Evaluation of Motion Analysis on Synthetic and Real-World Image Sequences

Xi Yang and Reinhard Klette
The *.enpeda..* Project
The University of Auckland, Auckland, New Zealand
xyan081@aucklanduni.ac.nz and *r.klette@auckland.ac.nz*

Abstract—**Motion analysis is of basic importance for vision-based driver assistance systems. Optical flow is a common image motion representation. This paper provides comparative evaluations of four different optical flow algorithms (called Horn-Schunck, CLG, BBPW, and TV-L_1 in the paper). The algorithms are tested for synthetic and recorded sequences on original image data, on edge maps, and on residual images after applying a smoothing operator. For testing robustness, the algorithms are also analyzed on synthetic sequences with artificial noises (Gaussian blur, Gaussian white noise, or constant brightness changes). A comparative discussion of the algorithms is provided at the end.**

Keywords: Motion analysis, optical flow, performance evaluation, Horn-Schunck, CLG, BBPW, TV-L_1

I. Introduction

Since the papers [1], [7] in 1994 and 1996, respectively, on performance evaluation of optical flow algorithms, many papers have been published on this topic. Those evaluations typically focus on the determination of the accuracy of algorithms using test sequences with available ground truth; the used sequences are either generated by a computer program, or 'engineered' in an indoor environment, such that motion ground truth is available in both cases. The performance is then judged by applying error measures, comparing results against ground truth. Those synthetic or 'engineered' sequences are typically short, only a few frames. Obviously, this is not adequate to evaluate the algorithms in depth for changing situations as occurring in a vision-based driver assistance systems (DAS) context [5].

This paper refers to four different optical flow algorithms, which are hierarchical Horn-Schunck [8], the combination of local and global (CLG) analysis in [3], BBPW, which is short for the four co-authors of [2], and (the improved) total-variation method TV-L_1 [14] which aims at using the L_1- rather than the L_2-metric.

For testing we use long synthetic sequences, which are available in Set 2 of EISATS [4] and real-world sequences recorded on Auckland's roads. All the used sequences contain more than 100 frames each. Ground truth is available for the synthetic sequences, and we discuss ways how to obtain approximate ground truth for the real-world sequences. Figure 1 illustrates a synthetic sequence.

The evaluation in this paper is mainly about the robustness of the algorithms, defined by average behavior on long image sequences.

For the synthetic sequences, not only a *default driving situation* (i.e., daylight, no rain or snow, no difficult lighting conditions, and so forth) is tested, but we also add artificial noises to imitate natural phenomena, such as illumination artifacts, or extreme weather.

The real-world sequences are taken under different driving situations, such as driving towards a wall, parallel to a wall, through a tunnel, or into a parking lot. These situations are characterized by some kind of "simple environment geometry". In this paper we illustrate for the case of driving towards a wall, how this may be mapped into some estimated ground truth.

II. Experimental Scheme

A. Optical Flow Algorithms

The four algorithms have been selected for being rather representative for different ways of calculating optical flow. The Horn-Schunck algorithm was historically first, still referenced frequently in today's publications, and characterized by local iterations. The CLG method

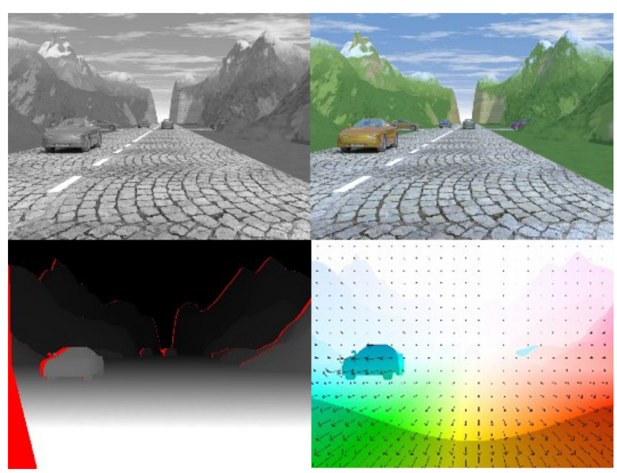

Fig. 1. Top left: gray-level image of Sequence 1 in Set 2 of EISATS. Top right: color image of this sequence. Bottom left: ground truth of depth (red pixels indicate occlusions). Bottom right: ground truth of optical flow. See [4].

combined local with global analysis, and is known for being tolerant to noise. The algorithm by Brox, Bruhn, Papenberg and Weickert (BBPW) implements a warping technique. Finally, TV-L_1 is based on total variation with respect to (basically) the L_1 metric, and has a high ranking on [10].

B. Evaluation Metrics

In our evaluation, we chose two quality metrics which are in use since the 1990s, and also on [10].

1) Angular Error: The *angular error* $E_{AE}(p)$ between two flow vectors $\mathbf{v_0}(\mathbf{p}) = (u_0, v_0)$ and $\mathbf{v_1}(\mathbf{p}) = (u_1, v_1)$ at pixel p is the angle between $(u_0, v_0, 1)$ and $(u_1, v_1, 1)$ in three-dimensional space. First, the vectors may be normalized:

$$\widetilde{\mathbf{v}} = \frac{(v_0, v_1, 1)^T}{\sqrt{(v_0^2 + v_1^2 + 1)^2}}$$

Then we obtain that

$$E_{AE}(p) = \arccos(\widetilde{\mathbf{v}}_0^T \cdot \widetilde{\mathbf{v}}_1)$$

This angular error (AE) is convenient for handling both very large and small velocity. If the evaluation is on sequence with given motion ground truth, then $E_{AE}(p)$ is the angle between estimated flow and true flow [1].

2) End Point Error: The *endpoint error* (EPE) is defined as the distance between flow endpoints, which is

$$\sqrt{(u_0 - u_1)^2 + (v_0 - v_1)^2}$$

If the evaluation is on sequence with given motion ground truth, then $\mathbf{v_0}(\mathbf{p}) = (u_0, v_0)$ is ground truth flow and $\mathbf{v_1}(\mathbf{p}) = (u_1, v_1)$ is the estimated flow.

Furthermore, we also use the mean angular error (MAE) and the mean end point error (MEPE) when evaluating the performance over a whole sequence.

3) Adding Noise: For robustness evaluation of the algorithms, [12] degraded given synthetic sequences by noise of varying intensity. We denote by $I_{in}(p,t)$ the image value at pixel position p at time t in the input (i.e., recorded) image data. Three types of noise are applied, Gaussian blur, Gaussian white-noise, and brightness changes which are constant with each of the frames. To reflect the effect of the noise, the amount of noise varies from image to image in the sequence. This evaluation process is sketched in Fig. 2.

4) Gaussian Blur Blur is happening often in real world driving sequences. To find out the algorithms' tolerance with respect to blur noise, an approximate blurring effect was generated using a Gaussian blurring convolution

$$I_{out}(p,t) = I_{in}(p,t) \times G(k)$$

where $G(k)$ represents a $k \times k$ Gaussian smoothing kernel [12]. The noise parameters are increased through the first half of the sequence. At the middle frame, the amount reaches its maximum. From the beginning of the second half, the amount of noise starts to decrease. Sample frames are shown in Fig. 3.

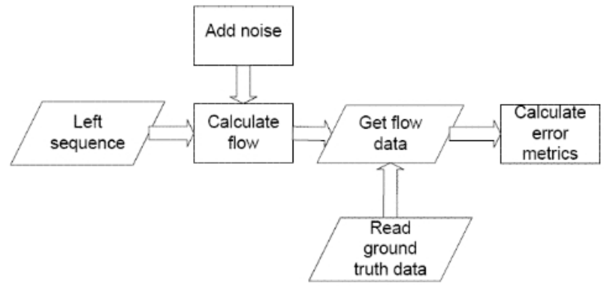

Fig. 2. The process of evaluating by adding various types of noise to the left sequence of a stereo sequence [12].

5) Gaussian White Noise This kind of noise is common in recorded images. There are always small amounts of Gaussian white noise present. We simulate this noise on synthetic sequences. The noise is added randomly for selected pixels. The random Gaussian (i.e., normal distribution) process is denoted by $N(\mu, \sigma)$, where μ is the expected value and σ is a varying standard deviation, to be changed from small to large. Following [12], the noise is defined as follows:

$$I_{out}(p,t) = I_{in}(p,t) + N(0, \sigma)$$

6) Constant Brightness Changes

This event happens frequently when driving on the road. For example, driving below trees, into or out of a shadow, turning at a corner, driving into a tunnel, and so forth. To simulate this, a constant brightness value was added or subtracted to or from all pixels of an image:

$$I_{out}(p,t) = I_{in}(p,t) \pm c$$

where c is a positive constant [12]. For odd frames, the constant c will be added, and for even frame numbers, c will be subtracted.

III. Evaluation on Synthetic Sequences

We first discuss results for the synthetic Sequence 1 of EISATS Set 2, as available on [4], and at the end briefly also for Sequence 2 of the same set.

Fig. 3. Sample images in a blurred sequence of 100 frames: Frames 1, 50 and 88.

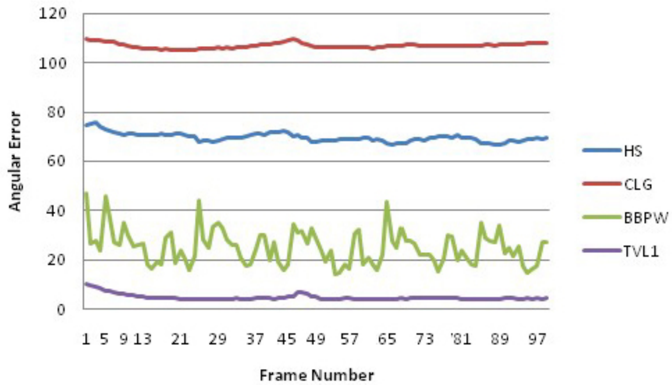

Fig. 4. Angular error for Sequence 1 of Set 2. No addition of noise.

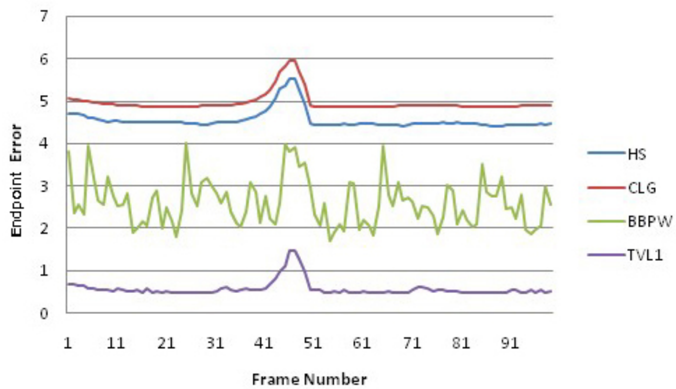

Fig. 5. End point error for Sequence 1 of Set 2. No addition of noise.

1) Original Sequence: Figures 4 and 5 illustrate the results for the original sequence.

From the graphs it is obvious that TV-L_1 has the best performance. The mean angular error of TV-L_1 equals $4{\cdot}77$, compared to Horn-Schunck with $69{\cdot}75$, CLG with $106{\cdot}91$, and BBPW with $25{\cdot}08$. The mean end point error of TV-L_1 equals $0{\cdot}56$, compared to Horn-Schunck wit $4{\cdot}55$, CLG with $4{\cdot}95$, and BBPW with $2{\cdot}58$.

Error values are high at the very beginning of the sequence because a vehicle appears in the scene. Near frame 45, there are peaks in both figures, and this is when a vehicle comes towards the camera and disappears soon from the left corner of the scene. This means that the algorithms could not handle new objects as well as existing moving objects. This phenomenon is more obvious by end point error, which is shown by a single peak over the whole sequence.

BBPW shows different trends (more variation in errors) then the others. Figure 6 shows a sample of a BBPW flow image for Sequence 1.

There are too many errors at the bottom left and right corners. These errors may occur because of the selected $gamma$ and $alpha$ values in the BBPW algorithm, and could be investigated further in future work. These

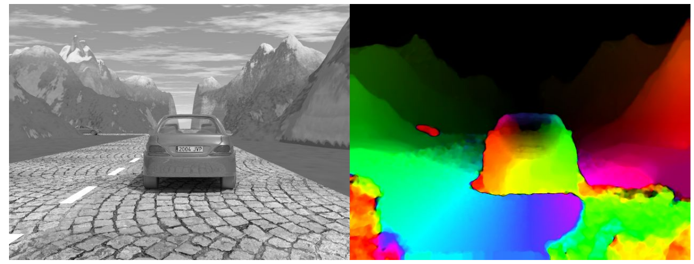

Fig. 6. Left: the first frame of Sequence 1. Right: BBPW flow image. The left and right bottom corners have too many errors.

errors may lead to fluctuations, and increase the error. The original image size is 640×480. We were cutting off the lower part of the image to remove those two error-infested areas, and tested BBPW on a resulting 640×360 sequence. The mean angular error reduces to $5{\cdot}93$, and the mean end-point error to $0{\cdot}32$.

2) Gaussian Blur Sequence: Figures 7 and 8 show test results when Sequence 1 of Set 2 was degraded by Gaussian blur. The CLG error values are the largest.

Gaussian blur defines a a smoothing operator which

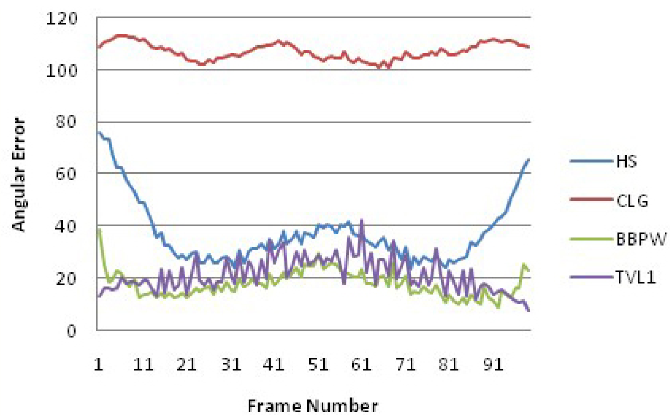

Fig. 7. Angular error for Sequence 1 of Set 2. With Gaussian blur.

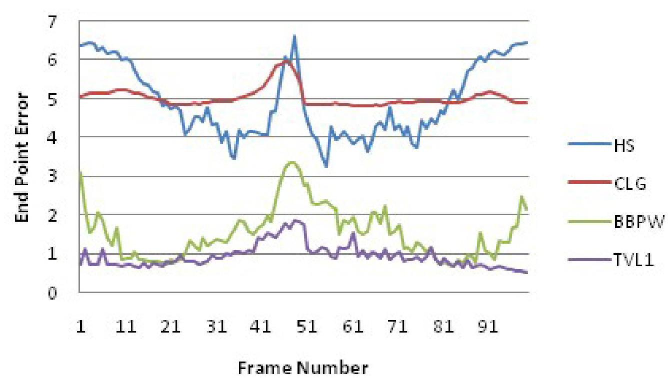

Fig. 8. End point error for Sequence 1 of Set 2. With Gaussian blur.

improves the overall performance. For example, the mean angular error of Horn-Schunck improves from 69·75 to 37·14. The two ”challenging” corners of BBPW now have been “smoothed out”, and the mean angular error is 17·65.

For Horn-Schunck, the start values of the AE are high, but as the blur increases, the AE values drop dramatically. The EPE is still sensitive to the oncoming vehicle, since it is close to the camera and disappears quickly; the EPE of Horn-Schunck fluctuates very much.

Applying a smoothing function prior to Horn-Schunck improve its performance significantly on this sequence, but Horn-Schunck is also very “sensitive” in the blurred case to objects appearing and disappearing in the scene. Another issue is that Horn-Schunck and CLG cannot compute the flow field on the road very well.

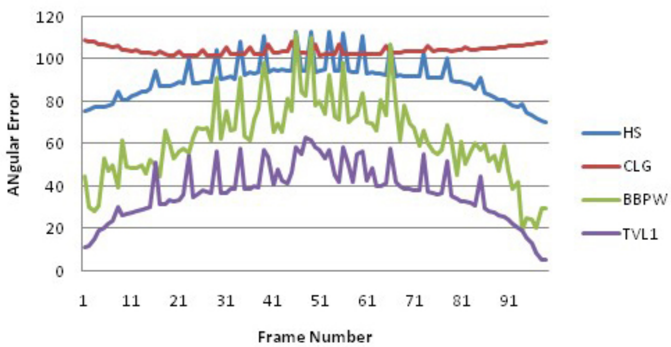

Fig. 9. Angular error for Sequence 1. With Gaussian white noise.

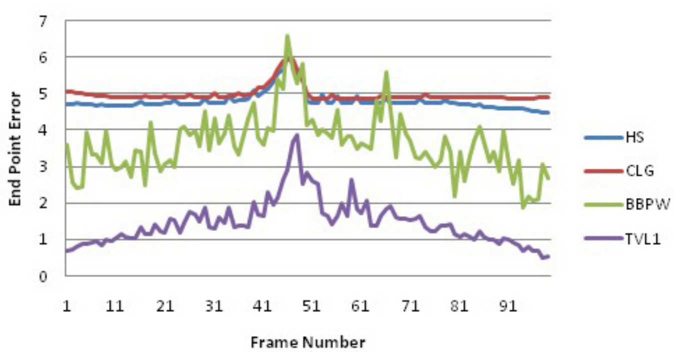

Fig. 10. End point error or Sequence 1. With Gaussian white noise.

3) Gaussian White-noise Sequence: Figures 9 and 10 show results for Sequence 1 with Gaussian white noise preprocessing.

The errors in the graph fluctuate all the time for all the algorithms, for AE and EPE. There are many peaks at certain intervals. These peaks are due to the amount of noise. We increased the amount of noise after every four or five frames. The graphs show that once the amount of noise is increased, the error values go up immediately. At the subsequent frame, the errors drop down again to their previous levels. Over the whole sequence, errors increase with the increase of noise.

Horn-Schunck and CLG’s performance are relatively stable, but produce large errors. BBPW errors fluctuate the most, and the difference between peak and bottom values is the largest. TV-L_1 has the best performance.

Around frame 45, all algorithms have suddenly peaks in error values for both AE and EPE, and in particular for the EPE.

4) Constant Brightness Change Sequence: Figures 12 and 13 show results for Sequence 1 when degraded by constant brightness changes. Figure 11 illustrates the changes in brightness over the whole sequence.

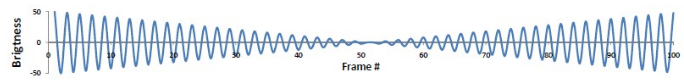

Fig. 11. Constant changes of brightness over the sequence [4].

At the very beginning of the sequence, the brightness difference between two consecutive frames is the largest. This big difference causes all the AE values to be larger than 150. As the brightness difference decreases between subsequent frames, especially between frame 30 and 60, the error values drop down as well. The AE and EPE come down to a normal level. As shown in Figure 12, AE values have an obvious turning point at frame 50, and

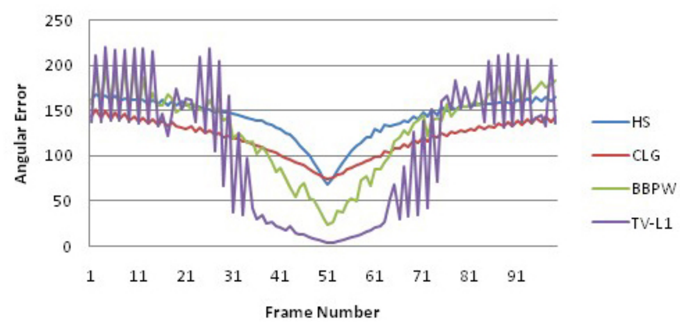

Fig. 12. Angular error for Sequence 1. Constant brightness changes.

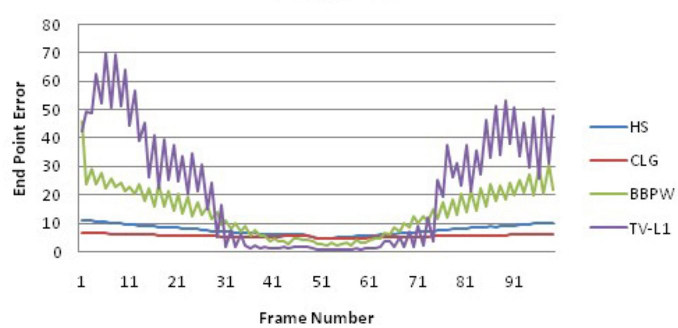

Fig. 13. End point error for Sequence 1. Constant brightness changes.

Sequence	Algorithm	MAE	MEPE
Original	Horn-Schunck	65·76	4·55
	CLG	106·91	4·95
	BBPW	25·08	2·58
	TV-L_1	4·77	0·56
Gaussian Blur	Horn-Schunck	37·14	4·94
	CLG	107·24	5·04
	BBPW	17·65	1·55
	TV-L_1	21·40	0·93
Gaussian White-noise	Horn-Schunck	89·99	4·80
	CLG	103·72	4·96
	BBPW	62·29	3·63
	TV-L_1	37·02	1·47
Brightness Change	Horn-Schunck	142·00	7·48
	CLG	119·12	5·47
	BBPW	129·56	13·81
	TV-L_1	107·58	21·72

TABLE I
MEAN AE (MAE) AND MEAN EPE (MEPE).

Sequence	Algorithm	MAE	MEPE
Original	Horn-Schunck	80·54	4·70
	CLG	78·14	4·67
	BBPW	65·32	5·08
	TV-L_1	37·31	3·37
Gaussian Blur	Horn-Schunck	87·91	4·79
	CLG	79·91	4·66
	BBPW	39·21	2·89
	TV-L_1	52·29	2·45
Gaussian White-noise	Horn-Schunck	81·36	4·86
	CLG	81·68	4·91
	BBPW	100·41	5·57
	TV-L_1	35·57	4·46
Brightness Change	Horn-Schunck	80·69	4·71
	CLG	78·43	4·67
	BBPW	57·90	4·76
	TV-L_1	38·50	3·41

TABLE II
MEAN ANGULAR ERROR (MAE) AND MEAN END POINT ERROR (MEPE) OF ALL TESTED SEQUENCES APPLYING SOBEL OPERATOR.

EPE values in Figure 12 are very low between frames 30 to 60.

5) Summary: Thus, all the algorithms have difficulties in dealing with brightness changes. Table I shows the mean AE and mean EPE for all cases, with or without added noise.

Obviously, no algorithm could deal with the brightness changes. All MAE values are above 100 in that case, even for TV-L_1, whose MAE was 4·77 on the original sequence. This is no surprise because the brightness constancy assumption is part of all four methods.

Gaussian blur is the only noise that improved the performances of some algorithms at some frames, rather than making them worse in general. The blur helps Horn-Schunck and BBPW to improve their performance significantly, where the MAE of Horn-Schunck improved by 56% compared to the original sequence! For BBPW, the blur improves its performance both on AE and EPE; the MAE improved from 25·08 to 17·65, and the MEPE improved from 2·58 to 1·55. On the other hand, Gaussian blur makes TV-L_1 much worse, especially on the AE values.

The most important outcome, demonstrated by these evaluation results, is that we need to find a way to deal with illumination changes. This problem always occurs in real world driving assistance systems.

Next, in an attempt to solve the problem, Sobel and residual operators will be tested for preprocessing.

6) Sobel Pre-Processing: In [6] it was pointed out that Sobel preprocessing was of benefit for correspondence analysis on images sequences of Set 1 on EISATS [4].

Table II shows MAE and MEPE of sequences preprocessed by the Sobel operator. Compared to the results without any pre-processing, the performance of all algorithms has improved on the brightness-changed sequence. CLG is the only algorithm that has improved its performance after applying all kinds of noise. However, the MEPE results are different: for Horn-Schunck and CLG, it has been improved; for BBPW and TV-L_1, MEPE results are worse.

7) Residual Pre-Processing: Following [13], the residual operator used in this evaluation is defined by firstly smoothing 40 times by the *cvSmooth()* function in OpenCV (i.e., this could also be done in one step with a larger smoothing kernel), after which the *cvSub()* function is used to generate the residual images.

Table III shows the MAE and MEPE for the various sequences resulting from this residual pre-processing.

As indicated in the table, the residual preprocessing improves the performance of all algorithms on the brightness-changed sequence, both in MAE and MEPE. For example, TV-L_1 has been improved from 107·58 to 30·17 on MAE, and from 21·72 to 2·97 on MEPE. Horn-Schunck and CLG perform similarly on the sequences when applying the residual preprocessing; the MAE values of Horn-Schunck and CLG fluctuate between 90

Sequence	Algorithm	MAE	MEPE
Original	Horn-Schunck	90·01	4·81
	CLG	99·21	4·90
	BBPW	82·43	1·87
	TV-L_1	29·25	2·94
Gaussian Blur	Horn-Schunck	93·51	4·84
	CLG	99·25	4·90
	BBPW	138·03	6·22
	TV-L_1	35·43	1·97
Gaussian White-noise	Horn-Schunck	97·24	4·88
	CLG	99·22	4·90
	BBPW	136·5	6·29
	TV-L_1	74·54	4·23
Brightness Change	Horn-Schunck	90·17	4·82
	CLG	99·20	4·89
	BBPW	140·53	6·40
	TV-L_1	30·17	2·97

TABLE III
MAE AND MEPE FOR RESIDUAL PREPROCESSING.

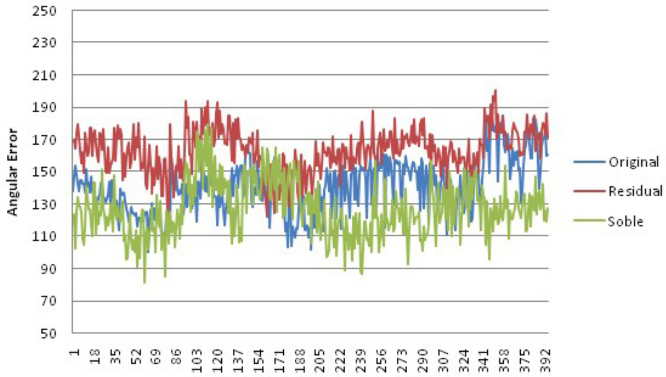

Fig. 14. AE of BBPW on the original sequence, or preprocessed with Sobel or residual operator.

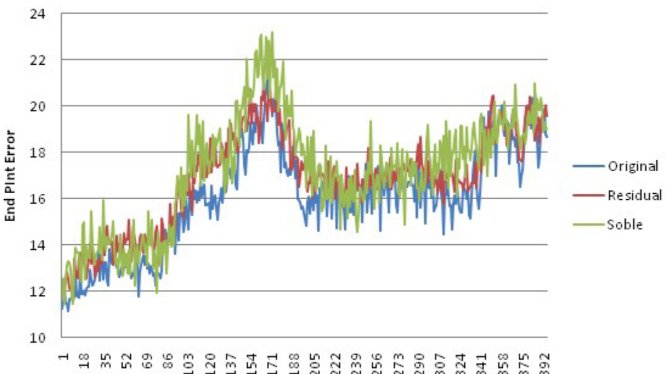

Fig. 15. EPE of BBPW on the original sequence, or preprocessed with Sobel or residual operator.

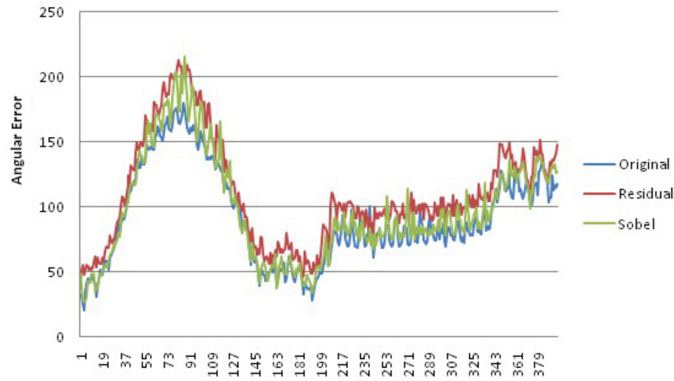

Fig. 16. AE of TVL_1 on the original sequence, or preprocessed with Sobel or residual operator.

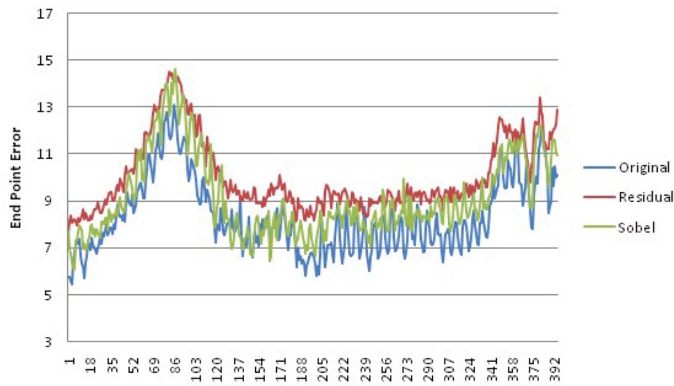

Fig. 17. EPE of TVL_1 on the original sequence, or preprocessed with Sobel or residual operator.

and 99, and the MEPE values are between $4{\cdot}80$ and $4{\cdot}90$. The reason is that residual preprocessing removes some low frequency elements from the image. Again, this is not surprising, because CLG is by design "partially Horn-Schunck".

For the other cases, all error values go up to some extent. BBPW performs here the worst, though its MEPE is the best on the original sequence. After cutting off the bottom section with those challenging corners, the MAE results still show that BBPW performs the worst. This result shows that for the given synthetic sequence, residual preprocessing is not reducing the effect of illumination changes for BBPW, while error values even increase in the other cases.

Comparing results for Sobel preprocessing with those when applying a residual operator, only BBPW's MAE on Gaussian white-noise and TV-L_1's MEPE on all sequences got worse. All the others perform actually better for Sobel preprocessing than using the residual operator.

8) Sequence 2: Finally for this section, we also briefly summarize our findings on Sequence 2 of Set 2 on EISATS [4]. This sequence is much longer than Sequence 1. It contains 396 frames. This sequence has less traffic, but has more plants, such as trees and grass. The plants cause more difficulties for computing corresponding pixels. From the results on Sequence 1, we know that BBPW and TVL_1 performed there better than CLG or Horn-Schunck. For discussing Sequence 2, we just focus on BBPW and TVL_1.

BBPW. The evaluations are carried out on the original sequence, the one pre-processed with the residual operator, and the one pre-processed with the Sobel operator. Results are shown in Figures 14 and 15. AE results prove again that the algorithm performs best for the Sobel operator, although the EPE tells something different.

Figure 15 shows that the error values keep increasing from Frame 1 to Frame 170 in all cases. This is because the ego-vehicle is driving uphill in those frames, and more and more sky area appears in the images. Sky and clouds are considered at infinity to the camera. Therefore, they are almost static or only have tiny movements, and affect the EPE a lot.

TVL_1. The evaluations are again done on the original sequence, pre-processed with residual operator, or preprocessed with Sobel operator.

Figures 16 and 17 show that both AE and EPE values keep increasing at the very beginning due to the no movement of sky and clouds, a problem similar to BBPW. The peak values of AEs are even above 200, double the MAE over the whole sequence.

It is obvious that neither BBPW nor TVL_1 were able to compute the movement of sky or clouds very well, even for the Sobel or residual operator. This is because sky and cloud areas do not provide sufficient information for optical flow computation. Luckily, this issue is not of much relevance for vision-based DAS.

IV. Evaluation on Real-World Sequences

We discuss cases of real-world sequences, recorded in Auckland at locations where the environment may be geometrically approximated by some simple models.

A. Driving Towards a Wall

We can estimate the ground truth for this kind of sequence. Let W be the width and H the height of the given frames (in pixels). The optical flow $\mathbf{u} = (u, v)$ is approximately

$$u = (\frac{S_t}{S_{t+\delta t}} - 1)(i - \frac{W}{2}) \text{ and}$$
$$v = (\frac{S_t}{S_{t+\delta t}} - 1)(\frac{H}{2} - j)$$

where S_t and $S_{t+\delta t}$ are the distances between camera and wall at time slots t and $t + \delta t$. These two distances can not be measured accurately for δt equals $1/30$ of a second because we do not know the exact speed of the vehicle at the required level of accuracy.

However, for estimating the value of $S_t/S_{t+\delta t}$, we can run one of the optic flow algorithms (say, TV-L_1) on two consecutive image frames first. Using all the calculated values u and v, we estimate $S_t/S_{t+\delta t}$. Finally, we use the estimated ratio $S_t/S_{t+\delta t}$ to have estimated ground truth; AE and EPE are then calculated with respect to those vectors.

There are changes in illumination in recorded sequences. We use residual preprocessing to reduce the impact of those changes. We discuss here results for two recorded sequences towards the same wall. These sequences have different numbers of frames, due to the ego-vehicle's different speed and different start distances to the wall.

To avoid the influence of other objects, the recorded scene was basically only showing the wall and some ground area, not any other objects. This way, the optical flow vectors are basically only related to distances. When the ego-vehicle was too close to the wall, the camera was out of focus sometimes, resulting in blurry images, and we did not record very close to the wall for that reason. (Though a small amount of blur may improve the performance of some algorithms, its use is not "fair" for all the algorithms.)

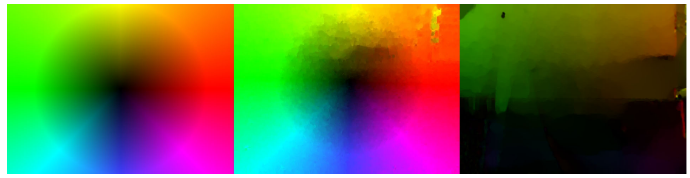

Fig. 18. Estimated ground truth and evaluation results for two selected subsequent frames when driving towards a wall. Left: Ground truth. Middle: Flow result generated by TV-L_1. Right: Flow result generated by BBPW.

These tests showed that TV-L_1 had by far the best performance. Thus, TV-L_1 was used to compute the ground truth.

Table IV shows MAE and MEPE values. TV-L_1 has outstanding performance on AE, and its MEPE is only worse than that of BBPW. In this evaluation, only TV-L_1 could show acceptable results. Figure 18 shows samples of results together with the estimated ground truth flow. We can see that the BBPW result fails to match visually the correct optical flow field. Horn-Schunck and CLG are even worse than BBPW. The approximate distance between camera and wall was about 5 meters.

Algorithm	MAE	MEPE
Horn-Schunck	90·01	4·81
CLG	99·21	4·90
BBPW	82·43	1·87
TV-L_1	29·25	2·94

TABLE IV
MAE and MEPE values of all four algorithms (after applying a residual operator) for both sequences of a driving-towards-a-wall situation.

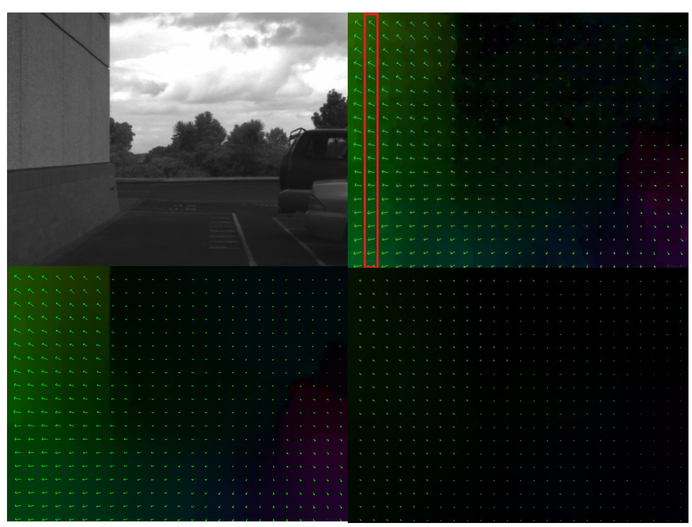

Fig. 19. Computed optical flow results when driving parallel to a wall. Top left: First frame of the used pair. Top right: Flow result generated by TV-L_1. Bottom left: Flow result generated by BBPW. Bottom right: Flow result generated by Horn-Schunck.

B. Driving Parallel to a Wall

Ground truth may also be estimated for this case, similarly to the driving-towards-a-wall case. However, without going into those details, we already know that the optical flow vectors "on the wall" should point backwards and the lengths of the vectors should vary depending on the distances between the projected surface points and the camera. The vectors in a vertical area, which have the same distance to the edge of the wall, should all have about the same length.

BBPW and CLG failed when applied to this sequence. Horn-Schunck performs a little bit better in this case compared to the driving-towards-the-wall sequence. TV-L_1 performs very well; see top right in Figure 19.

C. Driving Through a Tunnel

The used tunnel is relatively short. For recording a sufficient number of frames in a sequence, we slowed down the ego-vehicle. Additionally, pre-processing (Sobel) was use to deal with the changes in lighting.

Figure 20 shows a frame of the sequence and the computed optical flow fields. In New Zealand we drive on the left-hand side. Therefore, the distance between the camera and the wall on the left is greater than the distance to the wall on the right. The larger the distances are, the longer the flow vectors are. The relative directions of velocities at pixels showing the side walls should point backwards; the direction of pixels on the top should point upwards; the direction of pixels on the planar road should point downward.

Comparing the computed optical flow fields with our estimations, only TV-L_1 could generate a fairly correct optical flow field. Horn-Schunck, CLG and BBPW all failed in this case.

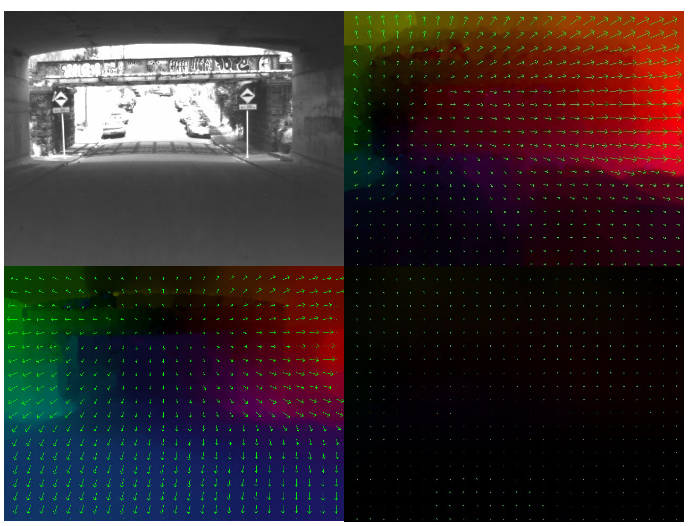

Fig. 20. Computed optical flow results when driving through a tunnel. Top left: First frame of the used pair. Top right: Flow result generated by TV-L_1. Bottom left: Flow result generated by BBPW. Bottom right: Flow result generated by Horn-Schunck.

D. Driving On a Planar Surface

This evaluation was done by recording sequences on the planar surface of a parking lot. The estimation of ground truth follows [9]. Again, only TV-L_1 generated good results. Driving at different speed was part of this evaluation. Obviously, when driving faster, the length of optical flow vectors should increase. Figure 21 proves that TV-L_1 results match this model. But we could also see that TV-L_1 failed if we drove too slowly.

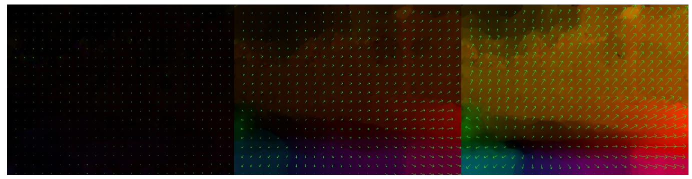

Fig. 21. Left: Optical flow field when driving around 5 km per hour. Middle: Optical flow field when driving around 25 km per hour. Right: Optical flow field when driving around 120 km per hour.

V. Conclusions

TV-L_1 performed best both for the used synthetic sequences and the real-world sequences.

All four algorithms proved to be very sensitive to new dynamic objects moving into the visual field of the camera (e.g., a vehicle coming towards the ego-vehicle). The error values go up significantly in all such cases. After applying noise to the synthetic sequences, the increase of end point errors becomes obvious. Errors also increase with the amount of noise.

Gaussian blur is the only noise that could possibly help those algorithms to improve their performance. A small amount of smoothing appears to be useful, especially for BBPW. The blur helps BBPW to deal with the "bad corners" in the optical flow fields of the synthetic sequences. Those corners, which are at the bottom left and right, have a major impact on error values. After those corners were cut off in the original sequence, BBPW performed even better than TV-L_1 on the synthetic sequences. BBPW performed obviously bad on "quite uniformly textured images", despite of also

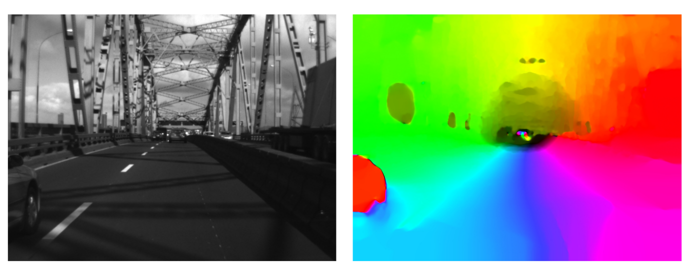

Fig. 22. Left: Driving over the harbor bridge in Auckland. Right: BBPW flow [11].

being often reasonable on real world sequences (see Fig. 22); but it also failed on sequences of extreme lighting situations ("very dark in the tunnel, and very bright behind the tunnel) such as shown in Fig. 20.

Illumination artifacts are basically a "disaster" for all algorithms. The angular errors of TV-L_1 are almost 20 times higher than for the original sequence. As illumination changes are reduced, the errors go down again.

If the amount of Gaussian white-noise is about constant, the algorithms could still compute optical flow fields as before. At the time when the amount of this noise increases, there appears a peak in error. After the change, the errors drop down to the level before.

Sobel and residual operators are helpful for dealing with these noises. Both operators reduce angular errors back to normal level, although they can not reduce end point errors much. In [12], Sobel has been proved to be the best operator to deal with illumination artifacts in most cases. In our evaluation, the results also prove that the overall performance of Sobel operator is better than that of residual preprocessing. Residual operators only performed better than Sobel in the case of TV-L_1 for some sequences.

On real world sequences, recorded with our test vehicle, TV-L_1 was the only (fairly) successful method, especially in cases of slow speed or objects in close distance.

Illumination artifacts exist in all recorded sequences. Before computing optical flow fields, the Sobel or residual operator was applied first. In cases of estimated ground truth, such as when driving towards a wall or parallel to a wall, TV-L_1 perfectly reflected the movement of pixels on the wall. The mean angular error was typically below 30. The only issue that has to be noted is that there should not be other objects in the image; in such cases, errors increase again.

Mean angular errors and end point errors of Horn-Schunck and CLG are relatively high. Mean angular errors of BBPW are high, but its mean end point error is often even better than that of TV-L_1.

Sky and clouds are a general problem for all the algorithms, either because of the minimal movement, or inadequate information for computing corresponding pixels. Even TV-L_1 could not provide good results in this case.

Acknowledgment

The authors thank James Milburn and Xuan Guo for their implementations of the BBPW and CLG algorithms, respectively. (Horn-Schunck was used as available in OpenCV, and TV-L_1 was implemented by the first author.) The authors also thank both reviewers for their comments.

References

[1] Barron, J.L., Fleet, D.J., Beauchemin, S.: Performance of optical flow techniques. *Int. J. Computer Vision*, **12**: 4377, 1994.

[2] Brox, T. Bruhn, A. Papenberg, N. and Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In Proc.*ECCV*, volume 4, LNCS 3024, pages 25–36, Springer, Berlin, 2004.

[3] Bruhn, A. , Weickert, J. and Schnoʹrr, C.: Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *Int. J. Computer Vision*, **61**:211–231, 2005.

[4] EISATS: The *.enpeda..* image sequence analysis test site. http://www.mi.auckland.ac.nz/EISATS.

[5] Franke, U., Rabe, C., Badino, H., and Gehrig, S.: 6D-vision - fusion of stereo and motion for robust environment perception. In: *DAGM*, LNCS 3663, pages 216–223, Springer, Heidelberg, 2005.

[6] Guan, S. and Klette, R.: Belief-propagation on edge images for stereo analysis of image sequences. In: *Robot Vision*, LNCS 4931, pages 291302, Springer, Berlin, 2008.

[7] Handschack, H. and Klette, R.: Evaluation of differential methods for image velocity measurement. *Computers Artificial Intelligence*, **14**:81-104, 1996.

[8] Horn, B. , and Schunck, B.: Determining optical flow. *Artificial Intelligence*, **17**:185–203, 1981.

[9] Liu, Z. and Klette, R.: Approximated ground truth for stereo and motion analysis on real-world sequences. In: *PSIVT*, LNCS 5414, pages 874–885, 2009.

[10] Middlebury College computer vision web site: see http://vision.middlebury.edu.

[11] Milburn, J.: Dense optical flow for night-vision stereo sequences. COMPSCI789 report, Computer Science, The University of Auckland, 2008.

[12] Morales, S. , Vaudrey, T. and Klette, R.: An in-depth robustness evaluation of stereo algorithms on long stereo sequences. In: *IEEE Intelligent Vehicles Symposium*, pages 347–352, 2009.

[13] Vaudrey, T., Wedel, A., and Klette, R.: A methodology for evaluating illumination artifact removal for corresponding images. In: *CAIP*, LNCS 5702, pages 1113-1121, Springer, Berlin, 2009.

[14] Wedel, A. , Pock, T. , Zach, C. and Bischof, H. and Cremers, D.: An improved algorithm for TV-L^1 optical flow. In Proc. *Dagstuhl Visual Motion Analysis Workshop*, LNCS 5604, pages 23–45, Springer, Berlin, 2009.