stability, refinement, and prediction of depth maps over multiple frames [BHA85]. We are considering the differences in start-up (when no depth information exists) versus updating an existing (and possibly inaccurate) depth map; in both situations we assume limited computational resources are available, yet increasing accuracy over time is required.

When an image sequence is first acquired, or the visible field changes dramatically (as in the case of coming around a corner), no depth map exists and the situation can be considered as a start-up. Under an assumption of a fixed limit on the computation that can be carried out between any pair of frames, a strategy has been developed to extract a coarse depth approximation from the first pair of frames using a coarse spatial resolution for the matching process. Each subsequent frame that is processed can use the previous estimate of depth to narrow the match area while increasing the match resolution, thereby maintaining constant computation, but finer accuracy in the depth estimates. As this process continues, temporal resolution can also be reduced as necessary. Thus, the approach employed involves a combined hierarchical spatial and temporal resolution as

frames continue to arrive.

The refinement strategy that we have just described for the start-up phase of depth map recovery can be generalized for updating, prediction, and error analysis. Under known sensor motion and known environmental depth, the image location and appearance of environmental features can be accurately predicted and matched from one frame to the next (leaving aside complex issues of image changes due to changes in lighting, highlights, shadows, shape distortion of surface patches, or occlusion). Thus, when one reaches the desired level (or limit) of spatial and temporal resolution, the updating process becomes one of prediction and verification of the environmental model. When predictions are not accurate, then depending upon the representation, the depth of either pixels, points, lines, regions, or surfaces could be refined in a focus-of-attention and refinement process for error reduction. Areas of the image and environment that do not behave as predicted become the focus of processing until their image dynamics over time can be properly predicted. In this manner one has an ongoing mechanism for verification of the current interpretation of the environment.

# II. IMAGE INTERPRETATION

Work on the VISIONS system for interpretation of static images continues [HAN78, PAR80, RIS84]. A rule-based system for generating initial object hypotheses from image data has been extended to permit information from multiple sources of low level data to be "fused" in a consistent manner. On the basis of the results in a forthcoming thesis by Weymouth [WEY86], we have refined the notion of schemas as a representation of knowledge. We are implementing a new schema system in Common LISP and translating existing schemas and their associated interpretation strategies into the new format. We are continuing to explore inferencing mechanisms based on the Shafer-Dempster-Lowrance idea of evidential reasoning [SHA76, DEM68, LOW82, WES83, WES85]. A recent development is a method for generating mass functions using explicit knowledge about the image domain without requiring that the range of values over which the mass functions are defined be either explicitly or implicitly discretized into "propositions".

## II.1. RULE-BASED HYPOTHESES FROM COMPLEX AGGREGATIONS OF IMAGE EVENTS.

In a recent paper [WEY83, RIS84, ARB86] we described a simple type of knowledge source for generating object hypotheses for particular regions in the image. Simple rules are defined in terms of ranges over a scalar feature, and complex rules are defined as combinations of the output of a set of simple rules. The scores of these rules serve as a focus of attention mechanism for other, more complex knowledge-based processes. The rules can also be viewed as sets of partially relundant features each of which defines an area of feature space which represents a "vote" for an object on the basis of this single feature value. The region attributes include color, texture, shape, size, image location,

and relative location to other objects. More recently, the approach has been extended to lines, with features including length, orientation, contrast, width, etc. In many cases, it is possible to define rules which provide evidence for and against the semantically relevant concepts representing the domain knowledge. While no single rule is totally reliable, the combined evidence from many such rules should imply the correct interpretation.

Most of the rules previously described are unary, accepting a region as input and returning a confidence for the object label. In addition, simple binary rules, defined over pairs of regions, were used to determine the similarity of the regions and to form aggregations of regions with similar properties. Typically, the rules operate on primitives formed by a single segmentation process (e.g. regions or lines) and result in the merging of the primitives into a more complete description, depending on the confidence returned by the rules. Forming more abstract groups of elements in this way has advantages when dealing with unreliable segmentation processes: fragmented elements can be grouped to form aggregates which perhaps more closely match object models.

Recently, we have extended this approach to include relational rules, which capture expected relations between the elements of multiple representation (e.g. regions, lines, surfaces) of the image data [BEL85]. Using rules of this form, sets of elements across the multiple representations can be selected and grouped on the basis of relational scalar measures associated with each rule. The result, assuming the confidence value returned by the rule is high enough, is the construction of complex aggregations of elements which satisfy user-specified relations across the multiple representations. One advantage of this approach is that it is modular and extensible; when new representations are added to the system, integration is accomplished by adding the appropriate rules.

In our preliminary work, we are concerned with relational rules defined over regions and lines. Since both are defined in a pixel-based representation, a convenient basis for the rules is intersection of the correspoding sets of pixels. Such relational rules, called intersection rules, are composed of three components:

1) a *relational filtering rule* for selecting lines which intersect a region based on relational measures;

2) a *ranking rule* which ranks the lines which intersect a region based on line attributes; and

3) a *combination function* which calculates the final score of the region-line aggregation based on the scores from the *filtering rule* and the *ranking rule.*

The relational measures are used to measure the type and degree of the relationship between a region and a line. Lines associated with regions are categorized into three types: boundary lines, interior lines, and lines which are neither interior nor boundary. The measures are:

1. interior-line-percentage: the ratio of line area interior to the region to total line area.

2. region-perimeter-percentage: the ratio of region boundary pixels covered by the line area to the region perimeter.

3. line-length-percentage: the ratio of the length of the region boundary covered by the line area to the total length of the line.

The relational filtering rule is then a complex line rule composed of a simple rule for

each relational measure; in many cases it simply removes certain combinations of regions and lines from further consideration. The ranking rule ranks each line on the basis of how well it satisfies the associated relational measure. The combination rule is supplied the scores from the relational filtering rule, the line ranking rule, and the relational measures and converts these into a confidence for the hypothesis supported by the rule.

These intersection rules can be used in some very diverse ways. One example is to use a filtering rule on interior-line-percentage to select only those lines which are interior to a region. The ranking rule could then be defined to select short, high-contrast lines. The score of the ranking rule could then be averaged to form a complex texture measure. Alternatively, a density measure could be calculated by counting the occurrences of lines which receive a high score from the ranking rule and then normalizing by the size of the region.

As an additional example, the line-length-percentage measure could be used to select lines which lie mostly on the boundary of the region. The ranking rule could then be defined to favor long lines. The scores from the ranking rule could then be averaged using region-perimeter- percentage as a weighting factor to form a simple shape measure.

A preliminary implementation of the extended rule system has been completed, several simple texture and shape rules have been written, and results have been obtained on urban house scenes and on road scenes. The results [BEL85] are quite promising. For example, we have been able to find roads in several roadscenes by using a rule which implements a simple shape measure. In the future, we intend to write additional rules and apply the system to a larger variety of images, develop new rule types, add additional representations for motion, depth, and surface segmentations, and incorporate the rule-based system into

the schema system currently being developed (see next section).

## II.2. SCHEMA NETWORKS AS A REPRESENTATION OF KNOWLEDGE

In the VISIONS system, scene independent knowledge is represented in a hierarchical schema structure organized as a semantic network [HAN78, WEY83, PAR80, HAN83, WEY86]. The hierarchy is structured to capture the decomposition of visual knowledge into successively more primitive entities, eventually expressed in symbolic terms similar to those used to represent the intermediate level description of a specific image obtained from the region, line, and surface segmentations. Each schema defines a highly structured collection of elements in a scene or object; each object in the scene schema, or part in the object schema, can have an associated schema which will further describe it. Each schema node has both a declarative component appropriate to the level of detail, describing the relations between the parts of the schema, and a procedural component describing image recognition methods as a set of hypothesis and verification strategies called *interpretation strategies.*

The schema system provides a hierarchy of memory structures, from vertices (or even pixels) at the bottom level through semantic objects at the top. A further division of knowledge into long term (LTM) and short term memory (STM) across the levels of hierarchy provides a convenient way of differentiating the system's permanent *a priori* knowledge base from the knowledge that it has received or derived from a specific image. The goal of the system is an interpretation, by which is meant a collection of objects at the top level of STM that is consistent with both the image date and the system's *a priori* knowledge of the world as represented in LTM.

A central problem of high-level vision is how to make use of knowledge, not just to categorize the results of lower levels of computation but also to guide those levels through the space of image analysis and feature extraction techniques. Practical systems will need to know about an extremely large number of objects – a prohibitive number for any system that attempts to find each object in each image. Furthermore, there is a computationally explosive number of low and mid-level image operations (segmentation algorithms, texture measures, line finders, rectangle finders, line grouping operators, etc. which collectively are termed 'knowledge sources') which might be applicable, especially when one realizes that for almost every object there might be a variation of certain operations that would be particularly well suited to recognizing *just that object*. As a result, the combinatorics of what low- and mid-level processes to apply and how to interpret their results is simply too great to expect any near-term increase in the power of computing systems to solve the problem by brute force computation. The high level vision system must control the work being done at the lower levels for computer vision to be computationally feasible in the near future. The goal of this research, then, is to provide a prototype knowledge-driven system called the Schema System, to interpret images and provide control.

The development of the schema system confronts many of the same issues that have come up in other interpretation and control domains, such as speech understanding [LES75, WOO78]. Among them are questions of the knowledge representation, the communication of information, error recovery and the selection of knowledge sources.

A doctoral dissertation by Terry Weymouth [WEY86] presents our most recent approach to these problems. This dissertation explores the information and control structures needed for knowledge-directed interpretation of natural outdoor scenes. A *schema network* represents object descriptions, relations among objects, and control knowledge.

Each node of the network, a *schema*, contains both a declarative structure and references to one or more *interpretation strategies*. The declarative portion of the schema describes the composition of an object including the spatial relations of its parts and their possible appearances in an image. The interpretation strategies are object-specific procedures for creating hypotheses of the existence of the object. In the interpretation strategies, the procedural representation of control information provides a natural form for expressing the dynamic nature of the image interpretation.

A *schema instance* is created when a schema is activated either by a top-down request for a goal or by bottom-up detection of key events in the image. Schema instances continually interact with one another either through a channel set up when a goal is requested or through hypotheses created in a blackboard data structure. Several schema instances can work simultaneously on relitively independent portions of the interpretation, thus exploiting the potential for parallelism. By selectively grouping line and region primitives into descriptions of parts of a scene, the cooperative activities of the schema instances construct the final interpretation network.

The system was tested on six images from four scenes. Parallel activation of schemas is simulated; overlaping of the timing in the actions of a set of interpretation strategies is illustrated in traces from the simulation. The resulting interpretations contain both the association between object structures and image events and three-dimensional descriptions of some of the objects in the scenes.

Currently, as a result of this experience, we have initiated another stage of schema development by building a schema shell for experimental development, and by restricting the inter-schema communication to be entirely through the blackboard. What follows is

ongoing work which has not yet been evaluted. In general, there will be three types of messages that go on the schema blackboard: Hypotheses, Goals and Personal Mail. An important issue is when is information propagated, i.e. at what point does one schema instance's hypothesis effect another. We have adopted the basic principle that *the decision whether information should be propagated from one schema to another or not resides in the reader (given the blackboard communication), not the writer.* A schema instance must make sure the hypothesis has been posted by the time it is strong enough that another schema might use it.

The first schema prototype being developed is structured to have a collection of seven types of interpretation strategies (IS's), each of which runs as its own concurrent process. The most important IS is the Object Hypothesis Maintenance Strategy (OHM) which is responsible not simply for creating a hypothesis, but also maintaining it as the interpretation process proceeds and deciding how the hypothesis relates to the rest of the system. The remaining six IS's are initial hypotheses (typically inexpensive processes that give a first estimate as to whether the object exists in the image, and if so where), hypothesis expansions (e.g. an algorithm that expands a roof hypothesis, given just a corner of the roof), hypothesis support, conflict resolution, negative information (in general, how to use the information that something *isn't* a particular object), and information from subparts and/or superparts.

A programming shell has been created for research implementation of schema sets. Schema sets are groups of concurrent processes whose goal is to label a given type object, operating from high-level contextual and relation knowledge, and intermediate feature knowledge. The object labeling is implemented procedurally, which permits strategies to be tailored to the object being labelled with little interference from globally imposed data

structuring. At the same time, the lack of a global controller imposes a great deal of structure on interprocess communication.

The purpose of the shell is to encourage development and testing of labeling strategies by optimizing research and programming and testing time. A prototype shell is in place and 5 object schema types are at different stages of development and testing under the current shell. Feedback from these preliminary schemas will lead to improvements in the shell structure itself. The implementation is in Common LISP on the TI Explorers, with low level data and image processing functions handled on VAX.

## II.3. INFERENCE NET

We are actively exploring the mathematical foundations of a knowledge representation framework within the domain of vision using the theory of evidential reasoning as developed by Dempster [DEM68] and Shafer [SHA76]. The Dempster-Shafer formalism for evidential reasoning supports an explicit representation of partial ignorance, uncertainty and conflict. The inferencing model allows "belief" or "confidence" in a proposition to be represented as a range within the [0,1] interval. The lower and upper bounds represent support and plausibility, respectively, of a proposition, while the width of the interval can be interpreted as ignorance.

The representation has two components [REY85]. The first part is static, and explicitly associates measureable properties of some feature of the image data, via knowledge sources, to labels which are to be assigned to abstractions of the image data. This association is made using the notion of a mass-function as defined by Shafer. These mass functions are generated via the notion of a possibility function which is defined using explicit knowledge about the image domain in question. Previous methods required that the range of values

over which the mass functions are defined to be either explicitly or implicitly discretized into "feature propositions".

The second part uses the static representation, just presented a frame of discernment, and the theory of evidence as developed by Shafer and by Lowrance [LOW82] to combine the mass functions (via Dempsters rule), and arrive at a consensus opinion for the purpose of determining the correct label of the image abstraction. Assumptions about the image domain are represented within the knowledge network via possibility functions; a conflict value detects when an assumption has been violated and is used as a represention of uncertainty within the system.

Our representation provides a simple mechanism for representing uncertain information and for pooling of partial evidence. Assumptions one makes about the domain provide the constraints on the relationship between primitives extracted from the image data and objects in the scene one is trying to reason about; we are interested in obtaining and pooling evidence which pertains to these constraints. These include intrinsic properties of the objects, which are expressed as unary constraints, and contextual constraints such as spatial relationships which are binary or in general n-ary relations (for example adjacency is a binary relation, betweeness is a ternary relation).

# III. INTERMEDIATE LEVEL VISION

The general strategy by which the VISIONS system operates is to build an intermediate symbolic representation of the image data using processes which initially do not make use of any knowledge of specific objects in the domain. The result is a representation of the image in terms of intermediate primitives such as regions, lines, and local surface patches with associated feature descriptors. These primitives may be directly associated with an object label (using the rule-based object hypothesis system as described in the previous section) or they may be grouped into more abstract descriptions. The grouping processes may be guided by high level contextual constraints (e.g. top-down) which effectively select certain groupings related to the interpretation goals, they may be guided by very general object-independent constraints (e.g. bottom-up), or they may be guided by both, changing their form depending on the constraints available.

In this section we summarize three areas of research whose focus is the construction of intermediate level primitives and their features.

## III.1. GEOMETRIC GROUPING OF STRAIGHT LINES

The extraction of lines based on significant intensity changes and perceived boundaries between areas is a difficult and important step in image understanding. We have developed a new approach to the extraction of straight lines based on geometric grouping. The primary goal is the extraction of straight lines from images in which there are fragmented intensity discontinuities. The secondary goal is the demonstration that the use of geometric organization is an important part of the line extraction process and therefore can produce improvements when combined with standard edge detection techniques.

The algorithm has two major components: edge detection and hierarchical grouping. There are many edge detection algorithms which might be used. The main requirements are that it produce measurements of the intensity contrast and direction of the edge. The two algorithms that have been used for selecting points are zero crossings of the Laplacian operator [MAR80, CAN83] and the Haralick operator [HAR84].

The hierarchical grouping process is based on scale (but there is no smoothing) and two steps which are performed at each level: linking and merging. The hierarchical representation is a compact representation which reduces the search space at each level for sequences of linked edges. It reflects the observation that "closeness" of lines is scale dependent and is a multi-scale representation of a line which may be straight only at large scales.

The linking process is based on intrinsic and geometric properties. It searches a space of lines for almost colinear pairs which are close to each other and links the appropriate endpoints. There are four criteria used for linking:

1. Similar gradient magnitude. The gradient magnitudes across the edge must be close to each other and in the same direction.

2. The lines must be approximately colinear. Lines 180 degrees apart are not linked.

3. The end points of two candidate lines must be close.

4. The lines must not overlap. If both endpoints of one line project within corresponding endpoints of the other, they are not linked.

The merging process consists of grouping and replacement. If a sequence of linked lines can be approximated sufficiently well by a straight line, then they are grouped and

are replaced by a straight line.

This approach has a number of advantages for extracting straight lines. The results shown in [WEI85] indicate that the principle of geometric grouping for extracting long straight lines gives significant improvement in the results obtained from standard edge detection algorithms. For example, line segments can be linked even when they are separated by gaps. We believe that the approach can be extended to curved lines, (see Section III.3) parallel lines, closed contours, and other geometric abstractions.

Although the algorithm is very robust in its extraction of straight lines, it has some problems which we are continuing to investigate. The ability of the algorithm to bridge gaps is simultaneously one of its strengths and one of its weaknesses. Gaps sometimes appear in a line because of changes in the lighting conditions along the line, (such as shadows and specular reflections) which in turn affects the magnitude of the gradient. These gaps should be bridged. Other apparent gaps are caused by the alignment of distinct lines (such as those on the top or bottom of a pair of shutters); such gaps are real and should not be bridged, yet at some level in the hierarchical representation they appear as one line. Methods must be found for analyzing the multi-scale representation and for determining what scales are appropriate and which are not appropriate.

The algorithm, like many others, relies on intensity gradient information to link lines, yet what we perceive as straight lines are not always collections of edges with similar intensity gradients. Finally, the algorithm will often find long lines in heavily textured areas because of accidental alignment of texture edges. We are investigating the possibility of using texture measures to inhibit the linking step.

## III.2. EXTRACTION OF CURVED LINES

Until recently the traditional method in computer vision for extacting straight and curved lines has been either through the use of the Hough transform or via "edge linking" algorithms applied to the output of some "significant edge pixel" algorithm. However, a novel approach for extacting straight lines was recently reported in Burns, Hanson and Riseman [BUR84], and involved a simple local computation (not involving any histogram methods), followed by a computation of connected components.

The central module of this algorithm was a grouping process using overlapping partitions on gradient orientation. In the context of extracting straight lines this process can be summarized as follows:

- Compute the gradient orientation at each pixel.
- Partition the 360 degree gradient orientation measure into non-overlapping sectors (normally 8 or 16 are used) and label the image according to the sector into which the gradient orientation falls.*
- Apply a connected components algorithm to the quantized gradient orientation, thereby producing regions with pixels having similar gradient orientation.
- Fit a straight line to the resulting "edge support" regions.

We have been investigating the application of this general approach to the problem of extracting semi-circular arcs, replacing gradient orientation with a curvature measure. Specifically we find "curve support regions" which are uniform with respect to a curvature

*Note that the process is actually somewhat more complex in that two sets of sectors are employed; the second set is applied with sectors rotated a half interval (see[BUR84]).

measure and as such can be abstracted from the image data as a part of a circle.

The curvature measure be used is given by the Kitchen-Rosenfeld curvature operator [KIT80] defined by:

$$K = I_{xx} \cdot I_x^2 + I_{yy} \cdot I_y^2 - 2I_{xy} \cdot I_x \cdot I_y (I_x^2 + I_y^2)^{3/2}.$$

In fact this measure only makes sense when applied to areas of locally maximum gradient magnitude, ie. zero crossings of some second derivative operator. Thus, this curvature algorithm can be summarized as follows:

- Apply the curvature measure along the zero-crossing contour.
- Partition the range of the curvature measure into sectors.
- Label each pixel according to the partition into which the curvature value falls.
- Produce regions by applying a connected components algorithm to the labels of the curvature partitions.
- Fit a semi-circle to each curve-support region.
- Repeat the above process for a second set of sectors rotated one-half sector.
- Each pixel then votes for one of the two regions which it is a part of in the two representations, specifically the region whose extracted curve is longest. The percentage of pixels within a region that vote for that region is the support of the region.
- Normally the regions selected are those whose support is greater than 50 percent.

Associated with each curve is a set of curve attributes, such as length, center, radius,

endpoint parameters, contrast and support. The algorithm is local in nature and is hoped to be robust in the face of moderate amounts of noise due to the coarse partitioning of the output curvature measure .

In summary, we are developing a system to derive local and piecewise circular descriptors of the image data. The approach utilizes local 2D operations and is computable in parallel. Curves are partitioned based on constancy of curvature rather than usual extrema methods. Also in contrast to other approaches, descriptors of neighboring segments are treated as independent, with the expectation that higher level processes will guide the next level of grouping. The system is designed to provide reliable local primitives (as opposed to pixel level events) for the purpose of moving up the abstraction hierarchy within the image understanding system.

## III.3. APPLICATION OF VANISHING POINTS TO 3-D MEASUREMENT

Perspective is an important cue to 3d spatial information such as the direction of lines or the orientation of surfaces. Human beings can perceive three-dimensional objects in space even when looking at two-dimensional images. A computer vision system must do likewise, but 3D shape, size and location cannot be recovered from a single image without additional information or assumptions. Vanishing points and vanishing lines can provide this information in the case of objects which are assumed to have parallel lines or parallel edges planar surfaces in the 3D world. Once the location of the vanishing point is detected, we can use it as a cue to calculate the distance and shape of the object to which the parallel lines belong [NAK80, NAK84a, BAR82].

Estimation of the errors in these features has practical significance and could be used in many ways. A modular process such as a knowledge source on perspective could use

this type of information in the form of constraints to generate and verify hypotheses. With some object models such as buildings, the dihedral angles between their surfaces (e.g. walls and roofs) are invariant shape features. An estimate for the relative orientation between surfaces which incorporates a model of error allows one to verify hypotheses in the face of imprecision in low-level processing. For example when analyzing a house scene, if two adjacent regions are temporarily labeled as house walls based on some property, say rectangular shape, the calculation of the mutual angle of the two surfaces can be used to verify this. This means that we must know whether the measured angle is outside the estimated range of error.

Parallel lines in 3d space are projected onto the image plane to lines which radiate from a single common point, called a vanishing point (VP). It can be used to calculate the size and orientation of objects with parallel lines. The vanishing line for a surface can be computed as the line passing through two VP's obtained from two sets of parallel lines. There are infinitely many sets of parallel lines which could be drawn in a given plane and the vanishing point for each set lies on this vanishing line.

The surface orientation of a plane is given by the unit normal vector perpendicular to the surface. The vanishing line (VL) of a plane gives a precise description of the unit normal. The distance from the VL to the center of the image plane corresponds to the angle of tilt of the surface away from the viewer. If the line goes through the center, then the normal to the surface is parallel to the viewing plane. The second angle of the surface is given by the orientation of the vanishing line; its normal is the projection of the normal to the surface onto the image plane. Thus, analysis of the errors in the distance and orientation of the vanishing line can be related to effect on the estimates for the surface orientation and line length. We have developed formulae for these errors as a function of

VP or VL errors, and we have used constraints based on real world knowledge to increase the precision of the estimates of surface orientation (NAK84b). The assumptions made are that the focal length of the camera and the depth of one point on a line are known. If the depth is not known at all, then the orientation can still be recovered, but only relative distances can be estimated.

The algorithm for locating vanishing points of a set of lines could take place as a form of Hough transform. Lines are extracted which are likely to be parallel (e.g. by selecting all lines which are nearly parallel in the image and spatially clustered. These lines are stereographically projected onto half of the Gaussian sphere and extended to semicircles. Peaks are located on the Gaussian sphere by simply thresholding. There is also the possibility of knowledge-directed selection of lines in the image (possible only pairs) which are assumed to be parallel in the world.

The estimation of the surface normal from the estimates for two vanishing points involves intersecting constrained regions on the Gaussian sphere. Each vanishing point estimate, which is an area on the Gaussian sphere, determines an annular set of possible directions for the normal to the surface. The intersection of these annular sets is the estimate for the normal.

For the application of geometric constraints in the case where two house walls are perpendicular, the estimate for the normal for one wall was rotated 90 degrees on the Gaussian sphere and intersected with the estimate for the normal to the other wall. In our experiments, there was significant reduction in the size of the error region estimate for the surface normal in the example used.

Although we assumed the perpendicularity between the planes in the two cases men-

tioned above, we can apply this method even if the dihedral angle is not a right angle. If the angle is given to be $\theta_S$, the other plane's normal must be in the belt that makes an angle $\theta_S$ with the given normal. We can again form the consistent range by taking the intersection of the belt and the area for the VP. These constraints can also be applied to more than two planes, for example when three planes meet in a trihedral angle.

# IV. THE UMASS IMAGE UNDERSTANDING ARCHITECTURE PROJECT

Our research group is designing and constructing a highly parallel architecture for computer vision with the goal of achieving real-time processing rates for low, intermediate and high level image interpretation tasks. This architecture consists of three tightly coupled layers that correspond to these levels of abstraction. These layers are the Content Addressable Array Parallel Processor (CAAPP) at the bottom, Intermediate and Communications Associate Processor (ICAP) in the middle, and the Symbolic Processing Array (SPA) on top. Attached to the SPA is a host processor.

The CAAPP is an associative square grid processing array that is designed to provide bi-directional parallel communication between symbolic and sensory processing [WEE83, WEE84, LEV84]. The ICAP is also an associative square array, and is tightly coupled to the CAAPP and SPA. The purpose of the ICAP is to perform intermediate level symbolic processing such as geometric grouping and to facilitate the flow of information and control between the CAAPP and SPA. The SPA is an array of processors which perform high level symbolic processing such as hypothesis generation and testing, schema processing, and knowledge source/blackboard processing.

The multilayer associative structure of the UMass architecture provides simultaneous parallelism at three different levels of abstraction with high bandwidth bi-directional flow of information and control between the levels. This permits the entire iconic to symbolic transformation process to take place within the architecture so that the top layer can provide a high level symbolic interface to the image interpretation process. At this level, images of the environment have essentially been transformed into a symbolic representation

of that environment.

The effort involves a custom VLSI implementation for the processing elements in the bottom two layers of the architecture; a systems hardware implementation for integrating the custom processors with off-the-shelf components in the top layer and host processor; a software development effort for creating a complete programming environment, simulators and tools for the system; and an algorithm development effort for implementing vision algorithms on the architecture. This project, particularly the VLSI implementation effort, will be shared with Hughes Research Labs.

IV.1. Hardware:

A test chip of the NMOS version of the CAAPP processing element has just been received from the MOSIS facility. We are currently preparing to test this chip. The layout for a CMOS version of the CAAPP processing element is about 60 percent complete. We will be examining the tradeoffs involved in going to a CMOS implementation. Although CMOS would increase the size of the layout, it would permit the use of the MOSIS scaleable rules, with a potential for significant size reduction and speed increase in the future.

The first pass on the design for the Intermediate and Communications Associative Processor (ICAP) has been completed. Unfortunately, to place this ICAP design on the same chip as the CAAPP cells will necessitate a greater number of pins than is currently available from MOSIS. Thus we are examining the tradeoffs of reducing the functionality of the ICAP to make it fit the pin limitations, versus placing the ICAP on a separate chip. The latter would double the size of the prototype circuit boards, but would provide greater processing flexibility.

## IV.2. Software and Algorithms

We are currently constructing an instruction-level functional simulator for the new CAAPP architecture. This simulator promises to provide considerably greater execution speed than the old simulator. The new simulator is being written in C as a stand-alone, portable system although its image formats will be compatible with the UMass Image Operating System (IOS) of the VISIONS project. Once the ICAP architecture is finalised, it will also be incorporated into the simulator.

An iconic to symbolic transformation process has been developed and tested for the CAAPP, using a version of the IOS to simulate the new CAAPP architecture, prior to construction of the new simulator. Several vision algorithms have been implemented in the simulator; these include an algorithm for computing approximations to large Gaussian convolutions, the Burns' line extraction algorithm, and the line grouping algorithm described in Section III.1.

A prototype slice of the UMass architecture is scheduled for completion in approximately 2 years. This will produce a symbolic representation of region and line image events, as well as surfaces, and can be interfaced to a LISP processor as a demonstration of the concept. The complete prototype could be built in two additional years. At the end of the first year, the software effort will produce simulators and tools for the bottom two layers of the architecture. The second year of the software effort will result in a transportable, stand-alone simulator for the entire architecture with associated environment and tools. After this, the software effort will concentrate on implementing vision processing tasks on the simulators and then transferring those implementations to the hardware as it becomes available. Additional enhancements to the environment and further tools will be developed

as necessary.

## REFERENCES


[ADI85a] G. Adiv, "Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Trans. Pattern Anal. Machine Intell.*, Volume PAMI-7, July 1985, pp. 384–401.

[ADI85b] G. Adiv, "Interpreting Optical Flow," Ph.D. Dissertation, Computer and Information Science Department, University of Massachusetts at Amherst, September 1985.

[ANA84] P. Anandan, "Computing Dense Displacement Fields with Confidence Measures in Scenes Containing Occlusion," *SPIE Intelligent Robots and Computer Vision Conference*, Volume 521, 1984, pp. 184–194; also *DARPA IU Workshop Proceedings*, 1984; and COINS Technical Report 84-32, University of Massachusetts at Amherst, December 1984.

[ANA85] P. Anandan and R. Weiss, "Introducing a Smoothness Constraint in a Matching Approach for the Computation of Optical Flow Fields," *Proc. of the Third Workshop on Computer Vision: Representation and Control*, October 1985, pp. 186–196; also in *DARPA IU Workshop Proceedings*, 1985.

[ARB86] A. Hanson and M. Arbib, Vision, Brain, and Cooperative Computation, to be published by MIT Press, Cambridge, MA 1986.

[BAR82] S.T. Barnard, "Interpreting perspective images," SRI Technical Note 271, 1982.

[BEL85] R. Belknap, E. Riseman, and A. Hanson, The Information Fusion Problem and Rule-Based Hypotheses Applied To Complex Aggregations of Image Events, Proc. DARPA

IU Workshop, Miami Beach, FL December 1985.

[BHA85] R. Bharwani, A. Hanson, E. Riseman, Refinement of Environmental Depth Maps over Multiple Frames, Proc. DARPA IU Worksop, Miami Beach, FL. December 1985.

[BUR84] J.B. Burns, A. Hanson, and E. Riseman, Extracting Linear Features, Proc. 7th ICPR, Montreal, 1984. Also COINS Technical Report 84-29, August 1984. To appear in IEEE PAMI.

[CAN83] J.F. Canny, Finding Edges and Lines in Images, MIT AI Lab Technical Report No. 720, June 1983.

[DEM68] A.P. Dempster, A Generalization of Bayesian Inference, Journal of the Royal Statistical Society, Series B, Vol. 30, 1968, pp. 205-247.

[GLA83] F. Glazer, G. Reynolds, and P. Anandan, Scene Matching by Hierarchical Correlation, Proc. IEEE CVPR, June 1983, pp. 432-440.

[HAN78] A. Hanson and E. Riseman, *VISIONS: A Computer System for Interpreting Scenes*, Computer Vision Systems (A. Hanson and E. Riseman, eds.) (1978), 303 - 333, Academic Press.

[HAN83] A. Hanson and E. Riseman, *A Summary of Image Understanding Research at the University of Massachusetts*, COINS Technical Report 83-35 (October 1983), University of Massachusetts at Amherst.

[HAR84] R.M. Haralick, Digital Step Edges from Zero Crossing of Second Directional Derivatives, IEEE Trans PAMI 6, January 1984, pp. 58-68.

[HOR81] B.K.P. Horn and B.A. Schunck, "Determining Optical Flow," *Artificial Intelli-*

*gence*, Volume 17, 1981, pp. 185–203.

[KIT80] L. Kitchen and A. Rosenfeld, A Gray Level Corner Detector, Tech. Report No. 887, Computer Science Center, University of Maryland, College Park, MD, 1980.

[LAW84] D.T. Lawton, Processing Dynamic Image Sequences form a Moving Sensor, Ph.D. Dissertation (TR 84-05), Computer and Information Science Department, University of Massachusetts, 1984.

[LES75] V.R. Lesser, R.D. Fennell, L.D. Erman, and D.R. Reddy, Organization of the Hearsay-II Speech Understanding System, IEEE Trans. on ASSP 23, pp. 11-23.

[LEV84] S.P. Levitan, Parallel Algorithms and Architectures: A Programmers Perspective, Ph.D. Dissertation (COINS Technical Report 84-11), Computer and Information Science Department, University of Massachusetts, May 1984.

[LOW82] J. Lowrance, Dependency Graph Models of Evidential Support Ph.D. Thesis, University of Massachusetts, Amherst, 1982; also COINS Technical Report No. 82-26.

[MAR80] D. Marr, and E. Hildreth, Theory of Edge Detection, Proc. of the Royal Society of London, B., 207, pp. 187-217.

[NAK80] H. Nakatani, et al. "Extraction of vanishing point and its application to scene analysis based on image sequence," 5th Int. Conf. on Pattern Recognition, pp. 370–372, 1980.

[NAK84a] H. Nakatani, T. Kitahashi, "Inferring 3–d shape from line drawings using vanishing points," 1st Intn'l Conf. on Computers and Applications, 1984.

[NAK84b] H. Nakatani, R. Weiss, and E. Riseman, "Application of Vanishing Points to

3D Measurement," Proc. SPIE, Vol. 507, 1984, pp. 164–169.

[PAR80] C.C. Parma, A.R. Hanson and E.M. Riseman, *Experiments in Schema-Driven Interpretation of a Natural Scene*, COINS Technical Report 80-10 (April 1980), University of Massachusetts at Amherst.

[PAV85] I. Pavlin, A. Hanson, and E. Riseman, Analysis of an Algorithm for Detection of Translational Motion, Proc. DARPA IU Workshop, Miami Beach, FL, December 1985.

[REY85] G. Reynolds, D. Strahman, N. Lehrer, Converting Feature Values to Evidence, Proc. DARPA IU Workshop, Miami Beach, FL, 1985.

[RIS84] E. Riseman and A. Hanson, A Methodology for the Development of General Knowledge-Based Vision Systems, Proc. of the IEEE Workshop on Principles of Knowledge-Based Systems, Denver, Colorado, December 1984.

[SHA76] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.

[STR84] T. Strat, Continuous Belief Functions for Evidential Reasoning, Proc. AAAI-84, pp. 303-313.

[ULL79] S. Ullman, The Interpretation of Visual Motion, MIT Press, Cambridge, MA 1979.

[WEE83] C. Weems, S. Levitan, D. Lawton, and C. Foster, A Content Addressable Array Parallel Processor and Some Applications, Proc. DARPA IU Workshop, Arlington, VA, June 1983.

[WEE84] C. Weems, S. Levitan, C. Foster, E. Riseman, D. Lawton, A. Hanson, Development and Construction of a Content Addressable Array Parallel Processor (CAAPP)

for Knowledge-Based Image Interpretation, Proc. Workshop on Algorithm-Guided Parallel Architectures for Automatic Target Recognition, Leesburg, VA July 16-18, 1984, pp. 329-359.

[WEI85] R. Weiss, A. Hanson, and E. Riseman, Geometric Grouping of Straight Lines, Proc. 1985, DARPA IU Workshop, Miami Beach, FL, 1985.

[WES82] L. Wesley and A. Hanson, The Use of an Evidential-Based Model for Representing Knowledge and Reasoning about Images in the VISIONS System, Proc. Workshop on Computer Vision, Rindge, NH, August 23-25, 1982.

[WES83] L. Wesley, Reasoning about Control: The Investigation of an Evidential Approach, Proc. 8th IICAI, Karlsrube, West Germany, August 1983, pp. 203-210.

[WES85] L. Wesley, Ph.D. Thesis, University of Massachusetts, Amherst, in preparation.

[WEY83] T.E. Weymouth, J.S. Griffith, A.R. Hanson and E.M. Riseman, *Rule Based Strategies for Image Interpretation*, Proc. of AAAAI-83 (August 1983), 429-432, Washington D.C. A longer version of this paper appears in Proc. of the DARPA Image Understanding Workshop (June 1983), 193-202, Arlington, VA.

[WEY86] T.E. Weymouth, Using Object Descriptions in a Schema Network for Machine Vision, Ph.D. Dissertation, Computer and Information Science Department, University of Massachusetts, Amherst. Also COINS Technical Report, (in progress), University of Massachusetts, Amherst.

[WOO78] W.A. Woods, Theory Formation and Control in a Speech Understanding System with Extrapolation Towards Vision, in Computer Vision Systems (A. Hanson and E. Riseman, Eds.), Academic Press, 1978.