Stereo-Vision-Support for Intelligent Vehicles - The Need for Quantified Evidence

Reinhard Klette

The *.enpeda..* Project, The University of Auckland
Auckland, New Zealand

Abstract. Vision-based driver assistance in modern cars has to perform automated real-time understanding or modeling of traffic environments based on multiple sensor inputs, using 'normal' or specialized (such as night vision) stereo cameras as default input devices. Distance measurement, lane-departure warning, traffic sign recognition, or trajectory calculation are examples of current developments in the field, contributing to the design of intelligent vehicles.
The considered application scenario is as follows: two or more cameras are installed in a vehicle (typically a car, but possibly also a boat, a wheelchair, a forklift, and so forth), and the operation of this vehicle (by a driver) is supported by analyzing in real-time video sequences recorded by those cameras. Possibly, further sensor data (e.g., GPS, radar) are also analyzed in an integrated system.
Performance evaluation is of eminent importance in car production. Crash tests follow international standards, defining exactly conditions under which a test has to take place. Camera technology became recently an integral part of modern cars. In consequence, perfectly specified and standardized tests ('camera crash tests') are needed very soon for the international car industry to identify parameters of stereo or motion analysis, or of further vision-based components.
This paper reports about current performance evaluation activities in the *.enpeda..* project at The University of Auckland. Test data are so far rectified stereo sequences (provided by Daimler A.G., Germany, in 2007), and stereo sequences recorded with a test vehicle on New Zealand's roads.

Key words: intelligent vehicle, vision-based driver support, stereo analysis, motion analysis, performance analysis, camera crash tests

1 Introduction

Current research in vision-based driver assistance asks for the generation of 'ground truth'[1] for real-world sequences, and its use for performance evaluation of various algorithms for stereo image sequence analysis.

[1] The term *ground truth* was coined in photogrammetry when comparing analysis results, derived from aerial imaging, against measured data ('on the ground'). The presence of a measurement error means that ground truth is not truth, but expected to be close to it.

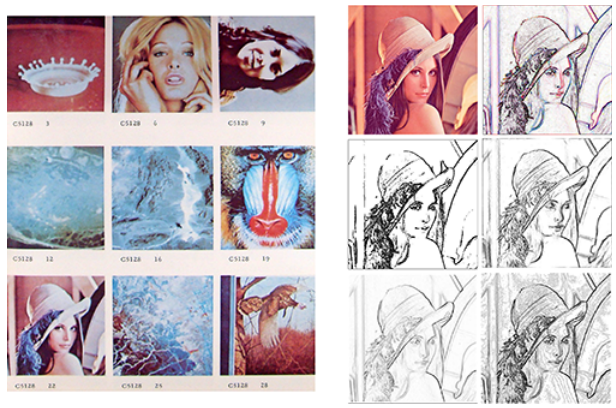

Fig. 1. Left: page in the 1976 report [21], offering eight color images and one multispectral image. Right: 'Lena' and results of various edge detectors. Those 1976 test images are still in use today when demonstrating research on low-level image processing.

Evaluations have a long history in image processing. In a first generation of test images in the 1970*s* (e.g., see [21] for images such as *Lena*, *Mandrill*, *peppers*, *tiffany*, or *zelda*; "copies of the IPI data base" were "supplied on magnetic tape, 9 track, 800 BPI, on 2400-ft. reels"; Fig. 1 shows nine of those test images), there were no stereo images, and no image sequences at all at that time in the test data base. Very short sequences of images became popular in the 1980*s*, such as those shown in Fig. 2, which allowed to compare results for optical flow. The lower left in Fig. 2 shows a calculated vector field (as obtained in a student assignment

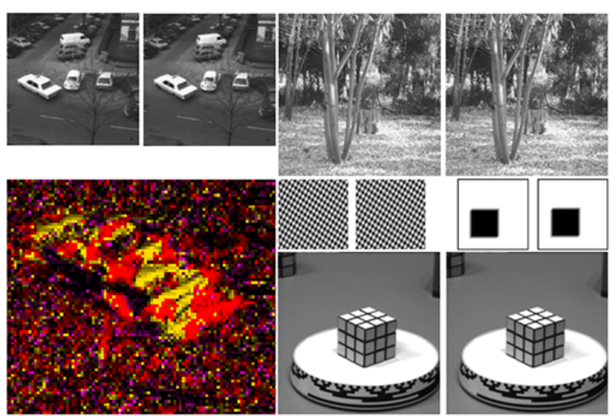

Fig. 2. Rubik cube on a microwave turntable at DEC, the 1971 "Hamburg Taxi", "SRI Trees", and two more "sequences" as used in the 1990*s*. Lower left: color representation of calculated optical flow for the taxi scene. These short sequences did not come with ground truth, and are still used sometimes today (e.g., for student assignments).

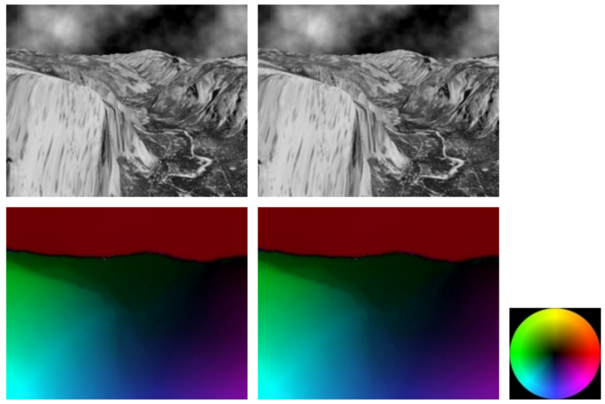

Fig. 3. Demonstration of calculated optical flow [23], using the 1984 Yosemite sequence as discussed on [2]. This sequence is still a popular way for demonstrating optical flow results.

in the 1990s in one of my classes) in common hue-intensity representation. The taxi sequence was actually recorded in 1971 (!) in the group of H.-H. Nagel [17].

The Yosemite sequence (by L. Quam [19]; see Fig. 3) "has been used extensively for experimentation and quantitative evaluation of optical flow methods, camera motion estimation, and structure from motion algorithms." [2] This is a synthetic sequence of 316×252 images, simulating a flight through a valley, with ground truth motion data (quantized to 8 bits) for rigid objects in the scene.

Test data for stereo analysis should be in standard binocular stereo geometry; [11] offered those based on using an optic bench in the lab and careful camera adjustments; see Fig. 4. There was no ground truth provided, and evaluation was based on subjective (visual) comparisons.

Automated stereo pair rectification [15] maps today stereo images into standard binocular stereo geometry [8]. This allows to generate sets of stereo images, ready for correspondence analysis. Laser-range finders may be used to generate ground truth for such stereo images by modeling real scenes [9].

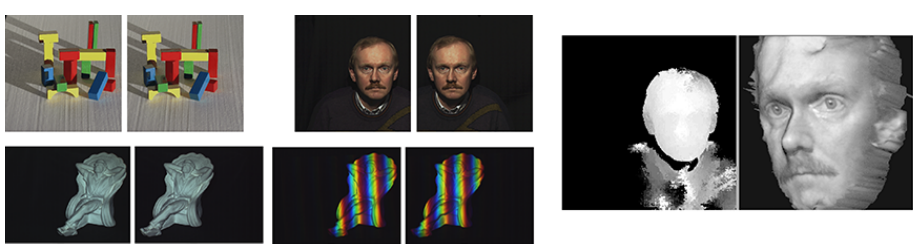

Fig. 4. Stereo pairs as used in the 1996 textbook [11] for evaluating the performance of various stereo matching algorithms. An example of a reconstructed face is shown on the right.

Fig. 5. Illustration of four stereo image sets on the Middlebury vision website: *map*, *sawtooth*, *venus*, and *Tsukuba* with depth map, illustrating ground truth as available for those data sets on this website.

The Middlebury vision page [16] (of D. Scharstein, R. Szeliski, et al.) provided in its 'pre-2008-web-server-crash' version only a few engineered samples of input images for stereo and motion analysis; see four stereo sets illustrated in Fig. 5. This website stimulates current progress in computer vision (and a web-server crash in August 2008 was followed with eagerness in the computer vision community worldwide). Currently the website is revised, now also featuring more data sets for performance evaluation, but still focussing on indoor, engineered, high contrast imagery.

Driver assistance systems (DAS), see, for example, [5], the monograph [3] of E.D. Dickmanns, or proceedings [22], have to deal with stereo image sequences

Fig. 6. An illustration for seven stereo night vision sequences available since 2007 on the *.enpeda..* website [4] for performance evaluation. Left: one out of close to 2,000 rectified stereo pairs in total. Right: screenshot of an avi showing one original sequence (lower left) and disparity data.

Fig. 7. Test vehicle HAKA1 with a pair of cameras for stereo image sequence capture, recording stereo sequences on Auckland's roads since July 2008.

recorded under any possible weather or lighting condition. See Fig. 6 for an illustration of DAS stereo sequences: seven rectified night-vision stereo sequences are available since 2007 on the *.enpeda..* website [4] for motion and stereo performance evaluation; the sequence data have been provided by Daimler AG (group of U. Franke) and prepared in 2007 by T. Vaudrey and Z. Liu for online presentation (with camera calibration and motion data for the ego-vehicle).

DAS sequences may contain unpredictable events and all kinds of variations in recorded image data, for example due to a partially 'faulty' camera, generating more blurry images in the left camera than in the right camera, or due to different brightness in left and right camera. More rectified stereo real-world sequences will be made available on the *.enpeda..* website [4] soon, including those recorded with a test vehicle (HAKA1, 'High Awareness Kinematic Automobile no. 1') in Auckland (see Fig. 7).

Obviously, it is a challenge to provide ground truth (3D environment, poses of agents) for such sequences. Three approaches appear to be possible options for satisfying the needs of *camera crash tests* as indicated in the Abstract of this paper:

(1) Post-modeling of recorded 3D environments: based on recorded stereo sequences, apply (possibly manual) 3D modeling software to generate a 3D dynamic model of the recorded scene.
(2) Accumulated evidence for 3D environments: in extension of the post-modeling approach, drive repeatedly into the same (static) 3D environment, and attempt to improve the 3D model (shape plus texture) by accumulation, merging, or unification of obtained 3D data (also using other sensors).
(3) Pre-modeling of recorded 3D environments: use 3D modeling approaches such as laser-range finders or sensor technology to generate an accurate 3D model (shape plus texture) of a defined environment and operating *agents* (vehicles or persons), and of poses of ego-vehicle and also of agents during recording.

This paper will report in the second section about work towards the first approach. For the second or third approach, see, for example, [6], where also a laser-range finder is mounted on a mobile platform, used for modeling city scenes. Laser-range finders allow very accurate 3D large-scale models, see [9]. For example, a particular area might be 3D modeled, such as a courtyard which is basically ‘static’, and this area may then serve as a ‘camera crash test site’, similar to crash test halls at car companies. For combining various sensors for 3D modeling, see, for example, [12]. Alternatively, large scale modeling may also

Fig. 8. Examples of manually specified rectangular regions for approximated ground truth: in original sequences of Set 1 on [4] (left) and in Sobel-based BP results (right).

utilize technology as developed for the generation of 3D maps [1], also discussed in [9].

2 Approximate Ground Truth

In a recorded stereo sequence, we may identify simple geometric shapes and identify their 3D location, using automated or manual measurements; see Fig. 8. (The figure also shows identified rectangular areas in depth maps calculated using belief propagation as specified in [7].) As a more general option [14], we may assume an approximate planar road surface, using known parameters of ego-vehicle and cameras (as saved for Set 1 on [4] in the *camera.dat* file and in the file header of every frame; see [13]).

2.1 Disparities on Road Surface

We assume that test sequences are ego-motion compensated, which means that the horizon is always parallel with the row direction in the images, and pixels on the same image row have the same depth value if a projection of the planar road surface.

A side-view of the camera setting is shown in Figure 9, where θ is the tilt angle, P is a road surface point which is projected into $p = (x_p, y_p)$ on the image plane, H is the height of the camera. It follows that

$$Z = d_e(OP_c) = d_e(OP)\cos\psi = \frac{H}{\sin(\theta+\psi)}\cos\psi \tag{1}$$

According to the stereo projection equations, the disparity d can be written as

$$d = \frac{b \cdot f}{Z} = \frac{b \cdot f}{\frac{H}{\sin(\theta+\psi)}\cos\psi} \tag{2}$$

where angle ψ can be calculated as follows, using focal length f and pixel coordinate y_p in the image:

$$\psi = \arctan\big(\frac{(y_p - y_0)s_y}{f}\big) \tag{3}$$

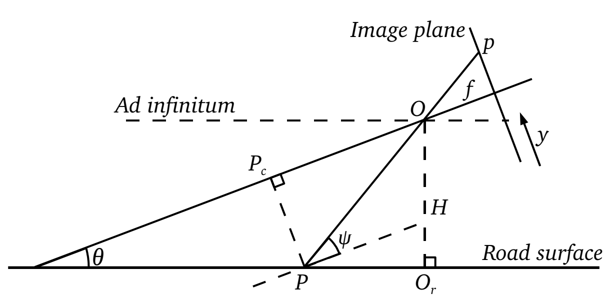

Fig. 9. Projection of a point P of the road surface.

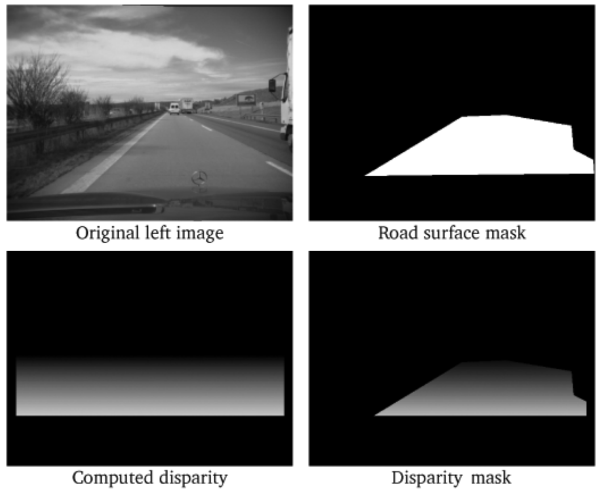

Fig. 10. Generation of a disparity mask: input image, generated road mask, depth map of a planar road, and resulting disparity mask.

Here, y_0 is the y-coordinate of the principal point, and s_y is the pixel size in y-direction. We can also compute the y-coordinate of a line that projects to infinity

$$y_{inf} = \frac{y_0 - f \cdot \tan\theta}{s_y}$$

This is the upper limit of the road surface, and points on it should have zero disparity (if no objects block the view).

Figure 10 illustrates the process of generating an approximated disparity map on road surface areas, also using manual input for a conservative outline of the road area in a given image. In the given camera setting (of the seven sequences), there is a yaw angle (0.01 radian) which makes the cameras looking a little bit to the left. This angle can be ignored because it only defines the right camera to be about 3 mm behind the left camera.

2.2 Recalibration of Tilt Angle

Although a camera tilt angle is already given for these sequences, we noticed that the angle is not always true when verifying the data. This problem might be caused by several reasons, for example, the road surface is changing (downhill, uphill), the car coordinate system is not parallel to the road surface in some situations (acceleration, braking), drivers of different weight, or driving with flat tires, or the installation of cameras may change for some reasons. (Actually, changes are easy to detect by reading the position of the Mercedes star in the given images.)

The outlined process for obtaining approximate stereo ground truth identified the importance of the tilt angle for the estimated values. We propose a method

Table 1. Results of tilt angle estimation for the given seven sequences.

Sequence name	Tilt angle (radian)
1: 2007-03-06_121807	0.01608
2: 2007-03-07_144703	0.01312
3: 2007-03-15_182043	0.02050
4: 2007-04-20_083101	0.06126
5: 2007-04-27_145842	0.06223
6: 2007-04-27_155554	0.06944
7: 2007-05-08_132636	0.05961

to estimate the average tilt angle for a given sequence of frames. This method is similar to the road surface stereo approximation, just in a reverse order. We estimate the tilt angle based on given depth at some feature points (i.e., with known disparities) which can be measured or identified manually.

See Figure 10 and assume a given pair of corresponding points, with disparity d. By Equation (2) we have that the tilt angle can be written as follows:

$$\theta = \arcsin\left(\frac{H \cos\psi \cdot d}{b \cdot f}\right) - \psi \tag{4}$$

where ψ is as given in Equation (3).

Altogether, at first, we randomly select five or six frames from a sequence of frames, then, we calculate or choose pairs of corresponding pixels on the road surface area, and obtain disparities between those. Each disparity (of one pixel pair) can be used to calculate a tilt angle using Equation (4), and a mean of those provides a tilt angle estimation; see Table 1 for results for the seven sequences.

2.3 2D Motion on Road Surface

Speed and direction (yaw rate) of the ego-vehicle are given for all frames of those seven sequences. The road is, obviously, static, what makes the calculation of relative movement of road surface points (with respect to the camera) straight forward.

Given a pixel p on the image plane at time t, which is projected to a road surface point P. Let P move to a new position P' at time $t + \delta t$, where δt is the time interval between two consecutive frames (called `CycleTime` in the seven sequences, either equals 0.04 s or 0.08 s). Then, P' is projected back to the image plane at p'; see Figure 11. The approximation of 2D motion (i.e., local displacement) at a pixel can then proceed as follows:

First, assume that the vehicle speed equals $\mathbf{v}$ at time t, and $\mathbf{v}'$ at time $t + \delta t$; the average speed during this time interval equals δt is $\overline{\mathbf{v}} = \frac{\mathbf{v}+\mathbf{v}'}{2}$, having δt very small in the sequences. Distances (in Z_{road} coordinates) of moving points are defined as follows:

$$d_Z(P, P') = |\overline{\mathbf{v}}| \cos(\overline{\varphi} + \varphi_c)\delta t = \frac{|\mathbf{v}_1| + |\mathbf{v}_2|}{2} \cos\left(\frac{\varphi_1 + \varphi_2}{2} + \varphi_c\right)\delta t$$

where φ_1 and φ_2 are the yaw angles of the ego-vehicle at t and $t + 1$, and φ_c is the yaw angle of the camera installation (see Figure 12). Therefore, the distance

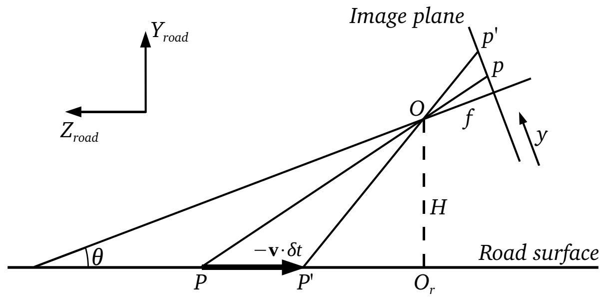

Fig. 11. Approximation of 2D motion in y-direction: P and P' is the same road surface point, just in two consecutive frames. P is projected into $p = (x, y)$ in the image plane, P' is projected into $p' = (x', y')$.

between the point P and the ego-vehicle becomes

$$Z_{P'} = d_Z(O_r, P') = d_Z(O_r, P) - d_Z(PP') = \frac{H}{\tan(\theta + \psi)} - d_Z(PP')$$

Then, the angle between the projection ray OP' and the optical axis of the camera may be determined as follows:

$$\psi' = \arctan\big(\frac{H}{d_Z(O_r, P')}\big) - \theta = \arctan\big(\frac{H}{d_Z(O_r, P) - d_Z(P, P')}\big) - \theta$$

where $d_Z(O_r, P) = \frac{H}{\tan(\theta+\psi)}$.

Therefore, according to Equation (3), the y-coordinate of 2D motion $\mathbf{u}$ at point P' can be written as

$$v = \big(\frac{f \cdot \tan(\psi')}{s_y} + y_0\big) - y_p$$

Thus, we are also able to specify the position of point P in x-direction as follows

$$X_P = \frac{Z_P \cdot x_p}{f}$$

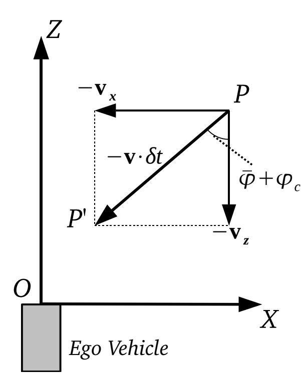

Fig. 12. Change in relative position between road surface point P and ego-vehicle.

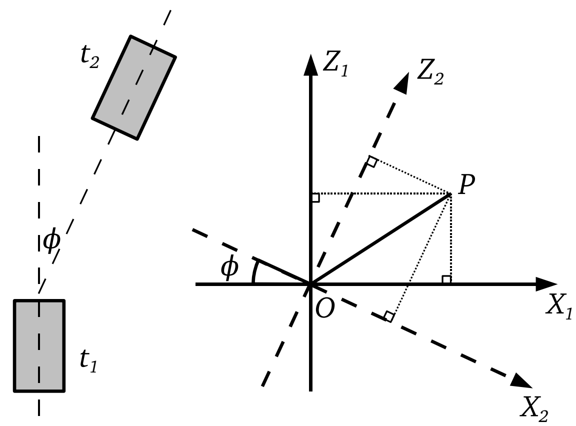

Fig. 13. A rotation of the ego-vehicle.

with $Z_P = \frac{H}{\sin(\theta+\psi)} \cos\psi$, which is actually already a known value from the previous stereo ground truth approximation.

The position of P' (for the next frame) can then be calculated by using speed $\mathbf{v}$ and time interval δt,

$$X_{P'} = X_P - |\mathbf{v}| \sin(\overline{\varphi} + \varphi_c)\delta t$$

Now we have the new relative position between the road surface point and the vehicle at time $t + \delta t$. - In a next step, we need to rotate the vehicle coordinate system by an angle according to the yaw rate given in the vehicle movement parameters; see Figure 13. Therefore, the final (relative) position equals

$$\begin{bmatrix} X^{\phi}_{P'} \\ Z^{\phi}_{P'} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} X_{P'} \\ Z_{P'} \end{bmatrix}$$

In a final step, point P is projected back to a pixel p' on the camera's image plane. Then, 2D motion is obtained by comparing locations of p and p', as follows:

$$\psi' = \arctan\big(\frac{H}{Z^{\phi}_{P'}}\big) - \theta$$

$$v = y'_p - y_p = \big(\frac{f \cdot \tan(\psi')}{s_y} + y_0\big) - y_p$$

$$u = x_{p'} - x_p = \frac{f \cdot X^{\phi}_{P'}}{\frac{H}{\sin(\theta+\psi')}\cos(\psi')} - x_p$$

2.4 Change in Depth for Image Features

As another option for modeling recorded scenes, we may use a scale-space based estimation of changes in depth [20]. Consider a disk of radius ρ moving towards

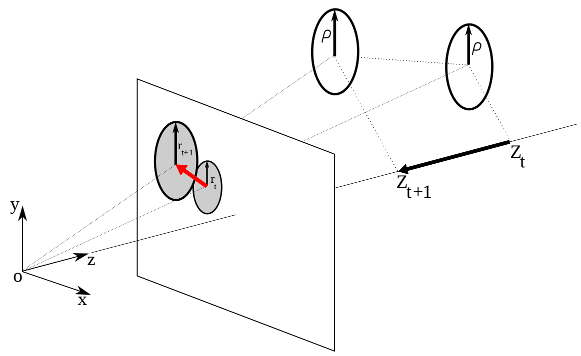

Fig. 14. Two projections of a moving disk, at times t and $t+1$.

an ideal pinhole-type camera of focal length f. Without loss of generality, let the radius move parallel to the Y-axis of the XYZ-camera coordinate system (i.e., $r = Y_c - Y_e$, for center P_c and an edge point P_e of the disk). A 3D point $P = (X, Y, Z)$ in the world (in camera coordinates) projects into a point $p = (x, y, f)$ in the image plane, with $x = f\frac{X}{Z}$ and $y = f\frac{Y}{Z}$. Point P_c projects into $p_c = (x_c, y_c, f)$, and P_e projects into $p_e = (x_e, y_e, f)$. The moving disk is at time t at distance Z_t, and projected into image I_t as a disk of radius r_t (see Fig. 14). We obtain the following for the area of this projected disk:

$$A_t = \pi r_t^2 = \pi\left(y_c - y_e\right)^2 = f\frac{\pi}{Z_t^2}\left(Y_c - Y_e\right)^2 = \pi f\frac{\rho^2}{Z^2}$$

Radius ρ of the disk is constant over time, thus, the product $A_t Z_t^2 \sim \rho^2$ will also not change over time.

We consider projections of the disk at times t and $t+1$. Because the ratio of square roots of areas is proportional to the inverse of the ratio of corresponding Z-coordinates of the disk, we are able to define a *z-ratio*

$$\mu_z = \frac{\sqrt{A_t}}{\sqrt{A_{t+1}}} = \frac{Z_{t+1}}{Z_t} \tag{5}$$

either by area or Z-values.

Such a z-ratio can also be defined just for a pair of projected points $P_t = (X_t, Y_t, X_t)$ and $P_{t+1} = (X_{t+1}, Y_{t+1}, Z_{t+1})$ (just by the ratio of Z-coordinates). Using the central projection equations for both projected points, we obtain for their *x-ratio* and *y-ratio* the following:

$$\mu_x = \frac{X_{t+1}}{X_t} = \frac{Z_{t+1}}{Z_t} \cdot \frac{x_{t+1}}{x_t} = \mu_z \frac{x_{t+1}}{x_t} \tag{6}$$

$$\mu_y = \frac{Y_{t+1}}{Y_t} = \frac{Z_{t+1}}{Z_t} \cdot \frac{y_{t+1}}{y_t} = \mu_z \frac{y_{t+1}}{y_t} \tag{7}$$

Altogether, this may also be expressed by the following *update equation*:

$$\begin{pmatrix} X_{t+1} \\ Y_{t+1} \\ Z_{t+1} \end{pmatrix} = \begin{pmatrix} \mu_x & 0 & 0 \\ 0 & \mu_y & 0 \\ 0 & 0 & \mu_z \end{pmatrix} \begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} \tag{8}$$

Fig. 15. Disks with radii defined by maxima of scale space characteristics.

with μ_x, μ_y, and μ_z as in Equations (6), (7), and (5) respectively. In other words, knowing μ_z and ratios $\frac{x_{t+1}}{x_t}$ and $\frac{y_{t+1}}{y_t}$ allows to update the position of point P_t into P_{t+1}. Assuming that P_t and P_{t+1} are positions of one tracked 3D point P, from time t to time $t+1$, we only have to solve two tasks: (1) decide for a technique to track points from t to $t+1$, and (2) estimate μ_z. If an initial position P_0 of a tracked point P is known then we may identify its 3D position at subsequent time slots. Without having an initial position, we only have a 3D direction P_t to P_{t+1}, but not its 3D position.

For identifying μ_z, an 'area of influence' is assigned to each tracked feature point, basically taking the role of a tracked disk.

For tracked points, a scale-space-based measure is computed for the 'extension of the local image structure' in a local (or semi-local) neighborhood. Such measures, computed independently for each pair of points, are used to determine a scale ratio (based on associated intensity profiles of scale characteristics of those feature points), which is finally used as an estimate of the z-ratio μ_z. For details, see [20]. Figure 15 illustrates disks assigned to tracked features.

3 Evaluation

We use quality metrics to measure the quality of calculated stereo correspondences or motion vectors with respect to approximated ground truth.

3.1 Stereo

The general approach of stereo evaluation is to compute error statistics based on given ground truth. We use the same error measurements as on [16], namely the *root mean squared error* between the disparity map $d(x,y)$ and the ground truth map $d_T(x,y)$, defined as follows:

$$E_R = \left(\frac{1}{n}\sum |d(x,y) - d_T(x,y)|^2\right)^{\frac{1}{2}} \tag{9}$$

where n is the total number of pixels, and the percentage of *bad matching pixels*, defined as follows:

$$E_B = \frac{1}{n} \sum (|d(x,y) - d_T(x,y)| > \delta_d) \tag{10}$$

where δ_d is the threshold of disparity tolerance.

Quality metrics for optical flow evaluation have to measure the result in a 2D space. We use the common *angular error* defined as the average angle between estimated optical flow vector $\mathbf{u}$ and the true flow vector $\mathbf{u}_T$,

$$E_{AE} = \frac{1}{n} \sum \arccos\left(\frac{\mathbf{u} \cdot \mathbf{u}_T}{|\mathbf{u}||\mathbf{u}_T|}\right) \tag{11}$$

where $|\mathbf{u}|$ denotes the length (magnitude) of a vector, and the *end point error* which measures the absolute distance between the end points of vectors $\mathbf{u}$ and $\mathbf{u}_T$,

$$E_{EP} = \sqrt{(u - u_T)^2 + (v - v_T)^2} \tag{12}$$

3.2 Examples of Results

The discussed approximate ground truth has been used in [7, 14] for evaluating stereo and motion analysis techniques, such as variants of dynamic programming (including Birchfield-Tomasi), belief propagation, semi-global matching, or variants of optical flow calculation (using sources in OpenCV [18] where available, D. Huttenlocher's belief propagation sources from [10], or our own implementation).

For example, Fig. 16 shows bad matches for Sequence 1 (of Set 1 on [4]), comparing a common dynamic programming approach with modifications, also using spatial or temporal propagation (only one of those, or both combined). The figure shows values for all the 300 stereo pairs of this sequence. It clearly indicates that temporal propagation (see DPt in the diagram) is of benefit if evaluating within the described road mask of estimated disparities.

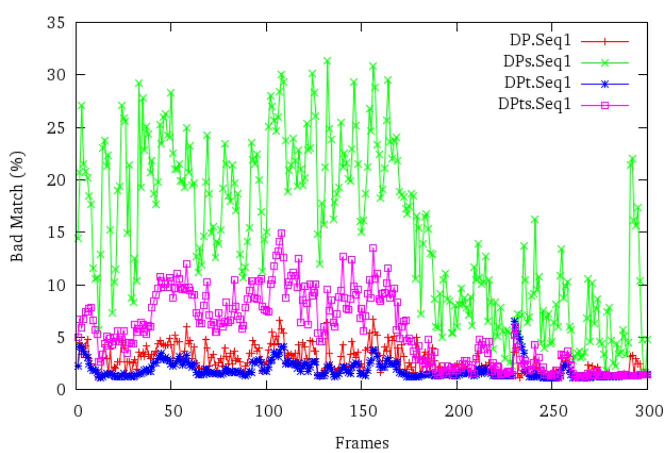

Fig. 16. Percentages of bad matches for dynamic programming stereo and its variants.

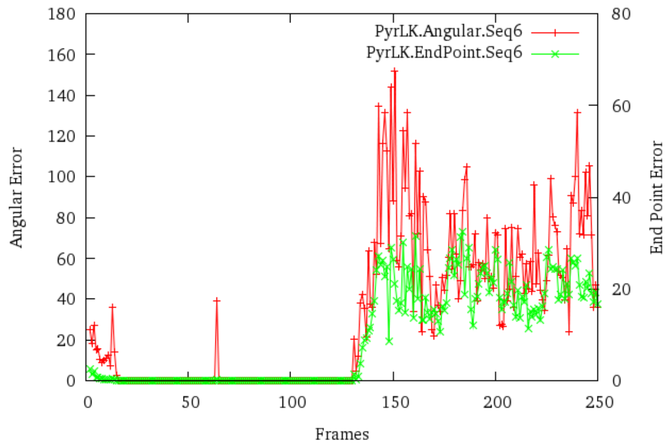

Fig. 17. Angular errors and endpoint errors for PyrLK on Sequence 6.

Figure 17 summarizes angular and end point errors of the pyramid Lucas-Kanade technique for all 250 frames of the left camera of Sequence 6.

We will not start a comparative discussion here, and point the reader to [7, 14]. The two examples of diagrams are given here to illustrate an important property of these evaluations based on real-world sequences: here we have long sequences, basically of arbitrary length, and we may use this for improving results (e.g., by applying a Kalman filter), but also for deriving statistically more relevant performance evaluation results.

4 Conclusions

Vision-based driver assistance systems have moved into modern cars in recent years, and there will be an 'exponential growth' in demands not only with respect to deriving accurate and real-time computer vision solutions, but also in evaluating these solutions, to ensure that they satisfy international standards (still to be defined by major car manufacturers).

This will require that testing is based on real-world data, without eliminating any possible visual effect, and with aiming at 'robust' testing. A vision system may be 'robust' if being fairly invariant with respect to changes in brightness or contrast; obviously, a smoke detection system should not have this type of 'robustness'. We conclude that 'robustness' needs to be defined for the particular needs of DAS.

Evaluation not only needs to be done *also* on stereo real-world sequences; we may expect that the car industry will define the state of the art in stereo and motion analysis with their (expected) quality standards very soon. Image analysis will also work on rainy days, even in the night, and so forth.

Acknowledgement: The author acknowledges valuable support of, or collaboration with (in alphabetic order) Je Ahn, Ali Al-Sarraf, Eduardo Destefanis, Shushi Guan, Zhifeng Liu, Jorge Sánchez, and Tobi Vaudrey.

References

1. 3D Reality Maps^{TM}, `http://www.realitymaps.de/`
2. Black, M.: Comments about the Yosemite sequence. `http://www.cs.brown.edu/~black/Sequences/yosFAQ.html`
3. Dickmanns, E.D.: *Dynamic Vision for Perception and Control of Motion.* Springer, London (2007)
4. *.enpeda..* Image sequence analysis test site, `http://www.citr.auckland.ac.nz/6D/`
5. Franke, U., Gavrila, D., Gorzig, S., Lindner, F., Paetzold, F., Wöhler, C.: Autonomous driving goes downtown. *IEEE Int. Systems*, 13:40–48 (1998)
6. Früh, C., Zakhor, A.: An automated method for large-scale, ground-based city model acquisition. *Int. J. Computer Vision* 60:5–24 (2004)
7. Guan, S., Klette, R.: Belief propagation for stereo analysis of night-vision sequences. Technical report, Computer Science Department, The University of Auckland (2008)
8. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK (2000)
9. Huang, F., Klette, R., Scheibe, K.: *Panoramic Imaging - Rotating Sensor-Line Cameras and Laser Range-Finders*, Wiley, Chichester (2008)
10. Huttenlocher, D.: Loopy belief propagation sources, `http://people.cs.uchicago.edu/~pff/bp/`
11. Klette, R., Koschan, A., Schlüns, K.: *Computer Vision.* Vieweg, Braunschweig (1996)
12. Klette, R., Reulke, R.: Modeling 3D scenes: paradigm shifts in photogrammetry, remote sensing and computer vision. Opening Keynote, In Proc. *Int. IEEE Conf. ICSS* (on CD, 8 pages), Taiwan (2005) (see also `http://www.citr.auckland.ac.nz/techreports/show.php?id=155`)
13. Liu, Z., Klette, R.: Performance evaluation of stereo and motion analysis on rectified image sequences. Technical report, Computer Science Department, The University of Auckland (2007)
14. Liu, Z., Klette, R.: Approximated ground truth for stereo and motion analysis on real-world sequences. Technical report, Computer Science Department, The University of Auckland (2008)
15. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections, Nature, volume 293, pages 133–135 (1981)
16. Middlebury vision website, `http://vision.middlebury.edu/`
17. Nagel, H.-H.: Image sequence evaluation: 30 years and still going strong. In Proc. *ICPR*, volume 1, pages 149–158 (2000)
18. Open Source Computer Vision Library, `http://www.intel.com/research/mrl/research/opencv/`
19. Quam, L.: Hierarchical warp stereo. In Proc. *DARPA Image Understanding Workshop*, pages 149–155 (1984)
20. Sánchez, J., Klette, R., Destefanis, E.: Estimating 3D flow for driver assistance applications. Technical report, Computer Science Department, The University of Auckland (2008)
21. Schmidt, R.: The USC - Image Processing Institute data base, revision 1. USCIPI Report 780, October (1976)
22. Sommer, G., Klette, R. (eds.): Robot Vision 2008. LNCS 4931, Springer, Berlin (2008)
23. Weickert, J., et al.: Online presentation of MIA Group, `http://www.mia.uni-saarland.de/OpticFlow.shtml`.