Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection

WILLIAM HOFF, MEMBER, IEEE, AND NARENDRA AHUJA, SENIOR MEMBER, IEEE

Abstract—**The goal of stereo algorithms is to determine the three-dimensional distance, or depth, of objects from a stereo pair of images. The usual approach is to first identify corresponding features between the two images and estimate their depths, then interpolate to obtain a complete distance or depth map. Traditionally, finding the corresponding features has been considered to be the most difficult problem. Also, occluding and ridge contours (depth and orientation discontinuities) have not been explicitly detected which has made surface interpolation difficult. The approach described in this paper integrates the processes of feature matching, contour detection, and surface interpolation. Integration is necessary to ensure that the detected surfaces are smooth. Surface interpolation takes into account detected occluding and ridge contours in the scene; interpolation is performed within regions enclosed by these contours. Planar and quadratic patches are used as local models of the surface. Occluded regions in the image are identified, and are not used for matching and interpolation. A coarse-to-fine algorithm is presented that generates a multiresolution hierarchy of surface maps, one at each level of resolution. Experimental results are given for a variety of stereo images.**

Index Terms—**Boundary detection, feature matching, integration, stereo vision, surface interpolation, three-dimensional segmentation, three-dimensional vision.**

I. INTRODUCTION

THE goal of stereo vision is the recovery of three-dimensional depth information from images taken from different viewpoints. In this paper we are concerned with the problem of computing the distance between the viewer and each point of the scene visible from two viewpoints, using two images recorded simultaneously from a pair of laterally displaced cameras. The usual paradigm for stereo algorithms includes the following steps:

1) Features are located in each of the two images independently.

2) Features from one image are matched with features from the other image. That is, for every feature in the left image corresponding to a certain point in the scene, a feature must be found in the right image such that it corresponds to the projection of the same scene point.

3) The disparity between features is used, together with the parameters of the imaging geometry (i.e., relative separation and orientation of the cameras), to determine the distance to the corresponding point in the scene.

4) The resulting depth points are often sparse whereas depth must be computed at every point in the scene. Thus, the depth points are interpolated to obtain a surface, or a complete depth map.

Although monocular cues expedite fusion [1], it may be sufficient to consider only syntactic or low level features of local gray level patterns to establish point correspondences across images [2]. Because of their simplicity, similar low level features occur commonly in the image. The search for the match of a point often yields multiple candidates, because there is little information that can be used to characterize low level features uniquely. Thus, there can be many possible matches for a given feature, and it is necessary to choose the correct match from among all the candidates. The matching step above incorporates a resolution of this ambiguity. Since the selected matches are crucial in determining the resulting surface map, this step, called the correspondence problem, has been considered to be the central and the most difficult part of the stereo problem.[1]

In addition to the problem of ambiguity, finding correspondences is difficult because some features may have no correct match due to image noise or occlusion. Also, the feature points may be sparse in places where there is little image variation. Finally, the disparity values may be noisy, which can be caused by uncertainty in the positions of the feature points.

This paper argues that the steps of matching and surface interpolation should be merged. The goal is to perform matching such that the interpolated surface is smooth except across ridge and occluding contours. First we review previous work and discuss the need for integration. Next we present a method for integrating surface interpolation and matching, and then a method for detecting ridge and occluding contours. A coarse-to-fine algorithm is outlined that implements the integration to produce a hierarchy of surface maps, corresponding to different levels of resolution. Results of the algorithm are presented on a variety of images. An early report on the work presented in this paper can be found in [15], [14]. See also [17], [20], [30] for other related work on integration.

Manuscript received February 3, 1987; revised June 1, 1988. This work was supported by the National Science Foundation under Grant ECS 8352408 and by Rockwell International.

W. Hoff was with the Coordinated Science Laboratory, University of Illinois, 1101 West Springfield Ave., Urbana, IL 61801. He is now with Martin Marietta Astronautics Group, P.O. Box 179, Denver, CO 80201.

N. Ahuja is with the Coordinated Science Laboratory, University of Illinois, 1101 West Springfield Ave., Urbana, IL 61801.

IEEE Log Number 8824799.

[1]The use of more than two views reduces but does not eliminate ambiguities. For this reason we will concentrate on the two view situation, although in principle the approach described here could be applied to more than two views.

II. The Need for Integration

In order to solve the correspondence problem, constraints must be used to select the correct match of a feature from among the candidate matches. Such constraints are often (but not always, e.g., [29]) expressed in terms of the relative disparities of nearby features. Thus, these constraints enforce specific models of spatial variation of depth, or surface smoothness. A realistic model of smoothness is suggested by the following observations. Objects in a scene have faces which are smooth in the sense that the surface normal varies slowly. The faces meet at surface ridges which are themselves smooth curves in three-dimensional space. This smoothness of the faces and ridges implies that the three dimensional boundary of any object against the background is also (piecewise) smooth. Thus, the image of a scene has the following structure. It contains regions corresponding to object faces. The border of a region may be composed of two kinds of segments: ridge segments, corresponding to surface ridges across which the surface slope is discontinuous, and occlusion segments across which surface depth is discontinuous. In the interiors of these regions both surface depth and slope vary smoothly in some predetermined sense (in our implementation smoothness means planar or quadratic variation in disparity; see Section III-A).

A. Constraints Used in Previous Work

Two types of constraints, and thus models of surface smoothness, have been used in most previous work on stereo. One popular constraint enforces the disparities of features in a window to have similar values. This constraint, which we call the *constant-local-disparity* constraint, has been used by a number of workers. In the Marr–Poggio–Grimson approach [3]–[5], the matching ambiguity at a point is resolved such that the chosen disparity value is close to the majority of the unambiguous points in the neighborhood. In the algorithm of Barnard and Thompson [6], a relaxation labeling technique [7] is used to resolve ambiguities, using the constraint that nearby points should have nearly the same disparity. In the algorithms of Medioni and Nevatia [8] and Ayache and Faverjon [9], line segments are extracted and matched. Again, similarity of disparity is used to resolve ambiguous matches.

Another constraint, called the *figural continuity* constraint, restricts disparity values along edges in the image. Mayhew and Frisby [10] use this constraint to resolve matching ambiguities along edge segments. Grimson [11] uses the figural continuity constraint along edges at a coarse resolution to enforce the same at finer resolution. Baker and Binford [12] use the figural continuity constraint along with an added restriction: if the edges along a certain epipolar line in one image are scanned left-to-right, their matches along the corresponding epipolar line in the other image should also have a left-to-right ordering. Ohta and Kanade [13] also use the same constraint.

There are two problems with each of these constraints. First, they make certain assumptions about the relationship of the detected image features to three-dimensional features, which may or may not hold. For example, an intensity edge segment may not lie on a single surface, but may cross an occlusion boundary. This happens when there are no strong intensity edges defining the boundaries of different surfaces. In this case, disparity will not vary smoothly along the edge segment and the figural continuity constraint will fail. Likewise, enforcing constancy of disparity over a window will be erroneous if the window contains an occlusion boundary.

The second problem with the use of these constraints is that even when the above assumptions about the features are met, namely, the edge segments or windows to which the constraints are applied come from a single surface, the constraints do not enforce surface smoothness in a true sense. For example, the constant-local-disparity constraint is too weak to truly enforce surface smoothness, because smoothness is actually determined by not only the values of disparity in an image region but also the spatial distribution of these values. The disparity values may actually span a wide range without violating surface smoothness constraint as would be the case for a slanted plane in which the disparities of features in the nearest part are larger than those of features located in the distant part. If the model requires that nearby disparities have similar values, i.e., that the histogram of these local disparities be uniform, then disparity selection is biased in favor of a frontoparallel surface.[2]

The figural continuity constraint correctly but weakly enforces the constraint of three-dimensional surface smoothness, because it enforces smoothness only along curves. Three-dimensional surface smoothness actually implies a stronger, two-dimensional smoothness in the image.

B. The Need for Integration

We will now argue that to enforce surface smoothness it is necessary to integrate the processes of matching and surface interpolation. Matching provides surface depth values at the locations of the matched features, which constrain the surface that will result after interpolation. Effectively, the matching process determines the final surface derived. Since the resulting surface is expected to be smooth, the correctness of the choice of matches is judged by the type of surfaces it produces. Therefore, the interpolation process should be involved in matching so as to make acceptable matching decisions. The matching and interpolation processes thus should be integrated.

[2]A pilot study was conducted to examine possible roles in human stereopsis of the surface-smoothness constraint and the constant-local-disparity constraint [14], [15]. A random dot stereogram was generated such that the use of the two different matching constraints would yield the perception of two different surfaces. The random dot stereogram portrays a surface whose height varies along the vertical axis as a cosine wave, and which is unambiguous everywhere except for a small region centered at the peak. This ambiguous region can be perceived as a smooth continuation of the cosine wave, or as a surface which is locally rough but has approximately constant height. We found that observers perceived the smooth surface much more readily than the rough surface. These results appear to support the hypothesis that the surface smoothness constraint is used.

Further, we argue that it is necessary to integrate contour detection with matching and surface interpolation. The presence of surface contours violates the assumption of local surface smoothness. It is important that contours be detected to avoid matching errors and to correctly interpolate the surface. In our approach, this was done by comparing adjacent surface patches for depth and orientation discontinuities.

III. An Integration Algorithm

In this section, we describe an algorithm that implements integration of matching, interpolation, and contour detection. We first present an overview of the algorithm and then briefly describe the major steps in the algorithm. The details of the implementation are given in the Appendix.

A. Overview of the Algorithm

The algorithm is shown schematically in Fig. 1. To reduce the number of ambiguous matches that must be explored, the algorithm works in a coarse-to-fine mode, and obtains depth maps at multiple resolutions. The algorithm starts with an initial coarse estimate of the surface map, e.g., a flat frontal surface at some depth.[3] A given coarse level surface predicts the locations of edge matches at the next finer level. Thus, only a relatively small window centered at the predicted location need be searched for each match. This multiresolution representation has been used by previous workers [18], [3]. It is used in our approach for the sake of efficiency, and is not central to the method.

To obtain edge images at different resolutions, the Laplacian-of-Gaussian ($\nabla^2 G$) operator of different sizes is used [19]. The finest resolution edges are obtained by locating the zero crossings of a small $\nabla^2 G$ operator convolved with the original stereo images. Coarser edge images are obtained by successively doubling the size of the operator and halving the size of the convolved image before detecting zero crossings.

At each resolution level, the following steps are performed. First, the algorithm matches individual edge points in one image with the corresponding points in the other image. Matching is driven from left to right, and from right to left, in two separate but identical processes. The result is that there are two sets of feature points, one for the left image, and one for the right image. Each feature point P_i in, say, the left image has a set of possible matches $M(P_i)$ in the right image, suggested by similarity of feature properties. Each candidate match in $M(P_i)$ determines a distinct point in three dimensional space. At most one of these candidate matches must be chosen from each set $M(P_i)$. The surface smoothness constraint is used

[3]The initial estimate of disparity is not critical, because the size of the matching window (15 pixels) is an appreciable fraction of the size of the image at the coarsest resolution (64 × 64). Also, as mentioned in the text, the matching window can be made larger, but at the cost of increased computation time.

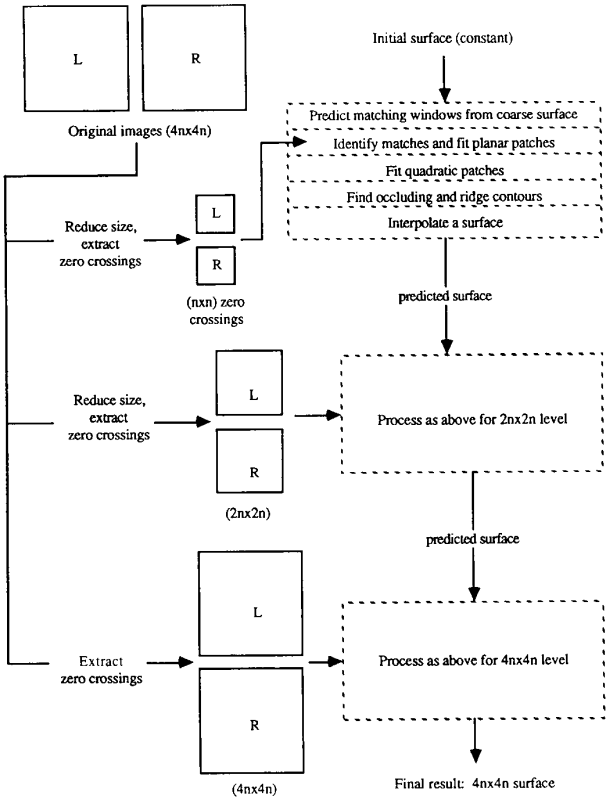


Fig. 1. Control flow of algorithm integrating matching, interpolation and contour detection.

to select a subset of matches, such that the corresponding 3-D points lie on a smooth surface, i.e., a surface whose surface normal varies slowly.

We have used specific parameterized functions as local models of the disparity surface, i.e., planar and quadratic patches. These surfaces are smooth because they are of low order. (Planar surfaces are also used by Eastman and Waxman [20].) The depth[4] of a point located at (x, y) according to the planar and quadratic models, $z_p(x, y)$ and $z_q(x, y)$, respectively, is given by

$$z_p(x, y) = a_1 x + b_1 y + c_1 \tag{1}$$

and

$$z_q(x, y) = a_2 x^2 + b_2 y^2 + c_2 xy + d_2 x + e_2 y + f_2. \tag{2}$$

[4]We actually fit surfaces in disparity space, rather than in depth. This is valid because a planar surface in depth is also planar in disparity space. To see this, consider a stereo pair of cameras located at $(X, Y, Z) = (+D, 0, 0)$ and $(-D, 0, 0)$ with their optical axes along the $+Z$-axis. The x and y coordinates of the projection in the left image of a scene point (X, Y, Z) are given by: $X = Zx/f - D$, $Y = Zy/f$, where f is the focal length of the camera. Now consider a plane whose depth Z is given by $Z = AX + BY + C$; substituting for X and Y terms of x and y gives $Z = A(Zx/f - D) + BZy/f + C$., i.e., $1/Z = (1 - Ax/f - By/f)/(C - AD)$. Since the stereo disparity is related to Z by the expression $2Df/Z$, the variation of disparity as a function of x and y is given by $2Df(1 - Ax/f - By/f)/(C - AD)$, which is planar in x and y.

These models can approximate any continuous surface, if the regions or patches of approximation are small enough. In practice, however, these regions must be sufficiently large to contain enough data points so that a reliable fit can be estimated. Thus, the surface reconstruction may have large errors for surfaces that are of higher order than quadratic. The amount of the error depends on the magnitude of the higher order coefficients, and the size of the patches used.

Planar patches are fit in circular image regions to the depth points centered at each (x, y) position on a regular grid in the image. A sparse grid is used to reduce the amount of computation. Up to two planes are found at each grid point that have the highest fit-rating, and pass the adequacy tests described in Section III-B. The planar patches represent a rough approximation to the surface. More importantly, they determine which combinations of matches are mutually consistent. Quadratic patches are then fit at each grid point, to the combinations of matches found by the previous step, using a standard least squares technique. The quadratic patches are fit to the points within the planar patches centered at the current grid point and neighboring grid points. The quadratic surface containing the most points is kept as the fit for that grid point.[5] All matches are now unambiguous, because those which contribute to a quadratic patch are taken to be the correct matches.

Next, depth and orientation contours are found by fitting bipartite planar patches, and detecting discontinuities between the two halves. The contours found from the left image are then combined from the contours from the right image. Finally, a smooth surface is interpolated away from contours to yield a piecewise smooth surface map at the given resolution. The process is then repeated at finer resolution using the current surface to predict matching locations of edges at the finer resolution. Processing at successively finer resolutions yields surfaces at increasingly fine resolution.

In summary, the processing steps for each resolution level are initial matching, fitting planar patches, fitting quadratic patches, detecting contours, and surface interpolation. We describe briefly the implementation of these steps below; see the Appendix or [14] for details.

B. Planar Fitting

The number of possible subsets of matches for the feature points contained in a local region R is

$$N_1 = \prod_{P_i \in R} \left(\text{card}\big(M(P_i)\big) + 1\right) \tag{3}$$

where $\text{card}(M(P_i))$ is the number of possible matches for point P_i and the $+1$ is for the possibility of no match. One way of selecting the best subset of matches is to test each combination of matches to see if they lie on a smooth surface. This exhaustive method is impractical if there are many ambiguous points and N_1 is very large. However, an alternative approach is practical: test many different surfaces to find the one that is most consistent with a subset of the possible matches.

This approach was taken, using planar surfaces. From (1), a plane is characterized by three parameters (a, b, c). We discretize the possible values of these parameters and evaluate each possible planar fit, using the Hough transform [21]. The plane that receives the highest fit-rating is chosen as the best estimate of the surface (actually, up to two planes are selected, and the best one is selected at a later step). Additional details are given in the Appendix. A similar Hough technique has been proposed by Clark [22], but he uses all the points on a given contour to find the best estimate of the planar parameters, whereas we use all points within a local area.

To obtain the Hough transform, a constant amount of work is required for each possible match, where the constant is proportional to the size of the Hough space. The total number of candidate matches is

$$N_2 = \sum_{P_i \in R} \text{card}\big(M(P_i)\big). \tag{4}$$

If a small parameter space is used, the total amount of work using the Hough transform is much less than if the exhaustive method discussed in the first paragraph were used.

To evaluate a particular planar fit, the matches contributing to the plane must be identified. If a point has more than one possible match, only the match closest to the planar surface may contribute to the score. In addition, no match may contribute to the score if it is beyond a certain distance to the plane, called the *outlier distance*. This takes into account the possibility of incorrect matches that are not on the surface. These points should not be allowed to affect the surface fit. The fit-rating for a particular plane (a, b, c) is given by

$$\text{fit-rating} = \sum_{z_i \in S} \left[D^2 - \big(z_i - z_p(x_i, y_i)\big)^2\right] \tag{5}$$

where D is the outlier distance, S is the set of matches contributing to the plane, z_i is the disparity of a match located at image point (x_i, y_i), and $z_p(x_i, y_i)$ is the equation of the plane evaluated at point (x_i, y_i).

The outlier distance may be derived from an *a priori* estimate of the noise in the depth values.[6] We assume that the major component of this noise is due to the uncertainty in position of the feature points. The fluctuation of the zero crossings about the true edge position is due to image noise and the blurring effect of the edge operator. We assume that the displacement error is normally distributed, with standard deviation σ_N. From the properties of the normal distribution, 95 percent of the time the error will be less than $2\sigma_N$. We assume that $2\sigma_N = \sigma_G$, where σ_G is the standard deviation of the Gaussian of the edge oper-

[5]The number of points is used as a criterion because reliability and accuracy increase with the number of points. In fact, it was found that discarding quadratic surfaces with less than 30 points removes most of the incorrect patches, caused by a local excess of spurious matches.

[6]Clark [22] calculates the displacement of the zero crossings exactly, but requires a complete scale-space map of the left and right images.

ator. This is consistent with the displacement of zero crossings observed in experiments we performed with white noise stereograms. The distance $2\sigma_N$ is used as the outlier distance, so that points with a displacement error greater than $2\sigma_N$ are assumed to be due to incorrect matches. Incorrect matches are ignored for the purpose of calculating the surface parameters.

The planar surface model breaks down for regions that are discontinuous. In the vicinity of the discontinuity, a sizable planar or quadratic fit will not be a good fit to the depth points. To detect this situation, and also the situation where the disparity of the surface is out of range of the matcher, two *adequacy* tests are used: 1) the squared error of the points to the fitted surface, and 2) the number of points in the region which are unmatchable.

The first measure depends on the *a priori* estimate of the noise in the depth values, σ_N. For a given number of poitns in a region, within the outlier distance to the surface, the probability of the sum of squared errors being less than or equal to ϵ^2 may be determined from the χ^2 distribution. The algorithm determines the maximum expected squared error for a 95 percent confidence level. If the squared error exceeds this value, the surface is rejected.

The second measure depends on an estimate of the probability of a point to be unmatchable. The surface patch should be rejected if there are too many unmatchable points. If the points are independent, then the number of unmatchable points follows the binomial distribution. If the number of unmatchable points in a region exceeds the value for a 95 percent confidence level, the surface patch is rejected.[7] The probability of a point to be unmatchable is partly a function of the disparity gradient of the surface due to perspective compression of the matching window. The acceptable fraction of unmatchable points was determined empirically by projecting a plane with a synthetic random dot texture onto left and right image planes, and measuring the fraction of unmatchable points.

C. Quadratic Fitting

The planar patches fit in the previous step represent a rough approximation to the surface. Further, some of the patches are ambiguous, with two possible planes assigned to a patch. To improve the accuracy of the surface reconstruction, and to resolve any ambiguities among the planar patches, quadratic patches are now fit. The quadratic patches may be fit over a larger region of the image than the planar patches, because they are of higher order and can better follow the surface curvature. The main purpose of the plane-fitting step is to determine which combinations of matches are mutually consistent, so that a quadratic surface may be fit to only those combinations.

A quadratic patch is fit at image location (x, y) to the points within the planar patches centered at (x, y) and neighboring locations, using a standard least-squares technique. If a planar patch is ambiguous, it contributes two different sets of matches to the quadratic patch. If there are n ambiguous planar patches, there are 2^n combinations of matches. Since we want the best fitting quadratic surface, we limit the number of combinations to 2, by taking the sets of planes that are most compatible, in the sense that their depths and orientations are similar. The quadratic surface containing the most points is kept as the fit for that location.[8] All matches are now unambiguous, because those which contribute to a quadratic patch are taken to be the correct matches. Details are in the Appendix.

The quadratic patches are also used to create the global surface, rather than interpolating directly from the (now unambiguous) depth points. The depth at each point of the final surface is computed as a weighted combination of the depths of the nearby patches.

D. Contour Detection

At occluding and ridge contours, the assumption of local surface smoothness is violated. One way to detect contours is to find places in the image where smooth surface patches cannot be fit, or where there is a large error in the fit. However, this method is not reliable because the patches may be missing or defective for other reasons, such as image noise, or lack of feature points. Also, patches may straddle contours, if the data points in the vicinity are sparse.

A more reliable method to detect discontinuities is to look for two adjacent surface patches that differ in depth or orientation. We fit bipartite circular planar patches, which are circular patches divided into two halves by a diameter of a given orientation. A plane is fit independently to the depth points within each semicircular half of the bipartite patch. If the two planes differ in depth (or orientation) by more than a threshold, then there is evidence for an occluding (or ridge) edge in the vicinity of the grid point, and in the direction of the diameter used to obtain the two semicircles. Details are given in the Appendix.

In our implementation, discontinuity detection was performed after matching and surface patch fitting, so that the matches had already been disambiguated. However, it can be performed concurrently with these processes, by using the methods for planar patch fitting described in Section III-B. In either case, surface patches that contain a detected discontinuity should be inhibited, because they

[7]The probabilities of the various points to be unmatchable are *not* independent, because the zero crossings are not scattered randomly over the image, but lie on continuous contours. Short segments of zero crossing contours may be unmatchable, and so points which are near an unmatchable point and on the same contour are more likely to be unmatchable than points which are not. However, the assumption becomes less erroneous if the regions over which statistics are taken are large. We will assume that the probabilities are independent, for the purpose of analysis here.

[8]The number of points is used as a criterion because reliability and accuracy increase with the number of points. In fact, it was found that discarding quadratic surfaces with less than 30 points removes most of the incorrect patches, caused by a local excess of spurious matches.

are adversely affected by the depth points on the other side of the contour, resulting in an incorrect estimate.

The result is a set of candidate edge points, lying near the centers of the bipartite planar patches where they were detected. These edge points are detected and localized based on local evidence. To accept an edge point we further require that it falls along a smooth (ridge or occluding) contour. This enforces the property that objects as well as their faces have smooth borders. This constraint is particularly useful since the edge points are often sparsely located and the local evidence may place the contours only approximately, somewhere in the interfeature area. The requirement of contour smoothness propagates the placement constraints on the edges posed by matched features occurring in different parts of the contour, thus possibly resolving correctly the local uncertainty.

In the implementation, cubic splines are fit to the locally detected edge points to obtain a smooth contour. Cubic splines are continuous up to the second derivative, and are not required to pass through the given points [23]. A more sophisticated method would integrate contour smoothness with contour detection. For example, a cost function could be defined such that one part consisted of the scores for the local placement of the edge point, and another part would be proportional to the local curvature of the contour, i.e., the curvature of the curve connecting the edge point and its neighbors along the contour. The contour would be placed such that it gives a minimum value for the cost function, i.e., it optimizes a combination of local placement scores and local smoothness.

An occluding contour is easier to detect and locate from the viewpoint where it is not adjacent to an occluded region, i.e., a region which is not visible to the other viewpoint and thus is unmatchable. If points within an occluded region are matched, they match at random, and occasionally the matches define small, false surface patches. Whether an occluded region lies next to an object boundary depends upon the viewpoint. In the left image, there is a region to the left of the left object boundary which is occluded from the right viewpoint. In the right image there is an occluded region to the right of the right object boundary. Therefore, in two identical and almost completely separate processes, we construct two surface maps: one based on the coordinate system of the left viewpoint, and the other based on the coordinate system of the right viewpoint. Each process detects only those segments of occluding contours which have an orientation such that there is no occluded region in that viewpoint. Matching is driven from left to right in one process, and from right to left in the other process. The result is that there are two sets of feature points, one for the left image, and one for the right image. Each feature point is labeled with one or more disparity values. The contours detected in the other viewpoint are then combined with the contours detected in the current viewpoint, to give a complete set of contours. The surface map of either viewpoint can be used to display the final result; in this work, the left viewpoint was used.

IV. Results

Results are presented for running the algorithm on a set of synthetic images and a set of real images. For each example, the results consist of a hierarchy of disparity maps, one at each level of resolution. "Occluded" and "unknown" regions of the surface are marked, as well as occluding and ridge contours. The results are evaluated on the basis of how close the disparity maps are to the true disparity maps, and how accurately the occluding and ridge contours are placed.[9]

For each stereo pair, the size of the finest level of resolution was specified, along with a constant estimate of disparity to be used for the coarsest level. The finest levels were either 256×256 or 512×512, and the coarsest level was always 64×64. We manually measured the range of disparities of the stereo pair and chose the midpoint to obtain the constant estimate.

Image regions which have no significant intensity texture generate very weak edges. The zero crossings detected for such regions are due to noise for the most part, and are uncorrelated. A threshold on the slope of the $\nabla^2 G$ convolution values across the zero crossings was used to eliminate weak edges. With no threshold, isolated patches were occasionally fit in these regions. These were incorrect, and appeared to be due to local groups of zero crossings that happened to lie on a smooth surface, by chance. The same threshold was used for most of the examples in this section. A higher or lower threshold was used for a few images because the gray level range for those images was different.

A. Synthetic Images

The synthetic images were generated by a program, which wraps image textures onto three-dimensional surface patches, and displays the result in a perspective view. The synthetic images have the advantage that the true disparity map is known, and the results can be quantitatively compared to the output of the stereo algorithm at every point.

The results on synthetic images are shown in Figs. 2–4. The original stereo pairs are shown in part (a) of each figure. Fig. 2(a) shows a concrete sphere on a table, Fig. 3(a) is a cube with a random dot texture, and Fig. 4(a) shows a cone and a cube with a reptile skin texture and a brick texture [24], respectively. The sizes of the original stereo pairs and the disparity range are given in Table I. The viewing directions are parallel in each example.

The contours and quadratic patches at the finest level of resolution are shown in (b) of each figure. The patches are centered at the locations of the star-like objects. In Fig. 2(b), the contours around the sphere are all occluding contours, and the thick band along the left side of the sphere represents the area of the left image that is occluded, i.e., not visible in the right image. In Fig. 3(b),

[9]The results are evaluated on the basis of disparity rather than distance, because converting disparity to distance would introduce calibration errors, and we wish to concentrate on the performance of the algorithm.

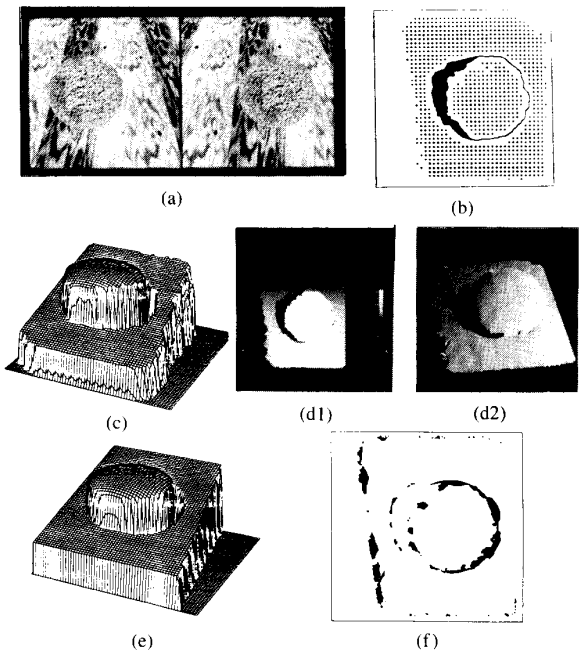


Fig. 2. (a) 256 × 256 synthetic image of a sphere on a table. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d1) Surface as intensity image. (d2) Shaded perspective view. (e) Ideal disparity surface. (f) Points with error magnitude >1.

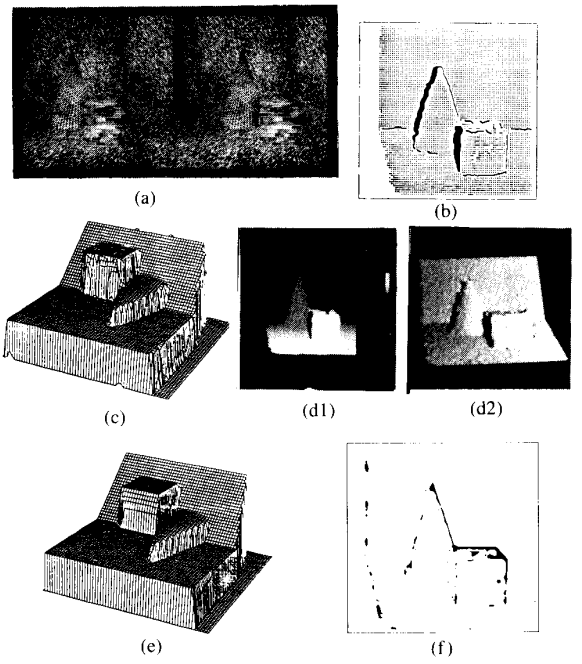


Fig. 4. (a) 512 × 512 synthetic image of a cone and a cube. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d1) Surface as intensity image. (d2) Shaded perspective view. (e) Ideal disparity surface. (f) Points with error magnitude >1.

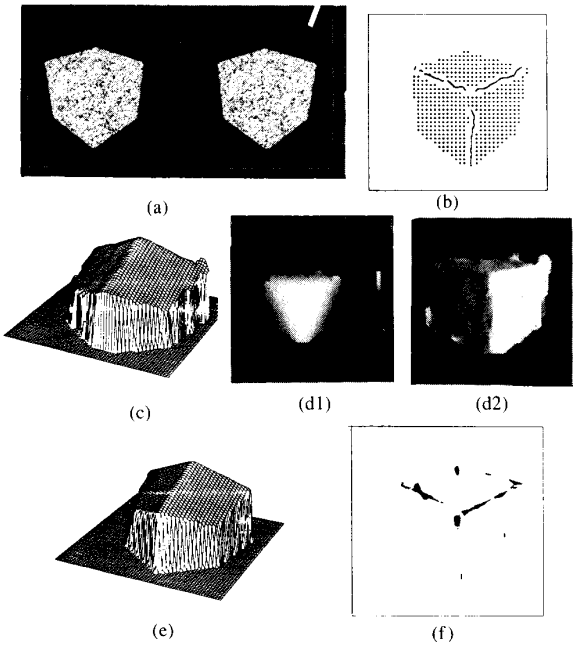


Fig. 3. (a) 256 × 256 synthetic image of a cube. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d1) Surface as intensity image. (d2) Shaded perspective view. (e) Ideal disparity surface. (f) Points with error magnitude >0.5.

TABLE I
SYNTHETIC EXAMPLES

Figure	Example	Size	Disparity Range
2	Sphere	256 × 256	45
3	Cube	256 × 256	15
4	Cone	512 × 512	40

the contours along the edges of the cube are all ridge contours. Fig. 4(b) contains ridge contours along the top edge of the cube and between the wall and the table. The other contours are all occluding contours.

The disparity surface is shown in (c) of each figure. Areas of the surface which are "unknown" are assigned the lowest height for display purposes—no patches could be fit to these areas, nor were there known patches in the vicinity to interpolate from. The surface appears to have a "hole" where depth values are unknown. "Occluded" areas are displayed as a height slightly above "unknown." The viewpoint for displaying the reconstructed surface is 25° above the horizontal plane, and ±25° from the axis running vertically through the image.

One can see a large unknown area on the top face of the cube in Fig. 4(c), and Fig. 4(b) shows that not many patches were fit in this area. This is because the brick texture in this area yields intensity edges which are predominantly horizontal. The algorithm does not attempt to match zero crossings which are near horizontal, because

the disparity of horizontal zero crossings is subject to large error. Thus, there are not enough points in the area to fit patches, and so the surface is unknown there.

Another way to display the resulting surface is to encode the height by an intensity value proportional to surface height. Yet another way is to show the shaded image of the surface from a given viewpoint and a given light source position. Part (d) of each figure shows the resulting surface as an intensity image, or shaded, or both. In these displays, black indicates "unknown" areas.

The ideal surface is shown in (e) of each figure. Points which are different from the ideal surface by more than one pixel of disparity are shown in Figs. 2(f) and 4(f). Most of these large errors are due to a small misplacement of the occluding boundary. Fig. 3(f) shows the points which are different from the ideal surface by more than 0.5 pixel of disparity. Since there are no occluding contours in this scene, there were no large errors.

B. Real Images

The results on real images are shown in Figs. 5–13. Most of the real stereo images were taken using a single camera at two different positions. The positions and orientations of the cameras were approximately, but not precisely measured. Several example images were obtained from other laboratories. Some of the images were not vertically registered, because the view directions were not exactly parallel, causing the epipolar lines to be nonhorizontal. Although it is not difficult in principle to calculate the positions and orientations of the epipolar lines from the camera parameters and imaging geometry, we found it easier to correct the images manually so that they were vertically registered. This was done by compressing or stretching one of the images in the vertical direction until it was aligned with the other. This procedure was just a first approximation to the task of correctly registering the two images, and was done so that the program could be run on examples that it otherwise would not be able to handle. The procedure was not intended to completely correct for all nonideal camera optics and viewing situations.

The baseball image (Fig. 5), the circuit board image (Fig. 9), and the books image (Fig. 10) were digitized directly from a TV camera with a 25 mm lens. The ruts image (Fig. 6), rocks image (Fig. 7), sandwich image (Fig. 8), and fruit image (Fig. 11) were taken with a 35 mm camera using a 50 mm lens, and were digitized from the negatives. The Pentagon image in (Fig. 12) was obtained from Prof. Takeo Kanade of Carnegie-Mellon University. The Renault image (Fig. 13) was obtained from Prof. Gerard Medioni of the USC Institute for Robotics and Intelligent Systems.

The original stereo pair is shown in (a) of each figure. Table II gives the size of the images, the disparity range in pixels, and approximate measurements for camera separation and subject distance. Also shown for each example are the contours and patches for the finest resolution level (b), and the surface for the finest resolution level

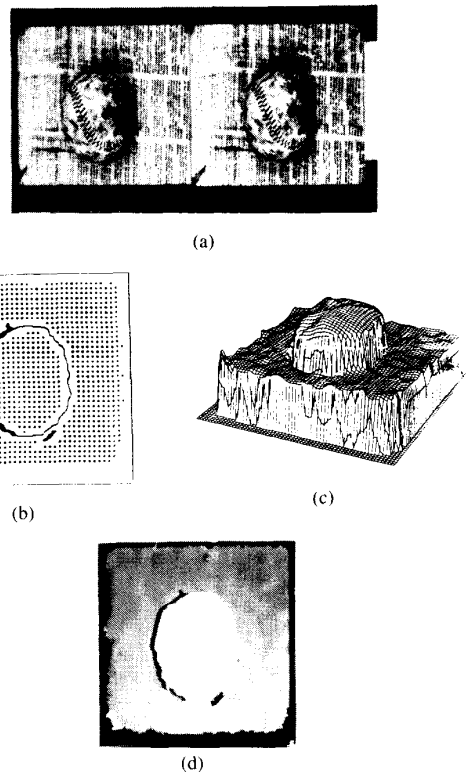


Fig. 5. (a) 256 × 256 real image of a baseball. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d) Surface as intensity image.

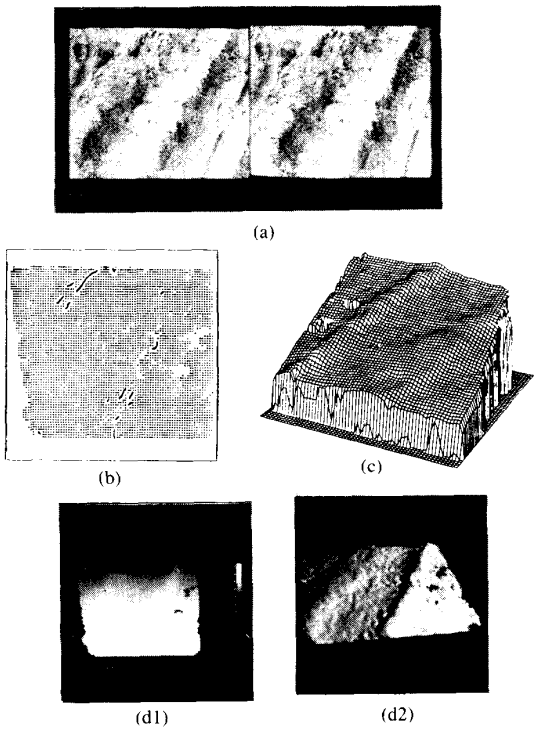


Fig. 6. (a) 512 × 512 real image of ruts. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d1) Surface as intensity image. (d2) Shaded perspective view.

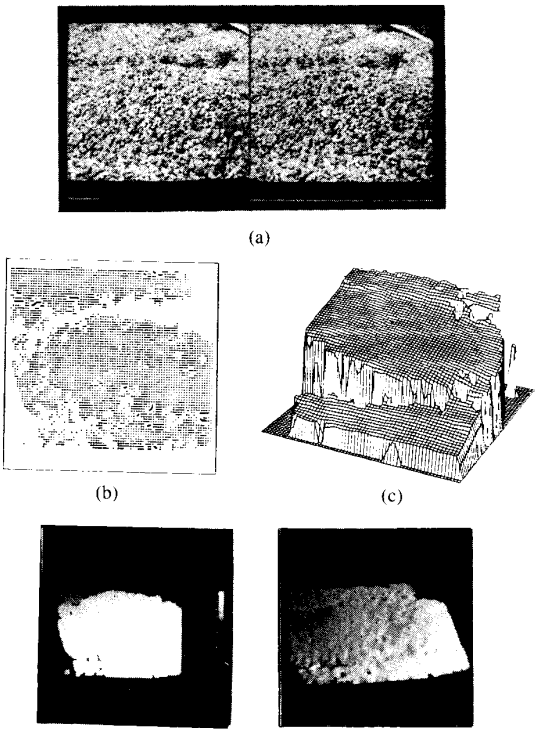


Fig. 7. (a) 512 × 512 real image of a mound of rocks. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d1) Surface as intensity image. (d2) Shaded perspective view.

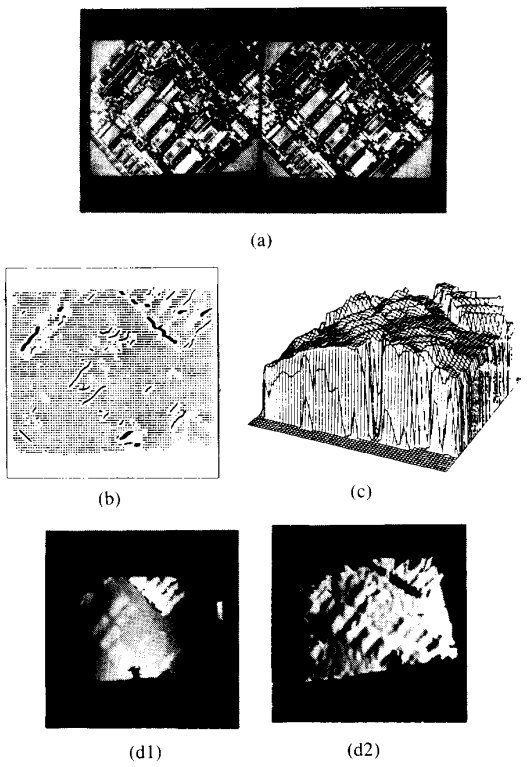


Fig. 9. (a) 512 × 512 real image of a circuit board. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d1) Surface as intensity image. (d2) Shaded perspective view.

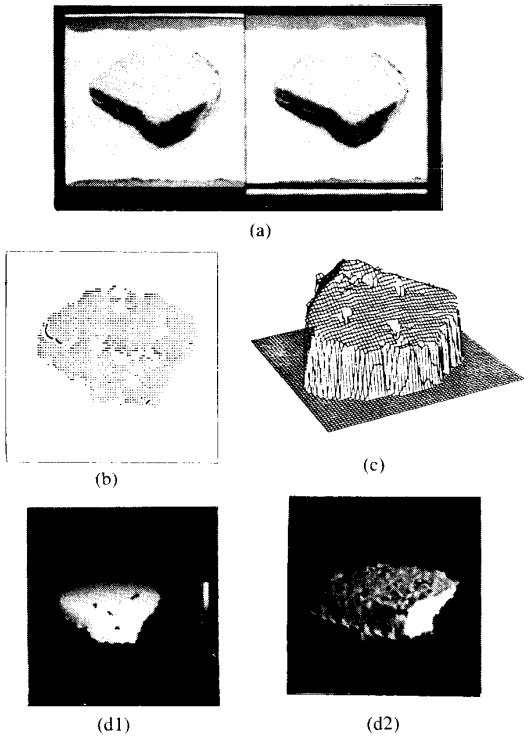


Fig. 8. (a) 512 × 512 real image of a sandwich. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d1) Surface as intensity image. (d2) Shaded perspective view.

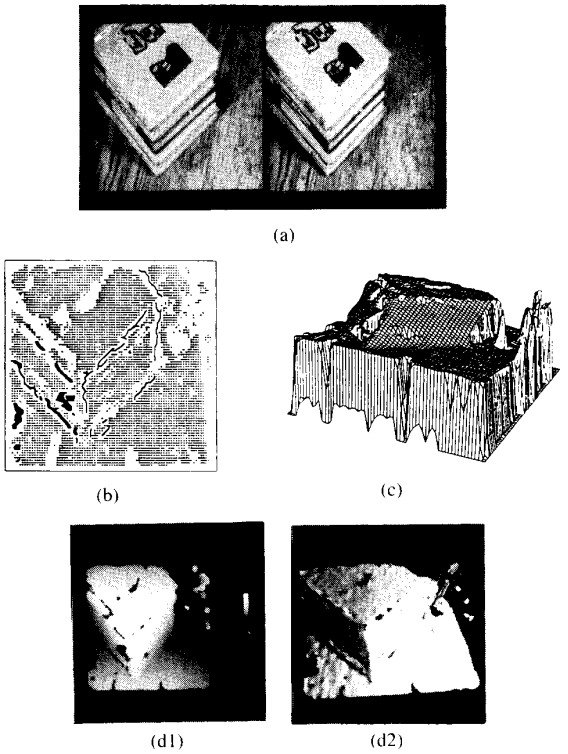


Fig. 10. (a) 512 × 512 real image of books. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d1) Surface as intensity image. (d2) Shaded perspective view.

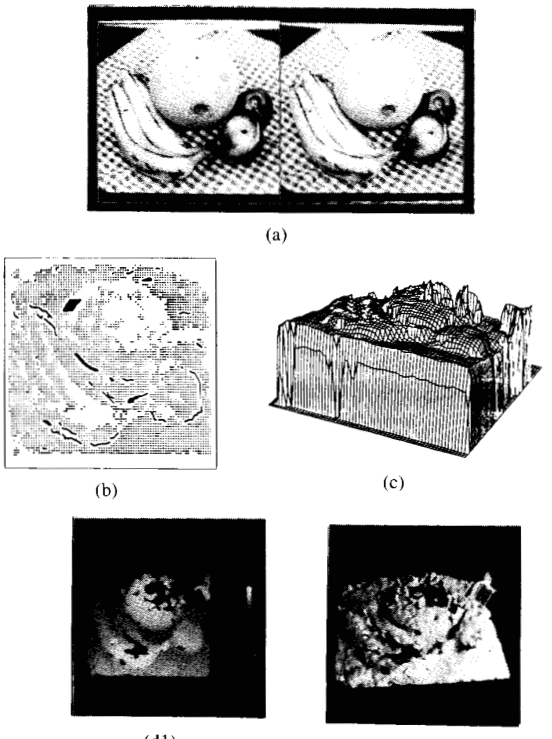


Fig. 11. (a) 512 × 512 real image of fruit. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d1) Surface as intensity image. (d2) Shaded perspective view.

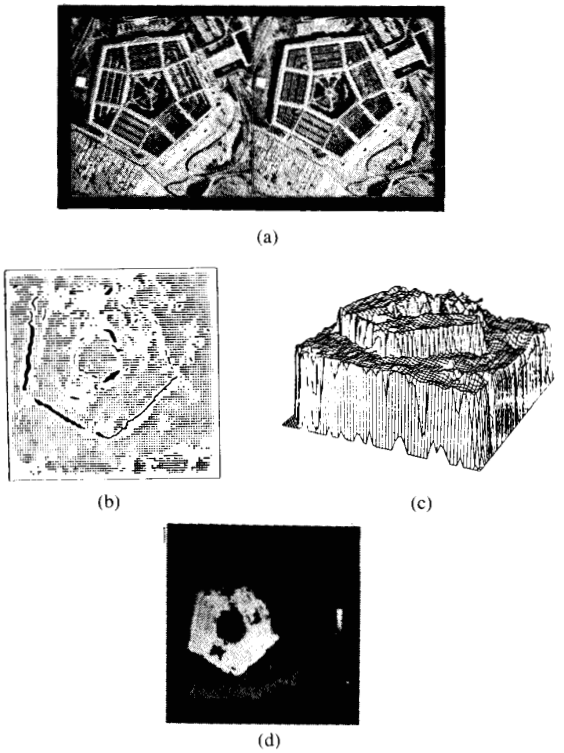


Fig. 12. (a) 512 × 512 real image of the Pentagon. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d) Surface as intensity image.

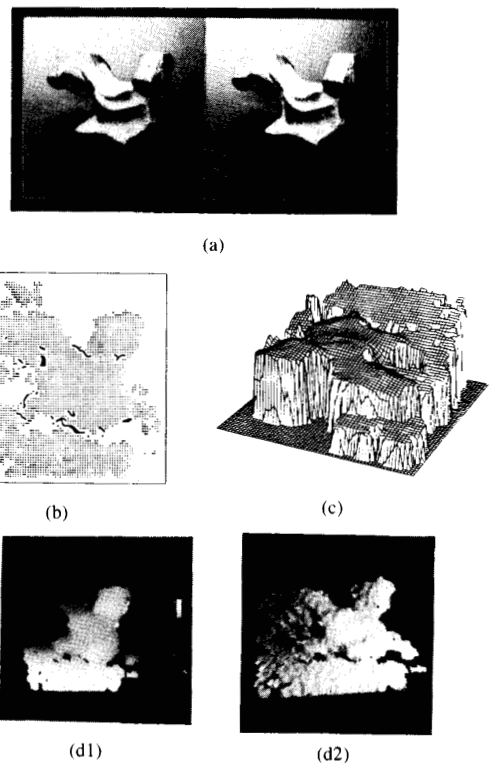


Fig. 13. (a) 512 × 512 real image of Renault auto part. (b) Quadratic patches and contours for the finest resolution level. (c) Reconstructed disparity surface. (d1) Surface as intensity image. (d2) Shaded perspective view.

(c). In (d), the surface is shown as an intensity image, or shaded, or both (black indicates "unknown" or "occluded").

Since accurate measurements were not taken of the camera parameters and object distances, the resulting surfaces cannot be compared point by point to the ground truth. However, the disparity was measured by hand at selected points, to confirm the accuracy of the results. The reconstructed surfaces were also qualitatively compared to the surfaces perceived by one of the authors (Hoff). The overall shapes of the surfaces appear correct, although there are problems in some places.

One problem is that there are "unknown" places in each surface, where surface patches could not be fit. One reason for this is the lack of significant image texture, most noticeably in the background of the sandwich example (Fig. 8). In that example, most of the zero crossings detected in the background were eliminated by the thresholding on the magnitude of the slope. The remaining zero crossings in the background were relatively uncorrelated, due to noise, and thus smooth surface patches could not be fit there. Another cause of "unknown" places is the presence of predominantly horizontal edges, for example in the topmost wing of the Pentagon building (Fig. 12). The algorithm does not attempt to match predominantly horizontal edges, because the disparity cannot be reliably measured. Finally, misregistration of the im-

TABLE II
REAL EXAMPLES

Figure	Example	Size	Disparity Range	Approximate Baseline	Approximate Subject Distance
5	Baseball	256 × 256	13	1 foot	4 feet
6	Ruts	512 × 512	67	1 foot	6 feet
7	Rocks	512 × 512	150	2 feet	10–20 feet
8	Sandwich	512 × 512	50[a]	3 inches	1.5 feet
9	Circuit Board	512 × 512	12	6 inches	1.5 feet
10	Books	512 × 512	46	1 foot	4 feet
11	Fruit	512 × 512	39	3 inches	1–2 feet
12	Pentagon	512 × 512	10	unknown	unknown
13	Renault	512 × 512	36[b]	unknown	unknown

[a]Disparity range of sandwich only.
[b]Disparity range of auto part only.

ages in the vertical direction can cause matching to fail. This happened in parts of the fruit example (Fig. 11), which were misaligned vertically by about 4 pixels. This could be due to nonideal camera optics.

The surfaces in the "unknown" places could be estimated by other methods; for example, by using the surface from the coarser levels, if known. Alternatively, a more global interpolation could be done: patches could be used from further away, or the surface could be interpolated using the depth points over the entire surface.

Occasionally, incorrect patches may be fit in the "unknown" regions described above. These are visible in the reconstructed surfaces of the rocks example (Fig. 7), books example (Fig. 10), and the fruit example (Fig. 11). Sometimes incorrect patches are caused by a local regularity in the image texture. For example, in the upper right corner of the books image (Fig. 10), the grain of the wood table is periodic, causing a large number of points to be mismatched. This is unavoidable because locally the patches are a good fit to the data, but globally they are not. The current algorithm performs only local matching and interpolation, and so is unable to recognize the error. A solution to this is to use more global information in deciding which patch is correct.

The circuit board in Fig. 9 does not appear planar in the disparity surface, as one would expect. However, fusion of this stereo pair is not perceived as planar to the human visual system, either. This may be due to distortion in the wide angle (25 mm) lens used to take the pictures, or it may be due to the transformation used to vertically register the images, as the angle between the view directions for this example is larger than any of the other image pairs. The surface appears to be correct by manually checking at isolated points.

Although the auto part is the main object in the example of Fig. 13, the table on which the auto part is resting is slightly dirty and has enough texture so that fusion of that area is possible. This shows the noise tolerance and generality of the method.

The contours found by the program are occasionally misplaced or missing, resulting in large disparity errors near occluding contours. The main reason for this is that the contours are detected and placed on the basis of local information, and the zero crossings in the vicinity of the contour may be sparse, or distorted by the blurring of different regions across the contour. Since we expect contours in the real world to be smooth and continuous, the detection and location of contours should be done while enforcing this constraint. The present algorithm only partially enforces smoothness, by fitting cubic splines to the detected contour points.

The edges detected are inaccurate many times, causing errors in disparity. Usually this is not a problem because the surface patches are fit to many points, and the errors tend to cancel out. However, in some cases artifacts are created, i.e., zero crossing contour segments which are present in one image but not in the other. This causes mismatched or unmatchable points. A smaller edge operator might give better localization, with fewer artifacts.

Occasionally the coarse levels provide an incorrect disparity estimate to the fine levels, causing the fine level to be unable to match the points. This usually happens when the surface at the coarse level is extrapolated into a unknown area, so that an estimate can be provided for that area. Since there are no points to constrain the surface there, a small error in the parameters of the known surface patch can cause a large error in the extrapolated surface. One solution to this would be to use a wider matching window in an effort to match the points, although this would be computationally expensive.

Finally, our algorithm would not work for images without texture. The simplest example would be a single vertical line segment in each image. The algorithm would not be able to fit a surface patch to the line segment, and so would not be able to match it. Since it is not uncommon for man made objects to have little or no surface texture, intensity edges may occur mainly along occluding or ridge contours in images of such objects. Thus, our algorithm alone is not capable of surface estimation in general, and complimentary stereo algorithms are necessary that could estimate surface from sparse image features.

V. ADVANTAGES OF INTEGRATION

The main advantage of integrating matching and surface interpolation is the implementation of the surface smoothness constraint. However, there are additional ad-

vantages: first, the availability of surface information allows occluded regions to be identified, so that matching of points within them is not attempted. Second, in the case of transparent surfaces, the features belonging to surfaces at different depths would be spatially intermixed in the image, and a two dimensional, local matching rule will not suffice. However, the surface fitting process only requires that the patches should have significant support from the depth points, and so it can yield multiple patches at any image location, one for each existing transparent surface at that location.[10] Because the depth points are separated in three dimensions, they can be identified as belonging to different surfaces. Finally, the disparity range over which matches are sought is completely adjustable, and does not have to be related to the parameters of the feature detector, e.g., $\nabla^2 G$, as done in the Marr–Poggio approach [3] and as pointed out by Mayhew and Frisby [25]. In principle, the range could be equal to the size of the image. This is because the number of false targets is irrelevant to the matching and surface fitting process; rather, the process relies on the existence of a set of depth points that define a smooth surface.

The advantage of explicitly detecting ridge and occlusion contours is that they can be constrained to be smooth because their counterparts in three dimensions are assumed to be smooth. This constraint is very useful because the feature locations in the image are usually sparse, particularly near a boundary of a steep surface. The contours could locally move in the interfeature (or "no-information") space without becoming inconsistent with the surface patches. The contour smoothness constraint makes it possible to propagate information between different parts of the boundaries to appropriately select the location of the contour when the local evidence does not lead to an unambiguous choice, or when it suggests a location that results in a large curvature. This should help in reducing the usually large number of depth errors that occur near surface boundaries. This step is implemented partially in the current implementation, in that although the contours are detected, only a coarse test of contour smoothness is used.

There are also advantages in using an explicit surface representation, at each resolution level. The explicit surface representation provides the common ground for interaction among different resolution levels. One such interaction is to use the coarse resolution surface to predict the locations of matches of finer resolution features. Another is to use the orientation of the coarse level surface to predict the difference in edge orientation between the left and right views. The difference in apparent orientation is due to the difference in perspective between the left and right views, when viewing the same three-dimensional edge. We can thus limit the candidate matches to those whose orientations are close to that expected, given the coarse level surface. A related effect is that of perspective compression. A surface region projected onto the left image will in general have a different image area than the same region projected onto the right image. Since the $\nabla^2 G$ operator generally yields edges of constant density, there will be fewer features in the compressed image region than in the uncompressed image region. If matching is done from the larger to the smaller region, this will cause unmatchable points. The coarse level surface can be used to predict the increase in the number of unmatchable points. This is important to know for feature matching. In our case, it is of immediate use since we have used the number of unmatchable points as a criterion for whether a surface patch is a good fit to a set of depth points.

Finally, the availability of explicit surface and boundary information at any given level makes it possible to change the focus of attention at the next finer level of processing. Thus, the algorithm may not spend a large amount of computation at the next finer level in processing an area which is relatively featureless. Rather, it may concentrate on areas near object borders, in order to more precisely locate them. The explicit knowledge of border locations may serve to guide the processing at finer levels, thus allowing a surface representation to be computed in a shorter time. This may relate well with the savings observed in fusion time in humans for scenes containing depth discontinuities, as reported by Gilliam *et al.* [26].

VI. Conclusions

We have presented an integrated approach to extracting surfaces from stereo. Along with performing matching and interpolation, depth and ridge contours are detected so as to enforce surface smoothness everywhere except across such contours. The contours are constrained to be smooth. The approach thus integrates matching, contour detection, and surface interpolation. These modules help exploit redundancy of information present in the image [27]. The integration approach is in contrast to existing stereo algorithms which complete the matching process before interpolating to obtain a dense depth map. As a consequence of integration, the computational effort is relatively uniformly distributed across the various integrated processes, unlike existing algorithms where most of the computation is devoted to feature point matching. The approach described is fairly domain independent since it uses no constraint other than the assumption of piecewise smoothness. The results are usually accurate to within a pixel of disparity. The errors occur mainly around the occluding boundaries, apparently because of errors in the locations of the boundaries.

The approach described lends itself to a parallel implementation since the processing in different parts of the image can be carried out in parallel. In fact, the algorithm was run on a network of Sun 3/160 and 3/75 workstations. At each of the major steps in the algorithm, the appropriate program is copied to each machine, along with the data needed by the program. Each machine independently processes a part of the image. Then the results are copied back to the original workstation and combined there. A

[10]The current implementation fits only a single patch at any location; thus, it does not handle transparent surfaces.

simple Shell script is used to automatically copy files to other workstations and execute the programs on them. Up to eight workstations were used simultaneously, although usually only about three were used. The factor of speedup provided by using multiple machines approached the number of machines as the number of machines used increased.

Zero crossings are detected using an array processor to implement the convolution via Fourier transforms. This program takes about 2 minutes to process a 512 × 512 image. The program which fits planar patches using the Hough transform is the most time-consuming part of the algorithm. It takes about 3 hours for the 512 × 512 level, on one machine. The program which fits quadratic patches takes about 50 minutes at the 512 × 512 level. The reason that these programs are so slow is the sheer number of patches—over 7000 at the 512 × 512 level. The program which finds contour points takes about 10 minutes at the 512 × 512 level, but this depends on the number of contour points present. The interpolation program takes about 45 minutes at the 512 × 512 level. The other programs have negligible execution times compared to these. Running an example usually took most of the night.

Although the implementation of this algorithm is quite slow, it is potentially quite fast, given the right architecture. The ideal situation would be to assign a processor to compute each patch. Additionally, special purpose hardware could be used to implement the Hough transform.

Appendix
Details of Implementation

A. Features Detected

The left and right images are each convolved with the $\nabla^2 G$ operator of the size (width of the Gaussian) corresponding to each resolution level. Zero crossings are then detected, and labeled with the orientation of the gradient. The result is a pair of edge images for each level. Due to subsampling, the effective width w of the $\nabla^2 G$ operator (the diameter of the central negative region) was the same for each level, i.e., 6 pixels.

We assume that the camera model is known, so that epipolar lines can be computed. The current implementation assumes horizontal epipolar lines, corresponding to parallel image planes. Searching for candidate matches is restricted to one dimension. The algorithm attempts to match only nonhorizontal zero crossings, since the disparity of horizontal zero crossings is subject to large error. In our experiments, a zero crossing at an angle of 22° or less from the horizontal axis was classified as horizontal.

For each zero crossing, candidate matches are found by searching a window of width $2w$ centered on the predicted location in the other image. Those zero crossings within ±35° of the expected orientation are taken to be candidate matches. The expected orientation is calculated from the surface orientation from the previous level. A large discrepancy in angle is allowed because experimentally we have found that the orientations of the right image zero crossings are not very close to those predicted by the orientations of the left image zero crossings and the camera geometry.

The reason for the discrepancy between actual and predicted orientation changes is that the zero crossings are not located at the true edge position, but are displaced if the image intensities are locally nonlinear, or if the edge is not straight, etc. In general, the displacement from the true position of the left image zero crossing and that of the right image zero crossing are different, because the local image structure is different due to differences in perspective compression, noise, etc. Therefore, the shape of the zero crossing contour is distorted from the left to the right image in ways other than predicted by perspective difference, and so the orientation changes between corresponding pieces of a contour may be different than predicted.

B. Fitting Planar Patches

Planar patches are fit in circular image regions centered at each point along a regular grid. The spacing of this grid is w, the filter size. A sparse grid is used to reduce the amount of computation. At each grid point (i, j), the quadratic surface estimate $z_{i,j}(x, y)$ obtained from the previous level is used to match the zero crossings in the circular region. Up to two planes are then fit to the depth values obtained by matching the points. This is done for a sequence of radii, starting at a radius of w, up to a maximum of $2w$. The largest possible disk is identified at each point under the constraint that the depth points in the disc are a good fit to a plane, using the two adequacy measures described in Section III-B.

Two planar fits are obtained instead of one, in order to delay the final choice until information from adjacent regions is available to reliably choose between the two. Also, if there are two surface estimates for this point from the previous level, the process is repeated for the second estimate, resulting in up to four planes for the region.

To ensure a reliable planar fit, the data points must be distributed over the entire region. This condition is tested by examining if the convex hull of the points on the image plane contains the region center; or equivalently, that the vectors to the points from the region center span an orientation range greater than 180°. If this condition is not satisfied, the plane solution is rejected.

A crucial part of this algorithm is the use of the Hough transform [21] to fit planar patches. Identifying the best-fitting planar patches in the vicinity of an image point requires selecting the most planar subsets of depth points among all possible combinations of mismatched and ambiguous points. Using a standard least squares method such as Gaussian elimination would lead to combinatorial explosion, because a different plane would have to be fit to each possible subset of depth points in a region. The Hough transform is a relatively inexpensive and robust method of fitting planes having least squared error.

To implement the Hough transform, a three-dimensional parameter space is set up with each dimension cor-

responding to a parameter in the equation of the plane:

$$z = ax + by + c. \tag{6}$$

For each depth point (x_i, y_i, z_i) in a circular region, cells in quantized parameter space are incremented at the locations corresponding to the solutions (a, b, c) of the equation

$$c = z_i - ax_i - by_i + \epsilon_i, \qquad \text{for } |\epsilon_i| < D \tag{7}$$

where ϵ_i represents the amount of error of fit of the point (x_i, y_i, z_i) to the plane represented by (a, b, c), and D is the outlier distance. The array is incremented at each location (a, b, c) by the amount $(D^2 - \epsilon_i^2)$. After all points have been considered in this manner, the maximum entry in the parameter array represents the solution with the minimum squared error.

In the case of ambiguous points, only one of the depth values contributes to any plane: the one which is closest in depth. There are two kinds of ambiguities: two left feature points matching the same right point, and two right feature points matching the same left point. Both kinds of ambiguities are taken into account. Because of outliers and ambiguous points, the solution found is not the true least squared error solution, but the solution with the least squared error among the points satisfying the above conditions.

An important advantage of the Hough transform is that it requires a constant amount of work for each depth value, and the amount of work is not exponential in the number of ambiguous points or mismatches, as would be the case with Gaussian elimination. A disadvantage of the Hough transform is the limited resolution of the parameter space. Higher resolution requires additional computation. One way of circumventing this problem is to use a Hough array with adaptive resolution; the resolution is dynamically increased in the parts of the parameter space where peaks are found at the coarser level. However, because the planes obtained are only local approximations, a very fine resolution is not crucial. In the implementation, the parameter space was $7 \times 7 \times 16$, with the first two dimensions used for the x and y slopes from -0.6 to 0.6, and the third dimension for the z offset. This allowed a resolution of 0.2 in the slope, and 1.0 in the z value. A maximum allowed slope of ± 0.6 is used because the probability of a point to be unmatchable increases with slope (as described earlier), and planes having greater slopes than this would have so many unmatchable points that they could not be distinguished from planes fit to random matches.

C. Fitting Quadratic Patches

To fit a quadratic surface centered at grid point (i, j), the following procedure is used. The planar patches centered at the neighbors of (i, j) are tested for mutual compatibility. Two neighboring planes are *compatible* if the depth and the orientation differences between them are less than certain thresholds. In the implementation, the disparity difference threshold is $w/2$ and the orientation difference threshold (i.e., the threshold on the difference in slopes) is 0.25. Two incompatible planar patches at neighboring grid points are likely to be separated by an occluding or ridge contour, and the two patches should not be part of the same quadratic surface. (These contours are found in later processing.)

To obtain all compatible planes at a given point (i, j), the planes in the neighborhood up to two grid points away from (i, j) are placed into sets, such that the members of each set are compatible with each other. This is done by the following procedure: for each plane in the neighborhood,[11] the parameters of the plane are transformed so that it is centered at (i, j). The transformed parameters are then compared with the averaged parameters of each set. If it is compatible with the average of one of the sets, then it becomes a member of the set; else it becomes a member of a new set. The two sets with the most members are now chosen, and the rest are discarded.

For each set, the planes in the set should be local approximations of the same quadratic surface. Therefore, the matches consistent with these planes should lie on or near the quadratic surface. These matches are obtained by taking the closest alternative to the plane within the outlier distance. A least squares quadratic surface is fit to these matches, using Gaussian elimination. As before, the squared error is compared to the maximum expected error as given by the χ^2 distribution, and the fit is rejected if the error is too large. The quadratic surface containing the most points is kept as the fit for the grid point (i, j).

D. Locating Contours

To find occluding contours, we use the model of a bipartite surface patch: a circular region with two independent smooth (planar) halves, separated by a depth discontinuity at the center. The approach is analogous to that described by Leclerc and Zucker [28] for finding discontinuities in image intensities: it is necessary to find the local structure of the image (or surface) about the discontinuity in order to locate the discontinuity accurately. Our approach differs in that a fixed threshold is used to signal a discontinuity, instead of a statistical test. To find ridge contours, the model is the same, except that it uses an orientation discontinuity instead of a depth discontinuity.

To obtain the depth points needed for the plane fitting, the closest matches to the surface specified by the nearest quadratic patch are used. Only matches within the outlier distance to that surface are used. To help increase the reliability of the planar fit, the radius of each semicircle is increased until it contains at least 15 points, from a minimum radius of $3w$ to a maximum radius of $5w$. The same two tests are used as before to decide if the planes are a good fit to the data points: 1) the binomial test, for the

[11]Currently, the neighboring planes are examined in arbitrary order. However, a more precise method would be to represent all neighboring planes as points in parameter space, and then search for clusters in this space.

number of unmatchable points, and 2) the χ^2 test, for the total error of fit of the points to the plane.

If the two planes are a good fit to the data points, and they differ in disparity (or orientation) by more than a threshold, then there is evidence for an edge point in the vicinity of the grid point. The threshold used was w for the disparity difference, and 0.25 for orientation difference (slope difference). To locate the edge point more accurately, the following procedure is used: the bisector of the bipartite circular patch forms an edge segment that partitions the matches into two sets, belonging to one side or the other. This edge segment is moved to and fro between the two sides. As the edge segment moves, some data points that were originally in one side of the bipartite patch become members of the other side. At each position, a score is computed, which is the average squared error of fit of the points to the surface whose side they are currently in. The edge point is placed at the position with the lowest score. This method yields good localization of the edge point, while saving the expense of fitting the edge detector at every point.

For occluding edge points, the above procedure is modified by establishing a "dead zone" adjacent to the edge segment, on each side of the segment. Any matches inside the dead zone are ignored and they do not contribute to the score. The reason for this is that the zero crossings near occluding contours are typically distorted by the contour and the matches are not reliable. The distortion is typically significant out to a distance of $w/3$, and so this is the width of the dead zone that was used.

The bipartite circular edge detector is applied four times at each grid point to detect edges at the four orientations: vertical, horizontal, and the two diagonals. If an edge is detected at a particular orientation, it is localized and given a score, as described above. The edge orientation at this grid point with the best score is retained.

This edge detector may have multiple responses, i.e., it may signal the presence of an edge at multiple locations in the vicinity of the true edge, along a line perpendicular to the true edge. However, the best score should ideally occur at the position of the true edge. Therefore, the edge points which are not local minima in the direction perpendicular to their orientations are suppressed.

False ridge contours are occasionally detected parallel to occluding contours, on either side. These arise because surface patches are occasionally fit across the occluding edge, forming a steep ramp. To eliminate these false ridge contours, the algorithm eliminates ridge edge points near occluding contours. In addition, patches which overlap contours are eliminated, because they are adversely affected by the depth points on the other side of the contour, resulting in an incorrect surface estimate.

E. Generating a Surface Map

The final step is to interpolate to obtain a complete depth map, and predict matching locations for features at the next finer level. The quadratic patches are a good local estimate of the surface, defined at the grid points. To interpolate the depth at each point P on the surface, the closest patch or patches to point P are used, which do not lie across any occluding or ridge contour. In the implementation, patches were used if their centers were up to $2w$ from point P. The computed height at point P is the average of the heights of individual patches weighted according to their distance to P. If there are no patches within $2w$ of P, then no attempt is made to interpolate a depth at P, and it is marked "unknown." If the point P is in an occluded region, no attempt is made to interpolate a depth, and it is marked "occluded." A reasonable guess for the depth at such a point can be made by extending the surface which is more distant out to P; this, however, was not done.

To predict matching locations for the next level, the set of quadratic patches is copied to a new grid, twice the size of the old. There are now new grid points which do not have quadratic estimates, and these must be interpolated from the existing ones. In addition, in the vicinity of occluding contours, we would like to provide two depth estimates—one from the high side of the surface, and one from the low side. This is done because the location of the occluding contour is known only coarsely, and we would like to be able to match the zero crossings in the vicinity of the contour.

To accomplish the above, the following procedure is repeated N times, to propagate the known estimates to the unknown areas ($N = 4$ in our implementation). At each grid point on the new level for which there are less than two estimates, examine the 8 neighboring quadratic patches. If these patches are mutually compatible (meaning their parameters have similar values), then average them to determine the quadratic estimate at the new point. If they are not all compatible, divide them into compatible sets, and average the ones in each set to determine up to two estimates.

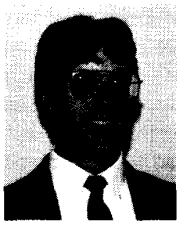
ACKNOWLEDGMENT

Thanks are due to anonymous reviewers who made very useful comments on the manuscript.

REFERENCES

[1] W. Richards, "Stereopsis with and without monocular contours," *Vision Res.*, vol. 17, pp. 967-969, 1977.
[2] B. Julesz, *Foundations of Cyclopean Perception.* Chicago, IL: University of Chicago Press, 1971.
[3] D. Marr and T. Poggio, "A theory of human stereo vision," *Proc. Roy. Soc. London*, vol. B 204, pp. 301-328, 1979.
[4] D. Marr, *Vision.* San Francisco, CA: Freeman, 1982.
[5] W. E. L. Grimson, *From Images to Surfaces.* Cambridge, MA: MIT Press, 1981.
[6] S. T. Barnard and M. A. Fischler, "Computational stereo," *ACM Comput. Surveys*, vol. 14, no. 4, pp. 195-210, Dec. 1982.
[7] A. Rosenfeld, R. Hummel, and S. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, pp. 420-433.
[8] G. Medioni and R. Nevatia, "Segment-based stereo matching," *Comput. Vision, Graphics, Image Processing*, vol. 31, pp. 2-18, July 1985.
[9] N. Ayache and B. Faverjon, "Fast stereo matching of edge segments

using prediction and verification of hypotheses," in *Proc. Computer Vision and Pattern Recognition*, June 1985, pp. 662-664.
[10] J. E. W. Mayhew and J. P. Frisby, "Psychophysical and computational studies towards a theory of human stereopsis," *Artificial Intell.*, vol. 17, pp. 349-385, Aug. 1981.
[11] W. E. L. Grimson, "Computational experiments with a feature based stereo algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 17-34, Jan. 1985.
[12] H. H. Baker and T. O. Binford, "Depth from edges and intensity based stereo," in *Proc. Int. Joint Conf. Artificial Intell.*, Aug. 1981, pp. 631-636.
[13] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 139-154, Mar. 1985.
[14] W. Hoff and N. Ahuja, "Extracting surfaces from stereo: An integrated approach," Univ. Illinois Coordinated Sci. Lab., Tech. Rep. ILU-ENG-87-2204, Jan. 1987 (a part of this report appears in *Proc. 1st Int. Conf. Computer Vision*, London, June 1987, pp. 284-294).
[15] ——, "Depth from stereo," in *Proc. Fourth Scandinavian Conf. Image Analysis*, Trondheim, Norway, June 18-20, 1985, pp. 761-768.
[16] ——, "Surfaces from stereo," in *Proc. DARPA Image Understanding Workshop*, Miami Beach, FL, Dec. 1985, pp. 98-106.
[17] S. Olsen, "Concurrent solution of the stereo correspondence problem and the surface reconstruction problem," in *Proc. Eigth Int. Conf. Pattern Recognition*, 1986, pp. 1038-1041.
[18] H. P. Moravec, "Towards automatic visual obstacle avoidance," in *Proc. Fifth Int. Joint Conf. Artificial Intelligence*, Cambridge, MA, 1977, p. 584.
[19] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London*, vol. B 207, pp. 187-217, 1980.
[20] R. D. Eastman and A. M. Waxman, "Using disparity functionals for stereo correspondence and surface reconstruction," *Comput. Vision, Graphics, Image Processing*, vol. 39, pp. 73-101, 1987.
[21] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley, 1973.
[22] J. Clark and P. Lawrence, "A theoretical basis for diffrequency stereo," *Comput. Vision, Graphics, Image Processing*, vol. 35, pp. 1-19, 1986.
[23] J. D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*. Reading, MA: Addison-Wesley, 1983.
[24] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. New York: Dover, 1956.
[25] J. E. W. Mayhew and J. P. Frisby, "The computation of binocular edges," *Perception*, vol. 9, pp. 69-96, 1980.
[26] B. Gilliam, T. Flagg, and D. Finlay, "Evidence for disparity change as the primary stimulus for stereoscopic processing," *Perception Psychophys.*, vol. 36, no. 6, pp. 559-564, 1984.
[27] M. Brady, "Artificial intelligence approaches to image understanding," in *Pattern Recognition Theory and Applications: Proc. NATO Advanced Study Institute*, Mar. 29-Apr. 10, 1981, J. Kittler, K. S. Fu, and L. F. Pau, Eds. Dordrecht, The Netherlands: Reidel, 1982.
[28] Y. Leclerc and S. W. Zucker, "The local structure of image discontinuities in one dimension," in *Proc. 7th Int. Conf. Pattern Recognition*, July 1984, pp. 46-48.
[29] K. L. Boyer and A. C. Kak, "Structural stereopsis for 3-D vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 144-166, Mar. 1988.
[30] T. E. Boult and L. H. Chen, "Analysis of two new stereo algorithms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Ann Arbor, MI, June 1988, pp. 177-182.



William A. Hoff (S'83-M'87) received the B.S. degree in physics from the Illinois Institute of Technology, Chicago, in 1978, and the M.S. degree in physics and the Ph.D. degree in computer science from the University of Illinois, Urbana-Champaign, in 1981 and 1987, respectively.

From 1978 to 1980 he was an engineer at Teletype Corporation, Skokie, IL. From 1983 to 1986 he was a Research Assistant at the Coordinated Science Laboratory at the University of Illinois. He is currently a Staff Engineer at Martin Marietta Astronautics Group, Denver, CO, where he is working on computer vision systems for space applications. His research interests include stereo vision, object recognition, and computer architectures for real-time computer vision.

Dr. Hoff is a member of the Association for Computing Machinery and the IEEE Computer Society.

Narendra Ahuja (S'79-M'79-SM'85) received the B.E. degree with honors in electronics engineering from the Birla Institute of Technology and Science, Pilani, India, in 1972, the M.E. degree with distinction in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1974, and the Ph.D. degree in computer science from the University of Maryland, College Park, in 1979.

From 1974 to 1975 he was Scientific Officer in the Department of Electronics, Government of India, New Delhi. From 1975 to 1979 he was at the Computer Vision Laboratory, University of Maryland, College Park, as a Graduate Research Assistant (1975-1978), as a Faculty Research Assistant (1978-1979), and as a Research Associate (1979). Since 1979 he has been with the University of Illinois as an Assistant Professor (1979-1984), Associate Professor (1984-1988), and Professor (1988-) in the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory. His research interests are in computer vision, robotics, image processing, and parallel algorithms.

Dr. Ahuja received the University Scholar Award (1985) and a Presidential Young Investigator Award (1984). He has coauthored the book *Pattern Models* (Wiley, 1983). He was Program Chair for the 1987 IEEE Workshop on Computer Vision. He is an Associate Editor of *Computer Vision, Graphics and Image Processing*, and a member of the Association of Computing Machinery and the American Association for Artificial Intelligence.