Disparity map determination by multiple adaptive windows

C. Demoulin and M. Van Droogenbroeck
Department of Electricity, Electronics and Computer Science,
Montefiore B28, Sart Tilman, B-4000 Liège, Belgium

Abstract— **The correspondence problem is one of main topics in stereo vision that, despites being studied for many years, is still in progress.**

In this paper, we present a new method that computes the disparity map. Our method is local (i.e. only information gathered in the close neighborhood is used) and is based on image statistics. More specifically it combines multiple adaptive windows and local statistical measures in order to optimize the quality of the computation of disparity maps. As the size of windows is critical for local methods, we propose an algorithm that modifies the size of windows surrounding the pixel of interest to capture enough information in regions with low texture energy. This process is based on statistical measures (mean and horizontal/vertical average deviation) taken in the windows of the original stereo images.

In the last section we show that our method performs very well compared to other existing local methods.

I. Introduction

Stereo matching algorithms have been developed for many stereoscopic applications such as 3D vision, teleconferencing, and reconstruction. One of the major problem in stereo matching is the correspondence problem. The objective is to determine a couple of pixels (p_L, p_R) (where L and R refer to the left and right images respectively) which are the projections of a point P in the real world.

Theoretically this is a 2D problem, but it can be reduced to a 1D problem by rectification. The rectification algorithm (see [4], [7] for rectification algorithms) applies a transformation matrix to the two images in order for them to be parallel to each other and at the same focal distance. Therefore, after rectification, p_L lies on the same line as p_R, but the distance between these pixels, called *disparity*, still has to be computed for all pixels. In the following we assume that all images have been rectified.

Methods computing the disparity maps have been categorized as either global or local. Moreover, some methods detect and use features, others do not depend on the image content. It is known that global methods based on features offer excellent performances. However they are computationally expensive and do not meet real-time requirements. For this purpose many alternatives have been proposed to lower the complexity at the price of performance degradation. For example some techniques only compute sparse disparity maps.

The method we propose estimates the complete disparity map. Since the problem is not left-to-right or right-to-left symmetric we have chosen the left image as a reference, and we are looking for the corresponding pixels in the right image. In addition our method is local and does not rely on features. The key idea is to evaluate the correlation between small windows surrounding a pixel of interest to enhance the quality of the disparity estimate.

The major challenges for our method, also met with other similar techniques, is to find the appropriate positions, sizes and shapes of the windows to have the most discriminant one. Many algorithms have been implemented during the last decade. In [8] Kanade and Okutomi proposed an algorithm that uses a single window with an adaptive size. A different scheme using two or more windows with fixed sizes was developed in [2], [5]. For each of them, the window with the smallest assignment cost is selected. Recently, these two principles (adaptive size and multiple windows) were mixed in [3]. Chan *et al*. have described an algorithm where nine windows are used. From these nine windows only the one with the lowest cost is selected and its size is reduced adaptively. Although offering good performances, this algorithm seems not to be very effective in flat regions like textureless areas.

Our scheme is somehow similar; it uses four windows surrounding the pixel of interest and, based on the evaluation of relevant statistical characteristics in the reference image, determines the best window.

This paper is organized as follows. In Section II we provide a description of our scheme. Section III details which statistics are chosen. In Sections IV and V we discuss the processes of the disparity determination and the aggregation. Section VI evaluates the performances, and Section VII concludes the paper.

II. Method overview

For all local methods, the most sensitive problems are to find the good window, and appropriate measures. Our method uses four rectangular windows surrounding the

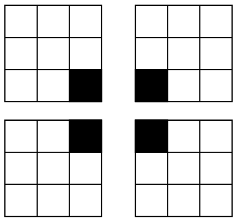

Fig. 1. Four windows used in our algorithm. The black rectangle represents the pixel of interest.

pixel of interest, as displayed in Figure 1, whose sizes are independently adapted.

Once the basic window shapes have been chosen, there still remain several steps described hereafter:
1. Determination of the window sizes in the reference image,
2. Decision on a cost function for each window,
3. Computation of a disparity map by costs comparison, and
4. Aggregation.

III. Determination of the window size

The discussion on the best window size is a difficult one. Windows with a fixed size are unable to handle borders (i.e. a local discontinuity) when the size is chosen to be large. Windows with a small size have the drawback of leading to poor performances for untextured areas.

Technically, in the first case the window covers regions with different disparities, and the resulting assignment cost is inappropriate. The use of multiple windows could potentially solve this problem as different windows will be able to cope with local inhomogeneities. For that reason, we use four windows.

In the second case (small sized windows), the cost calculated on the window could be the same for multiple locations in the second image because the level of intensity variations from one window to another is low. Only larger windows will capture enough pixel variations to be able to provide a satisfactory disparity estimate. On the one hand a window must be small enough to avoid any effects of projective distortion (as explained in [8]). But on the other hand larger sizes increase the computation times.

To determine the window size, we use two criteria :
- The value of the horizontal and vertial variances, and
- The value of the correlation cost.

The first criterion is developed in III-A, and the second one in III-B.

A. Statistical measures

We use rectangular windows with horizontal and vertical axes as most of the energy contained in an image is concentrated along these axes. We compute an estimate of the horizontal and vertical average deviations for each window. As pixel values are realizations of random variables or processes, depending on the assumptions made, we have no access to the real values, and one has to satisfy himself with an estimated value. The measures taken are the estimates of the mean, and the horizontal and average deviations defined as follows:
1. Horizontal average deviation:

$$\hat{E}_H = \frac{1}{W_H W_V} \sum_{m=-W_H/2}^{W_H/2} \sum_{n=-W_V/2}^{W_V/2} \| I_L(i+m,\, j+n) - \hat{\mu}_{j+n} \|$$

2. Vertical average deviation:

$$\hat{E}_V = \frac{1}{W_H\, W_V} \sum_{m=-W_H/2}^{W_H/2} \sum_{n=-W_V/2}^{W_V/2} \| I_L(i+m,\, j+n) - \hat{\mu}_{i+m} \|$$

where
- symbol $\wedge$ denotes an estimate,
- W_H and W_V represent the horizontal and vertical dimensions of the window,
- I_L and I_R designate the intensity pixel in the left and right images,
- i and j are the column and line coordinates of the center pixel of the window, and
- $\hat{\mu}_{j+n} = \frac{1}{W_H} \sum_{m=-W_H/2}^{W_H/2} I_L(i+m,\, j+n)$ and $\hat{\mu}_{i+m} = \frac{1}{W_V} \sum_{n=-W_V/2}^{W_V/2} I_L(i+m,\, j+n)$ are the means computed on a horizontal and vertical line respectively.

When the horizontal average deviation $\hat{E}_H$ is small, the window covers a region with low horizontal variation. Note that to raise the signal to noise ratio (in untextured area) during the correlation cost computation process, we have to increase the horizontal and/or the vertical window size.

B. Correlation cost

Next, to find the best window size and to compute the disparity associated to each window, we have to choose a cost function. After many experiments, we decided to use an extension of the SSD (Sum-Squared Difference) appropriate for adaptive windows, defined as follows :

$$C\left(I_L, I_R, W_H, W_V, d\right) = \frac{1}{(W_H\, W_V)^2} \sum_{m=-\frac{W_H}{2}}^{\frac{W_H}{2}} \sum_{n=-\frac{W_V}{2}}^{\frac{W_V}{2}} [I_L(i+m,\, j+n) - I_R(i+m-d,\, j+n)]^2$$

where we assume that the image intensities of the projection p_L and p_R of a real point P are the same, and where d denotes the disparity.

This cost function has two properties :

- When the window covers a low-textured region, the numerator is low, and the cost gives the priority to large windows,
- But, when the window covers regions of various disparities, the numerator is high, and the cost gives the priority to smaller window.

C. *Window size determination*

The sizes of each window are determined as follows : the growing process lasts until $\hat{E}_H > \frac{T}{W_H W_V}$. The same principle yields for $\hat{E}_V$. But we immediately stop the growing process if the cost computed for the new window size is higher than the one computed for the previous window size. The last case occurs when the window covers region of various disparities.

IV. Computation of the disparity maps

To determine the disparity of the pixel of interest, we take the disparity associated to the window with the smallest cost C.

V. Aggregation

The disparity map contains some gaps. As in [10], we use a median filter with a 5×5 rectangular window. This introduces a kind of inter-lines consistency in the process.

VI. Experiments

To evaluate our method, we have used a benchmarking tool found on the Middlebury Stereo Vision web page [1]. This site provides left, right, and groundtruth images of scenes containing untextured areas (see [10] for more information).

As suggested on this site, we have tested our algorithm on four images (*tsukuba*, *sawtooth*, *venus*, and *map*) with constant parameter settings across all four images. These parameter values are:

1. $T = 1440$.
2. W_H and W_V are initialized at 5. These values are chosen to guarantee that the statistical parameters are computed on a population which is large enough to obtain significant statistics.
3. horizontal and vertical sizes are upper bounded to 30 pixels.

Figure 2 shows the results obtained on *tsukuba* and Figure 3 those obtained with *sawtooth*.

As can be seen, the algorithm performs well in untextured regions. On the other hand there are some errors in

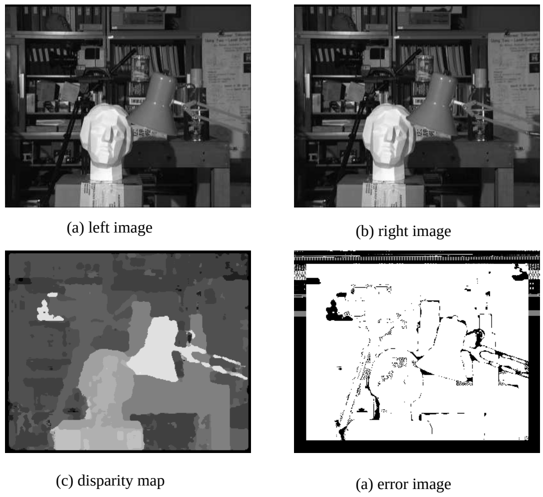

Fig. 2. Results obtained on *tsukuba*: (a) left original image, (b) right image, (c) disparity map obtained after aggregation, and (d) error image. Pixels drawn in black in the error image indicate that the difference between the real disparity (found in [1]) and the disparity computed is larger than 1.

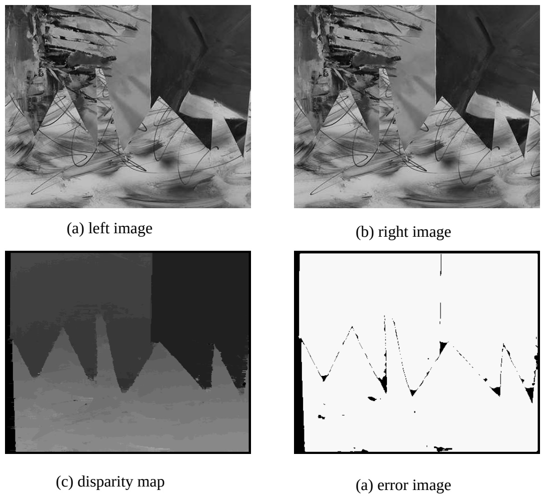

Fig. 3. Results obtained on *sawtooth*: (a) left original image, (b) right image, (c) disparity map obtained after aggregation, and (d) error image. Pixels drawn in black in the error image indicate that the difference between the real disparity (found in [1]) and the disparity computed is larger than 1.

regions containing two untextured regions separated horizontally. In this particular case, the adaptive process stops although it should continue when two regions have an equal disparity. A simple solution to this problem consists in imposing the ordering constraint.

We have also compared our algorithm to other existing methods. In the table available on the web site, our method comes at the $16th$ place out of 35 algorithms (these results were obtained on the 7th of January 2005). But we must mention that many algorithms in this list use global meth-

	Tsukuba		
Algorithme	all	untex.	disc.
[13]	1.51 1	0.65 1	7.22 1
YOUR METHOD	3.15 3	2.81 3	8.93 2
[12]	2.35 2	1.65 2	12.17 3
[11]	3.36 4	3.54 4	12.91 5
[6]	4.25 7	4.47 8	15.05 7
[14]	3.49 5	3.65 5	14.77 6
[10][1b]	6.49 9	11.62 9	12.29 4
[9]	3.95 6	4.08 7	15.49 8
[10][1c]	5.23 8	3.80 6	24.66 9

	Sawtooth		
Algorithme	all	untex.	disc.
[13]	1.15 1	0.29 3	5.47 2
YOUR METHOD	1.25 2	0.07 1	5.40 1
[12]	1.28 3	0.23 2	7.09 3
[11]	1.61 6	0.45 5	7.87 4
[6]	1.32 4	0.35 4	9.21 5
[14]	2.03 7	2.29 9	13.41 8
[10][1b]	1.45 5	0.72 6	9.29 6
[9]	2.45 9	0.90 8	10.58 7
[10][1c]	2.21 8	0.72 7	13.97 9

Fig. 4. Comparative tables of local method algorithms. The comparison is shown for two images: Tsukuba and Sawtooth. For each image, the first column corresponds to all the pixels, the second to the untextured areas, and the third to borders. The numerical results represent the percentage of "bad" pixels, i.e. pixels whose absolute disparity error is larger than 1.

ods and not local methods. To have a fair comparison, we have considered only local methods. After this restriction to local methods, the comparison is given in Figure 4.

The comparison is shown for two images: Tsukuba and Sawtooth. For each image, the first column corresponds to all the pixels, the second to the untextured areas, and the third to borders. The numerical results represent the percentage of "bad" pixels, i.e. pixels whose absolute disparity error is larger than 1. From these tables it can be seen that our algorithm reaches the second place, but it improves on the best method for pixels near depth discontinuities.

VII. Conclusions

We have presented a new area-based method which uses four windows. Their sizes are determined by an adaptive process parametrized by several local statistics taken on the reference image. After having detailed our method, we have used a benchmarking tool that shows that it performs very well compared to other local methods.

For future work, we may impose the ordering constraint to further improve the quality of computed disparity maps.

References

[1] Middlebury stereo vision page. `http://www.middlebury.edu/stereo`.

[2] A. Bobick and S. Intille. Large occlusion stereo. *International Journal on Computer Vision*, 33(3):181–200, September 1999.

[3] S. Chan, Y. Wong, and J. Danie. Dense stereo correspondence based on recursive adaptive size multi-windowing. In *Image and Vision Computing New Zealand 2003*, volume 1, pages 256–260, Massey University, Palmerston North, New Zealand, 2003. November 26-28.

[4] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[5] Davi Geiger, Bruce Ladendorf, and Alan Yuille. Occlusions and binocular stereo. *International Journal on Computer Vision*, 14(3):211–226, April 1995.

[6] H. Hirschmuller. Improvements in real-time correlation-based stereo vision. In *Proceedings of IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 141–148, Kauai, Hawaii, December 2001.

[7] F. Isgro and E. Trucco. Projective rectification without epipolar geometry. In *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 94–99, Fort Collins, Colorado, USA, June 23-25 1999.

[8] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window : Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intellignece*, 16(9):920–932, September 1994.

[9] S. H. Lee, Y. Kanatsugu, and J.-I. Park. Hierarchical stochastic diffusion for disparity estimation. In *Computer Vision and Pattern Recognition 2001*, 2001.

[10] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal on Computer Vision*, 47(1/2/3):7–42, April-June 2002.

[11] O. Veksler. Stereo matching by compact windows via minimum ratio cycle. In *International Conference on Computer Vision 2001*, 2001.

[12] O. Veksler. Fast variable window for stereo correspondence using integral images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 556–565, Madison, Wisconsin, June 18-20 2003.

[13] K.-J. Yoon and I.-S. Kweon. Locally adaptive support-weight approach for visual correspondence search. In *Computer Vision and Pattern Recognition 2005*, 2005.

[14] L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. In *Pattern Analysis and Machine Intelligence 2000*, 2000.