2

ETL-0549

Dynamic Image Interpretation for Autonomous Vehicle Navigation, Final Report

Edward M. Riseman
Allen R. Hanson

University of Massachusetts
Computer & Information Science Department
Amherst, Massachusetts 01003

August 1989

Approved for public release; distribution is unlimited.

Prepared for:

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209-2308

U.S. Army Corps of Engineers
Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060-5546

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
ETL-0549

6a. NAME OF PERFORMING ORGANIZATION
University of Massachusetts

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
U.S. Army Engineer Topographic Laboratories

6c. ADDRESS *(City, State, and ZIP Code)*
Computer & Information Science Department
Amherst, Massachusetts 01003

7b. ADDRESS *(City, State, and ZIP Code)*
Fort Belvoir, VA 22060-5546

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Defense Advanced Research Projects Agency

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DACA76-85-C-0008

8c. ADDRESS *(City, State, and ZIP Code)*
1400 Wilson Boulevard
Arlington, Virginia 22209-2308

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
62301E			

11. TITLE *(Include Security Classification)*
Dynamic Image Interpretation for Autonomous Vehicle Navigation -- Final Report

12. PERSONAL AUTHOR(S)
Edward M. Riseman and Allen R. Hanson

Autonomous navigation, [illegible] vehicles,

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 2/26/85 TO 7/12/89

14. DATE OF REPORT *(Year, Month, Day)*
1989 August

15. PAGE COUNT
59

16. SUPPLEMENTARY NOTATION
Previous Report in Series: ETL-0516. Dynamic Image Interpretation for Autonomous Vehicle Navigation, 1987 End of Year Technical Report

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Scene Interpretation, Spatial Reasoning,
Sensor Motion
(SDW)

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This report presents the results of the Dynamic Image Interpretation for the Autonomous Vehicle Navigation project for the time period 2/26/85 to 7/12/89. The purpose of the project is to develop algorithms and tools to enable a robotic ground vehicle to navigate autonomously through realistic landscapes.

In this final annual report, we summarize our accomplishments in constructing robust algorithms to be used for vehicle navigation as well as tools that have been developed to more efficiently utilize these algorithms. K[illegible] (S)

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Linda H. Graff

22b. TELEPHONE *(Include Area Code)*
(202)355-2818

22c. OFFICE SYMBOL
CEETL-RI-T

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Abstract

This report presents the results of the Dynamic Image Interpretation for the Autonomous Vehicle Navigation project from the time period 2/26/85 to 7/12/89. The purpose of the project is to develop algorithms and tools to enable a robotic ground vehicle to navigate autonomously through realistic landscapes.

In this final annual report, we summarize our accomplishments in constructing robust algorithms to be used for vehicle navigation as well as tools that have been developed to more efficiently utilize these algorithms.

Contents

List of Figures

List of Tables

Preface

This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and monitored by the U. S. Army Engineer Topographic Laboratories (ETL) under Contract DACA76-85-C-0008, titled "Knowledge-based Vision Techniques-Task D." The DARPA Program Manager is LTC Robert Simpson and the ETL Contracting Officer's Representative is Ms. Linda Graff.

Summary

Over the course of this contract from February 26, 1985 to July 12, 1989, our research has fallen into two broad categories: Motion and Mobile Robot Navigation. We first summarize our work on motion and then that on mobile robot navigation.

Work on Motion

Our research on motion has led us to develop a variety of motion algorithms, and in most cases, apply them to real-world image sequences including the domains of robot arm workspaces, indoor hallways, and outdoor sidewalk/road scenes.

Anandan constructed an algorithm for determining feature point correspondences between frames that allowed the computation of dense displacement fields with associated confidences. The algorithm can also be used to effectively track points during motion. Glazer developed two algorithms for the efficient computation of image motions using hierarchical multiresolution methods operating over image data pyramids. Adiv developed an algorithm (to date, the only one that exists) for general sensor motion (five degrees of freedom) in an environment with objects undergoing independent general motion. He also analyzed the conditions under which the determination of these motion parameters would be ambiguous. In related work, Snyder analyzed the effects of uncertainty in the location of the FOE and of feature points in the image on the computation of depth, and showed how this analysis could be used to provide *quantitative* predictions for constraining the search window used for matching these points in future frames. He also analyzed the relative efficacy of motion and stereo for depth computations.

Much of our work has centered on the recovery of depth from assumed translational motion. Pavlin developed an efficient algorithm for extracting the focus-of-expansion (FOE) from a sensor undergoing pure translational motion (i.e., two degrees of freedom) to an accuracy of a few degrees. Bharwani, et al. used Pavlin's work to develop a multi-frame algorithm for depth extraction under known translational motion which iteratively predicted the image motion of a feature point in future frames, determined correspondence by a search over the limited predicted area, and then refined the depth estimate using the new match. Difficulties with this algorithm led us to develop a general motion algorithm by combining the optical flow computation of Anandan and the motion parameter estimation component of Adiv's algorithm. This algorithm seems to be able to predict depth with an error of about 10%.

Other techniques we developed to extract depth from motion are those due to Balasubramanyam, Snyder, and Weiss using stereoscopic motion, Pavlin using assumptions of constant general motion, Williams and Hanson using grouped geometric structures, and Sawhney and Oliensis using the image traces of points undergoing purely rotational motion. A further aspect of our research has been the collection of extensive motion data with ground truth of known precision. These data were collected on the Autonomous Land Vehicle (ALV) at Martin Marietta's Denver, Colorado test site, and are presently available

to the general vision community.

Work on Mobile Robot Navigation

In the pas‘ ıobile robots have been *constrained* to operate in either an indoor or an outdoor enviru..ment, but not both. Special purpose representations and *ad hoc* sensor techniques geared toward tasks of narrow focus have dominated these efforts. Our mobile robot effort has addressed the problem of enabling a mobile automaton to navigate intelligently through indoor and outdoor environments.

Our first attempt to construct such a "cosmopolitan" robot was the development of the Autonomous Robot Architecture (AuRA) by Arkin which makes use of a "meadow" map for global path planning. This map serves as the robot's long term memory and contains imbedded *a priori* knowledge to guide sensor expectations.

Arkin's work has been used by Fennema to further investigate the problem of navigating intelligently through arbitrary environments. He uses model–based processing of the visual sensory data as the primary mechanism for obstacle avoidance, movement through the environment, and measuring progress towards a given goal. The modular building blocks of the system include the planning and plan monitoring modules, a set of vision modules, a 3–D modelling system, a 2–D feature matching and fitting system, and finally a 3–D pose refinement system for updating the robot's location and orientation.

The world model is developed in a 3–D solid modelling package, GeoMeter, developed by Connolly, Weiss, et al. GeoMeter serves as a system for representing both polyhedral solid objects (such as buildings) in terms of basic geometrical entities such as vertices, faces, and edges, as well as curved surfaces. It has been used to construct a 3–D model of both indoor and outdoor environments.

An important problem in model–driven 3–D interpretation is how to use approximate knowledge of the location and orientation of the sensor, models of objects in the environment, and the results of low–level vision to determine the *image–to–model correspondence*. The approach we have taken is to separate 2–D model–to–image matching from the determination of the 3–D pose parameters. Mechanisms for optimal 2–D model matching, used to locate landmarks derived from the world model and to estimate the robot's current position, are the subject of research by Beveridge, et al., who determine correspondences between the model and the data lines such that an optimized spatial fit will produce the lowest match error. Methods for determining the "pose," i.e., the position and orientation, of the robot with respect to a world coordinate system have been developed by Kumar.

The successes in actual robot experimentation to date at UMass have been modest, but are increasing in power and robustness, and are beginning to have real significance. Successful navigation of both an outdoor sidewalk and an indoor hall using the approaches of Fennema, Beveridge, Kumar, et al. has been achieved. The algorithm is quite robust working with (unchanging) environments in the presence of significant path edge discontinuities (doorways, vehicle tracks, clutter etc.). To date, obstacle avoidance on vehicle runs has been handled using ultrasonic data. Dead-reckoning information is used minimally in our system as our goal is to serve as a testbed for vision algorithms.

Many of the issues involved in the mobile vehicle research can be seen as complementary to those of other areas in our vision and robotics groups. The use of perceptual and motor schemas in the proposed vehicle architecture exploits many of the concepts used in both the VISIONS scene interpretation group and the work being done in the Laboratory for Perceptual Robotics' distributed programming environment. Multi-sensor integration, certainly crucial for the vehicle's domain, will benefit from the work being done on the integration of vision, touch, and force sensing. Our research on developing parallel implementations of robust vision algorithms is certainly synergistic with our development of parallel architectures for real time vision processing.

1 Introduction

One of the key features of an object that usually distinguishes it from other objects in the environment is its movement relative to them. Even when an object is camouflaged by its similarity in appearance to other objects, any independent movement of the object immediately gives it away. In addition, if there is relative movement between the camera and the object, the viewer is automatically provided with several distinct views of the object and therefore with 3D structures and their dynamic characteristics.

The two most common methods of obtaining two images from two distinct views are **stereopsis and motion**. Stereopsis is when two images are obtained simultaneously by two cameras. Motion is when several images are taken one after another by a single camera moving with respect to the environment. In most applications of stereopsis, it is common to orient the cameras such that their image planes are perpendicular to the ground plane and their optical axes are parallel to each other. Usually the displacement between the camera locations is horizontal and parallel to the image plane.

Given two images obtained from either stereo or motion, the task is to combine them to provide 3D information about the objects in the image. The process usually consists of two stages – the establishment of the *correspondence* between the points in the two images to provide a *disparity* and then a *depth* map, followed by some process that uses the depth information to discover and describe the surfaces in the 3D environment.

Before we proceed further, we define a few key terms. The *correspondence problem* is the task of identifying events in the two images as images of the same event in the 3D environment. The *disparity* is the distance between the locations in the two images of the two corresponding events. When the optical axes are parallel to each other, the *depth* of a point is its distance along the optical axis from the image planes.

Motion processing can be broadly divided into two categories:

1. the camera moves and the environment is stationary, and

2. there are independently moving objects in the scene.

The first case is easier to analyze and process, as can be seen from the large number of techniques that have been developed for this purpose.

The most common approach taken towards motion analysis is one in which the processing proceeds bottom-up. The movement of individual points in the images is computed first, followed by a process that determines the motion of the camera, as well as the location, 3D structure, and motion of the objects in the scene.

One important term used in motion research is **optical flow**. Optical flow can be broadly defined as the vector field representing the changes in the positions of the images of environmental points over time. Strictly speaking, it is necessary to distinguish between the *optical flow*, which is the field of instantaneous 2D velocity vectors of the points in the image on the image plane, and the *displacement field*, which is the field of discrete displacement vectors connecting the location of the same image-point in successive image frames. However, when the time interval between the frames is small enough, the displacement field is a good approximation to the optical flow. The usual approach to motion analysis consists of two steps—the computation of optical flow followed by its interpretation to provide the 3D structure and motion of the objects in the scene as well as the motion of the camera. The computation of optical flow is similar to the correspondence problem mentioned earlier. In fact, it is common to regard the correspondence problem in stereopsis as a special case of motion correspondence. However, in stereopsis, the knowledge of the relative locations of the cameras constrains the search for corresponding points in a manner that is not possible in motion analysis. Finally, we mention one important limitation of current approaches to motion analysis. Most of the techniques for motion analysis deal with only two frames. Some initial approaches to multi-frame analysis are described in the body of this report.

Identifying image "events" that correspond to each other is the primary task of both motion and stereo analysis. The term "events" is used here in a broad sense, to mean any identifiable structure in the image - e.g., image intensities in a neighborhood, edges, lines, texture markings, etc.

The techniques that rely on the similarity of the light intensity reflected from a scene location in the two frames as the basis for determining correspondence are called *intensity-based* approaches. Methods that identify stable image structures, and use them as tokens for finding correspondences are referred to as *token-based* approaches.

The most popular way of solving the correspondence problem is to divide it into one or two parts. The first is the local correspondence problem, which provides partial or complete constraints on the displacement of a point in the image, based on image information in the immediate neighborhood of that point. Usually the local correspondence is solved (partially or fully) *independently* at all points of interest in the image. The second part, where used, consists in applying a non-local constraint on the flow field. This is usually an assumption of the spatial smoothness of the flow field, or one that is derived from the geometry of rigid bodies in motion. This constraint can be either global or semiglobal, depending on whether or not explicit boundaries are recognized, across which the constraint is not allowed to propagate.

It is also possible to impose on top of this framework for the computation of displacement fields, a multi-frequency, multi-resolution approach. In this approach the images are pre-processed with a set of band-pass filters which are spatially local and which decompose the spatial frequency-spectrum in the image in a convenient way. The outputs from the corresponding filters applied to the two images are matched, and the matching results from the different filters at the same location in the image are combined using a consistency constraint.

The primary goal of motion analysis is to determine the 3-dimensional structure of the objects in the environment and the relative movement of the camera and the objects in the scene. The determination of the 3-dimensional image displacements or velocities of the image-points is only one (although an important one) of the steps involved. The interpretation of the displacement (or velocity) fields to determine the 3D structure of the environment and the relative 3D motion between the objects and the camera is another important step.

In this report we describe the techniques and algorithms we have developed for using motion analysis to determine environmental structure and sensor motion. We also describe how we have used these techniques in concert with other methods developed in our group to address the problem of intelligently navigating an autonomous mobile robot through a 3D environment.

2 Motion Research

2.1 The Reliable Computation of Optical Flow: A Smoothness Constraint and a Confidence Measure

Although our hierarchical correlation algorithm [40] for the computation of dense displacement fields proved to be an efficient and reliable technique, there are still a number of situations where the algorithm makes mistakes. These situations arise in areas of the image without significant intensity variations and at occlusion or motion boundaries. Our previous work [5] attempted to identify such situations through the use of a confidence measure which indicated the reliability of a match vector. The recent work of Anandan uses a relaxation process to improve matches with low confidence based on neighbouring matches with higher confidences.

In his recently completed doctoral dissertation [8], Anandan provides a unified framework for extracting a dense displacement field from a pair of images, as well as an integrated system based on a matching approach. This framework appears to be sufficiently general to encompass both gradient–based and correlation–matching approaches. It consists of a hierarchical scale-based matching scheme using bandpass filters, orientation-dependent confidence measures, and a smoothness constraint for propagating reliable displacements. His integrated system for the extraction of displacement fields uses the minimization of the sum-of-squared-differences (SSD) as the local match-criterion, computes confidence measures based on the shape of the SSD surface, and formulates the smoothness assumption as the minimization of an error functional. This overcomes many of the difficult problems that exist with other techniques.

The SSD measure which is to be minimized is expressed as

$$SSD(x_0, y_0; \delta_x, \delta_y) = \sum_{i,j=-n}^{n} W(i,j)\,(I(x_0+i, y_0+j) - J(x_0+\delta_x, y_0+\delta_y))\,.$$

Here I and J are the intensity functions describing the first and second images, respectively, W is a weighting function, n is the radius of the match window, and δ_x and δ_y are the x– and y–components, respectively, of the displacement of the pixel located at (x_0, y_0) in the first image. In practice, W is taken to be a Gaussian, and n is chosen to be 2.

The error functional consists of two terms: one, called the approximation error, measures how well a given displacement field approximates the local match estimates; the other, called the smoothness error, measures the global spatial variation of a given displacement field. The finite-element method is used to solve the minimization problem. The approach also gives information for extracting occlusion boundaries in some situations.

The confidence measure that was described in [5] was a scalar value between 0 and 1 that indicated the reliability of the displacement vector at a pixel in the image. One such value was provided for each pixel. This measure was derived by studying the properties of the error-surface obtained during the process of computing the displacement at a pixel. However, the image displacement vector is a 2–D quantity. Hence, it is appropriate to have a 2–D confidence measure associated with the displacement vector.

In his previous work [5], Anandan observed that the error-surface allowed us to distinguish between situations in which completely reliable information regarding the displacement vector (i.e., at high curvature points along image contours) is available, those in which we have only partial information (i.e., at edge locations where only the displacement perpendicular to the edge can be reliably measured), and situations where there is no reliable information (i.e., at homogeneous intensity areas of the image). The new confidence measure is a vector quantity which uses these distinctions.

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

- The new confidence measure is best described as a two-dimensional vector. It is convenient to describe the vector in terms of the local orthogonal basis vectors $\hat{e}_{max}$ and $\hat{e}_{min}$, which are the principal directions for the SSD surface. The displacement vector D can be decomposed in terms of its components along these basis vectors, and confidence measures c_{max} and c_{min}, given by the principal curvatures of the SSD surface, are associated with these components. The details of their computation are given in [6]. It is worthwhile to note that these are no longer bound to be between 0 and 1. The formulation of the smoothness constraint described below requires that these values be allowed to vary between 0 and ∞.

- The process of improving an unreliable match estimate based on its neighbours is formulated as a smoothness constraint on the displacement vector field. The smoothness constraint consists of two errors, E_{smooth} and E_{approx}, whose sum is minimized. E_{smooth} measures the spatial variation of the displacement field, i.e., the smoother the variation, the smaller the error. It is taken to be:

$$E_{smooth}(\{\mathbf{U}\}) = \int\int \left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right) dx\, dy,$$

where $\{\mathbf{U}\}$ is the set of displacement vectors $\mathbf{U}(x,y) = (u(x,y), v(x,y))^T$, derivatives are represented by $u_x = \partial u/\partial x$, etc., and the integration is over the whole image. E_{approx} measures the deviation of the smooth displacement field from the initial field provided by the matching process:

$$E_{approx}(\{\mathbf{U}\}) = \sum_{x,y} \left[c_{max}(\mathbf{U}\cdot\hat{e}_{max} - \mathbf{D}\cdot\hat{e}_{max})^2 + c_{min}(\mathbf{U}\cdot\hat{e}_{min} - \mathbf{D}\cdot\hat{e}_{min})^2\right].$$

The definition of this error makes it clear that the low confidence estimates are allowed to vary more than the high confidence estimates. Hence, the smoothing process modifies the initial displacement values at locations of low confidence measures more than those at the locations of high confidence measures.

The smoothness constraint translates into a minimization problem which is solved using the finite-element method, since this permits the inclusion of known discontinuities in the displacement field. The application of this method leads to a local relaxation algorithm, which iteratively updates the displacement vector field [8].

Anandan has also shown that the functional minimization problem formulated in his matching technique converges to the minimization problem used in gradient-based techniques (e.g., Glazer's technique discussed in the next section). In particular, by relating an approximation error functional used in his matching approach to the intensity constraints used in the gradient-based approaches, he explicitly identifies confidence measures which have thus far been implicitly used in the gradient-based approach. Finally, he suggests the ways that algorithms operating on a pair of frames can be developed into multiple-frame algorithms, and discusses their relationship to spatio-temporal energy models. Anandan's algorithm has been applied to many image sequences. In Figure 1, we show a pair of images, in which both the camera and the dinosaur have moved independently from one image to the next.

In Figure 2, we show the corresponding optical flow determined by Anandan's algorithm.

2.2 Glazer's Hierarchical Algorithms

Glazer's recently completed thesis [41] presents an approach to motion detection using multi-resolution methods in a hierarchical processing architecture. Two motion detection algorithms are developed and analyzed. The hierarchical correlation algorithm utilizes

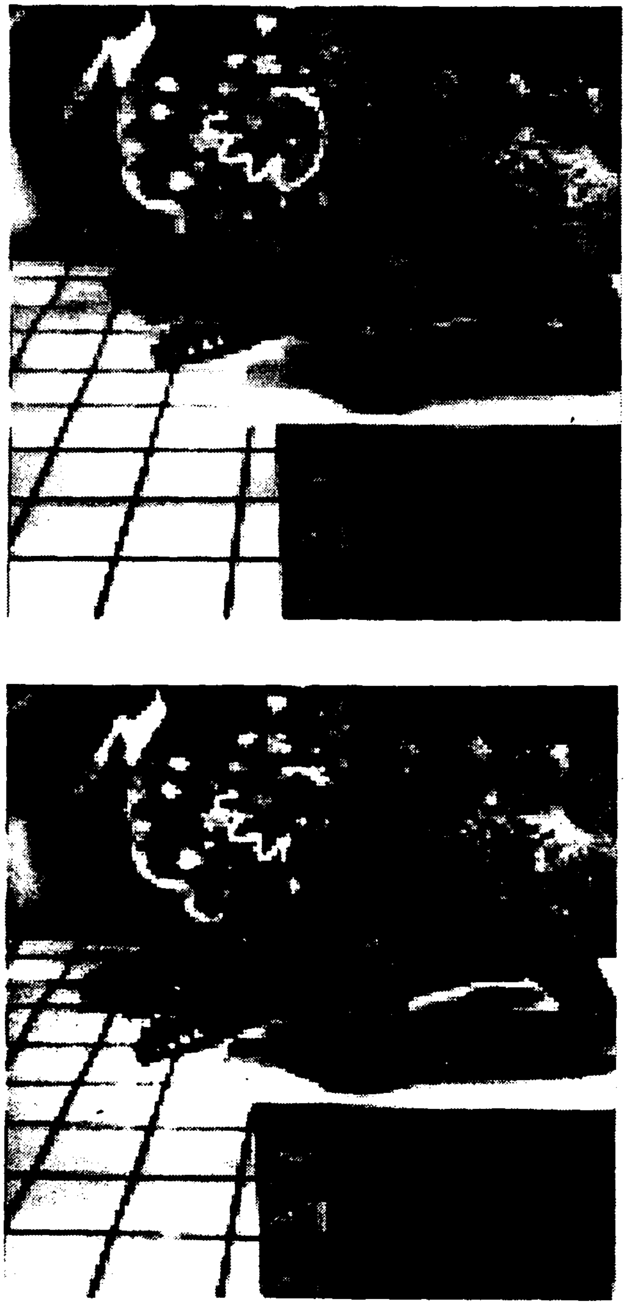


Figure 1: The Dinosaur–Image Experiment.
The input images (128 × 128), with Frame 1 at top, Frame 2 at bottom. The camera motion is a translation to the right, along with a rotation about the vertical axis. The independent moticn of the dinosaur is primarily rotational about the vertical axis.

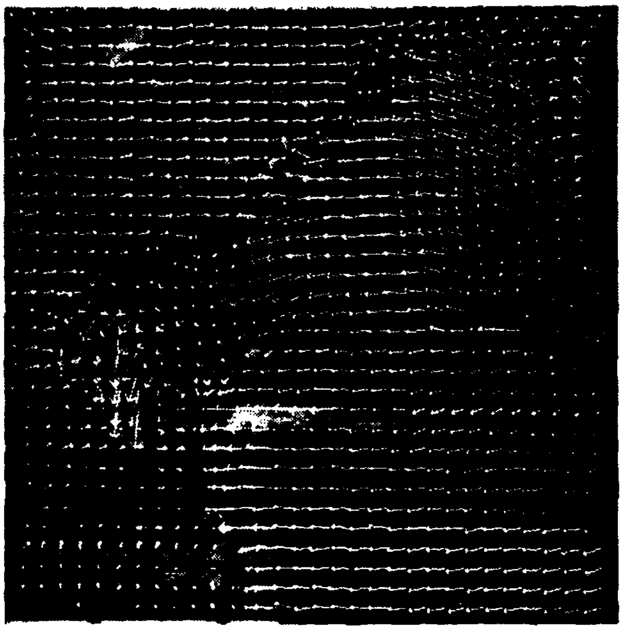


Figure 2: Displacment Field Using Anandan's Algorithm.
The smoothed displacement vector field computed using Anandan's algorithm for the dinosaur-image, superimposed on Frame 1. In order to enhance visibility, only a 32 × 32 sample of the displacement is shown.

a coarse-to-fine control strategy across the resolution levels and overcomes two disadvantages of single–level correlation: large search areas requiring expensive searches, and repetitive image structures which cause incorrect matches. The hierarchical gradient-based algorithm [42], generated over low-pass image pyramids, extends single-level gradient algorithms to the computation of large displacements. Within each level, the next refinement of the displacement field is obtained by combining a local intensity constraint and a global smoothness constraint. The mathematical formulation involves the minimization of an error functional consisting of two terms, corresponding to the intensity and the smoothness constraints mentioned above. The minimization problem is solved using the finite-difference approach which leads to a multi-resolution relaxation algorithm. A formal analysis of the hierarchical gradient algorithm is presented, including the basic equations for computing a refined disparity vector, the discrete representations and computations for solving these equations, and a geometric interpretation of the resulting relaxation algorithm. The experimental results show that the two algorithms have comparable accuracy and a cost analysis shows that the hierarchical gradient algorithm is less costly.

2.3 The Computation of General Motion for Independently Moving Objects from Optical Flow

The segmentation of an image into independent objects is one of the most difficult problems in computer vision. Adiv [1,2] has developed an algorithm which performs this segmentation when the objects are independently moving. His algorithm has two main stages. In the first stage, the optical flow field (obtained, e.g., via Anandan's algorithm) is partitioned into connected segments of flow vectors, where each segment is consistent with a rigid motion of a roughly planar surface. Such a segment is assumed to correspond to part of only one rigid object. This initial organization of the data is utilized in the second stage without the assumption that the surfaces are planar. Segments are then grouped under the hypothesis that they are induced by a single rigidly moving object and/or by the sensor motion. This is done by computing the optimal motion parameters and related error measure for each segment by employing a least–squares approach that minimizes the deviation between the measured flow fields and that predicted from the estimated motion and structure. Based on the fundamental equations for optical flow:

$$\begin{aligned} u &= \frac{T_X - xT_Z}{Z} - \Omega_Z y + \Omega_Y(1+x^2) - \Omega_X xy \\ v &= \frac{T_Y - yT_Z}{Z} + \Omega_Z x - \Omega_X(1+y^2) + \Omega_Y xy, \end{aligned}$$

the error function to be minimized is:

$$\begin{aligned} \sum_{i=1}^{n} \ W_i \ & [\Big(\alpha_i - \frac{T_X - x_i T_Z}{Z_i} + \Omega_Z y_i - \Omega_Y(1+x_i^2) + \Omega_X x_i y_i\Big)^2 + \\ + \ & \Big(\beta_i - \frac{T_Y - y_i T_Z}{Z_i} - \Omega_Z x_i + \Omega_X(1+y_i^2) - \Omega_Y x_i y_i\Big)^2], \end{aligned}$$

where the translation vector is (T_X, T_Y, T_Z), the rotation vector is $(\Omega_X, \Omega_Y, \Omega_Z)$, and for each i between 1 and n (α_i, β_i) is the optical flow vector computed at pixel (x_i, y_i), with W_i its weight. Z_i is the spatial depth of the corresponding environmental point. The task is to find the translation, rotation, and spatial depth which minimize this function. This step essentially involves grouping segments of the flow field which are consistent with the same motion parameters. Therefore the output of Adiv's algorithm is a set of object masks, as well as the motion parameters of each of these independent objects. Numerous experiments with real data show this algorithm to have quite good performance. In Figure 3, we show the results of Adiv's algorithm when applied to the image pair of Figure 1. We recall that there is general, independent motion of the objects which are imaged. In this example, we see good qualitative agreement between the segmentation of the image using Adiv's algorithm, and the actual objects in the scene.

2.4 Inherent Ambiguity in the Motion Analysis of Noisy Flow Fields

Owing to the presence of noise and other image imperfections, the optical flow in an image sequence will not be exact. The work of Adiv [3,4] mathematically examines the robustness of algorithms which compute general motion from optical flow. The analysis focuses on ambiguities that are inherent in the sense that they are true of all algorithms, and can only be resolved if constraining assumptions or other sources of visual information are employed.

Two sources of ambiguity which arise from noisy flow fields are examined. The first ambiguity is in recovering the motion parameters from a noisy flow field generated by a rigid motion. Motion parameters of the sensor or a rigidly moving object may be extremely difficult to estimate because there may exist a large set of significantly incorrect solutions which induce flow fields similar to the correct one. Adiv shows that if the field of view corresponding to the region containing the interpreted flow field is small, and the depth variation and translation magnitude are small relative to the distance of the object from the sensor, then the determination of the 3-D motion and structure can be expected to be very sensitive to noise and, in the presence of a realistic level of noise, practically impossible. He also experimentally found that there was a relationship between the location of the focus of expansion (FOE), the point where the sensor velocity vector intersects the image plane, and the degree of ambiguity.

The second ambiguity is in the decomposition of the flow field into sets of vectors corresponding to independently moving objects. Two independently moving objects may induce optical flows which are compatible (modulo the noise) with the same motion parameters; hence, there is no way to refute the hypothesis that these flows are generated by one rigid object. Adiv shows that the standard rigidity assumption [61] is not appropriate for noisy flow fields. He proposes that a weaker assumption is more effective, namely that a connected set of flow vectors, consistent with a rigid motion of a *planar* surface, is induced by a rigid motion.

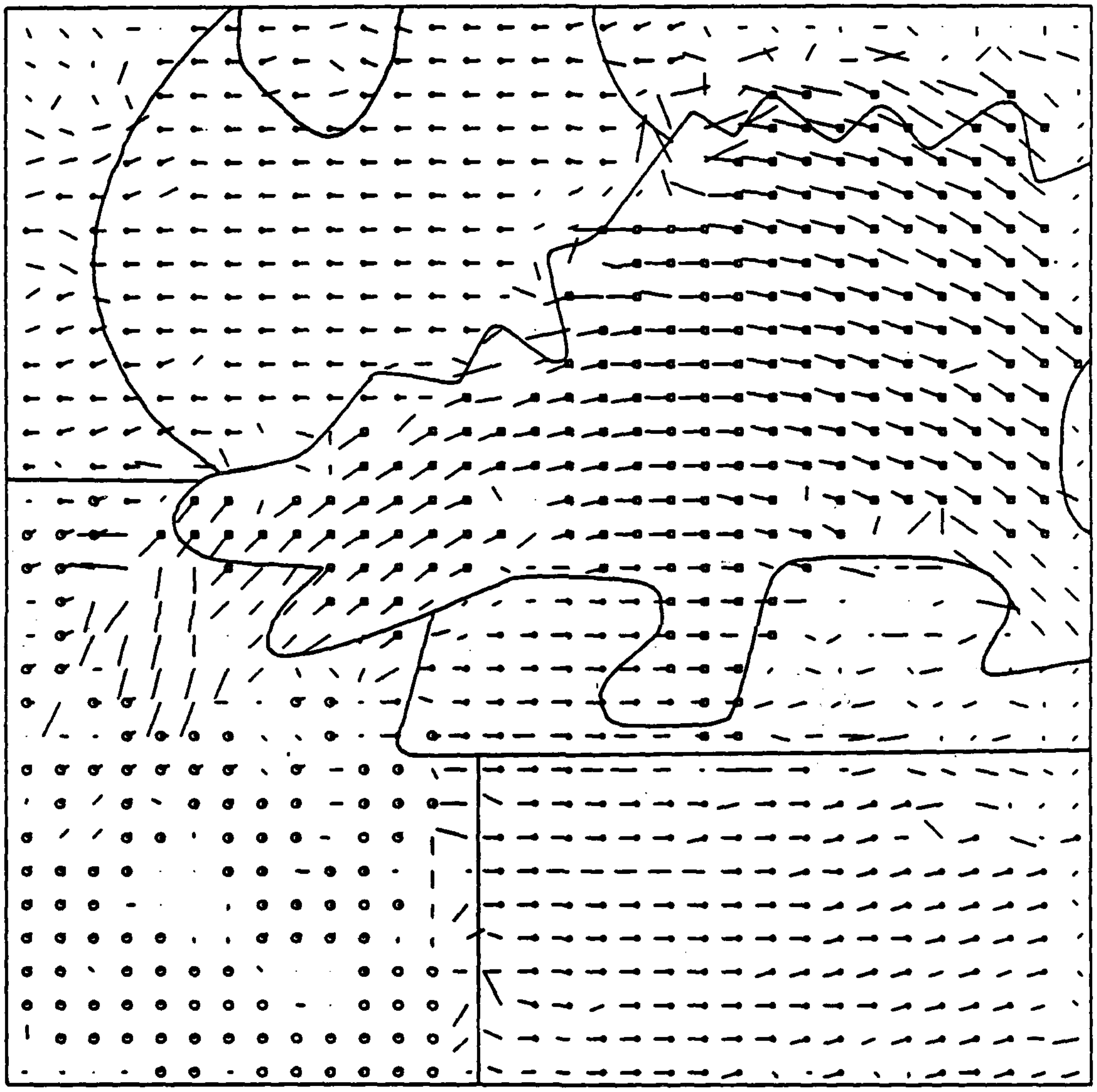


Figure 3: Adiv's Algorithm.
The grouping of the flow vectors into segments is shown by using various shapes of the vector tails. Vecotrs without a tail are ungrouped. In addition, the "correct" boundaries are shown.

In related work, Snyder [55,56,58] has considered the general effect of uncertainty in the position of image points on algorithms which attempt to compute environmental structure from motion. He analyzes the case of uniform translational sensor motion in a rigid environment. He finds analytical expressions for the uncertainty in depth which follows from uncertain image point positions, and for the search region for these points in subsequent frames of a multiple image sequence. The former result can be used to associate a confidence measure with the depth of each environmental point, and the latter can be used to constrain the search region for a point of interest in subsequent frames.

2.5 Recovery of Depth from Approximate Translational Motion

In this section, we describe our early attempts at recovering environmental depth from approximate translational motion. As we will note, although the first few algorithms we developed appeared at first sight to give good results, extensive experimentation on real motion sequences convinced us that the conditions necessary for these algorithms to give accurate depth values are only rarely satisfied in realistic motion scenarios, so the utility of these earlier algorithms seems to be very restricted. In Section 2.5.3, we analyze the reasons for the failure of these algorithms and present an algorithm which does not suffer from the same inadequacies. It appears very promising for the accurate determination of both environmental depth and sensor motion.

One of our earliest attempts to recover the FOE in the case of approximate translational motion was the algorithm of Pavlin [51]. In this algorithm, the global search for the FOE requires the computation of the sum of errors (e.g., via correlation) associated with the displacement of a set of feature points in two or more frames. A sparse sampling of the possible location of the FOE provides a global error function whose minimum localizes the direction of motion. The accuracy and robustness of this algorithm was found to be a function of the number of points that are tracked and contribute to the error function, which of course must be traded off against the amount of computation that can be tolerated for real-time motion analysis.

As we will note later in Section 2.5.3, the basic assumption of this algorithm, namely that the sensor motion was purely translational, is rarely satisfied in practical situations, so that this algorithm is of limited utility.

2.5.1 Refinement and Prediction of Image Dynamics and Environmental Depth Maps over Multiple Frames

The algorithm developed by Bharwani, et al. [28,29] was an attempt to iteratively refine the depth map of the environment over multiple frames so as to obtain increasingly more precise depth estimates. The algorithm assumes uniform translational sensor motion between adjacent frames of the multiple image sequence. Although our preliminary results on synthetic image sequences appeared promising, extensive experimentation with this algorithm on real image sequences showed that the assumption of uniform translational motion central to this algorithm is rarely valid. As a result, the practical utility of the Bharwani algorithm appears to be restricted to highly controlled environments where the motion of the sensor can be very precisely constrained.

2.5.2 Registration

As we have noted in the previous two sections, the assumption of uniform translational motion is typically violated to such an extent that the algorithms we developed were of little use in practical motion situations. Since the violation of the assumption of uniform translational motion implies the existence of rotational components in the sensor motion, our next attempt to find robust, accurate algorithms focused on finding and removing these rotational components, a process called *Registration*. We developed an algorithm which attempted to do this, but exhaustive experimentation showed that the removal of the rotational sensor motion components was a fragile and numerically unstable process. Indeed, we found that even very small rotational components to the motion (on the order of a few tenths of a degree) could not be effectively removed. Hence, this approach to the determination of sensor motion parameters and environmental depth was seriously flawed.

All the problems we found with these algorithms led us to develop an algorithm which could effectively deal with the existence of rotational as well as translational components to the sensor motion. That is, we sought to develop an algorithm which could deal with general sensor motion. This is described in the next section.

2.5.3 Processing Approximate Translational Motion for a Robotic Vehicle

As we have noted earlier, our previous research in motion analysis led us to attempt to deal with a real application subsystem for the Carnegie–Mellon University robotic vehicle [60]. The goal was to detect obstacles in the path of the vehicle at distances beyond the limits of the ERIM laser range sensor (i.e. at distances beyond 40 feet). Initial results from Bharwani's algorithm implied the possibility of extracting usable depth of obstacles at distances between 40 and 80 feet. By applying an FOE extraction algorithm prior to the depth extraction algorithm, there was an expectation that an effective subsystem could be developed. To accomplish this in actual imaging situations on a moving vehicle turned out to be far more difficult than anticipated.

In dynamic imaging situations where the sensor is undergoing primarily translational motion with a relatively small rotational component, it might seem likely that "approximate" translational motion algorithms can be effective in determining depth. Although translational motion was the dominant form of motion and was approximately constant over a long sequence of frames, there usually were local variations due to irregularities in the road surface (bumps, holes, and undulations), as well as minor rotation of the vehicle as it translates. This was often manifested in changes in the location of the FOE (i.e. effectively it produces a different translational motion), and in rotational motions that had to be removed if correct values of depth were to be extracted from the feature displacements. An attempt to correct for these effects via a relatively simple preprocessing algorithm (registration of the image sequence), without utilizing full analysis of the general motion problem, also led to difficulties. The issues and our experimental efforts to deal with what we initially considered to be the relatively simple problem of approximate translational motion are discussed in [34]. In this paper, we show quantitatively that even small rotations can significantly affect the computation of the FOE. This is shown both theoretically for the case of an environment which can be approximated as a frontal plane and experimentally for a real image sequence.

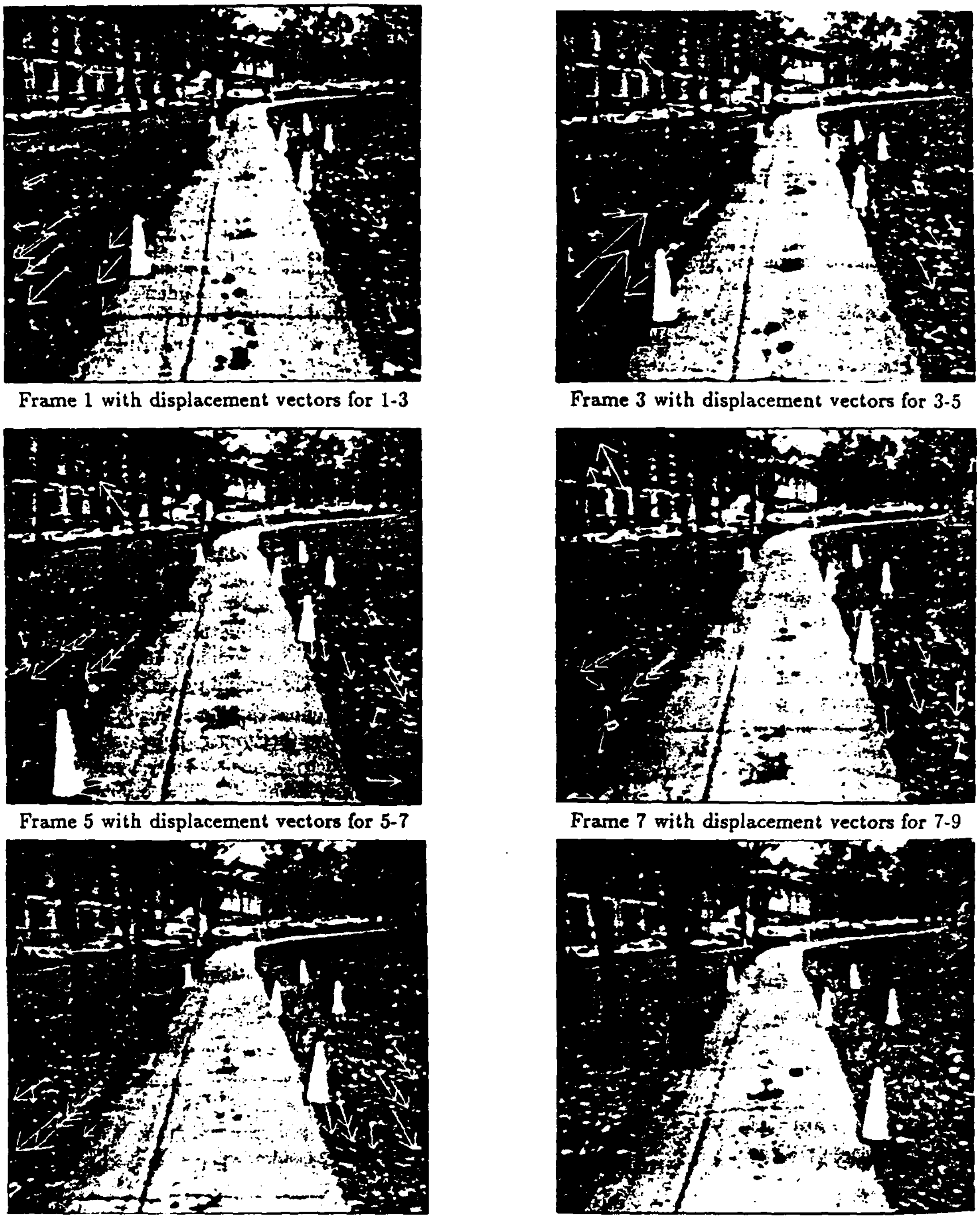
These problems led us to compare the efficacy of a general motion algorithm obtained by combining the previously described Anandan and Adiv algorithms with a new translational motion algorithm obtained by using a weighted Hough transform technique. The latter algorithm finds all the possible intersections of the displacement vectors, and corresponding to each intersection votes in a Hough array. The number of votes corresponding to each intersection is an increasing function of the length and confidences of the displacement vectors which intersect. This ensures that longer displacement vectors and more reliable displacement vectors are weighted more heavily. The smallest region in the Hough array with at least ρ (taken to be 0.1 in the experiments) fraction of the votes is then chosen as the region for the location of the FOE. The depth of points is then calculated using the *time–adjacency* relationship:

$$\frac{Z}{\Delta Z} = \frac{D}{\Delta D},$$

where Z is the depth of the 3–D point P, D is the distance from the FOE of the corresponding image point p, ΔD is the distance p moves between the initial and final frames, and ΔZ is the inter–frame sensor displacement.

We found [34] that the depths of points in a real image sequence were obtained with an error of about 9% for the general motion algorithm and of about 20% for the weighted Hough transform algorithm. In Figure 4, we show six frames from a motion sequence taken with the Carnegie–Mellon (CMU) robotic vehicle. In Table 1, we show the ground truth and experimental depth values for a number of objects in this image sequence. In Table 2, we show the results for the motion parameters obtained from the same algorithm. In Table 3, we show the average error in depth for points on the obstacles (traffic cones) in this image sequence. We conclude that while the FOE might be approximately extracted, most real situations require general motion analysis to reliably determine the depth of points, even when sensor motion is primarily translational with only small amounts of rotation. One obvious hardware solution (at significantly increased cost) is the use of a gyro-stabilized platform so that sensor motion typically will be much closer to the case of pure translational motion.

We have also developed algorithms which represent alternatives to this approach. These are described in the next sections.



Frame 1 with displacement vectors for 1-3

Frame 3 with displacement vectors for 3-5

Frame 5 with displacement vectors for 5-7

Frame 7 with displacement vectors for 7-9

Frame 9 with displacement vectors for 9-11

Frame 11

Figure 4: The Sequence of Image Frames Taken With the CMU Robotic Vehicle.

		1-3	1-3	3-5	3-5	5-7	5-7	7-9	7-9	9-11	9-11
Object	pts.	Exptl	True	Exptl	True	Exptl	True	Exptl	True	Exptl	True
cone1	1	65.7	76	58.3	72	61.7	68	50.3	64	61.2	60
	2	66.9	76	67.7	72	60.5	68	63.6	64	59.6	60
cone2	3	61.4	76	67.2	72	65.1	68	63.0	64	63.9	60
	4	60.8	76	82.3	72	56.2	68	61.7	64	61.7	60
cone3	5	50.2	56	59.2	52	46.3	48	40.8	44	38.4	40
	6	51.1	56	49.6	52	46.1	48	41.0	44	38.5	40
cone4	7	59.3	56	53.8	52	44.4	48	35.9	44	37.9	40
	8	46.3	56	53.3	52	47.6	48	41.8	44	39.8	40
can	9	44.1	46	44.4	42	47.6	38	41.8	34	39.8	30
	10	*	46	*	42	*	38	*	34	*	30
	11	*	46	*	42	*	38	*	34	*	30
cone5	12	31.0	36	32.2	32	26.0	28	22.0	24	20.0	20
	13	31.1	36	31.1	32	26.3	28	22.5	24	20.8	20
	14	31.9	36	30.9	32	28.5	28	21.9	24	20.5	20
cone6	15	18.1	21	*	17	**	**	**	**	**	**
	16	18.4	21	*	17	**	**	**	**	**	**
	17	18.9	21	-29	17	**	**	**	**	**	**
	18	18.6	21	-42	17	**	**	**	**	**	**

		1-3	1-3	3-5	3-5	5-7	5-7	7-9	7-9	9-11	9-11
Object	pts.	Exptl	True	Exptl	True	Exptl	True	Exptl	True	Exptl	True
can	9	38.1	46	43.3	42	36.5	38	32.2	34	29.6	30
	10	40.4	46	@	42	@	38	30.3	34	32.4	30
	11	44.2	46	42.5	42	@	38	32.7	34	@	30

Table 1: Depth Values of Some Points Over a Sequence of Frames Using the General Motion Algorithm.

The two tables used 100 and 200 points respectively. Depths are in feet. * and @ indicate respectively that the point was not among the top 100 or 200 Moravec points. ** indicates that the point is absent in thje image-pair.

100pts	1–3	3–5	5–7	7–9	9–11
U	-0.09	-0.09	-0.09	-0.09	-0.09
V	-0.25	-0.25	-0.25	-0.25	-0.25
W	-0.96	-0.96	-0.96	-0.96	-0.96
A	-0.19	0.17	-0.10	-0.04	-0.03
B	0.39	0.56	0.53	0.49	0.43
C	-0.30	0.01	0.07	0.06	0.28

200pts	1–3	3–5	5–7	7–9	9–11
U	-0.09	-0.16	-0.09	-0.09	-0.09
V	-0.25	-0.21	-0.25	-0.25	-0.25
W	-0.96	-0.96	-0.96	-0.96	-0.96
A	-.19	0.11	-0.10	-0.03	0.03
B	0.41	0.17	0.53	0.49	0.43
C	-0.22	-0.52	0.10	0.07	0.31

Table 2: Motion Parameters Obtained Using the General Motion Algorithm. The frame pairs are at 4 ft. intervals. The results have been tabulated for 100 and for 200 Moravec points. (U,V,W) is the unit translation vector, and (A,B,C) is the rotational vector in degrees.

Frame-Pair	Average Error
1-3	12.4 %
3-5	6.9 %
5-7	8.2 %
7-9	9.2 %
9-11	5.4 %

Total Average Error = 8.9 % .

Table 3: Average Errors in Depth For Points on the Obstacles.
The obstacles are the traffic cones in Figure 4. The results are for the general motion algorithm of Adiv.

2.6 Stereoscopic Motion Analysis and the Detection of Discontinuities

By carrying out motion analysis with imagery from a pair of sensors—stereoscopic motion—the additional constraints can significantly reduce the complexity of the analysis on a theoretical level. Balasubramanyan and Snyder [23,24,25] have developed an algorithm to extract the parameters of motion in depth: the single component T_Z of translation in depth (i.e. parallel to the line of sight) and the two components Ω_X and Ω_Y of rotation in depth (i.e. rotations that are not around the line of sight). This is achieved by building upon the work of Adiv for segmenting the flow field into rigid independently moving objects [1], and the formulation of Waxman and Duncan [62]. The latter authors show that the ratio of the relative optical flow between a stereo pair of images to the disparity between them is a linear function of the image coordinates:

$$\begin{aligned} \frac{\Delta\alpha}{\delta} &= \Omega_Y x_\ell - \Omega_Y y_\ell - \frac{T_Z}{Z} \\ \frac{\Delta\beta}{\delta} &= 0, \end{aligned}$$

where $\Delta\alpha$ and $\Delta\beta$ are the components of the relative optical flow between the two images, δ is the disparity between the two images, (x_ℓ, y_ℓ) is the coordinate of a point p in the left frame, Ω_X, Ω_Y, and T_Z are the three motion-in-depth parameters, and Z is the spatial depth of the corresponding environmental point P.

The algorithm proceeds in four steps:

1. Extract the relative optical flow field between the left and right images using the difference between the two optical flow fields, along with the disparity field.

2. Use Adiv's algorithm to segment the monocular optic flow corresponding to the left sensor. This segmentation is therefore performed using only motion information in the 2-D image plane in order to obtain a grouping of the flow vectors, where each segment corresponds to the motion of a roughly planar surface.

3. Merge the segments on the 2-D image plane (obtained from the segmentation step) based on a least-square minimization to compute the motion-in-depth parameters for each of the merged regions. The output at this stage is a grouping of the image into regions that correspond to the same set (within some normalized value of the deviation) of motion-in-depth parameters.

4. Minimize the following error functional over each set of relative flow vectors corresponding to a single segment or possibly a merged set of them:

$$E(\Omega_X, \Omega_Y, T_Z) = \sum_{i=1}^{n} W_i \left[\left(\frac{\Delta\alpha}{\delta} \right)_i - \Omega_Y x_i + \Omega_X y_i + \frac{T_Z}{Z_i} \right]^2 ,$$

where (x_i, y_i) denotes an image point in the set in question, and n is the number of elements in the set.

The algorithm was run on synthetic data with general motion of both the sensor and independently moving objects. It shows good performance with ideal images (i.e., no noise), but shows some degradation of performance with increasing noise. A representative example of the results obtained are given in Figure 5 and Table 4. Work is currently underway to test the effectiveness of this algorithm on real scenes.

One of the most important problems in stereo and motion processing is the recovery of depth and motion boundaries. A number of algorithms for computing optic flow make a global smoothness assumption that tends to unnaturally smooth across depth and motion discontinuities. This makes later detection of these boundaries very difficult. On the other hand, knowledge of these discontinuities is very important for the flow and disparity computations to be correct, especially at occlusion boundaries.

One approach to this problem is to integrate motion and stereo data. Balasubramanyam and Weiss [26] use information in both the stereo and motion sequences at two time instances to define a confidence measure in the presence of motion and depth discontinuities. This measure can be applied early, prior to the full computation of flow and disparity fields. The general idea is to use coarse disparity and flow estimates from hierarchical correlation processes [10] to locate and label depth and motion discontinuities; smoothing is then inhibited across these boundaries. Discontinuities that are continuous (i.e. unbroken) in the other dimension are favored. The results of running this algorithm on both synthetic and real stereo-motion imagery are presented in [26]. We give an example in Figure 6.

2.7 Smoothness Constraints for Optical Flow and Surface Reconstruction

The computation of optical flow normally requires a constraint on the variation of the flow fields from constancy. Snyder [57] has given an axiomatic derivation of the possible smoothness constraints under a small number of physically reasonable assumptions. He shows that there are only four possible smoothness constraints which are quadratic in first derivatives of the optical flow, and either first or second derivatives of the image intensity function that satisfy these assumptions. He also gives a novel geometric interpretation of these smoothness constraints, and shows that only two of the four are physically sensible.

2.8 Analysis of Constant General Motion

Another way to introduce additional constraints to the problem of general motion analysis in an effort to achieve practical, robust algorithms is via Shariat's formulation: constant but arbitrary general motion of a rigid object [54]. This leads to a set of difference equations across a sequence of images, relating the positions of a feature in the image plane to the motion parameters of the projected point. The solution obtained is a set of 5th

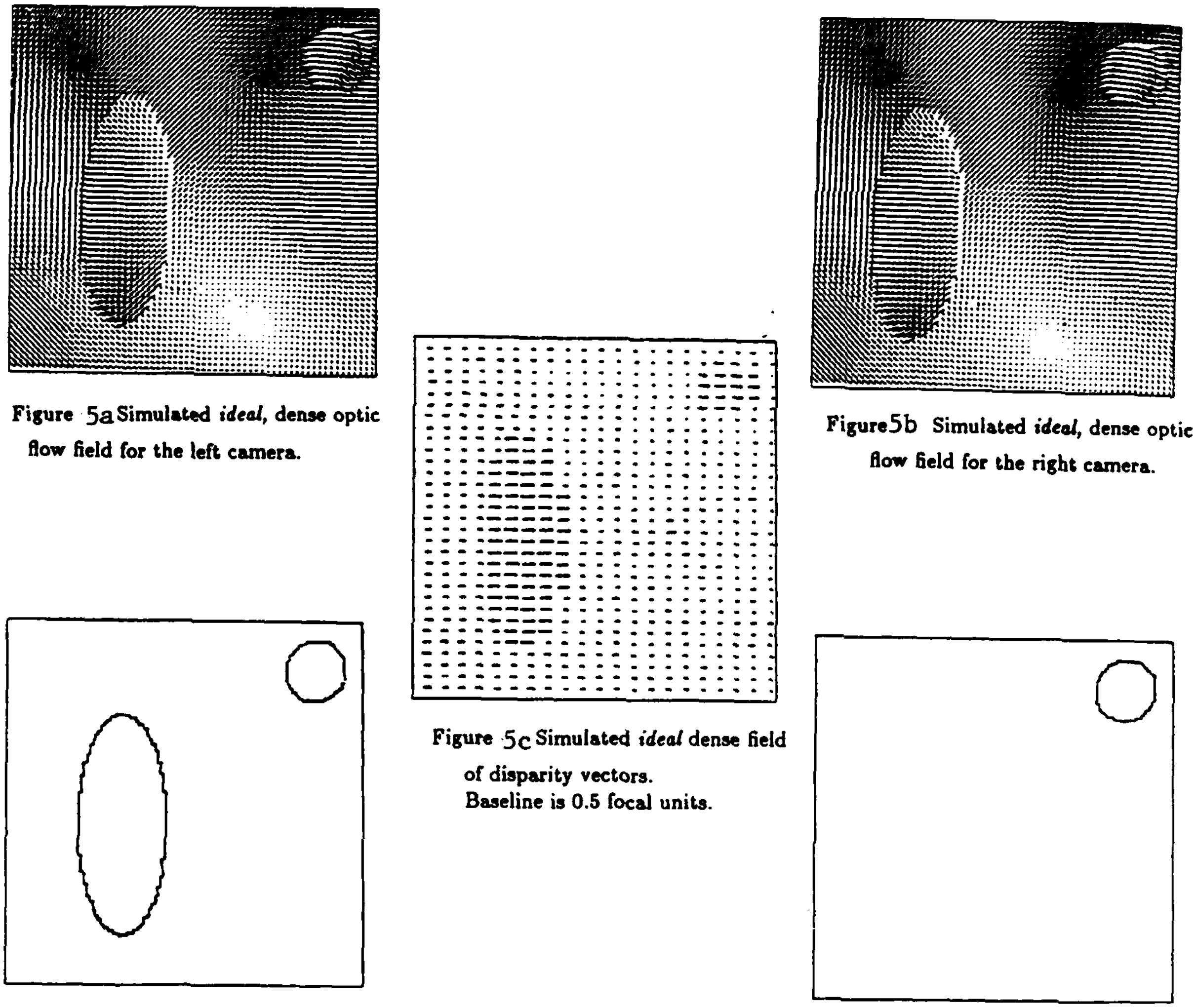


Figure 5a Simulated *ideal*, dense optic flow field for the left camera.

Figure5b Simulated *ideal*, dense optic flow field for the right camera.

Figure 5c Simulated *ideal* dense field of disparity vectors.
Baseline is 0.5 focal units.

Figure 5d Result of segmentation performed using Adiv's algorithm [1].

Figure 5e Result of merger in the optimization step of the algorithm. Note that the independently moving sphere is picked out.

Figure 5: Stereoscopic Motion.
The algorithm of Balasubramanyam and Snyder applied to a noisy optical flow field. The camera motion is completely general; the sphere is moving independently with no motion-in-depth components, while the ellipsoid is stationary.

<table>
<tr><td rowspan="3"><em>sphere</em></td><td rowspan="3"><em>size</em>
2,2,2</td><td rowspan="3"><em>position</em>
9,9,30</td><td colspan="2"><em>Object Translation</em>
<em>(focal units)</em></td><td colspan="2"><em>Object Rotation</em>
<em>(radians)</em></td></tr>
<tr><td><em>Input</em></td><td><em>Computed</em></td><td><em>Input</em></td><td><em>Computed</em></td></tr>
<tr><td>$T_X = 0.50$
$T_Y = -0.5$
$T_Z = 0.00$</td><td>$T_Z^{comp} = 0.11$</td><td>$\Omega_X = 0.00$
$\Omega_Y = 0.00$
$\Omega_Z = -0.19$</td><td>$\Omega_X^{comp} = 0.04$
$\Omega_Y^{comp} = 0.02$</td></tr>
<tr><td></td><td><em>size</em></td><td><em>position</em></td><td colspan="2"><em>Object Translation</em>
<em>(focal units)</em></td><td colspan="2"><em>Object Rotation</em>
<em>(radians)</em></td></tr>
<tr><td><em>ellipsoid</em></td><td>2,5,2</td><td>-3,-1,20</td><td colspan="4">stationary</td></tr>
<tr><td><em>plane</em></td><td colspan="2">Z = X + 0.5Y + 50</td><td colspan="4">stationary</td></tr>
<tr><td rowspan="3"><em>camera</em></td><td colspan="3"><em>Camera Translation</em>
<em>(focal units)</em></td><td colspan="3"><em>Camera Rotation</em>
<em>(radians)</em></td></tr>
<tr><td><em>Input</em></td><td colspan="2"><em>Computed</em></td><td><em>Input</em></td><td colspan="2"><em>Computed</em></td></tr>
<tr><td>$T_X = 0.50$
$T_Y = 0.05$
$T_Z = 1.0$</td><td colspan="2">$T_Z^{comp} = 1.2$</td><td>$\Omega_X = 0.02$
$\Omega_Y = -0.02$
$\Omega_Z = 0.04$</td><td colspan="2">$\Omega_X^{comp} = 0.03$
$\Omega_Y^{comp} = -0.01$</td></tr>
</table>

Table 4: General Camera Motion with Independent Object Motion.

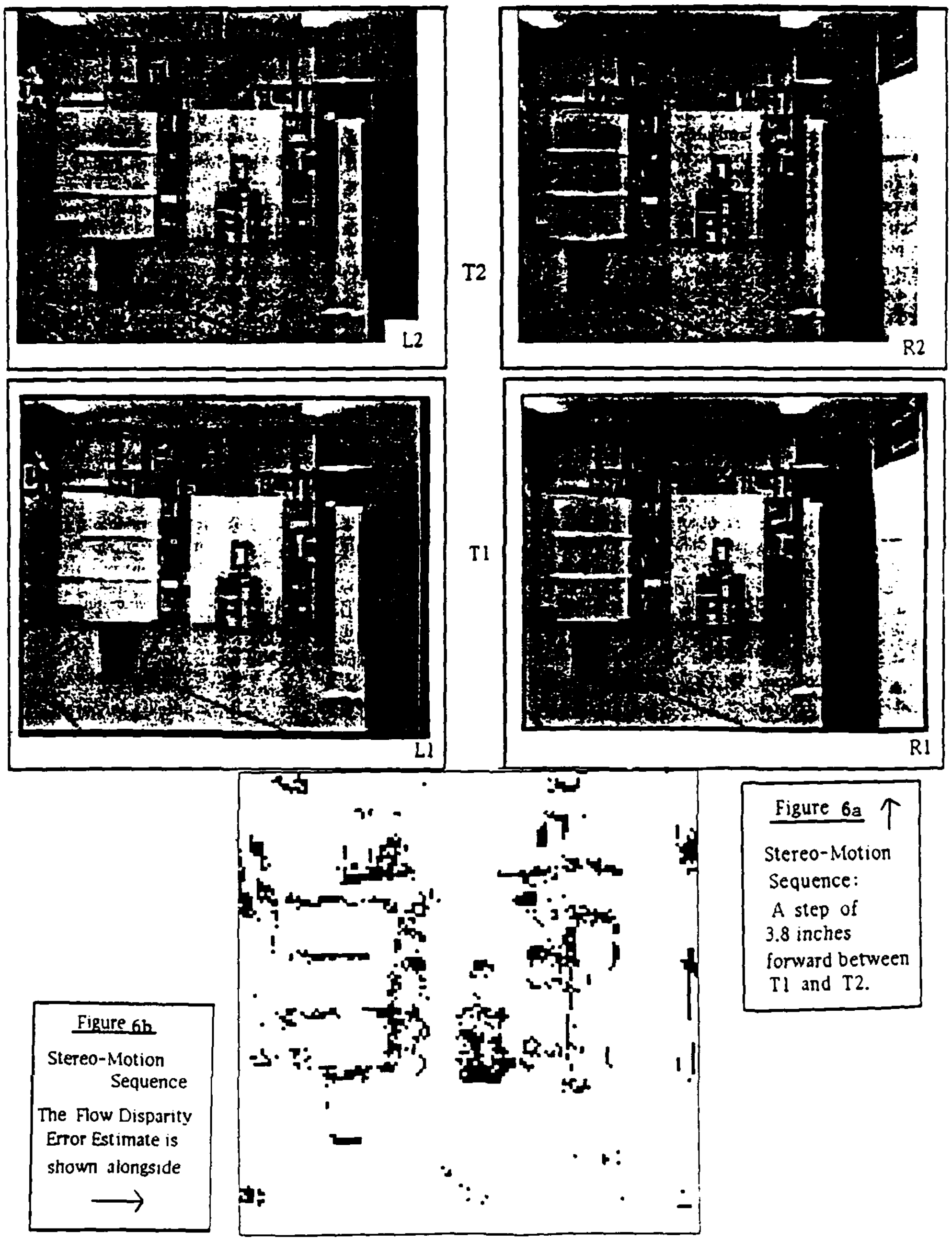


Figure 6a ↑

Stereo-Motion Sequence: A step of 3.8 inches forward between T1 and T2.

Figure 6b

Stereo-Motion Sequence

The Flow Disparity Error Estimate is shown alongside →

Figure 6: The Balasubramanyam and Weiss Algorithm.
The four video images at the top of the page are the stereo motion sequence. The images from the left (right) camera are on the left (right). The earlier stereo pair is at the bottom, while the later stereo pair is at the top. The binary image at the bottom of the page shows the flow disparity estimate.

order non-linear polynomial equations in the unknown motion parameters. The solution requires a Gauss-Newton non-linear least-squares method with carefully designed initial guess schemes. Pavlin [52] has derived a closed-form solution for the rigid object trajectory by integrating the differential equations describing the motion of a point on the tracked object. The integrated equations are non-linear only in angular velocity, and are linear in all other motion parameters. These equations allow the use of a simple least-square error minimization criterion in an iterative search for the motion parameters.

2.9 Token-Based Approaches to Motion and Perceptual Organization

The problems cited previously with respect to the extraction of motion and depth information using traditional optical flow techniques have led us toward the exploration of methods for combining the local flow/displacement fields with larger token-like structures. It is our position that the inherently local measurement of visual motion provided by optical flow is insufficient to meet the varied requirements of dynamic image understanding. The approach we developed involves computing the correspondence between tokens of arbitrary spatial scale produced by perceptual organization processes. Such tokens often map directly to environmental structure, and descriptions of their movement often correlate more closely with the motion of physical objects than does the local motion information contained in the displacement field. A token match represents more than just a spatial displacement; also explicit in this representation are the time-varying values of those parameters which define the token, or which can be extracted from the structure of the token.

The work of Williams and Hanson [65,66] describes work in progress toward this goal. The premise of this work is that the structure obtained from perceptual organization processes can be combined with the local motion information contained in the flow field to provide a more robust estimate of motion and depth parameters. The approach can be viewed as augmenting the rather limited use of spatial structure in traditional approaches with the richer descriptive vocabulary of spatial structure provided by the perceptual organizational processes over both space and time. In this sense, the spatially organized structures (such as lines, regions, curves, vertices, intersections, rectangular groups, etc.), which are actively constructed from the image can be considered to be interest operators of large spatial extent.

In their first paper [65], a method for computing the temporal correspondence between straight line segments is presented. We consider the two frame case here, but the method is extensible, and has been extended, to multiple frames. A straight line perceptual organization process developed by Boldt and Weiss [27,64] is applied to both frames independently to provide straight lines in each frame. A displacement field is also computed from the two frames using the algorithm developed by Anandan [9,10]. After filtering the straight lines on length and constrast to reduce the line set in both images, the displacement field is used to construct a search area in Frame 2 for each line in Frame 1. Since a one-to-one correspondence between lines is unlikely, a minimal mapping approach [61] is used to compute the correspondence between the Frame 1 and Frame 2 line sets; such a mapping is called a minimal bipartite cover. The similarity measure used to compute the cover involves the similarity and spatial separation of the candidate token matches. By computing the connected components of the bipartite graph, the global matching problem is conveniently divided into smaller, individually tractable pieces which reflect the scope of potential interactions. A simple blind search of the subgraphs is used to extract the bipartite cover minimizing the positional and similarity discrepancy metric.

The matching results obtained are quite good. The system has been run repeatedly on successive frames of several multi-frame sequences. In the multi-frame case, a directed acyclic graph is constructed which represents the splitting and merging patterns of line segments over time. Work is in progress to analyze the trajectories of the tokens over time. In Figure 7, we show the first frame of an image sequence of a soccer ball, the computed displacement field, the line tokens, and the output of the matching process for selected lines.

In their second paper [66], a method for computing depth from the line correspondences is described using the temporal change in the length of virtual lines constructed from the intersections of the Boldt lines [27]. They use virtual lines because the length of the original lines is not reliable, although their orientation and lateral displacement are quite precise. This "looming" method is also generalized to areas. The method is generally applicable to any structure whose total extent in depth is small compared to the depth of its centroid (that is, for those cases in which perspective projection can be approximated by scaled orthographic projection [59]) and which does not exhibit any independent motion. The technique does not depend on the complete determination of the egomotion parameters of the sensor, but it does require the computation of the component of the sensor's translation in the direction of motion. An analysis of the sensitivity of the algorithm to errors in the measured variables is planned for the near future; experimental results on real image sequences suggest that the algorithm may be quite robust. In Figure 8, we show the first frame of an indoor motion sequence taken by our mobile robot. In Figure 9, we show the line segments used to define virtual lines and virtual regions. In Tables 5 and 6, we show the experimental results for depth using the virtual lines and virtual regions, respectively. The error in depth seems to be around 5%; this is a promising result, but further experimentation is necessary.

2.10 3–D Interpretation of Rotational Motion from Image Trajectories

The research of Sawhney and Oliensis [53] addresses the problem of finding the motion parameters of independently moving objects in their natural coordinate system. They analyze an extended time sequence of images of an object rotating uniformly around an axis of arbitrary location and orientation, and demonstrate how the abstraction of continuous descriptions of multi-frame data can lead to the recovery of scene motion and structure.

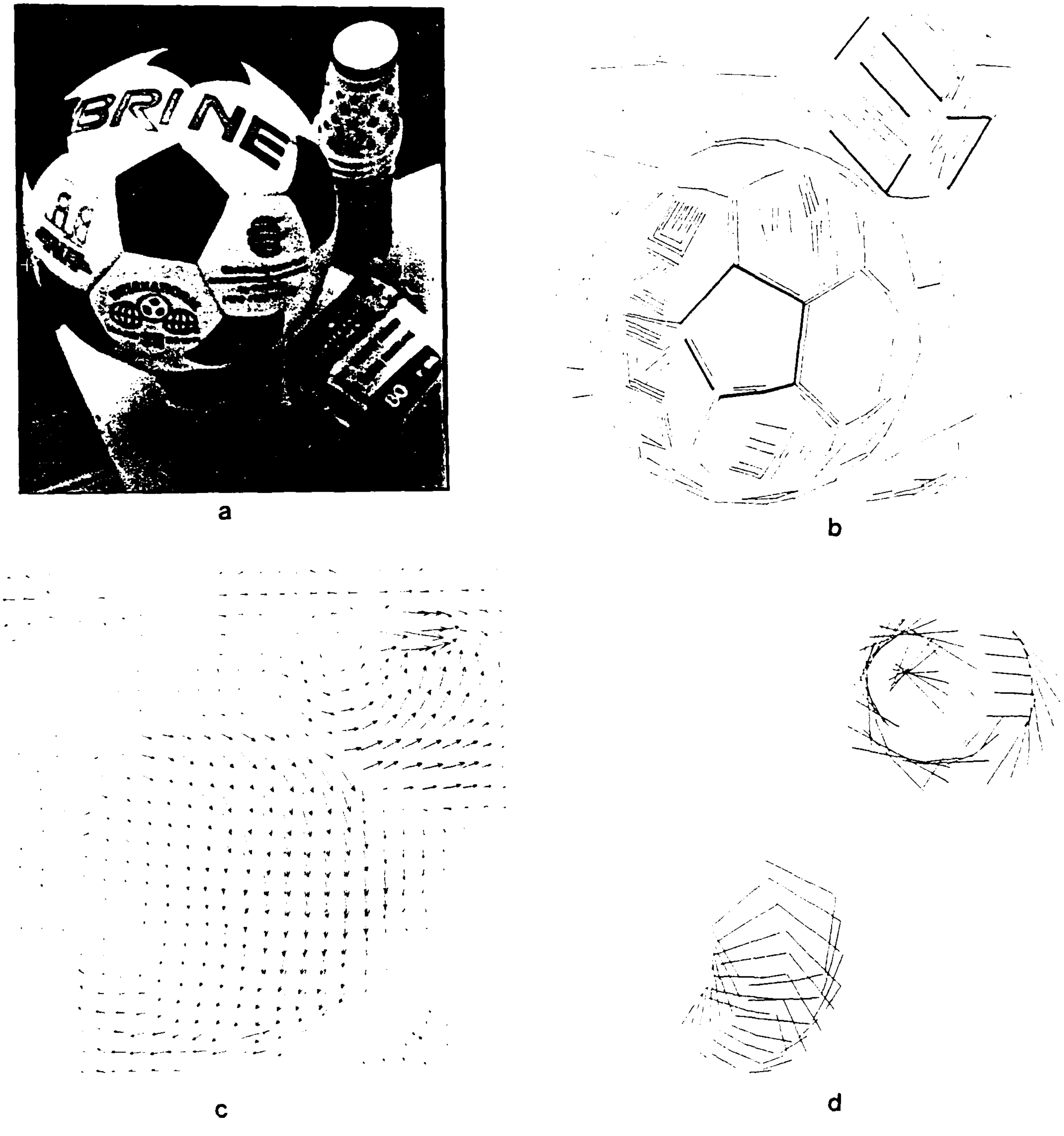


Figure 7: Application of the Algorithm of Williams and Hanson.
7a. The first frame of a motion sequence containing multiple independently moving objects. 7b. The Line Tokens Computed For the First Frame. Line tokens which will be used to illustrate the output of the matching process are displayed thick. 7c. The displacment field computed for the first and second frame of the sequence. Note the rotation of the box and the soccer ball. 7d. The output of the patching process for selected lines.

Figure 8: Mobile Robot Image Sequence.
The first frame of a motion sequence taken by a mobile robot moving down the hallway.

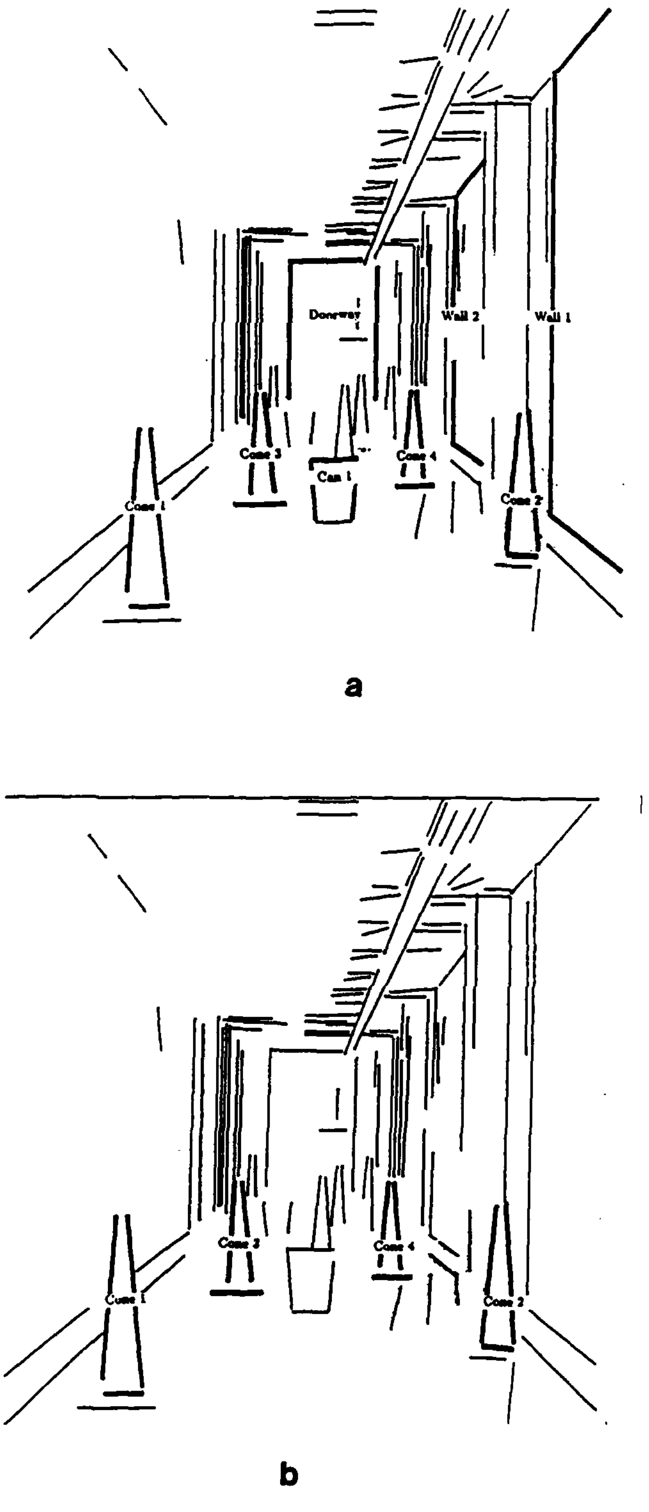


Figure 9: Line Segments Found by the Williams and Hanson Algorithm. 9a. The line segments used to define *virtual* lines. 9b. The line segments used to define *virtual* regions.

Virtual Line	Depth (ft.)	Ground Truth (ft.)	% Error	t
Cone 1	19.1	20.0	4.5	1
Cone 2	23.6	25.0	5.6	3
Cone 3	28.3	35.0	19.1	1
Cone 4	42.1	40.0	5.3	7
Can 1	29.0	30.0	3.3	7
Wall 1	27.7	27.1	2.2	2
Wall 2	48.8	48.7	0.2	7
Doorway	88.8	87.1	2.0	7

Table 5: Comparison of the Computed and the Ground Truth Depth for the Virtual Lines.

Virtual Region	Depth (ft.)	Ground Truth (ft.)	% Error	t
Cone 1	20.1	20.0	0.5	1
Cone 2	25.8	25.0	3.2	3
Cone 3	35.5	35.0	1.4	1
Cone 4	40.0	40.0	0.0	7

Table 6: Comparison of the Computed and the Ground Truth Depth for the Virtual Regions.

Image traces of 3–D feature points are generated from image point correspondences over a sequence of frames. These traces are described by continuous curves that are obtained by fitting conic arcs to the set of points. The goal is motion-based grouping of image traces to provide constraints (unavailable in only a few frames) sufficient to extract the motion parameters of independently moving objects in their natural coordinate system.

2.11 A Motion Data Set from the Autonomous Land Vehicle (ALV)

A major difficulty with the analysis of motion algorithms has been the lack of motion data with ground truth of known precision. In particular, these data have not been collected for robot vehicles operating under realistic conditions in outdoor environments. Thus, the proper scientific evaluation of motion algorithms intended for practical application has been impossible.

In response to this general problem, our group decided to collect a reasonably large data set from the ALV [35,36]. Motion sequences of about 30 frames each were collected at five different outdoor sites with different road surfaces, including on-road, dirt-road, and off-road scenarios. Data from the video camera, laser range finder, and land navigation system (LNS) were recorded simultaneously under stop-and-shoot and move-and-shoot scenarios. *Ground truth data for the 3–D environment* were obtained using traditional surveying methods, while the LNS provided ground truth data for the motion parameters. This motion data set is available to the general community and can be obtained by contacting Ms. Valerie Cohen at the University of Massachusetts (UMass) at Amherst (E-mail address is VCohen@CS.UMass.EDU).

3 Mobile Robot Navigation

Vision-based mobile robot navigation is a relatively recent addition to the VISIONS research group at UMass. We have acquired a mobile robot (called HARV) that will enable us to develop a testbed for many of the vision algorithms that we have developed and continue to develop. The robot is to be operated both indoors and out, providing a wide variety of scenes for analysis. The integration of robot planning, perception, and motor control systems for effective navigation is the focus of continuing work, beginning with the work of Arkin [13] and continuing with the more recent work of Fennema [37,38,39].

3.1 AuRA—the Autonomous Robot Architecture

Arkin developed an integrated system, the UMass Autonomous Robot Architecture (AuRA) [11,12,13,14,15,16,17,18,19,20,21,22], to support this research effort. It incorporated both global and reflexive schema-based path planning strategies and utilized *a priori* knowledge stored in long-term memory, when available, to assist the vehicle's attainment of its navigational goals.

AuRA has five major components: the planning, cartographic, perception, motor, and homeostatic subsystems. A block diagram of AuRA is presented in Figure 10.

The purpose of the hierarchical planning subsystem is to handle the task of path planning in both indoor and outdoor environments. The cartographic subsystem maintains the information in long- and short-term memory (which store *a priori* and acquired world knowledge, respectively), and supplies it on demand to the planning and perception modules. The perception subsystem processes all the sensory information from the environment, interprets it, and delivers it to the cartographic subsystem. The motor subsystem controls the motion of the vehicle. Finally, the homeostatic subsystem is concerned with maintaining a safe internal environment for the robot.

The chief navigational issues addressed in the work of Arkin, and also that of Fennema, include path following, landmark recognition for vehicle localization, and obstacle avoidance. A new fast line finding algorithm [46] was used for hall and sidewalk navigation and for localization purposes. Our depth-from-motion algorithms are used for obstacle avoidance, and can also provide information for landmark identification when coupled with top-down knowledge of expected landmark locations. A new fast region segmentation algorithm [32] has found potential application in both path following and vehicle localization. A description of all these algorithms and their use within AuRA can be found in [13].

Arkin is now at the Georgia Institute of Technology, continuing the development of AuRA. Fennema has built on our experience with Arkin's systems to develop new systems for model-directed navigation, described in the next section.

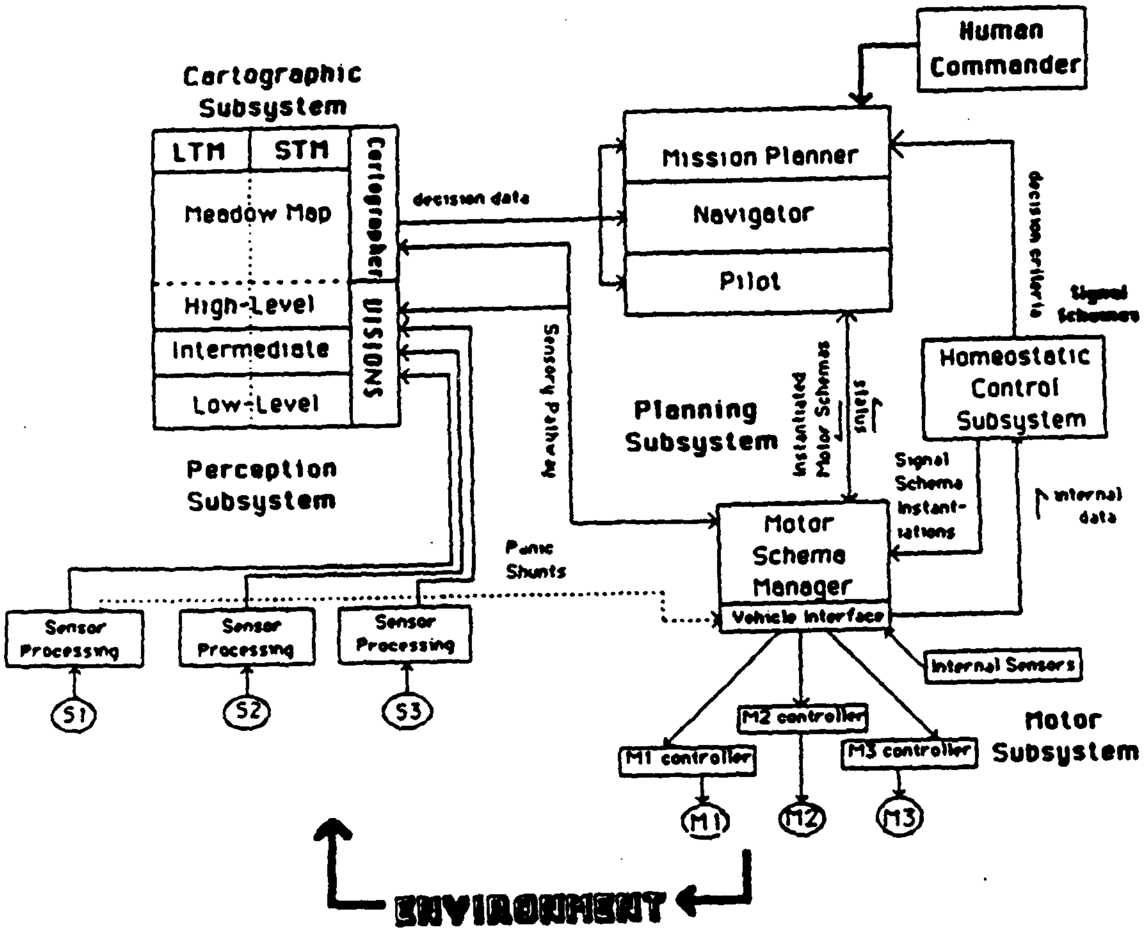


Figure 10: System Architecture of AuRA.
A block diagram of the UMass Autonomous Robot Architecture (AuRA).

3.2 Planning and Control via Milestones for Model—Directed Navigation

Our mobile robot, HARV, begins with an accurate, but incomplete, model of the world implemented in GeoMeter (Section 3.3). Each task given to HARV is translated by a command interpreter and problem solver which ultimately produces a set of navigational goals. The execution of these goals is accomplished by a tight interweaving of planning, perception, and action, orchestrated by a dynamic planning and execution scheme [37,38,39]. This subsystem works with plans, each represented as a sequence ($M0\ A1\ M1\ \ldots An\ Mn$) of milestones Mk and proposed actions Ak. Milestones are constructed from perceivable events, and are used to verify the successful completion of a particular phase of the plan. As used here, milestones are composed of 3–D landmarks (perceivable physical events) and their expected location with respect to the robot at the completion of the appropriate phase of the plan. They allow the progress of the plan to be monitored and to trigger replanning before the next action is taken when perception and the milestone do not agree.

Planning, perception, and execution are directed by the plan–and–monitor executive in such a way as to dynamically modify and refine the plan to fit the actual results of each action and the details of the perceived environment. The principal activities involved in this process are planning, milestone recognition, determination of location, and execution of primitive actions. Interweaving perception, planning, and action in this way makes specific what task is expected of perception, and provides a way of focusing the available knowledge to that end. The result is a distribution of perception and perceptual reasoning into all aspects of navigation.

The actual motion in response to the plan is produced by the plan–and–execute module. This motion is controlled using perceptual servoing. Perceptual servoing determines the robot's motion by enforcing control at several levels: action–level servoing ensures accurate execution of each primitive action; plan–level servoing uses vision to ensure that the accumulation of primitive actions conforms to a plan; and goal–level servoing ensures that overall action is directed to the goal. Each level uses model–directed vision and compares what is sensed to what is expected, and issues corrective actions to minimize any difference. The detailed explanation of each of these can be found in the work of Fennema, et al. [37,38,39].

3.3 GeoMeter

Models of the vehicle's environment are built using GeoMeter, a three–dimensional solid modelling package developed jointly by UMass and the General Electric Research and Development Center [33]. GeoMeter is implemented in CommonLisp and is oriented towards image understanding research (although it has many other potential applications). It currently runs on several types of workstations, including Symbolics LISP machines, TI Explorers, VAX workstations, and SUN workstations. Work is under way to allow it to run on the Sequent Balance 2000.

GeoMeter adopts the language of simplicial complexes in algebraic topology for describing surfaces. It provides generality and an explicit representation of edges, vertices, and faces. Each of these serve as a type of geometric primitive, and can be parametrized as a smooth function from a point, unit interval, and triangle to $\mathbf{R}^3$, respectively. Surfaces are constructed as the union of these primitives, and are denoted by a sum of simplices. This representation produces a triangulation of the surface, where the triangles are not necessarily planar.

GeoMeter has two basic parts: a geometric section, and an analytic section. The three basic entities which the geometric section uses to represent sets of points are the *vertex*, the *edge*, and the *face*. These are then composed to represent solid objects. Topological structures are then used to define the connectivity between the sets of the model, and solid objects are built hierarchically starting with vertices, then edges, then faces.

The analytic section of GeoMeter is devoted to the manipulation of polynomials and transcendental functions. This is of interest because these functions permit the exact description of curved surfaces, and also because such manipulations provide a mechanism for performing algebraic deduction, which is useful in reasoning about geometric relations.

We have surveyed a portion of the UMass campus and have used GeoMeter to construct a 3–D model, including buildings, sidewalks, lampposts, telephone poles, etc. This model has been annotated with properties of objects and surfaces which are useful to the planning and vision routines used by our mobile robot HARV. Although this cannot include every visible entity (e.g., dirt patches within grassy areas), most of the significant stationary objects in the environment have been represented in the model. Finally, the entire model has been placed in a space–organizing data structure, which divides 3–D space into "locales," or space packets, that are used for planning and for locating the robot. In Figure 11, we show how GeoMeter models the area around our building.

3.4 2–D Model Matching

An important problem in model–driven 3–D interpretation is how to use approximate knowledge of the location and orientation of the sensor, models of objects in the environment, and the results of low-level vision to determine the image–to–model correspondence. The approach we have taken is to separate 2–D model–to–image matching from the determination of the 3–D pose parameters (see section 3.5). We believe this approach will be more robust.

Beveridge, et al. [30,31] assume that a 2–D model has been supplied with rough constraints on its image position (e.g., via an approximate 3–D location in a modelled environment). This substantially reduces the search space of possible model–image line correspondences. The goal here is to determine correspondences between model and data lines such that an optimized spatial fit will produce the lowest match error. The search must be carried out across the space of possible line correspondences. This involves dealing with the complexities of grouping fragmented data and missing or erroneous lines. The rotation and translation of the model that minimizes the error in spatial fit for a given set

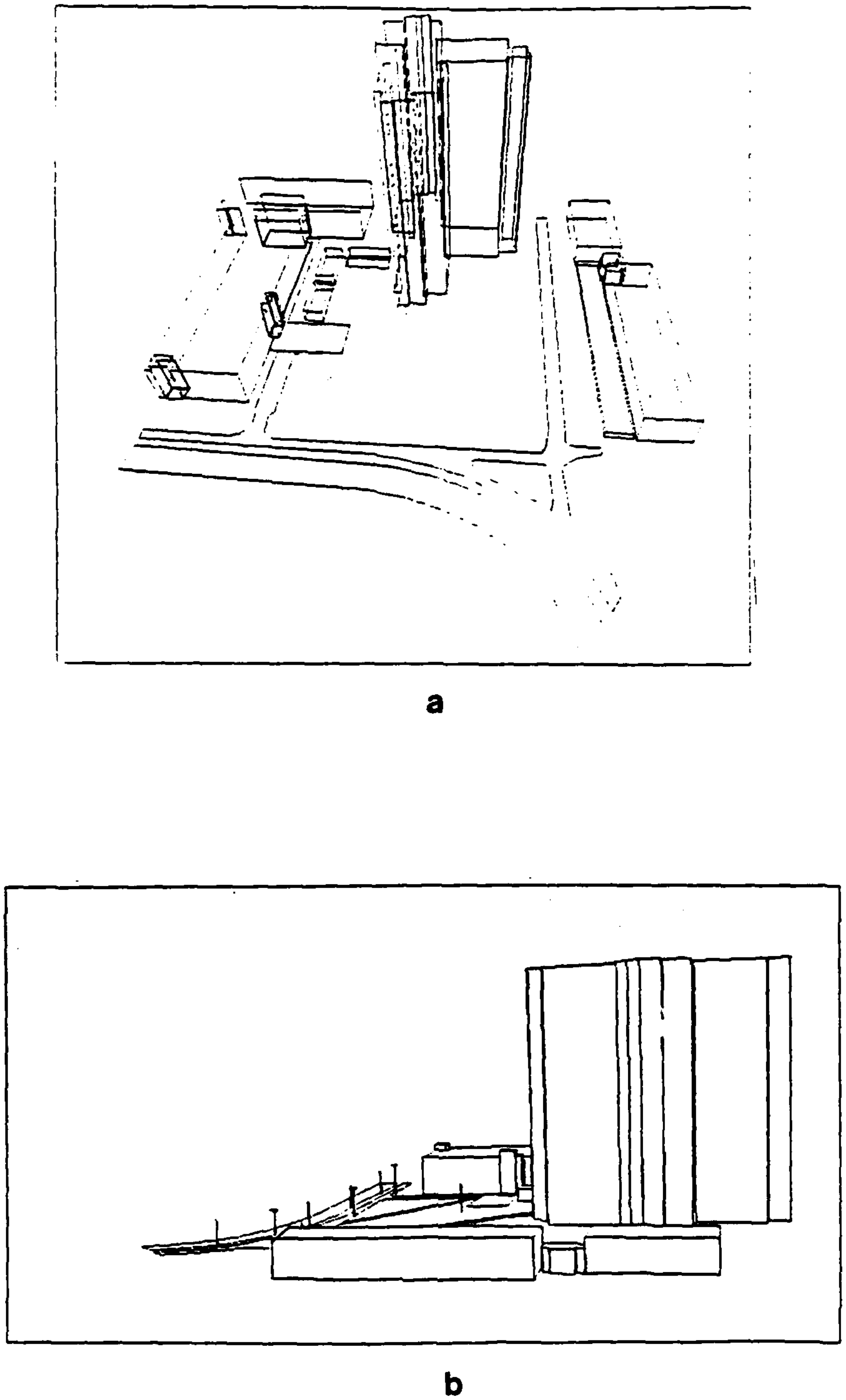


Figure 11: The GeoMeter Model of the Area Around the Graduate Research Center at UMass.

11a. Geometer model of the area around the Graduate Research Center. 11b. A more detailed Geometer model (with hidden lines removed) of the same area shown in 11a. Note that additional landmarks, such as telephone poles, have been added.

of line correspondences is computed via a closed–form solution.

In more detail, the basic steps of the model–matching algorithm are:

1. Determine the search space of correspondences. Lacking constraints on model position, all data line segments possibly correspond to every model line segment. If constraints are available, only associations of model and data lines satisfying these constraints need be considered.

2. Determine promising model positions if the search space is large. Use these positions to determine constrained search subspaces made up only of correspondences consistent with the estimated position. A promising model position may be found either through a generalized Hough transform or by identifying prominent features. The generalized Hough technique involves an analysis of the space of possible two–dimensional spatial transforms necessary to bring the model and the data into alignment. The identification of a prominent feature may involve finding a distinctive part of a model such as a corner, then using that to position the model as a whole.

3. For each of the constrained search spaces (sets of possible model–data correspondences) obtained above, use iterative refinement to determine a best match. After each iteration, perturb the correspondence, adding or deleting one or several data lines, and then determine the new best–fit model position and related match error. If the match error is thereby reduced, adopt the improved match; stop when the match can no longer be improved. The best of the resulting matches is taken as the final match.

This algorithm has achieved interesting results when used on images from our mobile robot domain. In Figure 12, we show a 512 × 512 image of the area around our building, taken from the mobile robot HARV. In Figure 13a, we show six navigational landmarks obtained using GeoMeter. In Figure 13b, we show the result of applying the 2–D model matcher to the image. We see that the matcher has correctly found the data segments which match the landmark lines.

3.5 3–D Pose Refinement

Kumar [47] has developed an optimization technique for finding the 3–D sensor pose given a set of correspondences between 3–D model lines and 2–D image lines. The 3–D pose is given by the rotation and translation matrices which map the world coordinate system to the sensor coordinate system. Using the output of the system described in the previous section, these algorithms allow updating of the mobile robot position via landmark recognition.

Previous researchers, e.g., Liu, et al. [50], have decomposed this problem into two stages: first solve for the rotation, and then solve for the translation. The problem with this approach is that the rotation and translation constraints, when used separately, are very weak constraints, such that even small errors in the rotation stage can be amplified

Figure 12: A 512 × 512 Image Taken With Our Mobile Robot HARV.

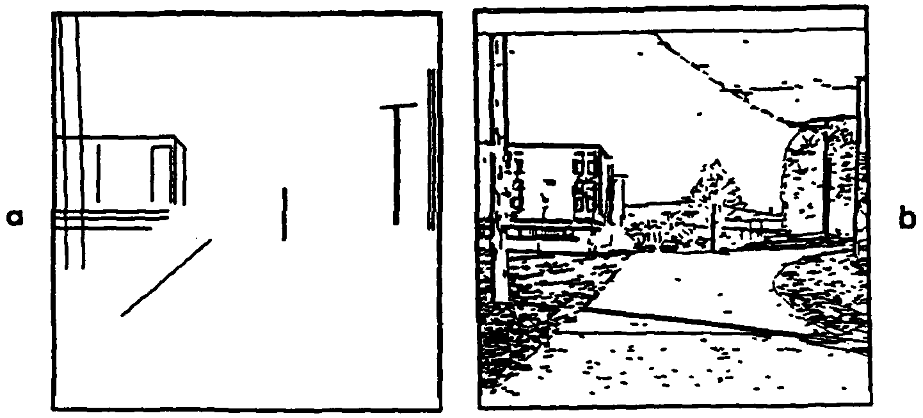


Figure 13: The 2-D Model Matcher.
13a. The six navigational landmarks projected onto the image plane. 13b. The data line segments matching the landmark lines.

into large errors in the rotation stage. In Kumar's work, rotation and translation are solved for simultaneously using an algorithm called "R–and–T." The regnant constraints for this approach are much tighter, and hence are much more immune to noise than previous approaches.

The technique used to solve for the optimal rotation and translation is adapted from the work of Horn [45] on the problem of relative orientation. Kumar minimizes the objective function, which measures the error between the data and a presumed rotation and translation, by first estimating the rotation and translation. He then linearizes the error term about this estimate and makes iterative adjustments to the rotation and translation that reduce this error. The iterations are continued until the algorithm converges to a minimum. This nonlinear least–squares optimization technique has much better convergence properties than does Liu, et al.'s solution method based on Euler angles. The algorithm has been tested on both synthetic and real images, with good results (see Table 7).

For practical applications, the issue of computational speed is critical. The acquisition of parallel hardware, a Sequent multiprocessor, will decrease the processing time required for both vision and motor tasks and is expected to enhance the real-time capabilities of the mobile robot project. We are in the process of porting our algorithms for robot navigation onto the Sequent, and will be doing timing experiments. An additional piece of hardware is the UMass Image Understanding Architecture (IUA) currently being developed under another DARPA–sponsored contract [63]. The IUA is a three level board (64 × 64) which has been designed to deal with with the different levels of computation that one typically finds in vision tasks, and should be able to operate at speeds that allow real–time vehicle control. When it is complete, we believe that real–time processing for most of the vision and robotics navigation algorithms will be feasible.

NOISE			ROTATION ERROR			TRANSLATION ERROR		
No. Lines	θ deg.	ρ pixels	$\delta\omega_x$ deg.	$\delta\omega_y$ deg.	$\delta\omega_z$ deg.	ΔT_x feet	ΔT_y feet	ΔT_z feet
Correct			0.00	0.00	0.00	0.00	0.00	0.00
5	1.0	1.0	0.24	0.15	0.04	0.21	2.03	1.16
5	5.0	5.0	1.20	0.79	0.19	1.08	10.14	6.20
5	1.0	5.0	0.24	0.16	0.04	0.21	2.04	1.18
5	5.0	1.0	1.19	0.78	0.19	1.08	10.14	6.20
10	1.0	1.0	0.21	0.08	0.05	0.02	1.73	0.08
10	5.0	5.0	0.72	0.27	0.31	0.18	6.33	0.48
14	1.0	1.0	0.07	0.06	0.08	0.03	0.77	0.02
14	5.0	5.0	0.34	0.30	0.39	0.17	3.80	0.12
30	1.0	1.0	0.03	0.05	0.06	0.06	0.48	0.06
30	5.0	5.0	0.16	0.24	0.31	0.32	2.39	0.32

Table 7: Average Absolute Error of Translation and Rotation for the R–and–T Algorithm.

The average for each experiment is taken over 100 samples of uniform noise.

4 Conclusions

This section presents the conclusions drawn from the research performed under this contract.

- **Most motion is not translational.** There is no such thing as uniform translational motion, except in very strictly controlled situations. In the absence of a gyro–stabilized sensor, there are usually rotational motion components in excess of 1° for real image sequences. Algorithms which assume uniform translational motion in order to calculate quantitative information can be expected to performly poorly in such realistic situations. They will therefore be of little use for tasks which require accurate quantitative information, such as computing structure from motion, unless the objects are quite close to the sensor. They may be of some use for more qualitative tasks such as avoidance of distant objects, or for navigation.

- **General Motion is Necessary.** In practical situations, general motion algorithms will be necessary for any quantitative task. Our combination of the Anandan and Adiv algorithms to obtain a general motion algorithm shows promise and seems to be able to find environmental depth with an error of less than about 10%.

- **Stereoscopic Motion May Be Useful.** This is an alternative to the general motion algorithms. Although we do not yet have much experimental data on algorithms which combine stereo and motion, we think the initial results are promising.

- **Longer Image Sequences Should Improve Robustness.** One way of achieving good performance for monocular image sequences is to use longer image sequences. The additional information and constraints provided by such sequences should lead to more robust results.

- **Algorithms Must Be Evaluated Scientifically.** Accurate ground truth is needed to have a quantitative metric for the evaluation of an algorithm's performance. The scientific evaluation of such an algorithm cannot be performed if you don't know what you were supposed to get.

- **Landmarks are Useful.** The use of landmarks in model–based vision appears to be feasible. This means that models of the environment are needed. The acquisition of such models is a non–trivial problem in itself.

- **The Decomposition of 2–D and 3–D Processing is Useful for Navigation.** The process of correspondence between image data and model data is complicated by sensory data that are noisy (e.g., skewed and translated lines), fragmented, and missing elements. The recovery of 3–D pose can be simplified if the problem is decomposed into 2–D optimization of line correspondences during model–matching, followed by 3–D optimization of the robot's position and orientation.

5 Recommendations

In this section, we detail our recommendations for the direction in which motion research supported by the present contract should head. First we outline the directions for motion research, and then we present recommendations for future research in mobile robotics.

5.1 Directions for Motion Research

- **Motion algorithms must be precise.** Motion algorithms that derive depth from an analysis of sensor motion must be capable of recovering the parameters of general motion with rotational accuracies of much less than one degree. If the algorithm cannot perform to this level, it will be difficult to recover the environmental depth of surfaces that are at medium distances from the sensor (for example, 40 feet or more from the sensor, when the sensor moves 2 feet between frames). The general motion algorithm of Adiv has shown significant promise in recovering the depths of outdoor objects with less than 10% error. The robustness of such general motion algorithms must be carefully evaluated on many sequences of controlled image data.

- **Motion algorithms must be compared with ground truth.** The motion data set obtained by us at Martin Marietta has known ground truth for both environmental depth and sensor motion parameters. It can therefore serve as the touchstone for the *scientific* evaluation of the accuracy of motion algorithms. This data set is being made widely available; we intend to utilize it extensively.

- **Motion and stereo should be used together.** Efforts to combine motion and stereo should be extended from the analysis of synthetic laboratory data and applied to real scenes. Such algorithms also promise the possibility of dealing with general motion. The goal here should be an algorithm that initially recovers a coarse approximation to surfaces over the first few frames of the image sequence, and then continuously refines the surface to form a better approximation. The detection of occlusion boundaries and depth discontinuities will be critical to the success of this effort. The performance of such algorithms should be compared with the performance of general motion algorithms (such as that of Adiv) for the recovery of sensor motion and environmental depth.

- **Trajectories should be used.** Long temporal sequences should be useful for any motion motion algorithm. The development of token–based tracking algorithms (such as the line–tracking algorithm of Williams and Hanson) is needed to extract the trajectories of tokens across sequences. As two tokens of the same type cross each other, as frequently occurs, the match becomes ambiguous and the tracking sequence is disrupted. If the image trajectories were fit via smooth curves, they could be unambiguously matched, and in fact their crossing and occlusion could be predicted.

Obtaining the trajectories of tokens can also provide critical information for the organization of moving objects and the recovery of their natural coordinate systems.

- **Approaches to top–down surface extraction for both static and moving objects should be investigated.** The goal here would be to make use of a static 3–D representation of the environment, and the approximate location of the vehicle, which is often available. In addition, the system could be provided with a model of objects that are capable of locomotion, such as people, cars, or bicycles. Thus, direct extraction of the motion parameters of a surface may be possible by using specific or general surface models. Furthermore, the extraction and refinement of the depth of the surfaces would be enhanced by jointly processing the image motions of a set of points, or an area, with the knowledge of the possible or probable surface models that can explain the image data.

5.2 Directions for Mobile Robot Research

- **Evaluate the efficacy of using accurate 3–D knowledge of the environment.** The 3–D representations and knowledge base must serve as a map for path planning and navigation, as well as for maintaining descriptions of objects for goal and landmark recognition. We intend to use this representation (using GeoMeter) to capture two local environments, the interior hallways of our building, and the outside of our building, for experiments in vehicle navigation and to test a variety of navigation tasks.

- **Use a wider range of knowledge about the environment, e.g., color and texture.** The model of the environment can be enriched with information that represents more qualitative spatial constraints than those obtained using a 3–D modeller (such as GeoMeter). This information can be captured in a manner similar to the road scene models in the current knowledge base of the VISIONS system. By using this methodology, the areas of the image which cannot be conveniently represented as wire–frame models, such as vegetation or distant mountains, can all be added to the tight geometric models to provide additional knowledge for object recognition and navigation.

- **Further develop landmark–based navigation strategies.** The ability to relate image events to stored models of objects and landmarks will be crucial to utilizing the knowledge of the environment that is stored in a map. If specific landmarks can be recognized, then their location on a map can be used to determine the location of the vehicle in the environment, or at least to reduce the uncertainty in the vehicle's position and orientation. In addition, this will be necessary to achieve goals, since the specification of goals will often involve relationships to objects. The accuracy of these landmark–recognition algorithms across a variety of landmark/object models and at a range of distances from the sensor should be evaluated. We expect to demonstrate that a 3–D model and model–based vision algorithms can be used to effectively navigate from an approximately known starting location to another desired location.

- **Supplement model–based algorithms with stereo and motion algorithms.** Model–based algorithms will not work well if an unmodelled object is encountered by the robot. Motion and stereo algorithms should therefore be used to supplement the static recognition of landmarks by providing the depth of points, lines, and surfaces as a function of bottom–up processing of an image sequence. This information would then be useful for such tasks as obstacle avoidance and the automatic acquisition of 3–D models.

- **Use learning to automatically acquire object and scene models.** The information required for object recognition strategies can be time–consuming if constructed entirely by hand. It is possible that a training set of interpreted scenes can be used to automatically acquire object schema knowledge. Some of the attributes of object classes such as color, texture, size, shape, or location relative to other objects may be automatically extracted via the use of multiple examples of instances in a training set. Geometric knowledge can also be acquired during exploration of an environment via motion and stereo processing. Thus object and scene mouels can be continuously acquired and refined during or after each navigational experience.

6 References

REFERENCES

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

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

[3] G. Adiv, "Inherent Ambiguities in Recovering 3-D Motion and Structure from a Noisy Flow Field, " *Proc. of the Computer Vision and Pattern Recognition Conference*, San Francisco, CA, pp. 70–77, June 1985.

[4] G. Adiv, "Inherent Ambiguities in Recovering 3–D Motion and Structure from a Noisy Flow Field," *IEEE Trans. Patt. Anal. Mach. Intel.*, Vol. PAMI–11 (5), pp. 477–489, May 1989.

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

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

[7] P. Anandan, "Motion and Stereopsis," COINS Technical Report 85-52, University of Massachusetts, Amherst, December 1985; also to appear (in Spanish) in *Vision por Computador*, (Carme Torras, ed.), to be published by Alianza Editorial, Spain.

[8] P. Anandan, "Measuring Visual Motion From Image Sequences," Ph.D. Dissertation, University of Massachusetts, Amherst, January 1987.

[9] P. Anandan, "A Unified Perspective on Computational Techniques for the Measurement of Visual Motion," *Proc. of the International Conference on Computer Vision*, London, England, pp. 219-230, June 1987.

[10] P. Anandan, "A Computational Framework and an Algorithm for the Measurement of Visual Motion," *International Journal on Computer Vision*, Vol. 2, pp. 283–310, 1989.

[11] R. Arkin, "Path Planning for a Vision-Based Autonomous Robot," *SPIE Conference on Advances in Intelligent Robotics Systems-Mobile Robots*, Cambridge, MA, October 1986, also COINS Technical Report 86-48, University of Massachusetts, Amherst, October 1986.

[12] R. Arkin, "Path Planning and Execution for a Mobile Robot: A Review of Representation and Control Strategies," COINS Technical Report 86-47, University of Massachusetts, Amherst, October 1986.

[13] R. Arkin, "Towards Cosmopolitan Robots: Intelligent Navigation in Extended Man-Made Environments," Ph.D. Thesis, Computer & Information Science Department, University of Massachusetts at Amherst, September 1987, also COINS Technical Report 87-80, University of Massachusetts, Amherst, September 1987.

[14] R. Arkin, A. Hanson, and E. Riseman, "Visual Strategies for Mobile Robot Navigation," *Proc. of IEEE Computer Society Workshop on Computer Vision*, pp. 176-181, Miami, FL, November 1987.

[15] R. Arkin, "Reactive/Reflexive Navigation for an Autonomous Vehicle," *American Institute of Aeronautics and Astronautics Computers in Aerospace*, pp. 298-306, Wakefield, MA, October 1987.

[16] R. Arkin, "Motor Schema-Based Navigation for a Mobile Robot: An Approach to Programming by Behavior," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 264-271, Raleigh, NC, March 1987.

[17] R. Arkin, E. Riseman, and A. Hanson, "AuRA: An Architecture for Vision-Based Robot Navigation," *Proc. of the DARPA Image Understanding Workshop*, pp. 417-431, Los Angeles, CA, February 1987, also COINS Technical Report 88-07, University of Massachusetts, Amherst, June 1987.

[18] R. Arkin, "Neuroscience in Motion The Application of Schema Theory to Mobile Robotics," in *Visuomotor Coordination: Amphibians, Experiments, Models and Robots*, (P. Evert and M. Arbib, Eds.), Plenum Publishing Co., 1988.

[19] R. Arkin, "Navigational Path Planning for a Vision-Based Mobile Robot," *Robotica*, 1988.

[20] R. Arkin, "Spatial Uncertainty Management for a Mobile Robot and its Role in Expectation-Based Perception," *Symposium on Robot Control '88 (Syroco)*, Karlsruhe, W. Germany, October 1988.

[21] R. Arkin, "Homeostatic Control for a Mobile Robot: Dynamic Replanning in Hazardous Environments," *Proc. of the SPIE Conference on Mobile Robots III, Cambridge Symposium on Advances in Intelligence Robotics Systems*, Cambridge, MA, 1988.

[22] R. Arkin, "Motor Schema-Based Mobile Robot Naviation," *International Journal of Robotics Research*, 1988.

[23] P. Balasubramanyam, "Extraction of Motion In Depth: A First Step in Stereoscopic Motion Interpretation," presented at the American Institute of Aeronautics and Astronautics Computers in Aerospace IV Conference, pp. 277-286, Wakefield, MA, October 1987.

[24] P. Balasubramanyam, "Computation of Motion-In-Depth Parameters Using Stereoscopic Motion Constraints," *Proc. of IEEE Computer Society Workshop on Vision*, pp. 349-351, Miami, FL, November 1987.

[25] P. Balasubramanyam and M. Snyder, "Computation of Motion in Depth Parameters: A First Step in Stereoscopic Motion Interpretation," *Proc. of the DARPA Image Understanding Workshop*, pp. 907-919, Cambridge, MA, April 1988.

[26] P. Balasubramanyam and R. Weiss, "Early Identification of Occlusion in Stereo-Motion Image Sequences," *Proc. of DARPA Image Understanding Workshop*, pp. 1032-1037, Palo Alto, CA, 1989.

[27] M. Boldt and R. Weiss, "Token-Based Extraction of Straight Lines," COINS Technical Report 87-104, University of Massachusetts at Amherst, October 1987.

[28] S. Bharwani, A. Hanson, and E. Riseman, "Refinement of Environmental Depth Maps over Multiple Frames," *Proc. DARPA IU Workshop,* pp. 413-420, Miami Beach, FL, December 1985.

[29] S. Bharwani, E. Riseman, and A. Hanson, "Refinement of Environmental Depth Maps Over Multiple Frames," *Proc. of the Workshop on Motion: Representation and Analysis*, pp. 73-80, Charleston, SC, May 1986.

[30] J. R. Beveridge, R. Weiss, and E. Riseman, "Optimizing Two–Dimensional Model Matching," *Proc. DARPA Image Understanding Workshop,* pp. 815-830, Palo Alto, CA, May 1989.

[31] J. R. Beveridge, R. Weiss, and E. Riseman, "Matching and Fitting Sets of 2D Line Segments to Broken and Skewed Data," Dept. of Computer and Information Science, University of Massachusetts, Amherst, Tech. Rep. in preparation.

[32] J. R. Beveridge, J. Griffith, R. Kohler, A. Hanson, and E. Riseman, "Segmenting Images Using Localized Histograms and Region Merging," *International Journal of Computer Vision*, Vol. 2, pp. 311-347, 1989.

[33] C. Connolly and R. Weiss, "Geometer: A System for Modelling and Algebraic Manipulation," *Proc. of DARPA Image Understanding Workshop*, pp. 797-804, Palo Alto, CA, May 1989.

[34] R. Dutta, R. Manmatha, E. Riseman, and M.A. Snyder, "Issues in Extracting Motion Parameters and Depth from Approximate Translational Motion," *Proc. DARPA Image Understanding Workshop*, pp. 945-960, Cambridge, MA, April 1988.

[35] R. Dutta, R. Manmatha, L. Williams, and E. Riseman, "A Data Set for Quantitative Motion Analysis," *Proc. of Conference on Computer Vision and Pattern Recognition*, pp. 159-164, San Diego, CA, June 1989.

[36] R. Dutta, R. Manmatha, L. Williams, and E. Riseman, "A Data Set for Quantitative Motion Analysis," *Proc. of DARPA Image Understanding Workshop*, pp. 714-720, Palo Alto, CA, May 1989.

[37] C. Fennema, E. Riseman, and A. Hanson, "Planning with Perceptual Milestones to Control Uncertainty in Robot Navigation," *Proc. of AAAI Spring Symposium Series*, pp. 19–23, Stanford University, Palo Alto, CA, March 1989. Also *Proc. SPIE International Society for Photographic and Industrial Engineering*, Cambridge, MA, November 1988.

[38] C. Fennema, A. Hanson, and E. Riseman, "Model Directed Mobile Robot Navigation," submitted to *IEEE Systems, Man, and Cybernetics*, 1989.

[39] C. Fennema, A. Hanson, and E. Riseman, "Towards Autonomous Mobile Robot Navigation," *Proc. of the DARPA Image Understanding Workshop*, pp. 219-231, Palo Alto, CA, May 1989.

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

[41] F. Glazer, "Hierarchical Motion Detection," Ph.D. Dissertation, Computer and Information Science Department, University of Massachusetts, Amherst, February 1987.

[42] F. Glazer, "Hierarchical Gradient-Based Motion Detection," COINS Technical Report 87-77, University of Massachusetts, Amherst, 1987.

[43] F. Glazer, "Computation of Optic Flow by Multilevel Relaxation," Coins Technical Report 87-64, University of Massachusetts, Amherst, 1987.

[44] A. R. Hanson and E. M. Riseman, "The VISIONS Image Understanding System," in *Advances in Computer Vision*, C. Brown (Ed.), Erlbaum Press, 1987. See also UMass, Amherst COINS Tech. Rep. 86–62, Dec. 1986.

[45] B. K. P. Horn, "Closed-Form Solution of Absolute Orientation Using Unit Quarternions," *J. Opt. Soc. Am.*, Vol. 4, pp. 629-642, 1987.

[46] P. Kahn, L. Kitchen, and E. Riseman "Real-Time Feature Extraction: A Fast Line Finder for Vision-Guided Robot Navigation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1988.

[47] R. Kumar, "Determination of Camera Location and Orientation," *Proc. of DARPA Image Understanding Workshop*, pp. 870-881, Palo Alto, CA, May 1989.

[48] D.T. Lawton, "Processing Translational Motion Sequences," *Computer Graphics and Image Processing*, Vol. 22, pp. 116-144, 1983.

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

[50] Y. Liu, T. Huang, and O. Faugeras, "Determination of Camera Location From 2D and 3D Line and Point Correspondences," *Proc. of CVPR*, pp. 82-88, Ann Arbor, MI, June 1988.

[51] I. Pavlin, A. Hanson, and E. Riseman, "Analysis of an Algorithm for Detection of Translational Motion," *Proc. DARPA IU Workshop*, pp. 388-398, Miami Beach, FL, December 1985.

[52] I. Pavlin, "Motion From a Sequence of Images," *Proc. of the DARPA Image Understanding Workshop*, pp. 930-937, Cambridge, MA, April 1988.

[53] H. Sawhney and J. Oliensis, "Description and Interpretation of Rotational Motion from Image Trajectories," *Proc. of DARPA Image Understanding Workshop*, pp. 992-1003, Palo Alto, CA, May 1989.

[54] H. Shariat, "The Motion Problem: How To Use More Than Two Frames," Ph.D. Dissertation, University of Southern California, Los Angeles, 1987. Also Technical Report IRIS 202, Institute for Robotics and Intelligent Systems, University of Southern Calfornia, Los Angeles, CA, October 1986.

[55] M. A. Snyder, "The Accuracy of 3-D Parameters in Correspondence-based Techniques," COINS Tech. Rep. 86-28, University of Massachusetts, Amherst, MA, June 1986.

[56] M. A. Snyder, "Uncertainty Analysis in Image Measurements," *Proc. DARPA Image Understanding Workshop*, pp. 681-693, Los Angeles, CA, January 1987.

[57] M. A. Snyder, "On the Mathematical Foundations of Smoothness Constraints for the Determination of Optical Flow and for Surface Reconstruction," *Proc. of DARPA Image Understanding Workshop*, pp. 1004-1011, Palo Alto, CA, May 1989. Also in *Proc. IEEE Workshop on Visual Motion*, pp. 107-115, Irvine, CA, March 1989, and UMass COINS Tech. Rep. 89–07, January 1989.

[58] M. A. Snyder, "The Precision of 3–D Parameters in Correspondence–Based Techniques: The Case of Uniform Translational Motion in a Rigid Environment," *IEEE Trans. Patt. Anal. Mach. Intel.*, Vol. PAMI–11 (5), pp. 523–528, May 1989.

[59] D. Thompson and J. Mundy, "Three Dimensional Model Matching From an Unconstrained Viewpoint," *Proc. of the IEEE Conference on Robotics and Automation*, Raleigh, NC, 1987.

[60] C. Thorpe, S. Shafer, and T Kanade, "Vision and Navigation for the Carnegie Mellon Navlab," *Proc. of the DARPA Image Understanding Workshop*, pp. 143-152, Los Angeles, CA, February 1987.

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

[62] A.M. Waxman and J.H. Duncan, "Binocular Image Flows: Steps Toward Stereo-Motion Fusion," *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol. PAMI-8, pp. 715-729, November 1986.

[63] C. Weems, S. Levitan, A. Hanson, E. Riseman, G. Nash, and D. Shu, "The Image Understanding Architecture," *International Journal of Computer Vision*, **2**, pp. 251-282, 1989. Also COINS Technical Report 87-76, Computer & Information Science, University of Massachusetts, Amherst, August 1987.

[64] R. Weiss and M. Boldt, "Geometric Grouping Applied to Straight Lines," *Proceedings on the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 489-495, Miami, Florida, June, 1986.

[65] L. Williams and A. Hanson, "Depth From Looming Structure," *Proc. of the DARPA Image Understanding Workshop*, pp. 970-980, Cambridge, MA, April 1988.

[66] L. Williams and A. Hanson, "Translating Optical Flow Into Token Matches," *Proc. of the DARPA Image Understanding Workshop*, pp. 1047-1051, Cambridge, MA, April 1988.