A Study on Stereo and Motion Data Accuracy for a Moving Platform

Sandino Morales[1], Young Woon Woo[2], Reinhard Klette[1] and Tobi Vaudrey[1]

[1] University of Auckland, Auckland, New Zealand
pmor085@aucklanduni.ac.nz
Home page: http://www.mi.auckland.ac.nz/
[2] Dong-Eui University, Department of Multimedia Engineering, Busan, Korea

Abstract. Stereo and motion analysis are potential techniques for providing information for control or assistance systems in various robotics or driver assistance applications. This paper evaluates the performance of several stereo and motion algorithms over a long synthetic sequence (100 stereo pairs). Such an evaluation of low-level computer vision algorithms is necessary, as moving platforms are being used for image analysis in a wide area of applications. In this paper algorithms are evaluated with respect to robustness by modifying the test sequence with various types of realistic noise. The novelty of this paper is comparing top performing algorithms on a long sequence of images, taken from a moving platform.

1 Introduction

The main task of computer vision is to use image data recorded by one or multiple cameras to understand the given 3D environment. In particular, stereo algorithms obtain 3D information about the scene geometry, and motion algorithms gather information about the 2D motion of the images. Both types of information are needed to reconstruct the 3D motion of the scene. These algorithms still represent a challenging task for the vision community. For mobile devices the challenge becomes even more difficult; moving background, change in lighting conditions, possible misalignments of cameras, and so forth, make the task of the algorithms even harder. However, the use of cameras has become a popular data sensor for moving vehicles. Vision-based stereo has already been used in a wide variety of vehicles, including wheelchairs (e.g., for the detection of obstacles, unevenness of the ground, detection of stairs and ropes or beams in the air [13]), or for forklifts, where the operator gets valuable information to deal with heavy loads at great heights [14], and in standard cars to assist a driver while driving on a road.

Thus, it is necessary to evaluate the performance of these algorithms to detect which one performs the best in different situations and to encourage their theoretical improvement. Several authors have evaluated stereo algorithms; for example [1] presented one of the earliest evaluations of stereo algorithms and [3] presented one of the most recent and representative evaluation papers so far. Several stereo algorithms were tested, but in both cases the experiments were done with small sets of images. Motion algorithms have also been evaluated, the approach presented in [17] influenced the evaluation of motion algorithms until recently. Now, [16] is the main approach for testing and comparing algorithms online. However, both publications focus on very short image sequences.

In this paper we evaluate the performance of stereo and motion algorithms over a long (i.e., 100 stereo pair) sequence. The analysis of long sequences allows the usage of temporal information (e.g., [5]). In order to test the robustness of the chosen algorithms, we added different kinds of noise to the sequence so that algorithms can be tested under different conditions [12]. As we are interested in testing the algorithms on a mobile platform (a wheelchair or car), we used a long sequence that is publicly available, with ground truth for motion and stereo, in Set 2 on the *.enpeda..* Image Sequence Analysis Test Site [4]. This sequence simulates a driving situation.

This paper follows [12] with respect to stereo algorithm evaluation, and extends these studies by including evaluations of optic flow algorithms on the same synthetic sequence.

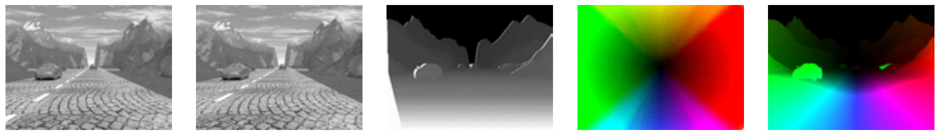
2 Stereo and Optic Flow Algorithms

Stereo Vision: is the process of understanding the 3D information of the environment from the available 2D data (images) by matching the corresponding projections of a 3D point in (at least) two images. The algorithms chosen for our analysis are as follows:
(i) *Dynamic programming stereo*: we compare a standard algorithm [6] (DP), against one with temporal (DPt), spatial (DPs), or temporal and spatial (DPts) propagation; see [7] for propagation details.
(ii) *Belief propagation stereo* (BP): we use a coarse-to-fine algorithm of [8] with quadratic cost function, as reported in [9];
(iii) *Semi-global matching* (SGM): characterizes one of the top performing stereo strategies, see [2]; we chose two cost functions, mutual information (SGM MI) or Birchfield-Tomasi (SGM BT) [10];

Motion Analysis: is estimated from a pair of images taken at slightly different times. Optical flow algorithms aim to detect the visible displacement of pixels in the image plane to understand the motion of the 2D projection of 3D motion of the visible objects (and background). The following are the algorithms used in our evaluation:
(i) *Horn-Schunck algorithm* (HS): we use the program as available in the OpenCV library [11];
(ii) *Combination of Local and Global* (CLG): optimization as in [18]; we use an implementation from the *.enpeda..* group; see acknowledgment;
(iii) *BBPW*: this is named after the initials of surnames of all the four co-authors of [15], we use an implementation also from the *.enpeda..* group; see acknowledgment.

Evaluation Approach: the algorithms were tested using the original sequence and with the same sequence corrupted with different types and magnitudes of noise. For the stereo algorithms we analyzed the stereo pair at each time frame, and for the motion algorithms only the left images.

Data Set and Visualization: For our experiments we use a long sequence of 100 synthetic stereo image pairs and ground truth data (for stereo and motion



(a) Left image (b) Right image (c) GT Disparity (d) Color Key (e) GT Flow

Fig. 1. Stereo image pair #40 of the sequence; (a) and (b) are original left and right images. (c) ground truth data in gray-scale encoding: light = close, dark = far, white = occlusion. (d) color key for encoding optic flow. (e) ground truth optic flow.

algorithms), which are all available on [4]. To visualize the stereo results we use gray scale encoding: light for closer objects and dark for objects further away. The color key that we use to visualize the motion results uses hue for direction and intensity for vector size; dark to light means small to large as seen in Figure 1(d).

Noise: A mobile platform has to deal with non-controlled environments. Thus, we consider it necessary to test the robustness of the algorithms in different situations. Therefore, we corrupt our data set with three different kinds of noise: brightness differences, blurring, and Gaussian white noise. As a consequence of the movement of the platform, brightness on images can change from one frame to another or even between the left and right image in the same frame of a stereo sequence. Blurring may be caused by differences in the focus of the lenses due to movements of the platform. Gaussian noise is present in images taken, even with modern camera technology. Note that we are aware that this may not be an extensive noise list, but it is sufficient to show the importance of testing algorithms in different conditions.

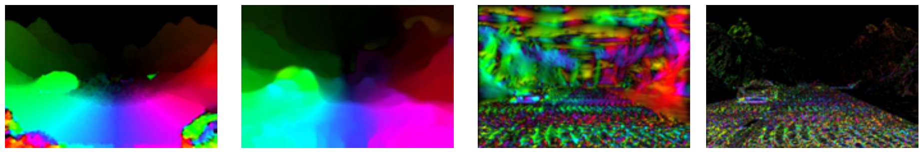
To alter the brightness of the images, we add a constant brightness c to each pixel of every image. Blurring was applied to the sequence by convolving the images with a Gaussian smoothing kernel of size k. Finally, the Gaussian noise was generated by adding at each pixel random Gaussian (normal distribution) white-noise $\mathcal{N}(\mu, \sigma)$, with a mean μ of zero, and a varying standard deviation σ. The parameters are varied over the sequence and presented in Table 1.

To evaluate the motion algorithms we modify the left images using the parameters defined for the right images (Table 1), with the exception of the brightness constant: where $c = t - 52$ is used for even t and $c = 51 - t$ for odd t.

Quality Metrics: Following [12] and [3], for the evaluation of stereo algorithms we use Root Mean Squared (RMS) and % Bad Percentage Pixels (BPP). For BPP we used two different thresholds. Motion algorithms were evaluated using the Average Angular Error (AAE) and the End Point Error (EPE) as

	Left Image		Right Image	
Noise Method	$1 \leq t \leq 50$	$51 \leq t \leq 100$	$1 \leq t \leq 50$	$51 \leq t \leq 100$
Brightness	$c = t - 50$		$c = 50 - t$	
Gaussian Noise	$\sigma = t$	No noise	$\sigma = t$	$\sigma = 101 - t$
Gaussian Blur	$k = 2t - 1$	No noise	$k = 2t - 1$	$k = 203 - 2t$

Table 1. Parameters of the different kind of noise added to the sequence.



(a) BBPW original (b) BBPW blurred (c) CLG brightened (d) CLG G. noise

Fig. 2. Results of top motion algorithms, with respect to EPE mean, for the original sequence and every kind of noise, on left images 41-42.

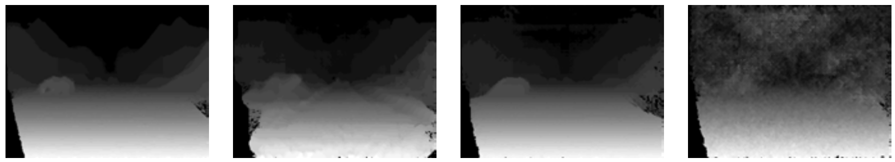
quality metrics, also used (for example) in [16]. The metrics used here allow analysis performance similar to the approach at Middlebury [2], but as we are working with long sequences, we can make statistical inference from the obtained data, such as the mean, zero-mean variance, maximum and minimum for each error metric over the sequence.

3 Results

As this is a brief conference publication, only the results for AAE for motion algorithms are presented. For stereo algorithms, just the RMS results for the original and bright altered sequences are shown; however, for our discussion we use the results obtained from all four of the metrics. We include resultant images obtained with the algorithms with all kinds of noise (see Figures 2 and 3). For a more detailed report concerning the stereo algorithms, see [12] (and more detailed results of the motion algorithms are available on the Technical Report No. 32 from [4]).

Noise-Free Results: The results obtained with the original sequence are the base for the robustness analysis, see Figures 4(a) and 6(a). For motion algorithms BBPW perform the best with both metrics followed by CLG. In the AAE graph it can be observed a considerable large difference in the magnitude among BBPW and the other two algorithms. It is worth to say that for EPE, CLG performs better than BBPW, and its range is the minimum one of all three techniques. For stereo algorithms, the best algorithms for both metrics are the SGM ones, followed by BP and finally the DP algorithms. The difference in magnitudes between the SGM algorithms and the other ones is noticeable. The best algorithm was SGM BT and the best among the DP algorithms is DP closely followed by the other three. For these sequences RMS and BPP show almost the same information.

Gauss Blur Results: BBPW outperforms the other motion algorithms with both metrics. With AAE all the algorithms improve their performance with respect to the noise-free results except for CLG when the blurring is maximum, see Figures 5(a) and 4(a); the change in magnitude is highly noticeable. The improvement is most likely because all the algorithms improve their performance on the road area. The three algorithms behave the best when the amount of blur is medium and peak when the blur is minimum and maximum. The same behavior is observed with EPE.



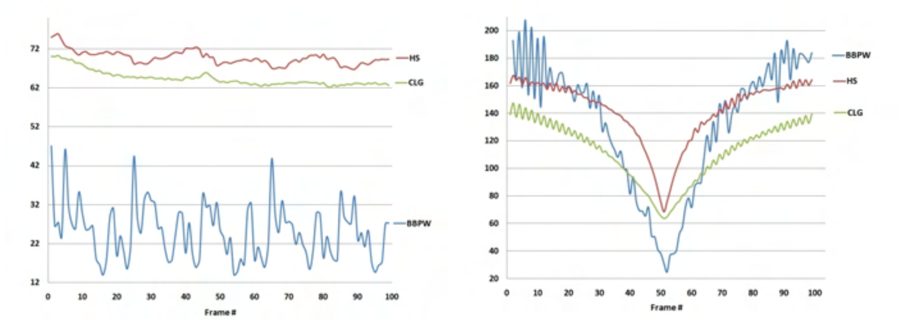
(a) SGM-BT: original sequence. (b) SGM-BT: blurred sequence. (c) SGM-MI: brightness altered. (d) SGM-BT: Gaussian noise added

Fig. 3. Stereo results of the top performing algorithms, for various kinds of noise and the original image sequence, here shown for image pair #40.

The interesting observation for the stereo algorithms is that when the blurring is in both images, the results are not so bad, but when the blurring is removed from the left image, the results get worse for all algorithms. Again SGM BT is the best, SGM MI seems to have the same problem if both images are blurred, or just one. For BPP the interesting point is that both DP and DPt ranked higher than SGM MI.

Brightness Difference Results: This was the noise that had the biggest impact on the results for both stereo and motion algorithms. The ones that perform the best with the original sequence are the worst in this case (for all the respective metrics). SGM BT and BP are tremendously affected, while SGM MI and the dynamic programming algorithms are relatively robust to this kind of noise. For the flow algorithms, CLG was the best and BBPW produced useless data until the difference in brightness is around 10%. See Figures 4(b) and 6(b).

Gauss Noise Results: The algorithms are very sensitive to this kind of noise too. In this case BBPW was the best for AAE and CLG for EPE, see Figure 5(b). For the latter metric there is a noticeable overlapping in the graphs for all of the algorithms. For stereo, SGM BT is the best algorithm, and among dynamic programming algorithms, DPt is the best and DP the worst. The difference between them is enough to make DPt the best overall (see Table 1(b)) dynamic programming algorithm. SGM BT and BP are relatively robust to this alteration of the images.

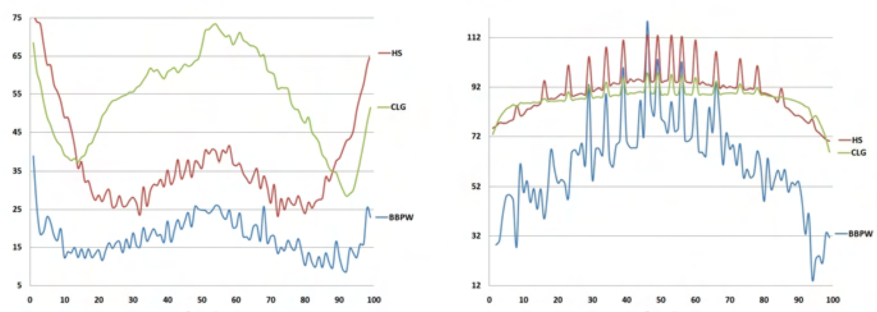


(a) AAE on original sequence. (b) AAE on brightness altered sequence.

Fig. 4. AAE results on original and brightness altered sequence.

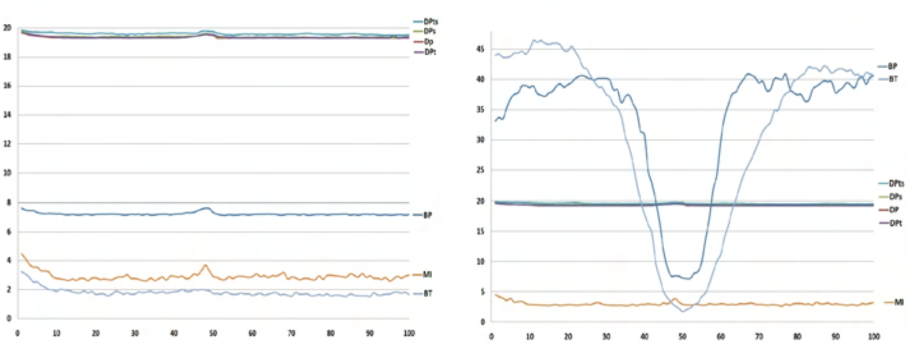
Summary: In the Table 2(a) we present the overall statistics for the motion algorithms when the results are analyzed with AAE. BBPW is clearly the best (respect to the mean and standard deviation) for the sequences analyzed here. However, note that its dynamic range is the largest one, due to the very bad results that were obtained with the brightness altered sequence. The bad results with this sequence were compensated with the improvement on the blurred sequence. It is worth noting that for EPE, in the overall statistics (not shown in this report), BBPW was the worst algorithm, once again due to results obtained with the bright altered sequence. CLG was the best for this metric. Finally, HS show an improvement with the blurred sequence, but it has also a bad performance with the brightness altered sequence.

For stereo algorithms (see Table 2(b) for RMS results) the one with better overall performance was SGM MI, with all the metrics used (for stereo algorithms) in this paper. Note that SGM BT outperform best than SGM MI in all



(a) AAE on blurred sequence. (b) AAE on Gauss. noise added sequence.

Fig. 5. Motion algorithm results for blurred and Gaussian noise sequence.



(a) RMS on original sequence. (b) RMS on brightnress altered sequence

Fig. 6. RMS stereo results on the original and blurred sequences.

(a) Overall motion results for AAE.

Algorithm	Mean	St. Dev.	Min.	Max.
BBPW	59.17	158.01	8.65	207.73
CLG	79.57	167.52	28.45	147.59
HS	84.72	187.83	23.16	167.36

(b) Overall stereo results for RMS.

Algorithm	Mean	St. Dev.	Min.	Max.
SGM MI	7.23	20.45	2.61	23.95
SGM BT	11.21	36.51	1.52	46.45
BP	15.41	39.73	6.99	40.92
DPt	20.56	41.47	19.30	29.49
DP	21.00	42.37	19.30	30.28
DPs	21.40	43.37	19.32	32.63
DPts	21.54	43.58	19.48	32.69

Table 2. Overall results of the four sequences.

the sequences except in the brightness altered one, where its performance was not good at all. The DP algorithms where the worst ones with RMS, with DPt performing the best among them. It is worth to say that DPt has a better overall performance than SGM MI for the Bad Pixel metrics. BP was always below the two best algorithms, showing its worst performance in the bright sequence as with SGM BT.

We briefly report about the average computational time of all the algorithms evaluated in this study. The algorithms were processed on an Intel Core 2 duo E800 at 3.00Ghz under a Windows platform; except for the SGM algorithms (the only time-optimized ones) which were processed on an Intel Centrino Duo at 2.4Ghz using a Linux OS. The dynamic programming algorithms process, on average, one stereo pair in 63.615 s, with DP being the fastest and DPts the slowest. BP was the slowest of the stereo algorithms with a processing time of 106.2235 s. SGM MI processed a stereo pair in 3.974 s. The fastest of the stereo algorithms was SGM BT with 0.741s per pair.

For the motion algorithms, CLG was the fastest processing one pair of images in 27.780 s; this is followed by HS, whose computing time was 39.038 s. The slowest motion algorithm was BBPW, with 90.383 s per pair of images.

4 Conclusions and future work

In this paper we presented an approach to evaluate the robustness of stereo and motion algorithms over a long synthetic sequence. In order to do this we tested several algorithms over a long synthetic sequence, which was corrupted with different kinds of noise. From our results it is clear that most of the algorithms are very sensitive to brightness differences. This has to be highlighted as changes in illumination is one of the most common problems that mobile devices have to deal with. The SGM BT stereo algorithm, whose results were the worst with this type of noise, was the best in all the other sequences. A similar behavior presented the BBPW motion algorithm. As a direct consequence of using long sequences we were able to observe that DPt represent a good option for the dynamic programming algorithms. The future work will include a wider set of noise types, more challenging sequences (real ones), a more in depth study on the quality metrics and a way to evaluate precisely the performance of the algorithms when there is no ground truth available.

Acknowledgments: The authors acknowledge valuable support by members of the *.enpeda..* team at Auckland university, in particular by Xi (Todd) Yang, James Milburn, and Xuan (Sean) Guo.

References

1. R. Mohan, G. Medioni, and R. Nevatia, “Stereo error detection, correction, and evaluation,” *Pattern Analysis and Machine Intelligence*, vol. 11, pp. 113–120, 1989.
2. ——. Middlebury stereo and motion evaluation. [Online]. Available: `http://vision.middlebury.edu/`
3. D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. of Computer Vision*, vol. 47, pp. 7–42, 2002.
4. *.enpeda..* image sequence analysis test site (EISATS). [Online]. Available: `http://www.mi.auckland.ac.nz/EISATS/`
5. T. Vaudrey, H. Badino, and S. Gehrig, “Integrating disparity images by incorporating disparity rate,” in *Robot Vision*, pp. 29–42, 2008.
6. Y. Ohta and T. Kanade, “Stereo by two-level dynamic programming,” in *Proc. IJCAI*, pp. 1120–1126, 1985.
7. Z. Liu and R. Klette, “Dynamic programming stereo on real-world sequences”, In *Proc. ICONIP*, LNCS, 2008.
8. P.F. Felzenszwalb and D.P. Huttenlocher, “Efficient belief propagation for early vision”, *Int. J. Computer Vision*, vol. 60, pp. 261–268, 2006.
9. S. Guan, R. Klette, and Y.W. Woo, “Belief propagation for stereo analysis of night-vision sequences”, in *Proc. PSIVT*, LNCS 5414, pp. 932–943, 2009.
10. H. Hirschmüller, “Accurate and efficient stereo processing by semi-global matching and mutual information”, in *Proc. Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 807–814, 2005.
11. Open Computer Vision Library (OpenCV). Available: `http://sourceforge.net/projects/opencvlibrary/`
12. S. Morales, T. Vaudrey, and R. Klette, “An in depth robustness evaluation of stereo algorithms on long stereo sequences”, *Tech. Rep. TR-30*, Available: `http://mi.auckland.ac.nz`, University of Auckland, 2008.
13. Y. Satoh, K. Skaue, “An Omnidirectional Stereo Vision-Based Smart Wheelchair”, in *EURASIP Journal on Image and Video Processing*, vol. 2007, Article ID 87646, 11 pages.
14. M. Seelinger, J. D. Yoder, “Automatic Pallet Engagement by a Vision Guided Forklift”, in *Proc. International Conference on Robotics and Automation (ICRA)*, pp. 4068-4073, 2005.
15. T. Brox, A. Bruhn, N. Papenberg and J. Weickert. “High Accuracy Optical Flow Estimation Based on a Theory for Warping”, in *Proc. 8th European Conference on Computer Vision (ECCV)*,LNCS 5414, vol. 4 pp. 25–36, 2004.
16. S. Baker, S. Scharstein, J. P. Lewis, S. Roth, M. J. Black and R. Szelisky. “A Database and Evaluation Methodology for Optical Flow”, in *Proc. IEEE 11th International Conference on Computer Vision (ICCV)*, pp. 1–8, 2007.
17. J. Barron, D. Fleet and S. Beauchemin. “Performance of Optical Flow Techniques”, in *International Journal of Computer VIsion (IJCV)*, vol. 12(1), pp. 43–77, 1994.
18. A. Bruhn, J. Weickert, C. Schnrr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. Updated version with errata. *Int. J. Computer Vision*, Vol. 61:211–231, 2005.