Efficient Stereo and Optical Flow with Robust Similarity Measures

Christian Unger[1,2], Eric Wahl[1], and Slobodan Ilic[2]

[1]BMW Group, München, Germany
[2]Technische Universität München, Germany
firstname.lastname@bmw.de, firstname.lastname@in.tum.de

Abstract. In this paper we address the problem of dense stereo matching and computation of optical flow. We propose a generalized dense correspondence computation algorithm, so that stereo matching and optical flow can be performed robustly and efficiently at the same time. We particularly target automotive applications and tested our method on real sequences from cameras mounted on vehicles.
We performed an extensive evaluation of our method using different similarity measures and focused mainly on difficult real-world sequences with abrupt exposure changes. We did also evaluations on Middlebury data sets and provide many qualitative results on real images, some of which are provided by the adverse vision conditions challenge of the conference.

1 Introduction

Dense stereo matching and the computation of optical flow in real-time are important for many computer vision tasks. In automotive applications very useful driver assistance systems can be realized based on this information – collision avoidance maneuvering, preventive pedestrian protection, longitudinal vehicle control or camera-based parking slot detection – are just some examples.

Many of those automotive applications rely on real-time stereo and optical flow. Therefore, in this paper, we focus on dense stereo, motion-stereo and optical flow that can be applied in real-time to challenging real-world sequences acquired by cameras integrated into vehicles. Since dense matching is very demanding in terms of processing power, we focus on highly efficient methods, but still try to maintain reasonable quality. In practice, stereo matching and in particular motion-stereo becomes difficult under sudden exposure or illumination changes (e.g. in garages), low-light scenarios, different weather conditions (rain, snow, etc.) or due to glare light effects. Standard block matching techniques that are fast usually use very simple similarity measures and exhibit lots of artifacts in such realistic scenarios. On the other hand robust similarity measures are better and may be used to improve results in such situations. However, the computational burden often prohibits their usage in real-time systems. A more generic and complicated problem than stereo is optical flow. It helps determining the motion of moving objects and has to face similar challenges as ordinary stereo. However, while for stereo the epipolar geometry can be used to constrain

possible matches to epipolar lines, in optical flow a large rectangular search region must be considered instead. This usually results in a high computational overhead. Therefore, the formulation of a highly efficient and robust approach based on block matching is much more difficult than for stereo.

In this paper we propose a generalized dense correspondence computation algorithm, so that stereo matching and optical flow can be performed robustly and efficiently at the same time. We generalize and extend the concepts of the efficient disparity computation approach given in [18], which was originally designed for highly efficient disparity retrieval. There, stereo matching is performed iteratively by alternating minimization and propagation phases at every pixel. We significantly increase the correspondence search range to a two dimensional area and, although based on window-based block matching, still maintain a surprisingly high efficiency. Consequently, our approach can be applied not only to stereo, but also to optical flow.

We demonstrate the effectiveness on challenging real-world and Middlebury data sets and the results underline a significant improvement in running times, while quality is not sacrificed.

In the rest of the paper, we will first review related work and explain briefly the ideas in [18], then present our method and finally show an exhaustive experimental evaluation.

2 Related Work

Traditional *correlation-based* or *local methods* [4, 8] can be implemented very efficiently [15, 19, 3] and are still widely used in many real-time applications. The main assumption of these approaches is that all pixels in the matching region originate from co-planar or even fronto-parallel scene points. This assumption is often violated in real world scenarios and results in inaccurate object boundaries [8, 17]. Some techniques have been introduced to improve the quality [21, 12, 8], but are often quite time consuming or have limited effect.

In global methods the stereo problem is formulated as an energy minimization problem and is solved using standard optimization techniques like in [3, 1, 17, 7, 9, 5, 20, 14, 13]. These approaches usually achieve much better visual and quantitative quality, but are also computationally quite expensive. Several attempts have been made to improve their running times using GPUs [1, 16]. However, such hardware is not available on many mobile platforms or vehicles.

Among the global methods, semi-global matching [7, 6] is known to be the most efficient approach and also comes with a reasonable quality. However, real-time processing requires specialized hardware [6] or enormous processing power and much memory. From this prospect, a generalization of semi-global matching from stereo towards optical flow is infeasible due to the increased amount of memory and number of cost function evaluations.

By contrast, in this paper, we demonstrate both memory and computationally efficient stereo and optical flow using robust cost measures.

3 Background

In [18] an efficient dense stereo matching algorithm was presented, which does not rely on an exhaustive search technique. Unlike related correlation based methods no a-priori maximum disparity is required. In their approach matching is done by a localized minimization technique at every pixel and uses SAD. Since the maximal disparity is not known, only neighboring disparity values are checked. The one which decreases the dissimilarity is retained. However, this procedure might be confused by local minima. Therefore a propagation step was introduced where the dissimilarity is also checked using the larger neighborhood, i.e for the larger disparity values. This allows jumping over some local minima and when done in several iterations may converge to the global minimum. It is evident that the pixel dissimilarity is evaluated in every iteration and at every pixel. However, the efficiency and quality was still higher than the one of traditional real-time correlation methods based on using integral images. This can be explained by very few cost function evaluations.

4 Method

These properties of [18] motivated us to use this method and to extend it. We first use robust similarity measures and later generalize it to the optical flow problem. Since the algorithm of [18] does not rely on box filtering or integral images [19] our goal is to preserve as much efficiency as possible, even though using complex cost functions.

4.1 Robust Similarity Measures

The convergence of [18] is mainly based on the use of matching costs which are aggregated over a support region (e.g. a square window). Therefore, we argue that the use of normalized cross-correlation (NCC) or Census Transform (CT) [22] is possible as long as matching windows are used. We chose to use the following robust cost functions.

Normalized Cross-Correlation (NCC).

$$E_{NCC} = \frac{\sum\sum(I_L(\mathbf{p}) - \bar{I}_L)(I_R(\mathbf{q}) - \bar{I}_R)}{\sigma_L \sigma_R} \tag{1}$$

with I_L and I_R being the left and right images, $\bar{I}_L$ and $\bar{I}_R$ being average pixel intensities in the correlation window computed as $\bar{I}_L = \sum\sum \frac{I_L(\mathbf{p})}{N}$ where $N = (w+1)^2$ and w is the window size. $\sigma^2_{I_L} = \sum\sum(I_L(\mathbf{p}) - \bar{I}_L)^2$ is the variance of the image intensities in the defined correlation window. I_R, $\bar{I}_R$ and σ_{I_R} are computed analogously.

Census Transform (CT). The Census filter computes a bit string for every image pixel. Every bit encodes a specific pixel of the local window centered around a pixel of interest. The bit is set to one if the pixel has a lower intensity than the pixel of interest. Later, the pixel-wise matching cost is defined as the Hamming distance of pairwise bit strings. In practice, we sum these Hamming distances over a small support region.

4.2 Optical Flow

For optical flow, we generalize the approach presented in [18], by modifying the individual processing steps. For every pixel location $\mathbf{p} = (x, y)^T$ we search for a flow vector $\mathbf{f} = (u, v)^T$, where u and v are the displacements in x- and y-direction. For the dissimilarity E of image pixels, we use matching costs based on SAD E_{SAD}, NCC E_{NCC}, or Census transform E_{CT}. The two-dimensional flow vectors are stored in a flow field $\mathcal{F}(\mathbf{p})$.

Optimization Procedure. One of the central ideas of the method is that at every pixel location, a steepest descent is performed. This means that at every pixel, the flow vector is modified using the *minimization* step. However, the minimization will stop at local, suboptimal minima. To alleviate this problem, a *propagation* is introduced, so that at every pixel, the flow vectors of adjacent pixels are evaluated.

Minimization Step. Let the current flow vector at $\mathbf{p}$ be $\mathbf{f}_0 = \mathcal{F}(\mathbf{p}) = (u_0, v_0)^T$ (which is $(0, 0)^T$ directly after initialization). The mapping for the iteration is then given as:

$$\mathbf{f}_{n+1} = (u_{n+1}, v_{n+1})^T := \operatorname{argmin}_{\mathbf{f} \in M} E(\mathbf{p}, \mathbf{f}) \tag{2}$$

with the modified vectors

$$M := \left\{ \begin{pmatrix} u_n + i \\ v_n + j \end{pmatrix} \middle| \, i, j \in \{-1, 0, 1\}, i^2 + j^2 \leq 1 \right\} \tag{3}$$

Please note that we do not include diagonal steps in M, because it improves the efficiency and the result is not notably affected. If $\mathbf{f}_{n+1} = \mathbf{f}_n$ the iteration is stopped and the flow field is updated.

Propagation Step. In the propagation at every pixel, the flow vectors from surrounding pixels are evaluated and the flow field is updated:

$$\mathcal{F}(\mathbf{p}) \mapsto \operatorname{argmin}_{\mathbf{f} \in N(\mathbf{p})} E(\mathbf{p}, \mathbf{f}) \tag{4}$$

with the neighboring flow vectors $N(\mathbf{p})$ (with $\mathcal{F}(\mathbf{p}) \in N(\mathbf{p})$). At this step, flow vectors may be spread through their local neighborhood. In practice, we alternate minimization and propagation steps for a few iterations until convergence is achieved (2-3 repetitions from experience).

Hierarchical Iteration. In the original formulation, the image pyramid was created only by scaling the horizontal dimension to reduce ambiguity in textureless regions. In case of optical flow, where we search also along the vertical axis, we scale both dimensions.

We start the matching at the lowest resolution. In every pyramid level, we perform the optimization procedure, which computes an estimated flow field. At next resolution, the optimization uses the upscaled flow field from the previous resolution as a starting point (in the beginning, all flow vectors are set to $(0,0)^T$):

$$\mathcal{F}^{(k+1)}(2x+i, 2y+j) = 2\mathcal{F}^{(k)}(x,y) \quad \text{with } i,j \in \{0,1\} \tag{5}$$

5 Results

In this section we present our experiments with dense stereo and optical flow methods applied to a number of real sequences and Middlebury data sets for quantitative results.

5.1 Robust Stereo Matching

Quantitative evaluation. In our experiments with the Middlebury data set *Art* from [10] we performed stereo matching using image pairs with different combinations of exposures or illuminations similar to [10]. The main result depicted in the graphs of Fig. 1 is that the tested cost measures are less effective for different illuminations than for exposure changes. The matching error depends on the amount of the illumination change between the image pair. On the contrary, the exposure change has less influence on the error variation. The Census Transform is very effective in this case and shows only slight variations between the combinations. It is interesting that in many cases local matching can keep up with semi-global matching and was in two cases even better. However, also SGM could be improved by Census Transform if more processing time is spent.

Qualitative evaluation. We performed tests on real world sequences provided by the 2011 DAGM Adverse Vision Conditions Challenge and imagery from our vehicle. In particular, we present results on the sequences *Exposure Changes* (see Fig. 2), *Groundplane Violation* (see Fig. 3) and a motion-stereo video from our application (see Fig. 4), because there are interesting differences noticeable. For the *Motion-Stereo* example we picked a very challenging data with incident sunlight. In practice, our sequences feature glare light effects, frequent exposure changes, specular reflections and specular highlights.

In Fig. 2 (*Exposure Changes*), Fig. 3 (*Groundplane Violation*) and Fig. 4 (*Motion-Stereo*) we show a comparison of traditional matching algorithms and our matching method with different similarity measures. As expected, Census Transform performs in overall better than the other similarity measures, and surprisingly, SAD performs also quite well in combination with a x-Sobel operator. The quality when using NCC is relatively bad, which might be explained by

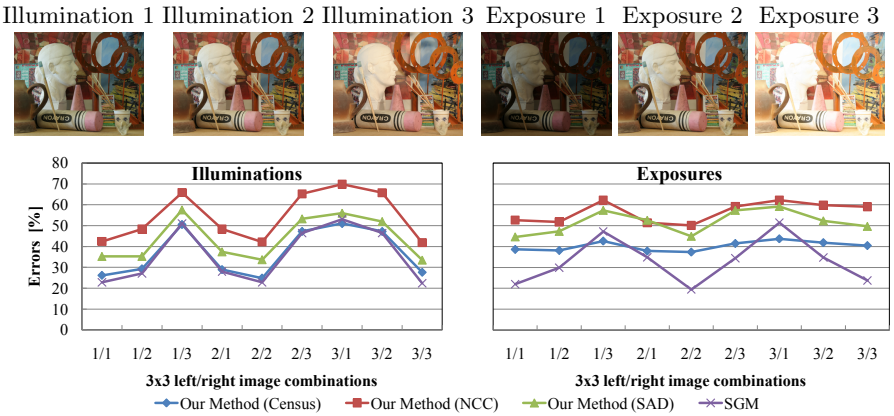

Fig. 1. Results on the stereo dataset *Art* for different exposures and illuminations. The x-axis denotes the different exposure/illumination combinations.

a high sensitivity in homogeneous regions. The semi-global method of [7] produces relatively good results, but in some difficult situations the density of the disparity maps is reduced in homogeneous regions (see Fig. 2 frame 89, Fig. 3 and Fig. 4).

5.2 Efficient Dense Optical Flow

We performed tests on the challenging real image sequences *Large Displacement* and *Exposure Changes* also provided by the 2011 DAGM AVCC and show results in Fig. 5. In *Large Displacement*, the vehicle in front (entering from the left) drives with a higher velocity than the cars in the background (which move from right to left). The sequence *Exposure Changes* is a video from a forward looking camera on a forward moving vehicle, where a sudden change in exposure takes place between frames 90 and 91. Our method with Census Transform shows again the best overall result, but also the SAD cost measure works surprisingly well on images that were previously filtered with a Sobel filter in x direction. We do not include the results of the Horn and Schunck algorithm [11], because it recovered only very localized motion. TV-L1 of [2] works relatively well, but is highly sensitive to exposure changes. Due to this reason we use Sobel-filtered images, but in this case the smoothness is negatively affected by image noise. At dramatic changes of the exposure time, none of the methods succeeded.

5.3 Efficiency

Table 1 shows the execution times in milliseconds of our single-threaded implementations on a standard Intel E8200 CPU and underlines the high efficiency of our methods. We tested stereo matching on images with a resolution of 1024x334 and a maximum disparity of 48 (for traditional methods that require an a-priori

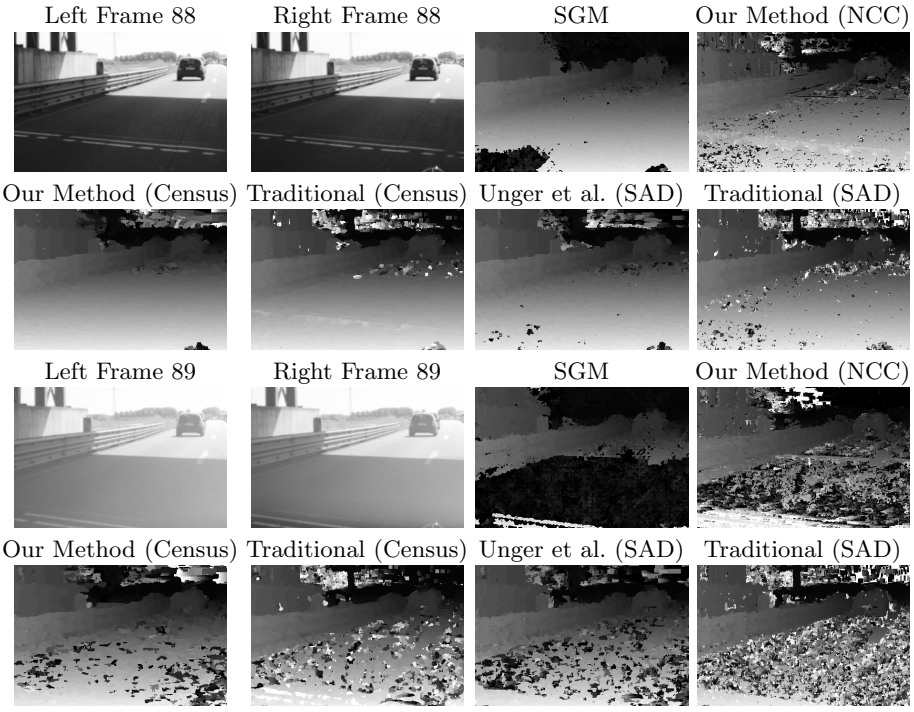

Fig. 2. Results on the sequence *Exposure Changes*.

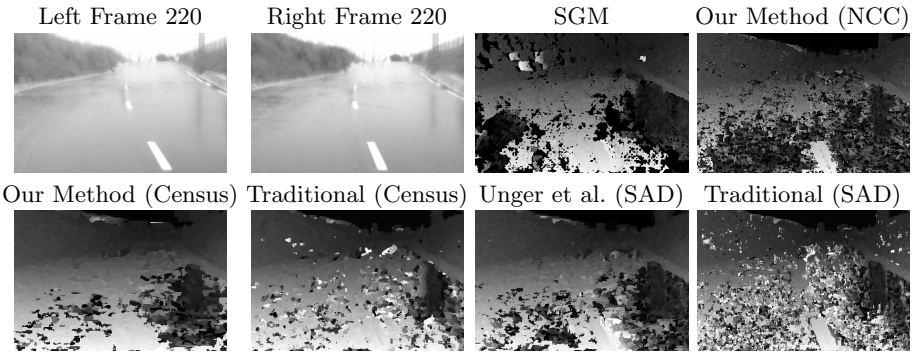

Fig. 3. Results on the sequence *Groundplane Violation*.

specification). Optical flow methods were also tested on 640x481 images and flow vectors with displacements of at most 30 pixels in each direction (the timings of [2] are not comparable, because their Matlab implementation took minutes and a GPU version reported as real-time [2] is of course not comparable to a CPU implementation). We also performed tests with down-scaled images (third of their

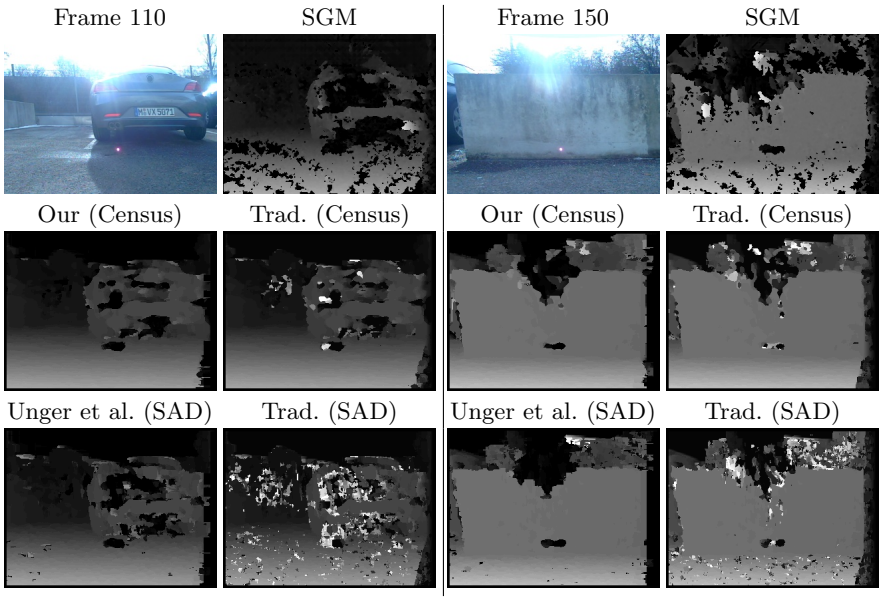

Fig. 4. Results on the motion-stereo sequences of our method (Our), traditional block matching (Trad.), semi-global matching (SGM) and the method of Unger et al.

original size). The timings underline the high efficiency of our proposed optical flow formulation which demonstrates that with small images it is possible to compute dense optical flow with large displacements in real-time on commodity hardware, even with robust cost measures. On higher resolutions, the performance gap to traditional block matching is extremely big: our proposal is up to 90 times faster due to our efficient search algorithm and the hierarchical setup.

6 Conclusion

We presented a generalized framework for dense stereo matching and optical flow which can be computed efficiently and robustly at the same time. We tested our method on a number of real world sequences and provided quantitative results on Middlebury data sets. The comparison of different similarity measures showed that robust ones, like census transform, are usually the best choice. However, in some situations traditional measures like SAD perform also quite well. The main strengths of our framework are its efficiency, which is maintained even when using more demanding robust measures, and its genericity, since complex problems like optical flow can be addressed, and its good results, obtained on very challenging real world data.

Acknowledgements: We would like to thank Daniel Scharstein and Richard Szeliski for providing stereo images with ground truth data. This work is supported by the BMW Group.

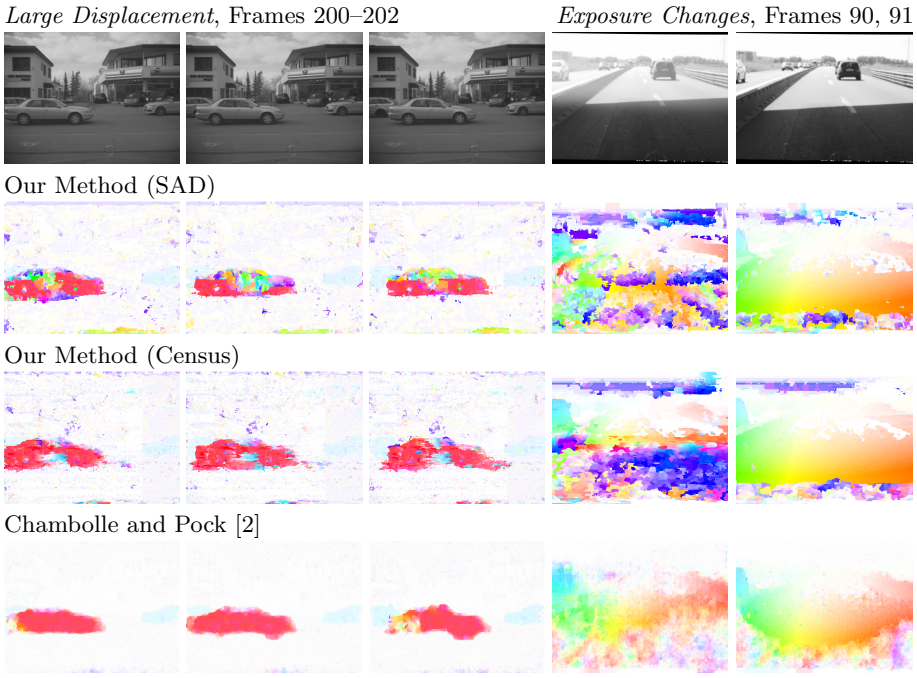

Fig. 5. Results on the sequences *Large Displacement* (flow fields were computed for frame pairs (200, 201), (201, 202) and (202, 203)) and *Exposure Changes* (frame pairs (90, 91) and (91, 92)). The image on the right denotes the color-coding of the computed flow vectors. **Please note that this figure is best viewed in color.**

References

1. Brunton, A., Shu, C., Roth, G.: Belief propagation on the gpu for stereo vision. In: Canadian Conference on Computer and Robot Vision. pp. 76–81 (2006)
2. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Tech. rep., Graz University of Technology (2010)
3. Di Stefano, L., Marchionni, M., Mattoccia, S.: A fast area-based stereo matching algorithm. Image and Vision Computing 22(12), 983–1005 (Oct 2004)
4. Faugeras, O., Hotz, B., Mathieu, H., Viéville, T., Zhang, Z., Fua, P., Théron, E., Moll, L., Berry, G., Vuillemin, J., Bertin, P., Proy, C.: Real time correlation-based stereo: algorithm, implementations and applications. Tech. Rep. RR-2013, INRIA (1993)
5. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. Int. J. Comput. Vis. 70(1), 41–54 (2006)
6. Gehrig, S.K., Eberli, F., Meyer, T.: A real-time low-power stereo vision engine using semi-global matching. In: ICVS. pp. 134–143 (2009)
7. Hirschmuller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: CVPR. pp. 807–814 (2005)
8. Hirschmüller, H., Innocent, P.R., Garibaldi, J.: Real-time correlation-based stereo vision with reduced border errors. IJCV 47(1-3), 229–246 (2002)

Table 1. The execution times of the different methods in milliseconds. Stereo matching was tested on images with a resolution of 1024x334 and a maximum disparity of 48 was used for methods that require it a-priori. Optical flow was tested on 640x481 images and flow vectors with displacements of at most 30 pixels in each direction. We also performed tests on images scaled to one third of their original size

Method	**Stereo** 1024x334	**Flow** 640x481	**Flow** 213x160
SGM	536	-	-
Unger et al. (SAD)	129	-	-
Our Method (SAD)	-	466	52
Our Method (Census)	403	1763	167
Our Method (NCC)	560	1812	179
Trad. Block Matching (SAD)	263	39744	641
Trad. Block Matching (Census)	2313	162769	2029
Trad. Block Matching (NCC)	2691	140648	1913
Horn and Schunk [11]	-	313	34
Chambolle and Pock [2]	-	N/A	N/A

9. Hirschmüller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: CVPR. pp. 1–8 (2007)
10. Hirschmüller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. TPAMI 31(9), 1582–1599 (2009)
11. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artif. Intell. 17(1-3), 185–203 (1981)
12. Hosni, A., Bleyer, M., Gelautz, M., Rhemann, C.: Local stereo matching using geodesic support weights. In: ICIP (2009)
13. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: ICPR. pp. 15–18 (2006)
14. van der Mark, W., Gavrila, D.M.: Real-time dense stereo for intelligent vehicles. IEEE Trans. Intell. Transport. Syst. 7(1), 38–50 (2006)
15. Mühlmann, K., Maier, D., Hesser, J., Männer, R.: Calculating dense disparity maps from color stereo images, an efficient implementation. IJCV 47(1-3), 79–88 (2002)
16. Rosenberg, I.D., Davidson, P.L., Muller, C.M.R., Han, J.Y.: Real-time stereo vision using semi-global matching on programmable graphics hardware. In: SIGGRAPH 2006 Sketches (2006)
17. Scharstein, D., Szeliski, R., Zabih, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vis. 47, 7–42 (2002)
18. Unger, C., Benhimane, S., Wahl, E., Navab, N.: Efficient disparity computation without maximum disparity for real-time stereo vision. In: BMVC (2009)
19. Veksler, O.: Fast variable window for stereo correspondence using integral images. In: CVPR. pp. 556–561 (2003)
20. Wang, L., Liao, M., Gong, M., Yang, R., Nister, D.: High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In: Proc. Int. Symp. 3D Data Proc., Vis., and Transm. (3DPVT). pp. 798–805 (2006)
21. Yoon, K.J., Kweon, I.S.: Locally adaptive support-weight approach for visual correspondence search. In: CVPR. pp. 924–931 (2005)
22. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: ECCV (2). pp. 151–158 (1994)