Affine Invariant Matching

by

Robert Hummel
Haim Wolfson
Courant Institute

March, 1988

New York University
251 Mercer Street
New York, New York 10012
hummel@nyu.edu, wolfson@acf8.nyu.edu

Work described in this paper was supported by Office of Naval Research Grants N00014-82-K-0381 and N00014-85-K-0077 Work Unit NR-4007006, and National Science Foundation Grants NSF DCR-83-20085 and IRI-8703335.

Researcher	Position	Vision Interests
J. Schwartz	Professor, Computer Science (on leave)	Model-based vision, matching
M. Sharir	Professor, Computer Science (visiting)	Shape representation
R. Hummel	Assistant Professor, Computer Science	Representations, feature extraction, knowledge aggregation
H. Wolfson	Research Scientist, Computer Science	Efficient matching, model-based vision
P. Wright	Professor, Computer Science	Automated manufacturing
E. Schwartz	Adjunct Professor, Computer Science Professor, Medical Center	Neurophysiology, retinal mapping, extra-striate physiology
D. Lowe	Assistant Professor, Computer Science (on leave)	Model-based vision
M. Bastuscheck	Research Scientist, Computer Science	3-D sensor design
J. Hong	Professor, Computer Science (visiting)	Shape representation
X. Tan	Research Scientist (visiting)	Shape matching
M. Landy	Assistant Professor, Psychology	Shape perception
R. Shapley	Professor, Psychology	Single cell recordings, visual pathway
A. Movshon	Associate Professor, Psychology	Neuroanatomical mapping, motion
G. Sperling	Professor, Psychology	Psychophysics: Motion perception, attention, computational vision

Table 1. Selected NYU Researchers in Computer Vision

AFFINE INVARIANT MATCHING

Robert Hummel and Haim Wolfson

Robotics Research Laboratory
Courant Institute, New York University
251 Mercer Street, New York, NY 10012

Abstract

We begin with a selected overview of computer vision research at New York University, emphasizing work on **representation of images**, and on **model-based vision**. We then focus on a technique for fast matching of shape descriptions of objects to preprocessed models. The matching process permits general affine transformations of the shape, and thus is applicable to the problem of recognizing flat, or nearly flat, objects viewed from an arbitrary angle and distance in 3-D. Different methods apply when the boundary of the objects consist primarily of line segments, or general (non-convex) curves, or (the most difficult case), convex curves. We also discuss the case of recognizing 3-D objects, as opposed to flat objects. The novelty of the approach and the computational efficiency is achieved by a hashing technique indexed over possible affine transformations.

1. A selected summary of Computer Vision research at NYU

In Table 1, we list a number of researchers, their titles and affiliations within NYU, and topics of research interest. Within recent years, computer vision research has become a speciality in the departments of computer science and psychology at New York University. Research in the psychology department has focused on computational vision studies through psychophysics and cell recordings, whereas research within the Courant Institute's computer science department has included high-level vision and sensor design within the Robotics Laboratory, and theories of low-level vision and mathematical analysis of feature extraction within the interdisciplinary "representation of images" sponsored research program. Most of the researchers listed in Table 1 independently direct a group of students, programmers, and other staff. We describe some of the activities of these groups below.

1.1. Representation of images

Professor Robert Hummel, in collaboration with Professor Michael Landy in the Psychology Department, has been directing a research project on low-level vision, in which they attempt to take a methodical, analytical, and mathematically sound approach to a comparative analysis of representations that can be useful for the analysis of images. The work is motivated by knowledge that human visual processing is mediated by center-surround receptive fields of various sizes in early stages of the visual pathway, and by directional and motion selective processing in subsequent stages. Accordingly, the research has focused on representational issues involving pyramid data structures, scale-space structures, and on methods for extracting information from low-level cues, such as depth from motion perception, and on methods for combining information from multiple cues to provide higher-level inferences.

Three methods are used to evaluate a representation. First, a proposed representation should be analyzed mathematically, to the extent possible. Questions of completeness, continuity, and stability are the standard mathematical questions that must be addressed. Next, the information content of a representation can be assessed by using psychophysics to determine the information that is used by the vision system to induce perceptual effects. For example, we can produce a reconstruction, and conduct experiments to compare the information content in an original image with the impoverished reconstruction. Finally, a representation can be shown to be worthwhile if a useful vision system can make use of the construct. We next describe how several subprojects support this methodology of representation evaluation.

1.1.1. Solving ill-conditioned inverse problems

Many problems in vision involve the solution of inverse problems. Interpolation, segmentation, motion extraction, shape-from-shading and other cues, and many other standard problems in vision can be viewed as inverse problems. Moreover, the problem of reconstructing image data from a representation is also frequently an inverse problem. At the Courant Institute, interest in inverse problems is a hallmark of research work over the past fifty years.

Nearly all inverse problems are ill-posed, in the sense that either no solution exists, or many solutions exist, or if one exists, there are many other candidate solutions which are nearly as good. The latter situation, when many images project to nearly the same representation, characterizes the case of an ill-conditioned inverse problem. A prototypical example is the deblurring problem: to recover the image data from a Gaussian-blurred version of the original image. In [1], we examined a novel filtering approach to deblurring. More practical instances of vision-related ill-conditioned inverse problems include reconstructions from zero-crossings, interpolation, and reconstructions from sparsely sampled filtered data. These topics have been addressed in subsequent studies.

One way to approach certain ill-conditioned inverse problems, formulated as variational minimization problems, is to add a regularization term. When done carefully, the result is an image or function that is not quite a solution to the original problem, but satisfies some smoothness constraints. By allowing for *controlled continuity*, it is even possible to allow for some discontinuities.

A major contribution of research effort by Professor Hummel and Dr. Moniot has been to show the utility of a different approach. The method, called ***minimization of equation error***, is applicable whenever the inverse problem involves the specification of information in scale-space. To date, the technique has been applied to deblurring [2] and to reconstructions from zero-crossings [3]. The results in both domains have been excellent. A principle advantage of the approach is that there is no explicit smoothness constraint. Instead, there is an assumption that the supplied information came about from a sampling in scale-space.

1.1.2. Scale-space representations

In a number of publications and presentations, researchers at NYU have become major proponents of the scale-space viewpoint on pyramid data structures [4,5]. The idea of using the Heat Equation to model the process of representing image data by a continuum of Gaussian-blurred images is not new, but has become the basis for much of our mathematical analysis of related representations. In particular, we have shown that the "evolution property of zero-crossings," the famous property that says that in scale-space zero-crossing contours are never created as one passes from fine to coarse resolution, is mathematically equivalent to the Maximum Principle for parabolic equations [5].

Zero-crossings in scale-space have become a popular proposed representation, especially since the zero-crossings are correlated with edge information. One difficulty with zero-crossings is that they do not vary continuously with the image data; a second problem, as we have established, is that the representation is unstable. Nonetheless, by working on reconstructions from zero-crossings, we have been able to establish that the information carried in the zero-crossings is a rich description of image information, and that the reconstructions contain many recognizable features of the original data. This suggests that the zero-crossings enhanced with slightly more information should suffice for a good representation.

Indeed, in more recent experiments, we have established that stable reconstruction is possible when zero-crossings are enhanced with gradient data along the zero-crossings. The fact that such a representation is complete was conjectured by Marr, established theoretically by us in a paper a couple years ago, but only established in stable numerical experiments by us this past summer.

It is also possible that the structure can be simplified to make for a more useful image representation, using alternatives to zero-crossings altogether. Specifically, the visual cortex seems to have cells that respond to local bandpass filters of visual data, and to directional derivatives of that data. By sampling data and derivatives of data in a Laplacian-of-Gaussian scale-space, we believe that an effective representation can be constructed. Others have similarly been concentrating on "oriented pyramids," for example using the "wavelet" transformation, and we find much of this work to be persuasive. We expect to use our reconstruction techniques and mathematical analytical methods to study these and related representations. Ultimately, we seem to be focusing on a representation involving sampling in scale-space, and cortical computations for reconstruction of a representation that encodes the complete Laplacian pyramid.

1.1.3. Kinetic depth effect and depth perception

Another way to probe the information that is retained by our image analysis system is to determine the cues that are important for certain perceptual effects. Professor Michael Landy, in collaboration with post-docs and students, has extended understanding of the kinetic depth effect, by conducting many experiments varying motion cues and evaluating depth perception. A key to this work is our ability to measure the perceptual effectiveness of the kinetic depth effect in particular psychophysical experiments, based on a collection of three-dimensional shapes. It is important to isolate the cues presented to subjects in order to assess independently the information that is used in the computation of shape parameters. For example, we are able to remove local dot density as a cue, and still retain shape identification from moving dots. Further, we have shown that while visual motion is the essential input stimulus that is responsible for the kinetic depth effect, the optic flow perception need not be computed by a Fourier energy detection system, such as the Reichardt model. Moreover, pairs of features do not seem to be important in depth perception. We conclude that the kinetic depth effect is based on a representation of optical flow that is preattentive and based on global computations. We are thus developing a model of optic flow computation feeding a model for generating a kinetic depth effect [6].

Another interest has related to depth perception from stereo. The cooperative psychology and computer science group has begun to develop a multiresolution model of stereo perception. However, the model suggested a number of experiments for psychophysical investigation, and these experiments have been conducted by students in our Psychology department. At interest is the kinds of interpolations that are used when the stereo information is sparse. By discovering that smooth interpolants seem to dominate, we have constrained models for stereo perception.

1.1.4. Knowledge aggregation

A large body of research in AI is concerned with the combination of knowledge from different sources of information. In computer vision, cues can come from edge data, texture, orientation, dynamic processing of temporal data, color, models, and other sensory sources. To make inferences from disparate sources of knowledge requires a representation of information in a fashion that permits incremental modification.

We have long been associated with relaxation labeling methods for image analysis [7]. We have developed a number of relaxation labeling models, and continue to show their effectiveness at handling low-level visual tasks. However, there are many related alternatives to relaxation labeling, and we have pointed out relationships between relaxation labeling, stochastic relaxation methods, brain state models, neural networks, and the Dempster/Shafer theory of evidence [8].

The latter topic, the "theory of evidence," has a large and dedicated set of advocates in the AI community. By bringing a mathematical and statistical viewpoint to this field, and by contrasting the methods involving the theory of evidence to other knowledge aggregation methods, we have been able to explain the foundations of the Dempster rule of combination, and show that the basis of the formula is Bayesian combination of opinions, where the state of the system is represented by the statistics of more than one opinion. Using this viewpoint, we are able to suggest alternative formulations, which end up looking remarkably like Kalman filtering [9, 10]. One of our extensions, presented at the last IJCAI meeting, incorporates a notion of parameterized independence, relaxing normal assumptions of complete (conditional) independence.

1.2. Computational neuroscience

Professor Eric Schwartz directs a large group concerned with computational neuroscience. Using studies of the visual cortex of monkeys, complex patterns of functional areas have been identified. In this project, we are especially concerned with the mappings along the visual pathway, and the algorithms that are suggested by the data structures created by the precise organization of information. Attempts at understanding the nature of visual cortex pose a wide range of problems in computer graphics, image processing, computational geometry, and numerical methods. In a series of studies, computer graphics and image processing methods have been used to develop accurate three-dimensional models of the retinotopic map, and to represent these maps by numerically flattening layers using a method of minimal metric error [11, 12]. Using results obtained from these methods and dioxyglucose studies, subsequent work has suggested a number of algorithmic methods that arise from the functional neuroanatomy. In particular, a novel computational method for stereopsis has been suggested based upon the striations in primary visual cortex [13, 14]. A shape representation scheme has also been presented, and recent work centers on pattern recognition interpretations of cortical functional maps.

1.3. Representing shape

In addition to studying representations of grayscale image data, members of the Courant Institute Robotics Laboratory have been concerned with more symbolic specifications of image constructs. In particular, we have been concerned with the specification of two and three-dimensional shape information. In this study, our principle evaluation criterion for a shape representation is its effectiveness in an object recognition system.

For description of 3-D shapes in a manufacturing environment, it is reasonable to assume the existence of 3-D depth data, obtained from a depth sensor. In this regard, the NYU Robotics Laboratory has pursued the development of a novel light-striped depth sensor, yielding simultaneous intensity and range data [15, 16]. Using depth data extracted by this depth sensor, and also by the laser-based "White Scanner," descriptions of 3-D objects have been developed.

For 2-D object description, Yaron Menczel in his thesis work demonstrated that shape information can be effectively encoded by a graph structure that is derived from the orientation of boundary points [17]. Specifically, each point in a region is labeled with an orientation tag based on the orientation of the nearest boundary point, and the orientation field is quantized to produce regions of similarly labeled points. The resulting graph is used for matching and recognition, and proved successful for applications such as character recognition with variable fonts and connected letters.

A principle motivation in the study of shape information is that shape identification is possible in the presence of occlusion and obscuration. Thus it is evident that shape analysis for this purpose should be local, enabling partial matching techniques. Since 2-D objects are fully described by their boundary curves, both globally and locally, and 3-D objects may also be represented by sets of *characteristic curves*, (e.g. ridges, curves of sharp intensity change, curves of specularity), considerable effort has been done to develop efficient curve representation and matching algorithms both in 2-D and 3-D (using range data, as described above). This effort was initiated by the work of Professors J. T. Schwartz and M. Sharir and their coworkers, and has been carried on and expanded in the last two years by Dr. H. J. Wolfson and graduate students associated with the Lab [18-22]. A landmark in application of this method in the 2-D case, and a clear demonstration of its sensitivity and robustness, was its use to assemble (graphically rather than physically) all the pieces of two intermixed hundred-piece commercial jigsaw puzzles from separate photographs of the individual pieces [23].

The work is founded on a new technique for geometrically hashing two-dimensional and three-dimensional curves [24-26]. Curves are represented by local features that are invariant to rigid transformations, and feature values are used to generate an attribute of a curve called its *footprint*, which enables us to efficiently index the appropriate local information to the object for recognition purposes.

This effort has been recently extended to local representation of 2-D curves in an affine invariant way [27]. We will elaborate on this issue in Section 2.

In a separate but related project, some particularly elegant work on affine invariant *global* shape representation has been completed recently by Professor J. Hong and Dr. X. Tan, who are currently visiting our Lab [28].

Finally, we have applied an interest in parallel algorithms to shape analysis. Much work in parallel image processing focuses on SIMD architectures, where many simple processors perform the same function distributed over an image. However, it is clear that in biology, there is a diversity of function in massive parallelism, so that models of multiple program streams (MIMD) are in a sense more realistic. We have done a limited amount of work on connected component algorithms for image analysis, focusing on issues of MIMD parallelism [29,30]. NYU has a large group of researchers involved in parallel algorithm development, parallel architecture studies, and parallel system design. In particular, the "Ultracomputer" project works closely with IBM's RP3 project to develop an MIMD shared-memory machine with a combining network, together with an highly parallel operating system based on Berkeley UNIX. Many image processing and vision research projects (for example, the matching algorithms based on the *footprint* technique), will be greatly facilitated by the accessibility of this unique parallel machine.

1.4. Model-based vision

Our work on representation and description of scenes has led to a large program of work on object recognition using the techniques of model-based vision. A number of methods progressing along complementary lines have been developed.

The curve matching techniques using *footprint* representations, which were mentioned in the previous section, led to the development of an experimental 2-D object recognition system enabling us to recognize overlapping 2-dimensional objects selected from large databases of model objects without significant performance degradation as the size of the data base increases [24]. Experimental results from databases of size about 100 make this technique appear quite promising.

A complete, working model-based system, SCERPO, was developed by Professor David Lowe while at NYU [31]. Several graduate students at NYU continue working on this system, under direction of Professor Lowe (although David has now returned to the University of British Columbia). For example, Robert Goldberg is extending SCERPO to the case where the models have articulation joints [32].

The existing, functioning, SCERPO system is one of the first to demonstrate the recognition of three-dimensional objects in single images taken from arbitrary viewpoints, incorporating fast matching methods using a grouping strategy. The objects are represented as polyhedral solids. Features are extracted from the scene by means of low-level edge feature extraction [33], followed by simple grouping and feature description operations. Matching is done by solving for the three-dimensional position and orientation of an object directly from two-dimensional image measurements, using an iterative, hill-climbing technique to find the best viewpoint parameters that will "project" the object model onto the locations of matched image features. An important point is that the evaluation of the quality of a match, is done in the projected, two-dimensional image space. Thus methods to recover 3-D object shape from image cues are unnecessary. Once a few initial matches have been formed, it can make quantitative predictions for the exact locations of further model features in the image. This provides a reliable method for evaluating the correctness of a match according to whether it is consistent with a single viewpoint [34].

A second aspect of the SCERPO research concerns the problem of perceptual organization. Human vision is able to detect many different types of significant groupings of image elements, such as parallelism, collinearity, proximity, or symmetry in an otherwise random set of image features. This perceptual organization capability has been missing from most computer vision systems. Since these image groupings reflect viewpoint-invariant aspects of a three-dimensional scene, they are ideal structures for bridging the gap between the two-dimensional image and the three-dimensional model. Probabilistic measures have been developed for evaluating the significance of instances of each of these image relations that can arise from projective invariance. The SCERPO system uses these significance measures to prioritize prototype matches for object recognition [35].

Yet another project in model-based object recognition from single 2-D images is being directed by Dr. Haim Wolfson. This system is based on affine invariant point, line, and curve matching, and uses the affine approximation of the viewing transformation to facilitate efficient matching procedures. The system will be able to deal efficiently with both polyhedral and non-polyhedral scenes with considerable occlusion. Some of the algorithms have been already successfully tested in recognition of flat objects in 3-D scenes from an arbitrary viewpoint [27,36], and it is currently being extended to enable recognition of general 3-D objects. This work will be addressed in detail in Section 2.

2. Efficient matching

We now present a design and results of work by H. Wolfson, in collaboration with H. Lamdan, and J. Schwartz, on object recognition in the presence of arbitrary affine transformations of the models.

We develop new techniques for model-based recognition of 3-D objects from unknown viewpoints. The method is especially useful for recognition of scenes with overlapping and partially occluded objects. An efficient matching algorithm, which assumes affine approximation to the perspective viewing transformation, is proposed. The algorithm has an off-line model preprocessing phase and a recognition phase to reduce matching complexity. The algorithm has been successfully tested in recognition of flat industrial objects appearing in composite occluded scenes.

2.1. Introduction

Recognition of industrial parts and their location in a factory environment is a major task in robot vision. Most practical vision systems are model-based systems (see the survey in [37]). Object recognition using model-based vision presents many challenges in image understanding, but offers the possibilities of well-formulated tasks and rigorous algorithm evaluation.

We consider the object recognition problem, where the vision system is faced with a composite scene of overlapping parts (thus partially occluding each other), taken from a data-base of known objects. The task is to recognize the objects in the scene and to specify their location and orientation.

No restriction on the viewing angle of the camera is assumed. We begin by considering the recognition of flat objects arbitrarily positioned in space. At the end of this section, we discuss the use of these methods for the general case of 3-D objects. The recognition is done from 2-D intensity images. The algorithms that we describe have been actually tested for the recognition of objects comprising composite scenes of industrial tools, such as pliers, wrenches, etc., (see Figs. 4-8).

Since we are concerned with recognition of partially occluded objects, the use of global features is precluded. Accordingly, we must describe our objects by a set of local features. This same conclusion is applicable to the human vision system, which is also capable of recognition in the presence of considerable occlusion. The local features can be points, line segments, curve segments, borders, or other structures developed from local description operations. Initially, we restrict ourselves to the use of special points, which we denote as *interest points*. The point sets of the various model objects are matched against the point set of the composite overlapping scene using a small number of corresponding points. Once a prototype correspondence is established, we find the best transformation in least-squares sense to establish the correct position of the model object in the scene image. A key aspect of our scheme is its computational efficiency, based upon a division into a preprocessing stage and a recognition stage. Our model point sets are preprocessed off-line independently of the scene information, thus enabling an efficient on-line recognition stage. A major advantage of the proposed matching algorithm is the ease with which both the preprocessing and recognition stages can be parallelized.

The problem of object recognition in 2-D scenes is a common one [24,38-41]. Three-dimensional object recognition systems are discussed in [37,42]. Recent image understanding results not mentioned in the above surveys include [31,43,44].

The method described here differs from other existing model-based matching systems. Our method, which uses a hashing scheme indexed on the affine transformation and model type, is more algorithmic and more parallelizable than Lowe's SCERPO system [31]. In [44], a clustering approach is used to discover the transformation between the model and the scene images. The hashing scheme here is more efficient and more predictable. In [43], there is an emphasis on the classification of the model and image features to reduce the complexity of matching, while the matching algorithm itself is straightforward. We, on the other hand, consider the case where no such effective classification can be done (this is also the assumption in [44]), and, hence, our emphasis is on the development of an efficient feature matching algorithm, which processes the models and the scene images independently allowing fast recognition. In case feature classification is possible it can be incorporated in our algorithm in a natural way to improve its efficiency.

2.2. Definition of the Problem

We initially assume that we view partially occluded flat objects from an arbitrary viewpoint. These initial assumptions are similar to those in [43]. We also assume that the depth of the centroids of the objects in the scene is large compared to the focal length of the camera, and that the depth variation of the objects are small compared to the depth of their centroids. Under these assumptions it is well known that the perspective projection is well approximated by a parallel (orthographic) projection with a scale factor (see for example p.79 in [45]). Hence, two different images of the same flat object are in an affine 2-D correspondence: namely there is a non singular 2×2 matrix $\mathbf{A}$ and a 2-D (translation) vector $\mathbf{b}$, such that each point $\mathbf{x}$ in the first image is translated to the corresponding point $\mathbf{Ax} + \mathbf{b}$ in the second image.

Our problem is to recognize the objects in the scene, and for each recognized object to find the affine transformation that gives the best least-squares fit between the model of the object and its transformed image in the scene.

2.3. Choice of 'Interest Points'

The matching algorithm, which is described in the next section, is based on matching 'interest points', extracted in both the scene image and the model. These should be database dependent, so that different databases of models will suggest different features for 'interest points'. For example, a data base of polyhedral objects naturally suggests the use of polyhedra vertices as 'interest points', while 'curved' objects suggest the use of sharp convexities, deep concavities and, maybe, zero curvature points. 'Interest points' do not have to appear physically in the image. For example, a point may be taken as the intersection of two non-parallel line segments, which are not necessarily touching. An 'interest point' does not necessarily have to correspond to a geometrical feature. The Moravec 'interest operator', based on high variance in intensity, is described in [46] and was used in [47].

The problem of extracting stable, useful 'interest points' is a delicate topic equivalent, in many ways, to the shape representation problem. Although a successful approach to this problem is required for any model-based vision system, we will assume here that a sufficient number of stable points can be extracted from the relevant images. Our emphasis then, in this paper, is on the matching problem, and not on the model representation or image description.

In our experiments with 2-D objects, we used points of sharp convexities and deep concavities along the borders (see Figs. 4c, 4d, 5b, 6b).

2.4. Recognition of a Single Model in a Scene

For the sake of clarity we describe our algorithm in the simpler situation, where the database consists only of one model. However, the presentation given here applies to the general case where a number of models may appear in the scene.

It is well known that an affine transformation of the plane is uniquely defined by the transformation of three non-collinear points (see, for example, [48]). Moreover, there is a unique affine transformation, which maps any non-collinear triplet in the plane to another non-collinear triplet. Hence, we may extract "interest points" on the model and the scene, and try to match non-collinear triplets of such points to obtain candidate affine transformations. Each such transformation can be checked by matching the transformed model against the scene. This is also the basic approach in [43].

However, the complexity of such a scheme is quite unfavorable. Given m points in the model and n points in the scene, the worst case complexity is $(m \times n)^3 \times t$, where t is the complexity of matching the model against the scene. If we assume that m and n are of the same magnitude, and t is at least of magnitude m, the worst case complexity is of order n^7. One way to reduce this complexity ([43]) is to classify the points in a distinctive way, so that each triplet can match only a small number of other triplets. We consider, however, the situation were such a distinction does not exist or cannot be made in a reliable way (see [44]). Hence, we present a more efficient triplet matching algorithm. Our method has the advantage that when distinguished points are classifiable, or when the transformation can be restricted to a smaller class, the complexity will be reduced.

The algorithm consists of two major steps. The first one is a preprocessing step which is applied to the model points. This step does not use any information about the scene and is executed off-line before actual matching is attempted. The second step, matching proper, uses the data prepared by the first step to match the models against the scene. The execution time of this second step is the actual recognition time.

In order to separate the computation into two such independent steps, we have to represent the model and scene point information in a way that is both independent and still allows comparison of corresponding structures.

The crucial observation is that once an affine basis is specified by a triplet of non-collinear points, then the coordinates of all the other points, given in the coordinate system of the triplet, are affine invariant. That is, if e_{10}, e_{01}, and e_{00} are three non-collinear points, then any other point v, with coordinates (ξ,η):

$$v = \xi(e_{10} - e_{00}) + \eta(e_{01} - e_{00}) + e_{00}$$

will still have coordinates (ξ,η) if the entire figure is translated by the affine transformation T:

$$Tv = \xi(Te_{10} - Te_{00}) + \eta(Te_{01} - Te_{00}) + Te_{00}$$

assuming the same triplet of points, now Te_{00}, Te_{10}, Te_{01} are chosen as the basis.

Accordingly, our data structure for representing a given object will be based on a collection of mappings from the set of all non-collinear triplets into a list of quantized coordinate pairs. Actually, we will form a hash table, where each quantized coordinate pair, i.e., a box represented by (ξ,η), contains a list of all object models and their basis triplets that contained an interest point that mapped to that box.

Our algorithm will efficiently compare these sets of coordinates belonging to different bases. The algorithm is as follows:

(A) Preprocessing

Assume we are given an image of a model, where m 'interest points' have been extracted. For each ordered non-collinear triplet of model points, the coordinates of all other $m-3$ model points are computed taking this triplet as an affine basis of the 2-D plane. Each such coordinate (after a proper quantization) is used as an entry to a hash-table, where we record the number of the basis-triplet with which the coordinates were obtained and the number of the model (in case of more than one model). The complexity of this preprocessing step is of order m^4 per model. New models added to the data-base can be processed independently without recomputing the hash-table.

(B) Recognition

In the recognition stage we are given an image of a scene, where n 'interest points' have been extracted. We choose an arbitrary ordered triplet in the scene and compute the coordinates of the scene points taking this triplet as an affine basis. For each such coordinate we check the appropriate entry in the hash-table, and for every pair *(model, basis-triplet),* which appears there, we tally a vote for the model and the basis-triplet as corresponding to the triplet which was chosen in the scene. (If there is only one model, we have to vote for the basis triplet alone).

If a certain pair *(model, basis-triplet)* scores a large number of votes, we decide that this triplet corresponds to the one chosen in the scene. The uniquely defined affine transformation between these triplets is assumed to be the transformation between the model and the scene. If the current triplet does not yield a model and triplet that scores high enough, we pass to another basis-triplet in the scene.

For the algorithm to be successful it is enough, theoretically, to pick any three non-collinear points in the scene belonging to one model. The voting process, per triplet, is linear in the number of points in the scene. Hence, the overall recognition time is dependent on the number of model points in the scene, and the number of additional 'interest points' which belong to the scene, but did not appear on any of the models. Although, in the worst case, we might have an order of n^4 operations, in most cases, especially when the number of models is small, the algorithm will be much faster. For example, if there are k model points in a scene of n points, then the probability of not choosing a model triplet in t trials is approximately

$$p = (1-(\frac{k}{n})^3)^t$$

Hence, for a given $\epsilon>0$, if we assume a lower bound on the 'density' $d=\frac{k}{n}$ of model points in a scene, then the number of trials t giving $p<\epsilon$ is of order $\frac{\log\epsilon}{\log(1-d^3)}$, which is a constant independent of n. Since the verification process is linear in n, we have, in this case, an algorithm of complexity $O(n)$, which will succeed with probability of at least $1-\epsilon$.

This method assumes no *a-priori* classification of the model and scene points to achieve matching candidates. If such information is available, it can be incorporated into our method by assigning weights to the correspondence of different triplets to the model, and by checking the triplets in an appropriate order.

Numerical errors in the point coordinates are more severe when the basis points are close to each other compared to the other model points in the scene. To overcome this problem, we may introduce the following procedure. If a certain basis triplet gets a number of votes, which, on one hand, are not enough to accept it as a 'candidate' basis, but, on the other hand, do not justify total rejection, we may change this triplet to another triplet consisting of points that were among the 'voting' coordinate pairs, and are more distant from each other than the previous basis points. In the correct case this procedure will result in a growing match, as the numerical errors become less significant. Even if a basis-triplet belonging to some model did not get enough votes due to noisy data, we still have chance to recover this model from another basis-triplet.

A major potential advantage of the suggested algorithm is its high inherent parallelism. Parallel implementation of this algorithm is straightforward.

2.5. Finding the Best Least-Squares Match

Suppose that for a particular basis triplet chosen in the scene, a high number of votes are obtained for a given *(model, basis-triplet)*. Each vote implies the existence of a match (by close proximity) of an interest point in the scene with some point in the specified model. This match is valid for the affine transformation that maps the basis triplet in the scene to the basis triplet of the model receiving many votes. We can then improve this affine transformation, and potentially find more matches by finding the optimal affine transformation for the set of matched points. This is efficiently accomplished if the measure of optimality is the sum of square distances in errors of the match (details are given in [36]). Other measures are also possible.

We incorporated this process of affine transformation improvement in our experiments. In Fig. 6c we see an example of a fit obtained by calculating the affine transformation from three basis points, and in Fig. 6d the same model is fitted using the best least-squares affine match, based on 10 points, all of which, by the way, were recovered as corresponding points by the transformation in Fig. 6c.

2.6. Summary of the Algorithm

Our algorithm can be summarized as follows:

(A) Represent the model objects by sets of 'interest points'.

(B) For each non-collinear triplet of model points compute the coordinates of all the other model points according to this basis triplet and hash these coordinates into a table which stores all the pairs *(model, basis-triplet)* for every quantized coordinate pair.

(C) Given an image of a scene extract its interest points, choose a triplet of non-collinear points as a basis triplet and compute the coordinates of the other points in this basis. For each such quantized coordinate pair, vote for the pairs *(model, basis-triplet)*, that appear in the hash table at that location and find the pairs which obtained the most number of votes. If a certain pair scored a large number of votes, decide that its model and basis triplet correspond to the one chosen in the scene. If not, continue by checking another basis triplet.

(D) For each candidate model and basis triplet from the previous step, establish a correspondence between the model points and the appropriate scene points, and find the affine transformation giving the best least-squares match for these corresponding sets. If the least-squares difference is too big go back to Step (C) for another candidate triplet. Finally, the transformed model is compared with the scene (this time we are considering not only previously extracted 'interest points'). If this comparison gives a bad result go back again to Step (C). (In our experiments we compared the boundaries of our objects at equally spaced sample points.)

This is a short summary of the basic algorithm. Of course, various improvements can be incorporated in its different steps. We discuss a number of possibilities in the next section.

2.7. Reduction of Complexity using Affine Invariants

When the number of 'interest points' on the models is large, various affine invariants can be exploited to reduce the complexity of the method presented in Section 4. We give one such example. We will use the following observation (see, for example, p.73 in [45]). Two straight lines which correspond in an affine transformation are 'similar', i.e. corresponding segments on the two lines have the same length ratio. The same statement holds for sets of parallel lines. Hence, if we have a set of points, which are located on parallel lines in a model, and another set of points on parallel lines in the scene, we can efficiently check the conjecture that some of these points correspond.

Let us see how the previous method can be modified for the case when the 'interest points' lie on a collection of lines. We have again two major steps.

(A) Model Preprocessing

Extract the 'interest points' on the model and group the points into a collection of lines. (A point may belong to different lines.) Take an ordered pair of points on a line and compute the coordinates of all other model points on this line taking this pair as the standard one-dimensional basis of the line. Each such coordinate is used as an entry to a hash-table, where we record the number of the basis-pair at which the coordinate was obtained, the number of the line, and the number of the model.

(B) Recognition

Extract sets of points positioned on the same line in the image. Choose a pair of points on such a line as a basis and compute the coordinates of the other points on the same line according to this basis. For each such coordinate check the appropriate entry in the hash-table, and vote for every triple of *(model, line, basis-pair)*, which appears there. A triple that scores a large number of votes gives the correspondence between the points on the appropriate lines. Correspondence of three non-collinear points (obtained from different lines, of course), already gives a full affine basis, and we proceed as before.

The worst case complexity of this approach is less by a factor of n, since we now are iterating over pairs of points, as opposed to triples of points in the scene. The expected complexity is also less, since we are more likely to choose a pair of points belonging to a single model, over choosing a triplet within a single model.

A different reduction in complexity occurs if we are confronted with the problem of recognition of objects that have undergone a similarity transformation, i.e., rotation, translation, and scale. This is the situation when the viewing angle of the camera is the same both for the model and the image of a scene. Such conditions can be achieved, for example, in a factory environment where the viewing angle of a camera on a conveyor belt can be kept constant.

Our algorithm is obviously applicable without modification to the case of a similarity transformation, since it is a special case of an affine transformation. However, the complexity of both the preprocessing and recognition stage can be reduced. The key observation here is that since the similarity transformation is orthogonal, two points are enough to form a basis which spans the 2-D plane. (The first point is assigned coordinates (0,0) and the second (1,0). The third basis point (0,1) is uniquely defined by these two points.) Hence, we may repeat the procedure described in Section 4 using basic pairs instead of basic triplets. This reduces the complexity of the preprocessing step by a factor of m, and the worst case complexity of the recognition step by a factor of n.

2.8. Line Matching

In the previous sections we dealt with point matching algorithms. However, extraction of points might be quite noisy. A line is a more stable feature than a point. Thus in scenes were lines can be extracted in a reliable way, e.g. scenes of polyhedral objects, we might be interested to apply similar procedures to lines.

All the point matching techniques given above apply directly to lines, since lines can be viewed as points in the dual space. Thus three lines that have no parallel pairs are a basis of the affine space; each line has unique coordinates in this basis, and we repeat exactly the same matching procedure. We can also make use of line segments to reduce the complexity of the point matching when lines can be stably extracted from the scene. We omit the details here.

2.9. Curve Matching

In this section, we extend the methods of the previous sections to the case where the extracted features are no longer simple 'interest points,' but instead are entire boundary curves. Since the shape of planar rigid bodies is completely described by their boundary, object recognition can be accomplished by matching these curves. Matching of curves that have undergone affine transformation was discussed in the works of Cyganski and Orr ([49,50]). Another elegant global curve matching method was recently developed by J. Hong and X. Tan [28]. Their methods, however, require knowledge of the full curve, and hence are unable to deal with occlusion. The method described in this section is based on local affine invariant features enabling recognition of partially occluded objects.

The curves are conveniently represented by vertices of their polygonal approximations. Ideally, the extraction includes a smoothing process, such as the one described in [18]. We discuss separately the cases of non-convex and convex curves.

2.9.1. Non-convex Curve Matching

As was pointed out in [51], non-convex planar regions are sometimes easier to handle than convex regions. In the case of an affine transformation, each concavity supplies us with a stable feature from which the affine transformation can be recovered. Specifically, consider the sketch of Fig. 1. The concavity depicted there is bounded by a single segment of the convex hull which we call the *concavity entrance*. It is a simple geometric observation that the concavity entrance is invariant under affine transformation. An additional point which is invariant under affine transformations is the concavity point most distant from the concavity entrance line. (If this point is not unique, we may choose the leftmost.) Thus, one can extract a concavity-based point triplet which is affine invariant. This basis triplet can be used in a recognition scheme similar to the point matching scheme.

The concavity entrances are computed as follows. First the convex hull of a polygonal approximation of the boundary curve is computed. The concavity entrance endpoints are those convex hull point pairs which are separated by polygon vertices not belonging to the convex hull. The computation of the (leftmost) boundary point most distant from the concavity entrance is simple. The complexity of the entire process is $O(n)$, where n is the number of polygon vertices (see [52], p.93).

Figure 1. A concavity entrance and basis triplet

The procedure for recognition of partially occluded non-convex objects in composite scenes may proceed exactly in the same way as as in the previously described point matching algorithm (see Section 2.6). Here, however, the the complexity is highly reduced, since we consider only basis triplets that are *concavity*-based. Moreover, since concavities may be differentiated by their shape even in the affine invariant case, we may further reduce the complexity of the algorithm by comparing only *basis triplets* based on affine invariantly similar concavities. To accomplish it we introduce a numerical affine invariant *shape characteristic* that we call a *footprint*. The footprint should be a continuous, stable, and easily computed representation of the concavity shape. To compute the footprint, we first normalize a concavity by applying the transformation which maps its triplet basis to a standard equilateral triangle. That is, the concavity endpoints are mapped to $(-1,0)$, $(1,0)$, and the third point to $(0,\sqrt{3})$. To each such normalized shape we assign a vector of numbers that we call the 'footprint.' One of the footprint schemes that we use is illustrated in Fig. 2. For some constant s (say $5 \le s \le 10$), we divide the upper half plane by $s+1$ rays based at the origin, with angle $\frac{\pi}{s}$ between two consecutive rays. Let a_i be the area of the 'normalized' shape between rays i and $i+1$. The footprint will be s-vector $(a_1, a_2, \cdots, a_s)$, where each component is quantized into one of a number of discrete bins.

Figure 2. The footprint of a concavity

We now proceed as before, constructing a hash table. Each footprint is used as an entry to the hash table, where the *model* and *concavity* numbers are recorded. In the recognition phase, each concavity is used to compute a footprint, and the appropriate entry in the hash table is accessed. For each pair, *(model, concavity)*, appearing in the hash table at that location, we compute the appropriate affine transformation to the associated model, and attempt to verify an instance of the model in the image.

If the concavity entrances are distinctive enough, the complexity of the recognition stage will be this time linearly dependent on the number of concavities in the scene and the number of scene vertex points, namely, $O(k \times n)$.

2.9.2. Convex Curve Matching

In case we have a model with a convex boundary curve, or if we wish to recognize non-convex models with all concavity entrances occluded, a different method is needed. (It is interesting to point out that for these cases, the point classification method of [43] is also unapplicable.) We must resort to a strategy with a greater time complexity, since there is no 'natural' affine base, as the one defined by a concavity. Specifically, to construct the hash table for a given convex model, we iterate over all pairs of boundary points. For each pair, we join the two points with a line, which will be called the 'base line.' There are two most distant curve points from the base line, one on each side of the base line. (If this point on one side is not defined uniquely, take the leftmost such point). Each most distant point together with the endpoints of the base line form an affine basis triplet. To each such basis triplet corresponds a convex region bounded by the base line and the convex body, containing all three basis points. This is the 'basis region' (see Fig. 3).

As before, we can use a footprint based on the normalized basis region to create an entry to a hash table. In this case, for each pair of boundary points, we have a separate footprint. Thus the hash table

Figure 3. A basis triplet of a convex object.

entries contain the identification of the model and the identification of the basis triplet. For recognition, we judiciously choose a pair of points in the scene on the boundary of a convex curve, find an associated convex 'basis region,' and compute the footprint. For this footprint (properly quantized), we check the appropriate entry in the hash table, and extract the pairs *(model, basis triplet)* appearing there. For each such relevant model with the appropriate basis triplet, we compute the corresponding affine transformation between the model and the scene, and verify their correspondence.

Observe that convex bodies usually intersect at concave angles (in [24] they are called breakpoints). Thus for the recognition step, it will be enough to examine only one pair of points (one base line) for each convex 'protrusion', delimited by two consecutive breakpoints. Hence, if a scene has k convex boundary subcurves (delimited by consecutive breakpoints), the recognition stage of the algorithm wil be of the order $k \times n$, where n is the number of object vertices. That is, the recognition phase will be very efficient.

2.9.3. 3-D objects

The methods described in this paper naturally extend to the recognition of 3-D objects. In the case where accurate 3-D depth data is available, an entirely similar procedure can be developed based on extraction and matching of points in 3-D, subject to a rigid transformation. Here, we discuss the more general problem of recognition of 3-D objects from a single 2-D view. We assume that the variation in depth of the objects is small compared to the depths of the object centroids, so that the perspective projection is well-approximated by an affine transformation.

We distinguish three approaches to the problem. The appropriate method will depend upon the complexity of the models, and the robustness of the feature extraction.

If the objects can be approximated by polyhedral solids, then we may build a database of 2-D models representing the 'almost' planar faces of each 3-D model. The problem then reduces to recognition of these flat surfaces, according to the methods of the previous sections. The faces may be partly obscured by the presence of other 3-D objects in the line-of-sight to the object. Complete identification of the object and its orientation can be verified by the consistent identification of other faces of the 3-D object in the appropriate locations [34].

Alternatively, we may discretize the space of viewing directions, and produce a nearly flat model of each 3-D object from each given viewing direction. Recognition proceeds by identifying objects and the viewing direction among the database of models, which may have been subjected to a similarity transformation. In this approach, there will be many models, but due to the fact a similarity transformation is sought rather than an affine transformation (Section 2.7), there will be a reduced complexity in the recognition phase.

Finally, in the same way that a triplet of points in the 2-D case can be used as an affine basis, four points on the surface of a 3-D model, providing they are non-coplanar, define a 3-D basis. Suppose that we choose four points in the scene, and posit a match with four corresponding points in a model. The correspondence defines a 3-D affine transformation of the object. The match can be verified quickly, by checking whether other points of the model appear in the scene according to the affine transformation. The checking can be made faster by a hashing scheme (although, one that is different than the hashing method presented earlier). However, methods to speed the search for an appropriate set of four points in the scene can contribute greatly to the efficiency of this approach.

3. Experimental Results

We have implemented the point matching, non-convex curve matching and best least-squares matching algorithms.

In the first set of figures we show recognition of industrial parts (pliers) in composite scenes. Here the point matching algorithm and least squares matching algorithm were applied. Figs. 4a and 4b are the original gray scale images of two models (pliers), and Fig. 4c and 4d show the extracted 'interest points' of the models, which are points of sharp concavities and convexities. In Fig. 6a we see an image of the pair of pliers of Fig. 4a rotated, translated and tilted at about 40 degrees (observe the different lengths of both handles in the image). The recognition algorithm was performed to obtain a number of matching basis-triplets. The corresponding affine transformations were calculated and for each such transformation the transformed model was superimposed on the scene of Fig. 6a. Fig. 6c shows such a transformation computed according to a basis triplet which gives a somewhat noisy match. This solution is significantly improved by the best least-squares match which is given in Fig. 6d, and was calculated using all the points which were recognized as model pointsby the basis triplet of Fig. 6c.

In Fig. 5 we see an image of a composite overlapping scene of the two pliers, the extracted 'interest points', and the recognition results. Note that in Fig. 5b we have additional 'interest points' that do not correspond to 'interest points' in the original models, but are created by the superposition of the two objects. Also, one can see that a number of the original 'interest points' are occluded in the scene.

The second set of figures deals with recognition of some household items. Fig. 7a is the original gray scale image of a pizza cutter. In Fig. 7b the concavity entrances are marked by the dashed lines, and the concavity basis triplets are displayed. Fig. 8a is a composite scene of the pizza cutter and a spatula. The image was taken by a significantly tilted camera resulting in an affine distortion of the model. In Fig. 8b the concavity entrances of the composite scene are marked, and the basis triplet points are displayed. The algorithm of Section 2.9.2 was applied to this scene, resulting in the recognition of the pizza cutter displayed in Fig. 8c.

Acknowledgements

Work described in this paper was supported by Office of Naval Research Grants N00014-82-K-0381 and N00014-85-K-0077 Work Unit NR-4007006, and National Science Foundation Grants NSF DCR-83-20085 and IRI-8703335.

References

[1] Hummel, Robert A., B. Kimia, and S. Zucker, "Deblurring gaussian blur," *Computer Vision, Graphics, and Image Processing* **38**, pp. 66-80 (1987).

[2] Hummel, Robert and Robert Moniot, "Solving ill-conditioned problems by minimizing equation error," *Proceedings of the IEEE First Insternational Conference on Computer Vision*, pp. 527-533 (1987).

[3] Hummel, Robert and Robert Moniot, "A network approach to reconstructions from zero-crossings," *Proceedings of the IEEE Computer Society Workshop on Computer Vision*, pp. 8-13 (November, 1987).

[4] Hummel, Robert, "The scale-space formulation of pyramid data structures," pp. 107-123 in *Parallel Computer Vision*, ed. Len Uhr, Academic Press, New York (1987).

[5] Hummel, Robert, "Representations based on zero-crossings in scale-space," *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, pp. 204-209 (June, 1986).

[6] Landy, M., "A parallel model of the kinetic depth effect using local computations," *Journal of the Optical Society of America (A)* 4, pp. 864-876. (1987).

[7] Hummel, Robert A. and Steven W. Zucker, "On the foundations of relaxation labeling processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-5, pp. 267-287 (May, 1983).

[8] Landy, Michael S. and Robert A. Hummel, "A brief survey of knowledge

aggregation methods," *Proceedings of the International Conference on Pattern Recognition*, pp. 248-252 (October, 1986).

[9] Hummel, Robert A. and Michael S. Landy, *A statistical viewpoint on the theory of evidence*, To appear: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. March, 1988.

[10] Hummel, Robert and M. Landy, "Evidence as opinions of experts," *Proceedings of the "Uncertainty in AI" Workshop*, pp. 135-143 (August 8-10, 1986).

[11] Wolfson, Estarose and Eric L. Schwartz, "Computing minimal distances on polyhedral surfaces ," *IEEE Pattern Analysis and Machine Intelligence*, (1988). In press.

[12] Schwartz, Eric L., Alan Shaw, and Estarose Wolfson, "A numerical solution to the generalized map makers problem," *IEEE Pattern Analysis and Machine Intelligence*, (1988). In press.

[13] Yeshurun, Yehezkel and Eric L. Schwartz, "Cepstral filtering on a columnar image architecture: a fast algorithm for binocular stereo segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , (1988). In press.

[14] Yeshurun, Yehezkel and Eric Schwartz, "Shape description with a space variant sensor: algorithms for scan-path,fusion and convergence over multiple scans," *IEEE Pattern Analysis and Machine Intelligence* , (1988). In press.

[15] Bastuscheck, M. and J. Schwartz, "Experimental implementation of a ratio image depth sensor," pp. p. 1-12 in *Techniques for 3-D Machine Perception*, ed. A. Rosenfeld, Elsevier (North-Holland) (1986).

[16] Carrihill, Brian and Robert A. Hummel, "Experiments with the intensity ratio depth sensors," *Computer Vision, Graphics, and Image Processing* **32**, pp. 337-358 (1985).

[17] Menczel, Y., "Description of shape using orientation and propogation flow," Ph.D. Thesis, Dept. of Computer Science, NYU (Oct., 1986).

[18] Schwartz, J.T. and M. Sharir, "Identification of Partially Obscured Objects in Two Dimensions by Matching of Noisy 'Characteristic Curves'," *The Int. J. of Robotics Research* **6** (2), pp. 29-44 (1987).

[19] Bastuscheck, C. M., E. Schonberg, J.T. Schwartz, and M. Sharir, "Object Recognition by 3-Dimensional Curve Matchin," *Int. J. of Intelligent Systems* **1**(2), (1986).

[20] Wolfson, H., "On Curve Matching," *Proceedings of the IEEE Computer Society Workshop on Computer Vision*, pp. 307-310 (November, 1987).

[21] Schwartz, J. and H. Wolfson, "Improved shape-signature and matching methods for model-based robotic vision," *Proceedings of the Workshop on Space Telerobotics* **II**, pp. 103-109 (Jan., 1987). NASA, JPL

[22] Kishon, E., "Simultaneous matching of curves in three dimensions," NYU Computer Science TR 263 (Nov., 1986).

[23] Wolfson, H., E. Schonberg, A. Kalvin, and Y. Lamdan, "Solving Jigsaw Puzzles Using Computer Vision," *Annals of Operations Research*, (1987). In Press.

[24] Kalvin, A., E. Schonberg, J.T. Schwartz, and M. Sharir, "Two Dimensional Model Based Boundary Matching Using Footprints," *The Int. J. of Robotics Research* **5**(4), pp. 38-55 (1986).

[25] Hong, J. and H. J. Wolfson, "An Improved Model-Based Matching Method Using Footprints," *Robotics Report 137*, Computer Science Div., Courant Inst. of Math., NYU, (1987).

[26] Kishon, E. and H. Wolfson, "3-D Curve Matching," *Proceedings of the AAAI Workshop on Spatial Reasoning and Multisensor Fusion*, pp. 250-261 (October, 1987).

[27] Lamdan, Y., J.T. Schwartz, and H. J. Wolfson, "Object Recognition by Affine Invariant Matching," *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, (June, 1988). To appear.

[28] Hong, J. and X. Tan, "The similarity between shapes under affine transformation," NYU Robotics Research Report 133 (Dec., 1987). Submitted.

[29] Hummel, Robert A., "Connected component labeling in image processing with MIMD architectures," in *Intermediate-level Image Processing*, ed. M. J. B. Duff, Academic Press, Bonas, France (1986).

[30] Hummel, Robert and Kaizhong Zhang, "Dynamic processor allocation for parallel algorithms in image processing," *Proceedings of the Optical and Digital Pattern Recognition session of the SPIE Conference on EO-Imaging, SPIE Vol. 754*, pp.

268-275 (January, 1987).

[31] Lowe, D.G., *Perceptual Organization and Visual Recognition,* Kluwer Academic Publishers (1985).

[32] Goldberg, R. and D. Lowe, "Verification of 3-D parametric models in 2-D image data," *Proceedings of the IEEE Computer Society Workshop on Computer Vision*, pp. p. 255-257 (Nov., 1987).

[33] Hummel, Robert and David Lowe, "Computing large-kernel convolutions of images," NYU Robotics Technical Report 84 (1986).

[34] Lowe, D.G., "The Viewpoint Consistency Constraint," *Int. J. of Computer Vision* **1** (1), pp. 57-72 (1987).

[35] Lowe, D., "Three dimensional object recognition from single two-dimensional images," *Artificial Intelligence* **31**, pp. 355-395 (1987).

[36] Lamdan, Y., J. T. Schwartz, and H. Wolfson, "On recognition of 3-D objects from 2-D images," *Proceedings of the IEEE Int. Conference on Robotics and Automation*, (April, 1988). To appear.

[37] Chin, R.T. and C.R. Dyer, "Model-Based Recognition in Robot Vision," *ACM Computing Surveys* **18** (1), pp. 67-108 (1986).

[38] Ayache, N. and O.D. Faugeras, "HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects," *IEEE TPAMI* **8** (1), pp. 44-54 (Jan., 1986).

[39] Bolles, R.C. and R.A. Cain, "Recognizing and Locating Partially Visible Objects: The Local Feature Method," *The Int. J. of Robotics Research* **1** (3), pp. 57-82 (1982).

[40] Turney, J.L., T.N. Mudge, and R.A. Volz, "Recognizing Partially Occluded Parts," *IEEE TPAMI* **7** (4), pp. 410-421 (July 1985).

[41] Grimson, W.E.L. and T. Lozano-Perez, "Model Based Recognition and Localization from Sparse Range or Tactile Data," *The Int. J. of Robotics Research* **3** (3), pp. 3-35 (1984).

[42] Besl, P.J. and R.C. Jain, "Three-Dimensional Object Recognition," *ACM Computing Surveys* **17** (1), pp. 75-154 (1985).

[43] Huttenlocher, D.P. and S. Ullman, "Object Recognition Using Alignment," *Proc. of the 1'st Int. Conf. on Computer Vision*, pp. 102-111 (1987).

[44] Thompson, D.W. and J.L. Mundy, "Three-Dimensional Model Matching from an Unconstrained Viewpoint," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 208-220 (1987).

[45] Klein, F., *Elementary Mathematics from an advanced standpoint ; Geometry*, Macmillan, New York (1925 (Third edition)). (English translation)

[46] Moravec, H., "Towards automatic visual obstacle avoidance," *Proceedings of the 5th IJCAI*, p. p. 584 (1977).

[47] Barnard, S. and W. Thompson, "Disparity analysis of images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, p. 333 (1980).

[48] Yaglom, I.M. and V.G. Ashkinuze, *Ideas and Methods of Affine and Projective Geometry*, (in Russian) 1962.

[49] Cyganski, R. and J.A. Orr, "Applications of Tensor Theory to Object Recognition and Orientation Determination," *IEEE PAMI* **7** (6), pp. 662-673 (Nov. 1985).

[50] Cyganski, R., J. Orr, T. Cott, and R. Dodson, "Development, implementation, testing, and application of an affine transform invariant curvature function," *Proceedings of the 1st Int. Conf. on Computer Vision*, pp. 496-500 (1987).

[51] Schwartz, J.T. and M. Sharir, "Some Remarks on Robot Vision," *Trans. of 3'rd Army Conf. on Applied Math. and Computing*, pp. 1-36 (May 1985).

[52] Mehlhorn, K., *Multi-dimensional Searching and Computational Geometry*, Springer (1984).