Recovering a boundary-level structural description from dynamic stereo

Arun P Tirumalai*, Brian G Schunck and Ramesh C Jain

We present a stereo algorithm to recursively compute a boundary-level structural description of a static scene, from a sequence of dynamic stereo images. This algorithm is based on connected line segments as the basic match primitive, which yields a description composed primarily of boundaries of objects in the scene. The algorithm is integrated into a dynamic stereo vision system to compute and incrementally refine such a structural description recursively, using belief measures. The approach is illustrated with a real dynamic stereo sequence.

Keywords: dynamic stereo, sensor fusion, environment modelling

Dynamic stereo vision deals with the problem of processing sequences of stereo images of a scene, acquired from different viewpoints, to recover the underlying structure. Dynamic stereo is useful to construct a complete map of the environment as only a portion of the actual environment is visible from each viewpoint. In addition, there is usually an overlap between the portions of the environment visible from two successive viewpoints. It is then feasible to utilize a prediction-verification approach to combine the individual estimates of features visible from both viewpoints to obtain a more accurate estimate.

In this paper we present a stereo algorithm to compute a boundary-level structural description of a scene, from a sequence of stereo images. This algorithm, based on connected line segments as the basic match primitive, is integrated into a dynamic stereo vision system, and is used to incrementally refine such a structural description using belief measures.

Artificial Intelligence Laboratory, University of Michigan, Ann Arbor, Michigan 48109-2110, USA
*Currently with the Department of Computer Science, University of South Carolina, Columbia SC 29210, USA

The problem of recovering structure from stereo vision has been studied extensively for several years. Various algorithms have been proposed based on matching points[1,2], line segments[3,4] and surfaces[5]. A comprehensive review of the structure from stereo problem is presented by Dhond and Aggarwal[6]. However, a satisfactory solution to the problem has not been discovered so far. A recent trend in stereo vision research is a move towards dynamic vision[7] where the goal is to combine multiple measurements, which are invariably noisy, to obtain reliable depth estimates. Related research is presented elsewhere[8–12].

The use of high-level features in stereo vision has been less popular due to the complexity of extracting and matching such high-level features from images. We depart from the traditional approach of attempting to resolve all ambiguities in matching in each stereo pair of images, based on global constraints such as 'locally smooth disparities'[1]. We utilize a notion of a belief assignment[13] to each stereo match, reflecting its reliability. We attempt to resolve ambiguities by enhancing the beliefs of the correct matches by recursively processing a stereo sequence.

COMPUTATIONAL FRAMEWORK

Images of the environment are acquired using a lateral stereo camera pair mounted on a mobile robot which moves under computer control along a predetermined path. The objective is to recursively obtain a boundary-level (wire-frame) representation of the depth map of the environment with respect to the initial camera position. A coordinate system placed at the optical centre of the right camera and aligned with the image plane is treated as the inertial coordinate system. The computations involved in our framework are detailed in block diagram form in Figure 1. The computations proceed as follows:

1. For each acquired stereo pair, the basic processing involves line-segment detection, segment triplet (connected triples of segments) detection

0262–8856/92/003191–06

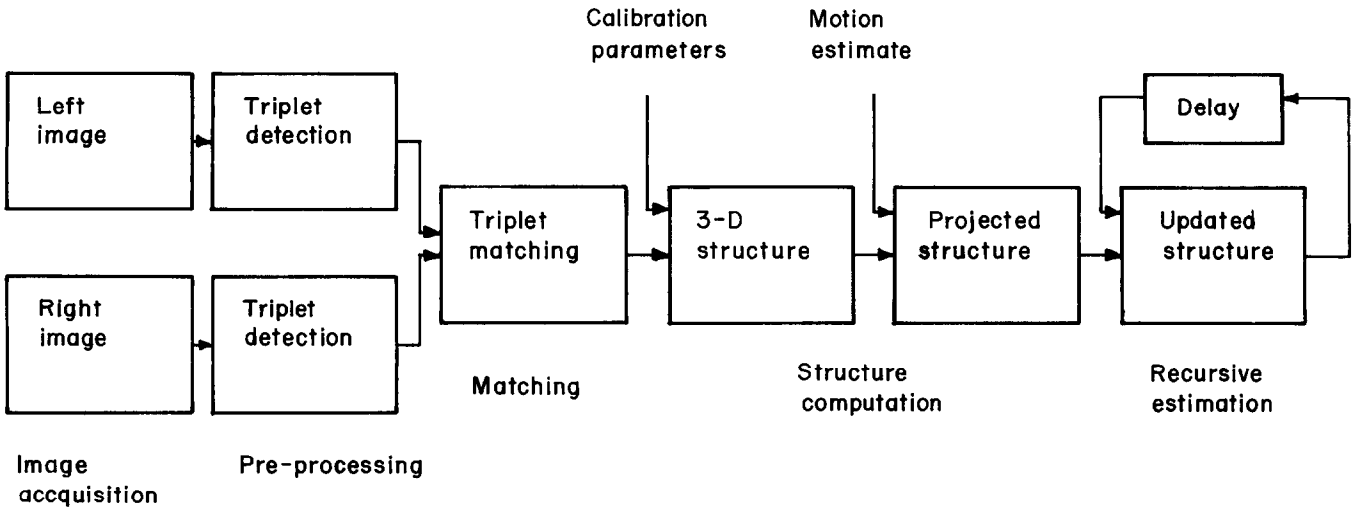

Figure 1. Computational framework

in each image, and then triplet matching between the left and right camera images. For each matched triplet in the image plane, we can compute a triplet in 3D space with respect to camera viewpoint.

2. By matching corner points on matched triplets between two stereo pairs acquired from two viewpoints, the camera motion between the two viewpoints is recovered.
3. The recovered motion is used to project the locations of the matched triplets from the second viewpoint to the first.
4. The matched triplets from the first viewpoint and the projected matched triplets from the second viewpoint are then fused, and the wire-frame representation with respect to the first viewpoint is updated.
5. Steps 1–4 are performed recursively as additional stereo pairs are acquired.

MATCHING STEREO PAIRS

The various computational stages in matching individual stereo pairs are segment detection, triplet extraction, and triplet matching. Each of these stages is now described in detail.

Segment detection

Edges are detected in each image using the Sobel operator. For each edge point, an approximate orientation is computed from the gradient value and a label, in the range 1–8 (corresponding to 0–360 degrees), is assigned. Next, an 8-connected component algorithm is used to group together edge points with the same label[14]. Finally, a continuous representation for each such connected component is computed in the form $aX+bY+c=0$, using linear regression. Several parameters are associated with each detected segment including the line segment parameters (a,b,c), the endpoints of the segment, the segment label, orientation angle, length, and the pixel coordinates of the bounding rectangle.

Triplet detection

One of the problems with dealing with line-segments as the basic matching primitive is that a one-to-one match between the end-points of the detected segments does not always exist. We circumvent this problem, to an extent, by choosing to match connected triples of segments. For each such triplet, there exists a central segment which forms a corner at each end-point with another segment as illustrated in Figure 2. The region enclosed by each such triplet (when the triplet bounds a convex region) typically lies on a surface in the scene. We seek matches between such triplets, primarily seeking a one-to-one match between the central segments of each triplet.

The first step in detecting triplets is to compute segment neighbourhoods, i.e. the list of all segments neighbouring a given segment, thus forming potential corners with it. We utilize bucketing techniques[3] for this task. The next step involves the computation of the

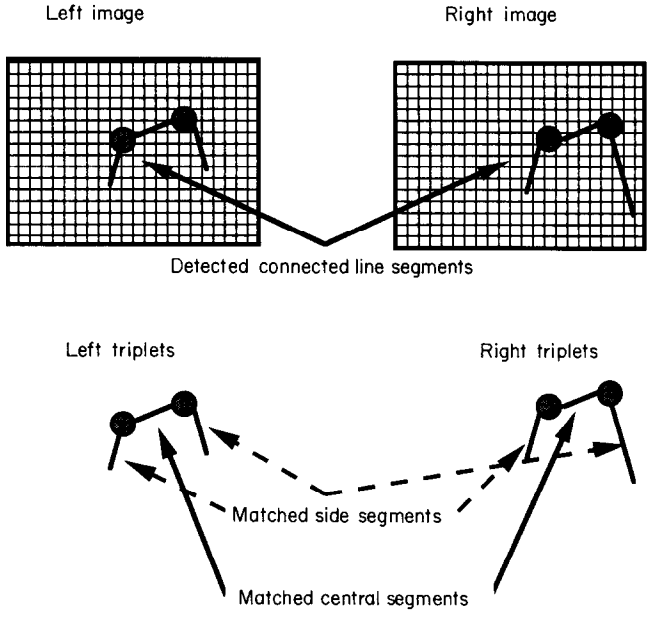

Figure 2. Segments triplets as the basic matching primitives

list of corners each segment forms with each of its neighbours. Following this step, we seek segments forming at least one corner near each end-point with a neighbour which directly yields the list of segment triplets. Such segments form the central segment in the triplet structure we wish to detect and match. The two segments forming corners at each end-point of the central segment correspond to the side segments of this triplet.

Matching triplets

For each triplet in one image, the mid-point of the central segment is first computed. Starting with the location of this mid-point in the other image, all windows within a rectangular region, bounded by a maximum allowed vertical offset and a maximum allowed disparity range, are searched for intersecting segments. Each such intersecting segment which happens to be the central segment of a triplet in the second image is a potential triplet match. Each such potential triplet match is verified based on similarity of labels of the segments forming the triplets, the structure of the triplets, and the vertical offsets between the corner matches.

For each triplet match, the central segment has both its end-points matched. The two side-segments have only one of the corners matched. The other corner is computed by first picking the end-point of the shorter side-segment and finding a corresponding location on the longer side-segment by scanning along the epipolar line. After this step, the side-segments also have disparities assigned to two points lying on them.

Belief assignments

For each matched triplet, we assign a belief of 0.6 to the central segment as it is considered to have been reliably matched. The two side-segments are assigned beliefs of 0.3. We actually retain multiple triplet matches if they exist. We attempt to resolve these ambiguities by assimilating additional stereo data. Each segment that is correctly measured in more than one viewpoint has its belief assignment enhanced. After all the stereo data has been assimilated, a threshold based on the resultant belief values is used to resolve the ambiguities in individual segment matches.

The final description of the underlying structure is in the form of segments in the image plane with known disparities at their end-points. This description can directly be transformed to 3D coordinates using the camera calibration parameters. Each segment also has a degree of belief assigned to it reflecting the reliability with which it has been matched.

REGISTERING STEREO IMAGES

In order to combine stereo data acquired from two viewpoints, it is necessary to relate the individual stereo measurements to a common coordinate system. In our experiments, the camera (which is mounted on a mobile robot) motion between successive viewpoints is approximately known. For each matched triplet, the end-points (corners) of each matched central segment can be considered to have been reliably matched. The 3D coordinates of these matched corner points can be computed from the camera calibration parameters. We attempt to establish a one-to-one correspondence of such corner points between two stereo frames, and then attempt to refine this motion estimate using the motion estimation algorithm described by Horn *et al.*[15] on these matched corner points.

ASSIMILATING STEREO DATA

In our experiments, we affix a camera centred coordinate system with respect to stereo frame 1. All measurements (new structural descriptions composed of connected segments) made in subsequent frames are first transformed to this coordinate system, using the recovered motion, and then assimilated with the existing structural description. Basically, we seek matches between segments on an individual basis. For those segments for which matches are found, the beliefs are updated (enhanced). The unmatched segments are also retained without having their beliefs enhanced, with the objective of finding matches in a subsequent frame. After the stereo data acquisition has been completed, only the segments with beliefs higher than a threshold are retained which yields the final structural description.

The assimilation of beliefs is based on Bernoulli's rule[13] of combination. Bernoulli's rule provides a method for combining two mass distributions (beliefs assigned to propositions of interest) $M(B_1)$ and $M(B_2)$ obtained from two independent sources to produce an update mass distribution $M(B_3)$ that represents a consensus opinion of the two sources. Mathematically, it is expressed as:

$$M(B_3) = M(B_1) + M(B_2) - M(B_1)M(B_2) \qquad (1)$$

EXPERIMENTAL RESULTS

In this section we present several results from experiments conducted with a laboratory sequence of stereo images.

Stereo image sequence acquisition

Our basic active vision system consisted of a camera mounted on a pan-tilt-translate (PTT) head. The PTT head in turn was mounted on a mobile robot which allowed image data acquisition from multiple viewpoints. We constructed an environment in our laboratory consisting of common, mostly polyhedral objects. Our objective was to recover a boundary-level structural description of this environment. A total of ten stereo pairs was collected by translating the robot directly forward along the optical axis of the camera 10 cm at a time, a total of about 100 cm. A parallel stereo camera configuration was used here. A camera centred coordinate system in frame 1 (placed at the optical centre of the right camera with X-Y axes aligned with the image plane) was chosen as the inertial coordinate system. Depth measurements made from frames 2–10 were first transformed to this inertial coordinate system, and then assimilated into the

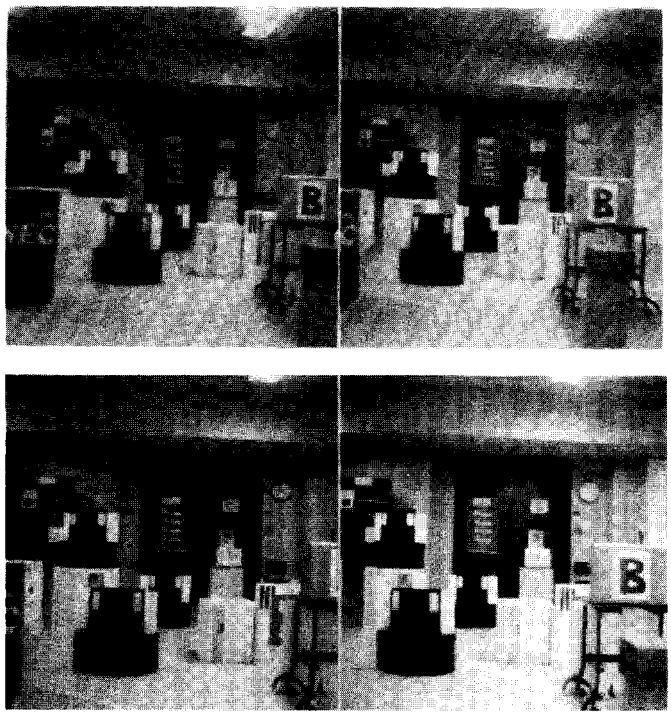

Figure 3. Left and right stereo camera images in frames 1 (top) and 10 (bottom)

current description in a recursive fasion. The laboratory scene is depicted in Figure 3, wherein the left and right stereo camera images from frames 1 and 10 are shown.

Matching individual stereo pairs

The first processing stage involved line segment feature detection. This involved edge detection (based on the Sobel operator), edge-grouping, and linear regression as described earlier. For each segment, the orientation, the length, and end-point locations were computed. These parameters were later used to hypothesize segment matches. Figure 4 depicts the gradient magnitude image (scaled to the range 0–255), along with the computed line segments from the right camera image in frame 1. The line segments shown here were reconstructed from the regression parameters which yielded a continuous representation for the segments.

The next processing stage involved the detection of

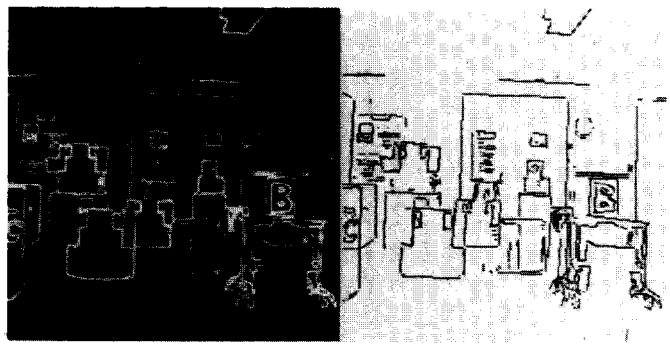

Figure 4. Sobel gradient magnitude (left) and the detected line segments (right) in the right camera image in frame 1

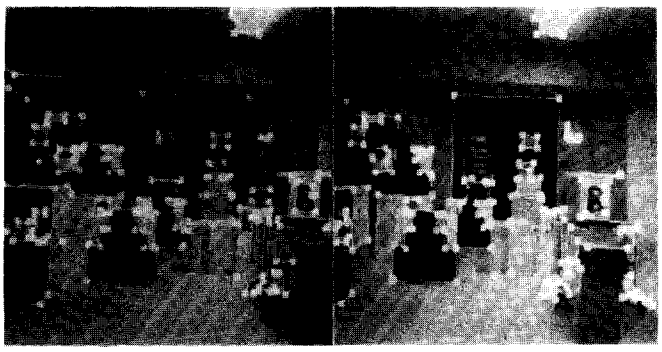

Figure 5. Connected line segments and the corresponding corners in the stereo pair in frame 1

connected line segments in each stereo pair, which was based on bucketing techniques as described earlier. Figure 5 depicts the connected line segments and the corresponding corners from the stereo pair in frame 1. Segment triplets were then computed from these connected segments.

At this point, triplets of connected segments were available in both the left and right camera images. Each triplet consisted of a central segment with a corner at each end formed with a neighbouring segment. The bucket lists (list of pixels intersecting a given segment), computed in the triplet detection stage, were used once again to hypothesize potential triplet matches as explained earlier. Figure 6 depicts the matched triplets between the left and right camera images in frame 1. We retained multiple triplet matches, if they existed. We assigned beliefs of 0.6 and 0.3 to the central segment and the two side-segments, respectively, for each triplet match.

From this point, we processed each segment on the matched triplets somewhat independently. A one-to-one match existed for two corners on either side of the central segment. For this central segment we retained the X, Y image coordinates of each matched corner, and also the disparity, computed from the triplet match. The two side-segments had only one corner matched. The disparity of the other end-point was computed as explained earlier. After this step, the 3D location of each segment (with respect to the inertial coordinate system) could be computed using the camera calibration parameters. As we were using a lateral stereo configuration, the disparity of a corner point and its location in image coordinates was adequate to recover its corresponding 3D coordinates.

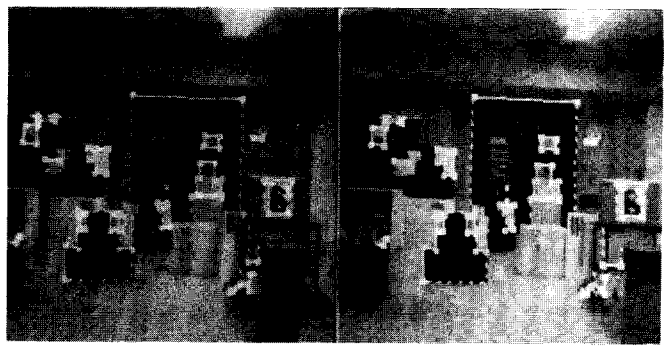

Figure 6. Matched triplets of connected segments and corners in the left and right camera images in frame 1 (———: central segment; – – – –: side segments)

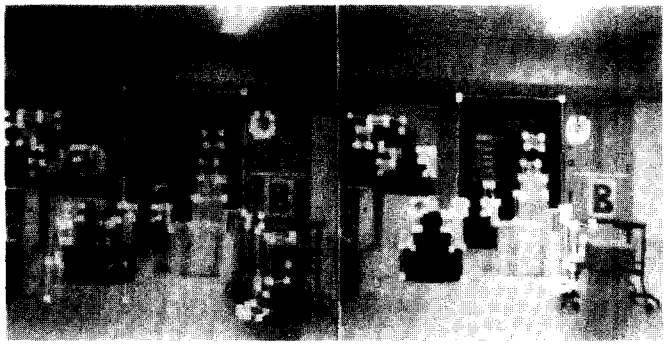

Figure 7. Matched (left) and projected (right) line segments in the right camera image in frame 8, superimposed on the right camera image in frame 1

Figure 9. Left: segments matched between frames 1 and 2; right: segments matched in only frame 1 or only frame 2

Registering stereo images

For each viewpoint, after the triplet matching process was complete, the corners on the central segment of each triplet could be considered to have been reliably matched. The 3D locations of these matched corners were known with respect to that viewpoint. In our experiments the motion between two viewpoints was approximately known. Using this approximate motion estimate, the detected corners from the second viewpoint were projected back to the first viewpoint. Next, we attempted to find one-to-one matches between these back-projected corners and the measured corners in the first viewpoint. This matching process was based on similarity of the segments composing the corners and also in the estimated disparities. After this step, we had a set of corners (points with known 3D locations), which were in correspondence between the two viewpoints. Using these matched corners, an improved motion estimate was derived as described earlier.

In Figure 7, the success of the motion recovery process in registering camera images between two viewpoints is illustrated. On the left, the matched line segments in the right camera image in frame 8 are shown, superimposed on the right camera image in frame 1. On the right, the same matched line segments are shown, after projecting them back to frame 1 using the recovered camera motion. Note the projected line segments coincide almost exactly with expected edge locations in the image.

Assimilating stereo data

Each matched segment in the new viewpoint was first back-projected to the viewpoint in the first frame using the improved motion estimate. Next, we sought one-to-one matches between the projected segments and the segments in the current estimate of the spatial layout. Segments for which matches were found were fused by simple averaging and the beliefs were updated by Bernoulli's rule, as described earlier. Segments which were not matched were also retained with the objective of finding matches from future frames.

In Figure 8, the matched segments from the right camera image in frames 1 and 2 (after projecting them back to frame 1) are depicted. Note that a 1–1 correspondence existed between some, but not all, of the matched segments in frames 1 and 2. Only those segments matched between the two frames would have their beliefs enhanced after the assimilation stage. In Figure 9, the matched and the unmatched segments are shown, superimposed on the right camera image in frame 1. Finally, in Figure 10, different views (perspective front view, side view, top view, and isometric view)

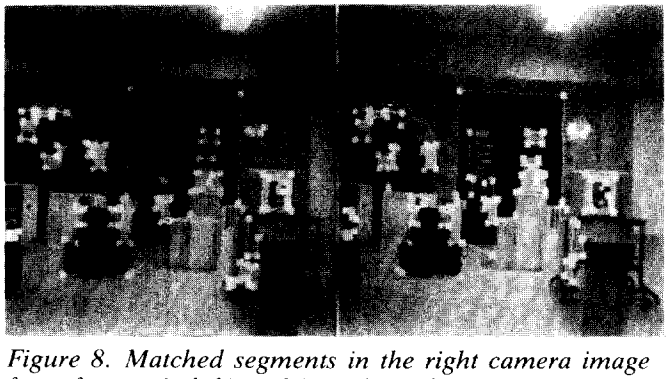

Figure 8. Matched segments in the right camera image from frames 1 (left) and 2 (right), after projecting them back to frame 1

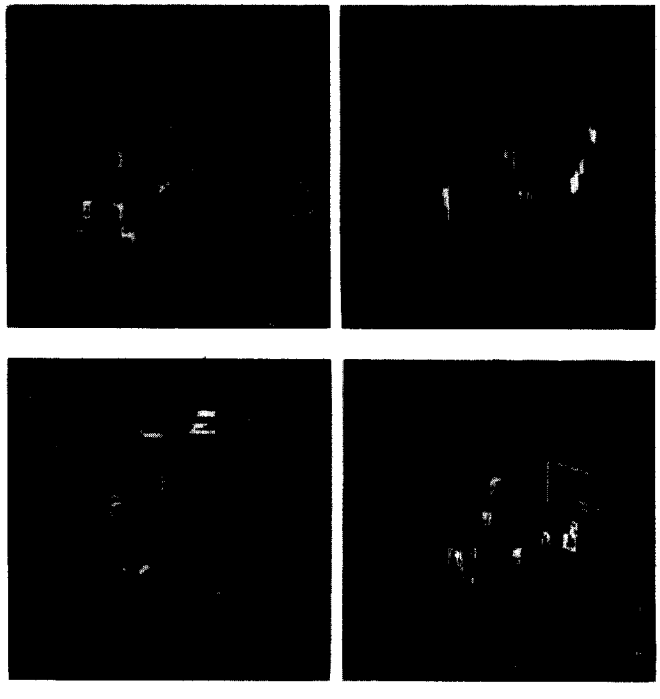

Figure 10. Different views of the reconstructed boundary-level structural description after processing frames 1–10. Top left: perspective front view; top right: side view; bottom left: top view; bottom left: isometric view

of the reconstructed boundary-level structural description, after processing frames 1–10, are depicted. Only segments with beliefs greater than 0.6 are shown.

CONCLUSIONS

We have presented a dynamic stereo algorithm to compute a boundary-level structural description of a scene. The basic approach is designed to work with scenes containing polyhedral objects. Extensions to this work under investigation include generalizing this framework to deal with curved objects, application of geometric reasoning techniques to compute higher-level representations from the boundary-level representations, the integration of other depth sensors (laser rangefinder, sonar) with the binocular stereo sensor used in this work, and the application of this work for mobile robot path planning.

ACKNOWLEDGEMENTS

This work was sponsored in part by a grant from ERIM-CAMRSS.

REFERENCES

1 **Grimson, W E L** 'Computational experiments with feature based stereo algorithm', *IEEE Trans. PAMI*, Vol 7 (July 1985) pp 17–31

2 **Hoff, W and Ahuja, N** 'Surfaces from stereo: integrating feature matching, disparity estimation and contour detection', *IEEE Trans. PAMI*, Vol 11 (February 1989) pp 121–136

3 **Ayache, N and Faverjon, B** 'Efficient registration of stereo images by matching graph descriptions of edge segments', *Int. J. Comput. Vision*, Vol 1 No 2 (1987) pp 107–131

4 **Medioni, G and Nevatia, R** 'Segment based stereo matching', *Comput. Vision, Graph. & Image Process.*, Vol 31 (July 1985) pp 2–18

5 **Marapane, S and Trivedi, M** 'Region-based stero analysis for robotic applications', *IEEE Trans. Syst. Man. & Cybern.*, Vol 19 No 6 (November/December 1989) pp 1447–1464

6 **Dhond, U R and Aggarwal, J K** 'Structure from stereo – a review', *IEEE Trans. Syst. Man & Cybern.*, Vol 19 No 6 (1989) pp 1489–1510

7 **Ballard, D H** 'Reference frames for animate vision', *Int. Joint Conf. on AI*, Detroit, MI (1989) pp 1635–1641

8 **Bolles, R C, Baker, H and Marimont, D H** 'Epipolar-plane image analysis: an approach to determining structure from motion', *Int. J. Comput. Vision*, Vol 1 (1987) pp 7–55

9 **Matthies, L, Kanade, T and Szeliski, R** 'Kalman filter-based algorithms for estimating depth from image sequences', *Int. J. Comput. Vision*, Vol 3 (1989) pp 209–236

10 **Moezzi, S and Weymouth, T** 'A computational model for dynamic vision', *Int. Conf. Robotics & Automat.*, Cincinnati, OH (May 1990) pp 1148–1153

11 **Tirumalai, A P, Schunck, B G and Jain, R C** 'Dynamic stereo with self-calibration', *3rd Int. Conf. on Comput. Vision*, Tokyo, Japan (December 1990) pp 466–470

12 **Zhang, Z and Faugeras, O** 'Building a 3-d world model with a mobile robot', *Int. Conf. on Patt. Recogn.*, Atlantic City, NJ (June 1990) pp 38–42

13 **Shafer, G** *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ (1976)

14 **Burns, J, Hanson, A and Riseman, E** 'Extracting straight lines', *IEEE Trans. PAMI*, Vol 8 No 4 (1986) pp 425–455

15 **Horn, B K P, Hilden, H M and Negahdaripour, S** 'Closed-form solution of absolute orientation using orthonormal matrices', *J. Opt. Soc. Am.*, Vol 5 No 7 (July, 1988) pp 1127–1135