Stanford Artificial Intelligence Laboratory
Memo AIM-248

May 1975

Computer Science Department
Report No. STAN-CS-74-462

A Fast, Feature-Driven Stereo Depth Program

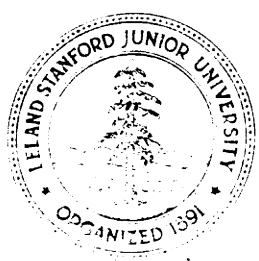
by

Karl K. Pingle
Arthur J. Thomas

Research sponsored by

Advanced Research Projects Agency
ARPA Order No. 2494

COMPUTER SCIENCE DEPARTMENT
Stanford University



Stanford Artificial Intelligence Laboratory
Memo AIM-248

May 1975

Computer Science Department
Report No. STAN-CS-74-462

A Fast, Feature-Driven Stereo Depth Program

by

Karl K. Pingle
Arthur J. Thomas

ABSTRACT

In this paper we describe a fast, feature-driven program for extracting depth information from stereoscopic sets of digitized TV images. This is achieved by two means: in the simplest case, by statistically correlating variable-sized windows on the basis of visual texture, and in the more complex case by pre-processing the images to extract significant visual features such as corners, and then using these features to control the correlation process.

The program runs on the PDP- 10 but uses a PDP- 11/45 and an SPS-41 Signal Processing Computer as subsidiary processors. The use of the two small, fast machines for the performance of simple but often-repeated computations effects an increase in speed sufficient to allow us to., think of using this program as a fast 3-dimensional segmentation method, preparatory to more complex image processing. It is also intended for use in visual feedback tasks involved in hand-eye coordination and automated assembly. The current program is able to calculate the three-dimensional positions of 10 points to within 5 millimeters, using 5 seconds of computation for extracting features, 1 second per image for correlation, and 0.1 second for the depth calculation.

This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC 15-73-C-0435 . The views and conclusions contained in this document are those of the author(s) and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Stanford University, ARPA, or the U. S. Government.

Reproduced in the U.S.A. Available from the National Technical Information Service, Springfield, Virginia 22151.

1. A Rationale for the extraction of depth information

It is, by now, a common-place of image-processing methods in artificial intelligence that three-dimensional information can play a crucial role in the interpretation of a visual scene. In a number of cases, (Guzman,1968;Falk,1970;Waltz,1972), some intuitions about the possible three-dimensional configurations of polyhedra and polyhedral vertices have been incorporated into programs which segment and interpret simple table-top scenes. In at least one case (Nevatia & Binford,1973; Nevatia,1974)) raw depth information, gained by scanning a scene with a laser, has been used with some success for the segmentation, description and recognition of complex curved objects. In the near future, we envision that three-dimensional information will be useful in two major areas: in the first, that of hand-eye assembly systems, it will be used to effect visual feedback; the second area is the provision of disambiguating information for more sophisticated and 'knowledgeable' vision programs, e.g. for an extenslon of the model-based line-fitter described by Grape (Grape,1973). We also envision its use in deciding where in a scene visually interesting structures exist, so as to minimise the need to apply more computationally expensive methods in those areas where there are none.

However, little work seems to have appeared so far on fast, efficient methods for gaining such depth information from TV images (but see, for example, Yakimovsky,1974; Hannah,1974; Bajcsy,1972). Those programs which have been described seem to embody a wish for very accurate spatial localisation, at the expense of speed. Our concern is much more with providing a 'quick and dirty' way of getting reasonable spatial resolution, albeit at the risk of occasional ambiguity or error. We contend that this is a reasonable approach on two grounds:

(a) since spatial information gained in this way is intended as a preliminary to more sophisticated processing, we can assume that any errors or inconsistencies will be caught by the higher-level routines;

(b) there is suggestive evidence from the human visual system that much of the extraction of three-dimensional information is carried out very fast at quite a low-level, with only a limited amount of guidance from other kinds of visual processing. The work of Julesz(Julesz,1971) shows that there exists in humans a well-developed capacity for extracting depth-information of reasonable resolution even in the absence of (monocular) form cues. However, it is not clear what the role of such low-level detection of disparity is in normal peception. It seems likely that in normal circumstances, a feature-guided correlation technique is being used. Recent neurophysiological evidence, (e.g. Blakemore,1970) shows that in the occipital cortex of the cat there exists a dual representation of visual features: so called 'orientation' columns define the location of a stimulus feature of given properties within the retinal field, while the 'depth' columns receive binocular input and encode the inter-ocular disparity and thus the distance of a given stimulus. In an early study, Attneave (1954) showed that much of the 'information' encoded in a visual configuration is concentrated at the places where sharp changes take place in the radius of curvature of the boundary, that is, at *corners* of varying degrees of sharpness.

One can make a case, then, for a correlation mechanism which is guided by a feature-detection pre-processing method. This is the kind of mechanism which we describe in the sequel.

What kind of efficiency and resolution should one expect from such a mechanism? An elementary calculation suggests that, given a 3 degree visual angle at one metre distance, and making some assumptions about the spatial acuity of the retinal system, that human beings should be able to achieve a spatial resolution of the order of 0.1-0.5 mm. Intuitively, one would expect relative depth acuity to be substantially better than absolute positional accuracy; human beings are plagued with the same kinds of calibration problems which we discuss in Section 6, but their hand-eye coordination is substantially better than positional accuracy might lead one to expect, because of visuo-motor feed back.

2. Correlation based on textural cues alone

An earlier study (Nevatia,1975) investigated the possibilities of achieving motion stereoscopy using only bulk-correlation techniques. This study investigated, in particular, various statistical measures for correlation, and some heuristics for tracking features through multiple views. Our work here is concerned with the inadequacy of unguided bulk-correlation techniques alone, and it combines those techniques with a feature-extraction algorithm. In that sense, our work here is both an extension and a critique of Nevatia's studies.

In this section we will describe methods of extracting from multiple-view TV pictures windows suitable for bulk-correlation. We will suggest why these simple-minded methods are inadequate, and how they may be improved upon by pre-processing the pictures in order to extract various kinds of important features by use of a corner finder described in section 3.

First, however, some discussion of our vision system is in order.

2.1, Hardware

The hardware for the stereo program consists of a large table with two television cameras mounted at one end and a turntable mounted flush with the table surface at the other end. The turntable can be rotated under computer control in either direction, and positioned with an accuracy of about 1/4 degree. The position of the table can be read by the computer with an accuracy of .O 175 degrees. Each of the TV cameras can input rectangular portions of their image of any size selected by the program up to the full image, which contains 333 image points horizontally and 256 vertically. Each image point is entered as a four bit intensity. Both cameras are mounted on motor driven pan-tilt heads and have a color wheel and a motor driven focusing system. One of them has a four lens turret, and the other a zoom lens. All of these can be servoed manually or under

computer control and the computer can read the position of any of the motors. Even though two cameras are available, we have chosen not to use them simultaneously to get a true binocular effect. This is because at present they are some 35 degrees apart, and we do not think that the techniques which we describe here, nor the mechanisms involved in human stereoscopic depth perception, can operate at this wide angle. The solution for body locus at such disparity levels is, we believe, at a very different computational level than its narrow-angle analogue, since it involves having much more information about, the geometric nature of physical objects. It is suggestive that, while young children have quite well-developed ability to grasp objects near them in space, they do not seem to develop continuity and conservation laws for such objects until considerably later (Piaget & Inhelder, 1967). Work is in progress at this laboratory on wide-angle methods (Baumgart, 1974; Ganapathy, 1975).

2.2, Coordinate Systems

There are two coordinate systems used by the program. The TV cameras have a two dimensional coordinate system on their image plane, with Y being the scan line and X being the sample point in the line. A three dimensional coodinate system is laid out on the table in inches, with the origin at one corner.

Calibration programs are available which generate a model of each camera. From this, a coordinate transform can be, calculated, based on the camera's current orientation. This transform and its inverse can then be used to translate any point in 3-space from table coordinates to the TV coordinates and any point in the TV image can be translated into its projection onto the table in the table coordinate system.

2.3, Taking Pictures

The stereo program could easily take live pictures as it runs. For debugging purposes, however, the program currently uses stored pictures. Another program, which is a modified version of our general purpose picture input program, is available to create stereo pictures. It reads in from two to ten pictures, with constant areas and camera position, from either camera. In between each picture, the turntable is rotated a fixed increment selected by the operator. The program *also* reads the current orientation of the camera in use, reads in the camera model stored by the calibration programs, and generates the proper coordinate transform for the pictures. A disk file is written containing the images, the exact position of the turntable for each image and the transform. As we mentioned above, these pictures are taken by rotating the scene, rather than by moving the camera or using two cameras. Taking a series of pictures which are only 0.5 degrees or so apart allows a reasonable depth resolution together with relative ease of tracking visual features from one frame to the next. In the conditions which we describe, rotating the object is closely analogous to rotating the tamer a.

2.4, Finding windows of interesting content

The first thing the program needs is a set of windows in the image to correlate on. These windows can be of any shape, but in both our work and Nevatia's, square windows were used. The windows must contain a variation in intensities to give us something on which to correlate, and the variation must be a pattern which is unique in this part of the image. Obviously, we cannot correlate on any area which has a constant intensity, nor can we correlate on totally random texture patterns. Also, a straight line, or series of parallel lines, will not give meaningful results, since the window can slide along them, getting similar results at any point. Once a set of windows is found,

the program can use correlation to track them through a set of images taken at different positions of the turntable. The relative depth, from the lens center, will be found using the shift in the position of the windows between the first and last images.

In his work, Nevatia applied to the image a variance operator which could detect windows containing intensity fluctuations. The operator was applied sequentially to the entire image and then the windows found were used for correlation. This method has several disadvantages, most of which result from the limited knowledge of the nature of the intensity fluctuations in the selected windows. This scheme will use windows containing random texture and straight lines, resulting in many windows where meaningful results cannot be obtained. Also, the windows may contain areas of different depth which, again, can cause problems with the correlation. Portions of the image whose depth may be of interest, such as the corners of objects, may be on the boundary of two windows and, thus, not be tracked using either of them.

To reduce the number of windows which will give poor results, we introduce the concept of windows containing features. A feature may be defined here as a set of gradienti in a small area which. exhibit a distinct pattern. For our work, these features are the intersections of intensity d&continuities where there is a significant difference in the directions of their gradients. This normally means the corners of objects, or at least places where the radius of curvature changes sharply. We search the image for suitable features and then define the coordinates of the features as the centers of the windows we will correlate on. Then, given the shift in the window, we can calculate not only the relative depth between two windows, but the actual location of the feature point in three dimensions.

There are several conceivable methods of extracting these features from the image. One would be to have a high level program which has traced the objects in the scene determine where

corners can be found in areas where it needs depth. It could then call the stereo program and give it TV coordinates of the corners in which it is interested. Currently no such program is available, although edge extraction programs do exist which will probably be modified in the future to perform this task. For the moment, the stereo program is left to find its own features by one of two methods.

The simplest feature extraction method allows the user of the program to move a cursor on the TV monitors by means of keyboard commands and, when the cursor is pointing at a feature of interest, to instruct the program to remember its coordinates. We envisage that this method will be used in the specification of interesting features of machine parts in an assembly task when visual tracking of specific points is desired.

The usual method of feature extraction, however, is for the program to apply **a variance *operator*** to the image to find the areas with sufficient intensity variation to be interesting and then to apply a ***corner finder*** to extract features automatically. The corner finder is not necessarily expected to find all features which could be correlated on, but rather the most distinct and relatively simple ones which can be found quickly and which will give good correlations. Our variance operator differs from Nevatia's in that it is designed to select windows which may contain edge segments which are small in area but large in intensity, while his is best at detecting areas with more widespread gradients, possibly with smaller intensities.

The variance operator consists of a 16x16 window which is applied to the image in a raster scan, with windows overlapped by 1/4 of their width on each side. At each application a histogram is generated giving the number of sample points in the window for each intensity level, as well as the average intensity in the window. Then, if A is the average, i is an intensity level, and H_i is the histogram count for that level, the variance becomes:

$$\text{VARIANCE} = [\Sigma_i |i-A| * Hi] / 256$$

All applications which result in a variance over a selected threshold cause the corner finder to be applied to the area.

3, The Corner Finder

The corner finder begins by applying a 3x3 *vector gradient operator* to every point in the window. At each point where the magnitude of the gradient is over a threshold, the routine saves the coordinates of the center of the operator, its intensity, the X and Y components of the gradient vector, *a* direction number from zero to seven specifying which half quadrant the vector is in, and the exact angle of the vector from the +X axis. A list of points is created for each direction number, pointing to the data for *all* gradient points with that number. After the operator has been applied to the whole window, a count is made of the number of directions which contain over three points. If there are less than two this window is rejected, since there is at most one edge direction represented, and the program continues with the variance operator. The window is also rejected if there are more than six of the eight directions with the proper count and the total number of gradient points is more than half the number of points in the window. This is assumed to be random texture, or an area too complex to be analyzed by this corner finder.

For each direction number, the gradient points are grouped into *line segments*. To be added to a given line, a point must have an angle close to the slope of the line and the gap between this point and the closest point on the line must be under a threshold, which is very small if the direction of the gap is perpendicular to the direction of the line but larger if the gap is collinear with the line. After finding all of the points for a given direction number which are in a line segment, the directions on each side of it are checked in case the line's slope is close to the boundary between the half quadrants. All lines with four or more points are kept. If there are less than two

of them, the window is rejected. The line equation for each line is calculated using the averages of the coordinates of the operator centers for the applications on the line and the directions of the vectors, each weighted by the intensity of the gradient for that application. Then the line is intersected with all other lines whose slopes differ from its slope by a large enough angle. All intersections which are inside the window and close enough to points on each tine are saved. Finally, all intersections which are close enough together are merged. If more than two distinct intersections are found, the window is rejected as being too complex. Otherwise, the intersections found, if any, are returned as the centers of features for this window.

4, Correlation

Once a set of features has been obtained, either manually, or by the corner-finder, each feature must be tracked through the images to locate it in the final one. This is done by reading in consecutive images and locating each feature In the new image by applying a correlation operator to it and the first image. For the first two images, the 16x16 window is centered on the feature point in the first image and the same coordinates In the second image. For the remainder of the images each window'is centered on the point where the feature is predicted to be, based on how far it shifted between the last two images.

Then the window is moved parallel to the X axis of the image in both directions, applying the correlation operator until the score begins to increase. Then similar scans are ptrfomed above and below the initial scan line until lines have been scanned on both sides which have a higher minimum score than one of the in-between lines. The lowest score found, of course, defines the new position of the feature. To speed up processing, the correlation is performed at *every* other point on the scan lines in all but the final image, since this has proven to locate the position accurately

enough for tracking In the intermediate images. In the final image, the shift is estimated to within .1 points by extrapolating beween the coordinates of the best and second best correlation scores.

If X_i and Y_i are corresponding points in the windows when applied to a pair of images, the correlation operator becomes:

$$C = \Sigma_i |A_i - B_i|^2$$

5, Depth Calculation

Figure 1 shows the geometry of the system α is the angle of rotation of the turntable between the initial and final images which is provided by the program which read in the images. C is the center of rotation of the turntable in the table coordinate system which is a constant and can be determined manually, since there is a grid surface on the entire table. L is the lens center in table coordinates, which is calculated from the camera model data. R is the image point, in TV coordinates, of a feature point T in the initial image which was provided by the corner finder. S is the image point, in TV coordinates, of the same feature point, now at U, in the final image, as determined by the correlation routine.

Our task is to calculate T in table coordinates and the length of line TL. By ignoring for the moment that L is above the table, we do all but the final calculations in the two dimensional case, using the plane containing the table top and, at the end, translating the results to the actual plane, LTU.

Rather than solve the equations involved to obtain an analytical solution, which appears to be fairly difficult since they are quadratics, we use iteration to find a solution to the accuracy we need. The iterative process is very fast, reducing the need for attempting an analytical solution.

Using the coordinate transform provided by the picture input program, we first transform

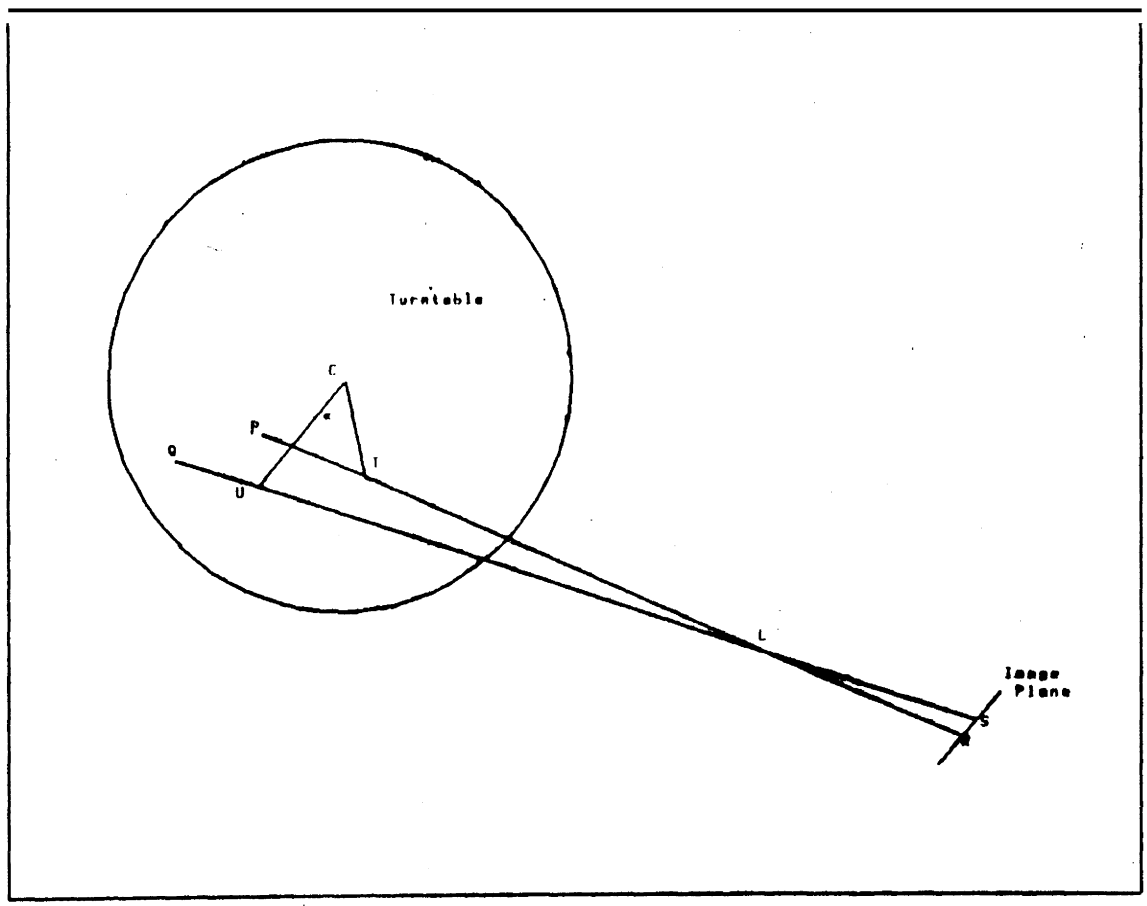


Figure 1: **Depth Calculation**

points R and S into points P and Q, their projections onto the table surface from point L. Then we find the line equations for LQ and LP. The problem now becomes finding points T and U on these lines satisfying the following equations:

$$(1)\ |U-C| = |T-C|$$

since the distance from the turntable center to the feature point is a constant during the rotation.

$$(2)\ |U-T|/|T-C| = 2*\sin(\alpha/2)$$

obtained, by elementary trigonometry, from the fact that, with UC and TC of equal length, the perpendicular bisector of UT will pass through C, bisecting angle α.

A point X on line LP is selected 3/4 of the way from L to P, since the feature must be close to that end to be on the turntable, and equation 1 is solved for a point Y on line LQ. The solution can take several forms:

(a) there is no solution which lies between L and Q. This means that the point we are trying to find lies nearer point P. The distance to P is halved and we try again.

(b) there is only one solution which lies between L and Q. It will be used. The other solution is for the case where the object is on the other side of the turntable center and below the table plane.

(c) there are two solutions which lie between L and Q and they are equal. This case occurs when the perpendicular bisector of UT is also perpendicular to LC. It will be used;

(d) there are two solutions which lie between L and Q and they are not equal. This case occurs because the feature point may be either in front of, or behind, the perpendicular to CL. If the angle from PC to QC has the same sign as α, the feature is behind the perpendicular; otherwise it is in front of it. The appropriate solution is used.

The proper solution for Y is selected and, from equation 2 we compute

$$A = |X-Y|/|X-C|$$

By comparing the magnitude of A to $2*\sin(\alpha/2)$, we determine which direction to move from X along LP to get the next starting point, X. If we move toward P, the interval is halved. If we move toward L, X is again chosen 3/4 of the way to L. This process continues until the interval along LP being searched is less than the accuracy we require, giving T in two dimensions. Then, using similar triangles, the height of T above the table and its distance from L is calculated.

In addition to finding the depth, features composed of lines of different depths can often be detected. The easiest case is seen in figure 2(a). A, B, and C are part of the same object, while D is part of a second object behind the first. As the scene rotates, the corner divides into two corners, as seen in figure 2(b). This will result a very high minimum score for the correlation.

In figure 3, A and B are the edge of one object and C is an object behind it. As the scene rotates the 'corner' will slide along line AB, resulting in the wrong depth being calculated. If the depth at other points of AB is known, and a higher level program has determined that AB is one edge of an object, the depth of the intersection will be inconsistant with the other depths, showing that C is part of another object. We could eliminate this case by not using features containing two collinear line segments, but we have chosen not to do this. Cues such as these two are often useful to programs which are using depth, as well as other information, to separate objects out of complex scenes.

6, Implementation and Results

As might be expected, the program works best when the distance between the intial and final position of the features in the TV image is large. The program can determine which of two features is further from the camera if the difference in their distance from the lens center is over

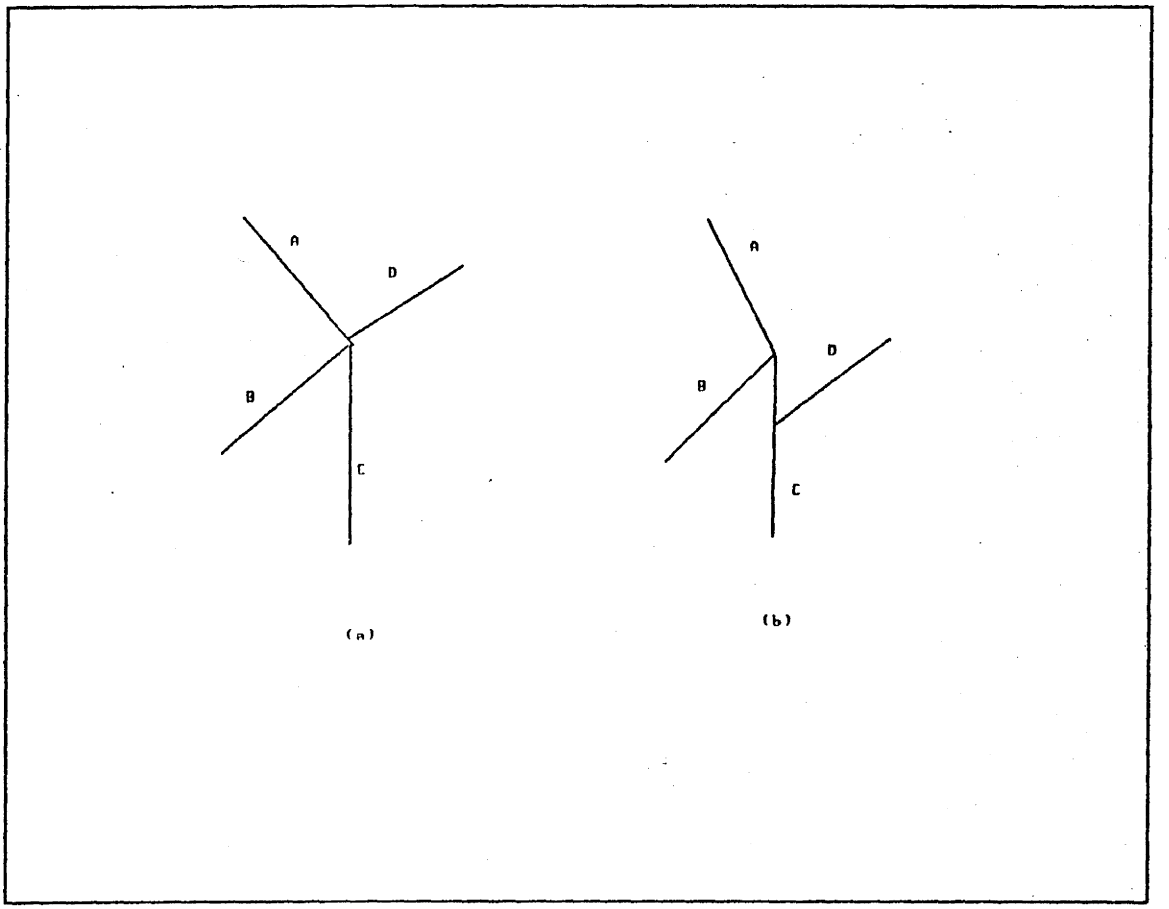


Figure 2: Break-up of an illusory Corner

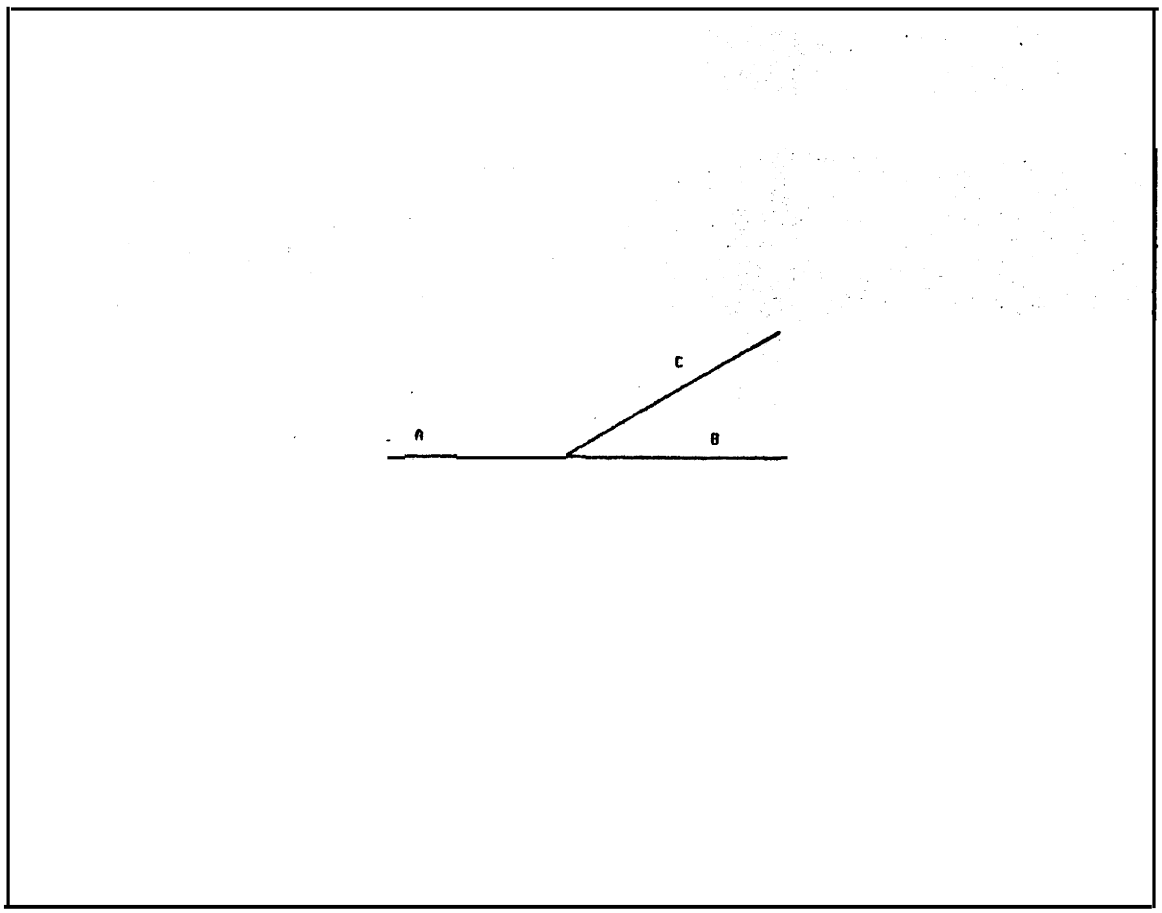


Figure 3: An ambiguous corner

1cm. Currently, the absolute location is not too accurate due to problems in generating accurate coordinate transforms. New calibration programs are under development which should result in greater accuracy, since the tracking routine is very accurate. The positional accuracy is now about 1 cm. at one metre. Relative depth resolution is considerably better, amounting to lmm. at one metre.

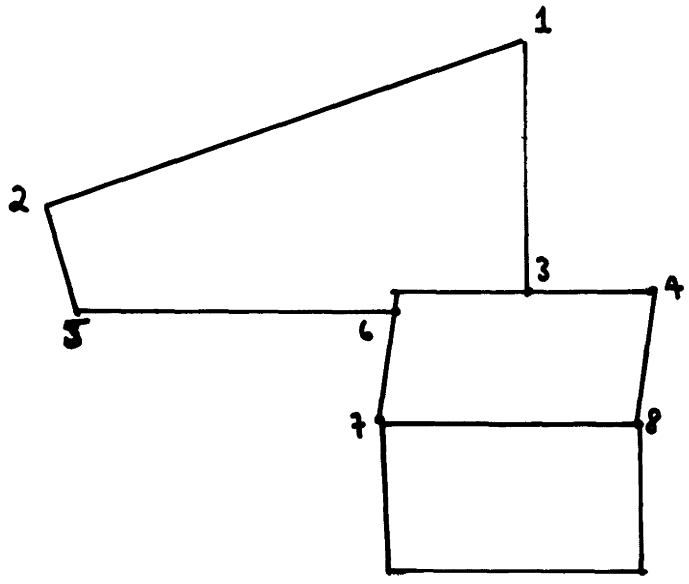
The outer level control of the program is currently written in SAIL, a dialect of ALGOL, for the PDP- 10. The correlation operator runs on a SPS-41, a small but very fast signal processor, under the control of a PDP-1 1/45. The initial image is copied from the 10 to the 11 and the 1 1 applies the variance operator to it. When an acceptable window is found, the gradient operator is applied to it and the various outputs of the operator for each point are returned to the 10, which finishes the corner finding process while the 11 continues looking for more potential features. After the features have been found a list of them is returned to the 11. Each image is sent to the 11 which then drives the SPS-41 to find the best correlation for each feature in the current pair of images and then signals the PDP-IO to transmit the next image, returning the final data to the PDP-10 after the last image. The SPS-41 can perform the correlation on a 16x 16 window in under 20 microseconds. Since the PDP-11 is faster than the PDP-IO, and is running only one program while the PDP- 10 is timesharing, the more parts of the program which are moved to the PDP-11, the faster the program will run. Currently, in a typical scene the program requires about 3-5 seconds to find 5-10 features, 0. 1 seconds per feature per image for the correlation and less than 0.1 seconds for calculating the depth of 6-12 features to a tolerance of 2.5 mm. When the hardware is finished, TV pictures will be transmitted directly to the PDP-11, which will then do all of the processing of the raw' data without the need to copy the images, which is relatively slow. Also, the variance and gradient operators will be recoded for the SPS-41 to greatly increase their speed.

7, Typical Results

Figures 4 and 5 show some typical results of running the program with ten views, each 0.5 deg. apart. Figure 4 is straightforward, but Figure 5 shows some examples of the difficulties that were mentioned above: points 4 and 6 are sliding points, which appear substantially different in the different views. Inaccuracies in the absolute distances (especially a general skewness in the picture) are due mainly to aberrations in the camera calibration model.

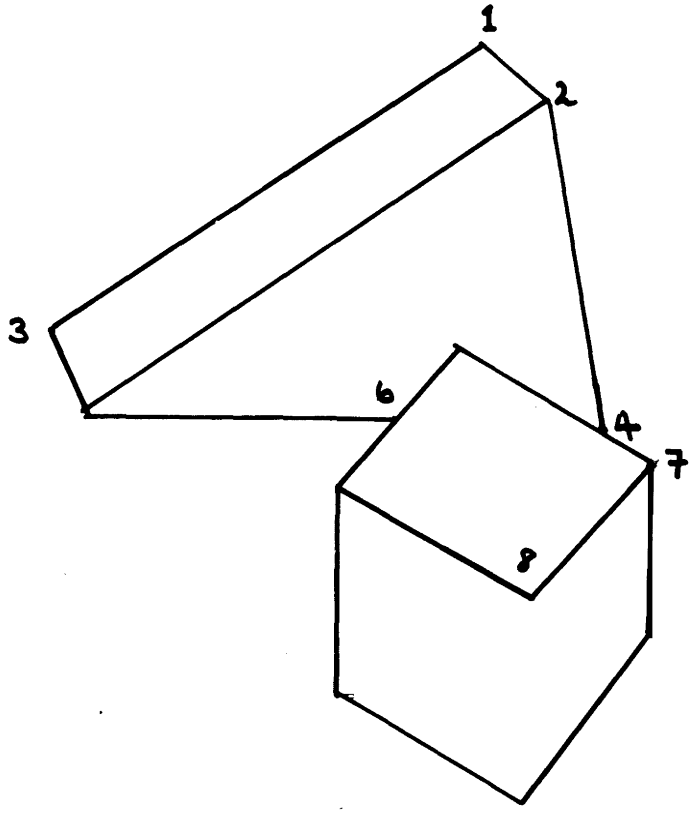
7, Conclusion

We have shown that it is possible to find the three-dimensional coordinates of a reasonable number of points within a table-top scene with a resolution acceptable for visual coordination and for disambiguation of complex features. The program that we have described works best on polyhedral objects, but is also usable on complex curved objects if the depth of some boundary points and/or conspicuous internal details is required. We envision using this program together with high level object seperation and recognition programs, and as an alternative to Nevatia's laser-ranging method. Further software and hardware development will allow the current program to be speeded up by an order of magnitude, at which point it will be ideally suited to visual feedback tasks in real time.



Point	Depth(ins.)
1	33.23
2	33.05
3	32.80
4	32.60
5	32.92
6	32.80
7	31.68
8	31.45

Figure 4.



Point	Depth(ins.)
1	32.03
2	31.23
3	33.08
4	32.64 (sliding)
5	32.73
6	30.08(sliding)
7	30.83
8	29.26

Figure 5.

Acknowledgments

We are very grateful to Ramakant Nevatia, who devloped an early version of the variance operator and the initial correlation programs from which our program has evolved; and to Botond Eross for programming the correlation operator for the SPS-41.

Bibliography

A ttneave,F.(1954) 'Some informational aspects of visual perception', *Psychol. Rev. 61:183.*

Bajcsy,R.(1972) 'Computer Identification of Textured Visual Scenes', *Stanford A.I. Memo No. 180.*

Baumgart,B.G.(1973) 'Image Contouring and Comparing', *Stanford AI Memo No.* 19'9.

Blakemore,C(1970) 'The Representation of S-dimensional visual space within the cat's visual cortex', *J.Physiol.Lond.,209:155.*

Bower,T.G.R.(1968) 'Morphogenetic Problems in Space Perception', *Proc. Assoc. for Research in Nervous and Mental Disease,* 48

Falk,G.(1970) 'Computer Interpretation of Imperfect Line Data as a three-dimensional scene',*Stanford A.I.Memo No. 132.*

Ganapathy,K.(1975) Forthcoming thesis, Stanford University.

Grape,G.R.(1973) 'Model based (Intermediate-level) Computer Vision', *Stanford A.I. Memo No. 201.*

Guzman,A.(1968) 'Decomposition of a visual scene into three-dimensional bodies', *Proc. Fall Joint Computer Conf. Was Aington.*

Hannah,M.J.(1974) 'Computer matching of areas in stereo images', *Stanford A.I. Memo No, 239*

Julesz,B.(197 1) *'The Foundations* of *Cyclopean* Perception', Chicago.

Nevatia,R.(1974) 'Structured Descriptions of complex curved objects for recognition and. visual memory', forthcoming thesis, Stanford University.

Nevatia,R.(1975) forthcoming.

Nevatia,R. & Binford,T.O.(1973) 'Structured Descriptions of Complex Objects', *Proc.* 3rd. *int. Joint Conf. on Artificial Intelligence, Stanford.*

Piaget,J. & Inhelder,B.(*1967) 'The Child's Conception of Space', New* York.

Waltz,D.(1973) 'Generating Semantic Descriptions froom drawings of scenes with shadows', ***MIT*** ***A.I.*** *Lab,* AI *TR-271.*

Y akimovsky,Y.(1974) 'Extracting depth information from a stereo pair', unpublished memorandum, Jet Propulsion Laboratory, Pasadena.