A Homogeneous Framework for Visual Recognition

Rick Kjeldsen, Ruud ML Bolle, Andrea Califano, Russell W. Taylor
Exploratory Computer Vision Group
IBM Thomas J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598

Abstract

A homogeneous paradigm for evidence integration is presented, and a vision system to recognize 3D objects is demonstrated using this paradigm. A new concept called *generalizedfeatures* supports a highly modular architecture, and allows a uniform treatment of features at all levels of recognition - from simple partial features to complex feature assemblies and 3D objects. Layered, concurrent parameter transforms vote for feature hypotheses on the basis of image data and previously reconstructed features. Additional transforms identify supporting or conflicting relationships between hypotheses. The entire reconstruction and indexing process occurs within *recognition networks,* which collect votes, fuse evidence from various sources and insure global consistency. The overall approach allows the system to completely avoid the common weak point of explicit, low-level scene segmentation. Features reconstructed by the vision system include surface regions and 3D surface intersection curves. Experimental results, including noise sensitivity, for real data from a laser range finder are presented.

1. Introduction

Of the many approaches which have been proposed for object recognition (see surveys [Binford, 1982, Besl & Jain, 1985]), few can be considered a general framework capable of supporting a wide variety of recognition strategies. Those which can, typically address only a portion of the problem [Bolle *et al.*, 1986] or propose differing approaches for each stage of the recognition process [Weems *et al.,*1989], In general there is a lack of consistency both from low to high level processing (e.g., from feature extraction to object recognition) and between various feature extraction mechanisms. The representation of hypotheses, the control structure and the representation of feature extraction knowledge can all vary. This poses several problems, mostly related to communicating and combining information:

- Communication between modules of the system is hindered, making it difficult for the various recognition pathways to cooperate.
- Multiple, unique internal representations make the comparison of hypotheses generated by different modules more difficult.
- Addition of new feature types is a nontrivial process, so the system is often locked into a static set of primitives (which in turn requires great care in choosing the primitives).
- The integration of different data sources (e.g. reflectance and range data) is made more complex.

This paper describes an approach which attempts to address these concerns. A uniform structure is defined, using a new concept of *generalizedfeatures.* This identifies exactly what types of information are needed to extract a new feature while putting as few limitations as possible on the feature extraction processes which may be used. Hypotheses throughout the system are represented and treated uniformly using an evidence integration scheme motivated by work in connectionist systems (recognition networks [Feldman & Ballard, 1981, Sabbah, 1985]). Together, these concepts create a framework in which it is relatively easy to combine disparate feature types, allowing them to interact yet retaining a high degree of modularity.

Figure 1 represents an overview of the approach. Recognition is structured as a hierarchy of layered and concurrent parameter transforms [Ballard, 1981]. Features are extracted in concurrent recognition pathways. Each pathway is made up of one or more stacked parameter transforms. The transforms generate feature hypotheses using data from the scene and features extracted in lower layers. Pathways can interact at any level to exchange information. Knowledge about interactions between features is used to generate evidential links between hypotheses throughout the system. A global interpretation is arrived at using an iterative process which fuses the evidence for and against each hypothesis.

The primary advantages of this approach are the modularity, which allows the system to be extended as needed, and consistency which provides the ability to

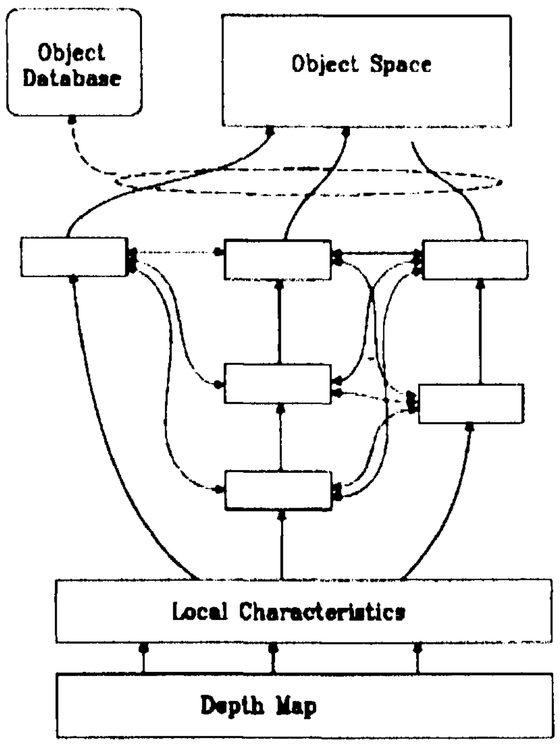

Fig. 1: Architecture Overview

integrate multiple types of evidence and provides a consistent conceptual framework.

1.1. Paper outline

Section 2 of this paper describes the generalized feature concept Section 3 discusses in detail the organization of the recognition process, that is, the use of parameter transforms and recognition networks. The remainder of the paper discusses a system which implements our ideas. The extraction of primitive features, such as, 3D curves and surfaces is discussed in Sect 4. In Sect. 5, we discuss our object matching strategy, and show its equivalence to the other recognition steps. Experiments are given in Sect. 6 that demonstrate the behavior of the system when analyzing scenes with multiple occluding objects, using depth data obtained from a laser range finder. We also discuss a qualitative analysis of the behavior of the system in the presence of noise.

2. The generalized feature approach

The term *feature* has been widely used in pattern recognition and artificial intelligence research over the past several decades. As a result, it no longer has a single canonical meaning. In the context of our vision system, feature denotes any entity that can be parameterized.

To aid discussion of the feature hierarchy in the system, we classify features as *local, partial, primitive* or *assembly* features, in order of increasing level of abstraction from the original data. For an example consider lines in three-space. The line is a *primitive* feature contained in many higher level feature *assemblies* such as parallelepipeds. But it is also formed from *partial* features such as orientation and position, which, in turn, are extracted from *local* features such as the output of an edge detector.

We have defined a Generalized Feature concept to impose a uniformity on the features in a system, allowing them to interact and creating a uniform conceptual approach. A generalized feature type is defined by a parameterization and a set of relationships to other features. The relationships are defined using two types of knowledge; knowledge about the characteristics of lower level features (or input data) which provide evidence for a feature, and knowledge about relationships between feature hypotheses.

Thus, in order to introduce a feature, we establish the particular parameterization and define procedures to compute these parameters using lower level data *{parameter transforms),* and procedures to identify evidential relationships with other features at any level of any path *{compatibility relations).* The generalized feature concept defines what types of knowledge are needed from these procedures without putting limitations on how that knowledge is obtained. It specifies that the parameter transforms will be initiated whenever a hypothesis survives iteration (as will be discussed shortly) and supplies the parameters of features (if any) which that hypothesis supports. The compatibility relations will be initiated with each newly created hypothesis and returns each existing hypothesis which should support or compete with it. Any procedure which meets these criteria can be used.

In the above example, a parameter transform will be triggered when a local discontinuity is identified and will return the parameters of the line. A line hypothesis will be created (unless it already exists) and passed to a compatibility relation. This will return, for example, all the previously existing line hypotheses which represent alternative interpretations of the discontinuity, and so should compete with it.

A feature type may have several parameter transforms or compatibility relations leading to it from various parameter spaces. Thus any particular hypothesis may receive support from multiple sources, potentially in different parameter spaces. For example, assembly hypotheses receive support from hypotheses in many of the primitive feature spaces.

3. Evidence Integration

The parameter transforms and compatibility relations identify the alternative feature hypotheses with the evidence for and against each. In order to evaluate hypotheses through-out the system and reach a global consensus we use a homogeneous, feature independent control structure. This takes the form of a recognition network [Feldman & Ballard, 1981, Sabbah, 1985] where nodes represent feature hypotheses and links represent the evidential relationships between features. Iterative refinement, similar to that used in connectionist systems, allows the network to determine which hypotheses best explain the data.

Hypotheses are collected in a *parameter space* associated with each feature type. Each parameter space is a sub-net of the recognition network. Parameter transforms and compatibility relations map from some input parameter space into some other parameter space [Ballard, 1981] and are used to accumulate evidence for feature hypotheses in a manner similar to the Hough transform [Hough, 1962].

The links in the network are (1) bottom-up connections as identified by the parameter transforms, and (2) lateral links between nodes, identified by the compatibility relations. Lateral links are inhibitory, if the hypotheses are conflicting, or excitatory, in case the hypotheses are supporting one another. Any link can have an associated weight representing the strength of the evidential relationship. Hypotheses and links are generated dynamically at run time.

Each node computes an activation level representing the confidence in the existence of the corresponding feature or object in the input. At each iterative step /, the activation level of a node, denoted by $AL_{de}\{i)\hat{}$ is computed as

$$AL_{node}(0) = 0 \qquad (3.1)$$
$$AL_{node}(i) = AL_{node}(i-1) + BU_{node} + LE_{node}(i-1)$$

where BU_{nod} represents bottom-up reinforcement (see [Sabbah *et al*1§86]). LE_{node}presents excitation from other hypotheses; $LI_{node,}$ inhibition from other hypotheses. The amounts of *LI* and *LE* depend on the activation levels of the competing/supporting nodes. *D* is a decay term that helps suppress spurious (noise) hypotheses.

A set of mutually inhibiting units in the network form a "winner-take-all" sub-network [Feldman & Ballard, 1981, Sabbah, 1985] where only one unit will survive. The function of this is twofold. It sharpens the response of the transforms

and it provides an implicit segmentation. Sharpened response comes about as a result of inhibition within a parametric neighborhood. Only the strongest unit in any neighboring cluster will survive. Implicit segmentation results when inhibition links are established on the basis of shared support If hypotheses which are supported by common image pixels inhibit each other, only those which do not share portions of the image will survive. These will represent a spatial segmentation of the image. This behavior can be generalized to other shared characteristics as well, for example primitive features shared between object hypotheses. We view this as a significant deviation from the "classical" segmentation schemes.

If a unit in a space survives, it and its associated input data points are used by the parameter transforms to generate bottom up votes for hypotheses in higher level parameter spaces. Survival is determined as a function of *(AL* - a L I where a is a parameter. Thus hypotheses with high activation and few competing alternatives survive. Survivors from the winner-take-all sub-networks are *stable coalitions* [Feldman & Ballard, 1981] of feature hypotheses which represent globally consistent interpretations of the input data.

We believe these methods provide a powerful approach to recognition. Consistency and modularity, which are the cornerstones of the approach, provide a host of secondary benefits:

*Extending the system to incorporate additional features becomes relatively easy. Thus a rich and varied feature set is possible.

*The ability to integrate multiple recognition pathways allows graceful degradation of performance as feature recognition mechanisms fail. For instance a planar surface and its bounding edges supply redundant information that allows for more robust recognition of bounded planar patches than either feature alone, however either one can supply information about planar regions in the absence of the other.

- Lateral inhibition eliminates the need for an explicit segmentation step.

- The architecture provides a natural vehicle for layered feature extraction. Use of multiple layers reduces the dimensionality of each step, while the iteration at each step drastically reduces the total number of hypotheses, compared to a single layer process.

4. A Vision System

Using these techniques we have developed a system capable of recognizing objects in depth maps of complex scenes. This section will describe the features we use and the parameter transforms which extract them.

4.1. System features

In selecting primitive features for the system, we have endeavored to select a set that will allow the creation of a large and varied object database. CAD primitives suggest a set of features useful for industrial part recognition. While we have chosen this particular feature set, it is important to note that our system has no dependency on any specific choice of features. With the generalized feature concept, as we expand the system's scope, features can be incrementally added or the feature set substantially changed with little effort.

Our surface feature set consists of planes and quadrics of revolution, specifically spheres, cylinders, and cones. A survey of industrial parts [Hakala *et al.,*1980] indicates that surface features alone should allow us to model the large majority (about 85%) of man-made parts.

To increase coverage, we also include curves in three-space, namely lines and conic-sections. These correspond to intersections and boundaries of surface patches. Because the information contained in curves is redundant to some extent with that in surfaces, we are capable of more robust recognition than with surfaces alone.

4.2. Local feature extraction

We define 3D points that lie on depth discontinuities as local curve features. Well-known edge detectors [Rosenfeld & Kak, 1982] are used to generate discontinuity maps; in combination with the range data this gives us a set of range points q near zero and first order depth discontinuities.

Local surface features are extracted from smooth surface approximations to the depth map. That is, least-squares second order polynomial approximations [Bolle *et al.,*1987] are made within *MxN* areas about range point q. From these approximations, the principal curvatures, K_{max} and K_{min}, and the associated principal directions in three space, X_{max} and Xmin, for each range point q are computed [DoCarmo, 1976]. Hence, the 3D location on the surface, and the principal curvatures and directions, are the local surface features.

43. Multiple window parameter extraction

To extract the parameters of complex geometric entities, one would like to devise an *M* x *M* operator that computes some parametric description of the curves and surfaces. To avoid interference from nearby local features, the size *M* of the operator should be small-but makes estimates of higher-order properties of the curves and surfaces inaccurate.

To solve these problems, we use the correlated information embedded in different windows that contain portions of a feature. For both curve and surface extraction, we use a set of nearby range points or windows to examine a more global neighborhood and extract the parameters of our primitive features. Specifically, we use all possible combinations of n windows in groups of *k* (within some *radius of coherence)* to generate the hypotheses. Though many spurious ones are generated, only those actually present collect sufficient evidence to survive. Hence, we extend the pure local parameter extraction to a somewhat more global process of parameter estimation, while maintaining the same order of computational complexity. This is illustrated in Fig. 2 for the case of ellipse finding using multiple windows. (The multiple window approach is described in detail in [Califano, 1988, Califano *et* a/., 1989].)

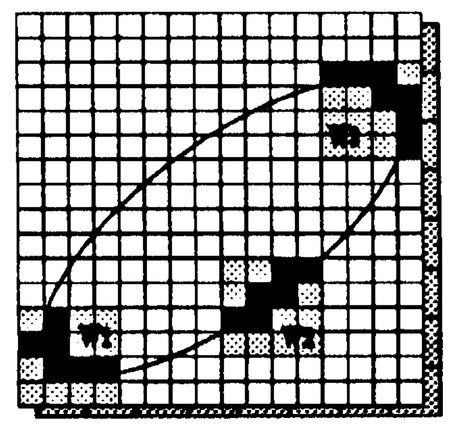

Fig. 2: Using three windows for finding an ellipse.

4.4. First level parameter transforms

Recognition is initiated by simultaneously generating hypotheses for five different low-level geometric features; (1) lines in three-space that correspond to linear surface intersections and occluding boundaries, (2) 3D planar curves formed by edge elements (i.e., planes that contain 3D curves), (3) points in three-space that correspond to sphere centers, (4) lines in three-space corresponding to axes of revolution (of cylinders,...), and , (5) planar surfaces.

Curves: Two 3D points determine a line, the further apart the more accurately the line is determined in the presence of spatial and quantization noise. Therefore, we compute the scatter matrices of combinations of two windows that contain range points on surface discontinuities. For each matrix, the eigenvector corresponding to the largest eigenvalue gives the orientation of the line. The mean vector $\bar{\mathbf{q}}$ determines the location of the line. Similarly, triplets of windows are used to determine a plane in three-space that could contain a 3D curve. For each scatter matrix, the eigenvector associated with the minimum eigenvalue gives the normal to a plane, while $\bar{\mathbf{q}}$ gives a point on the plane.

Surfaces: Consider two range points $\mathbf{q_a}$ and $\mathbf{q_b}$ that lie on a quadric of revolution (not an oblate ellipsoid). Then, the plane P_a containing $\mathbf{q_a}$ and spanned by $\mathbf{X}^{\mathbf{a}}_{\mathbf{min}}$ and the normal $\mathbf{N^a}$ contains the axis of revolution. Similarly, for plane P_b spanned by the normal at point $\mathbf{q_b}$ and the direction of minimum curvature. Hence, the intersection of these two planes gives us a hypothesis for the axis of revolution. Now, suppose that the two points $\mathbf{q_a}$ and $\mathbf{q_b}$ both lie on a spherical surface patch. The point of intersection of $\mathbf{N^a}$ and $\mathbf{N^b}$ (or their point of closest approach) creates the sphere center hypothesis. For the generation of nodes in the network of planar surfaces, we use a different technique. Three points $\mathbf{q_a}$, $\mathbf{q_b}$, $\mathbf{q_c}$ and their local neighborhoods are used to compute scatter matrices of the points. Based on the eigenvalues and eigenvectors, hypotheses about planar surfaces are generated.

4.5. Higher-level parameter transforms

When a first-level hypothesis survives in its winner-take-all sub-network, it becomes an input to a higher-level parameter transform. This transform generally reexamines the set of image points, $\{\mathbf{q}_n\}$, which support the hypothesis.

Planar curves: Reconstruction of planar curves from the curve-containing plane (CCP) is done in the following way. A new system of coordinates (x', y', z') is introduced with x' and y' coordinate axes contained in the CCP. Since $z'=0$ for all $\{\mathbf{q}_n\}$, the problem is reduced to the extraction of a conic in two-space. A conic is fit to triplets of neighborhoods on the CCP using a technique described in [Califano, 1988, Califano *et al.*, 1989, Bookstein 1979]. The conic in three-space is completely determined from the parameters of the conic in two-space and the CCP coordinate transform.

Surfaces: The radius R associated with each point in $\{\mathbf{q}_n\}$ of a sphere center hypothesis $\mathbf{p}$ is given by $R = \|\mathbf{p} - \mathbf{q}\|$.

To classify the quadric associated with an axis L, we transform the problem from one of classifying 3D quadric surfaces of revolution into one of classifying conic sections. That is, we determine which 2D curve is revolving about L to form the solid of revolution. For each range point in $\{\mathbf{q}_n\}$ of an axis, L, there exists a set $\{\mathbf{r}_j\}$ of $N \times M$ range points that is used to compute the smooth surface approximation. Using the line L parameterized by the orientation vector $\mathbf{w}$ and location vector $\mathbf{p}$, these points can be mapped into a set of 2D points $\mathbf{s}_j = \{(\upsilon_j, \omega_j)\}$ by setting

$$\upsilon_j = (\mathbf{r}_j - \mathbf{p})^t \mathbf{w}, \quad \omega_j = (\mathbf{r}_j - \mathbf{p})^t (\mathbf{I} - \mathbf{w}\mathbf{w}^t) (\mathbf{r}_j - \mathbf{p}) . \quad (4.1)$$

This "unwrapping" process is illustrated in Fig. 3. Standard linear regression techniques are applied to fit a 1D quadric function $\omega(\upsilon) = \beta_0 + \beta_1 \upsilon + \beta_2 \upsilon^2$ to the unwrapped points $\{\mathbf{s}_j\}$. The coefficients β_0, β_1, and β_2 of this equation are used to estimate the parameters of the revolving curves, see [Sabbah *et al.*, 1986, Bolle *et al.*, 1987].

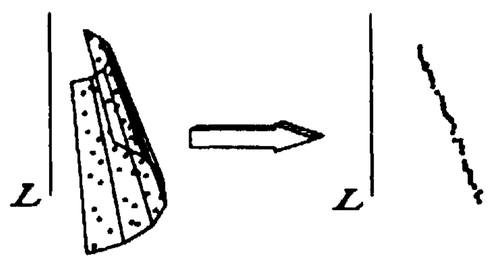

Fig. 3: Unwrapping points

5. Object recognition

Features reconstructed in lower-level parameter spaces are combined into *assembly hypotheses* in the object sub-network which represent complete or partial 3D objects. The parameter transform into this level identifies which assembly hypotheses a feature should support. Support links are created from features to assemblies, inhibition links are identified between assembly hypotheses and iterative refinement identifies the best alternatives. Hence, the homogeneous approach of the system is maintained for object recognition as well.

The parameter transform from the feature spaces to the object space can be thought of as *indexing* into a database of object models with a feature, to determine which assembly hypotheses it suggests. This involves *matching* the image features to features of object models on the basis of intrinsic characteristics (e.g., surface type) and relative position. When an image feature matches a model feature, we create a object hypothesis containing a *binding* between the two.

Our system architecture provides several advantages which allow us to streamline the indexing task. The goal of indexing is now simply to bind an image feature to every possible model feature. The correct hypotheses will collect evidence from more features than their competitors and will survive evidence integration. In reality we must do some pruning in order not to overload the network. Most hypotheses, however, have very little evidence in their favor. Because eliminating highly unlikely alternatives is far easier than distinguishing between the best alternatives, the simple techniques described later can do sufficient pruning without putting to great a burden on the transform.

Additionally, we have no need to determine which features belong to a single object *before* recognition. Just as the refinement step provides an implicit segmentation of the image during feature extraction, it also provides a partitioning of the features during object recognition. We specify that hypotheses which share support from common primitives compete, thus assembly hypotheses which survive evidence integration will not share primitives, and a partition on the

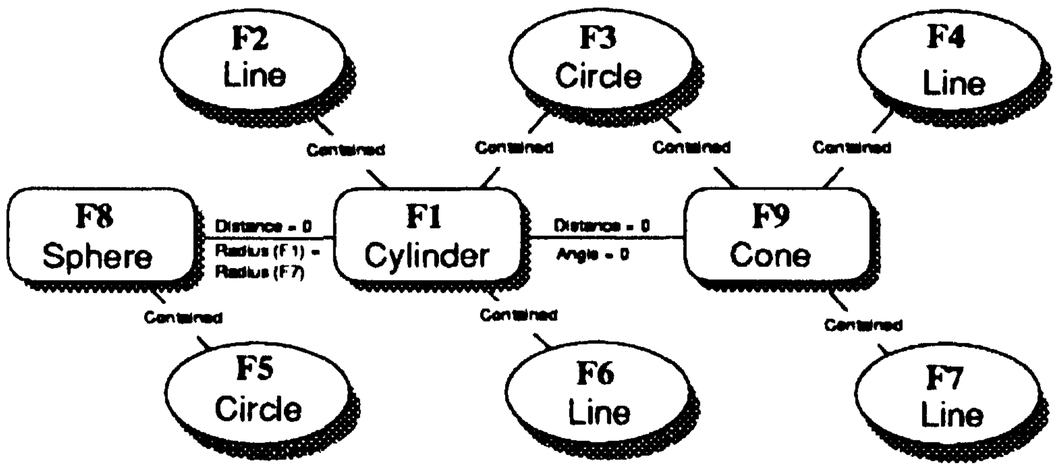

Fig. 4: Feature graph of object model

feature set is created. This avoids a major stumbling block of many systems [Binford, 1982, Knapman, 1987].

5.1. Object models

Object models in the database are represented by a feature graph (see, for example, Fig. 4). Nodes F in the graph represent the primitive features of the object; surfaces and 3D curves. Arcs R_{ij} represent coordinate-free geometric relationships between features F_i and F_j. For example, R_{18} in Fig. 4 indicates F_1 and F_8 are related by the distance from the center of the sphere to the axis of the cylinder, and by the relative size of their radii. We have defined a set of possible relationships between any pair of feature types which may be used in a graph.

Features of a model are sorted into layers, organized according to the resolution at which we expect them to be reconstructed. The first layers contain those features likely to be found at a coarse resolution and successively later layers contain progressively finer features. This "multiple resolution" representation prunes the search for matching features as will be described below.

5.2. Object hypotheses

Each hypothesis in the object parameter space represents an instantiation of an object model from the database. It is identified by the set of bindings $\{B(F_i, f_j), ...\}$ between features F_i of the object model and features f_j found in the image.

5.3. Indexing

Indexing consists of two steps, checking for a match with unbound model features of existing object hypotheses and checking features of models in the database. If a match is found we create a new hypothesis, or extend an existing one, to include the new evidence.

In either case, we only need to check for matches between image features and features in *active* layers of an object model. For uninstantiated models in the database, only the first (coarsest resolution) layer is active. Instances of models in object hypotheses can have one or more active layers. Layers are activated whenever sufficient features of the previous layer have been bound to image features. To avoid an explosion in the number of hypotheses with very little evidence, only models with a sufficient percentage of matched features in the first layer are instantiated.

5.4. Feature matching

Matching an image feature to a feature in a model requires checking two pieces of information, intrinsic feature characteristics (e.g., feature type) and position relative to other features. Since a feature's characteristics are implicit in its parameterization, the first is simply a comparison with the properties specified for the feature in the model. Checking the relative position or size of image features, on the other hand, requires a fair amount of computation and is much more time consuming.

The primary feature characteristic is feature type. The amount of pruning done by feature type search depends on the number of different types [Ettinger, 1988]. Many systems have used a relatively small set, often just curves and junctions (e.g., [Shapiro *et al.*, 1977, Sugihara, 1979]). We use a richer set (six different feature types and the potential for many more), which greatly improves the pruning ability of this step. Following [Ettinger, 1988], assuming that the features are equally distributed among the feature types, then given a typical scene and our present model base, with six feature types we will explore .02% (1/5000) of the search space that would be required with two.

6. Experiments

Experiments have been run on some twenty images of varying complexity. Early images were artificially generated and relatively simple (see, e.g., [Bolle *et al.*, 1987]). More recently, we have been using images generated with a laser range finder [TAC, 1986]. Images have varied from 32×32 to 256×256.

We present an experiment using a 64×64 depth map generated from the scene in Fig. 5. It contains four simple objects; a pencil sharpener, a battery, a tape box, and a half golf ball. One end of the pencil sharpener and the golf ball are resting on the box. Figure 6 shows the depth map generated from the scene. Shadows are present behind each object due to the range finding technique used (triangulation). These are most noticeable behind the sphere. In these regions, we have no depth information at all.

Low-level processing includes finding the zero and first-order discontinuities in the original data, and computing quadric smooth surface approximations about each point. With this size image, these take about two minutes on a Symbolics 3650.

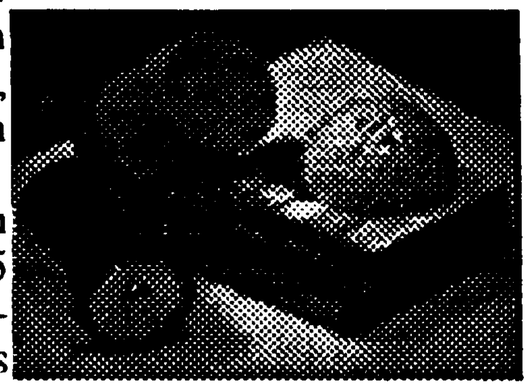

Fig. 5: Real world scene.

Feature extraction takes on the order of 12 iterations and 15 minutes real time. Most surface and discontinuity features present in the image were successfully reconstructed. As an example, 17 axis orientation

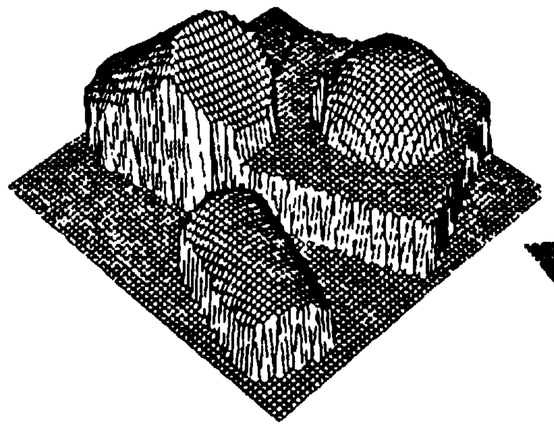

Fig. 6: Depth map.

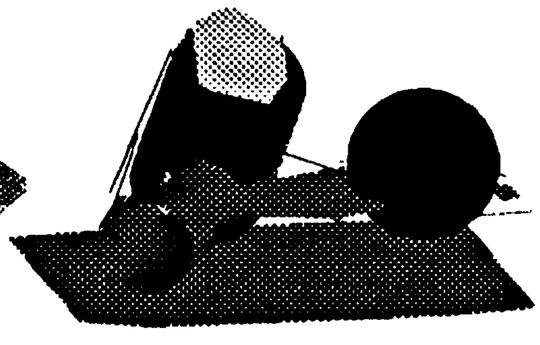

Fig. 7: Reconstructed surface features.

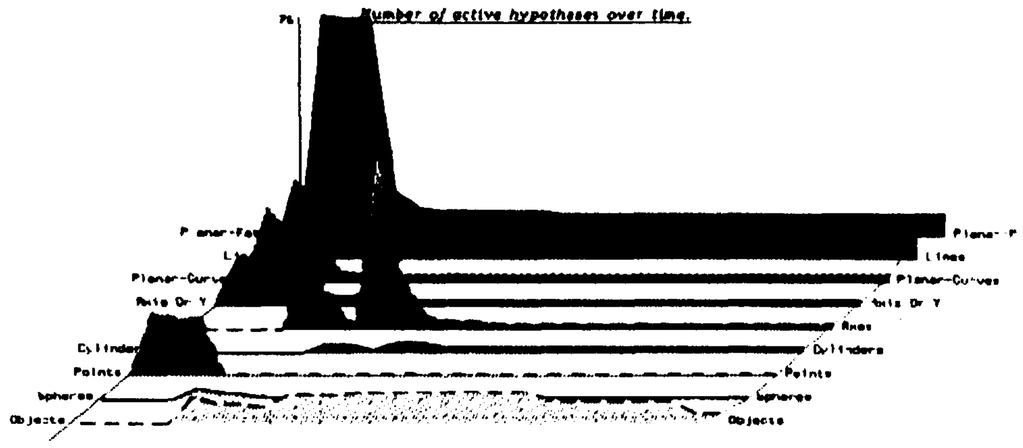

Fig. 8: Number of active hypotheses over time.

hypotheses (partial features of axes of quadrics of revolution) were generated in the first-level quadric of revolution reconstruction process. Of these two survived and generated 89 hypotheses for axes of revolution. Only the two correct axes survived to reconstruct the solids of revolution, i.e. hypotheses for cylinders and cones. Evidence integration causes a dramatic implosion in the number of hypotheses, as can be seen in Fig. 8, which shows the number of hypotheses in each parameter space over time.

Figure 7 shows the results of feature reconstruction. To generate this figure we gave the parameters of the surviving primitive features to a CAD system.

For this experiment, there were 11 object models in the database, each containing from two to 20 features (averaging 12) and from one to 110 relationships (averaging 34). Model features were divided into two layers of resolution (see Sect. 5). The models of the objects in the image were a simple sphere (two features, spherical surface and a circular limb), a cylinder segment (7 features and 8 relationships), and a box (18 features and 63 relationships). Other models in the database included, for example, a bottle, an L-brackct and two different computer mice.

The four objects were all successfully identified. The large cylinder segment was identified on the basis of five features, the cylindrical surface, the bounding end plane, the limbs and the circle formed by the intersection of the bounding plane and the cylinder. The smaller cylinder, sphere and box hypotheses were similarly bound to all their reconstructed features. The total indexing time for all features was on the order of four minutes.

In an attempt to investigate performance degradation with noise, we incrementally added gaussian noise to the image of the previous experiment and, without changing any control parameters, observed the behavior of the system.

Performance degraded gracefully. Two general trends were observed: some features were no longer recognized; and the parameters of reconstructed features started to vary, causing errors in object matching. At high noise levels, some erroneous features were also detected. With white Gaussian noise of standard deviation $\sigma = 0.3$ pixels (significant when compaired to the dynamic range within a feature estimation window - typically 3-4 pixels) we were no longer able to recognize the surface of the small cylinder. All other features were reconstructed and three of the four objects were recognized. With noise of standard deviation $a = 0.4$ all but two features were still reconstructed, however the parameters of some were inaccurate. In spite of this, three of the four objects were still recognized, though the hypotheses had fewer image-to-object feature bindings. With noise of standard deviation $a = 0.6$ pixels (Fig. 9) some recognition was still possible. The cylinder surfaces were no longer recognized, some object limb and the conies disappeared. However the planar and spherical surfaces and most of the line discontinuities were still reconstructed with accurate parameters. The box and the sphere were still recognized as objects.

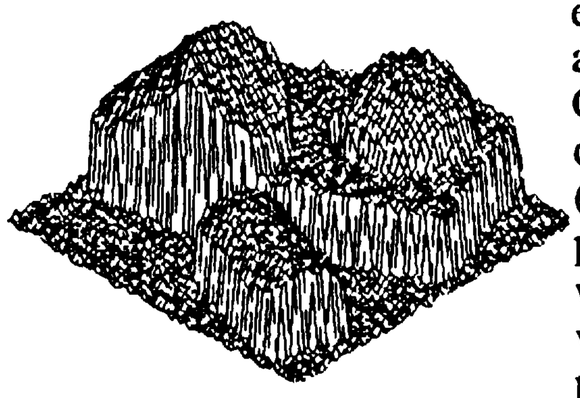

Fig. 9: Degraded depth map.

The performance at high noise levels could have been improved substantially if we had adjusted the control parameters to compensate. This points out the need for automatic adjustment of system parameters on the basis of image characteristics.

7. Conclusion

The vision framework presented here, while certainly not providing a definitive solution to "the vision problem," does successfully address the critical problems mentioned in the introduction. We have defined a consistent representation for hypotheses where the evidence for each is made explicit. We have provided a mechanism to combine this evidence into a globally consistent interpretation. At the same time we have tried to put as few limitations as possible on the feature extraction mechanisms, allowing them to vary (as they necessarily must) during the course of the recognition process. The result is a powerful approach, which as demonstrated, is capable of supporting reasonably complex recognition tasks. This approach helps alleviate several problems which have hindered vision researchers in the past, including segmentation, evidence integration, expendability, global consistency and hypothesis evaluation/comparison.

Future work will focus on several areas. First is to continue work within the framework itself. This will include a more careful analysis of the dynamics of the network including analysis of network stability, especially in the presence of feedback. Some current work focuses on dynamic adjustment of important variables including parameter space quantization. Other areas will use the framework to explore topics concerning the recognition process itself. These include more powerful object modeling and indexing techniques, extensions of the multi-window approach and exploration of multi-resolution recognition. One particularly promising approach made easier by the architecture is in the area of sensor fusion. Because hypotheses can receive support from features in varied parameter spaces, we will be able to combine evidence from features extracted from different input sources, thus providing a vehicle for sensor fusion. All of this work will be made easier by the consistent framework we have described.

References

[Ballard, 1981] D.H. Ballard, "Parameter nets: A theory of low level vision," in *Proc. 7th Int. Joint Conf. On Artificial Intell., Aug.* 1981, pp. 1068-1078.

[Besl & Jain, 1985] P.J. Besl and R.C. Jain, "Three-dimensional object recognition," Computing Surveys, Vol. 17, No. 1, March 1985, pp. 75-145.

[Binford, 1982] T.O. Binford, "Survey of Model-Based Image Analysis Systems," *The Int. Journal of Robotics Research,* Vol. 1, No. 1, Spring 1982, pp. 18-64.

[Bolle *et al.,* 1986] R.M. Bolle, D.B. Cooper, "On Optimally Combining Pieces of Information, with Application to Estimating 3-D Complex-Object Position from Range Data," *IEEE Trans, on Pattern Analysis and Machine Intell.,* Vol. 8, No. 5, September 1986, pp. 619-638.

[Bolle *et al,* 1987] R.M. Bolle, R. Kjeldsen, and D. Sabbah, "Primitive shape extraction from range data," in *Proc. IEEE Workshop on Comp. Vision,* Nov.-Dec. 1987, pp. 324-326; also *IBM Tech. Rep. RC12392,* AI Systems Group, IBM T.J. Watson Res. Center, July 1987.

[Bolles *et al.,* 1983] R.C. Bolles, P. Horaud, and M.J. Hannah, "3DPO: A three-dimensional part orientation system," in *Proc. 8th Int. Joint Conf. on Artificial Intell.,* Aug. 1983, pp. 1116-1120.

[Bookstein, 1979] F.L. Bookstein, "Fitting conic sections to scattered data," *Comp. Graphics and Image Processing,* Vol.9, No. 1, Jan. 1979, pp.56-71.

[Brooks, 1983] R.A. Brooks, "Model-based three-dimensional interpretations of two-dimensional images," *IEEE Trans, on Pattern Analysis and Machine Intell.,* Vol. 5, No. 2, March 1983, pp. 140-150.

[Califano, 1988] A. Califano, "Feature recognition using correlated information contained in multiple neighborhoods," in *Proc. 7th Nat. Conf on Artificial Intell.,* July 1988, pp. 831-836.

[Califano *et al.,* 1988] A. Califano, R.M. Bolle, and R.W. Taylor, "Generalized neighborhoods: A new approach to complex feature extraction," submitted to *IEEE Conf. on Comp. Vision and Pattern Recognition,* Nov. 1988.

[DoCarmo, 1976] M.P. DoCarmo, *Differential geometry of curves and surfaces.* New Jersey: Prentice-Hall, 1976.

[Ettinger, 1988] G J Ettinger, "Large hierarchical object recognition using libraries of parameterized model subparts," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition,* June 1988, pp 32-41.

[Feldman & Ballard, 1981] J.A. Feldman and D.H. Ballard, "Connectionist models and their properties," *Cognitive Science,* Vol. 6,1981, pp. 205-254.

[Hakala *et al.,* 1980] D.G. Hakala, R.C. Hillyard, P.F. Malraison, and B.F. Nource, "Natural quadrics in mechanical design," in *Proc. CAD/CAM VII,* pp. 363-378.

[Hough, 1962] P.V.C. Hough, *Methods and Means for Recognizing Complex Patterns,* U.S. Patent 3069654, 1962.

[Knapman, 1987] J. Knapman, "3D model identification from stereo data," in *Proc. First Int. Conf. on Computer Vision,* June 1987, pp. 547-551

[Rosenfeld & Kak, 1982] A. Rosenfeld and A.C. Kak, *Digital Picture Processing.* New York: Academic Press, 1982.

[Sabbah, 1985] D. Sabbah, "Computing with connections in visual recognition of origami objects," *Cognitive Science,* Vol. 9, No. 1, Jan.-March 1985, pp. 25-50.

[Sabbah *et al.,* 1986] D. Sabbah and R.M. Bolle, "Extraction of surface parameters from depth maps viewing planes and quadrics of revolution," in *Proc. SPIE Conf. Intell. Robots and Comp. Vision,* Oct. 1986, pp. 222-232.

[Shapiro *et al.,* 1977] R. Shapiro and H. Freeman, "Reconstruction of curved-surface bodies from a set of imperfect projections," in *Proc. 5th Int. Joint Conf on Artificial Intell.,* Aug. 1977, pp. 22-26.

[Sugihara, 1979] K. Sugihara, "Range-data analysis guided by junction directory," *Artificial Intelligence,* Vol. 12, No. 1,1979, pp 41-69.

[TAC, 1986] Technical Arts Corporation, *100X 3D Scanner: User's manual & application programming guide.* Redmond, W A: 1986.

[Weems *et al.,* 1989] C. C. Weems, S. P. Levitan, A.R. Hanson and E.M. Riseman, "The Image Understanding Architecture," *Intl. Jou. of Computer Vision,* Vol. 2, No. 3,1989, pp 251-282.