Proceedings of SPIE—The International Society for Optical Engineering

Volume 848

Intelligent Robots and Computer Vision

Sixth in a Series

David P. Casasent, Ernest L. Hall
Chairs/Editors

Sponsored by
SPIE—The International Society for Optical Engineering
in association with
IEEE Industrial Electronics Society
Society of Instrument and Control Engineers of Japan
Cooperating Organizations
Center for Optical Data Processing/Carnegie Mellon University
Electro-Optics Technology Center/Tufts University
IEEE Robotics and Automation Council
Sira Ltd.—The Research Association for Instrumentation (UK)

2-6 November 1987
Cambridge, Massachusetts

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA
Telephone 206/676-3290 (Pacific Time) • Telex 46-7053

Model-based Object Recognition Using the Connection Machine

Lewis W. Tucker
Carl R. Feynman, Mike Drumheller, Donna M. Fritzsche, and David Waltz
Thinking Machines Corporation
Cambridge, Massachusetts

Abstract. This paper reports on a model-based object recognition system and its parallel implementation on the Connection Machine[1] System. The goal is to be able to recognize a large number of partially occluded, two-dimensional objects in scenes of moderate complexity. In contrast to traditional approaches, the system described here uses a parallel hypothesize and test method that avoids serial search.

The basis for hypothesis generation is provided by local boundary features (such as corners formed by intersecting line segments) that constrain an object's position and orientation. Once generated, hypothetical instances of models are either accepted or rejected by a verification process that computes each instance's overall confidence.

Even on a massively parallel computer, however, the potential for combinatorial explosion of hypotheses is still of major concern when the number of objects and models becomes large. We control this explosion by accumulating weak evidence in the form of votes in position and orientation space cast by each hypothesis. The density of votes in parameter space is expected to be proportional to the degree to which hypotheses receive support from different local features. Thus, it becomes possible to rank hypotheses prior to verification and test more likely hypotheses first.

1 Introduction

The goal of model-based object recognition is to find instances of known objects in novel scenes. This is an important problem, but current solutions do not approach human performance and thus have limited applications.

Massively parallel architectures improve computer performance by several orders of magnitude. The purpose of this investigation is to examine ways in which this parallelism may be applied to the problem of object recognition.

1.1 Background

Traditionally, the simplest approach to object recognition has relied on classification techniques based on global measures such as an object's area, perimeter, color, and other characteristics of the object as a whole. While it may be possible to employ global measures in controlled environments, segmentation of an image into regions corresponding to objects to be recognized is in general a very difficult task and these measures may not be applicable when objects touch or partially occlude each other.

Local features, such as boundary segments, do not suffer from this problem, and their spatial relationships provide a major source of information for recognizing objects in complex scenes. Thus, the problem may be posed as one of constraint satisfaction — given a description of the local features of each model, find the parameters of a spatial transformation for the model for the "best" pairing of features in the image with those "expected" by each model. Constraints provided by one feature pairing are propagated throughout each model in the course of searching for a solution.

Finding this transformation is a difficult combinatorial problem that grows exponentially with the number of image and model features. Consequently, tree-based search techniques have been employed in an attempt to limit the number of potential solutions that must be explored. Constraints provided by the spatial relationships between features are used to prune entire subtrees from consideration [Grimson and Lozano-Perez, 1987]. Implausible solutions lead to backtracking and further search. However, for any significant number of features, it is still necessary to explore a large number of solutions.

Other approaches concentrate on the accumulation of weak evidence supporting a spatial transformation for each model. Ballard's [1981] use of a Hough transform for an arbitrary shape is one such example. Problems with Hough transforms include the often very large memory requirement for multi-dimensional accumulator arrays and the confusion caused by false peaks.

A third approach relies on perceptual groupings and the viewpoint consistency constraint [Lowe 1986*a*]. Image features are grouped according to proximity, parallelism, and colinearity and used to predict the transformation required to bring the model into alignment with its corresponding features. Lowe [1986*b*] has used this technique to recognize 3-D objects using only their 2-D projections. In a similar manner, salient features such as characteristic boundary segments have been used to generate either hypotheses to be tested or votes for the position of each matching object [Turney et al. 1985].

[1] Connection Machine is a registered trademark of Thinking Machines Corporation.

In general, these previous approaches have treated each model-object in turn. As a consequence, the time to recognize objects in a scene grows linearly with the size of the model database.

In this paper, we present a new, highly parallel approach to recognition. The goal is to design a system capable of recognizing objects in scenes given a database of known object models.

2 Parallel Object Recognition

Three principal problems facing object-recognition are addressed by the system described here. The first is the issue of how object recognition may be performed on a massively parallel system. The second concerns overcoming the combinatoric problem associated with establishing correspondence between image and model features. The third problem addressed concerns how the recognition system scales when the number of known models increases. Does the time to recognize objects in a scene increase with the number of models in the database? Is the recognition hindered as the size of the model database is increased?

2.1 Hypothesis Generation and Test

We approach the problem as being one of hypothesis generation and test. Image features in the scene serve as events while features of each model function as expectations waiting to be satisfied. Hypotheses arise whenever an expectation is satisfied by an event. Following a principle of least commitment, these hypotheses may be based on very weak evidence provided they are subject to verification prior to acceptance.

Parallel hypothesis generation and test may be briefly described as follows: Image features are extracted from the scene and broadcast one at a time. All models, in parallel, determine if the broadcast image feature event corresponds to one or more of their feature expectations. Wherever a match between an event and expectation is detected, a hypothesis is generated which states that a given model exists at a specified location and orientation. Once all image features are processed, hypotheses are verified in parallel. This verification takes the form of creating an instance of the model according to the parameters of a spatial transformation specified by each hypothesis. In a sense, a match between a feature in a model and a corresponding feature in image, creates a hypothesis that binds an instance of the model to a specific location and orientation in the scene. This binding projects all features of each model to specific locations and orientations in the scene. Verification may then be performed for all instances in parallel by simply having each instance feature check an expected location in the scene. Competitive matching between instances for each image feature resolves any conflicts that might arise between hypotheses.

For parallel hypothesis generation and test to be effective, however, a number of conditions must be met:

- The total number of features in the model database must be less than the number of processors. Using the Connection Machine System, it is possible to think of operating on databases of several thousand objects.
- Features used in hypothesis generation must completely constrain the parameters of a model-to-scene coordinate system transformation. In the 2-D system described here, the match between an image corner and an object corner is sufficient to describe a hypothetical translation and rotation which brings the corresponding corners into alignment.
- For an object to be hypothesized and subsequently identified, at least one of these constraining features must be detectable in the scene. Since an object with k corners has k opportunities to be recognized, this is not considered to be an unreasonable restriction.

2.2 Object Domain and Feature Extraction

The domain chosen for study is that of 2-D objects of fixed scale. Objects in the scene may touch or partially occlude each other and no assumptions are made regarding the completeness of each object's border. In this initial study, up to 100 arbitrarily chosen objects make up the model database. From 1-10 objects will be in each scene. A typical scene is given in Figure 1a.

Local boundary segments form the primary basis for matching and hypothesis generation. A Canny edge detector [Canny 1986], adapted for parallel execution, is used to extract and label edge points. A parallel curve decomposition and least-squares approximation algorithm is used to segment connected edge components into simple line segments. Corner features are generated wherever line segments intersect within some distance of their endpoints. These line segments and corners make up the feature set employed by the current system (Figure 1b).

The number of processors available for processing these images is equivalent to the number of pixels in the scene (256x256 in this example).

2.3 Architecture of the Connection Machine

The Connection Machine System (CM), with its large number of processing elements, supports this parallel approach and proved to be a major influence on many aspects of the overall system. It is a massively parallel architecture having up to 65,536 physical processors and supported by a general interprocessor communications network. Each processor has 64K bits of local storage, sufficient for storing the data structures required by the system. The software environment of the Connection Machine supports several high-level programming languages with parallel instruction extensions, including C*, *LISP, and FORTRAN. The system described here was written in *LISP, a parallel extension of Common Lisp. Operating system support for program editors, file system access, and program execution was provided by a Symbolics Lisp Machine front-end. The use of a familiar software environment is a major factor contributing to the ease with which the Connection Machine is operated.

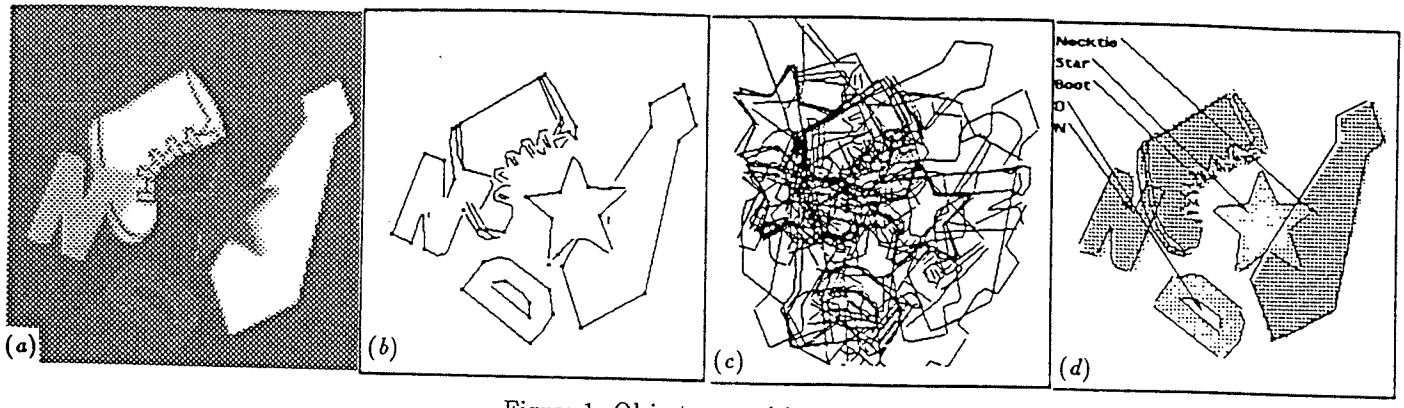

Figure 1: Object recognition steps.

(*a*) Typical scene containing five objects to be identified. (*b*) Straight line segments fit to intensity edges, and corners found at intersecting segments form the primary feature representation of models and scenes. Note that boundaries are not necessarily complete. (*c*) Several hypotheses generated for the scene given in Figure *a*. Each hypothesis is generated in response to a single match between an image and model corner feature. (*d*) Final results: Five objects are identified and located by the recognition system.

An additional feature relevant to this application is the support of a virtual processor mechanism. Transparent to the user, this mechanism permits configuration of machines with many more virtual than physical processors. In the system described here, the typical machine configuration utilized 64K virtual processors executing on actual hardware having either 8K or 16K physical processors.

A complete description of the Connection Machine architecture is beyond the scope of this paper and the reader is referred to [Hillis 1987, TMC 1987] for further details.

2.4 Model Representation

Since the primary basis for generating hypotheses is the determination of a match between image and model features, models are represented simply as a set of features, assigned one per processor (Figure 2). A model is therefore represented by a set of processors. These features are derived from a single image of the object in isolation. This representation was chosen over a single processor per model representation so that each feature could in parallel, independently participate in matching and verification. In addition, since the the number of processors devoted to each model is directly related to the complexity of the object, the time to perform model-based operations is the same for all objects.

Spatial relationships between a model's features are defined by their position and orientation in an object-centered coordinate system.

Instances of models likewise utilize this processor set representation. The difference between an instance and a model is that in each instance, the location and orientation of features is given in the image rather than model-centered coordinate frame. Thus verification is simply a matter of having each instance feature seek corresponding image features in predicted locations in the scene.

2.5 Hypothesis Generation and Instantiation

As used in the context of this paper, an object hypothesis is defined to be a conjecture that an instance of a named model exists at a given location and orientation in the input scene. As noted above, a hypothesis arises whenever a "good" match is found between image and model features that have sufficient information content. In two dimensions, the intersection of two line segments (i.e., an image corner) is one such feature. Each corner is described by a vertex location, included angle, and orientation of the angle's bisector. The match between an expected and observed image corner is sufficient to define a spatial transformation that would bring an instance of the model into alignment with the image corner.

This points out the importance of perceptual organization. "Higher-level", composite features such as "image corners" are not only more selective in matching appropriate models than simple features in a database of objects, but they also provide additional constraints sufficient for predicting position and orientation. The number of these composite features based on connectivity or proximity increases only linearly with the number of simple features and therefore does not significantly increase the amount of information that must be processed.

However, as is shown in Figure 1c, the use of such weak evidence is expected to generate a large number of hypotheses - most irrelevant and some that are nearly identical. Although each hypothesis is a simple data structure (containing only a model-id and parameters specifying a position and rotation), the processor requirement for instantiation places a limit on how many hypotheses can be tested at a time. Testing a hypothesis requires generation of a model instance translated and rotated in image space. Since each instance is a copy of an object model they typically require between 20 and 30 processors to represent the features of each model. Given

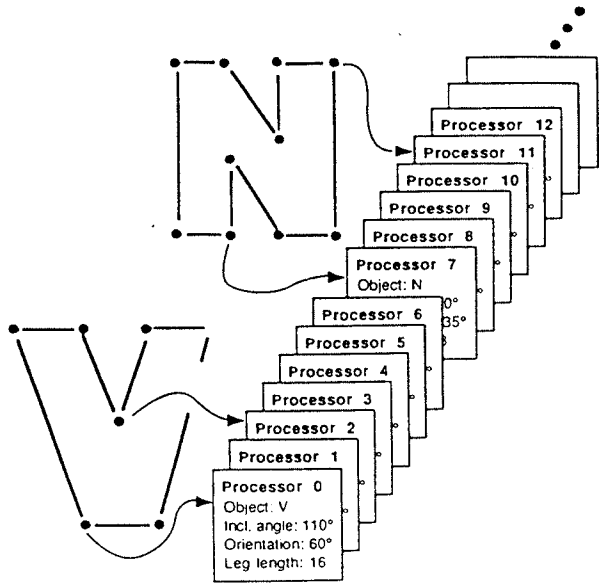

Figure 2: Object representation as a set of features assigned one per processor

a machine size of 64k processors, between 2,000 and 3,000 instances are tested as a group. If the correct solution is not found at first, incorrect hypotheses are eliminated and the next block of hypotheses is instantiated until the solution is found. Thus, several iterations of instantiation and testing might be required if all hypotheses are to be examined.

2.6 Scene Analysis and Hypothesis Acceptance

As previously described, verification of a hypothesis may be considered to be primarily a process of template matching. All features of an instance reference specific locations in the scene to find corresponding image features. A "goodness of fit" measure is assigned to the instance as a whole by computing the average score of its individual feature matches.

If a single object is to be identified, it would be a simple matter to pick the instance showing the best "goodness-of-fit" score. When several objects appear in a scene, however, it is nearly impossible to accept hypotheses that simply pass a threshold criterion since there may be conflicts which lead to implausible solutions. These conflicts arise from the fact that in most cases, multiple hypotheses are generated for each image feature. Since accidental alignment of object features is considered rare, it is reasonable to restrict the assignment of each image feature to that hypotheses having the greatest overall confidence. A procedure is therefore employed which enforces a unique assignment of features to objects. Once the set of hypotheses is consistent with this restriction, individual hypotheses are accepted provided they exceed some threshold confidence level (see Figure 1d).

In the current system, no attempt has made to explicitly detect and account for occlusions. Thus, it is expected that there will be some difficulty distinguishing between heavily occluded and poorly matched objects.

2.7 Hypothesis Ranking by Accumulation of Mutual Support

As previously described, sets of hypotheses are instantiated and tested in turn until the "correct" solutions have been found. If it is possible to define a procedure less computationally expensive than instantiation which is capable of producing an ordering of hypotheses, the total cost can be reduced. Ideally, this ordering would place all "correct" solutions within the first instantiated set. (The actual ordering within the set is inconsequential since all instances may be verified in parallel.)

The following observation led to the selection of this ordering function. In general, each model has several features that serve as sources for hypotheses and this leads to the generation of several, nearly identical hypotheses. The degree of redundancy is indicative of the amount of support offered a particular hypothesis from independently concurring hypotheses.

Mutual support may be determined using a voting scheme in parameter space in a manner related to the Hough transform. Hypotheses pertaining to each model object cast votes in an accumulator in this space for their predicted position and orientation. Clusters in parameter space correspond to mutually supporting hypotheses. Prior to instantiation, hypotheses can then be ranked according to the amount of support indicated by a local density estimate of the region surrounding each hypothesis.

Cluster formation in high dimensional spaces, however, is expensive both in memory and time, particularly if we wish to do this for all models at the same time. Therefore three projections along x, y, and orientation dimensions in the form of simple 1-D histograms, are calculated and the final density score for each hypothesis is estimated by forming the product of the local density in each projection. In this way, only hypotheses having support in both position and orientation receive the top scores. In order to compare scores between models, each histogram is normalized by the number of expected features of each model.

3 Experimental Results

Experiments were run using simple objects from the following set:

1. 26 plastic block letters and numbers
2. 40 cutout cardboard figures
3. 34 common household or office items (staplers, pens, etc)

Models were created by digitizing each object in turn, and extracting the significant features to form the descriptions of each object. Supporting software provided for the saving and loading of model descriptions to and from disk. Since models were automatically generated from actual image data, it was expected that there would be a certain degree of associated error with each model.

Ten objects were selected from this set and five test scenes each containing five objects were composed. The example given in Figure 1 is one such test scene. As shown, objects were permitted to touch but occlusion was purposefully restricted in order to avoid problems associated with template-matching verification schemes.

Object databases were formed from this set of 100 objects by randomly selecting objects to be added to the initial object set such that databases having 10, 30, 50, 70 and 100 objects were created. For each database, the five test scenes were examined by the system and results tabulated. In all cases, the system correctly located and identified all objects in each scene.

Table 1 tabulates for each database the number of hypotheses, time to completion, and the best, worst, and median rank of the hypotheses that led to a correct identification. Figure 3 shows how the rankings for a typical scene changed with increasing the size of the model database.

As expected, the number of hypotheses generated increased linearly with the size of the database, from a minimum of 1,519 for the 10-object database to 17,877 for the 100-object database.

The ranking data indicates the effectiveness of the evidence accumulation and clustering in ordering hypotheses prior to instantiation. It should be noted that the rank of a given hypothesis shows how large the instantiated set must be to include the correct solution.

The worst rank-score of a correct hypothesis was 2,326. Thus in all cases tested, the correct interpretation was always in the first instantiated set.

The median rank increased at a slower rate than the increase in the size of the database from a rank of 73 (10-object database) to 451 (100-object database). The best rank increased only slightly from 4 to 14 as the size of the database increased from 10 to 100 model objects.

The average time to identify all five objects in each scene, given a 100-object database was 48 seconds using a 8,192 processor CM-2. This time included all aspects of the system from image acquisition and feature extraction, through hypothesis instantiation and object identification. The image resolution used was 256x256 pixels, which required the Connection Machine to be configured with a virtual processor ration of 8:1. Consequently, execution of this test on a full 65,536 processor Connection Machine is estimated to be in the range of 5-10 seconds.

DB size (objects)	Number of Hypotheses	Median Rank	Best Rank	Worst Rank	Time (secs)
10	1519	62	4	183	47
30	4506	171	8	761	49
50	8362	219	10	1083	52
70	11194	312	12	1557	52
100	17877	451	14	2326	52

Table 1: Results from databases containing between 10 and 100 objects. Tests were run on an 8K-processor CM2 with a virtual processor ratio of 8:1.

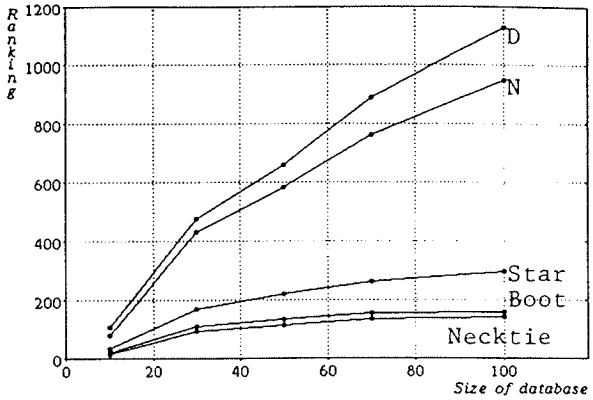

Figure 3: Rank of hypotheses that led to correct interpretations for scene in Figure 1 over a range of database sizes

4 Discussion

In this paper, we have demonstrated a parallel, 2-D object recognition system capable of recognizing objects selected from a database of up to 100 objects. The system is based on hypothesis generation and a cluster detection in parameter space. Crucial to the design of the system is the representation of a model object as a collection of features assigned one-per-processor. This permits simultaneous matching and verification of model features and provides the basis for hypothesis generation.

In our experience the system reliably recognizes objects in novel scenes and performance degrades gracefully in the presence of noise and significant occlusions. In cases where the system fails to find the correct interpretation, the problem is usually traceable to one of the following sources of error. Simple template matching is used for verification. The potential therefore exists for finding an accidental alignment of features from several objects that match well with a model's template. Another source of confusion arises whenever the borders of different models largely superimpose. Finally, since the system is designed to find "acceptable" interpretations before exploring all possibilities, the discovery of an acceptable but non-optimal solution may preclude the discovery of the correct solution. These errors are rarely observed in simple scenes and no attempt to directly address the problems associated with occluding objects has been made in this study.

One of the goals of this system was to investigate how an increase in database size effects object recognition. For small databases (up to 20 objects) it was feasible to instantiate and verify all hypotheses. However, as the size of the object database increased, it became necessary to order hypotheses so that the more likely hypotheses were tested first. Hypotheses were ordered by the degree of mutual support each received as indicated by a parameter space clustering technique.

From our experience with this system, a number of observations can be made. The number of hypotheses and the size of the instantiated set containing the correct solution appears to increase linearly with the number of objects in the database. Ranking of hypotheses proved to be effective in

pruning the set of hypotheses such that 86% were eliminated prior to instantiation and verification. Hypotheses leading to the correct interpretation for over half of the objects in each scene proved to be within the top 4% of this ordered set. These results demonstrate the strength of this parameter space clustering technique.

Parallel hypothesis generation and parameter space clustering together together provide an efficient alternative to serial search and may form a uniform framework for combining evidence from multiple features. We are currently extending the system to recognize curved objects by using extrema of curvature. It is felt that additional, more robust features can only improve performance and permit even larger object databases to be examined.

5 Conclusion

This paper has attempted to show how object recognition may be accomplished in parallel using the Connection Machine System. With this massively parallel architecture, some old ideas (hypothesis generation and test) have been given new meaning, and we may begin to explore the larger problems associated with object recognition, knowledge representation, and recall from memory.

6 Acknowledgements

We would like to thank Todd Cass for implementing Canny's edge detector on the Connection Machine; Tommy Poggio, for thoughtful discussions and encouragement; and Harry Voorhees, for assistance in preparing this manuscript.

7 References

Ballard D. H., "Generalizing the Hough Transfrom to Detect Arbitrary Shapes," *Pattern Recognition*, Vol. 13, pp. 111–122, 1981.

Canny, John F., "A Computational Approach to Edge Detection", IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), Vol. PAMI-8, No. 6, pp. 679–698, Nov. 1986.

Hillis, W. Daniel, "The Connection Machine," *Scientific American*, Vol. 256, pp. 108-115, June 1987.

Lowe, D. G., "The Viewpoint Consistency Constraint," New York University Technical Report No. 244, Sept. 1986*a*.

Lowe, D. G., "Three-Dimensional Object Recognition from Single Two-Dimensional Images," *Artificial Intelligence*, Vol. 31, pp. 355–395, 1987).

Grimson, W. Eric L. and Tomas Lozano-Perez, "Localizing Overlapping Parts by Searching the Interpretation Tree," *IEEE Trans. PAMI*, Vol. PAMI-9, No. 4, pp. 429–482, July 1987.

Thinking Machines Corp., "Connection Machine Model CM-2 Technical Summary," Tech. Report HA87-4, April 1987.

Turney, J. L., T. N. Mudge, and R. A. Volz, "Recognizing Partially Occluded Parts," *IEEE Trans. PAMI*, Vol. PAMI-7, No. 4, pp. 410–421, July 1985.