3—5

An Efficient Matching Algorithm for Segment-Based Stereo Vision Using Dynamic Programming Technique

Toshio Ueshiba * Yoshihiro Kawai * Yutaka Ishiyama †
Yasushi Sumi * Fumiaki Tomita *

* Electrotechnical Laboratory † Stanley Electric co., Ltd., R&D

Abstract

An efficient matching method for segment-based stereo vision is proposed. A *potential matching graph* which describes the connectivity between candidate matching pairs of segments is built. Establishing correspondence is then reduced to a problem of searching for the optimal path that maximizes a similarity measure. The optimal path is found efficiently without the adverse effects of combinatorial explosion by using a dynamic programming technique. The validity of the method is confirmed by experiments with actual images.

1 Introduction

Stereo matching is one of the most important and difficult problems in low-level vision and has attracted much attention over the decades in the vision community[2]. The numerous methods proposed so far can be characterized in terms of matching tokens, constraints and optimization methods[5].

In order to reduce matching ambiguity, various constraints between matching tokens should be fully exploited and the matching similarity should be evaluated in a global context. Low-level features such as edge points have been used as tokens in some studies, which include a coarse-to-fine strategy[3], PMF algorithm using disparity gradient limit[9] and dynamic programming techniques[8]. Using higher-level features with rich attribute information, however, is more desirable to reduce the number of ambiguous candidate matches and computational overhead.

Segment-based stereo[7, 1, 4] is one of the promising directions using high-level features as matching tokens. This approach is more suitable for high-level tasks such as model building and object recognition because it produces a much more structured description of the scene than a disparity map. Unfortunately, establishing globally optimal correspondence that fully takes into account adjacency and connectivity between segments often gives rise to combinatorial explosion[4]. Though a method which initiates the matching process at a local correspondence and integrates neighboring corresponding tokens based on the smoothness of disparity was proposed[1], the result is dependent on the selection of initial matching and there is no guarantees that it will be globally optimal.

In this paper, we propose a new stereo matching algorithm which attains both an efficient computation and an optimal result. All the potential matching between left and right segments are represented by a graph description called a *potential matching graph*. The optimal correspondence is established by finding the optimal path that maximizes a global similarity measure in this graph. This search process is performed efficiently by using a dynamic programming technique. The validity of the proposed algorithm is confirmed by experiments with actual images.

2 Boundary Representations of Images

In our method, stereo matching is performed between two boundary representations (*B-rep*) extracted from left and right input images. The B-rep is a collection of boundaries of image regions obtained by extracting edge points. Each boundary is divided into edge segments at feature points (e.g. branches, corners, etc.). Each boundary is also segmented at maxima/minima of vertical coordinate (y-value), which guarantees that the y-value of the edge points in each segment is monotone increasing/decreasing along the segment and greatly simplifies the matching procedure. Each segment is then oriented so that its surrounding region is seen on its right side (Fig. 1).

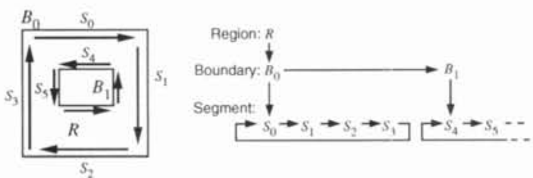


Figure 1: B-rep of the image

*Address: 1–1–4 Umezono, Tsukuba-shi, 305–8568 Japan. E-mail: ueshiba@etl.go.jp

†Address: 1–3–1 Edanishi, Aoba-ku, Yokohama-shi, 225–0014 Japan.

3 Stereo Matching

3.1 The Potential Matching Graph

Stereo matching is performed by establishing correspondences between the left and right segments in B-reps obtained from the input images. For this purpose, graph representations of potential matchings are constructed as follows; We fix one boundary B in the left B-rep and enumerate all the potentially corresponding right segments for each left segment in B by using an epipolar constraint and the similarity of their shape, intensity and direction. The potential matching between the left and right segments is called a *pair*. We then construct a directed graph called a *potential matching graph* whose nodes are pairs and whose arcs represent connectivity between the pairs (Fig. 2). See [6] for details about evaluating connectivity.

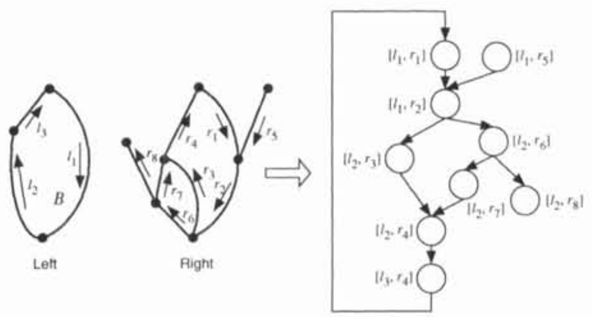


Figure 2: Graph representation of potential matching

For each pair i of the left and right segments, we compute its similarity value

$$S_{pair}(i) \equiv d_i \left(1 - \frac{\Delta I_i}{I_{max}}\right) \qquad (1)$$

where d_i is the length of matching portion, ΔI_i is intensity difference between the vicinities of two segments and I_{max} represents the maximum intensity. This $S_{pair}(i)$ yields a large value when these two segments have a large matching portion and their intensity difference is small.

3.2 The Correspondence Path

S_{pair} defined by (1) reflects only the local similarity between the left and right images. In order to obtain reliable results, however, the similarity must be evaluated more globally. Since the right segments corresponding to contiguous left segments belonging to a left boundary B must also be contiguous, the correct matching pairs must be connected to each other and form a path, called a *correspondence path*, in the potential matching graph. We therefore regard the path that maximizes the following global matching similarity

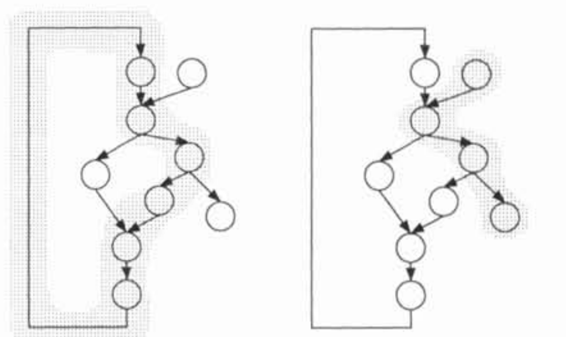
$$S_{path} \equiv \sum_{i \in path} S_{pair}(i). \qquad (2)$$

as the correct matching for B. The matching problem is then reduced to searching for the optimal path with the maximum value of S_{path}.

There are two kinds of correspondence paths;

Loop path A path forming a closed loop in the graph (Fig. 3 (a)). This happens when potentially corresponding right segments are found for all the left segments along B.

Non-loop path A path starting from a node pointed at by no other nodes and terminating at a node pointing at no other nodes (Fig. 3 (b)). This happens when potentially corresponding right segments are partially missing due to the occlusion or incompleteness of edge detection.



(a) loop path (b) non-loop path

Figure 3: Loop/Non-loop path

3.3 Search for Optimal Paths

Since S_{path} is defined as a sum of S_{pair} values, the optimal path from pair i to pair j passing through pair k is a concatenation of the two optimal paths from i to k and from k to j. From this observation, we can find the optimal path efficiently by using the following dynamic programming technique; Each node is assigned two slots: S_{pair} for the similarity value of the pair and S_{sum} for the intermediate value of S_{path}. Starting from a given node, the graph is traversed in depth-first order (Fig. 4). When back-tracking upward, S_{sum} of node i is computed recursively as

$$S_{sum}(i) = S_{pair}(i) + \max_{j \in \mathrm{next}(i)} \{S_{sum}(j)\}$$

where next(i) stands for a set of nodes pointed at by i. Finally S_{path} value of the optimal path is obtained in the S_{sum} of the starting node. The optimal path itself can be found by tracing the nodes with maximum S_{sum} values from the starting node. Since each node is visited only once, the computational complexity of this process is linear to the numbers, that represent the complexity of the scene, of nodes and arcs of the graph.

The nodes where the search process described above is initiated are determined as follows;

- We initiate the search for non-loop paths at all the nodes pointed at by no other nodes.
- We initiate the search for loop paths at one node pointed at by other nodes and pointing to other

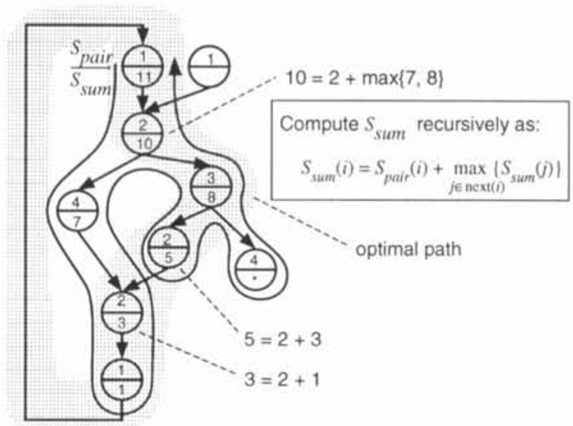


Figure 4: The optimal path search

nodes. Each node visited during the search process is marked. The search is repeated until no nodes pointed at by other nodes and pointing to other nodes remain unmarked.

If we choose a node which all the loop paths pass through, the optimal loop path for the boundary can be detected in only one search. For instance, we can check two loop paths in Fig. 2 at the same time if the search process begins at node $[l_1, r_1]$. For this reason, as a heuristic effective in many cases, we select a node with a maximum or minimum y-coordinate value.

3.4 Integration of Optimal Paths

Up to now, for one left boundary, we have obtained multiple optimal paths for each starting node in the search process described above. Multiple optimal paths are also obtained in cases where the left boundary has two or more potential matching graphs. In order to determine the sequence of the corresponding right segments uniquely, we sort the obtained optimal paths in descending order of their S_{path} values and pick them from the top of the list while keeping their corresponding portions from overlapping each other along the left boundary.

Let us consider the example shown in Fig. 5. We have three optimal paths $Path1$, $Path2$ and $Path3$ for the left boundary B where $S_{path}(1) > S_{path}(2) > S_{path}(3)$. First, $Path1$ is picked. Next $Path2$ is rejected because it overlaps $Path1$ on B. Therefore $Path3$ is selected which does not conflict with $Path1$.

3.5 Establishing One-to-One Correspondences

As a result of the above-described processing, a sequence of corresponding right segments is uniquely determined for each left boundary (left-to-right correspondence). Multiple left boundaries, however, may have the same right segments selected in their optimal matching. In order to establish one-to-one correspondences, we compute right-to-left correspondences in the same way by exchanging the roles of the left and right images and eliminate matching pairs with incompatible left-to-right and right-to-left correspondences. Refer to [6] for details.

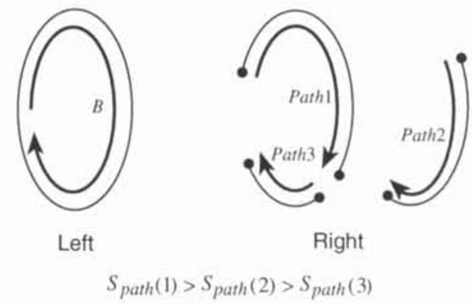


Figure 5: Integration of the optimal paths

4 Experimental Results

The computational time performance of the proposed algorithm was evaluated using actual images and compared with that of the exhaustive method which enumerates all the paths in the graph and then selects one with largest S_{path} value. Table 1 shows the CPU time (UltraSPARC-I, 167MHz) required for the search and integration of the optimal paths described in subsections 3.3 and 3.4. It can be observed that the proposed algorithm is not only much faster but is also immune from large variations in computational time. In contrast with this, the exhaustive method is sometimes very slow even for scenes with medium complexity.

Figures 6 and 7 show the matching results respectively corresponding to scene1 and scene3 in Table 1. These results are outputs of the path integration process described in section 3.4. In spite of not processing to establish one-to-one correspondences, the 3D shapes of the scene are reconstructed quite faithfully.

	num. of nodes	num. of arcs	exhaustive method (msec)	proposed method (msec)
scene1	5096	2121	923.8	619.8
scene2	594	287	87.8	21.8
scene3	2165	815	131.2	129.3
scene4	1408	803	198.7	65.1
scene5	1215	627	43.6	34.9
scene6	828	492	6590.0	57.0
scene7	1099	640	568.2	30.7
scene8	941	604	325.4	30.8

Table 1: CPU time required for the optimal path search

5 Conclusions

We have described a new matching algorithm for segment-based stereo vision. The features of this algorithm are summarized as follows;

- Potential matching is represented by a directed graph. Matching similarity is then globally evaluated by searching for the optimal path that yields the maximum value of the sum of the similarity of segment pairs.

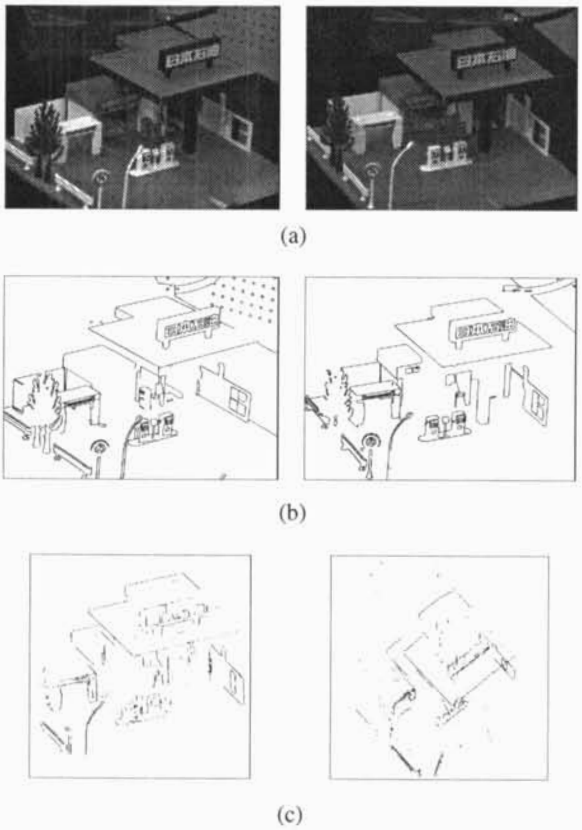


Figure 6: Matching result (1): *(a) Input images. (b) B-reps. (c) 3D reconstruction.*

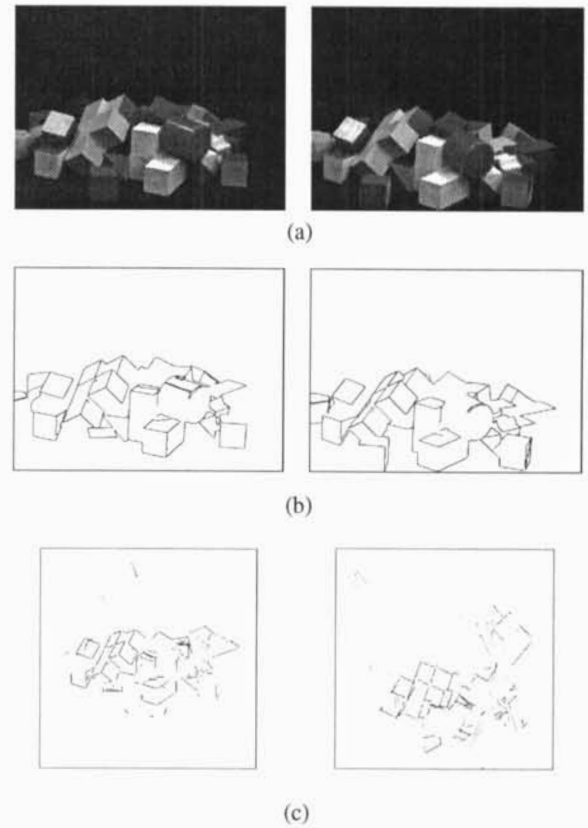


Figure 7: Matching result (2): *(a) Input images. (b) B-reps. (c) 3D reconstruction.*

- This search is efficiently performed by recursive computation of the global similarity value using a dynamic programming (DP) technique.

Since our DP process is a one-dimensional search along the boundaries in contrast with a hierarchical 2D search across epipolar lines[8], it is efficient and guarantees an optimal result.

A stereo vision system using our proposed algorithm has already been successfully applied to object recognition tasks[10, 11]. We also expect that this method will make segment-based stereo applicable to a wide range of fields such as robotics where a quick response is crucial.

References

[1] N. Ayache and B. Faverjon. Efficient Registration of Stereo Images by Matching Graph Descriptions of Edge Segments. *International Journal of Computer Vision*, 1(2):107–131, 1987.

[2] U. R. Dhond and J. K. Aggarwal. Structure from Stereo -A Review. *IEEE Trans. on Systems, Man, and Cybernetics*, 19(6):1489–1510, 1989.

[3] W. E. L. Grimson. Computational Experiments with a Feature Based Stereo Algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(1):17–34, Jan 1985.

[4] R. Horaud and T. Skordas. Stereo Correspondence Through Feature Grouping and Maximal Cliques. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(11):1168–1180, Nov 1989.

[5] G. A. Jones. Constraint, Optimization, and Hierarchy: Reviewing Stereoscopic Correspondence of Complex Features. *Computer Vision and Image Understanding*, 65(1):57–78, 1997.

[6] Y. Kawai, T. Ueshiba, Y. Ishiyama, Y. Sumi, and F. Tomita. Stereo Correspondence Using Segment Connectivity. In *Proc. 14th Internatinal Conference on Pattern Recognition*, volume 1, pages 648–651, 1998.

[7] G. Medioni and R. Nevatia. Segment-Based Stereo Matching. *Computer Vision, Graphics, and Image Processing*, 31(1):2–18, 1985.

[8] Y. Ohta and T. Kanade. Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(2):139–154, Mar 1985.

[9] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. PMF: A Stereo Correspondence Algorithm Using a Disparity Gradient Limit. *Perception*, 14:449–470, 1985.

[10] Y. Sumi, Y. Kawai, T. Yoshimi, and F. Tomita. Recognition of 3D Free-Form Objects Using Segment-Based Stereo Vision. In *Proc. 6th International Conference on Computer Vision*, pages 668–674, 1998.

[11] Y. Sumi and F. Tomita. 3D Object Recognition Using Segment-Based Stereo Vision. In *Proc. 4th Asian Conference on Computer Vision*, volume 2, pages 249–256, 1998.