

Efficient Disparity Computation without Maximum Disparity for Real-Time Stereo Vision

Christian Unger¹

<http://campar.in.tum.de/Main/ChristianUnger>

Selim Benhimane³

<http://www.vision.inst.ac.uk/~pp>

Eric Wahl¹

Eric.Wahl@BMW.de

Nassir Navab²

<http://campar.in.tum.de/Main/NassirNavab>

¹ BMW Group

Munich, Germany

² Chair for Computer Aided Medical Procedures

Technische Universität München
Garching, Germany

³ Metaio GmbH

Munich, Germany

Abstract

In order to improve the performance of correlation-based disparity computation of stereo vision algorithms, standard methods need to choose in advance the value of the maximum disparity (MD). This value corresponds to the maximum displacement of the projection of a physical point expected between the two images. It generally depends on the motion model, the camera intrinsic parameters and on the depths of the observed scene.

In this paper, we show that there is no optimal MD value that minimizes the matching errors in all image regions simultaneously and we propose a novel approach of the disparity computation that does not rely on any a priori MD. Two variants of this approach will be presented. When compared to traditional correlation-based methods, we show that our approach improves not only the accuracy of the results but also the efficiency of the algorithm. A local energy minimization is also proposed for fast refinement of the results.

An extensive comparative study with ground truth is carried out on classical stereo images and the results show that the proposed method clearly gives more accurate results and it is two times faster than the fastest possible implementation of traditional correlation-based methods.

1 Introduction

Dense stereo matching in real-time is important for many fields of applications that require an on-line dense three-dimensional representation of the observed scene [8, 22]. Also the processing of large images or long image sequences needs computationally efficient algorithms [10]. However, for automotive and other mobile applications, the hardware requirements must be as low as possible. This usually restricts the available processing power as well as the number of cameras.

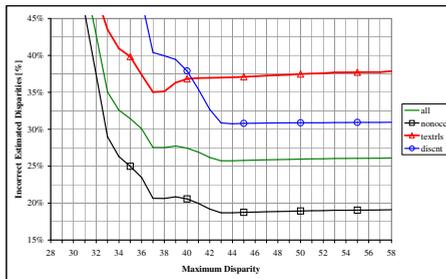


Figure 1: Curves plotting the percentage of incorrect disparities estimated with classical methods on the different image regions of the Teddy dataset as a function of the maximum disparity value used. The curves show that the maximum disparity needs to be set according to the region type of the image. When considering all pixels (all), the optimal value of the maximum disparity is equal to 43. The same value is optimal when considering non occluded regions (nonocc) and regions close to discontinuities (discnt). However, this value is not optimal when considering textureless regions (textrls) where the value 37 provides a lower percentage of wrong disparities.

Typical commercial implementations of such systems use one or two cameras together with a computationally feasible algorithm to compute depth information [8, 22]. If two cameras are used, local methods based on correlation can be implemented very efficiently [12]. On dedicated hardware, methods such as local correlation, dynamic programming, semi-global matching or even belief propagation can be implemented for real-time application [6, 8, 18, 24]. However, optimized local methods are among the fastest ways in order to perform dense matching solely on general purpose CPUs without special hardware. In this case, decisions must be made upon the values of some parameters, particularly for the maximum disparity (MD).

We show that the choice for a fixed MD influences the quality of the depth-map: setting it too high introduces false matches and setting it too low will produce gross errors at close objects. But also a seemingly ideal value will not result in the best possible result. The relationship between MD and matching errors is shown in Fig. 1 for the standard dataset Teddy¹. The figure depicts that there is no optimal fixed MD setting that minimizes all individual errors at once. For example, if the value 37 is used the errors of textureless regions are minimized but this value will cause higher errors in the other regions. The optimum may be obtained if the MD is set to 37 in textureless regions and to 43 in the rest of the image. In general, the MD should be variable and as close as possible to the true disparity since a higher MD only increases the possibility of false matches, especially in regions with weak texture.

In some cases, the choice of MD is complicated. Generally, the disparity is proportional to the distance between camera centers and the focal length and it is inverse proportional to the depth of the point. Especially at motion-stereo [15], a choice for a fixed MD either restricts the practical applicability (the minimally allowed depth of the points is a function of the camera interframe displacement) or results in an increased number of errors and an inefficient use of processing power (if the MD is set too high).

We propose a novel method for efficient dense stereo matching without the need of

¹<http://vision.middlebury.edu/stereo/>

the choice of MD, which is based on local correlation. Moreover, we present a fast post-processing technique that is based on energy minimization and is suited to refine the obtained results.

We successfully reduce the matching errors compared to traditional local correlation since our approach uses locally a minimal MD. Furthermore, our approach is simple to implement and the iterative search technique might be integrated into other stereo matching algorithms as well. Finally, our proposal is even faster than the fastest possible implementation of local correlation with integral images [7, 23]. That is because we significantly reduce the number of required correlations which ultimately saves processing time.

1.1 Related Work

A lot of work has been done in the field of dense stereo matching. Traditional *local correlation-based methods* [8] can be implemented very efficiently [7, 17, 23]. Several techniques, for example [12], have been introduced to improve the quality of these methods but are sometimes quite time consuming [26].

In *Graph Cuts* [5, 14] and *Belief Propagation* [9, 13, 21] stereo matching is formulated as an energy minimization problem, where a function is computed that minimizes a global energy functional. Both methods produce really good results, but are relatively slow. *Dynamic Programming* [20] and *Scanline Optimization* [16, 19] reduce the computational effort by processing individual scanlines. However, this may lead to streaking effects and most formulations of dynamic programming require an ordering constraint to be fulfilled [4]. Another efficient and promising idea is *Semi-Global Matching* [10, 11, 18], where a minimum of a global energy functional is estimated using local computations.

Recently, region-based methods [2, 3, 25] have received great success. These methods first segment the input image and then estimate the correspondence between regions rather than pixels.

Most of those methods can be implemented very efficiently, and can even be used in real-time applications if they are running on dedicated hardware [6, 8, 18, 24]. They provide results with a very high quality, but need to set the MD in advance to preserve the real-time performance. However, we will concentrate on approaches that can be used for real-time applications with no dedicated hardware and without a priori knowledge about the MD.

2 Traditional Disparity Computation

In traditional correlation-based methods, to each pixel $\mathbf{p} = (x, y)^T$, the disparity associated with the minimum cost is assigned:

$$\mathcal{D}(\mathbf{p}) = \arg \min_{0 \leq d \leq d_{max}} E(\mathbf{p}, d) \quad (1)$$

where \mathcal{D} is the depth-map, d_{max} is the MD and $E(\mathbf{p}, d)$ is the dissimilarity (lower values indicate higher similarity). We consider E as a matching cost summed over a support region around the pixel:

$$E(\mathbf{p}, d) = \sum_{u=-w}^w \sum_{v=-w}^w C_0(\mathbf{p} + (u, v)^T, d) \quad (2)$$

$C_0(\mathbf{q}, d)$ is for example the squared intensity difference (SD) or the absolute intensity difference (AD). For some separable formulations of E (especially for the sum of absolute in-

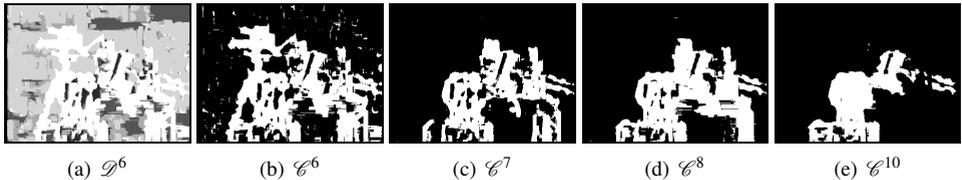


Figure 2: Depth-maps computed with wrong values for MD using the Tsukuba dataset (see also Fig. 4). One depth-map with $d_{max} = 6$ (a) and four thresholded depth-maps with $d_{max} \in \{6, 7, 8, 10\}$, thresholded using d_{max} (b-e). In general, \mathcal{C}^n is computed from \mathcal{D}^n .

tensity differences (SAD)), its values can be computed very efficiently using integral images [23].

3 Disparity Computation without MD

We introduce a novel algorithm for disparity computation, which does not require a beforehand specified MD. Our proposal is particularly suited for usage in real-time correlation-based stereo methods. We consider that pairs of images are rectified, so that corresponding epipolar lines are horizontal and that pixel coordinates are integer valued – i.e. $(x + 1, y)$ and $(x, y + 1)$ are neighbouring pixels of (x, y) .

3.1 Basic Idea

We depict our very initial ideas in the hope that they are of further scientific use. However, this subsection is not mandatory for understanding the rest of the paper. Our early idea originates from thresholded depth-maps computed with intentionally wrong values for MD (see Fig. 2): we computed depth-maps using different values for the MD d_{max} using local correlation and extracted all pixels whose disparity is equal to d_{max} (a “thresholded depth-map”). Similar to Fig. 2 we observed that there are “regions” being present among many thresholded instances. In our first analyses we focussed on tracking these regions: We first performed stereo correspondence with some small MD d_{max}^0 . Then we extracted the set of pixels \mathcal{C} whose disparity is equal to the MD d_{max}^0 . By assuming that the set of pixels with a disparity greater than d_{max}^0 is close to \mathcal{C} , we were able to formulate an approximate criterion to compute pixels that may have disparity $d_{max}^0 + 1$. We continued this process by incrementing the MD $d_{max}^{n+1} = d_{max}^n + 1$ iteratively until no pixel was assigned the current MD.

This formulation, although able to greatly reduce the number of required matching cost computations, is not able to compete with the traditional method in terms of execution time, due to an inefficient conceptual layout: according to [17], the innermost loop should run over the domain of disparity values and not over the domain of coordinates. Fortunately, we were able to derive a superior approach which is described below.

3.2 Algorithm Description

Instead of determining disparities by a brute-force search within the whole disparity domain, we focus on an iterative algorithm that stops at the right value. We perform two operations at each pixel: a minimization followed by a propagation-step. The minimization basically

follows a line search strategy and allows us to find the “next” local minimum of the matching cost function. Since matching costs have many local minima we introduce a propagation-step in order to find further, more optimal minima using disparity values of neighbouring pixels. We embedded these steps into a hierarchical setup, which will also be described in detail.

Minimization For every pixel \mathbf{p} , we determine an *intermediate optimal* disparity:

$$\mathcal{D}(\mathbf{p}) \mapsto \min \{d \mid d \in \mathbb{N}_0, d \geq \mathcal{D}(\mathbf{p}), E(\mathbf{p}, d+1) \geq E(\mathbf{p}, d)\} \quad (3)$$

Using this formula, we “step down the hill” and thus search for the next optimal disparity by iteratively incrementing the current disparity.

Propagation As the minimization-step will only return the first (and possibly not optimal) minimum of the dissimilarity function (see Fig. 3(a)), we additionally propagate the disparities of adjacent pixels:

$$\mathcal{D}(\mathbf{p}) \mapsto \arg \min_{d \in N(\mathbf{p})} E(\mathbf{p}, d) \quad (4)$$

with the neighbouring disparities $N(\mathbf{p})$. $N(\mathbf{p})$ should at least contain the disparities of the left and right neighbours and must fulfill $\mathcal{D}(\mathbf{p}) \in N(\mathbf{p})$. Disparity values will be propagated through their local neighbourhood in this step. This is the reason why the minimization- and propagation-steps should be run alternately. In practice, only a few iterations are required (2-4).

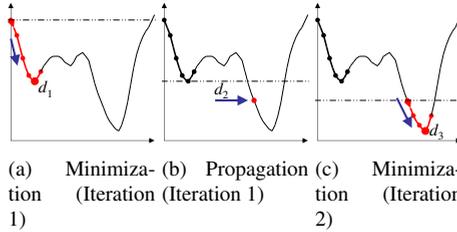


Figure 3: An example for a dissimilarity function (vertical axis) over disparity (horizontal axis). (a) Only the first minimum d_1 of the function is found by minimization. (b) Then, the adjacent disparity d_2 is propagated since it further reduces the dissimilarity. (c) Another minimization finds the optimal minimum d_3 .

Hierarchical Setup A hierarchical implementation stabilizes execution times, because disparities can be found with an almost constant effort. However, it must be noted that a hierarchical setup may obey the drawback of losing thin foreground objects and that errors at low resolutions may have severe impacts at higher resolutions. To minimize artifacts from false matches at low resolutions, we apply the hierarchical approach only to the horizontal dimension. Basically, at every resolution, the depth-map is initialized with the scaled up disparities from the previous resolution (in the beginning, the depth-map is initialized with zeros). For our algorithm, it is important to scale up disparities properly. Since the search direction of the minimization-step is in positive disparity orientation, it is beneficial to *underestimate* the

actual disparity. Let σ be the scale factor, for example $\sigma = 2$. For scaling up, we use the following formula:

$$\mathcal{D}'(\sigma x, y) = \sigma \mathcal{D}(x, y) - \sigma + 1 \quad (5)$$

$$\mathcal{D}'(\sigma x + k, y) = \sigma \min(\mathcal{D}(x, y), \mathcal{D}(x + 1, y)) - \sigma + 1 \quad \text{with } k = 1, \dots, \sigma - 1 \quad (6)$$

To summarize, the minimization- and the propagation-step is applied to every pixel of the image/scanline. This process is repeated until the disparities reach a fixed point. Then, the whole procedure is applied to the next resolution using the scaled up disparities.

4 Local Energy Minimization for Refinement

Local correlation-based methods produce errors in regions near depth discontinuities [12, 23] due to the assumption of constant disparity within the support region. This assumption is violated at object borders. There are techniques to reduce such errors but they do not eliminate them completely and are sometimes hindering for real-time application, in terms of execution time. The most effective remedy is to abandon matching windows, and to use pixelwise matching. However, to treat instabilities caused by pixelwise matching [1], many global methods minimize an energy functional, such as

$$E(\mathcal{D}) = E_D(\mathcal{D}) + \lambda E_S(\mathcal{D}) \quad (7)$$

where E_D measures how well the depth-map \mathcal{D} matches with the input images and E_S is a so called smoothness term, penalizing disparity variations [19].

Based on the assumption that the depth-map of a local method is a rough estimate of the ideal solution, we focus on enhancing a previously computed depth-map. To maximize the efficiency, we perform a winner-take-all optimization at every pixel (which is different to scanline optimization [19]).

The general idea is to propagate disparities through their neighbourhood. This way, the computational scheme is similar to the approach presented in section 3.2.

Algorithm For every pixel $\mathbf{p} = (x, y)^T$, we determine the best matching disparity value:

$$\mathcal{D}(\mathbf{p}) \mapsto \arg \min_{d \in N(\mathbf{p})} C(\mathbf{p}, d) \quad (8)$$

with the neighbouring disparities $N(\mathbf{p})$ as defined in section 3.2. Our pixelwise matching cost is defined as:

$$C(\mathbf{p}, d) := C_0(\mathbf{p}, d) + \tau(\mathbf{p})\rho(d - \mathcal{D}(\mathbf{p} - \mathbf{r}_x)) + \tau(\mathbf{p})\rho(d - \mathcal{D}(\mathbf{p} - \mathbf{r}_y)) \quad (9)$$

with

$$\rho(t) := \begin{cases} 0 & t = 0 \\ P_L & |t| = 1 \\ P_H & \text{otherwise} \end{cases} \quad \tau(\mathbf{p}) := \begin{cases} \gamma & \Delta \mathcal{I}(\mathbf{p}) > \Theta_{\mathcal{I}} \\ 1 & \text{otherwise} \end{cases} \quad (10)$$

C_0 is the (possibly truncated) absolute intensity difference or Birchfield and Tomasi's sampling invariant dissimilarity [1]. The parameters \mathbf{r}_x and \mathbf{r}_y point to the previously processed

pixel/scanline (for example, if the image scanlines are processed from bottom to top, \mathbf{r}_y may be set to $(0, -1)$). In (9) we use only two neighbours in order to avoid that depth discontinuities are penalized twice. The penalties P_L and P_H should be chosen such that $P_L < P_H$ to improve the recovery of slanted surfaces. The function τ helps to align discontinuities to intensity edges if $0 < \gamma < 1$ because it lowers the penalty if the intensity gradient $\Delta \mathcal{I}$ is high.

Through \mathbf{r}_x and \mathbf{r}_y , the solution depends on the ordering in which pixels are processed. This also affects the possibilities how object borders may be adapted. In practice, we process every scanline in both directions, such that $\mathbf{r}_x \in \{(1, 0)^T, (-1, 0)^T\}$ (to support the propagation in both directions equally).

Occlusion Detection On top we try to improve disparities near depth discontinuities. Generally, there are occluded pixels in the left image, if there is a positive disparity gradient in positive x-direction. The number of occluded pixels is given by the difference of the two disparities. We implement this efficiently in a relatively simple way using an array. At every pixel $\mathbf{p} = (x, y)$ we mark the entry at index $x - \mathcal{D}(\mathbf{p})$. But, if the entry has already been marked, the pixel is considered occluded.

5 Recommendations for an Efficient Implementation

In accordance to [17], we optimize scanlines individually to benefit from caching in CPUs.

We achieved good running times by remembering the maximally tested disparity for every pixel, in order to reduce redundant computations. In this way, we discard disparities smaller than the maximally tested disparity.

Another important optimization is the use of SIMD² instructions (in all methods: the traditional local correlation-based method and our proposals). However, to keep the code maintainable, we optimized the dissimilarity measure only (in our case: sum of absolute differences over an 8×8 window). We used so called *compiler intrinsics* to avoid cryptic assembler instructions.

6 Results

We chose to evaluate our disparity computation algorithm using well known stereo images with ground truth. To ensure a fair comparison, we decided to not improve the methods by e.g. multiple or shiftable windows and used a simple matching cost. Our goal was to evaluate the effects of a different disparity computation algorithm. We also show that stereo matching with no MD is possible without compromising quality or speed. Accordingly, we understand our algorithm not as a standalone method, but as a way to speed up existing real-time implementations. Furthermore, we applied our local energy minimization on depth-maps of our method as well as on those of the traditional method.

6.1 Stereo Images with Ground Truth

Fig. 4 presents a qualitative comparison using stereo image pairs with ground truth [19]. Obviously, some of the errors made by the traditional disparity computation method are avoided by our algorithm (for example, near the camera in the background), because our

²Single Instruction, Multiple Data

Method	Tsukuba	Venus	Teddy	Cones
Traditional correlation-based	11.17	14.16	25.96	22.50
Traditional + Refinement	5.09	4.37	19.61	18.18
Traditional + Refinement with Occlusion Detection	3.71	4.02	18.84	17.50
Our Algorithm	9.11	11.65	22.60	22.39
Our Algorithm + Refinement	4.99	4.73	18.43	17.86
Our Algorithm + Refinement with Occlusion Detection	3.93	4.54	17.00	17.15
Hirschmüller [12]	4.25	1.53	–	–
Graph-Cuts (gray scale)	4.43	4.53	25.93	17.76
Graph-Cuts (color)	4.13	2.66	17.65	15.97

Table 1: The errors of the methods (percentage of disparities that differ by more than 1; we ignore a border of 18 pixels) Correlation with our efficient disparity computation algorithm in conjunction with our refinement (local energy minimization with occlusion detection) comes close to the graph-cuts method.

Method	Tsukuba	Venus	Teddy	Cones
Traditional Correlation	32	52	98	98
Our Algorithm	31	44	47	48
Our Algorithm + Refinement with Occlusion Detection	46	74	73	85
Graph-Cuts	> 1000	> 1000	> 1000	> 1000

Table 2: Execution Times at standard data sets in milliseconds. Our algorithm is up to twice as fast as the fastest possible implementation of the traditional disparity computation.

algorithm manages to stop searching earlier than the traditional brute-force search. The local energy minimization appears to improve depth-maps independent from the actual stereo method and is therefore well suited as a robust post-processor.

In the quantitative evaluation (see Tab. 1), we used a border of 18 pixels. For Graph-Cuts and our methods we used constant parameters across all four images (for Graph-Cuts and the traditional method, the MD was chosen optimally for every image pair). We applied the Graph-Cuts method to color and gray scale images. The numbers depict, that our disparity computation algorithm is clearly better than the traditional brute-force method. Post-processing depth-maps of local correlation-based methods with our local energy minimization results in a quality close to Graph-Cuts, while being much more efficient (see Tab. 6.2).

6.2 Execution Times

The execution times, see Tab. 6.2, are an evidence for the efficiency of our new algorithm. Naturally, to some degree the computational effort of our approach depends on the image content: the computational effort spent can be directly estimated from the final depth-map. In practice, the additional overhead of our refinement routine is about 50% (compared to our method). Ideally, the time saved by efficient disparity computation may be invested in fast refinement. Interestingly, the additional overhead of enabled occlusion detection was below 1 millisecond.

All execution times were obtained using 32-bit single-threaded executables (compiled using MSVC90) on an Intel dual-core machine with 2.67 GHz.

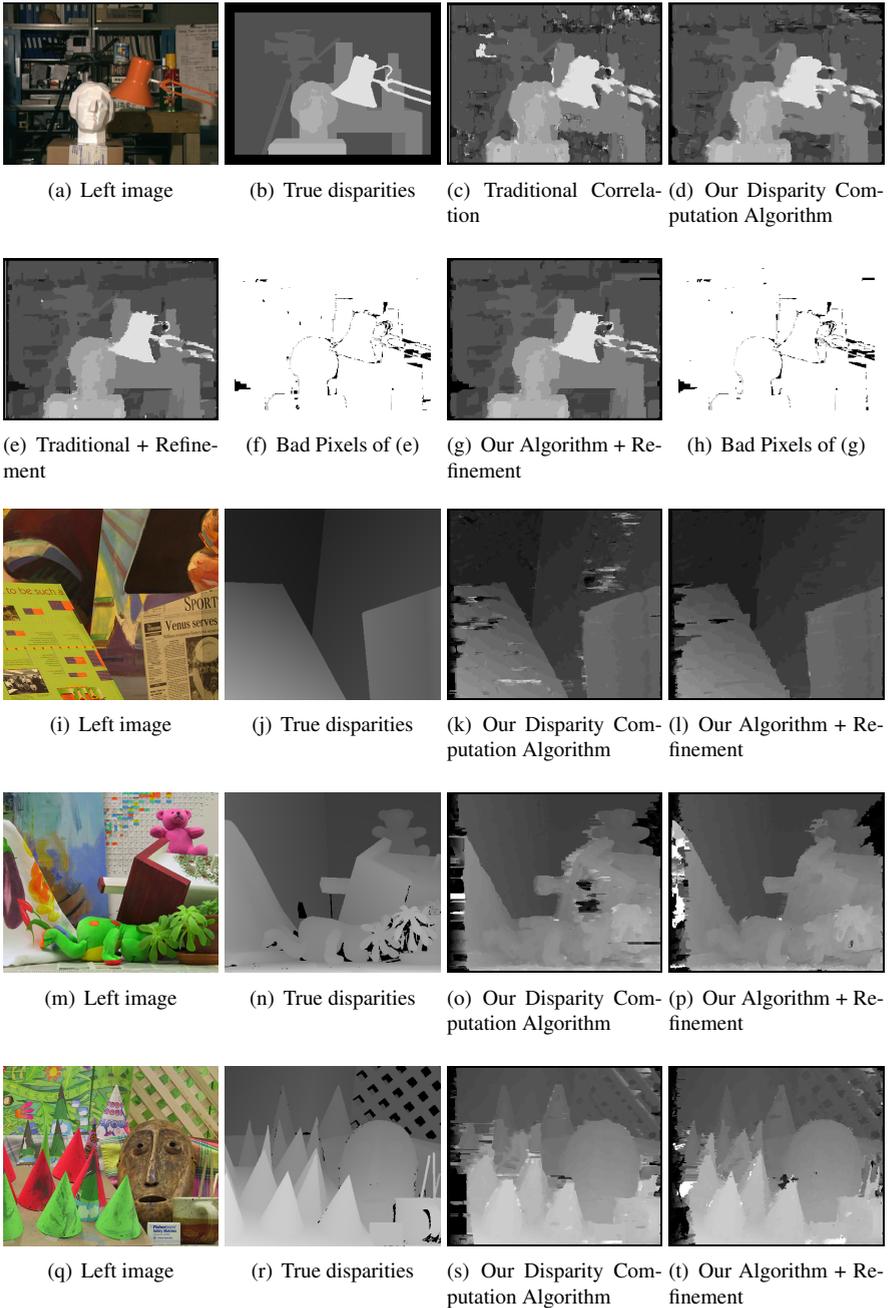


Figure 4: Comparison of the methods. Our refinement method was implemented with the described occlusion detection.

7 Conclusion

In this paper we show that the value of the maximum disparity (MD) has a great impact on the solution in local methods. In general, the MD should be *locally* as close as possible to the true disparity.

With our novel algorithm, disparities may be computed without specifying a MD. Rather than performing a brute-force search, we iteratively perform minimization- and propagation-steps. We not only circumvent the difficult task of determining the optimal search range for the disparity computation, but also present a solution that is applicable in real-time *without* special hardware. Unlike existing techniques, we do not need any information about the MD which makes our method extremely useful in motion-stereo setups.

Furthermore, we propose a local energy minimization technique that is suitable for fast refinement of the results. In the comparison, we show that the combination of our methods results in a quality close to Graph-Cuts and is even real-time.

In the future, we plan to explore further improvements of the local energy minimization method. Another interesting venue to explore is to integrate our disparity computation method into other stereo approaches.

Acknowledgements

We would like to thank Daniel Scharstein and Richard Szeliski for providing stereo images with ground truth data.

References

- [1] Stanley T. Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *ICCV*, pages 1073–1080, 1998.
- [2] Stanley T. Birchfield, Braga Natarajan, and Carlo Tomasi. Correspondence as energy-based segmentation. *Image Vision Comput.*, 25(8):1329–1340, 2007.
- [3] M. Bleyer and M. Gelautz. A layered stereo matching algorithm using image segmentation and global visibility constraints. *Photogrammetry and Remote Sensing*, 59: 128–150, 2005.
- [4] M. Bleyer and M. Gelautz. Simple but effective tree structures for dynamic programming-based stereo matching. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2008.
- [5] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *ICCV*, pages 377–384, 1999.
- [6] Alan Brunton, Chang Shu, and Gerhard Roth. Belief propagation on the gpu for stereo vision. In *Canadian Conference on Computer and Robot Vision*, pages 76–81, 2006.
- [7] L. Di Stefano, M. Marchionni, and S. Mattoccia. A fast area-based stereo matching algorithm. *Image and Vision Computing*, 22(12):983–1005, Oct 2004.

-
- [8] Olivier Faugeras, B. Hotz, Hervé Mathieu, T. Viéville, Zhengyou Zhang, Pascal Fua, Eric Théron, Laurent Moll, Gérard Berry, Jean Vuillemin, Patrice Bertin, and Catherine Proy. Real time correlation-based stereo: algorithm, implementations and applications. Technical Report RR-2013, INRIA, 1993.
- [9] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.
- [10] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, pages 807–814, 2005.
- [11] Heiko Hirschmuller. Stereo vision in structured environments by consistent semi-global matching. In *CVPR*, pages 2386–2393, 2006.
- [12] Heiko Hirschmüller, Peter R. Innocent, and Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *IJCV*, 47(1-3):229–246, 2002.
- [13] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, pages 15–18, 2006.
- [14] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *ICCV*, pages 508–515, 2001.
- [15] Ye Lu, J. Z. Zhang, Q. M. J. Wu, and Ze-Nian Li. A survey of motion-parallax-based 3-d reconstruction algorithms. *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, 34(4): 532–548, 2004.
- [16] S. Mattoccia, F. Tombari, and L. Di Stefano. Stereo vision enabling precise border localization within a scanline optimization framework. In *ACCV*, 2007.
- [17] Karsten Mùhlmann, Dennis Maier, Jürgen Hesser, and Reinhard Männer. Calculating dense disparity maps from color stereo images, an efficient implementation. *IJCV*, 47(1-3):79–88, 2002.
- [18] Ilya D. Rosenberg, Philip L. Davidson, Casey M. R. Muller, and Jefferson Y. Han. Real-time stereo vision using semi-global matching on programmable graphics hardware. In *SIGGRAPH 2006 Sketches*, 2006.
- [19] Daniel Scharstein, Richard Szeliski, and Ramin Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47:7–42, 2002.
- [20] Christoph Strecha, Rik Fransens, and Luc Van Gool. Combined depth and outlier estimation in multi-view stereo. In *CVPR*, pages 2394–2401, 2006.
- [21] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Trans. PAMI*, 25(7):787–800, 2003.
- [22] W. van der Mark and D. M. Gavrila. Real-time dense stereo for intelligent vehicles. *IEEE Trans. Intell. Transp. Syst.*, 7(1):38–50, 2006.
- [23] Olga Veksler. Fast variable window for stereo correspondence using integral images. In *CVPR*, pages 556–561, 2003.

- [24] Liang Wang, Miao Liao, Minglun Gong, Ruigang Yang, and David Nister. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In *Proc. Int. Symp. 3D Data Proc., Vis., and Transm. (3DPVT)*, pages 798–805, 2006.
- [25] Z. Wang and Z. Zheng. A region based stereo matching algorithm using cooperative optimization. In *CVPR*, pages 1–8, 2008.
- [26] Kuk-Jin Yoon and In-So Kweon. Locally adaptive support-weight approach for visual correspondence search. In *CVPR*, pages 924–931, 2005.