COMPUTING 3-D STRUCTURE OF RIGID OBJECTS USING STEREO AND MOTION

Thinh V. Nguyen, PhD.
MULTISIGNAL TECHNOLOGY CORPORATION
4662 Katella Ave., Suite H, Los Alamitos, CA 90720

ABSTRACT : *This paper presents our work performed as a step toward an intellegent automatic machine vision system for 3-D imaging. The problem considered here is the quantitative 3-D reconstruction of rigid objects. Motion and stereo are the two cues to be utilized in our system.*

The system basically consists of 3 processes: (i) the low-level process to extract image features, namely the corner points, (ii) the middle-level process to establish the correspondence in two modalities: stereo (spatial) and motion (temporal) and (iii) the high-level process to compute the 3-D coordinates of the corner points by integrating the spatial and temporal correspondences. Once the final correspondence is obtained, the corner points in the 3-D space can be determined easily as the intersection of two lines connecting the light sources to the corresponding points in the left and right views.

I. INTRODUCTION

Our problem is stated as follows:

Given 2-D images from a frame sequence of a moving rigid object taken by stereo cameras, perform a quantitative 3-D reconstruction on the object, i.e. determine the 3-D coordinates of the object control points.

The main goal of our research is to develop a strategy to combine information extracted from two independent sources: stereo and motion. Each source carries different cue on the object 3-D structure. Our work exploits the complementary nature of these two sources to reconstruct the object.

II. OUR APPROACH

Our approach is a hierarchical one. We break the solution into three distinct processes: low-level process, middle-level process and high-level process.

1) **Low level process:** *Feature extraction*

The low-level process segments the images into relevant features. These features will be used as the image descriptors for subsequent processing.

In the problem we are considering, there are images of rigid objects with regular geometrical shapes, i.e. shapes with well defined lines, curves and corners. The features to be extracted, therefore, should be related to these characteristics. One type of feature that is particularly useful for the processing of rigid objects is the corner point.

A corner point in an image is defined as a point which is an edge point and has significant change in the edge direction. Several researchers have proposed many

corner detector algorithms. After several experiments, we found that the Zuniga-Haralick [3] corner detector performed reasonably well for a variety of scenes.

2) **Middle level process:** *Correspondence*

The input to our correspondence process is the list of corner points detected in the segmentation (feature extraction) process. The spatial correspondence due to stereo will be carried out for two views (left and right) at each frame instant. The temporal correspondence due to motion will be carried out for two consecutive frames of the same view. Therefore, each correspondence process will involve four images grouped in four pairs. We solve the spatial correspondence by the epipolar line technique and the temporal correspondence by the relaxation matching method.

The epipolar line technique requires that the geometry of the two stereo cameras is known. For each point in the right image (the use of the right image is purely arbitrary), the equation of the epipolar line (in the left image) is computed. All points that lie in the neighborhood of this line are obtained as the spatially corresponding points.

The temporal correspondence is determined by a cooperative relaxation matching algorithm similar to the matching technique by Barnard and Thompson [1]. Our matching algorithm differs than that of Barnard and Thompson in two aspects: (i) the initial similarity measure is based on neighborhood interdistances and (ii) the matching is carried out in two directions (forward and reverse), only those points that are matched consistently in both directions are kept. This method eliminates the need of selecting proper probability threshold and also reduces false matches.

The details of these algorithms are described in [2].

3) **High level process:** *Integration*

The main task in the high level process is the resolution step. The resolution step uses the consistency principle. Correspondence must be consistent for both stereo and motion. In other words, if two points P_L and P_R of the left and righ image at frame i are two spatially corresponding points, then at frame j, Q_L and Q_R must also be spatially corresponding where Q_L and Q_R are the two temporally corresponding points of P_L and P_R respectively.

The spatial and temporal matchings will form loops. A loop is a closed matching sequence. Each loop will essentially pass through four points in the four images as illustrated in Figure 1. There are two kinds of loops:

. Shared loops: Loops which share same point(s).

. Single loops: Loops which do not share any points with any other loops

Conceptually, single loops are usually stable loops which represent correct matching sequences of all the four points in four images. There are cases, however, that single loops pass through incorrect corresponding points due to error in temporal correspondence or noisy conditions. We, therefore, propose the use of single loops only as a guide to search for correct loops, single or shared.

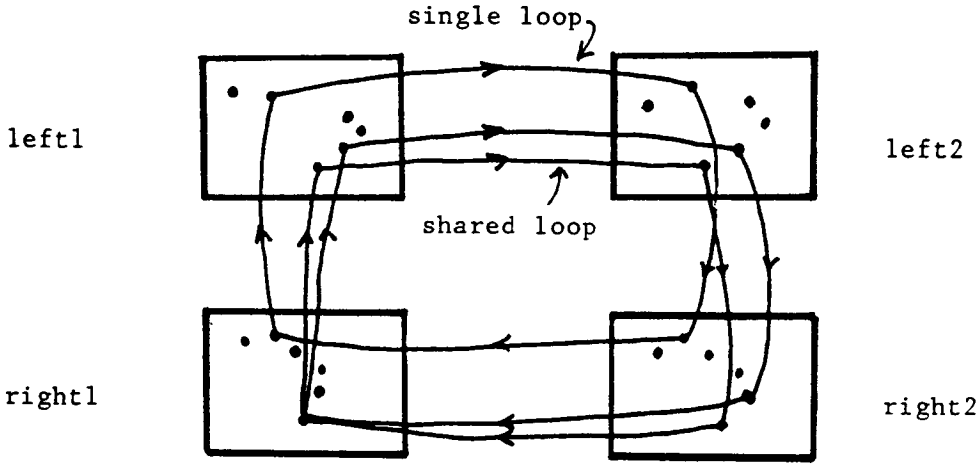

Figure 1 : Example to show some loops

Let : right1,left1 denote the two stereo views of the first frame.
right2,left2 denote the two stereo views of the second frame.
N denote the total number of points in the right1 view.

Our resolution algorithm then consists of the following steps :

Step 1: Determine single loops - For each point in the right1 image, find all closed loops going in the clockwise direction (the selection of clockwise direction is arbitrary). Select only single loops from these closed loops. Repeat for all N points in the right1 image. Let K denote the number of single loops.

Step 2: Determine initial object points - For each stereo pair in each single loop, compute the 3-D coordinates of the points in space. The result of this step is a set of 3-D coordinates of K points in frames 1 and 2. These K points will serve as a basis for further refinement.

Step 3: Find all possible matches - For each point in the right1 image, find all the possible matches in other images.

Step 4: Sort out potential candidates - From all possible loops for each point in the right1 image, select the best L loops (L is a small number relative to N) . The criterion for determining "best" loop is the *deviation from rigidity*. This deviation from rigidity is determined by computing the difference of the sums of distance errors from the two 3-D points in two consecutive frames. The distance is from the point to the K points found in step 2.

Step 5: Applying rules to select the best pairs - From the L best candidate pair found in step 4, we then apply the following heuristic rules to select our final best pairs because the smallest deviation of rigidity does not necessarily lead to correct loop due to potential errors and mismatches.

. Rule 1: If the smallest error is three times smaller than the second smallest error then the pair having the smallest error is the final best pair.

. Rule 2: If any single loop in step 2 is one of the candidate pairs, and if the relative difference between the corresponding error and the smallest error is less than 0.2, and if this corresponding error is less than a threshold then the single loop in question is the final best loop.

Step 3, 4 and 5 are repeated for all points in the right1 image. From these loops, we then remove redundant loops because some of these loops may be the same loop by keeping only the loops which have the first point in the right1 image the same as the corresponding starting point of that loop.

From the final best loops, the 3-D coordinates of the object point can be determined easily as the intersection of the two lines connecting the two focus points of cameras and the two stereo points on the image planes. Let (x,y,z) and (x',y',z') denote the coordinates of two temporally corresponding points , the object motion can then be modeled as an affine transform . The affine transformation is a mapping to transform the three coordinates (x,y,z) to a new set of coordinates (x',y',z'). The overall effect of rotations and translations results in the following equations :

$$\begin{aligned} x' &= a_0x + a_1y + a_2z + a_3 \\ y' &= b_0x + b_1y + b_2z + b_3 \\ z' &= c_0x + c_1y + c_2z + c_3 \end{aligned} \qquad (1)$$

When the number of object control points is sufficiently large (greater than 5 for example), we can use the computed coordinates to estimate the a_i,b_i parameters ($i = 0,1,2,3$) in the equations (1) using the least squares method. These parameters can then be used to predict object coordinates so that other tasks (e.g. tracking) can be performed.

III. RESULTS

We used simulated images to test our technique. The simulated images were random dots in space. We arbitrarily placed these dots on the surface of an ellipsoid. For the results reported here the lengths of the ellipsoidal axes are 500 mm, 400 mm and 500 mm in the x,y, and z directions respectively.

The computed 3-D coordinates are then compared to the true 3-D coordinates. The error is computed as the distance between the computed point and the true point. We counted the number of points that have errors in 3 groups. Group 1 consists of points that have small error (from 0 to 30 mm) . Group 2 consists of points that have medium error (from 30 mm to 75 mm). Group 3 consists of points that have large error (greater than 75 mm). The number of generated points is 10. Uniform random noises are added to displace these points. These points are also randomly deleted in both image planes. For 10 runs, the total number of points in both frames is 200.

The camera focus length is 16 mm. The two cameras are displaced by 500 mm, 0 mm and 50 mm in the x,y and z directions respectively. The right camera is rotated 10^o in the y direction . The object is placed at a distance of 2800 mm in the z direction . Motion parameters of the object are : translation (-50 mm, 0 mm, 50 mm) , rotation (5^o ,5^o , 5^o) in the x,y and z directions respectively.

The result of one typical simulation is shown in Table 1.

No. of deletions	Noise strength (pixels)	Group1	Group 2	Group 3
(0,0)	0	131	0	1
(0,0)	1	131	0	1
(0,0)	2	103	26	3
(0,0)	3	71	36	16
(0,0)	4	52	38	26
(0,0)	5	37	53	32
(1,1)	0	118	0	1
(1,1)	1	118	0	1
(1,1)	2	95	18	2
(1,1)	3	64	31	13
(1,1)	4	45	35	12
(1,1)	5	38	41	20
(2,2)	0	103	1	2
(2,2)	1	103	1	2
(2,2)	2	82	16	4
(2,2)	3	60	35	10
(2,2)	4	34	43	17
(2,2)	5	29	36	22

Table 1: Number of computed points in 3 groups.

IV. CONCLUSION

This paper describes our work in determining the 3-D structure of rigid objects using stereo and motion. The image features are the corner points. Correspondences of image features are carried out for stereo (spatial correspondence) and motion (temporal correspondence) using two independent methods. The two types of correspondence are then integrated to produce the final correspondence which provides the 3-D coordinates of the image features. Our integration technique exploits the rigidity constraint and heuristic rules. Results obtained by simulation show that our technique works reasonably well even under several noise sources.

ACKNOWLEDGEMENTS

The work described in this paper is supported by NASA Marshall Space Flight Center under Contract No. NAS8-37308. The guidance and support of NASA personnel, in particular, Mr. Glenn Craig, are sincerely acknowledged.

REFERENCES

[1] Barnard, S.T. and Thompson, W.B., "Disparity Analysis of Images," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, Vol. PAMI-2, No. 4, July 1980, pp. 333-340.

[2] Nguyen, T.V., "Computing Range and 3-D Structure of Rigid Objects using Stereo and Motion," MULTISIGNAL TECHNOLOGY CORPORATION, *Technical Report TR-87.001*, July 1987.

[3] Zuniga, O.A. and Haralick, R.M., " Corner Detection Using the Facet Model, *Proc. IEEE Computer Society - Computer Vision and Image Processing*, 1983, pp. 30-37.