Robust Stereo Aggregation with Large Windows

Peer-reviewed author version

GERRITS, Mark & BEKAERT, Philippe (2006) Robust Stereo Aggregation with Large Windows. In: Proceedings, IEEE Workshop on Content Generation and Coding for 3D-Television..

Handle: http://hdl.handle.net/1942/1721

Robust Stereo Aggregation with Large Windows

Mark Gerrits and Philippe Bekaert
Hasselt University
Expertise Centre for Digital Media
Transnationale Universiteit Limburg
Wetenschapspark 2, 3590 Diepenbeek, Belgium
{mark.gerrits, philippe.bekaert}@uhasselt.be

Abstract *We present a new window-based stereo matching algorithm which focuses on robust outlier rejection during aggregation to allow for windows of arbitrary size. Working from the assumption that depth discontinuities occur at colour boundaries, we segment the reference image and consider all window pixels outside the image segment that contains the pixel under consideration as outliers and greatly reduce their weight in the aggregation process. We developed a variation on the recursive moving average implementation to keep processing times independent from window size. Together with a robust matching cost and the combination of the left and right disparity maps, this gives us a robust local algorithm that approximates the quality of global techniques without sacrificing the speed and simplicity of window-based aggregation.*

1. Introduction

The stereo correspondence problem is an important challenge in computer vision. Much work is increasingly being done on stereo algorithms that produce dense disparity maps, as these can be used for view synthesis and video based rendering. A thorough survey and taxonomy of dense stereo techniques was provided by Scharstein and Szeliski[10].

Fast area-based approaches focus mostly on the aggregation step but run into problems when deciding the window size to be used during cost aggregation. Small windows do not contain enough information and lead to noisy results, while large windows contain enough texture information but encompass pixels at different depths near depth discontinuities, which leads to overblown foreground objects, as pixels near depth discontinuities become bimodal and will display a strong preference towards the foreground disparity, even if they are part of the background (the foreground fattening effect[10]). In this paper we present a method to avoid this through segmentation-based outlier rejection.

2. Related Work

In 2001, Scharstein and Szeliski published a taxonomy and evaluation of dense stereo algorithms [10]. This work further illustrated the intuitive notion that while local techniques excell at achieving high speeds, global techniques are better suited to generate high quality disparity maps. Consequently, most recent work has focused on developing global algorithms. But significant work has also been done on local methods.

Adaptive-window methods change the size and shape of their window adaptively for each pixel. Kanade and Okutomi [6] evaluated the local variation of intensity and disparity at each pixel to select an appropriate window. Their window shape was limited to rectangles and therefore ran into problems near arbitrarily shaped depth discontinuities. The method was also computationally expensive and relied heavily on a sufficiently accurate initial disparity estimation. To improve performance, multiple-window methods [3][7] use a small number of predefined windows amongst which they choose the optimal one.

By assigning different support-weights to different pixels in the window, Prazdny [9] and Xu et al. [12] tried to overcome this problem. The former assigned weights to neighbouring pixels iteratively while the latter used radial computations. Both these methods are dependent on an initial disparity estimation, which needs to be accurate enough. Yoon and Kweon [13] eliminated this reliance by using a non-iterative approach. They based their weights on the photometric and geometric relationship with the pixel under consideration. They achieved good results but at a high computational cost. Their technique was also susceptible to image noise.

In recent years, segmentation-based techniques have proven adept at correctly handling edges. Though they often come at a computational cost, they have proven to be some of the highest quality algorithms to date.

Tao et al.[11] proposed an analysis-by-synthesis method to maintain depth discontinuities. A reference image is segmented based on color and each image segment is then iteratively warped to the other views. The depth within an image segment is assumed to be smooth and representable by a plane-plus-parallax model.

Zhang and Kambhamettu[14] presented a stereo matching algorithm with integrated 3D scene flow computation. The algorithm consists of a hierarchical rule-based matching scheme employing color segmentation to enforce depth discontinuities. Scene flow is estimated via an energy minimization procedure and later applied as constraints on the depth estimation to make it more accurate and robust.

3. Algorithm

3.1 Overview

We start out by applying a robust function to our per-pixel matching costs to reduce the influence of all outlier pixels. This gives us the disparity volume to aggregate over.

During aggregation, all pixels inside the window whose disparities differ greatly from the central pixel under consideration, should be considered as outliers. But it is exactly these disparities we are trying to determine. We solve this problem by making the assumption that depth discontinuities occur across colour discontinuities and use a segmentation of the image to ignore outliers.

Finally we improve our results by combining the left and right disparity maps.

3.2 Robust Matching Costs

The need for robust matching costs becomes clear when we look at the problem as one of pure outlier contamination. After all, when aggregating the matching costs, pixels with very high matching costs will disrupt the average, especially near depth discontinuities where they will exert too much influence. Scharstein and Szeliski[10] noted this and experimented with truncated matching costs, which provided a small improvement. We chose to use the Geman-McClure function[4], a proven technique to handle outliers:

$$\rho(x) = \frac{x^2}{x^2 + \sigma^2}$$

Beyond a certain point, determined by σ, its influence begins to descend and smoothly converges to zero. The transformed matching cost $\rho(x)$ converges to 1. Therefore, no matter how large the raw costs become, after applying Geman-McClure, they will never exceed 1.

3.3 Segmentation Based Outlier Rejection

We assume that depths vary smoothly within any image segment with homogeneous colour. Based on this assumption, we can disregard or diminish the influence of those pixels within the aggregation window that fall outside the image segment which contains the central pixel under consideration. We use Comaniciu and Meer's mean shift algorithm [2] to segment the reference image.

Unlike other segmentation based techniques, we do not impose that all pixels in the same segment must share the same depth or lie on a simple, locally fitted surface such as a plane. Instead, we use the segmentation as a guide for robust aggregation. Ideally, any pixels outside the image segment should be considered outliers. However, these outlier pixels are not completely ignored in our aggregation but receive a small weight λ compared to the pixels inside the image segment. We do this to protect our algorithm from oversegmentation artefacts. By weighing the pixels outside of the window with a small weight, we can aggregate enough information in highly textured areas while still remaining accurate around depth discontinuities.

Because we aggregate across a window and thus not necessarily across all pixels in a segment, we are protected from some artefacts of undersegmentation, where depths from one object will cut into another object because an image segment crosses an object boundary. In the hypothetical worst-case scenario where the whole image is one big segment, our technique will still score equally well as the normal aggregation technique combined with the Geman-McClure function, whereas other segmentation based techniques would run into severe problems.

3.4 Disparity Map Combination

To improve the accuracy of our results, we calculate a depth image for both stereo images and combine them to eliminate some final oversegmentation artefacts. Depending on which view we are computing the disparity map for, we will warp the other disparity map back to this view. Undersegmentation faults will lead more frequently to overly high disparities than overly low disparities (because of the foreground fattening effect). Therefore, assuming that any undersegmentation fault only occurs in one of the two views, we take the minimum of both disparity maps.

This technique has the added advantage of improving disparities in occluded areas, as pixels in these areas will usually have too high disparities as they try to move out from under the occluding object to match with similar pixels in the background object.

3.5 Implementation

Most window-based aggregation techniques thank their high performance speeds to the fact that they can be implemented as recursive moving average filters with running times independent of the window size.

While our segmentation based outlier rejection allows for windows of arbitrary size without suffering from the foreground fattening effect, the recursive moving average

(a) Ground truth (b) SSD + MF from [10] (c) State of the art, [1] (d) Our method

Figure 1: Comparison of our technique with others

filter implementation will no longer work in this case. When aggregating the disparity rows in the classic recursive implementation of the moving average filter, the aggregated value A_{i+1} for pixel $i+1$ equals A_i - $C_{i-w/2}$ + $C_{i+w/2}$, where C_x is the matching cost at pixel x. Unfortunately, when working with segments, pixels i, $i+1$, $i-w/2$ and $i+w/2$ can all fall in different segments.

Algorithm 1 Segmented Moving Average

1. For each row:
   (a) For each segment s: T_s = 0
   (b) For each pixel i in row j:
      i. $T_{s_{i+w/2,j}} = T_{s_{i+w/2,j}} + C_{i+w/2,j}$
      ii. $T_{s_{i-w/2,j}} = T_{s_{i-w/2,j}} - C_{i-w/2,j}$
      iii. $A^r_{i,j} = T_{s_{i,j}}$
      iv. $A^s_{i,j} = A^s_{i-1,j} + C_{i+w/2,j} - C_{i-w/2,j}$
2. For each column:
   (a) For each segment s: T_s = 0
   (b) t = 0
   (c) For each pixel j in column i:
      i. $T_{s_{i,j+w/2}}$ = $T_{s_{i,j+w/2}}$ + $A^s_{i,j+w/2}$
      ii. $T_{s_{i,j-w/2}}$ = $T_{s_{i,j-w/2}}$ - $A^s_{i,j-w/2}$
      iii. t = t + $A^r_{i,j+w/2}$ - $A^r_{i,j-w/2}$
      iv. $A_{i,j}$ = $\lambda \times (t - T_{s_{i,j}}) + T_{s_{i,j}}$

A brute force implementation of the segmentation-based moving average filter would be far too slow to be of any practical use. Therefore we developed a variation on the moving average algorithm, so that our aggregation speeds are again independent from the window size, allowing us to use large windows without any speed penalties.

Our solution is explained in simplified form in Algorithm 1. Trivial precautions that need to be taken at the borders of the image are left out for clarity. For each segment s, we keep track of a running average T_s. As the edges of our aggregation interval move through different segments, we update the corresponding averages. To find the aggregated cost A^r of the central pixel in the interval, we simply check which segment the pixel falls into and look up its average. At the same time, we also perform regular recursive moving average computation (A^s) so we can combine the aggregated value inside the segment with the aggregated value outside the segment, weighed by a factor λ.

4. Results

Figure 1 and Table 1 show our result on the Tsukuba data compared to other techniques. Figure 2 shows the result of our technique on two other standard Middlebury data sets. Even though our algorithm is significantly faster and less complex, our results approach those of global correspondence techniques. Calculating the final disparity map with 51×51 windows and $\lambda = 0.01$ for the Tsukuba stereo pair took 1.26 seconds in a C++ implementation on a 3 GHz Pentium 3 computer. Approximately 35% of that time was spent on segmentation, 20% on calculating the per-pixel matching costs, 25% on aggregation and 15% on combining the two disparity maps.

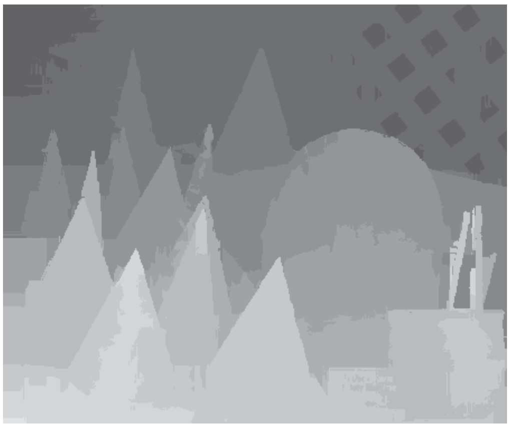
Algorithm	Tsukuba	Venus	Teddy	Cones
Segm+visib [1]	1.57	1.06	6.54	8.62
AdaptWeight [13]	1.85	1.19	13.3	9.79
GC+occ [8]	2.01	2.19	17.4	12.4
Our method	2.27	1.22	19.4	17.4
Reliablty-DP [5]	3.39	3.48	16.9	19.9
GC [10]	4.12	3.44	25.0	18.2
SSD+MF [10]	7.07	5.16	24.8	19.8

Table 1: Percentage of badly labeled disparities of several techniques, including ours, on the Middlebury test case

5. Conclusion

In this paper, we have shown how local, window based stereo aggregation can be performed with arbitrarily sized windows without suffering from the foreground fattening effect. We used a combination of robust matching costs

(a) Teddy, 30 disparities



(b) Cones, 16 disparities

Figure 2: Results of our algorithm on some of the Middlebury datasets

based on the Geman-McClure function, and segmentation-based outlier rejection. We developed a variation on the recursive moving average filter to keep running times independent of the window size. By combining the left and right disparity map, we further improved our results.

Using these techniques, we approach the results of global methods without sacrificing the simplicity, flexibility and speed of local aggregation methods.

Acknowledgements The authors acknowledge financial support on a structural basis from the ERDF (European Regional Development Fund), the Flemish Government and the Flemish Interdisciplinary institute for BroadBand Technology (IBBT), and from a research grant by the EU (IST-2-511316-IP ”Racine-IP”)

Furthermore we would like to thank Tom Mertens for providing the impetus for this research and Tom Haber and Cedric Vanaken for their help.

References

[1] M. Bleyer and M. Gelautz. A layered stereo algorithm using image segmentation and global visibility constraints. *ICIP*, 2004.

[2] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 2002.

[3] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997.

[4] S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of International Statistical Institute*, 1987.

[5] M. Gong and Y.H. Yang. Near real-time reliable stereo matching using programmable graphics hardware. *CVPR*, 2005.

[6] T. Kanade and M. Okutomi. A stereo matching algorithm with and adaptive window: Theory and experiments. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(9), 1994.

[7] S. B. Kang, R. Szeliski, and C. Jinxjang. Handling occlusions in dense multi-view stereo. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1, 2001.

[8] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. *ICCV*, 2001.

[9] K. Prazdny. Detection of binocular disparities. *Biological Cybern*, 52, 1985.

[10] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47, 2002.

[11] H. Tao, H. Sawhney, and R. Kumar. A global matching framework for stereo computation. *In International Conference on Computer Vision*, 1, 2001.

[12] Y. Xu, D. Wang, T. Feng, and H.Y. Shum. Stereo computation using radial adaptive windows. *Proc. Int'l Conf. Pattern Recognition*, 3, 2002.

[13] Kuk-Jin Yoon and In-So Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[14] Y. Zhang and C. Kambhamettu. Stereo matching with segmentation-based cooperation. *ECCV*, 2002.