AN ALGORITHM FOR STEREO-IMAGE MATCHING WELL ADAPTED FOR A PARALLEL SYSTEM

Xiao-Wei TU and Bernard DUBUISSON

Compiègne University, U.A. CNRS Heuristique & Diagnostic des Systèmes Complexes
B.P.649, 60206 Compiègne, France

ABSTRACT

In stereovision analysis, the successful implementation of a matching algorithm of features in images is considered as the key step. A new algorithm of feature matching in a pair of stereo images well adapted for parallel processing is presented. First, we establish a triangulation geometry between two camera models and a point in 3-D scene, which permits us to realized the algorithm in a parallel manner, then after a brief presentation of feature extraction, representation and implementation, the efficient stereo feature matching algorithm with two independent stages based on edge features is proposed. Finally, we give some possible improvements and some conclusions.

Key words: Stereovision, Feature extraction, Zero-crossings, Epipolar line, Feature Matching, Global consistency, Disparity.

1. INTRODUCTION

If the same scene is observed from two (or more) different positions (viewing points), three dimensional locations of the objects presented in the scene can be measured. This technique is termed 'Stereo vision' or 'Binocular vision' ('Multiviews vision'). In fact, it is the most important way in which humans capture range information. The stereo image analysis in the field of computer vision attracts a lot of attention of researchers, and it began earlier than any other ranging methods because it works in a large variety of conditions. For example it has been used for obtaining terrain shapes from aerial photographs, where active ranging methods cannot be applied. Numerous applications in a variety of domains can be found in litterature, such as change detection, target acquisition and tracking, passive navigation(1,2) and scene reconstruction(3).

In stereo image analysis task, the major difficulty is to find correspondences between two images.

In this paper, we propose a matching algorithm based on edge features. For each epipolar line, the optimal solution of feature correspondence is searched recursively. The global consistency of matches is confirmed in entire images by using an accumulation table counter. In section 2, we will describe briefly a camera model as well as the relation between image features and the points in 3-D space. The method of feature extraction is also shown. Feature matching in a local independent way and gobal consistency consideration are illustrated in section 3 and some conclusions and possible improvements of the algorithm are discussed in section 4. It should be noted that the successful implementation of our system in laboratory condition, which will be easily generalised, allows us to have first hand experience for launching recently a research project on navigable robot RoMo SAPIENS, the later will be equiped with three cameras on board.

2. CAMERA MODEL AND FEATURE EXTRACTION

2.1 Triangulation Calculation

For obtaining the three dimensional information, a stereovision approach has been undertaken in which a dual camera model is necessary and a correspondence among feature points in the pair of images should be established.

In order to adapt the computational method as well as to limit the complexity in task of camera modelisation, we suppose that the focal axes of two cameras are aligned in parallel to each other, the focal lengths of the cameras are equal and that the baseline is horizontal and parallel to the scan lines in images. In Fig.1, we show this specified configuration. Fig.1 shows the sketch of the 3-D geometry of the problem, we call E_j the epipolar line which is the intersections of the two image planes with an epipolar plane, defined to be the plane passing through an object point and the two camera foci O_1 and O_2. The direction of Y-axis is coplanar with the optical axes of the cameras. It is evident that, in this configuration, all epipolar lines are parallel to the baseline, and that they are confused with horizontal scanlines of the images. Given this accurate knowledge of camera model, we search the possible matched feature points only in two image lines having the same Y-axis value in each other, rather than the whole image. The calculation formulas are shown in the figure, in which L is the length of the baseline, D the disparity value. Normally, we expect that L is as long as possible under the constraint that the most features seen in left image are also appeared in the right one. X_a and X_b are the local coordinates in X-axis of the corresponding pixels in two images respectively, Y is the coordinate in Y-axis of the pixels and f is the focal length of the two cameras. We also assume, as in Fig.1, that we know how to express the camera coordinates in terms of a common coordinate system.

2.2 SELECTION OF FEATURES TO BE MATCHED

As we have explained, the major problem in 3-D information derivation is to find the correspondence in a pair of images. It implied to have an unambigous match by mean of some 'intermediate corresponding features' such as regions, straight line segments, special curves such as ellipses, circles and the most natural primitives; interesting points.

In our approach, we use edges with relatively arbitrary form as features to be matched. We think that regions are too sensitive in segmentation step to be used as primitives; when the order of operation using split and merge approach in a method of region segmentation is slightly changed, the resulting image partition is totally different; and moreover the segmentation is not stable when there is some interferences in light condition, since in this case, the corresponding regions in images are not always the projections of the same physical surface patch of real scene. We believe also that some methods of special curve approximation blur the exact location of contours in two images, and that they lose at least necessary precision. It is also necessary that the objects in the scene have some form of contours with special characteristics. Considering these facts, we apply a Laplacian-of-Gaussian [(4)] operator (LoG operator) in two images at first stage, then, zero-crossing points are detected, and are considered to be the edge points. Zero-crossings are the points at which the second derivatives of an intensity image change their sign. It is believed that this procedure gives edges with sub-pixel precision[(4)], which is very important in derivation of 3-D information in subsequent processing. When obtaining edge points, we begin to assign the number tags (or labels) to all vertically connected contours. The procedure to select the vertical connected edges scans each epipolar line to find the first point of an edge. If there is one in position (x,y), it explores only three directions namely (x-1,y+1; x,y+1;x+1,y+1) in next line for tracing the next edge point. That is why the maximal skew angle between any allowed edge and y-axis is 45°. It can be noted that neighboring pixels in the same epipolar line cannot be considered as the points in a same edge. Once the end edge point is encountered, and if the number of points in actual edge is superior to a prefixed threshold value, the edge is acceptable and, the edge points are given a number to replace the gray values in the contour image, which serves also as the label of this edge. In the next section, we can see that finding correspondence is done solely for these approximatively vertically oriented edges because these contours give the matches without ambiguity and other edges with different orientations can be eventually reconsidered with a third camera not to be installed on the same baseline in order to complete information. It should be emphasized that at this point, we have two stereo contour images with accuracy vertical edge positions, since only the features oriented across the epipolar lines provide an accurate match.

3. CORRESPONDENCE FINDING IN A PAIR OF IMAGES

Any successful matching algorithm consists of two general stages implicitly or explicitly: match features with the accordance of their local characteristics and global coherence among the matches, although the whole strategy to find correspondence can be different one to the other. In our case, the epipolar line matching is considered to be the first step in matching process, and discussed in 3.1. The use of an accumulation table to verify statistically the global edge matches among these lines is presented in 3.2.

3.1 EPIPOLAR LINE MATCHING

The camera geometry decides that the corresponding points must lie in corresponding row (with the same Y axis value of epipolar lines), which reduces the search space for possible matches from two dimensions to one, without reference to any other rows (and so rows could be processed in a parallel manner).

When scanning each epipolar line in the left and the right image, we need at first to construct a data structure corresponding to the edge points in the lines for further representation and processing. The data structure we have proposed is composed of two lists whose elements are the records (one to one map to the edge points in two epipolar lines) containing some fields. The informations in these fields are: the label of the edge containing the point, the coordinate value in X-axis of the point, the pointer to the next point in the same epipolar line and the pointer to the corresponding point in another image. The matching becomes now how to find the corresponding record in two lists. In the following, we call the records representing the edge points in the epipolar line under consideration as points for simplicity.

3.1.1 ASSUMPTIONS AND DEFINITIONS

We assume that the pointer fields linking the corresponding points in the records of two lists have the value NULL initially, which means that there is no match at all before matching process. There are two procedures to be defined in the algorithm.

First, LINE_MATCH(): a recursive procedure for selecting allowed matches points between the left and the right epipolar lines by calculating the goodness of two match sets of points. This procedure must be applied systematically to enumerate all the allowed matches at each level of the recursion, all allowed possibilities must be explored.

Second, POINT_MATCH() a procedure which evaluates the goodness of a particular selected pair of possible matched points by a cost value, the larger the value is, the more is the difference between two points.

The symbol '^' means the 'match between' two sets of points.

Let R and L be two sets of ordered points according to their x-axis coordinates in an arbitray right and left epipolar line including an empty point, and $R_r^t(i)$, $R_l^t(i)$, $L_r^t(j)$ and $L_l^t(j)$ be their four subsets separated by point i in R and j in L with recursive depth t, the indices r and l represent right and left part in an epipolar line delimited by i ou j. (i,j) and $*$ represent random pair of points in set RxL and empty point in R or L respectively.

Operator '->' means: fetch a value of any fields in a point. For example, i->X_coordinate is to draw the value of coordinate of point i above X-axis in the epipolar line.

3.1.2 DESCRIPTION OF THE ALGORITHM

Let us describe the algorithm of epipolar line matching as following:

1.For each epipolar line in two images, two lists of data structure as explained above are constructed with evalution of initial values for all points in them.

2.Considering all pairs of points (i,j) on the left and the right epipolar line, respectively, compute recursively the goodness of the matching between two sets of points R and L by using the following relations:

$$\begin{aligned}\mathrm{LINE\_MATCH}(R \wedge L) = {} & \mathrm{LINE\_MATCH}(R_r^1(i) \wedge L_r^1(j)) \\ & + \mathrm{LINE\_MATCH}(R_l^1(i) \wedge L_l^1(j)) \\ & + \mathrm{POINT\_MATCH}(i,j) \qquad t=1\end{aligned}$$

$$\begin{aligned}\mathrm{LINE\_MATCH}(R_r^{t-1}(i') \wedge L_r^{t-1}(j')) = {} & \mathrm{LINE\_MATCH}(R_r^t(k') \wedge L_r^t(l')) \\ & + \mathrm{LINE\_MATCH}(R_l^t(k') \wedge L_l^t(l')) \\ & + \mathrm{POINT\_MATCH}(k',l') \qquad t>1\end{aligned}$$

$$\begin{aligned}\mathrm{LINE\_MATCH}(R_l^{t-1}(i') \wedge L_l^{t-1}(j')) = {} & \mathrm{LINE\_MATCH}(R_r^t(k'') \wedge L_r^t(l'')) \\ & + \mathrm{LINE\_MATCH}(R_l^t(k'') \wedge L_l^t(l'')) \\ & + \mathrm{POINT\_MATCH}(k'',l'') \qquad t>1\end{aligned}$$

where (i',j'), (k',l') and (k'',l'') are three pairs of points in R_r^{t-2}(i'')XL_r^{t-2}(j''), R_r^{t-1}(i')XL_r^{t-1}(j') and R_l^{t-1}(i') XL_l^{t-1}(j'), as refered in Fig.2.

What we are trying to do actually is to search the minimal value of the difference returned by LINE_MATCH(R^L) with different choice of corresponding pair (i,j) in recursive levels, which implies usually the best match between two sets of points in two epipolar lines.

3. Goto 1, until that the last epipolar line in images has been treated.

It should be pointed out that the procedure POINT_MATCH(i,j) performs two operations, one is to calculate the similarity between possible matched pair (i,j) and to return the value, the other one is an affection of pointer, namely: i->pointer = j means i is corresponding to j; in addition, we impose that the value of POINT_LINE(*,j), and POINT_LINE(i,*) be any possible maximal values and *->pointer=NULL; i->pointer = * = NULL and j ->pointer = * = NULL mean that i or j has no corresponding point in his homologue line and that if any point has no correspondance, there is a penalty in total sum value of LINE_MATCH(R^L) because of this possible maximal value.

There are two cases in which the recursive exploring can not occur (see Fig.3).

a. There is only one point in either actual R_r^t(i), R_l^t(j), L_r^t(i) or L_l^t(j) , and the returned value is simply calculated by the following formulas:

MIN(LINE_MATCH(R_r^t(i) ^ L_r^t(j))) = MIN(POINT_LINE (i,j)) for 1<= i <= M, where M is the number of points in left epipolar line (the same for R_l^t(i)^L_l^t(j)).

MIN(LINE_MATCH(R_r^t(i)^L_r^t(j))) = MIN(POINT_LINE (i,j)) for 1<= j <= N, where N is the number of points in right epipolar line (the same for R_l^t(i) ^ L_l^t(j)), where MIN() is used to calculate minimal value among all possible considerations.

b. There are only one point in both R_r^t(i) and L_r^t(j) ou R_l^t(i) and L_l^t(j):

MIN(LINE_MATCH(R_r^t(i)^L_r^t(j))) = POINT_MATCH(i,j) (the same for R_l^t(i)^L_l^t(j)).

Now, let us see the concrete value of POINT_MATCH(), the similarity measure related to the procedure can be implemented in various ways. The simplest method is to take an absolute distance metric in X-axis between two points in left and right image. That is:

POINT_MATCH(i,j) = | i->x_coordinate - j->x_coordinate |

POINT_MATCH(*,j) = POINT_MATCH(i,*) = 512 (size of the images)

It is entirely flexible to include other more intelligent or heuristic measurement. Anyway, the correspondence finding process gives an optimal solution with minimal match cost in an epipolar line.

To reduce the number of matchable candidates for fast processing, we can limit the search area in right epipolar line within certain distance in one side of the point position in left epipolar line supposing that the disparity value is smaller than a predefined value called d (see FIG.2). When the partial value of LINE_MATCH(R_r^t(i)^L_r^t(j)) or LINE_MATCH(R_l^t(i)^L_l^t(j)) in recursive depth t is greater than the value of LINE_MATCH(R^L) which has been previously calculated with certain match state and is actual minimum, the current match (i,j) is abandoned, another pair of (k,l) will be considered. It is evident that this additional consideration accelerates the search speed of minimal value LINE_MATCH(R^L).

3.2 Global consistency in the images

The optimal value of LINE_MATCH(R^L) in each epipolar line does not mean that the match of points in the line is correct in a global way, so it is important to consider all edge matches in this stage for general coherence and avoidance of any coincidence in individual lines. Here , we use an accumulation bidimensional table called ACCTAB with row indice representing the edge number in the left image, and column indice, the edge number in the right image. When we have a pair of points (i,j) corresponding to each other in two epipolar lines, the value of ACCTAB(L_i,L_j) will be increased, where L_i and L_j are the labels of two edges which include the points i and j respectively. The labels can be calculated directly as shown previously with L_i = i->label, L_j = j->label in the record operations. Now, we will discuss in detail the manner in which how the incrementation of the elements in table ACCTAB(L_i,L_j) is realized. The calculation of the elements of table ACCTAB is done according to the local edge pattern of i and j in two images. The updating relation is:

$$\text{ACCTAB}(L_i, L_j) \leftarrow \text{ACCTAB}(L_i, L_j) + \text{DELTA}$$

where, DELTA is an interline measure of similarity and can take value 1, 1/2, 1/3, 1/4 and 1/5, which is the difference between the local pattern of i and j , and gives in the same time the reliability about the matched pair i and j in a particular line. In Fig.4, we show several local patterns of i and j in the edge images and their DELTA values. It is evident that when the local patterns of all pairs of i,j along the edges labelled L_i and L_j are exactly the same, then the ACCTAB(L_i,L_j) element corresponding to the pair will be increased rapidly, otherwise the ACCTAB(L_i,L_j) element will be grown slowly, an element ACCTAB(L_i,L_j) can be stable after some line processing, when this occurs, there is not any corresponding pair of points (L_i,L_j) within these lines. While this process for all epipolar lines is completed, the elements in ACCTAB(L_k,L_l) which have large values (locally maximum) or values above a prefixed threshold are selected, this implies the correct match between edges labelled by Label-k and Label-l. We can also deal with the situation where two edges in one image corresponding to two or more edges in another image, this sort of the case is possible when in an image, an edge is broken into two parts due to noise contaminated in the original image or failure of performance in preprocessing method. The algorithm allows us to cope with the situation supposing that the broken edges have an acceptable length.

It is also convenient to note that the global consistency match by using an accumulation table operates very fast in processing time, and , in addition, provides the possibility of matching all epipolar lines in a parallel environment without any global constraints.

4. IMPROVEMENTS AND CONCLUSIONS

We have tested our algorithm for a pair of stereo images which are taken successively by a CCD camera displacing about 20 cm in the direction strictly along to the imagery baseline, analogous to two cameras, with 70 mm focal length.

It is instructive to compare our approach with others: we think that the algorithm is less time-consuming in global coherent matching. On each epipolar line it gives an optimal solution; quite on the contrary, most of the methods search a possible match in a limited space on the same epipolar line, and only a limited number of paths are explored; hence, once they find a local relatively better solution, the matched pair is determined no matter what they have found is not the best match in a long term for further processing, the best match would also be discarded when it is not among the several minimal local costs thus far.

As we said, we can improve our algorithm by introducing three constraints in search process: 1. We provide a prefixed maximal disparity value in epipolar lines, which avoids to search all points in left epipolar line for a correspondence of a point in right epipolar line and vice versa. 2. When the partial evaluation of value LINE_MATCH() is superior to the actual minimal value of LINE_MATCH(R^L) having been obtained previously, a pair of (i,j) probably corresponding is discarded, no further recursive level will be explored. 3. The permutation of matching $R_l^t(i)$^$L_r^t(j)$ or $R_r^t(i)$^$L_l^t(j)$ is not permissible. These constraints cut considerable time of matching, but quantitive analysis appears difficult.

To be able to cope with the new problems encountered in navigable robot, we shall improve our algorithm in order to include region information presented in two images in a global way instead of using only edge continuity constraint, and accelerate the speed of epipolar line match, in addition to the use of the order constraint implicitely contained in the actual algorithm, employing more heuristic rules, and assumptions. Our method is able to match scenes much more complex than that reported in this paper. In fact, there is not any necessary modification in this generalisation.

Finally, we believe that matching edges alone is inherently limited, and higher level monocular cues detection modules are likely to be developped, and should be implemented, but our approach as suggested in this paper may constitute a major part of such a more complete system.

References

1. D.Gennery, Object detection and measurement using stereo vision, in Proc. Image Understanding Workshop, College Park, Md,. Apr. 1980, pp 161-167.

2. Toshifumi Tsukiyama and T.S.Huang, Motion stereo for Navigation for automous vehicles in man-made environments, Pattern Recognition, Vol. 20, N°1, pp 105-113, 1987.

3. Martin Herman and Takeo Kanade, Incremental reconstruction of 3D scene from multiple, complex Images, Artifical Intelligence 30 (1986) pp 289-341.

4. Andres Huertas and Gerard Medioni, Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-Gaussian Masks, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, N°5, September 1986, pp651-664

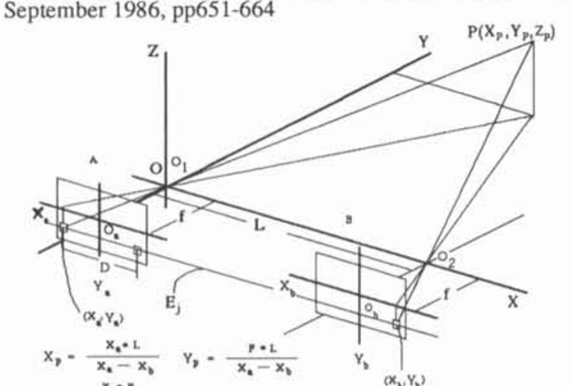

Figure 1. Configuration of a special binocular image system

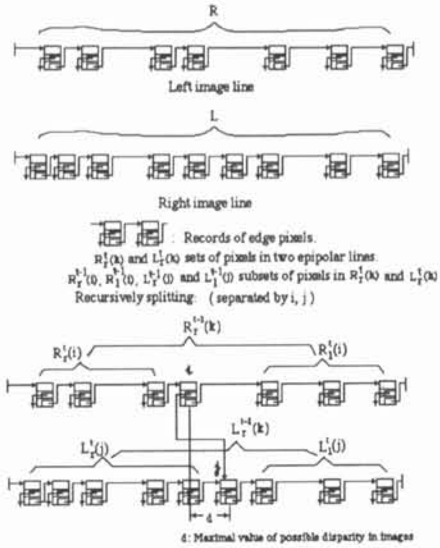

Figure 2. Matching two epipolar lines

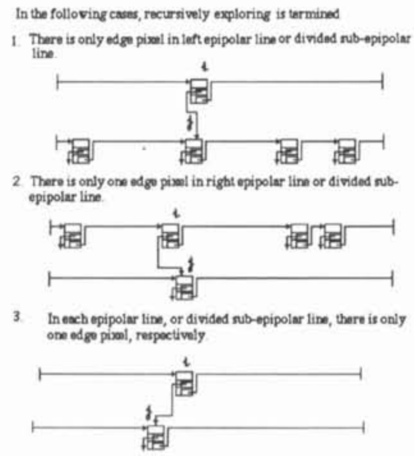

Figure 3.

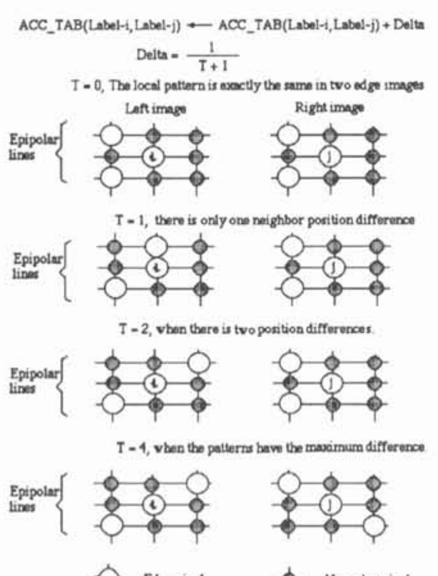

Figure 4. Local pattern of the edge pixels and T, Delta values