

## 選択的探索によるステレオ対応づけ

アーデシャ・ゴシュタスビ

ケンタッキー大学 情報工学科

新しいステレオ対応づけアルゴリズムを示す。まず画像中のエッジをその照度と照度勾配の強度および方向に基づいて分類する。次に画像中の同種類のエッジ同士の対応づけを行なう。第1段階として画像中の境界線同士の対応づけを行ない、第2段階として対応する境界線に属する点同士の対応づけを行なう。このアルゴリズムはエッジを選択的に探索するため、実行が高速化される。

## Stereo Correspondence by Selective Search

*Ardeshir Goshtasby*

Department of Computer Science, University of Kentucky

Lexington, Kentucky 40506-00270, USA

A stereo correspondence algorithm is described which first classifies edges in the images using their intensities, gradient magnitudes, and gradient directions, and then establishes correspondence between edges of the same type in the images. In this algorithm, correspondence is achieved in two steps. In the first step, correspondence is established between contours in the images, and in the second step, correspondence is established between points in corresponding contours. This algorithm carries out the search selectively in order to achieve fast speed.

## I. INTRODUCTION

Stereo is a method of determining the depths of objects in a scene using two images obtained from two similar cameras that are displaced horizontally by a small distance. In addition to stereo, there are other methods that can determine depth also. Generally, methods for depth perception are classified into active and passive.

Active methods use special lighting and determine depth either by time of flight or by triangulation. Methods based on time of flight use a scanning mirror to direct a beam of light at a point in the scene and measure the phase shift of the coaxially returned light to determine the time the light takes to travel from the scanner to an object and back. This time is then used to determine the depth of the object in the scene [16] [30]. In methods based on triangulation, a grid of light is projected to the scene, and by imaging geometry the depths of the grid points in the scene are computed [14] [17] [23] [29] [31]. Although computation of depth by active methods is reliable and fast, the obtained depth values are usually coarse in resolution, low in accuracy, short in range, and limited in scope. These methods are applicable to only indoor scenes where controlled lighting can be utilized.

Passive methods, on the other hand, do not need any special lighting as they use natural lighting in the scene to determine depth and therefore, are applicable to both indoor and outdoor scenes. Passive methods are based on principles such as stereo, focusing, texture gradient, perspective, and shading [18]. Among the passive methods, stereo produces the most accurate depth measurements. Depth perception by stereo requires determination of positions of corresponding points in the images and the camera geometry.

The most difficult problem in stereo is determining correspondence between points in the images. Zero-crossings of the second directional derivatives provide an abundance of points in the images that can be used in the matching. This method, however, produces too many edges and matching is very difficult. Since zero-crossings in an image are points where the gradient is either maximum or minimum, we should remove the minimum gradient edges (known as the phantom edges) because they do not truly represent edges in an image [5]. Small and noisy edges contribute to the matching error also. By removing the phantom edges and noisy edge contours from the images we obtain fewer edges, but the matching is still difficult because some edge contours appear different in the images even though they correspond to each other. Geometric difference between the images is unavoidable since the images are from different viewpoints of a 3-D scene. Some points may even be occluded in one image. An important problem in stereo is to identify the occluded points. For occluded points, no correspondence exists and matching should be avoided.

Different stereo correspondence algorithms have been developed. Many algorithms use one form of constraint or another to match the images. These algorithms produce a correspondence configuration that satisfies the constraints. In the following, we first review the existing stereo correspondence algorithms and then describe a new algorithm which uses information about the intensity, edge direction, edge magnitude, and shape of contours to establish correspondence between points in the images.

## II. PAST WORKS

Early algorithms in stereo correspondence use area-based correlation to determine correspondence between points in the images [7] [10] [22] [24]. These algorithms use intensities in the images, and by template matching determine correspondence between small areas in the images. Methods using area-based correlation have constantly been improved [8] [12], and in spite of some criticisms against them, they have been most successful in real applications. Many commercial systems have been implemented using the area-base correlation method [11] [26] [34] [35].

The greatest criticism about the area-based correlation is that when an area containing parts of two objects that are at different depths is used in the matching, that area will appear different in the images due to occlusion; therefore, a mismatch may be obtained. Another criticism is that the intensity at a point on a non-matte surface may appear different from two different viewpoints, and even though two areas truly correspond to each other, they may appear different, resulting in a mismatch. To reduce the effect of intensity difference between the images, gradients of the images, instead of the original intensities, have been used [3].

The difference between an area-based method and an edge-based method is that in an area-based method, information in a small neighborhood of a point is used in the matching, and although there may be dissimilarities between corresponding areas, there also are similarities that can be used in the matching. In an edge-based method, information in the neighborhood of a point is discarded and matching is carried out solely based on the information at the point. To compensate for the discarded information, various geometric constraints are used to assist the matching process [1].

Most edge-based methods use zero-crossings in the images as points for matching. Some algorithms carry out the matching in multiple scales. In these algorithms, matching is first carried out at a low resolution where matching is easy, and then the result is propagated to higher resolutions where matching is more difficult [6] [9] [20] [25] [28] [32]. Although the idea of multiple scales--which uses edges at different spatial frequencies for matching--is a good one, there is no method known that can determine true edges at certain frequencies in an image. Usually, to obtain edges at a given spatial frequency, the image is artificially blurred with an appropriate Gaussian filter. Gaussian filtering, however, blurs sharp edges as well as the blurred ones, and the obtained edges no longer represent edges of a certain spatial frequency in the original image. They are edges of a certain frequency in the blurred image.

Another problem with the multi-scale matching scheme is that edges from low frequency filters move from their true positions and propagation of that information to subsequent levels could mislead the matching process. In general, matching of edges obtained from a rather large filter should be avoided because many edges are displaced. Propagation of such inaccuracies to subsequent levels could mislead the system, unless a continuum of filter sizes is used in the process. A con-

tinuum of filter sizes, however, requires a tremendous amount of computation, which will make the stereo correspondence process slow and impractical.

In contrast to the area-based correlation methods which have used intensities in small areas for matching, so far the edge-based methods have used only the positions of the edges for matching. The objective of this study is to use different sources of information at the edge points such as the intensities, edge magnitudes, edge direction, and shape of connected edges to determine correspondence between points in the images. Before we describe our algorithm, we list below some of the desired properties that a stereo correspondence algorithm should have. Then we evaluate our algorithm against these criteria.

1. The algorithm should be able to identify occluded points in the images and avoid them in the matching.
2. It should produce global correspondence as well as local correspondence.
3. It should be applicable to any type of imagery.
4. It should match as many points in the images as possible.
5. It should avoid matching of noisy edges.
6. It should match points from the original images rather than the blurred images.
7. It should be fast.

The first item, occlusion, is very difficult to handle and many stereo correspondence algorithms simply ignore it. Hoff and Ahuja [13] describe a method that integrates feature matching and scene reconstruction in order to predict occlusion. Goshtasby [8] describes an algorithm that avoids matching of occluded points by carrying out a two-way search.

The second item assumes figural continuity and that assures correspondence between objects in the images in addition to correspondence between points in the images. Item number three makes the algorithm practical, so that it might be commercialized for general use. Algorithms that tend to work on only specific images, such as on random dot stereograms, will have very little practical value, because in real life it is not always possible to artificially add random texture to a scene.

Item number four makes it possible for the output of a stereo matcher to be usable in reconstruction and further analysis of the scene. The denser the depth map, the more accurate the reconstructed scene. Items number four and five are contradictory in nature because many edges could be due to noise and should be avoided in the matching. Noisy edges are usually weak and although some good edges are weak also, it is very difficult to distinguish between good edges and noisy ones. Therefore, a safe strategy is simply to discard the weak edges. It is better to have fewer matches than to have some possibly incorrect matches.

Item number six is required in order to obtain accurate depth measurements from the computed disparities. Methods that smooth the images first before carrying out the matching allow some edges to move from their true positions, and although correct matching may be obtained, the computed depth at the points will be incorrect. If image smoothing has to be performed before the matching, the smoothing filter size should be as small as possible in order to minimize movement of the edges.

Item number seven considers the practical aspect of an algorithm and requires that the algorithm be fast. A matching algorithm that requires a few hours on a supercomputer has no practical value and would be contradictory to the true spirit of computer vision which tries to automate visual tasks that are tiring or time consuming for humans.

In the following, an algorithm is described which first classifies edges in the images into various categories and then matches edges of the same category in the images. In this algorithm, search is carried selectively rather than exhaustively, to achieve high speed.

### III. EDGE CLASSIFICATION

Given a point in one of the images, we know that the corresponding point exists on the same epipolar line (row) in the other image. Often there are many candidate edges for a match. The difficulty in stereo correspondence is to find the correct match among the candidate ones. In this study, before carrying out the matching, we classify the edges using their intensities, edge magnitudes, and edge directions. Then, given an edge point in one of the images, we carry out the search among the edges that have about the same properties in the other image.

Stereo images are images of a scene obtained at the same instant from two similar cameras at slightly different viewpoints. Therefore, corresponding points should have close intensity values. Otherwise, either signal to noise ratio is small because of the low quality of the imaging system, or the point under study belongs to a specular and sharply curved surface which produces different brightness levels when viewed from different viewpoints. If a high quality imaging system is used and the scene does not contain sharply curved specular surfaces, we should expect that corresponding points produce about the same intensities. Similarly, corresponding points have close edge magnitudes and edge directions.

Classification of edges makes it possible to search for an edge only among a select set of edges that have about the same properties. Classification of the edges should take place with proper threshold values. For example, edge directions in the range  $\frac{\pi}{6}$  to  $\frac{5\pi}{6}$  and in the range  $\frac{7\pi}{6}$  to  $\frac{11\pi}{6}$  define two thresholding intervals. Edges falling in the ranges  $\frac{11\pi}{6}$  to  $\frac{\pi}{6}$  and  $\frac{5\pi}{6}$  to  $\frac{7\pi}{6}$  correspond to almost horizontal contours and should be avoided in the matching.

The gradient magnitude threshold values for classification of the edges may be obtained as follows. When we determine a histogram of the edges using their gradient magnitudes, if values at the 75, 70, 45, 40, and 10 percentiles are  $m_1, m_2, m_3, m_4,$  and  $m_5,$  respectively, then a set of intervals for classification of the edges could be: 1) values greater than  $m_2,$  2) values in the range between  $m_1$  and  $m_4,$  and 3) values between  $m_3$  and  $m_5.$  Edges with magnitudes less than  $m_5$  may correspond to noise and should be avoided in the matching. Overlapping intervals are used because the two images may not have exactly the same edge properties. Some edges may fall on one end of an interval in one image and fall on the

other end of the adjacent interval in the other image. Therefore, overlapping values should be used between adjacent intervals in order to allow classification of edges with properties falling on the thresholding boundaries. Intensities may be used to classify edges also. For example, those edges with intensities above the 45 percentile and those below the 55 percentile make two categories of edges.

If we classify the edges using two edge direction intervals, three edge magnitude intervals, and two intensity intervals, we obtain twelve categories of edges. If it happens that a category still contains a large number of edges, a finer set of thresholding intervals may be used. Note, however, that when a small thresholding interval is used, since the edge property along a contour may not be constant, the contour may break into two or more pieces, and because short contours are not used in the matching, a sparser disparity map will be obtained.

Once edges are classified into different categories, first correspondence is established between contours of the same category in the images, and then points in corresponding contours are matched. In the following, we describe the matching algorithm in more detail.

#### IV. THE ALGORITHM

In this section we describe an algorithm that determines correspondence between contours in the images. Two contours in the images cannot correspond to each other if they do not share at least one epipolar line (row). Among contours in the images that have common rows, the ones having the most similar properties are assumed to correspond to each other. Similarity between two contours is determined using their intensities, edge magnitudes, edge directions, and shapes.

Assuming two contours in the images share rows from  $r_1$  to  $r_2$ , we determine the dissimilarity between the contours by:

$$E = \frac{1}{(r_2 - r_1 + 1)(\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4)} \left[ \alpha_1 \sum_{i=r_1}^{r_2} |intL(i, j) - intR(i, k)| + \alpha_2 \sum_{i=r_1}^{r_2} |edmL(i, j) - edmR(i, k)| + \alpha_3 \sum_{i=r_1}^{r_2} |eddL(i, j) - eddR(i, k)| + \alpha_4 \sum_{i=r_1}^{r_2} |L(i, j - jmean) - R(i, k - kmean)| \right]. \quad (1)$$

In this formula,  $intL(i, j)$ ,  $edmL(i, j)$ ,  $eddL(i, j)$  show the intensity, edge magnitude, and edge direction at point  $(i, j)$  on a contour in the left image. Similarly,  $intR(i, k)$ ,  $edmR(i, k)$ ,  $eddR(i, k)$  show the intensity, edge magnitude, and edge direction of the corresponding point on the same row in the right image.  $jmean$  and  $kmean$  show the column averages of the common segments of the contours under consideration in the left and right images, respectively.  $L(i, j - jmean)$  shows the difference of column number of the contour in the left image at row  $i$  and  $jmean$ , and  $R(i, k - kmean)$  shows the difference of column number of the corresponding contour in the right image at row  $i$  and  $kmean$ .  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ , and  $\alpha_4$  show the relative importance of the intensity, edge magnitude, edge direction, and shape of contours in computation of the dissimilarity measure. In relation (1), by changing the value of index  $i$  from  $r_1$  to  $r_2$ , we follow two corresponding contours and determine their weighted differences with respect to intensity, edge magnitude, and edge direction point by point. Note that the dissimilarity measure is normalized with respect to the length of the common segment between the contours. Otherwise, shorter contours will produce smaller dissimilarity measures. The fourth term in relation (1) shows the shape dissimilarity between two contours. In this measurement, the two contours are aligned and then the cumulative distance between corresponding points in the two contours is determined.

Since small and noisy contours could cause a mismatch, we consider two contours as candidates for a match if they share more than a certain number of rows, i.e.  $r_2 - r_1 > d$ . A very large threshold value avoids matching of many good segments, while a very small threshold value may mistakenly match a short and noisy contour with a long and authentic contour. We will find the proper value for  $d$  in Section V.

Given a contour in the left image, we match that contour with contours in the right image that have at least  $d$  rows in common. For each candidate match, we determine the dissimilarity between the contours, and the match that produces the smallest dissimilarity measure is taken to be the correct one.

If the camera axes are parallel and aligned so that points in infinity have zero disparity, then when two points  $L(i, j)$  and  $R(i, k)$  correspond to each other we should have  $j \geq k$ . This shows that a point corresponding to point  $L(i, j)$  lies on a contour with column numbers smaller than or equal to  $j$ . If the maximum disparity of two images is known to be, say  $D$ , obtained for example from the focusing information of the cameras [15], we should look for contours with column numbers in the range from  $j - D$  to  $j$  for a match. This reduces the number of candidate matches and therefore, increases the matching reliability. Of course, if the maximum disparity between the images is not known, the entire set of contours in the right image should be searched for a match.

If for a given contour in the left image, we do not find a contour in the right image in the specified range of columns, we conclude that the edge is occluded. Along with correctly identifying occluded contours, this algorithm could identify a non-occluded contour as an occluded one if the contour corresponding to the given contour in one image was not extracted in the other image. This could happen when in the classification process, a contour is segmented into little pieces and the pieces are removed before matching is carried out. On the other hand, if the contour is really occluded from one image, but there are some contours present in that image in the range where the search is being carried out, a mismatch will be obtained. As we will see, however, the number of such mismatches may be arbitrarily reduced by classifying the edges into arbitrarily finer categories of contours.

Once correspondence is established between contours in the images, correspondence between points on the contours is easy because points on corresponding contours that fall on the same row correspond to each other.

Overall, the algorithm can be summarized into the following six steps:

1. Determine the zero-crossings of the images with a small Gaussian operator, such as  $\sigma=1$ .
2. Remove the phantom edges from the images.
3. Remove noisy contours of length less than  $l$  (say 10) pixels from the images.
4. Determine the intensity, edge magnitude, and edge direction of the remaining zero-crossings.
5. Classify the edges using their intensities, edge magnitudes, and edge directions.
6. For each category of edges,
  - a) Find the correspondence between contours using the dissimilarity measure of (1).
  - b) Find the correspondence between points in corresponding contours.

Next, we investigate the sensitivity of the parameters of the algorithm to the computed disparities.

## V. RESULTS AND DISCUSSION

In this section we investigate the sensitivity of the algorithm to its parameters. We vary these parameters and determine how the behavior of the algorithm changes. Let's denote the intensity, edge magnitude, and edge direction threshold values by  $I$ ,  $M$ , and  $D$ , respectively; the minimum number of common rows required for two contours to be a candidate match by  $d$ ; and the degree of importance of the intensity, edge magnitude, edge direction, and shape of edge contours in determination of the dissimilarity measure by  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ , and  $\alpha_4$ , respectively. Next, we investigate the influence of these parameters on the reliability of the obtained correspondences.

$d$  shows the minimum number of rows that two contours should share in order to be candidates for a match. Due to the classification error, a contour in one image may be segmented into two or more pieces in the other image, and it is still necessary to find a correspondence between the segments. When  $d$  is too small, a very small segment from one image may be matched to many larger segments in the other image, and since a small segment contains very little information, a mismatch may be obtained. On the other hand, when  $d$  is too large, many good contour segments will not be selected for matching.

Since we allow edge directions as small as  $\frac{\pi}{6}$ , a contour of length ten and direction  $\frac{\pi}{6}$  will occupy  $10\sin\frac{\pi}{6}$  or 5 rows.

Therefore, when the preprocessor removes all contours of length less than or equal to ten, the smallest value that we can allow for  $d$  is five. As  $d$  increases, contours with a higher number of common rows in the images are used in the matching and therefore, the matching will become more reliable. Increasing  $d$ , however, reduces the number of contours that are matched and therefore a sparser disparity map will be obtained. A value from five to ten seems to be a reasonable compromise.

Parameters  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ , and  $\alpha_4$  show the influences of the intensity, edge magnitude, edge direction, and shape of contours in determining the dissimilarity between two contours. When a scene contains specular and curved surfaces, the influence of intensity should be reduced, because specular surfaces look different when viewed from different viewpoints. Also, if two dissimilar cameras or digitization processes are used to obtain the images, since intensities in the images may appear different,  $\alpha_1$  should be given a smaller weight compared to the other parameters.

The difference between edge directions in corresponding contours is mostly due to the geometric difference between the images. In scenes with sharp depth variations  $\alpha_3$  should be given a smaller value than when the scene contains smoothly varying surfaces since dissimilar contours may be obtained. Note that edge direction should be measured with degrees rather than radians in order to maintain comparable weights for dissimilarities computed from edge directions compared to edge magnitudes. Edge magnitude varies with both intensity and geometric difference between the images.

Shape of contours is another measure which can be used to determine the correspondence between contours. If the geometric difference between the images is not large (distance between the cameras is small and depth in the scene varies smoothly),  $\alpha_4$  should be given a higher weight compared to  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$ . Again, note that the term corresponding to  $\alpha_4$  in (1) produces small values corresponding to the other terms. This is because the cumulative distance between two dissimilar contours, when the contours are aligned at their centers of gravity, is much smaller than the differences of the contours using their intensities, edge magnitudes, or edge directions. To make the four terms in (1) to contribute equally to the dissimilarity measure, we should allow  $\alpha_4$  to be an order of at least ten greater than  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$ .

Although the behavior of the algorithm changes by varying the parameters of the algorithm, the algorithm stays stable over a long range of the parameters. In an experiment, we even allowed three of the four parameters to be zero and the algorithm still was successful in finding the correct matches. Only 2% error was obtained when only edge magnitude was used. Once the edges are classified into different categories, very little information is needed to establish correspondence between the contours.

Parameters  $I$ ,  $M$ , and  $D$  classify the contours into various categories. If edges are finely classified, a more accurate matching will be obtained. However, since fine classification of contours may cause a contour to be segmented into two or more fragments, each belonging to a different class, and since small fragments are not used in matching, fewer matches are generated, resulting in a sparser depth map. Coarsely classifying the edges results in more matched edges but also in a greater number of mismatches.

In Figures 1 and 2 we show the results of applying this algorithm on two sets of images. The parameters used in both cases were:  $I_1 \leq 55\%$ ,  $I_2 \geq 45\%$ ,  $10\% \leq M_1 \leq 60\%$ ,  $50\% \leq M_2$ ,  $\frac{\pi}{6} \leq D_1 \leq \frac{5\pi}{6}$ ,  $\frac{7\pi}{6} \leq D_2 \leq \frac{11\pi}{6}$ ,  $\alpha_1 = \alpha_2 = \alpha_3 = 1$ ,  $\alpha_4 = 10$ , and  $d = 5$ .

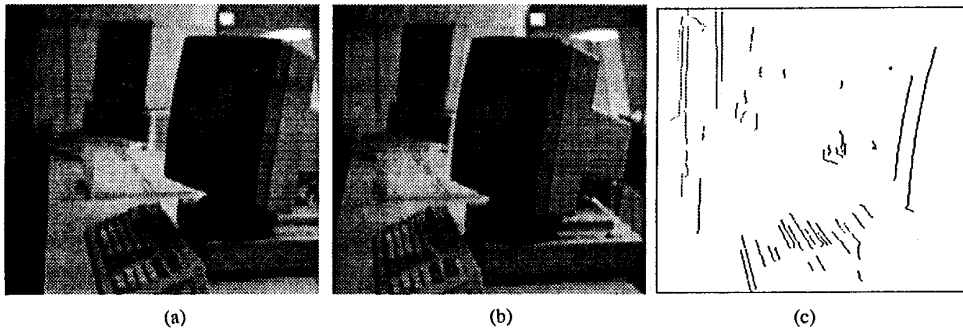


Figure 1. (a) and (b) are the left and right images of a stereo pair, and (c) shows the computed disparities. Darker lines show larger disparities.

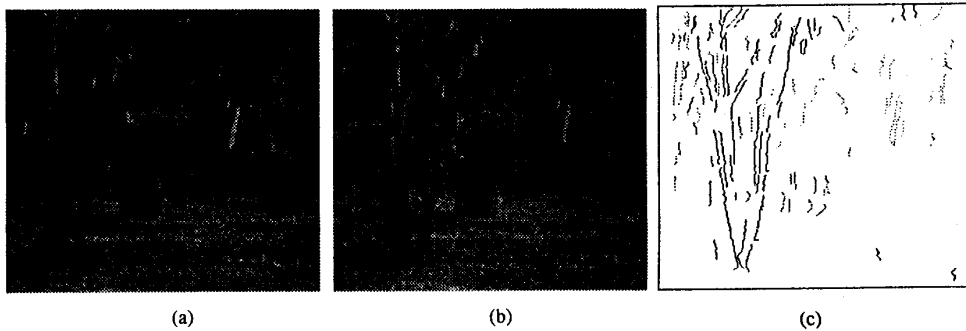


Figure 2. (a) and (b) show stereo images of an outdoor scene, and (c) shows their disparities.

We next evaluate the performance of the algorithm against the criteria given in Section II.

This algorithm establishes global correspondence as well as local correspondence between images. Once correspondence is determined between points in the images, we know that correspondence between contours has already been established. This higher level information which is obtained for free, may be used in analysis of the scene, for example for object recognition.

There is no restriction on the type of images that this algorithm can handle. In general, though, the algorithm performs better on man-made scenes where the images contain mostly long contours. In images of highly textured outdoor scenes (such as the one in Figure 2), since a larger number of short contours are obtained, and since short contours are not used in the matching, a sparse depth map will be obtained. If short contours are used in the matching a dense depth map will be obtained, but at the same time the probability of mismatch increases.

Matching of edges due to noise may be avoided by removing the weak edges because edges obtained from noise are usually weak. In the above experiments, 10% of the smallest gradient edges were removed from the images. The lower bound of parameter  $M_1$  is used to remove edges in the images obtained from noise. If noise is known to be strong, a larger percentage of the edges should be removed from the images.

Item number four works in the reverse of items number one and five. In other words, if we allow our algorithm to match as many points in the images as possible, we increase the probability of finding matches for noisy and occluded edges. On the other hand, if we try to avoid matching of occluded and noisy contours, we have to sacrifice matching of many good but weak edges also. This algorithm allows the user to select the degree of compromise between items one, four, and five by choosing a proper set of input parameters. For example, if the lower bound of parameter  $M_1$  is increased, a sparser depth map will be obtained but matching of edges due to noise will be avoided.

The proposed algorithm uses zero-crossing edges that are obtained from a small Gaussian operator. A small operator is used in order to minimize artificial blurring of the images, and in turn provide highly accurate edge positional values. A small operator will produce a large number of edges, but that is no problem for our algorithm because it can divide the edges into different categories and match one category of edges at a time.

The proposed algorithm is very fast compared to many well known algorithms [2] [4] [6] [19] [20] [26] [27]. Once zero-crossings of the images are determined, determination of intensity, edge magnitude, and edge direction at the zero-crossings takes about one minute, classification of the edges into different categories takes about another minute, and matching of the edges takes about two more minutes, for images of size  $240 \times 256$  on a MicroVax computer. Although this response time is not still real-time, a real-time response may be achieved if portions of this algorithm are implemented in

hardware, or a more high-speed computer is used.

## VI. CONCLUSION

A stereo correspondence algorithm was described which is based on the observation that when a small number of edges exist in two images, a reliable and fast matching will be obtained. Therefore, this algorithm first divides edges in the images into various categories and then matches edges in the same category. Since an arbitrarily large number of categories may be chosen, the number of edges that are to be matched at one time can be arbitrarily reduced. This mechanism provides a highly reliable and fast matching.

The algorithm has a number of parameters that can be changed to control its behavior. The user has the ability to choose the number of edge categories, the minimum length of a contour, and the weakest edge that should be used in the matching. Although these parameters somewhat change the behavior of the algorithm, the behavior of the algorithm stays the same over a broad range of the parameters. Therefore, the parameters may be fixed at appropriate values (such as the ones used in Figures 1 and 2) and the algorithm be applied to a broad range of images without having to change the parameter values.

In this algorithm, high computational speed is achieved by selectively carrying out the search. When an edge is taken from one image, a search is carried out in the other image only among a selected set of edges that have about the same properties. This algorithm can determine the correspondence between stereo images of size 240×256 in only a few minutes on a MicroVax computer.

## REFERENCES

- [1] Arnold, R. D. and T. O. Binford, "Geometric Constraints in Stereo Vision," *SPIE Image Processing for Missile Guidance*, Vol. 238, 1980, 281-292.
- [2] Ayache, N. and B. Faverjon, "Efficient Registration of Stereo Images by Matching Graph Descriptions of Edge Segments," *Int. J. Computer Vision*, 1987, 107-131.
- [3] Baker, H. H. and T. O. Binford, "Depth from Edge and Intensity Based Stereo," *Int. J. Conf. Artificial Intelligence*, 1981, 631-636.
- [4] Barnard, S. T. and W. B. Thompson, "Disparity Analysis of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 4, 1980, 333-340.
- [5] Clark, J. J., "Authenticating Edges Produced by Zero-Crossing Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 1, 1989, 43-57.
- [6] Eastman, R. D. and A. M. Waxman, "Using Disparity Functionals for Stereo Correspondence and Surface Reconstruction," *Computer Vision, Graphics, and Image Processing*, Vol. 39, 1987, 73-101.
- [7] Gennery, D. B., "A Stereo Vision System for an Autonomous Vehicle," *5th Int. J. Conf. Artificial Intelligence*, 1977, 576-582.
- [8] Goshtasby, A., "A Refined Technique for Stereo Depth Perception," *Proc. Workshop on Computer Vision: Representation and Control*, 1984, 125-129.
- [9] Grimson, W. E. L., "A Computer Implementation of a Theory of Human Stereo Vision," *Proc. R. Soc. Lond.*, B-292, 1981, 217-253.
- [10] Hannah, M. J., *Computer Matching of Areas in Stereo Images*, Ph.D. Thesis, July 1974, Computer Science Department, Stanford University.
- [11] Hannah, M. J., "Bootstrap Stereo," *Proc. Image Understanding Workshop*, 1980, 201-208.
- [12] Hannah, M. J., "Test Results from SRI's Stereo System," *Proc. Image Understanding Workshop*, 1988, 740-744.
- [13] Hoff, W. and N. Ahuja, "Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 2, 1989, 121-136.
- [14] Hu, G. and G. C. Stockman, "3-D Surface Solution Using Structured Light and Constraint Propagation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 4, 1989, 390-402.
- [15] Jarvis, R. A., "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 5, No. 2, 1983, 122-139.
- [16] Jarvis, R. A., "A Laser Time-of-Flight Range Scanner for Robotic Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 5, No. 5, 1983, 505-512.

- [17] Kak, A. C., K. L. Boyer, R. J. Safranek, and H. S. Yang, "Knowledge-Based Stereo and Structured Light for 3-D Robot Vision," in *Techniques for 3-D Machine Perception*, A. Rosenfeld (Ed.), Elsevier Science Publishers, North Holland, 1986, 185-218.
- [18] Levine, M. W. and J. M. Shefner, "Depth Perception," in *Fundamentals of Sensation on Perception*, Addison-Wesley Publishing Co., 1981, 250-274.
- [19] Marr, D. and T. Poggio, "Cooperative Computation of Stereo Disparity," *Science*, Vol. 194, 1976, 283-287.
- [20] Marr, D. and T. Poggio, "A Computational Theory of Human Stereo Vision," *Proc. R. Soc. Lond.*, B-204, 1979, 301-328.
- [21] Marr, D., G. Palm, and T. Poggio, "Analysis of a Cooperative Stereo Algorithm," *Biological Cybernetics*, Vol. 28, 1978, 223-239.
- [22] Moravec, H. P., "Rover Visual Obstacle Avoidance," *Proc. Int. J. Conf. Artificial Intelligence*, 1981, 785-790.
- [23] Mundy, J. L. and G. B. Porter III, "A Three-Dimensional Sensor Based on Structured Light," in *Three-Dimensional Machine Vision*, Takeo Kanade (Ed.), Kluwer Academic Pub., 1987, 3-61.
- [24] Nevatia, R., "Depth Measurement by Motion Stereo," *Computer Graphics and Image Processing*, Vol. 5, 1976, 203-214.
- [25] Nishihara, K., "Practical Real-Time Imaging Stereo Matcher," *Optical Engineering*, Vol. 23, No. 5, 1984, 536-545.
- [26] Panton, D. J., "A Flexible Approach to Digital Stereo Mapping," *Photogrammetric Engineering and Remote Sensing*, Vol. 44, No. 12, 1978, 1499-1512.
- [27] K. Prazdny, "Detection of Binocular Disparities," *Biological Cybernetics*, Vol. 25, 1985, 93-99.
- [28] Quam, L. H., "Hierarchical Warp Stereo," in *Readings in Computer Vision*, M. A. Fischler and O. Firschein (Eds), 1987, 80-86.
- [29] Rioux, M., "Laser Range Finder Based on Synchronized Scanners," *Applied Optics*, Vol. 23, No. 21, 1984, 3837-3844.
- [30] Sampson, R., "3-D Range Sensor-Phase Shift Detection," *Computer*, Vol. 20, No. 8, 1987, 23-24.
- [31] Stockman, G. C., and S-W Chen, *Experiments in 3-D Data Acquisition Using A Camera and Projector*, TR No. 85-023, Michigan State University, 1985.
- [32] Terzopoulos, D., "Multilevel Computational Processes for Visual Surface Reconstruction," *Computer Vision, Graphics, and Image Processing*, Vol. 24, 1983, 52-96.
- [33] Thrope, C. E., "An Analysis of Interest Operators for FIDO," *Proc. Workshop on Computer Vision: Representation and Control*, 1984, 135-139.
- [34] Vincent, R. K., M. A. True, and D. V. Roberts, "Automatic Extraction of High Resolution Elevation Data Sets from Digitized Aerial Photos and Their Importance for Energy Mapping," *Proc. NCGA Mapping and Geographic Information Systems*, 1987, 203-210.
- [35] Yakimovsky, Y., and R. Cunningham, "A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras," *Computer Graphics and Image Processing*, Vol. 7, 1978, 195-210.