Approximate Ground Truth in the Real World for Testing Optical Flow Algorithms

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

Abstract—Vision-based driver assistance requires basic vision modules for stereo analysis or optic flow calculation. Driving situations change frequently, and methods need to be evaluated in the real world. The paper proposes ways for evaluating optic flow techniques by using estimated geometries for surrounding buildings or ‘road furniture’. It demonstrates the value of such an evaluation by discussing four different optic flow techniques.

Keywords: Ground truth, optical flow, performance evaluation, Horn-Schunck, CLG, BBPW, TV-L1

I. INTRODUCTION

Evaluations of optic flow (since the papers [1], [7] in 1994 and 1996) 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. Used 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 demonstrates ways how to evaluate optic flow in the real world and illustrates the techniques by discussing 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-L1 [14] which aims at using the L1 - rather than the L2-metric. For testing we use real-world sequences recorded on Auckland’s roads. All the used sequences contain more than 100 frames each.

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

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 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-L1 is based on total variation with respect to (basically) the L1 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 \( E_{\text{AE}}(p) \) between two flow vectors \( \mathbf{v}_0(p) = (u_0, v_0) \) and \( \mathbf{v}_1(p) = (u_1, v_1) \) 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:

\[
\mathbf{\tilde{v}} = \frac{(v_0, v_1, 1)^T}{\sqrt{(v_0^2 + v_1^2 + 1)^2}}
\]

Then we obtain that

\[
E_{\text{AE}}(p) = \arccos(\mathbf{\tilde{v}}_0^T \cdot \mathbf{\tilde{v}}_1)
\]

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 \( E_{\text{AE}}(p) \) 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 \( \mathbf{v}_0(p) = (u_0, v_0) \) is ground truth flow and \( \mathbf{v}_1(p) = (u_1, v_1) \) 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.
### III. Defined Road Geometries

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 $u = (u, v)$ is approximately

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

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 $\delta t$ 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.)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MAE</th>
<th>MEPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horn-Schunck</td>
<td>90.01</td>
<td>4.81</td>
</tr>
<tr>
<td>CLG</td>
<td>99.21</td>
<td>4.90</td>
</tr>
<tr>
<td>BBPW</td>
<td>82.43</td>
<td>1.87</td>
</tr>
<tr>
<td>TV-L$^1$</td>
<td>29.25</td>
<td>2.94</td>
</tr>
</tbody>
</table>

#### TABLE I

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

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 I 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 1 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.

#### 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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MAE</th>
<th>MEPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horn-Schunck</td>
<td>90.01</td>
<td>4.81</td>
</tr>
<tr>
<td>CLG</td>
<td>99.21</td>
<td>4.90</td>
</tr>
<tr>
<td>BBPW</td>
<td>82.43</td>
<td>1.87</td>
</tr>
<tr>
<td>TV-L$^1$</td>
<td>29.25</td>
<td>2.94</td>
</tr>
</tbody>
</table>

#### TABLE I

MAE and MEPE values of all four algorithms (after applying a residual operator) for both sequences of a driving-parallel-to-a-wall situation.
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-L₁ performs very well; see top right in Figure 2.

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 used to deal with the changes in lighting.

Figure 3 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-L₁ could generate a fairly correct optical flow field. Horn-Schunck, CLG and BBPW all failed in this case.

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-L₁ 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 4 proves that TV-L₁ results match this model. But we could also see that TV-L₁ failed if we drove too slowly.

IV. DEFINED CHANGES OF VIDEO DATA

A further option of evaluation on real-world data is given by studying changes in results when modifying the video data in a parametrized model.

1) 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.

2) 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.

3) 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 by a Gaussian (i.e., normal distribution) process, denoted by \( N(\mu, \sigma) \), where \( \mu \) is the expected value and \( \sigma \) is a varying standard deviation, to be changed from small to large. Following [12], the noise is defined as follows:

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

4) 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.
TV-L$_1$ performed best in the described tests for 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, the increase of end point errors becomes obvious. Errors also increase with the amount of noise.

On the tested 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$\degree$. 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.

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. BBPW performed obviously bad on “quite uniformly textured images”, despite of also being often reasonable on real world sequences (see Fig. 5); 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. 3.

Illumination artifacts are basically a “disaster” for all algorithms. The angular errors of TV-L$_1$ are almost 20 times higher than for an 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 (with respect to smoothing) operators are helpful as preprocessing steps 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.

**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.)

**REFERENCES**