Progress in Computer Vision at the University of Massachusetts

Edward M. Riseman
Allen R. Hanson

COINS TR90-100

October 1990

Progress in Computer Vision at the University of Massachusetts[1]

Edward M. Riseman and Allen R. Hanson
Computer Vision Research Laboratory
Dept. of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

ABSTRACT[1]

This report summarizes progress in image understanding research at the University of Massachusetts over the past year. Many of the individual efforts discussed in this paper are further developed in other papers in this proceedings. The summary is organized into several areas:

1. Mobile Robot Navigation
2. Motion and Stereo Processing
3. Knowledge-Based Interpretation of Static Scenes
4. Image Understanding Architecture

The research program in computer vision at UMass has as one of its goals the integration of a diverse set of research efforts into a system that is ultimately intended to achieve real-time image interpretation in a variety of vision applications.

1. Mobile Robot Navigation

The initial focus of the mobile robot navigation project (Fennema and Hanson 1990b) has been on the development of a system for goal oriented navigation through a partially modeled, unchanging environment which contains no unmodeled obstacles. This simplified environment is intended to provide a foundation for research into navigation in more complicated domains. The guiding philosophy of this project is a tight coupling between planning, perception, and plan execution. Incremental planning and vehicle motion, guided by the relationship between the internal model and the external world provided by perception, serve to keep the vehicle accurately located within the environmental reference frame.

1.1 Experiments in Planning and Plan Execution

In a recent experiment (Fennema and Hanson 1990b), the vehicle successfully navigated a multi-leg course, from the robot laboratory to an office, moving approximately 50 feet through two doorways and coming to rest within an inch of its intended goal. Both rooms and the hallway were accurately (but incompletely) modelled using a hierarchical volumetric representation of space. Volumes are represented by their surrounding faces, and faces contain information about the visual appearance of the face. A more complete description of the world representation may be found in (Fennema and Hanson 1990a; Fennema and Hanson 1990b). Clearly, many more experiments, under widely varying environmental conditions (both indoors and outdoors), must be run before the robustness of the techniques can be established.

1.1.1 Plan Generation

Planning is carried out over the world model using traditional planning techniques (e.g. A* search, freespace representations, etc.) to generate a sequence of plan 'sketches' (incompletely specified

[1]This research has been supported in part by the Defense Advanced Research Projects Agency under RADC contract F30602-87-C-0140 and Army ETL contracts DACA76-89-C-0016 and DACA76-89-C-0017.

plans). Plans are generated in a depth first manner, with more detailed plans closer to the vehicle's current location. Associated with each plan sketch is a milestone, which can be thought of as a precondition for the execution of the plan sketch. These milestones are typically specified as landmarks which must be verified visually (and the vehicle's position relative to them determined) prior to the execution of the next step in the plan. The milestones form the basis for 'plan-level servoing', discussed below.

1.1.2 Plan Execution and Perceptual Servoing

The rationale for not fully developing detailed plans prior to moving the vehicle is that plans fail. Obstacles in the planned path, irregular or slippery surfaces, uneven tire inflation, or unexpected externally induced vehicle motions can throw the vehicle off course, causing inaccurate execution. To reduce the errors caused by these unexpected events, the execution of each action is controlled by 'servoing' on prominent visual features in the environment. These features may be objects, such as prominent buildings, or they may be local features, such as easily identifiable corners, door frames, or baseboards. Servoing occurs on three nested levels:

'action-level servoing' is used to maintain the accuracy of each primitive action executed by the vehicle but does not relate the vehicle's position to its progress towards the goal;

'plan-level servoing' uses the milestones defined in the plan to relate the current location of the robot to the plan and environmental model;

'goal-level servoing' attempts to relocate the robot when it becomes lost; this level of servoing is not discussed here.

1.1.2.1 Action-Level Servoing

The Denning Mobile Robotics vehicle used in these experiments can execute two 'primitive' actions directly: TURN θ and a straight line MOVE d. Neither of these actions can be reliably executed in 'open loop' mode. The straight line MOVE action, for example, results in a curved path when executed and the vehicle can be significantly off the intended straight line trajectory at the end of the action. In actual experiments, executing a MOVE 40' has resulted in the vehicle being as much as a foot off the intended straight line after 20', with the error increasing.

In action-level servoing, the primitive action MOVE 40', for example, is broken up into a sequence of smaller moves, say MOVE 2'. The 2D appearance information contained in the environmental model is used to generate two dimensional correlation templates for prominent visual features. From the predicted location of these features in the image, a search window is established and the templates are correlated with the image to establish their image location. Using the measured discrepancy between predicted and actual locations, the heading of the vehicle is modified to reduce the error and the next sub-action is executed. The process is repeated until the primitive action is complete. In actual experiments, the use of action-level servoing has maintained the vehicle within 1/4" of the intended straight line motion over a 40' move. Details on action-level servoing and more complete experimental results may be found in (Fennema and Hanson 1990a).

1.1.2.2 Plan-Level Servoing

Plan-level servoing is designed to ensure proper execution of a plan step prior to initiating the next step by relating the progress of the vehicle towards the goal to the environmental model. This is accomplished by matching the milestones defined in the plan sketch to the image (2D matching) followed by a 3D pose refinement step to determine the relationship between the vehicle and the environmental frame of reference.

Two different approaches to 2D model matching have been developed. The most recent approach has been to use the same feature extraction and 2D correlation methods used in action-level servoing. Since the vehicle has been tracking these points during action-level servoing, it is unlikely that there will be a large discrepancy between where the vehicle believes it is and where it actually is. Consequently, plan-level servoing involves only the additional step of 3D pose refinement, using the matched points, in order to recover its position. In an initial set of indoor experiments described in (Fennema and Hanson 1990a), the vehicle was able to recover its position to within a quarter of an inch after being displaced up to 6 inches from where it thought it was, using landmarks approximately 30 feet away. Additional experiments with this technique are planned for the near future.

An earlier approach (Beveridge, Weiss et al. 1989; Beveridge, Weiss et al. 1990) matched straight lines extracted from the image to model lines projected to the image plane using the assumed location of the vehicle. During this past year, the two-dimensional matching scheme has been extended to include determination of scale, as well as the rotation and translation parameters yielding the best fit. The model-to-image line correspondences determined during 2D matching are used as the input to the 3D pose computation step. The emphasis over the past year has been on improving both the reliability and efficiency of the search processes.

The 3D pose refinement technique developed earlier works with either points or lines (Kumar 1989; Kumar and Hanson 1990a; Kumar and Hanson 1990b; Kumar and Hanson 1989a) and has been extended to be robust in the presence of outliers. The robustness is achieved at a computational cost, since the median of the error function is minimized by combinatorial methods over the subset space of all matched image and model lines. However, the method is capable of handling up to 49.9% outliers. In a recent paper (Kumar and Hanson 1990a), the superiority of the least-median squares algorithm over traditional least-mean squares algorithms as well as those based on statistical M-estimation techniques was established. The sensitivity of pose refinement and other related 3D inference methods to inaccurate estimates of the image center and focal length has been theoretically established and experimentally validated (Kumar and Hanson 1990b). The results show that for 'small' field of view imaging systems, incorrect knowledge of the camera center does not affect the recovered location of the camera significantly. The error in the recovered orientation of the camera is linearly related to the error in the estimate of the location of the center of the imaging system.

1.2 Automated Model Extension

The construction of positionally accurate environmental models is a time consuming, tedious task. Ultimately, the only feasible approach for vehicles which are required to interact with large scale changing environments is to provide them with methods for automatically acquiring their internal models during goal-oriented activities or unrestricted exploration.

Two preliminary experiments have been performed using the 3D pose refinement algorithm to extend a partial model from a set of known points to include unknown points; these experiments are described in more detail in (Kumar and Hanson 1990b). The known model points are used to locate new points in the world coordinate system from pose refinement and triangulation over the induced stereo baseline obtained from a pair of 3D poses (e.g. location and orientation of the camera for each image). In both experiments, an image sequence was obtained for which the three-dimensional location of a set of points in the environment was known (the model). Image features are tracked over a sequence of frames using a token-based line tracker (Williams and Hanson 1988a; Williams and Hanson 1988b; Williams and Hanson 1988c), which provides the token correspondences. The 3D pose estimation algorithm described earlier is applied to each frame to map each feature into a stable world coordinate frame. The 3D pseudo-intersection of the rays passing through the camera center and the image feature point in each image frame is found using an optimization technique which minimizes the sum-of-squares of the perpendicular distances from the 3D pseudo-intersection point to the rays. In effect, this induces a stereo baseline between frames from which the 3D coordinates of the unknown features can

be obtained by triangulation. Note that the computation of the location of new points in the world coordinate system is not sensitive to accurate estimation of the image center.

1.3 Automatic Acquisition of Environmental Models

The problem of acquiring models or modifying incorrect models is an important aspect of object recognition and navigation. The major functional requirements of modeling for these tasks are: accurate prediction of visual features, accurate surface orientation and curvature, and accurate feature dimensions.

We are currently building a system for acquisition of models from image data under known motion generated by a camera mounted on an arm in a robot workcell. In order to obtain accurate depth and curvature information, an extension of the Giblin and Weiss algorithm (Giblin and Weiss 1987) is being used. This algorithm computes depth and curvature by tracking contours in three successive images. The surfaces need not be smooth and the algorithm can use creases (tangent discontinuities) as well as extremal contours and surface markings. This produces 3-D contours and curvatures to which a surface can be fit.

There are many types of surface that one can fit to this 3-D data. We have chosen a representation based on a vertices, edges, and faces. This type of model is supported by Geometer (Connolly 1989; Connolly, Kapur et al. 1989) which provides an environment that includes both planar and algebraic faces.

1.4 Image-Based Navigation Using 360 Degree Views

Mobile robot navigation has proven to be a difficult task, and when a system must be capable of automatically acquiring 3D environmental models, it is currently beyond the state-of-the-art. We are developing a quite novel approach to robot navigation (Hong, Tan et al. 1990) that allows environmental information to be acquired in terms of a set of images of the world taken at a set of target locations. The robot navigates through the world by moving between neighboring target locations using an image-based local homing algorithm. Such an approach is feasible only because the system utilizes an unusual imaging system that provides 360 degree views of the scene in an extremely compact form.

The imaging system is comprised of a spherical mirror mounted above a video camera that is pointed upwards so that a 360^{o} hemispherical view of the world is obtained as a circular extreme "fish-eye" image. This spherical imaging system so greatly distorts the scene during projection that the image changes dramatically as the robot moves, while maintaining visibility of the whole environment so that both objects that are in the path of movement as well as objects just passed will remain visible. There is, however, a projective invariant on the horizon line, or in this case the horizon circle. As the robot moves on a planar ground surface, distinctive world features (i.e. landmarks) that project to points on the horizon circle remain on the circle. In addition, each feature other than the points directly in front of and behind the robot slide around the horizon circle as a function of the robot movement and surface distance. A one-dimensional circular "location signature" is extracted from the hemispherical image by sampling along the horizon circle at angular intervals (in our experiments 1^{o}), allowing any resolution image to be compressed into a 360-byte location signature.

Large scale navigation is then decomposed into a sequence of small-scale navigation tasks by local homing. Around each target location, there is a "capture radius" that allows comparison of landmarks in the current and target location signatures to determine a motion to reduce the difference and thereby home in on the local target in a series of small steps. Thus, a compact 360^{o} representation of the environment and an image-based qualitative homing algorithm allows a mobile robot to navigate without explicitly inferring three-dimensional structure from the image. Experiments in typical indoor

rooms and corridors have been successful along paths that involve as many as 17 target locations for incremental homing. This research is an ongoing effort and the feasibility of sampling two-dimensional space for general goal-oriented navigation is being examined.

2. Motion and Stereo Processing

2.1 A Framework for the Integrated Processing of Stereo and Motion

Work is in progress on understanding the *dynamics* of a scene as viewed by a stereo pair of cameras undergoing arbitrary motion. This subsumes both the analysis of static stereo imagery at one time instant to obtain the static disparity between the two images and thereby depth, and the analysis of a monocular motion pair to obtain the optic flow for a pair of frames and thereby relative motion and depth. Thus, we are specifically interested in a reconstruction paradigm which can be categorized as *binocular motion*, in order to obtain additional constraints on the recovery of motion and depth without depending on one unique (and possibly erroneous) source for the depth.

A promising approach utilizes the ratio of the relative flow between the image pairs to the disparity as a function of the motion in depth parameters (Balasubramanyam and Snyder 1988; Waxman and Duncan 1986). The vectors parallel to the real instantaneous 3D velocity scaled by the depth of the point, located at the image of the 3D point, can be extracted using purely image measurable quantities. This field of scaled 3D vectors is called the *p-field*. The p-field is interesting from the point of view of binocular motion since it implies that at the image level, where normally only 2D entities were available, it is now possible to examine and exploit the nature of 3D phenomena directly.

We are currently examining the use of the p-field as a framework within which to represent both the problem of occlusion and discontinuity (Balasubramanyam and Weiss 1989) detection and flow/disparity computation as well as computation of the 3D motion itself. For instance, observing that the p-vector is a scaled version of the real 3D motion vector, it seems more appropriate to impose smoothness on this vector since this is closer to the assumption of smooth 3D motion, rather than on flow smoothness. This was briefly examined in (Scott 1986) but not within the framework of binocular motion.

It may be possible to use the p-field for the interpretation of available flow and disparity information for the estimation of the motion parameters. For instance, in the case of ideal pure translation, the p-field directly yields the direction of translation. In the case of general motion, we are examining several possible algorithms for the computation of the motion parameters.

2.2 Smoothness Constraints For Optical Flow & Surface Interpolation

Gradient-based approaches to the computation of optical flow often use a minimization technique incorporating a smoothness constraint on the optical flow field. Smoothness constraints are also of interest in surface interpolation, where they are known as ``performance functions." All known smoothness constraints used to compute optical flow have a subtle property, namely that they do not mix derivatives of different components of the optical flow field.

Snyder (Snyder 1990) presents an analysis of smoothness constraints which do not satisfy this 'decoupled' property, but rather in which derivatives of different components of the flow can interact. By using representation theory of the group of Euclidean motions in the image plane, he uses the single assumption that the smoothness constraint is invariant under this group of transformations to generate a *complete* list of all possible invariant smoothness constraints.

The constraints are represented as type (p,q), by which it is meant that they are quadratic in p^{th} derivatives of the optical flow field, and in q^{th} derivatives of the grey level image intensity function.

This is done explicitly for the values $0 \leq p,q \leq 2$. It appears that of these smoothness constraints, excepting those linear combinations which are decoupled, are new. In addition, he finds all invariant ``performance measures" used in surface interpolation, when the performance measure is quadratic in no higher than fourth derivatives of the object function.

2.3 Orientation Statistics for Modeling 3D Lines and Planes

One useful technique for deriving orientation information from static images is the estimation of a unit vector perpendicular to a number of derived unit vectors. For instance, under perspective projection a ray pointing towards the intersection of a group of converging image line segments is perpendicular to their projection plane normals. This has applications in finding vanishing points and in locating the focus of expansion of a pure translational flow field. Furthermore, the normal to a planar surface is perpendicular to the direction of all lines lying on that surface.

The problem of estimating a vector mutually perpendicular to several unit vectors can be characterized as estimating the polar axis of a great circle on the unit sphere. Bingham's distribution, which represents the intersection of a 3D Gaussian distribution with the surface of the unit sphere, is introduced to describe both equatorial and bipolar distributions of unit vectors. Statistical parameter estimation based on Bingham's distribution can be used to solve for the polar axis of a great circle of points and to represent the statistical uncertainty in the orientation estimate, but the procedure is somewhat expensive computationally. Collins and Weiss (Collins and Weiss 1990) develop a more convenient alternative based on linear-least-squares plane fitting. In addition, they consider the problem of estimating the orientation and uncertainty of the cross product of two uncertain unit vectors. The tentative solution is to form a Gaussian approximation to the ``intersection" of two equatorial Bingham distributions.

The above methods are illustrated using two examples (Collins and Weiss 1990). The first involves reconstruction of planar surfaces using stereo line correspondences . If relative pose of the stereo cameras can be described as a pure translation, then the orientation of lines in the world can be computed as the cross product of the projection plane normals of its two corresponding images, one in each image plane. Given a set of lines hypothesized to lie on a single planar surface, the plane orientation and uncertainty can be computed as the pole of a great circle formed from the uncertain line orientation estimates.

The second example involves the analysis of vanishing points (Collins and Weiss 1989). The images of parallel 3D lines converge to a vanishing point in the projective image plane. A ray constructed from the camera focal point towards the vanishing point has the same 3D orientation as the original world lines. The line orientation and its approximate confidence region on the unit sphere is estimated as the polar axis of a great circle of projection plane normals. Furthermore, surface plane orientations are hypothesized as the cross product of these uncertain line directions.

2.4 Analysis of the Limits of Robustness of Correspondence-Based Structure from Motion

In spite of extensive research in correspondence-based motion analysis, a comprehensive algorithm-independent study of the theoretical limits on the accuracy of the computation of environmental depth is *not available*. In response to this situation, Dutta and Snyder have been examining (Dutta and Snyder 1990) the robustness of correspondence-based approaches to structure from motion.

Their analysis shows, in an *algorithm--independent* way, that small absolute errors in image displacements cause absolute errors in rotational motion parameters significant enough to lead to large relative errors in the determination of environmental depth. Even if the motion parameters are known almost exactly, such as by sophisticated navigation systems, small errors in image displacements still

lead to large errors in depth for environmental points whose distance from the camera is greater than a few multiples of the the total translation in depth of the camera.

2.5 Comparative Results of Four Motion Algorithms

In an earlier paper, Sawhney (Sawhney and Oliensis 1989) presented a new technique for recovering motion and structure through image trajectories of rotational motion. A closed form solution was presented for the problem of recovering the 3D circular trajectory of a point given its conic trajectory in the image plane. It was also demonstrated that when small sections of the conic arc in the image are used as the input for a trajectory description, one obtains very unreliable estimates of the underlying trajectory. Hand-grouped sets of trajectories were used and it was conjectured that if spatio-temporal data from proximal points could be grouped and trajectories fit to the grouped data, reliable combined trajectory descriptions and accurate 3D results could be obtained.

An algorithm has been developed which uses commonality of motion to first incrementally group point tracks and then fits conic sections to a subset of these using an optimization technique over a joint error measure. The error measure uses a parameterization which makes the common and independent parameters of each trajectory explicit. The closed form solution was presented in (Sawhney and Oliensis 1989).

The algorithm has been applied to several image sequences; the results are summarized in (Sawhney and Hanson 1990; Sawhney and Oliensis 1990). In addition, the results have been compared with two other motion algorithms: Adiv (Adiv 1985), Horn's relative orientation algorithm (Horn 1988), as well as Kumar's pose refinement algorithm (Kumar and Hanson 1989a; Kumar and Hanson 1989b). The results are preliminary and represent a continuing effort in understanding robust 3D reconstruction from monocular motion. More accurate results applied to a more varied data set awaits precise calibration of our cameras.

3. Interpretation of Static Scenes

3.1 Learning 3D Object Recognition Strategies

A general system for object and scene interpretation, called the Schema System, has evolved as part of a long-term research effort at UMass (Draper 1989; Draper, Brolio et al. 1989; Hanson and Riseman 1978; Hanson and Riseman 1987; Riseman and Hanson 1984). The results of successful experiments in the outdoor scene domain has led to the not surprising conclusion that a declarative representation of knowledge would be more useful for future work, and in particular, automatic mechanisms for learning object recognition strategies (Hanson and Riseman 1989).

The basis for recognizing objects in complex outdoor scenes varies widely in terms of the processes utilized, the reliability of the information extracted, the efficiency of the underlying mechanisms, and the manner in which the evidence is combined into an object hypothesis. All of this information is certainly object- and domain-dependent. Some objects can be distinguished on the basis of color, while others can only be identified by scene and object context. Three-dimensional information about shape or texture of some objects might be recovered through bottom-up vanishing point analysis, while the locations of other objects are more easily determined by model-based point matching.

The problem of learning how to recognize an object is being addressed in (Draper and Riseman 1990) The system is given a description of the object and a set of user-interpreted training images. The task is to build the most efficient object recognition strategy possible within performance constraints set by the user. Three-dimensional 3D object recognition is approached within a generate-and-verify paradigm. The task of learning to generate the minimal necessary set of hypotheses is phrased as a search problem. The task of learning to verify a hypothesis is cast as a classification problem, followed by graph optimization.

3.2 Perceptual Organization of Occluding Contours

Contours corresponding to surface boundaries are readily perceived or completed by human observers even when local evidence in the form of measurable image brightness gradients is completely absent. A classic example of the former is the Kanizsa triangle, in which the illusory contours of the 'occluding' triangle are visually compelling, even though there is scant evidence for their existence. An example of completion occurs when one surface is partially occluded by a second (opaque) surface.

Williams (Williams 1990) has developed a system for perceptually organizing surface boundaries based on figural clues alone, although results have only been demonstrated in the 'Colorforms' domain and other simple scenes. The system has, however, successfully extracted Kanizsa's occluding triangle and has correctly analyzed relatively complex scenes containing multiple occluding surfaces. Detailed results are presented in Williams (Williams 1990). The current system is designed to complete gaps in the straight sections of occluded contours but isn't yet able to cope with more complex occlusions, such as missing corners or missing sides.

In Williams' system, the mechanisms of occlusion of one surface by another are captured in a set of integer linear constraints. These constraints ensure that the outputs of a contour grouping process is physically valid and consistent with the image evidence. Among the many feasible solutions, the most compelling is the solution which best explains the presence and form of the image structure. The problem of computing a complete and consistent surface boundary representation is thus reduced to solving an integer linear program.

3.3 Perceptual Organization of Curves

Dolan (Dolan and Weiss 1989) is extending the perceptual grouping mechanisms developed by Boldt (Boldt, Weiss et al. 1989) for straight lines to the case of general curves. Like the straight line system, the curve grouping algorithm relies on the Gestalt principles of proximity and good continuation and employs an iterative token-based approach to search for and describe significant curve structures (including straight lines, conic arcs, inflections, corners, and cusps).

The system iterates a cycle of linking, grouping, and replacement over a range of perceptual scales, but within each iteration processing occurs independently at each token. Each token is linked to other tokens that are likely to be its neighbors along some contour. Sequences of linked tokens are analyzed and classified based on the geometric structure they exhibit. Appropriate replacement tokens are then generated to explicitly describe and replace each sequence. Beginning with initial edge tokens (unit tangents centered at edge locations), curved structure is discovered in a bottom-up, local-to-global fashion and a multi-scale description results. The computational complexity inherent in any grouping process is managed here by searching locally within a perceptual window (which defines the local scale) and by explicitly replacing a sequence of tokens by a single token at the next scale.

Since the work previously reported in (Dolan and Weiss 1989), a parallel version of the grouping algorithm has been implemented in anticipation of parallel hardware. Here, the grouping process is simultaneously applied to the perceptual window (i.e. context) around each token for potential grouping and replacement, and parallel replacement of the aggregate tokens is assumed to take place simultaneously. A consequence of a highly distributed and parallel grouping process is that redundant descriptions arise because the contexts of nearby tokens overlap, and overlapping aggregate tokens are produced. Dolan is currently developing methods to identify and eliminate such redundancies by representing multiple types of relationships in the link graph; this will allow redundancy, as well textural structures to be dealt with in this parallel framework.

3.4 View Variation of Line Segment Features

Model-directed object recognition becomes much more difficult when the viewpoint of the three-dimensional object is unknown. A popular approach is based upon the use of multiple two-

dimensional views of three-dimensional structures, and is referred to under a variety of terms such as ``aspect graphs", ``generic views", and ``characteristic views" (Burns 1987; Burns and Kitchen 1987a; Burns and Kitchen 1987b; Burns and Kitchen 1988; Ikeuchi 1987). If such systems are going to be effective, a clear understanding is required of the manner in which the features of 2D projections vary as a function of the 3D viewing position of the object. It is important to find metric features of an object whose variation is small over a large range of views in order to constrain the number that must be stored.

Burns (Burns, Weiss et al. 1990) presents an analysis of the variation of point-set and line-segment features with respect to view. Although there are well known special-case invariants for four points, such as the cross ratio, there is no scalar invariant for an arbitrary number of points in general position, whether one uses true perspective, weak perspective or orthographic projection. The paper focuses on variation of features with respect to views of line segment pairs under weak perspective, a commonly used projection model in 3D recognition. The variation is a function of both the particular feature and the particular configuration of 3D line segments. The features studied are: the relative orientation, size and position of one line segment with respect to another, and the affine coordinates of one endpoint in terms of the other three.

The information in the view-variation analysis allows determination of semi-invariant features of an object over areas of the 3D viewing sphere, i.e. features which have a small variation over a large fraction of views. The relationships between the range of feature variation and the fraction of views are presented in a series of graphs for the features described above, and for varying instances of 3D line segments pairs. The mathematical analysis embodied in this paper is generally relevant to techniques for matching 3D models to 2D images.

3.5 Recovering Shape from Shading

Shape from shading has traditionally been considered an ill--posed problem. However, in recent work, Oliensis (Oliensis 1990a; Oliensis 1990c) has demonstrated that the solutions to shape from shading are often well--determined, with little or no ambiguity. For the case of illumination that is symmetric around the viewing direction (i.e. the light source is behind the camera), it was shown in (Oliensis 1989) that there is in general a *unique* solution to shape from shading. This proof is valid for *general* Lambertian objects (without holes), and is the first proof that the problem of shape from shading can be well--posed in general. These arguments were extended to the case of general illumination direction in (Oliensis 1990c), where it was demonstrated that, in this case also, the solutions to shape from shading are strongly constrained over much of the image. These results follow from a combination of local uniqueness theorems and global arguments concerning the properties of the flow of characteristic strips, both derived from the mathematical theory of dynamical systems theory. The essential constraints restricting the solution space are shown to be provided by the *singular points* in the image. Also, characteristic strips are given a simple interpretation as space curves, and demonstrated to be *independent of the viewing direction.*

It has long been an open question whether the image of the occluding boundary provides additional constraints on the solution to shape from shading. In (Oliensis 1990c), it is proven analytically that the answer to this question is negative. Specifically, for a local image patch containing a portion of the boundary, the problem of shape reconstruction is shown to be ill--posed. Shape reconstruction is actually more ambiguous in the neighborhood of an occluding boundary segment than it is in the neighborhood of an interior image curve. The proof, which applies to a Lambertian surface illuminated from a general light source direction, is based on recasting the basic characteristic strip equations of Horn in a form that is completely *non--singular* on the occluding boundary.

Also, an example is presented in (Oliensis 1990c) in which a small image region bordering the image of the occluding boundary yields an ambiguous shape reconstruction, even though the image contains

both singular points and the whole of the occluding boundary. This example demonstrates that shape from shading can be well--posed *and* ill--posed simultaneously: although the shape corresponding to most of the image is actually uniquely determined, the shape corresponding to the specified small image region is ill--determined. It is argued that, in general, these 'ill--posed' regions are probably small fractions of the image, but that they can occur frequently, in images both with and without visible occluding boundaries, and in practice may lead to instabilities and errors in shape reconstruction algorithms.

Finally, Oliensis has developed (Oliensis 1990b; Oliensis 1990c) a new local algorithm for reconstructing shape from shading using a general quadratic surface model. The new constraints for shape from shading should be investigated for their potential for robust surface reconstruction.

4. The Image Understanding Architecture

The Image Understanding Architecture (IUA) consists of three levels of tightly coupled array of parallel processors (Weems, Levitan et al. 1989). Work on the IUA has advanced in four areas in the preceding year through cooperative efforts by Hughes Research Labs and the University of Massachusetts. A generalized routing algorithm for the low-level processor has been developed, an Apply compiler for the low level has been implemented, the IUA simulator and tools have been enhanced, and assembly of the prototype system has begun. We have also started development of a data parallel C for the low level, continued planning of the next generation of the DARPA IU Benchmark, and started the development of the second generation IUA and the design of the third generation.

4.1 Routing On The CAAPP

The low-level processor of the IUA is a square mesh of processing elements, augmented with a second (reconfigurable) mesh, called the Coterie Network (Weems and Rana 1990). Normally, a mesh network is considered to be ill-suited for permutation routing because of the square-root of N diameter of the network. Other architectures, such as the Connection Machine and Masspar have devoted a significant amount of additional hardware to support fast permutation routing. These additions take the form of hypercube or crossbar networks with sophisticated controllers. On the other hand, the Coterie Network merely adds a set of switches and some pre-charge logic to each processor, with no increase in the number of physical connections between processors.

Using the Coterie Network and the fast summary capabilities (Rana and Weems 1990) of the CAAPP, we have developed an adaptive routing algorithm that is at worst an order of magnitude slower than routing on the Connection Machine, and in many cases is up to two orders of magnitude faster. The algorithm is actually a collection of different algorithms, each suited to optimizing performance on different types of permutation. The first step is to quickly identify some gross features of the data to be routed, such as the density of messages and the average distance that they will travel, and then select the appropriate algorithm.

Most of the routing algorithms are straightforward and will not be repeated here. The reader is referred to (Herbordt and Weems 1990a; Herbordt and Weems 1990b; Herbordt, Weems et al. 1990) for the details. However, one algorithm is particularly novel, as it uses the Coterie Network to route data in a manner similar to the MIMD wormhole routing technique (Dally and Seitz 1987) on the SIMD CAAPP.

Each message has a header that knows its destination. Assuming that there is no blocking, a message header enters the network along a row, with message packets following it like a train of railroad cars behind an engine. When the header reaches the column of its destination address, it switches direction and the packets follow it along the column. The message is then consumed by the destination processor.

If the path of the header is blocked at any point by another message, the header must stop and wait for a clear path. In addition, the trailing packets must also wait. The problem with this approach in a normal SIMD system is that the header must notify each processor containing a trailing packet. Such notification takes as many steps as there are packets in the longest blocked message. When the path clears, the notification must be repeated so that the train can start to move again.

However, with the Coterie Network we can dynamically form a bus that maps onto a train of packets. Each train is connected to an independent bus with the header designated the bus master. When the header is blocked it merely sends a one-bit message out on the bus, and every trailing packet is notified in a single cycle. When the blockage clears, one more cycle is required to tell the trailing packets to advance.

4.2 Apply Compiler for the CAAPP

We have implemented a version of the Apply language (Hamey, Webb et al. 1987) for the low-level processor of the IUA. The compiler generates C code that can be integrated with the C that is currently used to program the CAAPP. It permits us to rapidly write functions for the CAAPP that perform local window based operations on images. We have also compiled most of the Webb library, supplied with Apply, for the CAAPP and tested their performance. The details of this effort can be found in (Scudder and Weems 1990).

4.3 Enhancements to IUA Simulator and Tools

The IUA simulator includes an elegant user interface that provides extensive interaction with the processors for debugging software (Weems and Burrill 1990). However, the interface has only been available on Sun workstations because it was written with the SunViews windowing system. The simulator has now been converted to run with X-Windows, which will greatly enhance its portability. In addition, the use of X allows a user to run a simulation on a powerful server while displaying the system status on a low-cost workstation or X-terminal.

The low level of the IUA was originally programmed using a FORTH interpreter because the simplicity and extensibility of FORTH allowed us to easily add data parallel constructs and run with low interpretation overhead. While suitable for developing simple applications, FORTH is severely limiting as applications grow in size. We have thus implemented a C preprocessor that allows programming of the CAAPP in a manner similar to the C-PARIS facility on the Connection Machine. That is, programs are developed using C control structures, but CAAPP operations are executed through calls to an extensive library of subroutines.

4.4 Prototype System Assembly

All of the custom chips used in the IUA have been assembled and tested at the Hughes Research Labs. A breadboard has been built that exercises CAAPP chips and ICAP chips running together and communicating with each other. The backplane for the prototype has been assembled and tested. In addition, the first processor board, containing 256 CAAPP processors and four ICAP processors has been assembled and is being tested prior to fabrication of the remaining 15 boards that will make up the prototype. Completion of the prototype and delivery to Umass is expected by the end of Summer 1990.

4.5 Current Efforts

We are now working on a data parallel extension to C that will explicitly support an image plane data type on the CAAPP level of the IUA. This language will permit the user to define image planes of different sizes, automatically mapping them onto virtual processors. We are also looking into creating a C* compiler for the CAAPP, in order to provide portability of code with the Connection Machine and Masspar.

As recommended by the DARPA IU Benchmark Workshop participants, much of the benchmark (Weems, Riseman et al. 1990; Weems, Riseman et al. 1988) has been recoded as a set of library routines which are called by the core of the benchmark. The most complex portion of the benchmark, the graph matcher, must still be recoded to achieve greater generality. We are also starting to plan the second level benchmark, which will incorporate tracking of moving objects over a sequence of images.

The development of the second generation of the IUA has already begun (Weems, Hanson et al. 1990). The only significant changes to the architecture are that 256 CAAPP processors will be placed on each custom chip, allowing 4K processors to be placed on a single board; and the ICAP processors will be upgraded to the TMS320C30 processor. The latter change makes each ICAP processor a 32-bit, 32 MFLOP device instead of the 16-bit, 5 MIPS devices that are currently used. With the TMS320C30 we will also be able to expand the memory at each node, and also support a multi-tasking kernel.

The third generation of the IUA is already being designed. While the second generation is mainly a technology enhancement of the first generation, the third generation will be substantially different in its architecture. The low level will change from bit-serial processors to 8-bit processors, each with a much larger on-chip cache, and additional support for floating point. The communication bandwidth at the intermediate level will be greatly enhanced (Rana, Weems et al. 1988). At the high level, we finally expect systems to become large enough that a RISC-based multiprocessor will be employed. Currently the prototype is small enough that only a single processor has been incorporated at the high level. A full-scale implementation of the third generation (256K CAAPP PE's, 1K ICAP processors, 64 SPA processors) would achieve roughly a terra-op in performance on 32-bit integer arithmetic.

5. References

Adiv, G. (1985). "Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects," IEEE PAMI, Vol. 7(4), pp. 384-401.

Balasubramanyam, P. and M. Snyder. (1988). "Computation of Motion-in Depth Parameters: A First Step in Stereoscopic Motion Interpretation," Proc. of DARPA Image Understanding Workshop, Cambridge, MA, pp. 907-920.

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

Beveridge, J. R., R. Weiss and E. Riseman. (1989). "Optimization of 2-Dimensional Model Matching," Proc. of DARPA Image Understanding Workshop, Palo Alto, CA, pp. 815-830.

Beveridge, J. R., R. Weiss and E. M. Riseman. (1990). "Combinatorial Optimization Applied to Variable Scale 2D model Matching," Proc. of IEEE Intl. Conf. on Pattern Recognition, Atlantic City, NJ, pp. 18-23.

Boldt, M., R. Weiss and E. Riseman. (1989). "Token-Based Extraction of Straight Lines," IEEE-SMC.(19)6, December 1989, pp.1581-1594.

Burns, J. B. (1987). "Recognition in 2D Images of 3D Objects from Large Model Bases Using Prediction Hierarchies," Proc. of International Joint Conference on Artificial Intelligence, Milan, pp. 763-766.

Burns, J. B. and L. J. Kitchen. (1987a). "An Approach to Recognition in 2D Images of 3D Objects from Large Model Bases," Dept. of Computer and Information Science, University of Massachusetts (Amherst), TR 87-85.

Burns, J. B. and L. J. Kitchen. (1987b). "Rapid Recognition out of Large Model Bases Using Prediction Hierarchies and Machine Parallelism," Proc. of SPIE Conf. on Intelligent Robots and Computer Vision.

Burns, J. B. and L. J. Kitchen. (1988). "Rapid Object Recognition from a Large Model Base using Prediction Hierarchies," Proc. of DARPA Image Understanding Workshop, Cambridge, MA, pp. 711-719.

Burns, J. B., R. Weiss and E. M. Riseman. (1990). "View Variation of Point Set and Line Segment Features," Proc. of 1990 DARPA Image Understanding Workshop, Pittsburgh, PA.

Collins, R. T. and R. Weiss. (1989). "An efficient and Accurate Method for Computing Vanishing Points," Proc. SPIE Workshop on Image Understanding and Machine Vision, Vol. 14, Optical Society of America, Washington, DC, pp.92-94.

Collins, R. T. and R. Weiss. (1990). "Orientation Statistics for Modeling 3D Lines and Planes," Proc. of 1990 DARPA Image Understanding Workshop, Pittsburgh, PA.

Connolly, C. (1989). "Geometer: Solid Modelling and Algebraic Manipulation," Dept. of Computer and Informtion Science, University of Massachusetts (Amherst), TR In Preparation.

Connolly, C., D. Kapur, J. Mundy and R. Weiss. (1989). "Geometer: A System for Modelling and Algebraic Manipulation," Proc. of DARPA Image Understanding Workshop, Palo Alto, CA, pp. 797-804.

Dally, W. J. and C. L. Seitz. (1987). "Deadlock Free Routing in Multiprocessor Interconnection Networks," IEEE Trans. on Computers, Vol. C-36(5).

Dolan, J. and R. Weiss. (1989). "Perceptual Grouping of Curved Lines," Proc. of DARPA Image Understanding Workshop, Palo Alto, CA, pp. 1135-1145.

Draper, B. (1989). "Integrating Top-Down Control with Intermediate-Level Vision," Proc. of SPIE Conference on Applications of AI - VII, Orlando, FL.

Draper, B., J. Brolio, R. Collins, A. Hanson and E. Riseman. (1989). "The Schema System," IJCV, Vol. 2(3), pp. 209-250.

Draper, B. and E. M. Riseman. (1990). "Learning 3D Object Recognition Strategies," Dept. of Computer and Information Science, University of Massachusetts (Amherst).

Dutta, R. and M. Snyder. (1990). "Robustness of Correspondence-Based Structure from Motion," Proc. of 1990 DARPA Image Understanding Workshop, Pittsburgh, PA.

Fennema, C. L. and A. R. Hanson. (1990a). "Experiments in Autonomous Navigation," Proc. of 1990 DARPA Image Understanding Workshop, Pittsburgh, PA.

Fennema, C. and A. R. Hanson. (1990b). "Towards Autnomous Mobile Robot Navigation," IEEE SMC.

Giblin, P. and R. Weiss. (1987). "Reconstruction of Surfaces from Profiles," Proc. of First IEEE International Conference on Computer Vision, London.

Hamey, L. G. C., J. A. Webb and I. C. Wu. (1987). "Low-Level Vision on Warp and the Apply Programming Model," Dept. of Robotics Institute, Carnegie Mellon University, TR CMU-RI-TR-87.

Hanson, A. and E. Riseman. (1978). "VISIONS: A computer System for Interpreting Scenes" in Computer Vision Systems (A. Hanson and E. Riseman, Ed.), New York: Academic Press.

Hanson, A. and E. Riseman. (1987). "The VISIONS Image Understanding System" in Advances in Computer Vision (C. Brown, Ed.), Hillsdale, NJ: Erlbaum Press.

Hanson, A. and E. Riseman. (1989). "Progress in Computer Vision at the University of Massachusetts," Proc. of DARPA Image Understanding Workshop, Palo Alto, CA, pp. 49-55.

Herbordt, M. C. and C. C. Weems. (1990a). "General Routing on the Lowest Level of the Image Understanding Architecture," Proc. of 1990 DARPA Image Understanding Workshop, Pittsburgh, PA.

Herbordt, M. T. and C. C. Weems. (1990b). "Message Passing Algorithms for a SIMD Torus with Coteries," Proc. of ACM Intl. Symposium on Parallel Algorithms and Architectures, Crete.

Herbordt, M. T., C. C. Weems and D. B. Shu. (1990). "Routing on the CAAPP," Proc. of IEEE Intl. Conf. on Pattern Recognition, Atlantic City, NJ, pp. 467-471.

Hong, J., X. Tan, B. Pinette, R. Weiss and E. M. Riseman. (1990). "Image Based Navigation Using 360 Degree Views," Proc. of 1990 DARPA Image Understanding Workshop, Pittsburgh, PA.

Horn, B. K. P. (1988). "Relative Orientation," Proc. of DARPA Image Understanding Workshop, pp. 826-837.

Ikeuchi, K. (1987). "Generating an Interpretation Tree from a CAD Model for 3D-Object Recognition in Bin-Picking Tasks," IJCV, Vol. 1(2), pp. 145-165.

Kumar, R. (1989). "Determination of Camera Location and Orientation," Proc. of DARPA Image Understanding Workshop, Palo Alto, CA, pp. 870-881.

Kumar, R. and A. R. Hanson. (1990a). "Analysis of Different Robust Methods for Pose Refinement," Proc. of IEEE Workshop on Robust Methods in Computer Vision, Seattle.

Kumar, R. and A. R. Hanson. (1990b). "Pose Refinement: Application to Model Extension and Sensity to Camera Parameters," Proc. of 1990 DARPA Image Understanding Workshop, Pittsburgh, PA.

Kumar, T. and A. Hanson. (1989a). "Robust Estimation of Camera Location and Orientation from Noisy Data Having Outliers," Proc. of IEEE Workshop on Interpretation of 3D Scenes, Austin, TX, pp. 52-60.

Kumar, T. and A. Hanson. (1989b). "Robust Estimation of Camera Location and Orientation from Noisy Data with Outliers," Dept. of Computer and Information Science, University of Massachusetts (Amherst), TR 89-120.

Oliensis, J. (1989). "Existence and Uniqueness in Shape from Shading," Dept. of Computer and Information Science, University of Massachusetts (Amherst), TR 89-109.

Oliensis, J. (1990a). "Existence and Uniqueness in Shape from Shading," Proc. of IEEE Intl. Conf. on Pattern Recognition, Atlantic City, NJ, pp. 341-345.

Oliensis, J. (1990b). "New Results in Shape from Shading," Proc. of 1990 DAROA Image Understanding Workshop, Pittsburgh, PA.

Oliensis, J. (1990c). "Shape from Shading as a Partially Ill-Posed Problem," Dept. of Computer and Information Science, University of Massachusetts (Amherst), TR 90-50.

Rana, D. and C. C. Weems. (1990). "A Feedback Concentrator for the Image Understanding Architecture," Proc. of IEEE Intl. Conf. on Pattern Recognition, Atlantic City, NJ, pp. 540-544.

Rana, D., C. C. Weems and S. P. Levitan. (1988). "An Easily Reconfigurable Circuit Switched Connection Network," Proc. of IEEE Intl. Symposium on Circuits and Systems, pp. 247-250.

Riseman, E. M. and A. R. Hanson. (1984). "A Methodology for the Development of General Knowledge-Based Vision Systems," Proc. of IEEE Workshop on Computer Vision: Representation and Control, pp. 159-170.

Sawhney, H. and A. R. Hanson. (1990). "Comparative Results of Some Motion Algorithms on Real Image Sequences," Proc. of 1990 DARPA Image Understanding Workshop, Pittsburgh, PA.

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

Sawhney, H. and J. Oliensis. (1990). "Image Description and 3D Interpretation from Image Trajectories Under Rotational Motion," Dept. of Computer and Information Science, University of Massachusetts (Amherst).

Scott, G. L. (1986). "Smoothing the Optic Flow under Perspective Projection," Proc. of CVPR, pp. 504-509.

Scudder, M. and C. C. Weems. (1990). "An Apply Compiler for the CAAPP," Dept. of Computer and Information Science, University of Massachusetts (Amherst).

Snyder, M. (1990). "On the Calculation of Rigid Motion Parameters from the Essential Matrix," Dept. of Computer and Information Science, University of Massachusetts.

Waxman, A. and J. Duncan. (1986). "Binocular Image Flow: Steps Towards Stereo-Motion Fusion," IEEE PAMI, Vol. 8, pp. 715-729.

Weems, C. C. and J. R. Burrill. (1990). "The Image Understanding Architecture and its Programming Environment" in Parallel Architectures and Algorithms for Image Understanding (V. K. Prassana-Kumar, Ed.), Orlando, FL: Academic Press.

Weems, C. C., A. R. Hanson, E. M. Riseman, D. Rana, D. B. Shu and J. G. Nash. (1990). "An Overview of Architecture Research for Image Understanding at the University of Massachusetts," Proc. of IEEE Intl. Conf. on Pattern Recognition, Atlantic City, NJ, pp. 379-384.

Weems, C., S. Levitan, A. Hanson, E. Riseman, J. Nash and D. Shu. (1989). "The Image Understanding Architecture," IJCV, Vol. 2(3), pp. 251-282.

Weems, C. C. and D. Rana. (1990). "Reconfiguration in the Low and Intermediate Levels of the Image Understanding Architecture" in Reconfigurable SIMD Parallel Processors (H. Li, Ed.), Englewood Cliffs, NJ: Prentice-Hall. Also COINS Technical Report TR 90-10, University of Massachusetts (Amherst).

Weems, C. C., E. M. Riseman, A. R. Hanson and A. Rosenfeld. (1990). "An Integrated Image Understanding Benchmark for Parallel Processors," J. Parallel and Distributed Computing.

Weems, C., R. Riseman, A. Hanson and A. Rosenfeld. (1988). "An Integrated Image Understanding Benchmark: Recognition of a 2 1/2 D "Mobile"," Proc. of IEEE Conf. on Computer Vision, Ann Arbor, MI.

Williams, L. R. (1990). "Perceptual Organization of Occluding Contours," Proc. of 1990 DARPA Image Understanding Workshop, Pittsburgh, PA.

Williams, L. R. and A. Hanson. (1988a). "Depth From Looming Structureq," Proc. of DARPA Image Understanding Workshop, Cambridge, MA, pp. 1047-1051.

Williams, L. R. and A. Hanson. (1988b). "Translating Optical Flow into Token Matches," Proc. of DARPA Image Understanding Workshop, Cambridge, MA, pp. 970-980.

Williams, L. R. and A. R. Hanson. (1988c). "Translating Optical Flow into Token Matches and Depth from Looming," Proc. of 2nd International Conference on Computer Vision, Tarpon Springs, FL, pp. 441-448.