Stereo Matching Using Global and Local Segmentation

Yoo-Kyung Kim, Sung-Yeol Kim, and Yo-Sung Ho
Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong, Buk-gu, Gwangju 500-712, Korea
Tel: +82-62-970-2263, Fax: +82-62-970-3162
E-mail: {waterloo, sykim75, hoyo}@gist.ac.kr

Abstract: In this paper, we present a new stereo matching algorithm using global and local color segmentation. After we perform a quad-tree decomposition to generate global and local color segments, we determine meaningful depth information between global and local disparities based on the disparity space distribution. Unlike the previous segment-based stereo matching methods, we consider globality and locality at the same time to handle textureless regions. Expereimental results demonstrate that the proposed scheme generates high-quality disparity maps.

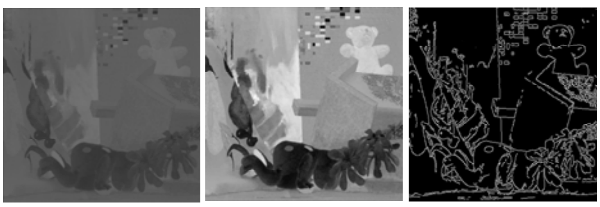
1. Introduction

Stereo matching has been studied in the field of the computer vision for many years. Recently, segment-based stereo matching methods have been proposed to generate depth maps. Although high-performance segment-based stereo matching algorithms have been developed, there are still rooms to solve mismatches between binocular images occurred in the textureless and disparity discontinuity regions [1]. In this paper, we focus on reduction of errors in textureless regions by exploiting global segmentation, since global information is less sensitive to locally ambiguous regions. We also consider local segmentation based on a quad-tree decomposition to deal with disparity discontinuity regions.

2. Global and Local Segmentation

The proposed method consists of three steps : merging homogeneous blocks to make global segments, generating local segments using a quad-tree decomposition, and calculating global and local disparities using the disparity space distribution.

Prior to segmentation, a bilateral smoothing filtering is applied to color images to remove noises. Bilateral filtering provides a consistent image while preserving color boundaries. In order to extract edges of a chrominance component, we employ the Canny algorithm. Since the chrominance component is more consistent over objects than the luminance component, its edges are more reliable to determine the homogeneous or discontinuous regions. To exaggerate the distribution of the chrominance component, we also use a histogram stretch algorithm. Figure 1 explains the edge extraction operation.



(a) Chrominance (b) Histogram Stretch (c) Edges

Figure 1. Edge Extraction.

In order to generate global segments, we divide the image into M x M blocks. When a block has no edges, we consider the block as a homogeneous region. For each block in the no edge region, we merge its 4-connected neighboring blocks iteratively. During merging, we avoid overlapping of previous blocks. We regard the merged blocks as global segments. We usually set M by 32 or 64. Although the size of the block is fixed, the merging procedure can find out homogeneous regions successfully. Figure 2 shows the merging process to make global segments.

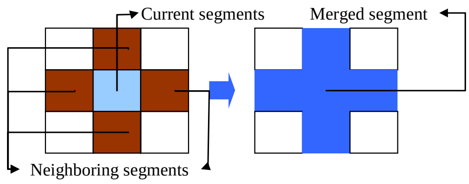


Figure 2. Merging Segments.

In order to generate local segments, we use the quad-tree decomposition. The decomposition splits the block including edges into smaller pieces, until the region of each piece is homogeneous [2]. As shown in Fig. 3, we set the maximum size of the block to M x M and the minimum size of the block to N x N. For each M x M block containing edges, we perform the quad-tree decomposition iteratively up to N x N blocks according to the presence of edges in each sub-block.

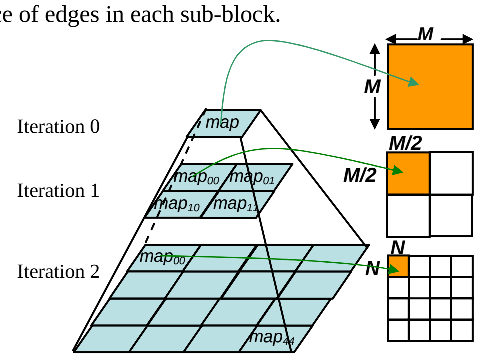
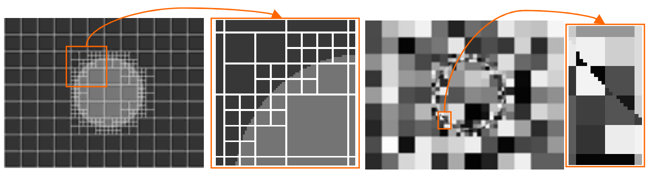


Figure 3. Quad-tree Map.

By the maximum and minimum size of blocks in the quad-tree decomposition, the iteration number, *iter*, is determined by

$$iter = log_2(M/2N) \qquad (1)$$

In general, the quad-tree map is a three-dimensional matrix whose size is $[iter][2^{iter}][2^{iter}]$.



(a) Quad-tree Decomposition (b) Local Segments

Figure 4. Local Segmentation.

As shown in Fig. 4(b), when the sub-block is in the boundary region which contains more than one object, we segment the sub-block into two segments based on the color difference. Since the block containing edges has already subdivided into the minimum blocks, it is enough to divide it into two segments. After color segmentation, we regard the decomposed blocks as local segments. Figure 4 shows the local segmentation operation.

3. Disparity Estimation

Since we segment the image into homogeneous regions, we assume that each segment has one disparity. In order to determine the disparity of each segment, we generate a disparity space distribution (DSD) for each segment [3]. We assign all pixels of each segment to the disparity with the maximum value in DSD. In order to calculate probabilities at every disparity for the segment, we find the corresponding matching region in the neighboring image by comparing stereoscopic images. We define the matching function by the largest sum of three continuous bins of the histogram.

After computing the disparity space distribution for global and local segments respectively, we refine disparities of homogeneous regions by comparing matching functions between global and local disparity for local regions. Finally, we assign the refined disparity value to each local segment.

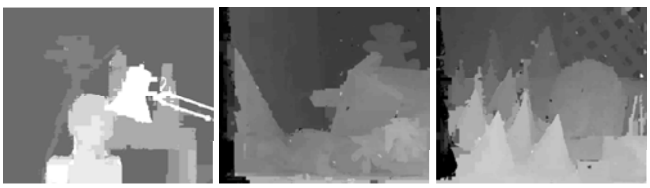
4. Experimental Results

We evaluated the performance of the proposed algorithm using three stereo images: “Tsukuba”, “Teddy”, and “Cones”. While Tsukuba is characterized by complicated indoor environments, Teddy and Cones are composed of slanted planes and multiple objects.

(a) SSD

(b) Dynamic Programming

(c) Graph Cuts



(d) Proposed

Figure 5. Results from Stereo Matching Algorithms.

The performance of our algorithm was confirmed by quantitative statistics based on the ground truth disparity map [4]. Bad matching means that the disparity is different more than one pixel with the corresponding ground truth disparity value. B_O, B_A, and B_D are defined as the percentage of bad matching pixels in non-occluded regions *O*, all regions *A*, and disparity discontinuity regions *D*, respectively.

We compared our algorithm to other stereo matching algorithms including the square sum of differences (SSD), dynamic programming, and graph cuts [4, 5, 6]. Figure 5 shows results from different stereo matching algorithms for the test data. We observed that the proposed scheme overcomes the problem of textureless regions. As shown in Fig. 5, SSD produced erroneous disparities around disparity discontinuity regions. Dynamic programming had horizontal streaks, especially in regions of low texture. Graph cuts carried out wrong disparity estimations for in-between objects, as shown in Fig. 5(a). Despite of the shortcomings of dynamic programming and graph cuts, they had good matching performance. Our proposed method showed the similar performance as graph cuts. Furthermore, we solved problems of in-between disparities. As a result, the proposed algorithm could enhance quality of the disparity map in textureless and disparity discontinuity regions.

Table 1 shows experimental results by our method and other stereo matching algorithms. Our hierarchical disparity estimation using the quad-tree decomposition had compatible results with other algorithms. Even though Cones suffered from the occlusion problems, the proposed method reduced the percentage of bad matching for disparity discontinuity regions.

Table 1. Performance Using Quantitative Statistics.

Algorithm	Tsukuba B_O B_A B_D	Teddy B_O B_A B_D	Cones B_O B_A B_D
Proposed	3.06 4.69 13.3	16.5 26.2 24.6	15.3 22.0 21.0
SSD	5.23 7.07 24.1	16.5 24.8 32.9	10.6 19.8 26.3
D.P.	4.12 5.04 12.0	14.0 21.6 20.6	10.5 19.1 21.1
Graph Cuts	1.94 4.12 9.39	16.5 25.0 24.9	7.70 18.2 15.3

5. Conclusion

In this paper, we have proposed a stereo matching algorithm formulating a hierarchical structure based on the quad-tree decomposition. The proposed scheme provides reasonable disparities for textureless regions by disparities of global segments. We also handle disparity discontinuity regions by exploiting a quad-tree decomposition and local segmentation.

Acknowledgments

This work was supported in part by MIC through RBRC at GIST, and in part by MOE through the BK21 project.

References

[1] J. Sun, N. Zheng, and H. Shum, “Stereo matching using belief propagation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 25, no. 5, pp.787-800, 2003.

[2] C. Faloutsos, H. Jagadish, and Y. Manolopoulos, “Analysis of the n-dimensional quadtree decomposition for arbitrary hyperrectangles,” *IEEE Trans. on Knowledge and Data Engineering*, Vol. 9, no. 3, pp.373-383, 1997.

[3] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-quality video view interpolation using a layered representation,” *SIGGRAPH*, pp.173-182, 2004.

[4] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame,” *International Journal of Computer Vision*, Vol. 47, no. 1, pp.7-42, 2002.

[5] A. Bobick and S. Intille, “Large occlusion stereo,” *International Journal of Computer Vision*, Vol. 33, no. 3, pp.181-200, 1999.

[6] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 23, no. 11, pp.1222-1239, 2001.