Improving Optical Flow using Residual and Sobel Edge Images

Tobi Vaudrey[1], Andreas Wedel[2], Chia-Yen Chen[3], and Reinhard Klette[1]

[1] The *.enpeda..* Project, The University of Auckland, Auckland, New Zealand
[2] Daimler Research, Daimler AG, Stuttgart, Germany
[3] National University of Kaohsiung, Taiwan

Abstract. Optical flow is a highly researched area in low-level computer vision. It is a complex problem which tries to solve a 2D search in continuous space, while the input data is 2D discrete data. Furthermore, the latest representations of optical flow use Hue-Saturation-Value (HSV) colour circles, to effectively convey direction and magnitude of vectors. The major assumption in most optical flow applications is the intensity consistency assumption, introduced by Horn and Schunck. This constraint is often violated in practice. This paper proposes and generalises one such approach; using residual images (high-frequencies) of images, to remove the illumination differences between corresponding images.

1 Introduction

Dense optical flow was first presented by Horn and Schunck [8]. Their approach exploited the intensity consistency assumption (ICA), coupled with a smoothness constraint. This was solved in a variational approach. Many more approaches have been proposed since this, most using this basic ICA and smoothness constraint. In recent years, the use of pyramids, warping and robust minimisation equations have improved results dramatically [3]. This has further been improved and computational enhancement in [21].

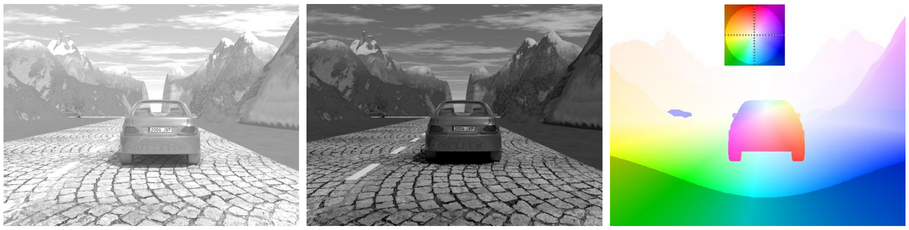

Fig. 1. Example frames from EISATS scene. Frame 1 (left) and 2 (middle) are shown with ground truth flow (right) also showing the color key (HSV circle for direction, saturation for vector length, max saturation at flow length 10).

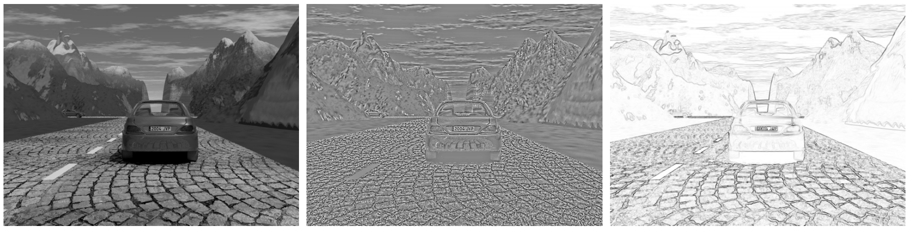

Fig. 2. Example for removing illumination artifacts due to different camera exposure in the frame 2 of EISATS set 2. Original (left) has its residual image (middle, computed using TV-L^2) and Sobel edge image (right) shown. Notice that the residual image retains more information than the Sobel image.

Previous studies have compared the results of optical flow algorithms against ground truth using various types of scenes [1, 2, 6, 11]. The earlier works in [2, 6, 11] use synthetically rendered scenes, and calculate the ground truth via ray-tracing. The more recent work of [1] calculates ground truth using structured lighting for real scenes. All of the scenes in these papers have been made publicly available. They are of good quality, but have a very limited number of frames (under 20).

None of these scenes are very difficult for the latest optical flow algorithms. The *Yosimite* scene from [2] has varying illumination in the sky, therefore most people do not use the sky for their evaluations. This means that most approaches still rely heavily on the ICA, and if this is violated the results become much worse. This was formally highlighted in [18], and then experimentally in [14]. This violation of the ICA is a major issue in real-world scenarios, such as driver assistance and security video analysis. A sample of the scene used, with ground truth, is shown in Figure 1. This demonstrates how lighting can differ dramatically between two frames.

For dealing with illumination artifacts, there are three basic approaches: simultaneously estimate the optical flow matching and model brightness change within the optical flow estimation [7], try to map both images into a uniform illumination model, or map the intensity images into images which carry the illumination-independent information (e.g., using colour images [12, 20]).

Using the first option, only reflection artifacts can be modeled without major computational expense. From experiments with various unifying mappings, the second option is basically impossible (or, at least, a very big challenge). The third approach has more merit for research; we restrain our study to using the more common grey value images.

An example of mapping intensity images into illumination-independent images is the structure-texture image decomposition [15] (an example can be seen in Figure 2). More formally, this is the concept of *residuals* [9], which is the difference between an intensity image and a smoothed version of itself. One of the first approaches, that exploited the residual images of [15], is $TV\text{-}L^1$ *improved* optical flow [19], which is an improvement to the original *TV-L*^{1} proposed in [21]. A residual is, in fact, an approximation of a high-pass filter, so only high frequencies remain present.

In this paper we generalise the residual operator by using *any* smoothing operator to calculate the low frequencies. Included in this study are three edge-preserving filters (TV-L^2 [15], median, bilateral [17]), two general filters (mean and Gaussian), and a gradient preserving filter (trilateral [4]). Furthermore, we use an edge detector as a reference (Sobel [16]). This paper shows experimentally that any residual image is better than the original image when illumination variance is causing issues.

2 Smoothing Operators and Residuals

Let f be any frame of a given image sequence, defined on a rectangular open set Ω and sampled at regular grid points within Ω.

f can be defined to have an additive decomposition $f(\mathbf{x}) = s(\mathbf{x}) + r(\mathbf{x})$, for all pixel positions $\mathbf{x} = (x, y)$, where $s = S(f)$ denotes the *smooth component* (of an image) and $r = R(f) = f - S(f)$ the *residual* (Figure 2 shows an example of the decomposition). We use the straightforward iteration scheme:

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

The concept of residual images was already introduced in [9] by using a 3×3 mean for implementing S. We apply the $m \times m$ mean operator and also an $m \times m$ median operator in this study. Furthermore, we use an $m \times m$ Gaussian filter, with σ for the normal approximation. The other operators for S are defined below.

TV-L^2 filter: [15] assumed an additive decomposition $f = s + r$ into a *smooth component* s and a *residual component* r, where s is assumed to be in $L^1(\Omega)$ with bounded TV (in brief: $s \in$ BV), and r is in $L^2(\Omega)$. This allows one to consider the minimization of the following functional:

$$\inf_{(s,r) \in BV \times L^2 \wedge f = s + r} \left(\int_{\Omega} |\nabla s| + \lambda ||r||^2_{L^2} \right) \tag{1}$$

The TV-L^2 approach in [15] was approximating this minimum numerically for identifying the "desired clean image" s and "additive noise" r. See Figure 2. The concept may be generalized as follows: any *smoothing operator* S generates a *smoothed image* $s = S(f)$ and a *residuum* $r = f - S(f)$. For example, TV-L^2 generates the smoothed image $s = S_{TV}(f)$ by solving Equ. (1).

Sigma filter: This operator [10] is effectively a trimmed mean filter; it uses an $m \times m$ window, but only calculates the mean for all pixels with values in $[a - \sigma_f, a + \sigma_f]$, where a is the central pixel value and σ_f is a threshold. We chose σ_f to be the standard deviation of f (to reduce parameters for the filter).

Bilateral filter: This edge-preserving Gaussian filter [17] is used in the spatial domain (using σ_2 as spatial σ), also considering changes in the colour domain (e.g., at object boundaries). In this case, offset vectors $\mathbf{a}$ and position-dependent real weights $d_1(\mathbf{a})$ define a local convolution, and the weights $d_1(\mathbf{a})$ are further scaled by a second weight function d_2, defined on the differences $f(\mathbf{x}+\mathbf{a}) - f(\mathbf{x})$:

$$
\begin{aligned}
s(\mathbf{x}) &= \frac{1}{k(\mathbf{x})} \int_{\Omega} f(\mathbf{x}+\mathbf{a}) \cdot d_1(\mathbf{a}) \cdot d_2 \left[f(\mathbf{x}+\mathbf{a}) - f(\mathbf{x}) \right] \, \mathrm{d}\mathbf{a} \\
k(\mathbf{x}) &= \int_{\Omega} d_1(\mathbf{a}) \cdot d_2 \left[f(\mathbf{x}+\mathbf{a}) - f(\mathbf{x}) \right] \, \mathrm{d}\mathbf{a}
\end{aligned}
\tag{2}
$$

Function $k(\mathbf{x})$ is used for normalization. In this paper, weights d_1 and d_2 are defined by Gaussian functions with standard deviations σ_1 and σ_2, respectively. The smoothed function s equals $S_{BL}(f)$. It therefore only takes into consideration values within a Gaussian kernel (σ_2 for spatial domain, f for kernel size) within the colour domain (σ_1 as colour σ).

Trilateral filter: This gradient-preserving smoothing operator [4] (i.e., it uses the local gradient plane to smooth the image) only requires the specification of one parameter σ_1, which is equivalent to the spatial kernel size. The rest of the parameters are self tuning.

It combines two bilateral filters to produce this effect. At first, a bilateral filter is applied on the derivatives of f (i.e., the gradients):

$$
\begin{aligned}
g_f(\mathbf{x}) &= \frac{1}{k_\nabla(\mathbf{x})} \int_{\Omega} \nabla f(\mathbf{x}+\mathbf{a}) \cdot d_1(\mathbf{a}) \cdot d_2 \left(||\nabla f(\mathbf{x}+\mathbf{a}) - \nabla f(\mathbf{x})|| \right) \, \mathrm{d}\mathbf{a} \\
k_\nabla(\mathbf{x}) &= \int_{\Omega} d_1(\mathbf{a}) \cdot d_2 \left(||\nabla f(\mathbf{x}+\mathbf{a}) - \nabla f(\mathbf{x})|| \right) \, \mathrm{d}\mathbf{a}
\end{aligned}
\tag{3}
$$

Simple forward differences $\nabla f(x,y) \approx (f(x+1,y) - f(x,y), f(x,y+1) - f(x,y))$ are used for the digital image. For the subsequent second bilateral filter, [4] suggested the use of the smoothed gradient $g_f(\mathbf{x})$ [instead of $\nabla f(\mathbf{x})$] for estimating an approximating plane $p_f(\mathbf{x},\mathbf{a}) = f(\mathbf{x}) + g_f(\mathbf{x}) \cdot \mathbf{a}$ Let $f_\triangle(\mathbf{x},\mathbf{a}) = f(\mathbf{x}+\mathbf{a}) - p_f(\mathbf{x},\mathbf{a})$. Furthermore, a neighbourhood function

$$
n(\mathbf{x},\mathbf{a}) = \begin{cases} 1 & \text{if } ||g_f(\mathbf{x}+\mathbf{a}) - g_f(\mathbf{x})|| < A \\ 0 & \text{otherwise} \end{cases}
\tag{4}
$$

is used for the second weighting. A specifies the adaptive region and is discussed further below. Finally,

$$
\begin{aligned}
s(\mathbf{x}) &= f(\mathbf{x}) + \frac{1}{k_\triangle(\mathbf{x})} \int_{\Omega} f_\triangle(\mathbf{x},\mathbf{a}) \cdot d_1(\mathbf{a}) \cdot d_2(f_\triangle(\mathbf{x},\mathbf{a})) \cdot n(\mathbf{x},\mathbf{a}) \, \mathrm{d}\mathbf{a} \\
k_\triangle(\mathbf{x}) &= \int_{\Omega} d_1(\mathbf{a}) \cdot d_2(f_\triangle(\mathbf{x},\mathbf{a})) \cdot n(\mathbf{x},\mathbf{a}) \, \mathrm{d}\mathbf{a}
\end{aligned}
\tag{5}
$$

The smoothed function s equals $S_{TL}(f)$. Again, d_1 and d_2 are assumed to be Gaussian functions, with standard deviations σ_1 and σ_2, respectively. The method requires specification of parameter σ_1 only, which is at first used to be the radius of circular neighbourhoods at $\mathbf{x}$ in f; let $\overline{g}_f(\mathbf{x})$ be the mean gradient of f in such a neighbourhood. Let

$$\sigma_2 = 0.15 \cdot || \max_{\mathbf{x} \in \Omega} \overline{g}_f(\mathbf{x}) - \min_{\mathbf{x} \in \Omega} \overline{g}_f(\mathbf{x}) || \tag{6}$$

(Value 0.15 was recommended in [4]). Finally, also use $A = \sigma_2$.

Numerical Implementation: All filters have been implemented in OpenCV, where possible the native function was used. For the TV-L^2, we use an implementation (with identical parameters) as in [19]. All other filters used are virtually parameterless (except a window size) and we use a window size of $m = 3$ ($\sigma_1 = 3$ for trilateral filter[4]). For the bilateral filter, we use color standard deviation $\sigma_1 = I_r/10$, where I_r is the range of the intensity values (i.e., $\sigma_1 = 0.2$ for the scaled images). The default value of $\sigma = 0.95$ is used for the Gaussian filter. All images are scaled to the range $-1 < h(\mathbf{x}) < 1$ using normalisation.

In our analysis, we also use Sobel edge images [16]; this operator provides a normalised gradient function. This is another form of illumination invariant images.

3 Optical Flow on EISATS Dataset

One of the most influential evaluations of optical flow in recent years is from Middlebury Vision Group [1]. This dataset is used to evaluate optical flow in relatively simple situations. To highlight the effect of using residual images, we used a high ranking (see [13]) optical flow technique called TV-L^1 optical flow [21]. The results for optical flow were analysed on the EISATS dataset [5]; see [18] for Set 2 (see below for details). Numerical details of implementation are given in [19]. The specific parameters used were:

Smoothness:	35	Duality threshold θ:	0.2	TV step size:	0.25
# of pyramid levels:	10	# of iterations / level:	5	# of warps / iteration:	25

The flow field is computed using $U(h_1, h_2) = \mathbf{u}$. This is to show that a residual image r provides better data for matching than for the original image f. We computed the flow using $U(r_1^{(n)}, r_2^{(n)})$ with $n = 1$, 10, 50, and 100 to show how each filter behaves. The results are compared to optical flow on the original images $U(f_1, f_2)$, and also for the Sobel-edge images. Figure 4 shows an example of this effect, obviously the residual image vastly improves optical flow results. In fact, the original image results are so noisy that they cannot be used.

EISATS Synthetic Dataset: This dataset was made public in [18] for Set 2 and is available from [5]. We are only interested in bad illumination conditions. We therefore use the altered data to resemble illumination differences in time, as performed in [14]; the differences start high between frames, then go to zero at

[4] The authours thank Prasun Choudhury (Adobe Systems, Inc.) and Jack Tumblin (EECS, Northwestern University), for their implementation of the trilateral filter.

n	TV-L^2	Sigma	Mean	Median	Bilateral	Trilateral	Gaussian	Original	Sobel
1 Ave.	7.58	7.74	7.69	7.36	6.80	6.34	7.71	55.04	1.35
ZMSD	7.59	7.76	7.71	7.38	6.83	6.36	7.72	69.41	1.84
Rank	5	8	6	4	3	2	7	9	1
10 Ave.	6.88	7.45	5.63	4.73	3.30	1.72	6.66	-	-
ZMSD	6.91	7.47	5.69	4.93	3.44	1.93	6.70	-	-
Rank	7	8	5	4	3	2	6	9	1
50 Ave.	5.17	5.59	2.83	3.85	1.47	1.72	2.83	-	-
ZMSD	5.24	5.67	3.27	4.16	1.75	1.93	3.23	-	-
Rank	7	8	5	6	1	3	4	9	2
100 Ave.	4.76	3.78	1.95	3.84	1.26	1.72	2.19	-	-
ZMSD	4.85	3.89	2.53	4.16	1.46	1.93	2.72	-	-
Rank	8	6	4	7	1	3	5	9	2

Table 1. Results of TV-L^1 optical flow on EISATS sequence. Results are shown for different numbers n of iterations. Statistics are presented for the average (Ave.), zero-mean standard deviation (ZMSD), and the rank based on ZMSD.

frame 50, then increase again. For all t (frame number) we alter the original image f using a constant brightness, using $f(\mathbf{x}) = f(\mathbf{x}) + c$. The constant brightness change is defined by: even values of t, $c = t - 52$, and odd values of t, $c = 51 - t$. An example of the data used can be seen in Figure 1, and the brightness change over the sequence can be seen in Figure 3.

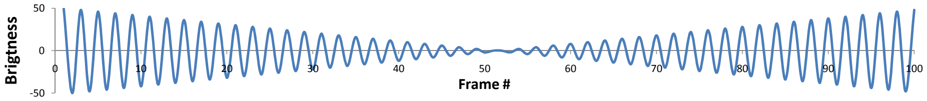

Fig. 3. Graph showing brightness change over sequence for EISATS dataset 2.

Results: To compare the results numerically, we calculated the end-point-error (EPE) as used in [1], which is basically a 2D-root mean squared error. The results can be seen in Figure 5. The zoomed out graph highlights that the results for the original image are unusable. The shape of the graph is appropriate as well, because the difference between intensities of the images gets closer together near the middle of the sequence, and further away near the end. The zoomed graph shows the EPE values between 0.5 and 7.

A major point to highlight is that at different frames in the sequence, there are different rankings for the filters. If you look, for example, at the $n = 100$ graph at frame 25, the rank is (best to worst): Sobel, trilateral, bilateral, sigma, TV-L^2, median, Gaussian, then mean. But if you look at frame 75 (roughly the same difference in illumination) the rank is (best to worst): mean, Sobel, bilateral, trilateral, median, sigma, TV-L^2, with Gaussian coming last; a completely different order! From this it should be obvious that a smaller dataset will not

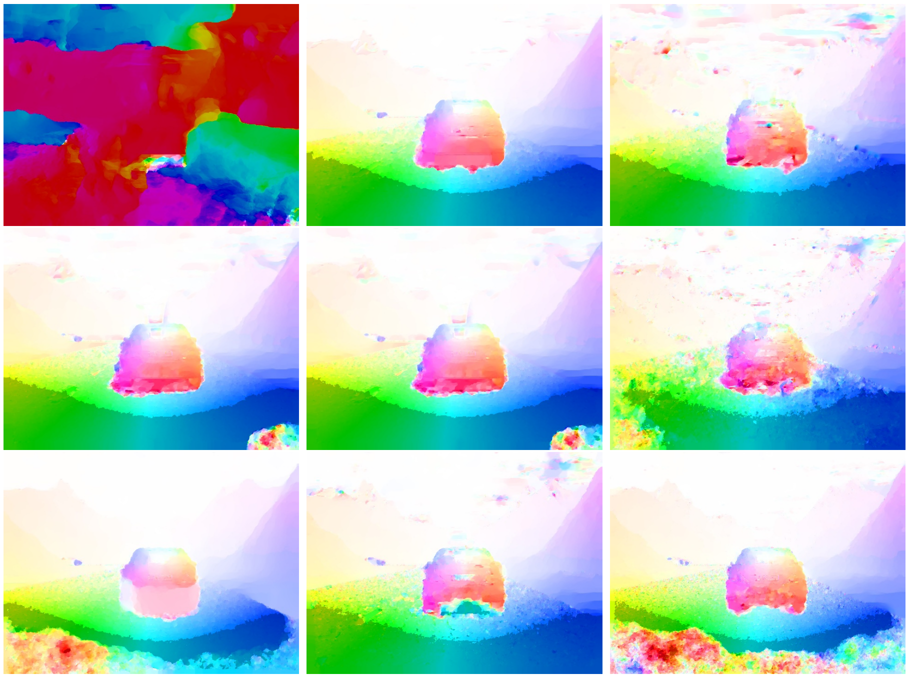

Fig. 4. Sample optical flow results on EISATS scene. Colour is encoded as in Figure 1. Top row (left to right): Using original images, Sobel edge images, and trilateral filter. Middle row (left to right): Gaussian, mean, and sigma filter. Bottom row (left to right): Median, bilateral, and TV-L^2 filter.

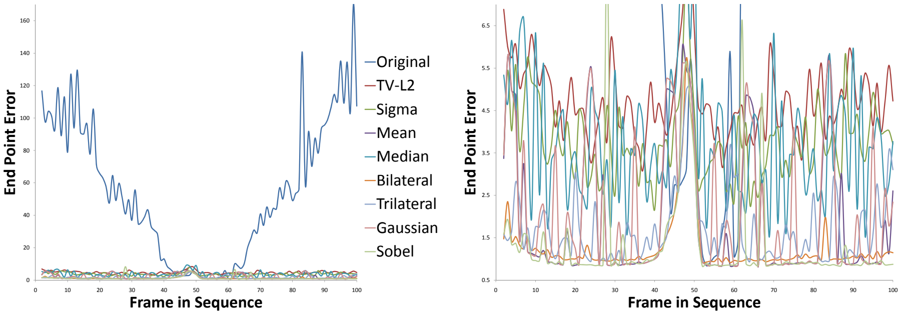

Fig. 5. End-Point-Error results over entire EISATS sequence. Filter iterations $r^{(n)}$ of $n = 100$ are shown. The left shows how different the magnitude is for the original sequence, and the right hand graph is zoomed in between 0.5 and 7.

pick up on these subtleties, so a large dataset (such as a long sequence) is a prerequisite for better understanding of the behaviour of an algorithm.

Since we have such a large dataset (99 results, 100 frames) we calculated the average and zero-mean standard deviation (ZMSD) for different iteration numbers $n = 1, 10, 50$, and 100. These results are shown in Table 1. Obviously, the original images are far worse than any residual image. From this table you can see that the order of the rankings shift around depending on the number of iterations for the residual image n. Another point to note is that the Sobel filter (which has only 1 iteration) is the best until 50 iterations; this is when bilateral filtering is the best. Simple mean filtering (which is much faster than any other filter) comes in at rank 5 after 50 iterations, and gets better around 100 iterations. It is notable that the difference between the average and ZMSD highlights how volatile the results are, the closer together the numbers, the more consistent the results.

4 Conclusions and Future Research

We have demonstrated how different residual images effect the results of optical flow. Any residual image is better than the original optical flow when using the illumination adjusted dataset. It turns out that bilateral filter may be the best, but a Sobel filter or mean residual image will work as well. So far, only simple residual images and Sobel images have been tested. Other smoothing algorithms and illumination invariant models need to be tested, such as those exploiting phase information. Finally, a larger dataset can be used to further verify the illumination artifact reducing effects of residual images.

References

1. Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M., and Szeliski, R.: A database and evaluation methodology for optical flow, in Proc. *ICCV*, pages 1–8 (2007)
2. Barron, J. L., Fleet, D. J., Beauchemin, S. S.: Performance of optical flow techniques. In *Int. J. of Computer Vision*, **12**(1): 43–77 (1994)
3. Brox, T., Bruhn, A., Papenberg, N., and Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In Proc. *ECCV*, pages 25–36 (2004)
4. Choudhury, P., and Tumblin, J.: The trilateral filter for high contrast images and meshes. In Proc. *Eurographics Symp. Rendering*, pages 1–11 (2003)
5. *.enpeda..* dataset 2 (EISATS): `http://www.mi.auckland.ac.nz/EISATS`
6. Galvin, B., McCane, B., Novins, K., Mason, D., and Mills, S.: Recovering motion fields: an evaluation of eight optical flow algorithms. In Proc. *9th British Machine Vision Conf.*, pages 195–204 (1998)
7. Haussecker, H. and Fleet, D. J.: Estimating optical flow with physical models of brightness variation. *IEEE Trans. Pattern Analysis Machine Intelligence*, **23**:661–673 (2001)
8. Horn, B. K. P., and Schunck, B. G.: Determining optical flow. *Artificial Intelligence*, **17**:185–203 (1981)

9. Kuan, D. T., Sawchuk, A. A., Strand, T. C., and Chavel, P.: Adaptive noise smoothing filter for images with signal-dependent noise. *IEEE Trans. Pattern Analysis Machine Intelligence*, **7**:165–177 (1985)
10. Lee, J.-S.: Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, **24**:255–269 (1983)
11. McCane, B., Novins, K., Crannitch, D., and Galvin, B.: On benchmarking optical flow, In *Computer Vision and Image Understanding*, **84**: 126–143 (2001)
12. Mileva, Y., Bruhn, A. and Weickert, J.: Illumination-robust variational optical flow with photometric invariants. In Proc. *Pattern Recognition - DAGM*, pages 152–162 (2007)
13. Middlebury Optical Flow Evaluation: `http://vision.middlebury.edu/flow/`
14. Morales, S., Woo, Y. W., Klette, R., and Vaudrey, T.: A study on stereo and motion data accuracy for a moving platform. In Proc. *Int. Conf. on Social Robotics (ICSR)*, to appear (2009)
15. Rudin, L., Osher, S., and Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D*, **60**:259–268 (1992)
16. Sobel, I., and Feldman,G.: A 3x3 isotropic gradient operator for image processing, in *Pattern Classification and Scene Analysis*, pages 271–272 (1973).
17. Tomasi, C., and Manduchi, R.: Bilateral filtering for gray and color images. In Proc. *IEEE Int. Conf. Computer Vision*, pages 839–846 (1998)
18. Vaudrey, T., Rabe, C., Klette, R., and Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In Proc. *IEEE Image and Vision Conf. New Zealand*, Digital Object Identifier 10.1109/IVCNZ.2008.4762133 (2008)
19. Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D.: An improved algorithm for TV-L^1 optical flow. In Post Proc. *Dagstuhl Motion Workshop*, to appear (2009)
20. van de Weijer, J. and Gevers, T.: Robust optical flow from photometric invariants. In Proc. *Int. Conf. on Image Processing*, pages 1835–1838 (2004)
21. Zach, C., Pock, T., and Bischof, H.: A duality based approach for realtime TV-L^1 optical flow, In Proc. *Pattern Recognition - DAGM*, pages 214–223 (2007)