# Graph-Cut versus Belief-Propagation Stereo on Real-World Images

Sandino Morales[1], Joachim Penc[2], Tobi Vaudrey[1], and Reinhard Klette[1]

[1] The *.enpeda..* Project, The University of Auckland, New Zealand
[2] Informatics Institute, Goethe University, Frankfurt, Germany

**Abstract.** This paper deals with a comparison between the performance of graph cuts and belief propagation stereo matching algorithms over long real-world and synthetics sequences. The results following different preprocessing steps as well as the running times are investigated. The usage of long stereo sequences allows us to better understand the behavior of the algorithms and the preprocessing methods, as well as to have a more realistic evaluation of the algorithms in the context of a vision-based Driver Assistance System (DAS).

## 1 Introduction

Stereo algorithms aim to reconstruct 3D information out of (at least) a pair of 2D images. To achieve this, corresponding pixels in the different views have to be matched to estimate the disparity between them. There exist many approaches to solve this matching problem, most of them too slow and/or inaccurate. In this paper we compare a graph cut method – which produces in [1,8,13] very good results but is quite slow – and belief propagation stereo which has proven in [5,13] to produce good results in reasonable running time. Both algorithms apply *global* 2D optimization by using information from potentially unbounded 2D neighborhoods for pixel matching, as opposed to, for example, *local* techniques (e.g., correlation-based), or *semi-global* scan-line optimization techniques (e.g., dynamic programming, semi-global matching). Furthermore, we are interested in analyzing various preprocessing methods (as suggested in [5,17]) in order to minimize common issues of real-world imaginary. We are in particular interested in eliminating a negative influence of brightness artifacts, which cause major issues for matching algorithms. This effect on stereo reconstruction quality is often neglected when looking at indoor scenes, with good lighting and cameras. As stated in [9], this kind of noise has a significant influence on the output of stereo algorithms. Following [5], we use the simple Sobel edge detector in order to improve the outcome of the algorithms. We also use *residual images* (i.e., images resulting from subtracting a smoothed version from an original image) that have proved to be of use for overcoming brightness issues, see [17]. The processing time of the algorithms it is also investigated, as this is of importance for most applications, such as vision-based driver assistance systems (DAS) and mobile robotics.

Performances of these two algorithms (BP and GC) have been compared in the past, but only for engineered or synthetic images; see, for example, [15]. Our study is focused on a comparison of the performance of both algorithms on long real-world

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 732–740, 2009.

image sequences; but we also investigate the performance of the algorithms over a long synthetic sequence (for behavior with respect to some systematic changes in this synthetic sequence, but not for ranking of methods; indoor or synthetic data do have limited relevance for the actual ranking of methods for real-world DAS). Those long test sequences allow us to better understand the behavior of the algorithms in general, and in particular the effects of previously proposed preprocessing methods. To overcome the lack of ground truth we use a sequence recorded with three calibrated cameras; thus we are able to use *prediction error analysis* as a quality measure [14]; we use the same approach to evaluate the performance of the algorithms on the chosen long synthetic sequence.

This paper is structured as follows: Section 2 specifies the implementations used in this paper of the graph cut and belief propagation algorithms; it also recalls prediction error analysis and informs about the chosen preprocessing methods. Section 3 presents and discusses the obtained results. Conclusions are stated in Section 4.

## 2 Approach for Evaluation

The experiments have been performed using a very recent graph cut implementation from V. Kolmogorov and R. Zabih[1] which can detect occlusions quite well; and a modified coarse-to-fine belief propagation algorithm of Felzenszwalb and Huttenlocher[2] as implemented for [5], focusing on more reliable (and time-efficient) matching, therefore using max-product, 4-adjacency, truncated quadratic cost function, red-black speed-up, and coarse-to-fine processing. Both algorithms were implemented under a C++ platform. For a detail discussion on belief propagation and graph cut algorithms see [7] and [3], respectively.

The outline of our experiments is as follows. We evaluate both algorithms over a synthetic and a real-world sequence, and compare results and computational time. Furthermore, we use two different preprocessing methods in order to improve the results. For the graph cut algorithm we also analyze the effect of different number of iterations (between 1 and 3). The algorithms were tested on an Intel Core2 vPro at 3.0 GHz with 4 GB memory using Windows Vista as the operating system.

**Data Set.** The *POV-ray* synthetic sequence (100 stereo pairs) with available ground truth is from Set 2 on [2], as introduced in [16]. The real-world sequence of 123 frames (recorded with three calibrated cameras using the research vehicle of the *.enpeda..* project, which is the *ego-vehicle* in our experiments) is from Set 5 on [2], as introduced in [10], and it is a fairly representative example of a daylight (no rain) outdoor sequence, containing reflections and large differences in brightness between subsequent stereo pairs, or between the left and right image of the stereo pair. The use of long sequences facilitates the recognition of circumstances that may affect the performance of an algorithm, as well as it helps to understand the *robustness* of an algorithms with

[1] See `http://www.adastral.ucl.ac.uk/vladkolm/software/match-v3.3.src.tar.gz`

[2] See `http://people.cs.uchicago.edu/~pff/bp` for original sources.

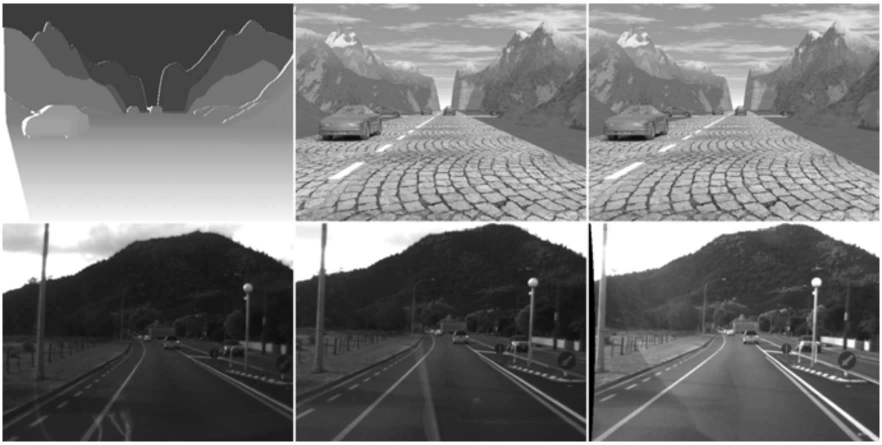

**Fig. 1.** Data sets. Synthetic sequence (upper row), form left to right: Ground truth disparity (dark = far, light = close, white = occlusion), left and right views of frame 43. Real-world sequence (lower row): from left to right, view of the left, center, and right cameras of frame 37.

respect to changes in circumstances (e.g., brightness differences, lighting artifacts, close objects, night, or rain). See Figure 1 for examples of the used data sets.

**Preprocessing Methods.** A vision-based DAS has to deal with input data that are recorded under uncontrolled environments. Among all the adverse conditions faced by outdoor image grabbing, brightness differences between the images of a stereo pair have a particularly negative influence on the output of the stereo algorithms [9]. In order to over come this almost unavoidable issue, following [5], we preprocess our sequences using a 3×3 Sobel edge operator (with a processing time of 0·06 s per stereo pair) to create an *edge sequence*. In [12], the simple Sobel operator proved to be the most effective edge-operator within a group of edge operators, tested for improving correspondence analysis on real-world data.

In [17], the authors used *residual images* to remove the illumination differences between correspondence images. We analyze whether there is an improvement in the output of the belief propagation and graph cut stereo algorithms using residual images as source data. Given an image $I$, we consider it as a composition $I(p) = s(p) + r(p)$, for pixel position $p \in \Omega$ (the image domain), where $s = S(I)$ denotes the *smooth component* and $r = I - s$ the *residual* one. We use the straightforward iteration scheme to obtain the residual component of the image $I$:

$$\mathbf{s}^{(0)} = I, \quad \mathbf{s}^{(n+1)} = S(\mathbf{s}^{(n)}), \quad \mathbf{r}^{(n+1)} = I - \mathbf{s}^{(n+1)}, \quad \text{for } n \geq 0.$$

In our experiments we use a 3×3 mean filter to generate the smooth component and $n = 40$ iterations (with a computational time of 0·07 s per stereo pair; see [17] for a reasoning for selecting mean filtering and $n = 40$). We refer to the sequence formed by residual images as *residual sequence*. See Figure 2 for a sample stereo pair of the real-world residual sequence and the edge synthetic sequence.

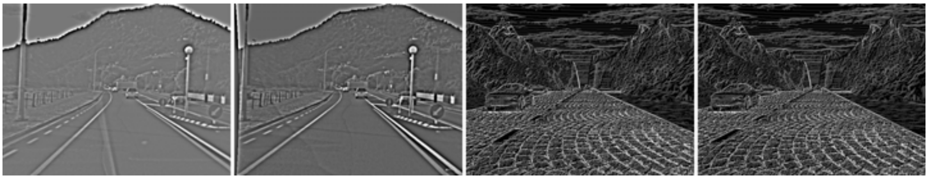

**Fig. 2.** Examples of the output of the preprocessing methods. Left: Residual stereo pair frame 37 of the real-world sequence. Right: Sobel edge stereo pair frame 43 of the synthetic sequence.

**Evaluation approach.** To objectively evaluate the performance of the algorithms over the real-world sequence (with non-available ground truth), the output of the algorithms is analyzed using the so-called *prediction error* [14]. This technique requires at least three images of the same scene: two of them are used to calculate a disparity map, while the third one is used for evaluation purposes. For consistency, the evaluation of the synthetic sequence is also performed with the prediction error. The third or virtual image used to evaluate the results is generated using the same pose of the left-most camera of the three-camera set-up in our research vehicle (while recording the real-world sequences) and the available ground truth.

We follow the method described in [10], where the (rectified) images recorded by the center and right-most camera are used as the input data of the stereo algorithms. The resultant disparity map and the center image are used to generate (by geometrical means) a virtual image as it would be recorded by the left-most camera. This virtual image is then compared with the actual left-most image in the following way: for each frame $t$ of the given trinocular sequence, let $\Omega_t$ be the set of all pixels in the left image $I_l$, such that their source scene point is also visible in the center and right images. Let $(x, y)$ be the coordinates of a pixel in $\Omega_t$ with intensity $I_l(x, y)$. The method above assigns to the pixel with coordinates $(x, y)$ in the virtual image $I_v$ an intensity value $I_v(x, y)$ (defined by the intensity of a certain pixel in the center image). Thus, we are able to compute the *root mean squared* (RMS) error between the virtual and the left image as follows:

$$R(t) = \frac{1}{|\Omega_t|}\Big(\sum_{(x,y)\in\Omega_t} [I_l(x,y) - I_v(x,y)]^2\Big)^{1/2}$$

where $|\Omega_t|$ denotes the cardinality of $\Omega_t$. A *high* RMS means there is more error.

The *normalized cross correlation* (NCC) is also used to compare left and virtual image, applying the following:

$$N(t) = \frac{1}{|\Omega_t|} \sum_{(x,y)\in\Omega_t} \frac{[I_l(x,y) - \mu_r][I_v(x,y) - \mu_v]}{\sigma_r \sigma_r}$$

$\mu_l$ and $\mu_v$ denote the means, and $\sigma_l$ and $\sigma_v$ the standard deviations of $I_l$ and $I_v$, respectively. A *low* NCC means there is more error.

Since we are dealing with a image sequence, the results can be graphed over time (see Figure 5 for an example). To summaries the large dataset, we compute the mean and the zero mean standard deviation (ZMSD - the standard deviation assuming a mean of zero) of the results in a sequence.

## 3 Results and Discussion

**Synthetic Sequence.** According to [1], GC only needs a few iterations to obtain acceptable results. Thus we test the algorithm with only one or three iterations over the three sequences. For all the sequences, differences in results for either one or three iterations are almost imperceivable, visually and statistically. The RMS metric reports a slight improvement using three iterations, and the NCC shows that the results are a bit better using just one (see Table 1). The computational time, on average, was 135·3 s and 386·7 s for one and three iterations, respectively. The latter result discourages the use of more than one iteration for this synthetic sequence.

Differences in GC results between the preprocessed sequences and the original ones are not consistent either. On one hand, NCC reports that the best performance is with the original sequence. On the other hand, the best RMS results are obtained with the edge sequence. Visually, NCC seems to report more accurately the behavior of the algorithm, as the results seems to suffer degradation with the preprocessed sequences (see Figure 3).

BP shows a different behavior. With RMS, the best overall results were obtained with the original sequence, while with NCC this sequence showed the worst performance. Again, by visual inspection, NCC seems to reflect better the performance of the algorithms, as the results get better (visually) using any of the discussed preprocessing methods. The average computational time was 98·5 s (parameters used: ITER $= 7$, LEVELS $= 6$, DISC$_K = 50$, DATA$_K = 35$, $\lambda = 0{\cdot}07$).

Summarizing (see BP values in Table 1), the metrics show contradictory results when using preprocessing methods. Visually, NCC seems to be the more appropriate metric; following the NCC results, GC has a better performance than BP on the original synthetic sequence, with one or three iterations; but, with preprocessing, BP produces results are as good as GC. See Figure 3.

**Real-world Sequence.** With GC, all the sequences reported worse RMS results with three iterations compared to one! NCC reports basically no change except for the edge sequence, for which the results are slightly worse for three iterations (see Table 2).

**Table 1.** Summarizing NCC and RMS results for the synthetic sequence

| Evaluation | | GC - 1 Iteration | | GC - 3 Iterations | | BP | |
|---|---|---|---|---|---|---|---|
| Approach | Sequence | Mean | ZMSD | Mean | ZMSD | Mean | ZMSD |
| NCC | Original | 0·76 | 0·76 | 0·75 | 0·75 | 0·68 | 0·68 |
| | Residual | 0·75 | 0·75 | 0·74 | 0·74 | 0·76 | 0·76 |
| | Sobel | 0·73 | 0·73 | 0·72 | 0·72 | 0·73 | 0·73 |
| RMS | Sobel | 36·04 | 36·07 | 36·02 | 36·04 | 36·02 | 36·04 |
| | Residual | 36·68 | 36·70 | 36·65 | 36·67 | 36·51 | 36·54 |
| | Original | 36·82 | 36·84 | 36·66 | 36·68 | 35·86 | 35·88 |

**Table 2.** Summarizing NCC and RMS results for the real world sequence

| Evaluation | | GC - 1 Iteration | | GC - 3 Iterations | | BP | |
|---|---|---|---|---|---|---|---|
| Approach | Sequence | Mean | ZMSD | Mean | ZMSD | Mean | ZMSD |
| NCC | Residual | 0·66 | 0·67 | 0·66 | 0·67 | 0·68 | 0·69 |
| | Sobel | 0·66 | 0·67 | 0·65 | 0·65 | 0·65 | 0·66 |
| | Original | 0·64 | 0·65 | 0·64 | 0·65 | 0·65 | 0·66 |
| RMS | Residual | 33·48 | 34·06 | 33·50 | 34·08 | 32·91 | 33·48 |
| | Sobel | 34·03 | 34·62 | 34·19 | 34·78 | 33·00 | 33·58 |
| | Original | 35·34 | 35·94 | 35·49 | 36·10 | 34·04 | 34·57 |

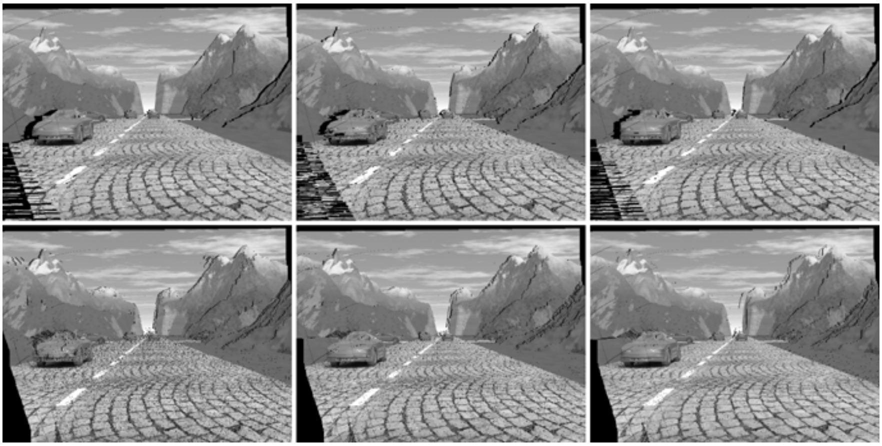

**Fig. 3.** Examples of the generated virtual view. Left to right: Original, edge map and residual sequences. Upper row: GC with one iteration. Lower row: BP.

In contrast, the preprocessing methods have a positive influence on the outcome of GC, the residual sequence having the best performance (no matter whether one or three iterations, and for both metrics). The mean computational time was 178·64 s and 390·72s for one and three iterations, respectively, per stereo pair. This result discourages the use of more than one iteration.

BP shows here a similar behavior as GC: the results improved with both preprocessing methods (with respect to both metrics), the residual sequence having the best results. The average NCC does not report any improvement with the edge sequence (when comparing with the original sequence); however, visually, the improvement is obvious when the difference in brightness is large between both images in an input stereo. The parameters used with the real world sequence were as follows: LEVELS $=$ 6, DISC$_K$ $=$ 500, DATA$_K$ $=$ 100, $\lambda = 0.3$, with an average computation time of 122·44 s per stereo pair of images.

Table 2 illustrates that BP outperforms GC in the overall results (as well as in computational time) even when comparing the best GC result (one iteration over the residual sequence) and the worst of BP (original sequence). Both algorithms

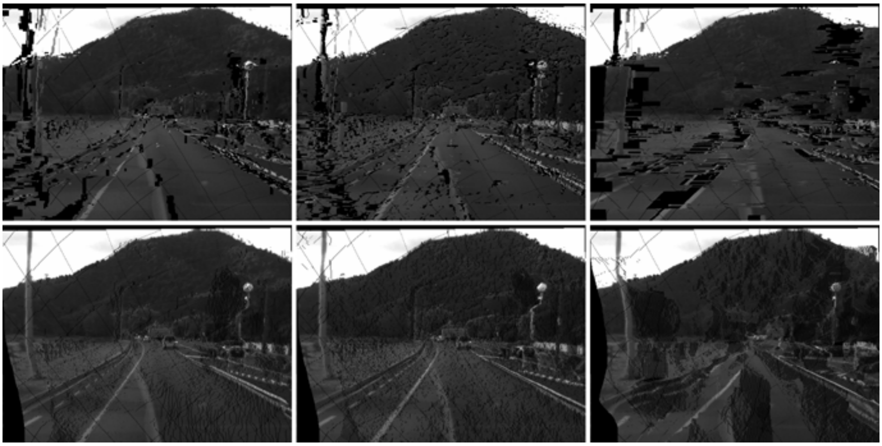

**Fig. 4.** Examples of the generated virtual view. Left to right: Residual, edge map and original sequences. Upper row: GC with 1 iteration. Lower row: BP.

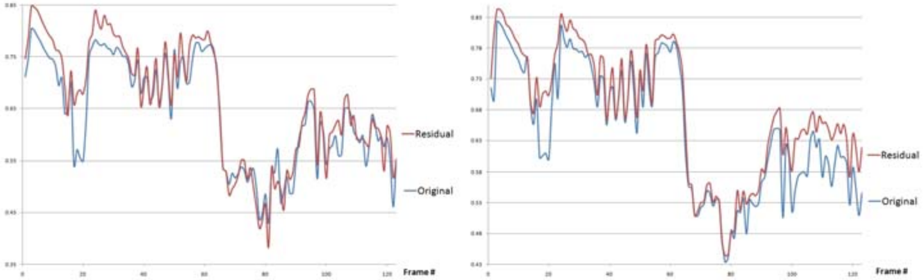

**Fig. 5.** NCC evaluation for the real-world sequence. Comparison shown between the original and the residual sequence. Left: GC with one iteration. Right: BP.

improve their results using preprocessed sequences, particularly, when the differences in brightness between both images in a stereo input pair are large. Figure 4 shows the calculated virtual views for frame #47 for the original (right) and the preprocessed sequences (left and center); the improvement can be detected visually. It is also interesting to note that there is a minimal (or no) improvement with the preprocessing of the original sequence when differences in brightness are only minor, meaning, that with fairly good balanced images, there would be no need of preprocessing.

**Summary.** The difference between the computational time between one and three iterations of GC, and the almost null benefit (or even a degradation in the obtained) results, discourages the use of more than one iteration, for both the real-world and the synthetic sequences (and its respective modifications). The preprocessing methods reported better results for both algorithms (except for GC when the synthetic sequence was evaluated), the residual image method having the best performance, see Figure 5. From Table 1, GC outperforms BP over the original synthetic sequence. However, BP had a better performance over the original real-world sequence, showing that it is misleading to evaluate over synthetic sequences when ranking stereo algorithms. This also tells us that more research needs to be done for studying the performance of stereo algorithms different circumstances (night, rain, etc.). For example, in a more recent comparison, GC has shown a better performance on sequences captured in the night, or when objects appear close to the ego-vehicle.

Note that the metrics reported different rankings when evaluating the synthetic sequence, NCC being the one that seems to confirm what can be concluded by visual inspection. This behavior (inconsistency in metrics) was not expected in images that have been recorded under perfect conditions. However, RMS is certainly a ‘very accurate’ measure, ‘asking for to much’, and seems to be misleading in evaluations.

## 4 Conclusions

In this paper we compare the performance of a belief propagation stereo algorithm with a graph cut stereo implementation, using two long sequences (real-world and synthetic) and two different preprocessing methods. We also tested the influence of the number of iterations for the GC algorithm. The different rankings obtained by the algorithms on either the real-world or the synthetic sequence support the usage of a wide class of data sets for testing the performance of the algorithms, to avoid some bias. The preprocessing methods proved to be a good option when dealing with real world images, as the results improved for both algorithms. For the synthetic sequence the metrics do not show consistent results, and when some improvement was detected, then it was only fairly minor. This is as expected, as there is no need to improve ‘perfect’ (good contrast and grey value distribution) images. We also noticed that there is no need to use more than one iteration with GC; if there was an improvement, it was almost imperceivable, statistically and visually. On the other hand, the difference in computational time is considerably large.

Future work may include the investigation of more metrics and preprocessing methods, as well as the usage of data sets with other adverse conditions such as rain, night time, and so forth.

## References

1. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Analysis Machine Intelligence 23, 1222–1239 (2001)
2. .enpeda.. image sequence analysis test site (EISATS), `http://www.mi.auckland.ac.nz/EISATS/`
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. Int. J. Computer Vision 70, 41–54 (2006)
4. Guan, S., Klette, R.: Belief-propagation on edge images for stereo analysis of image sequences. In: Sommer, G., Klette, R. (eds.) RobVis 2008. LNCS, vol. 4931, pp. 291–302. Springer, Heidelberg (2008)
5. Guan, S., Klette, R., Woo, Y.W.: Belief propagation for stereo analysis of night-vision sequences. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 932–943. Springer, Heidelberg (2009)

6. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 82–96. Springer, Heidelberg (2002)
7. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Analysis Machine Intelligence 26, 65–81 (2004)
8. Kolmogorov, V., Zabih, R.: Graph cut algorithms for binocular stereo with occlusions. In: Paragios, N., Chen, Y., Faugeras, O. (eds.) Handbook of Mathematical Models in Computer Vision, pp. 423–438 (2006)
9. Morales, S., Vaudrey, T., Klette, R.: An in depth robustness evaluation of stereo algorithms on long stereo sequences. In: Proc. IEEE Intelligent Vehicles Symp., pp. 347–352 (2009)
10. Morales, S., Klette, R.: A Third Eye for Performance Evaluation is Stereo Sequence Analysis. In:Proc. CAIP (to appear, 2009)
11. Ohta, Y., Kanade, T.: Stereo by intra- and inter-scanline search using dynamic programming. IEEE Trans. Pattern Analysis Machine Intelligence 7, 139–154 (1985)
12. Al-Sarraf, A., Vaudrey, T., Klette, R., Woo, Y.W.: An approach for evaluating robustness of edge operators on real-world driving scenes. In: IEEE Conf. Proc. IVCNZ 2008, Digital Object Identifier 10.1109/IVCNZ.2008.4762096 (2008)
13. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Computer Vision 47, 7–42 (2002)
14. Szeliski, R.: Prediction error as a quality metric for motion and stereo. In: Proc. Int. Conf. Computer Vision, vol. 2, pp. 781–788 (1999)
15. Tappen, M., Freeman, W.: Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In: Proc.9th IEEE ICCV, vol. 2, pp. 900–906 (2003)
16. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behavior on synthetic and real-world stereo sequences. In: Proc. Int. Conf. Image Vision Computing, New Zealand. IEEE Xplore, Los Alamitos (2008)
17. Vaudrey, T., Klette, R.: Residual images remove illumination artifacts for correspondence algorithms!. In: Proc. Pattern Recognition - DAGM (to appear, 2009)