

Image Understanding Workshop

Proceedings of a Workshop
Held at
Los Angeles, California

February 23-25, 1987

Volume I

Sponsored by:

**Defense Advanced Research Projects Agency
Information Science and Technology Office**

This document contains copies of reports prepared for the DARPA Image Understanding Workshop. Included are results from both the basic and strategic computing programs within DARPA/ISTO sponsored projects.

**APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED**

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the Defense Advanced Research Projects Agency or the United States Government.

Distributed by
Morgan Kaufmann Publishers, Inc.
95 First Street
Los Altos, California 94022
ISBN 0-934613-36-2
Printed in the United States of America

CARLSON

..7A

1632

.I4

Feb. 1987

v.1

TABLE OF CONTENTS

	<u>Page</u>
AUTHOR INDEX	i

VOLUME I

SECTION I - PROGRAM REVIEWS BY PRINCIPAL INVESTIGATORS

"Vision in Dynamic Environments", Azriel Rosenfeld, Larry S. Davis, John (Yiannis) Aloimonos; University of Maryland	1
"USC Image Understanding Research - 1986", R. Nevatia; University of Southern California	8
"Image Understanding Research at SRI International", Martin A. Fischler and Robert C. Bolles; SRI International	12
"Image Understanding: Intelligent Systems", Thomas O. Binford; Stanford University	18
"Image Understanding Research at CMU", Takeo Kanade; Carnegie-Mellon University	32
"MIT Progress in Understanding Images", T. Poggio and the Staff; Massachusetts Institute of Technology	41
"Summary of Progress in Image Understanding at the University of Massachusetts", Allen R. Hanson and Edward M. Riseman; University of Massachusetts at Amherst	55
"Recent Progress of the Rochester Image Understanding Project", Jerome A. Feldman and Christopher M. Brown; University of Rochester	65
"Image Understanding and Robotics Research at Columbia University", John R. Kender, Peter K. Allen, Terrance E. Boulton; Columbia University	71

TABLE OF CONTENTS

	<u>Page</u>
<u>SECTION I - PROGRAM REVIEWS BY PRINCIPAL</u> <u>INVESTIGATORS (Continued)</u>	
"Developments in Knowledge-Based Vision for Obstacle Detection and Avoidance", K.E. Olin, F.M. Vilnrotter, M.J. Daily and K. Reiser; Hughes Research Laboratories ..	78
"Detecting Obstacles in Range Imagery", by Michael J. Daily, John G. Harris, and Kurt Reiser; Hughes Artificial Intelligence Center	87
"Model-Directed Object Recognition on the Connection Machine", D.W. Thompson and J.L. Mundy; General Electric Corporate Research and Development	98
"Environmental Modeling and Recognition for an Autonomous Land Vehicle", Daryl T. Lawton, Tod S. Levitt, Christopher C. McConnell, Philip C. Nelson, Jay Glicksman; Advanced Decision Systems	107
"Honeywell Progress on Knowledge-Based Robust Target Recognition & Tracking", Bir Bhanu, Durga Panda and Raj Aggarwal; Honeywell Systems & Research Center	122
<u>SECTION II - TECHNICAL REPORTS PRESENTED</u>	
"The ITA Range Image Processing System", David G. Morgenthaler, Keith D. Gremban, Mitch Nathan, John D. Bradstreet; Martin Marietta Denver Aerospace	127
"Vision and Navigation for the Carnegie Mellon Navlab", Charles Thorpe, Steven Shafer, Takeo Kanade and the members of the Strategic Computing Vision Lab, Carnegie-Mellon University	143

TABLE OF CONTENTS

	<u>Page</u>
<u>SECTION II - TECHNICAL REPORTS PRESENTED (Continued)</u>	
"Vision-Based Navigation: A Status Report", Larry S. Davis, Daniel Dementhon, Ramesh Gajulapalli, Todd R. Kushner, Jacqueline Le Moigne, Phillip Veatch; University of Maryland	153
"Information Management in a Sensor-based Autonomous System", Grahame B. Smith and Thomas M. Strat; SRI International	170
"Tools and Experiments in the Knowledge-Directed Interpretation of Road Scenes", Bruce A. Draper, Robert T. Collins, John Brolio, Joey Griffith, Allen R. Hanson, Edward M. Riseman; University of Massachusetts at Amherst	178
"Models of Errors and Mistakes in Machine Perception - Part 1. First Results for Computer Vision Range Measurements", Ruzena Bajcsy, Eric Krotkov, Max Mintz; University of Pennsylvania	194
"Automating Knowledge Acquisition for Aerial Image Interpretation", David M. McKeown, Jr., Wilson A. Harvey; Carnegie-Mellon University	205
"Using Generic Geometric Models for Intelligent Shape Extraction", Pascal Fua and Andrew J. Hanson; SRI International	227
"Stereo Correspondence: A Hierarchical Approach", Hong Seh Lim and Thomas O. Binford; Stanford University	234
"Symbolic Pixel Labeling for Curvilinear Feature Detection", John Canning, J. John Kim, Azriel Rosenfeld; University of Maryland	242
"Searching for Geometric Structure in Images of Natural Scenes", George Reynolds, R. Ross Beveridge; University of Massachusetts at Amherst	257

TABLE OF CONTENTS

	<u>Page</u>
<u>SECTION II - TECHNICAL REPORTS PRESENTED (Continued)</u>	
"Detecting Runways in Aerial Images", A. Huertas, W. Cole and R. Nevatia; University of Southern California	272
"A Report on the DARPA Image Understanding Architectures Workshop", Azriel Rosenfeld; University of Maryland	298
"Image Processing to Geometric Reasoning: Military Image Analysis at GE FESD", M.S. Horwedel; General Electric Federal Electronic Systems Division	303
"Image Understanding Technology and its Transition to Military Applications", David Y. Tseng; Hughes Research Laboratories; Julius F. Bogdanowicz; Electro-Optical and Data Systems Group, Hughes Aircraft Company	310
"Context Dependent Target Recognition", Teresa M. Silberberg; Hughes Artificial Intelligence Center	313
"Precompiling a Geometrical Model into an Interpretation Tree for Object Recognition in Bin-picking Tasks", Katsushi Ikeuchi; Carnegie-Mellon University	321
"Analytical Properties of Generalized Cylinders and Their Projections", Jean Ponce, David Chelberg, Wallace Mann; Stanford University	340
"Surface Segmentation and Description from Curvature Features", T.J. Fan, G. Medioni, and R. Nevatia; University of Southern California	351
"From Sparse 3-D Data Directly to Volumetric Shape Descriptions", Kashipati Rao and R. Nevatia; University of Southern California	360

TABLE OF CONTENTS

	<u>Page</u>
<u>SECTION II - TECHNICAL REPORTS PRESENTED (Continued)</u>	
"Object Recognition Using Alignment", Daniel P. Huttenlocher, Shimon Ullman; Massachusetts Institute of Technology	370
"Parallel Recognition of Objects Comprised of Pure Structure", Paul R. Cooper and Susan C. Hollbach; University of Rochester ...	381
"A Framework for Implementing Multi-Sensor Robotic Tasks", Peter K. Allen; Columbia University	392
"Shape from Darkness: Deriving Surface Information from Dynamic Shadows", John R. Kender, Earl M. Smith; Columbia University ...	399

VOLUME II

SECTION III - OTHER TECHNICAL REPORTS

"Vision and Visual Exploration for the Stanford Mobile Robot", Ernst Triendl and David J. Kriegman; Stanford University	407
"AuRA: An Architecture for Vision-Based Robot Navigation", Ronald C. Arkin, Edward M. Riseman, Allan R. Hanson; University of Massachusetts at Amherst	417
"Guiding an Autonomous Land Vehicle Using Knowledge-Based Landmark Recognition", Hatem Nasr, Bir Bhanu and Stephanie Schaffer; Honeywell Systems and Research Center	432
"The CMU Navigational Architecture", Anthony Stentz, Yoshimasa Goto; Carnegie-Mellon University	440
"Qualitative Navigation", Tod S. Levitt, Daryl T. Lawton, David M. Chelberg, Philip C. Nelson; Advanced Decision Systems	447

TABLE OF CONTENTS

	<u>Page</u>
<u>SECTION III - OTHER TECHNICAL REPORTS (Continued)</u>	
"Interpretation of Terrain Using Hierarchical Symbolic Grouping for Multi-Spectral Images", Bir Bhanu and Peter Symosek; Honeywell Systems & Research Center	466
"Design of a Prototype Interactive Cartographic Display and Analysis Environment", Andrew J. Hanson, Alex P. Pentland, and Lynn H. Quam; SRI International	475
"The Image Understanding Architecture", Charles C. Weems, Steven P. Levitan; University of Massachusetts at Amherst	483
"Constructs for Cooperative Image Understanding Environments", Christopher C. McConnell, Philip C. Nelson, Daryl T. Lawton; Advanced Decision Systems	497
"Interpreting Aerial Photographs by Segmentation and Search", David Harwood, Susan Chang, Larry S. Davis; University of Maryland	507
"Initial Hypothesis Formation in Image Understanding Using an Automatically Generated Knowledge Base", Nancy Bonar Lehrer, George Reynolds, Joey Griffith; University of Massachusetts at Amherst	521
"Goal-Directed Control of Low-Level Processes for Image Interpretation", Charles A. Kohl, Allen R. Hanson, Edward M. Riseman; University of Massachusetts at Amherst	538
"Active Vision", John (Yiannis) Aloimonos, Isaac Weiss; University of Maryland; Amit Bandyopadhyay; SUNY Stony Brook	552

TABLE OF CONTENTS

	<u>Page</u>
<u>SECTION III - OTHER TECHNICAL REPORTS (Continued)</u>	
"An Integrated System That Unifies Multiple Shape From Texture Algorithms", Mark L. Moerdler and John R. Kender; Columbia University	574
"DRIVE - Dynamic Reasoning from Integrated Visual Evidence", Bir Bhanu and Wilhelm Burger; Honeywell Systems & Research Center	581
"What Is A "Degenerate" View?", John R. Kender, David G. Freudenstein; Columbia University	589
"The Role and Use of Color in a General Vision System", Glenn Healey and Thomas O. Binford; Stanford University ...	599
"Using a Color Reflection Model to Separate Highlights from Object Color", Gudrun J. Klinker, Steven A. Shafer, and Takeo Kanade; Carnegie-Mellon University	614
"Recognizing Unexpected Objects: A Proposed Approach", Azriel Rosenfeld; University of Maryland	620
"Parallel Algorithms for Computer Vision on the Connection Machine", James J. Little, Guy Blelloch, and Todd Cass; Massachusetts Institute of Technology	628
"Solving The Depth Interpolation Problem on a Fine Grained, Mesh- and Tree-Connected SIMD Machine", Dong J. Choi; Columbia University	639
"Survey of Parallel Computers", Hong Seh Lim and Thomas O. Binford; Stanford University	644

TABLE OF CONTENTS

	<u>Page</u>
<u>SECTION III - OTHER TECHNICAL REPORTS (Continued)</u>	
"Evidence Combination Using Likelihood Generators", David Sher; The University of Rochester	655
"Multi-Modal Segmentation using Markov Random Fields", Paul B. Chou and Christopher M. Brown; The University of Rochester	663
"Minimization of the Quantization Error in Camera Calibration", Behrooz Kamgar-Parsi; University of Maryland	671
"Uncertainty Analysis of Image Measurements", M.A. Snyder; University of Massachusetts at Amherst	681
"Results of Motion Estimation With More Than Two Frames", Hormoz Shariat and Keith E. Price; University of Southern California	694
"Tracing Finite Motions Without Correspondence", Ken-ichi Kanatani and Tsai-Chia Chou; University of Maryland	704
"A Unified Perspective on Computational Techniques for the Measurement of Visual Motion", P. Anandan; University of Massachusetts at Amherst	719
"Hierarchical Gradient-Based Motion Detection", Frank Glazer; University of Massachusetts at Amherst	733
"Shape Reconstruction on a Varying Mesh", Isaac Weiss; University of Maryland	749
"Stereo Matching Using Viterbi Algorithm", G.V.S. Raju; Ohio University, Thomas O. Binford; Stanford University, and S. Shekhar; Stanford University	766

TABLE OF CONTENTS

	<u>Page</u>
<u>SECTION III - OTHER TECHNICAL REPORTS (Continued)</u>	
"Steps Toward Accurate Stereo Correspondence", Steven D. Cochran; University of Southern California	777
"Stereo Matching by Hierarchical, Micro-canonical Annealing", Stephen T. Barnard; SRI International	792
"The Formation of Partial 3D Models from 2D Projections - An Application of Algebraic Reasoning", D. Cyrlluk, D. Kapur, J.L. Mundy, and V. Nguyen; General Electric Corporate Research and Development ..	798
"Spectral and Polarization Stereo Methods Using a Single Light Source", Lawrence Brill Wolff; Columbia University	810
"Surface Curvature and Contour from Photometric Stereo", Lawrence Brill Wolff; Columbia University	821
"Qualitative Information in the Optical Flow", Alessandro Verri and Tomaso Poggio; Massachusetts Institute of Technology	825
"Algorithm Synthesis for IU Applications", Michael R. Lowry; Stanford University	835
"Generalizing Epipolar-Plane Image Analysis for Non-Orthogonal and Varying View Directions", H. Harlyn Baker, Robert C. Bolles, and David H. Marimont; SRI International	843
"Detecting Dotted Lines and Curves in Random-Dot Patterns", Richard Vistnes; Stanford University	849
"Learning Shape Computations", John (Yiannis) Aloimonos and David Shulman; University of Maryland	862

TABLE OF CONTENTS

	<u>Page</u>
<u>SECTION III - OTHER TECHNICAL REPORTS (Continued)</u>	
"Local Shape from Specularity", Glenn Healey and Thomas O. Binford; Stanford University	874
"Finding Object Boundaries Using Guided Gradient Ascent", Yvan Leclerc and Pascal Fua; SRI International	888
"Detecting Blobs as Textons in Natural Images", Harry Voorhees and Tomaso Poggio; Massachusetts Institute of Technology	892
"Reconstruction of Surfaces from Profiles", Peter Giblin and Richard Weiss; University of Massachusetts at Amherst	900
"Using Occluding Contours for Object Recognition", Willie Lim; Massachusetts Institute of Technology	909
"Parallel Optical Flow Computation", James Little, Heinrich Bulthoff and Tomaso Poggio; Massachusetts Institute of Technology	915
"Using Optimal Algorithms to Test Model Assumptions in Computer Vision", Terrance E. Boulton; Columbia University	921
"A Parallel Implementation and Exploration of Witkin's Shape from Texture Method", Lisa Gottesfeld Brown and Hussein A. H. Ibrahim; Columbia University	927
"Localized Intersections Computation for Solid Modelling with Straight Homogeneous Generalized Cylinders", Jean Ponce and David Chelberg; Stanford University	933
"Line-Drawing Interpretation: Straight-Lines and Conic-Sections", Vic Nalwa; Stanford University	942

TABLE OF CONTENTS

	<u>Page</u>
<u>SECTION III - OTHER TECHNICAL REPORTS (Continued)</u>	
"Line-Drawing Interpretation: Bilateral Symmetry", Vic Nalwa; Stanford University	956
"Estimation and Accurate Localization of Edges", Fatih Ulupinar and Gerard Medioni; University of Southern California	968
Edge-Detector Resolution Improvement by Image-Interpolation", Vishvjit S. Nalwa; Stanford University	981
"Detection, Localization and Estimation of Edges", J.S. Chen and G. Medioni; University of Southern California	988

Object Recognition Using Alignment

Daniel P. Huttenlocher
Shimon Ullman
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract. This paper presents an approach to recognition where an object is first *aligned* with an image using a small number of pairs of model and image features, and then the aligned model is compared directly against the image. For instance, the position, orientation, and scale of an object in three-space can be determined from three pairs of corresponding model and image features. By using a small fixed number of features to determine position and orientation, the alignment process avoids structuring the recognition problem as an exponential search. To demonstrate the method, we present some examples of recognizing flat rigid objects with arbitrary three-dimensional position, orientation, and scale, from a single two-dimensional image. The recognition system chooses features for alignment using a scale-space segmentation of edge contours. Finally, the method is extended to the domain of rigid objects in general.

Introduction

Object recognition involves identifying a correspondence between part of an image and a particular view of a known object. This requires matching the image against stored object models to determine if any of the models could produce a portion of the image. Even for a single model, a given object can appear very different depending on its position and angle with respect to the viewer. First, from a particular view part of an object will generally be occluded. Second, an object may be distorted by projection into the image plane (e.g., foreshortening). Finally, an object may itself undergo transformations such as having parts which move independently, or being stretched or bent. Most recognition systems assume that objects are rigid, and do not undergo any transformation [12] [4] [9] [2]. Some systems allow for perspective projection [16], and some have parameterized models which can articulate at certain points [7].

The presence of more than one object in an image also complicates the recognition problem. First, objects may occlude one another. Second, different objects in the image must somehow be individuated. In the case of touching

and overlapping objects this generally cannot be done prior to recognition, but rather must be part of the recognition process itself.

The Task

In this paper we consider the problem of matching a two-dimensional view of an object against a potential model. The viewed object can have arbitrary three-dimensional position, orientation, and scale, and may be touching or occluded by other objects (3D from 2D recognition). First we consider the domain of flat rigid objects such as the widget shown in Figure 1. While the viewed object is flat, the problem is not two-dimensional because a flat object positioned in three-space can undergo distortion such as foreshortening when projected into the image plane.

Like the general recognition task, this task suffers from problems of occlusion and of individuating multiple objects in an image. There is also a limited kind of shape distortion caused by projecting a rigid object into the image. We then consider extending the recognizer to the domain of rigid objects in general, such as the personnel carrier shown in Figure 2.

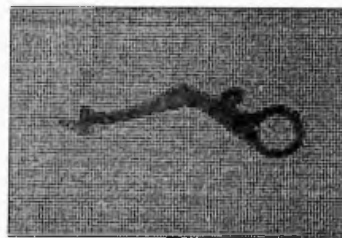


Figure 1. A flat widget used in recognition.

The current task cannot be handled by recognition systems which assume rigid objects with no distortion [12] [4] [9] [2], or by systems which only allow parameterized variation of rigid models [7]. The task is similar to that of Lowe, who addresses the problem of three-dimensional recognition from a single two-dimensional view [16]. However the task considered by Lowe is more restricted, because it is

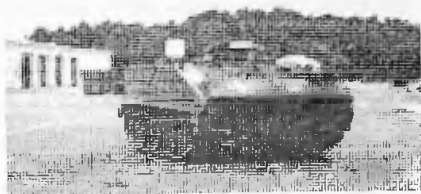


Figure 2. A personnel carrier used in recognition.

assumes that objects are polyhedral, and are viewed such that parallel surfaces appear more or less parallel.

In order to solve this task, we present a new approach where the recognition process is divided into two stages. In the first stage, a rigid object is *aligned* with an image using a small number of model and image features. In the second stage, the alignment is used to transform the model into image coordinates. The key observation underlying the alignment operation is that the position and orientation of a rigid object can be determined from a small number of position and orientation measures. Once the position and orientation have been determined, the model can be compared directly with the image. In contrast, current recognition systems search for the largest set of model and image feature pairs which are consistent with a single position and orientation of a rigid object. The number of such sets is exponential, requiring the use of various techniques to limit the search.

Matching Models and Images: Previous Approaches

In this section we briefly discuss the limitations of some recent recognition systems with respect to the recognition task described above (for a more general review see [3]). These systems all exploit rigidity by noting that for a given position and orientation of a rigid object, there must be a single transformation which maps each model feature onto its corresponding image feature. This transformation consists of a three-dimensional rotation and translation in 3D from 3D recognition, and a solution to the perspective viewing equation in 3D from 2D recognition [16].

Recognition is generally structured as a search for the largest pairing of model and image features for which there exists a single transformation mapping each model feature to its corresponding image feature [12] [4] [16] [7] [9]. For i image features and m model features there are at most $p = i \times m$ pairs of model and image features. Because of occluded image points, and image points which do not correspond to the model, any subset of these p pairs could be the largest set of matching model and image points, and thus the number of possible matches is exponential in the size of p . Two methods are used to limit this space of

possible matchings of model and image features.

The first method of limiting the possible matches is to use the identity of features to restrict the pairing of model and image features. However, even in the ideal case where each model feature has only a single corresponding image feature, there may be multiple matches to consider because of image features which actually correspond to other objects in the scene. Shape descriptions such as SLS [6], codons [14], and the curvature primal sketch [1] are all intended to produce relatively unique features for use in recognition.

The problem with using the identity of features in recognition is that there is a tradeoff between the uniqueness of a feature and the robustness with which it can be recognized. Since systems which rely heavily on the identity of features must use relatively unique features, they tend to be sensitive to noise and occlusion in the image.

For instance, the LFF [4] and 3DPO [5] recognition systems form feature descriptions by using local clusters of features. A "focus feature" in each cluster is chosen for use in matching. This feature is described in terms of its type (e.g., corner, hole), and the type, distance, and angle of the other features in the cluster. The use of local feature clusters yields relatively unique features. However, it is difficult to ensure that each cluster is composed of features from a single object, making the system sensitive to the position and orientation of neighboring and occluding objects.

The second method of limiting the possible matches is to use relations between features to eliminate inconsistent pairs of model and image features [12] [9] [4]. For instance, in order for two pairs of model and image features (m_1, i_1) and (m_2, i_2) to be part of a consistent set, the distance between the image features i_1 and i_2 must be the same as the distance between the model features m_1 and m_2 , within some error bound. Similarly, the angle between orientation measures for any pair of image features must match the angle between the corresponding pair of model features.

The problem with using relations between features in recognition is that the relations must be measurable in the image. Since relations such as distance and angle are not invariant under projection, three-dimensional recognition systems which use these relations require three-dimensional data. Relations which are invariant under projection tend to be much weaker than distance and angle relations.

It has been demonstrated that distance and angle relations can be used to greatly limit the number of possible matches of a model to an image, for both 2D from 2D [12] [4] and 3D from 3D [13] [5] recognition tasks. However, the method cannot readily be extended to handle 3D from 2D tasks. In addition to empirical demonstrations of the power of distance and angle relations among features, it has been

shown that simple bounded-noise distance constraints can be used to develop an $O(n^2)$ time algorithm for recognizing an isolated object with n features which can undergo two-dimensional rotation and translation. [2].

The SCERPO system [16] [17] is the only one to address the problem of three-dimensional recognition from a single two-dimensional view. Image edges are grouped together using proximity and parallelism. These local groupings are then used to form pairs of model and image features. A given set of feature pairs are consistent if there is a solution to the perspective viewing equation which maps each model feature onto its corresponding image feature. There are no simple pairwise checks such as distance and angle relations which can be used to incrementally test the consistency of a set of pairs given an added pair. Instead, the perspective viewing equation is solved for each added pair of model and image points. This is done using Newton-Raphson iteration, which allows a solution to be modified to account for a new pair.

None of these recognition systems effectively address the task of recognizing objects which have arbitrary three-dimensional position, orientation and scale, from a single two-dimensional view. Each system restricts the recognition task in order to limit space of possible matches of models and images (e.g., using distance and angle relations, or requiring parallel surfaces to appear parallel). In the next section we present a new approach for matching models and images which does not structure recognition as an exponential search. We then demonstrate this method on several images.

The Alignment Method of Recognition

We have seen above that recognition can be viewed as a search through the space of all possible positions and orientations of all possible objects. The idea of the *alignment* approach is to separate this search into two stages. In the first stage, the position, orientation, and scale of an object are found using a minimal amount of information, such as three pairs of model and image points. In the second stage, the alignment is used to map the object model into image coordinates for comparison with the image.

There are two major advantages of this approach. First, by using a small fixed number of model and image features, we avoid structuring recognition as a search through an exponential space. Second, a given alignment can be used to match multiple models against the image. If a group of objects are stored such that they are aligned with one another, then a single alignment computation will map the entire group of objects into the image.

The key observation behind the approach is that the alignment can be performed with a small amount of infor-

mation. For example, three image points and three corresponding model points are sufficient to determine the position, orientation and scale of a rigid object in three-space. Similarly, two points and an orientation measure can also be used to solve for the three-dimensional alignment.

In the case of two-dimensional recognition, only two pairs of corresponding model and image points are needed to align a model with an image. Consider two pairs, (m_1, i_1) and (m_2, i_2) , such that model point m_1 corresponds to image point i_1 and model point m_2 corresponds to image point i_2 . First the model is translated in x and y such that m_1 is coincident with i_1 . Then it is rotated about the new m_1 such that the edge m_1m_2 is coincident with the edge i_1i_2 . Finally the scale factor is computed to make m_2 coincident with i_2 . These two translations, one rotation, and a scale factor will make each unoccluded point of the model coincident with its corresponding image point, as long as the initial correspondence of (m_1, i_1) and (m_2, i_2) was correct.

For 3D from 2D recognition, the alignment method is somewhat more complicated. In the final section of the paper, we show how to use three pairs of model and image points to position and orient a model in three-space given a single two-dimensional view, assuming orthographic projection. A transformation from the model to the image consists of 2-dimensional translation, three-dimensional rotation, and a linear scale factor which is proportional to the viewing distance. Under normal viewing conditions, orthographic projection plus scale is a reasonable approximation to perspective viewing. Under high perspective distortion – when the object occupies much of the field of view – the approximation is poor. However, under such conditions people also appear to be bad at recognition.

Consider an object, O , with three-dimensional positional freedom, and a two-dimensional image, I , which contains a view of O (perhaps along with other objects). We are interested in using the alignment computation to find O in the image. The feature detector discussed in the next section returns a set of pairs of matching object and image features, P . Any three-tuple of the pairs in P specifies a possible alignment of the object with the image. In general some small number of these alignments will define the true position and orientation of the object, and the rest will be due to incorrect matchings of model and image points. Thus the recognition problem is to determine which alignment in P defines the transformation mapping the most model points onto their corresponding image points.

Given a set of pairs of model and image features, P , we solve for the alignment specified by each triple in P . For some triples there will be no possible alignment of the three model and image features. Each remaining triple specifies a transformation mapping model points to image points. An alignment is scored by using the transformation to map

the model edges into the image, and correlating the transformed model edges with the image edges. The best alignment is the one which correlates the most model edges with image edges.

For m model features and i image features, the number of pairs of model and image features, p , is at most $i \times m$. With a good labeling scheme, the number of pairs, p , will be much smaller, approaching m when each model point has one corresponding image point. Given p pairs of features, there are $\binom{p}{3}$, or $O(p^3)$, three-tuples of pairs, each of which specifies a possible alignment of the model and the image. Each alignment is scored by mapping the model edges into the image. If the model edges are of length l , then the worst case running time of the algorithm is $O(lp^3)$. Thus by structuring the recognition process as an alignment stage followed by a comparison stage, it is transformed from the exponential problem of finding the largest consistent set of model and image points, to the polynomial problem of finding the best triple of model and image points.

Shape Features: Segmenting Edge Contours

The recognition system uses edge-based shape features for matching models against images. The input to the system is a grey-level image. The image is processed by an edge detector [8], and the edges are chained together using an eight-way chaining algorithm. Chains with low overall edge strength are discarded. Thresholding whole chains rather than individual edge points produces a more stable output. Finally, edge chains with unambiguous nearest neighbors are merged together if they can be connected by a smooth spline.

Once pieces of edge contour have been chained together, simple shape descriptors are derived using the local curvature of the edge contours. The curvature is computed as the change in angle (per unit arclength) between local tangent vectors at neighboring pixels. The tangents are computed using the least squares best fit line over a small local neighborhood.

A relatively stable segmentation of an edge contour is obtained by breaking the contour at zero crossings of curvature (inflection points in the contour). Zero curvature regions are also identified, segmenting the contour into straight, positive curvature and negative curvature pieces. Zero crossings of curvature were chosen as segmentation points because they are stable under three-dimensional rotation and projection, except in the degenerate case where they disappear (when the contour projects into a line). Maximum curvature points, which are often used to segment edge contours [10] [16], are not at all stable under rotation and projection.

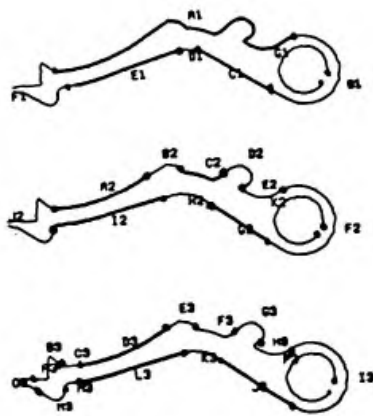


Figure 3. A simple curvature scale-space segmentation of a widget.

A hierarchy of curve segmentations can be obtained by smoothing the curvature at different scales, and using the smoothed curvature to segment the edge contour. Smoothing in curvature space preserves only those inflection points in the edge contour (zero crossings of curvature) which are significant at a given scale. Thus coarser scale segments correspond to merging neighboring segments at finer scales, producing a scale-space [18] [19] hierarchy of segments. Since coarser scales of smoothing do not introduce zero crossings which were not present at finer scales, the hierarchy forms a tree of segments from coarser to finer scales.

Figure 3 shows a three-level scale-space curvature segmentation of the edge contours of the widget in Figure 1. Each part of the Figure shows the same contour, segmented according to the curvature smoothed at different scales (using Gaussian filters of size $\sigma = 7, 20$, and 40 pixels, respectively). The coarsest scale is at the top of the Figure and the finest scale is at the bottom. The endpoints of each segment are delimited by a dot, and zero curvature regions are shown in bold.

Each segment of edge contour is classified according to whether it is curved or straight. The curved segments are further classified by the degree of closure: open or closed, and the smoothness of the contour: smooth or unsmooth, yielding a total of five types of segments. Richer descriptions are then obtained by combining the classifications at multiple scales of smoothing.

The three segmentations in Figure 3 form a (multi-rooted) tree, where each region at a coarse scale corresponds to one or more regions at each finer scale. Figure 4 shows this scale-space segment tree. Each segment in the tree is indicated by its label from Figure 3, and by the type of segment: straight, curve, and open-curve.

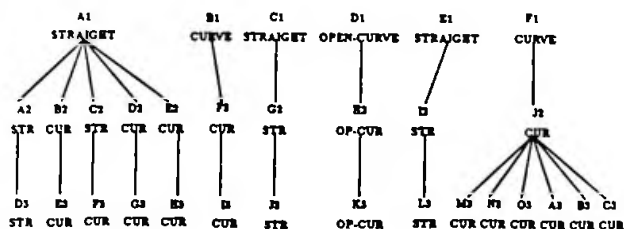


Figure 4. Multi-scale segmentation tree corresponding to the curvature scale-space segmentation in Figure 3.

Multi-scale descriptions are formed using the types of segments at a given scale, plus the structure of the hierarchy at the next finer scale. Two aspects of the tree structure are used. First, a segment is classified according to whether it corresponds (primarily) to one or many segments at the next finer scale: **single** or **multiple**. Second, a multiple segment is classified according to whether or not the finer scale segments form a regular pattern. A pattern consists of a repetition of the same type of segment, in either the same or opposite directions of curvature. This yields the classification **irregular**, **regular-same**, and **regular-opposite**.

For example, using this multi-scale description, the straight segment A1 at level 1 in the tree is differentiated from the other straight segments C1 and E1 at the same level, because A1 is composed of multiple, irregular segments at the next level whereas C1 and E1 are each composed of a single segment.

Using the multi-scale description, at the coarsest scale the widget is composed of seven segments, only two of which can be confused with one another (the two straight segments C1 and E1). The segments are, A1: straight, multi, irregular; B1: curve, single, smooth; C1: straight, single; D1: open-curve, single; E1: straight, single; F1: curve, multi, irregular; G1: closed-curve, single, smooth. Thus, the result of this multi-level description is relatively unique classification of the edge segments, even though the individual features are coarse, and sparsely distributed in the image.

Points for use by the alignment algorithm are obtained from the features as follows. For closed segments of contour, the center of the region defined by the contour is used. This point is found from the intersection of the major and minor axes of the region. For straight segments the endpoints are used, and for other curved segments the middle of the curve is used. Since the endpoints of the segments are at inflection points or the ends of zero curvature regions, they are relatively stable, making it reasonable to use endpoints and midpoints for matching.

Given a model and an image, the multi-scale description of both the model and the image are computed. The

coarse level descriptions are then used to determine which model features can potentially match which image features. These possibly matching features form the set of pairs, P , for the alignment algorithm. In scoring how well an alignment matches an image, fine scale model features are used.

Examples

In this section we show several examples of the recognizer processing grey-scale images of widgets. The model is the multi-scale description of the widget shown in Figure 3 and Figure 4. The model is just the result of processing the image of the isolated widget in Figure 1 in the same manner as any image. Thus for flat objects, models are formed directly from an image of an object.

The recognizer is implemented on a Symbolics 3650, and takes from 2-5 minutes for each of the examples shown in this section. A multi-scale description of the edge segments is formed and used to define alignment points in the image, as described in the previous section. Possible alignments are then computed using all three-tuples of model and image point pairs. For each alignment, the model is mapped into the image and the transformed model edges are correlated with the image edges. The alignments are ranked based on the percentage of the model edge contour for which there is a corresponding image edge contour.

Each example is illustrated by a Figure with four parts: i) the grey-level image, ii) the thresholded image edges extracted by the edge detector, iii) the model edges, iv) the edges of the aligned model superimposed on the image. Part iv) also indicates the points which were used in computing the alignment of the model with the image.

The example in Figure 5 shows two widgets in the plane. The top widget has been flipped over, and thus cannot be recognized using only two-dimensional transformations. The recognizer finds two distinct positions and orientations of the model which match 99% and 98% of the model edge contour to image edges. These two matches are shown in part iv) of the Figure.

Another position and orientation is found at the alignment stage, but is eliminated because the correlation with the image is poor, and the image edges are accounted for by a better alignment. This alignment is shown in Figure 6 superimposed with the image edges. This alignment is found because the two straight edges are indistinguishable, and the three points used for alignment were the two straight edges and the bend.

Figure 7 shows a widget which has been tilted approximately 30 degrees by resting one end of it on a block, for-shortening the image. The recognizer finds a single best position and orientation, which is shown in part iv) of the Figure.

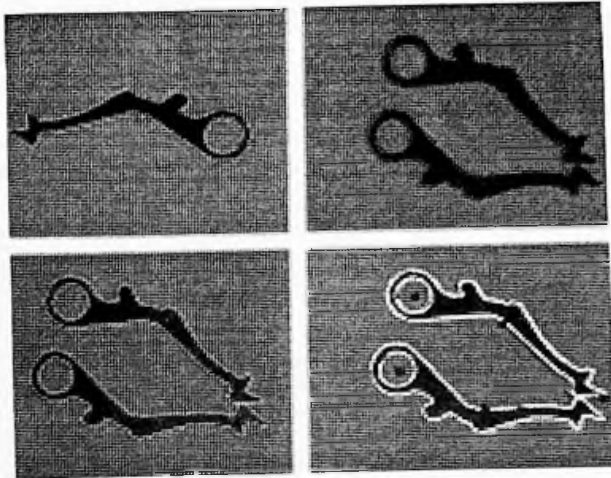


Figure 5. Matching a widget against an image of two widgets in the plane. See the text for a description.

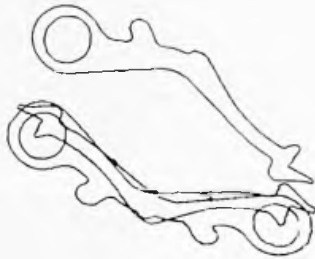


Figure 6. An alignment of a widget with an image which does not match the fine scale model features with image features.

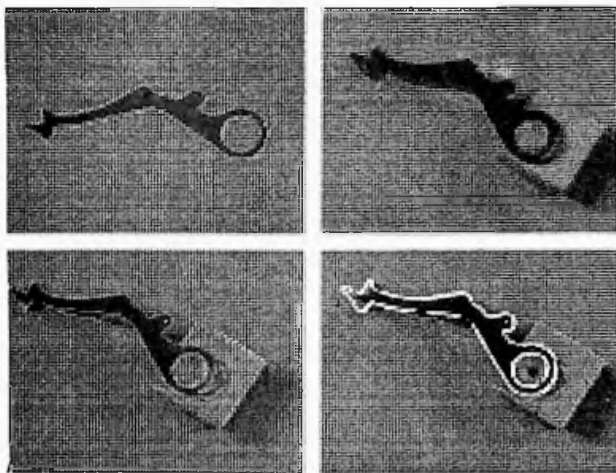


Figure 7. Matching a widget against an image of a forshortened widget.

The next example, in Figure 8, shows another widget which has been tilted out of the image plane — for instance the circular end is now an oval. The best match is shown in part iv) of the Figure.

Finally, we demonstrate the ability of the recognizer to find partly occluded objects. As long as three features are visible in the image, the alignment algorithm will be able to

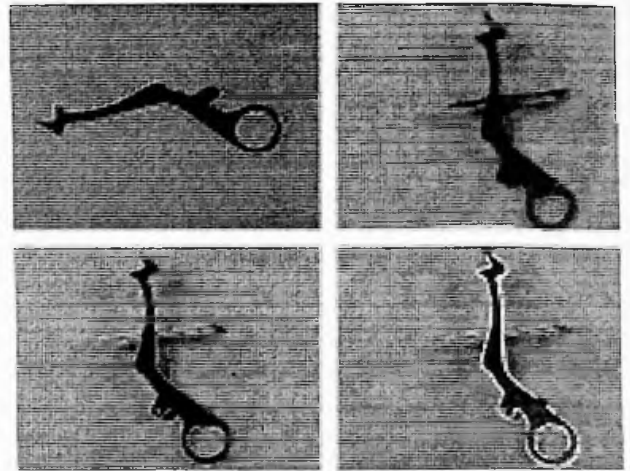


Figure 8. Matching a widget against an image of a tilted widget.

align the model with the image. Figure 9 shows a widget which has a pile of smaller widgets obscuring the circular end. The best alignment matches 80% of the model edge contour to image edges, and is shown in part iv) of the Figure. Figure 10 shows two widgets obscured by each other and several smaller objects. The matcher finds two distinct positions and orientations, which are shown in part iv) of the Figure.

From these examples we see that the alignment algorithm finds a small number of reasonable matches of widgets to images, even when the widget is forshortened, scaled, and partly occluded. The scoring method of mapping model edges into the image provides a simple method for finding the best alignment.

Recognizing Non-Flat Objects

Extending the alignment method to recognize non-flat objects such as the personnel carrier in Figure 2 is relatively straightforward. The only difference between using a three-dimensional model and a two-dimensional model is the need to eliminate portions of the object which are not visible from a given position and orientation. Thus, after computing a possible alignment, model edges and surfaces which are not visible from the specified position and orientation must be removed before the model is mapped into the im-

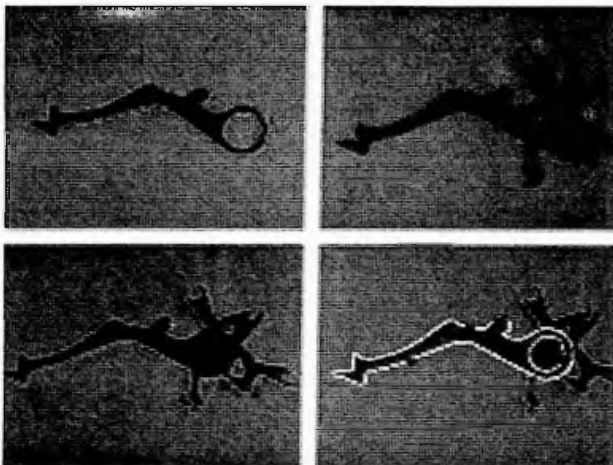


Figure 9. Matching a widget against an image of a partly occluded widget.

age (cf. [11]).

For a three-dimensional model, it must also be ensured that the three points used in alignment are all visible given the position and orientation of the object which they specify. Thus for each possible alignment, the three model points must be checked to make sure that they are not on hidden edges or surfaces.

Rather than using a single three-dimensional model of an object, it is possible to use multiple planar views, with one view for each position from which a different set of object surfaces are visible. Using multiple local alignments, such a planar model can be matched to a variety of different

Using just a single alignment, a planar model of a three-dimensional object will only match images which are taken at close to the same viewing angle as the model view. The distortion in using a planar model comes from the fact that the model actually contains multiple object planes, even though the points in the model are only two-dimensional. However by using multiple alignments, one for each plane of the object, a flat model can be matched to images from a wide range of different viewpoints.

One method of finding such regions for local alignment is to triangulate the set of model features used in alignment, and separately align each triangular region. To map the model to the image, all the model points inside a given triangle are transformed using the alignment for that triangle. To the extent that a triangle falls on a nearly planar part of an object, this will produce a correct alignment for that region. By aligning each locally neighboring triple of model features found in the triangulation, rigidity is preserved for

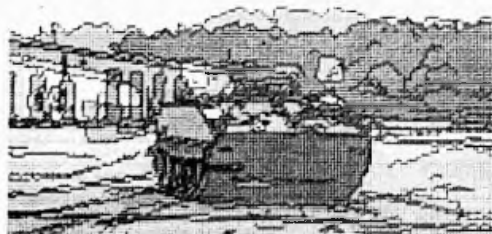
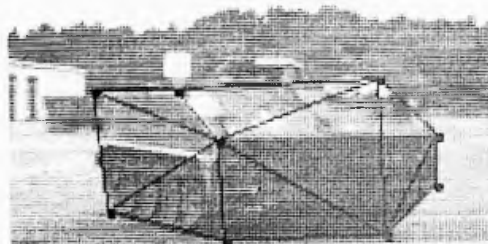
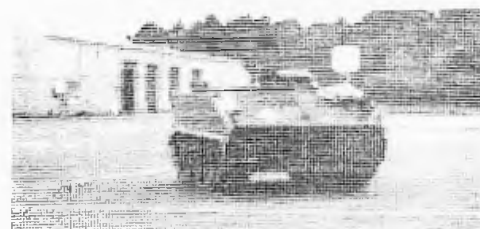


Figure 11. Local alignment of two views of a personnel carrier.

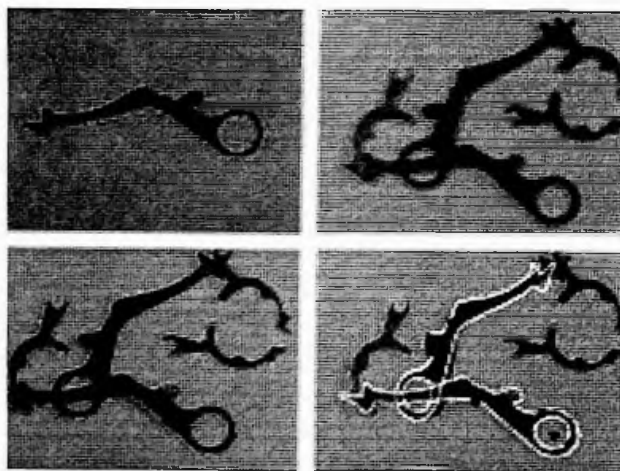


Figure 10. Matching a widget against an image of two partly occluded widgets.

images of the object.

each triangle, but not for the object as a whole.

Points in a given triangle will not be mapped into the image correctly if they are not on the same plane of the object as the three points defining the triangle. However, such points will usually be transformed to a point near their correct position because the alignment will be partially correct. Therefore the locally-rigid alignment process can be iterated, starting with a two-point or three-point alignment, and using the partial match to pick potential corresponding model and image points for computing more refined, and less globally rigid, alignments.

To show how this local alignment algorithm works, the image of a personnel carrier in Figure 2 is matched with another view taken from a slightly different position, shown in Figure 11a. Simple three-dimensional alignment of these images is insufficient because the side of the personnel carrier has become substantially foreshortened whereas the front has not. Figure 11b shows the triangulation used for matching the two images (the correspondence between the two images was specified by hand). Figure 11c shows a synthetic image formed by transforming the image in part b according to the specified triangles. Figure 11d shows the images in parts a and c superimposed. To better see how well the two personnel carriers match, the intensity edges of the two images are also superimposed on the images. The match is good enough to even bring small details such as the white cross painted on the front into correspondence.

The 3D from 2D Alignment Method

In this section we present the alignment method in detail. We show that the position, orientation, and scale of an object in three-space can be determined from a two-dimensional image using three pairs of corresponding model and image points. This method approximates perspective viewing by orthographic projection plus a scale factor which is proportional to the viewing distance. As mentioned above, this is a reasonable approximation except when the object is very large with respect to the viewing distance.

Consider three model points a_m , b_m and c_m and three corresponding image points a_i , b_i and c_i , where the model points specify three-dimensional positions, (x, y, z) , and the image points specify positions in the image plane, $(x, y, 0)$. The alignment task is to find a transformation which maps the plane defined by the three model points onto the image plane, such that each model point coincides with its corresponding image point. If no such transformation exists, then the alignment process must determine this fact.

Since the viewing direction is along the z -axis, an alignment is a transformation which positions the model such that a_m projects along the z -axis onto a_i , and similarly for

b_m onto b_i , and c_m onto c_i . The transformation consists of translations in x and y , and rotations about three orthogonal axes. There is no translation in z because all points along the viewing axis are equivalent under orthographic projection. Instead, distance from the viewer is reflected by a change in scale.

The first step in finding an alignment is to translate the model points so that one point projects along the z -axis onto its corresponding image point. Using the point a_m for this purpose, the model points are translated by $(x_{a_i} - x_{a_m}, y_{a_i} - y_{a_m}, 0)$, yielding the model points a'_m , b'_m and c'_m . This brings a'_m , the projection of a'_m into the image plane, into correspondence with a_i , as illustrated in Figure 12a.

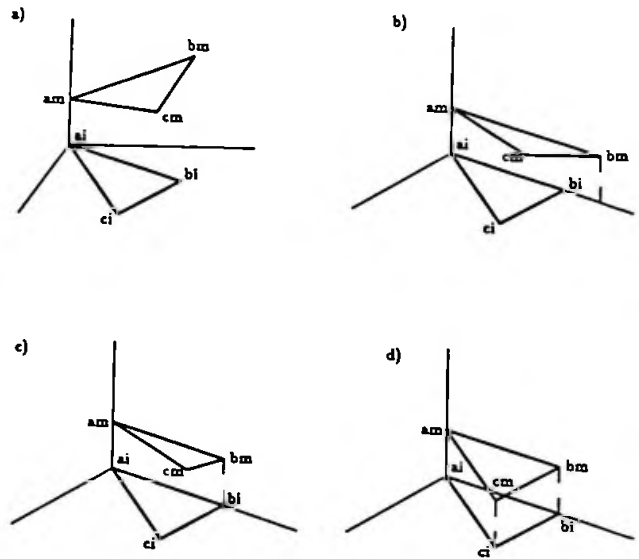


Figure 12. The alignment process: a) the points a_i and a_m are brought into correspondence, b) the ab edges are aligned, c) the points b_i and b_m are brought into correspondence, d) the points c_i and c_m are brought into correspondence.

Now it is necessary to rotate the model about three orthogonal axes to align b_m and c_m with their corresponding image points. First we align one of the model edges with its corresponding image edge by rotating the model about the z -axis. Using the $a'_m b'_m$ edge we rotate the model by the angle between the image edge $a_i b_i$, and the projected model edge $a'_m b'_m$, yielding the new model points b''_m and c''_m , as illustrated in Figure 12b.

To simplify the presentation, the coordinate axes are now shifted so that a_i is the origin, and the x -axis runs along the $a_i b_i$ edge.

Since b''_m , the projection of b''_m into the image plane, lies along the x -axis, it can be brought into correspondence with b_i by simply rotating the model about the y -axis. The

amount of rotation is determined by the relative lengths of $a_m b_m$ and $a_i b_i$, because the model must be rotated such that the projected model edge is the same length as the image edge. If the model edge is shorter than the image edge, then there is no such rotation, and hence the model cannot be aligned with the image.

Thus, the model points b''_m and c''_m are rotated about the y -axis by ϕ to obtain b'''_m and c'''_m , where

$$\cos \phi = \frac{\|b_i \cdot (1, 0, 0)\|}{\|b_m \cdot (1, 0, 0)\|}$$

for $0 \leq \cos \phi \leq 1$. Note that since a_i is the origin, the length of the edges is just b_i and b_m . The result of this rotation is illustrated in Figure 12c.

Finally, c'''_m is brought into correspondence with c_i by rotation about the x -axis. The degree of rotation is again determined by the relative lengths of model and image edges. However, in the previous case, the edges were parallel to the x -axis, and therefore the length was the same as the x component of the length. In this case, the edges need not be parallel to the y axis, and therefore the y component of the lengths must be used. Thus, the rotation about the x -axis is θ , where

$$\cos \theta = \frac{\|c_i \cdot (0, 1, 0)\|}{\|c_m \cdot (0, 1, 0)\|} \quad (2)$$

for $0 \leq \cos \theta \leq 1$.

If the model distance is shorter than the image distance, there is no transformation which aligns the model and the image. Furthermore, if the rotation does not actually bring c'''_m into correspondence with c_i , then there is also no alignment. This latter case can result because the rotations are those which will bring the points into alignment if there actually is a consistent solution. If there is no solution then it may still be possible to solve for the rotations, but they will not bring all three points into alignment.

The final rotation brings the plane defined by the three model points into correspondence with the image plane, as illustrated in Figure 12d. This combination of translations and rotations can now be used to map the model into the image, in order to determine if the object is in fact in the image at this position and orientation.

Now we show how to solve for a linear scale factor as a sixth unknown. First we note that scaling the model changes only the final two rotations (about the x - and y -axis), and not the x and y translations, or the rotation around the z -axis. This is because scaling changes the distances between the three model points, and only the rotations about the x - and y -axis depend on the distances between points.

The final two rotations align b_m with b_i , and c_m with c_i . The alignment of b_m involves movement of b_{mi} along the x -axis, whereas the alignment of c_m involves movement of c_{mi} in both the x and y directions.

Since the movement of b_{mi} is a sliding of along the x -axis, only the x -component, x_b , changes. The change is given by the rotation ϕ about the y -axis, as in (1). With a scale factor, s , this becomes

$$x'_b = s x_b (\cos \phi) \quad (3)$$

Similarly the movement of c_{mi} in the y direction is given by the rotation θ about the x -axis, as in (2). With a scale factor this becomes

$$y'_c = s y_c (\cos \theta) \quad (4)$$

The movement of c_{mi} in the x direction is given by the rotations about both the x - and the y -axis. From the matrix for a combined rotation about the x - and y -axis we obtain

$$x' = (x \cos \theta + y \sin \phi \sin \theta)$$

Thus with the scale factor, the the x component of c_m is

$$x'_c = s(x_c \cos \theta + y_c \sin \phi \sin \theta) \quad (5)$$

Now we have three equations in the three unknowns, s , θ , and ϕ . One method to solve for s is to substitute for $\cos \theta$, $\sin \theta$, and $\sin \phi$ in (5). From (3) we know that,

$$\sin \phi = \frac{1}{s x_b} \sqrt{s^2 x_b^2 - x'^2_b} \quad (6)$$

And similarly from (4),

$$\sin \theta = \frac{1}{s y_c} \sqrt{s^2 y_c^2 - y'^2_c} \quad (7)$$

Substituting (6) and (7) into (5) and simplifying yields

$$s^2 (x_b x'_c - x_c x'_b)^2 = (s^2 x_b^2 - x'^2_b) (s^2 y_c^2 - y'^2_c)$$

Expanding out the terms we obtain

$$s^4 (x_b^2 y_c^2) - s^2 (x_b^2 y'^2_c + x'^2_b y_c^2 + (x_b x'_c - x_c x'_b)^2) + x'^2_b y'^2_c$$

a quadratic in s^2 . While there are generally two possible solutions, it can be shown that only one of the solutions will specify possible values of $\cos \phi$ and $\cos \theta$.

Having solved for the scale of an object, the final two rotations ϕ and θ can be computed using (1) and (2) modified to account for the scale factor,

$$s(\cos \phi) = \frac{\|b_i \cdot (1, 0, 0)\|}{\|b_m \cdot (1, 0, 0)\|}$$

and

$$s(\cos \theta) = \frac{\|c_i \cdot (0, 1, 0)\|}{\|c_m \cdot (0, 1, 0)\|}$$

Thus solving for scale involves the computation of s , and slight a modification to the computation of the final two rotations.

For planar models, the three-dimensional alignment task only involves mapping points from one plane to another. Therefore, a planar model can be aligned with an image using only two-dimensional operations. In effect, the actual three-dimensional rotation and translation of the object are simulated in the plane. This *planar alignment* specifies x and y translation, planar rotation, x scale, y scale,

and a shear which is a scaling in x proportional to distance from the y -axis (or vice versa). The details of this alignment method are given in [15].

Summary

Recognition is generally viewed as a search through the space of possible positions and orientations of objects. The idea of the *alignment* approach is to separate this search into two stages. In the first stage, the position, orientation, and scale of an object are found using a minimal amount of

information, such as three pairs of model and image points. In the second stage, the alignment is used to map the object model into image coordinates for comparison with the image.

There are two major advantages of this approach. First, by using a small fixed number of model and image features, we avoid structuring recognition as search through an exponential space. Second, by storing object models such that they are aligned with one another, a single alignment computation can be used to map multiple objects into the image.

The key observation behind the approach is that the alignment can be performed with a small amount of information. For example, three points are sufficient to determine the position, orientation and scale of a rigid object in three-space from a single two-dimensional image. Similarly, two points and an orientation measure can also be used to solve for this alignment.

We have implemented a recognizer using the alignment method. This system chooses features for alignment using a scale-space segmentation of edge contours. Coarse scale segments are described both in terms of their shape, and the structure of the scale-space hierarchy at the next finer level. This produces relatively unique features for use in finding possible alignments of a model and an image. Fine scale segments are then used to determine how well a given alignment matches the model with the image.

To demonstrate the recognition method, several examples were shown of the recognizer finding a widget with arbitrary three-dimensional position and orientation, in a two-dimensional image. From these examples it can be seen that the alignment algorithm finds a small number of reasonable matches of widgets to images, even when the widget is forshortened, scaled, and partly occluded.

Acknowledgments

This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence

research is provided in part by the System Development Foundation and in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N0014-85-K-0124.

References

1. Asada, H. and Brady, M. 1984. The Curvature Primal Sketch, MIT Artificial Intell. Lab. Memo, No. 758.
2. Baird, H. 1986. *Model-Based Image Matching Using Location*. Cambridge, Mass: MIT Press.
3. Besl, P.J. and Jain, R.C. 1985. Three-Dimensional Object Recognition, *ACM Computing Surveys* bf 17(1), pp. 75-154.
4. Bolles, R.C. and Cain, R.A. 1982. Recognizing and Locating Partially Visible Objects: The Local Feature Focus Method, *Int. J. Robotics Res.* 1(3), pp. 57-82.
5. Bolles, R.C. and Horaud, P. 1986. 3DPO: A Three-Dimensional Part Orientation System *Int. J. Robotics Res.* 5(3), pp. 3-26.
6. Brady, M. and Asada, H. 1984. Smoothed Local Symmetries and Their Implementation, MIT Artificial Intell. Lab. Memo, No. 757.
7. Brooks, R.A. 1981. Symbolic Reasoning Around 3-D Models and 2-D Images, *Artificial Intell. J.* 17, pp. 285-348.
8. Canny, J. 1986. A Computational Approach to Edge Detection, *IEEE Trans. Pat. Anal. and Mach. Intel.* 8(6), pp. 679-698.
9. Faugeras, O.D. and Hebert, M. 1986. The Representation, Recognition, and Locating of 3-D Objects, *Int. J. Robotics Res.* 5(3), pp. 27-52.
10. Fischler, M.A. and Bolles, R.C. 1986. Perceptual Organization and Curve Partitioning, *IEEE Trans. Pat. Anal. and Mach. Intel.* 8(1), pp. 100-104.

11. Foley, J.D. and VanDam A. 1984. *Fundamentals of Interactive Computer Graphics*. Reading, Mass: Addison.
12. Grimson, W.E.L. and Lozano-Pérez, T. 1984. Model-Based Recognition and Localization from Sparse Range or Tactile Data, *Int. J. Robotics Res.* **3**(3), pp. 3-35.
13. Grimson, W.E.L. and Lozano-Pérez, T. 1987. Localizing Overlapping Parts by Searching the Interpretation Tree, to appear in *IEEE Trans. Pat. Anal. and Mach. Intell.*.
14. Hoffman, D.D. and Richards, W.A. 1986. Parts of Recognition, in *From Pixels to Predicates: Recent Advances in Computational and Robotic Vision*, edited by A.P. Pentland. Norwood, N.J.: Ablex.
15. Huttenlocher, D.P. and Ullman, S. 1987. Recognizing Rigid Objects by Aligning Them with an Image, MIT Artificial Intell. Lab. Memo, No. 937.
16. Lowe, D.G. 1985. *Perceptual Organization and Visual Recognition*. Boston: Kluwer.
17. Lowe, D.G. 1986. Three-Dimensional Object Recognition from a Single Two-Dimensional View, Courant Institute Robotics Report, No. 62.
18. Witkin, A.P. 1983. Scale-Space Filtering, in *From Pixels to Predicates: Recent Advances in Computational and Robotic Vision*, edited by A.P. Pentland, pp. 5-19. Norwood N.J.: Ablex.
19. Yuille, A.L. and Poggio, T. 1983. Scaling Theorems for Zero Crossings *IEEE Trans. Pat. Anal. and Mach. Intell.* **8**(1), pp. 15-25.