Performance Evaluation of Stereo and Motion Analysis on Rectified Image Sequences

Zhifeng Liu and Reinhard Klette

Computer Science Department, CITR, The University of Auckland
Private Bag 92019, Auckland 1142, New Zealand

Abstract. This paper introduces into seven real-world road driving stereo sequences (provided by Daimler AG, Germany; now freely available for academic research); it also informs about their use for performance evaluation in some experiments using common stereo and motion analysis algorithms. Often, such algorithms are tested on a few frames only, or on synthetic sequences, but not on long real-world sequences. The provided sequences have 250 to 300 stereo pairs each; they have been used at Daimler AG for testing 6D vision, which fuses disparity and motion data (by using a Kalman filter). In this paper we introduce into those seven sequences (and inform about the download web site); we discuss a few approaches how to use those sequences for testing either stereo or motion algorithms, or both combined together. Certainly, those sequences do have many more potentials for future performance evaluations for stereo and motion analysis algorithms.

1 Introduction

Stereo and motion analysis play a central role in computer vision. Many algorithms in this field have been proposed and carefully studied; see, for example, [2, 3, 16] and the website *vision.middlebury.edu*. However, performance evaluation of these algorithms is typically done on either short real-world, or slightly longer but synthetic image sequences, instead of sequences modeling real-world applications by showing diverse situations at some representative length. Performance evaluation is an important subject in computer vision [12].

After years of research on ego-motion estimation [1], the automobile industry has all the tools for producing rectified stereo image sequences. In Section 2, we will introduce seven night-vision, gray-level image sequences, provided recently by Daimler AG Germany. We briefly highlight main features for each of those sequences, which define goals when analyzing those sequences. For downloads, see *citr.auckland.ac.nz/6D*.

This short article only allows a few comments on the intended or already performed use of those sequences for testing stereo and motion algorithms. (See also [7].) Others may find those sequences also of interest.

The paper is structured as follows. Section 2 specifies the seven sequences. Section 3 discusses challenges in general terms, and informs about a few experiments, showing stereo and motion analysis results, and results for combined (i.e., 6D) analysis for these sequences. We also inform about methods how to visualize results in some efficient ways. Section 4 informs on evaluation strategies defined on contents in these sequences.

In this paper, the vehicle which is used to host the stereo camera platform for capture image sequences is called *the host car*.

2 Real-world sequences

Daimler AG Germany [6] provided recently to the authors seven stereo sequences for research purposes. They are captured with a calibrated pair of night-vision cameras near Stuttgart. Each sequence contains 250 or 300 frames, and features different driving environments, including highway, urban road and rural area. The ego-motion of the stereo platform has been correctly estimated and compensated [1], so that the view angle is always (about) parallel to the horizon. Furthermore, camera calibration is used for geometric rectification, such that image pairs are characterized by "standard epipolar geometry" [11]. A few "black strips" around borders of images are caused by this compensation or rectification. Figure 1 shows an example of one stereo pair in such a sequence.

Fig. 1. Sequence 2007-03-06_121807, called *Sequence 1* or *Construction-Site Sequence*.

The resolution of images in these sequences is 640×481. They are saved in PGM[1] grayscale format, and available in both big endian and little endian, for convenient usage in different software.

The stereo night vision camera parameters are provided in file *camera.dat*. For each sequence, calibration parameters for left and right camera are also provided

[1] Portable graymap file format.

in *calib_k0.bog* and *calib_k1.bog* under *systemData* directory. The vertical and horizontal size of each pixel of the camera sensor, focal length, and coordinates of principle point (i.e., intersection of optic axis with image plane) are all given in *pixel*. The horizontal size of a sensor cell defines this unit.

The vehicle's movement status is given for each frame. The information can be extracted from the comment in the corresponding PGM file header as shown below:

```
P5
# bigEndian
#[Units are rads, metres and seconds]
#[Inertial Sensor]
#YawRate= 0.004398
#YawAngle= 0.000000
#Speed= 8.329330
#[Other Image Data]
#TimeStamp: 1802550.000000
#CycleTime: 0.080000
#ImageNumber: 111
#[Wheel Data]
#WheelRPM_FL: 240.500000
#WheelRPM_FR: rm242.000000
#WheelRPM_RL: 235.500000
#WheelRPM_RR: 237.000000
#
640 481
3585
```

The `TimeStamp` might be wrong (e.g., identical time stamps in several subsequent frames). The `CycleTime` in the header is always correct (which denotes the interval between two consecutive pairs of frames). However, it may be either 0.04, or 0.08 (as in the example above; in this case there was a frame skip). For more detailed explanations, see download website *citr.auckland.ac.nz/6D*.

Here is a brief introduction for each sequence, including driving environment, main objects, or special features.

1. Sequence 2007-03-06_121807 (Figure 1) features normal traffic density on an Autobahn, with a standard safety fence between opposite traffic directions. However, normal lanes have been cut in width and shifted to the right, with several slow large trucks in the right lane.
2. Sequence 2007-03-07_144703 (Figure 2) features medium traffic in an urban area. The host car goes straight and turns then to the left. There are a few pedestrians walking on the footpath, some cars in parking lots, or waiting to join the traffic.
3. Sequence 2007-03-15_182043 (Figure 3) features a rural road with only incoming traffic. Starting from the 156th frame, a squirrel appears in the scene and runs across the road, in front of the host car. The scene appears dark

Fig. 2. Sequence 2007-03-07_144703, called *Sequence 2* or *Save-Turn Sequence*.

Fig. 3. Sequence 2007-03-15_182043, called *Sequence 3* or *Squirrel Sequence*.

Fig. 4. Sequence 2007-04-20_083101, called *Sequence 4* or *Dancing-Light Sequence*.

and wet. Vehicles in a distance can only be distinguished by head lights, with reflection on vehicles and road surface.

4. Sequence 2007-04-20_083101 (Figure 4) shows a one-way road (two-lane highway) in mountainous area. The host car follows a small car and passes a few larger trucks. There are many shadows of trees on road and vehicles. Illu-

Fig. 5. Sequence 2007-04-27_145842, called *Sequence 5* or *Intern-on-Bike Sequence*.

Fig. 6. Sequence 2007-04-27_155554, called *Sequence 6* or *Traffic-Light Sequence*.

Fig. 7. Sequence 2007-05-08_132636, called *Sequence 7* or *Crazy-Turn Sequence*.

mination changes significantly several times. The lighting between left and right images also varies.

5. Sequence 2007-04-27_145842 (Figure 5) features a straight country road with both incoming and outgoing vehicle traffic. A cyclist drives toward the main

road from a perpendicular direction and turns left at the intersection. This simulates a dangerous situation between the cyclist and the vehicle.

6. The host car stops during the first half of Sequence 2007-04-27_155554 (Figure 6) in front of a traffic light. It then goes into the opposite lane because of a construction site which blocks the other lane. At the end of the road construction area, another vehicle and a cyclist appear in the scene. The sequence features a road in a forest, with fine texture caused by trees.
7. In the first 30 frames of Sequence 2007-05-08_132636 (Figure 7), the host car is waiting on the roadside. Later, it goes into the central lane and turns left at the intersection while an oncoming vehicle is driving towards it. The sequence features a very dangerous situation where both vehicles almost collided.

These sequences, selected by Daimler AG, Germany, provide a multi-facetted challenge for stereo and motion analysis algorithms.

3 Approaches and Examples

Filtering or edge detection are important processes, to be considered for these sequences. However, here we only sketch briefly the three "core approaches".

3.1 Stereo Analysis

Typically, we are interested in dense (approximate) depth maps rather than in sparse depth data. Difficulties in finding corresponding points in pairs of stereo

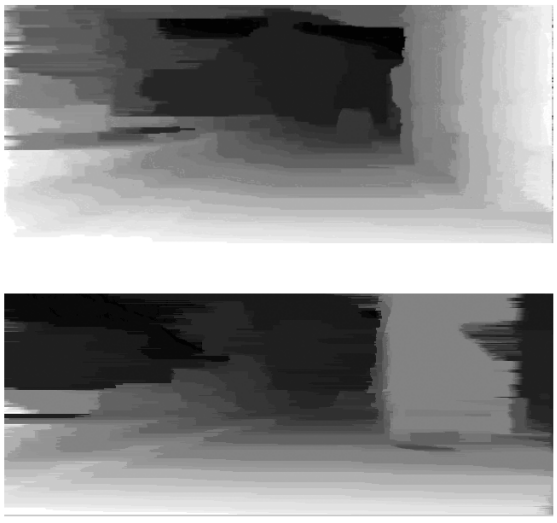

Fig. 8. Two dense depth maps calculated by applying dynamic programming on the Dancing-Light Sequence, using a double-propagation strategy. Courtesy by Darren Troy, Auckland.

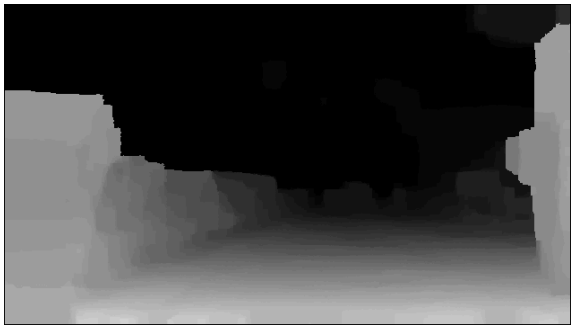

Fig. 9. Edge detection followed by belief-propagation allows to calculate this dense depth map for the Construction Site Sequence; see [7].

images do have many reasons (see, for example, [11], also for some of the earlier algorithms which have been designed for stereo analysis; for recent techniques, see [16] and the website *vision.middlebury.edu*).

We illustrate two examples of stereo analysis approaches. [5] proposed a special dynamic programming approach, basically for a pair of stereo images, in which the disparity matrix of a line is used as additional input for the calculation of the disparity map of the subsequent image line. This can now be generalized when having sequences of stereo pairs: additionally, the disparity matrix of the same line, but for the previous stereo pair, is also used. See Figure 8 for a result using this approach of propagating results within the same pair of images, and also along the time scale. This allowed for a substantial improvement.

Figure 9 illustrates another extension of a well-known approach due to the specific properties of the given image sequences. Here, belief-propagation was not performed on the given image pairs but on Sobel edge images of those. See [7] for details.

Instead of using gray-level representations of depth maps (as in these two examples), it also proved very informative (e.g., at Daimler AG, Germany) to use a color scheme, such as green for further away, and red for very close. This color scheme also reflects common color interpretation where red means danger and green stands for peace.

3.2 Motion Analysis

Motion estimation for these sequences should provide information about movements of objects (speed, trajectory) as relevant for each sequence, often for identifying possible courses of conflict. Motion analysis starts in computer vision typically with optic flow calculation [2, 9, 13], assuming that this leads to approximate calculations of the local displacement (of corresponding points between two continuous image frames). However, a few experiments with those sequences will reveal immediately the difficulty in applying optic flow algorithms successfully to those sequences, which are often blurry or fine textured (e.g., in trees).

Two typical basic assumptions behind optic flow algorithms are as follows: the brightness of the scene should be about constant, and local displacements must be small. However, both are not satisfied in those sequences. For example, the illumination changes often in one sequences, due to shading patterns of trees, or there is even different lighting for left and right camera. Object segments also move often very fast within images.

Again, sequences allow stabilization of results along the time scale (see, for example, [10]), and larger motion vectors can also be analyzed by using hierarchical approaches (see, for example, [14]). It is also recommended to apply relatively "advanced" edge detection first (such as a Canny edge detector, and not just the Sobel operator), and then analyze motion vectors only along those edges. See Figure 10 for an illustration.

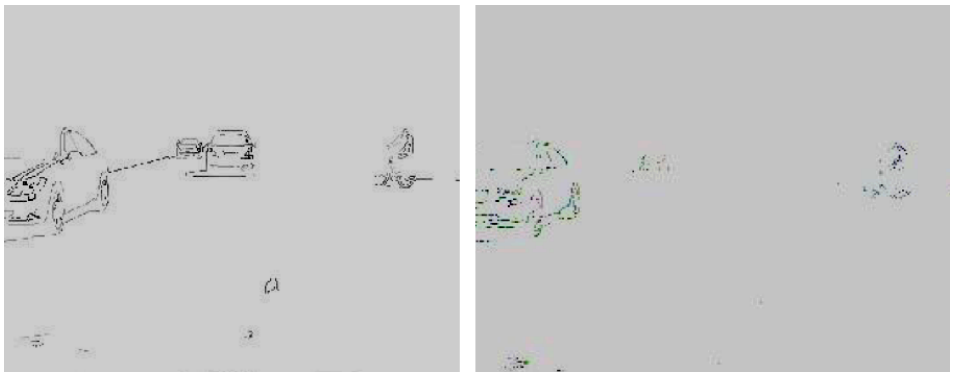

Fig. 10. Left: Canny edge image (inverted) for the Intern-on-Bike Sequence. Right: Only those edge pixels where the optic flow magnitude exceeds a threshold. Direction of optic flow is shown by hue, and length of vector by intensity (image also inverted). Courtesy by Xuan Guo, Zhongxia Ma, and Hao Xue, Auckland.

3.3 6D Analysis

Fusing stereo and motion analysis results together (i.e., 3D plus 3D, called *6D vision* in [6]), into one consistent interpretation of the scene, may allow to extract objects and their movement.

An *intersection approach* for fusion is to take all those pixels where motion information is evaluated as being reliable, and use the depth values at those pixels for combining motion and depth (or, vice-versa). However, this only allows to label very sparse pixel. Another idea is to segment depth maps, and to assign uniform motion vectors to those segments.

The use of Kalman filtering (see the book [8] or the online-tutorial [17]) is recommended to "smooth" and stabilize the movement of extracted features, and to generate more precise estimates. For example, [4] proposed this for the tracking of detected feature points, and [6] generalized this for an intersection approach.

Figure 11 illustrates a simple *background removal strategy* which uses camera calibration data which are provided for the seven test sequences (with respect to the camera's coordinate system, which is registered with respect to the car's coordinate system):

```
   [INTERNAL]
F = 820.428 # [pixel] focal length
SX = 1.0 # [pixel] pixel size in X direction
SY = 1.000283 # [pixel] pixel size in Y direction
X0 = 305.278 # [pixel] X-coordinate principle point
Y0 = 239.826 # [pixel] Y-coordinate principle point

   [EXTERNAL]
B  = 0.308084 # [m] width of baseline of camera rig
LATPOS = -0.07 # [m] lateral position of camera
HEIGHT = 1.26 # [m] height of cameras
DISTANCE = 2.0 # [m] distance of camera to rear axe
TILT = 0.06 # [rad] tilt angle
YAW = -0.01 # [rad] yaw angle
ROLL = 0.0 # [rad] roll angle
```

The camera's coordinate system is left handed: looking into driving direction along the z-axis, the x-axis points to the right, and the y-axis to the sky. The car coordinate system to camera coordinate system transform is the translation defined by "latpos", height, distance, followed by a rotation defined by tilt, yaw, and roll. A positive tilt means looking downward, a positive yaw means looking to the right, and a positive roll means clockwise.

Static *background* is anything what moves just (about) opposite to the movement of the host car. For calculated motion vectors, only those remain where the frame-to frame motion and the related depth information does not indicate a background situation. These are shown as dark dots in Figure 11.

Fig. 11. 6D vision result, showing only moving object points which are not classified as being static background. Clustering of the shown dots allows identification of those three cars and of the bicyclist. Courtesy by Xuan Guo, Zhongxia Ma, and Hao Xue, Auckland.

4 Proposed Evaluations

For evaluating stereo or motion algorithms, or combined 6D vision results on these real-world sequences, we divide meaningful features or objects (visible in these image sequences) below into examples of categories. Each group has its own characteristic properties and level of difficulty to be analyzed. By testing the ability to detect objects of different groups correctly in those sequences, an algorithm's performance can be demonstrated. (Such an algorithm is not defined by one task such as edge detection or optic flow calculation, but by a complex task of understanding a group of objects, or particular features in this group.) *For evaluation, each algorithm has to run on all seven sequences.*

Vehicles. The ability to detect any vehicles in any situation *within the space of relevance* is the key feature for algorithms to be used to assist in road driving. (A car parking along the roadside may start any moment.) In general, moving vehicles at close or medium distance (as in Figure 12) can be detected applying either stereo or motion analysis. The numbers of false-positive or false-negative counts defines the **first evaluation criteria**. The sequences contain various situations of vehicles where detection appears to be difficult:

1. *Vehicles that are far away from the host car.* Their contours are not very visible in dark environments such as in the Squirrel Sequence (see Figure 13), especially if also partially occluded. This group of events is certainly difficult to be detected by either stereo or motion analysis alone. Disparity and local displacement values are very small. Individually, analysis results may be confused with noise, but combined (6D) they form possibly a detectable cluster.
2. *Vehicles with similar speed and direction as the host car.* A constant relative position to the host car causes that local displacement vectors are close to zero. Of course, this identifies "similar motion". If the vehicle is not too far away from the host car, stereo analysis may be used to identify a constant distance. Object 2 in Figure 14 is a typical example in this group.

Fig. 12. Construction-Site Sequence, pair 185 of frames.

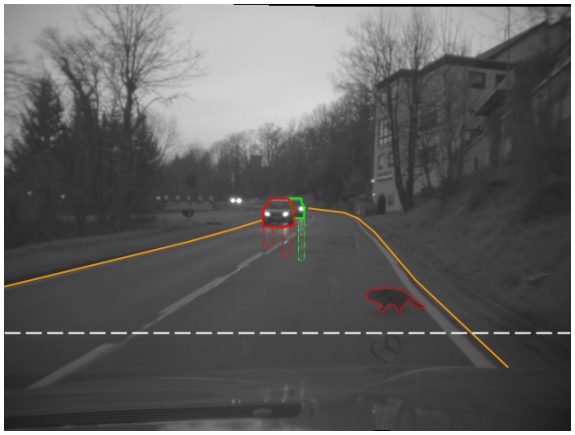

Fig. 13. Squirrel-Sequence, frame 178, left camera.

3. *Vehicle causing large local displacements.* A vehicle which is close to the host car, and which moves into a significantly different direction, will fit in general into this group. Such as Object 1 in Figure 14; the faded vehicle is Object 1's new position in the next frame. The local displacement between both green "T" marks is 35 pixel, and many motion-analysis algorithms will fail to detect that. Fusing stereo disparity and motion data in one Kalman filter is a recommended solution. Note that the perpendicularly approaching bicycle was about at the same position in the image for some time (constant viewing angle, but increasing in size), and at the moment of its turn it also falls into the category of a vehicle with large local displacements.
4. *Stopping or parking vehicles.* Vehicles which are not moving could be treated as background, but they still need to be detected and interpreted by a driver assistant system! For example, the car on the right-hand side of the road in Figure 15 is waiting to join the traffic. Stereo disparity can provide its

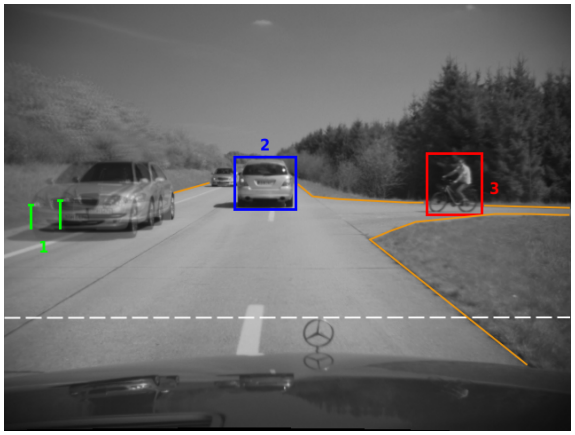

Fig. 14. Intern-on-Bike Sequence, frame 132, left camera.

Fig. 15. Save-turn sequence, frame 86, left camera.

geometry and position, but other statistical systems are needed to infer that this object is likely to join the traffic.

The identification of the trajectory (speed and direction) of the identified vehicle in world coordinates is the **second evaluation criteria**.

Humans and animals . Pedestrians and animals of relevant size belong to this group. (The cyclist counts as a vehicle.) Generally, members in this group have a much lower average speed than vehicles, but their motion pattern tends to be more stochastic. The bicyclist goes toward the main road, slows down when near the intersection, then turns the direction to avoid danger. The squirrel's behavior is opposite: it takes only 0.8s (20 frames) from it first appearance in the scene to be on the road surface, and another 0.32s (8 frames) to the center of the lane. Within this short period of time, an analysis system should find out the object's size, trajectory, and decide whether it is safe to continue going, or not. However, there are not many cases of humans or animals in those seven sequences, and, for this reason, we do not propose an evaluation criteria for this category.

Road edges, road surface, and safety fences. Road edges are defined by the road shoulder, lane marks, or other obstacles limiting the *free space* [15]. Lane detection is a standard task for highway driving.

For an ideal stereo algorithm and a perfectly flat road, surface depth is linearly distributed. Such a distribution defines a base plane for all moving objects, and object movement would be constrained to 2D. This may help when analyzing trajectories.

In real-world sequences, the surface of a road may not to be easily detectable by its depth distribution properties, because surface texture can affect the stereo algorithm, and some objects may obstruct the camera's sight. Road edges can be used to extrapolate the surface in between. Safety fences are shown some of the sequences (see Figures 12 and 16).

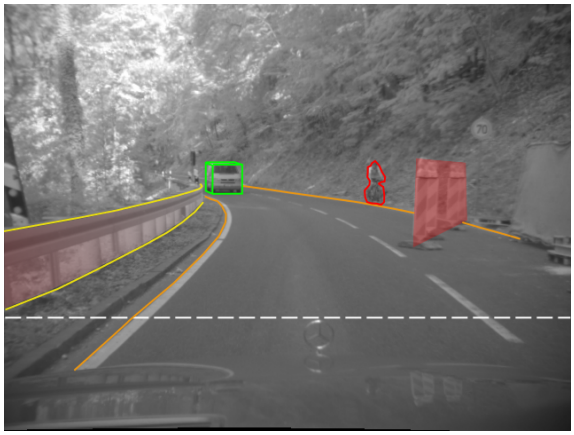

Fig. 16. Traffic-Light Sequence, frame 204, left camera.

The **third evaluation criteria** will be further specified on the download page for exact localization of road edges, safety fences, and road surface.

Real-world interference . There are a few events in those seven sequences which interfere with stereo and motion analysis. It is a challenge to find solutions to overcome them, and to deduct correct results.

Shadows on the road (see Figure 4) may significantly change the surface texture pattern and cause false window matching. Shadow on moving vehicles has an even worse effect, especially in motion analysis, because it may generate some highly incorrect optic flow patterns.

Changes in illumination is another problem in these sequences. Shadow, lighting angle, and many other factors may cause illumination differences between two subsequent frames, or between left and right images. Constant illumination is a basic assumption in many stereo or motion detection algorithms. The elimination of changes in illuminations, and the reduction of impacts of shadows defines the proposed **fourth evaluation criteria**. The processed image sequences can be used as input for algorithms for criteria one to three, and should lead to improved results.

5 Conclusions

This paper describes main features of seven real-world stereo image sequences. We briefly discussed a few stereo and motion estimation approaches and ways towards combined (i.e., 6D) analysis in the context of those sequences. A few examples of results were given for illustrating the potential of those sequences for use in performance analysis.

The paper suggests evaluation criteria to be used for testing the performance of algorithms on these seven real-world sequences.

Acknowledgements: The sequences were provided by Uwe Franke (and his colleagues at Daimler AG, Germany) at first only for teaching activities of the authors, and are now released into public domain. They can be freely used in academic research; see download site at *citr.auckland.ac.nz/6D*. Special thanks also go to Tobi Vaudrey (Auckland) who compiled those seven sequences while being in a research internship at Daimler AG in Germany.

References

1. H. Badino. A robust approach for ego-motion estimation using a mobile stereo platform. In Proc. *Int. Workshop Complex Motion*, pages 198–208, LNCS 3417, Springer, Berlin, 2006.
2. S. Baker and I. Matthews. Lucas-Kanade 20 years on: a unifying framework. *Int. J. Computer Vision*, **56**:221–255, 2004.
3. S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In Proc. *Int. Conf. Computer Vision*, to appear, 2007.
4. T. Dang, Ch. Hoffmann, and Ch. Stiller. Fusing optical flow and stereo disparity for object tracking. In Proc. *IEEE Int. Conf. Intelligent Transportation Systems*, pages 112–117, 2002.
5. S. Forstmann, Y. Kanou, J. Ohya, S. Thuering, and A. Schmitt. Real-time stereo by using dynamic programming. In Proc. *Computer Vision Pattern Recognition Workshop*, Volume 3, pages 29–36, 2004.
6. U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6D-vision - fusion of stereo and motion for robust environment perception. In Proc. *DAGM*, pages 216–223, 2005.
7. S. Guan and R. Klette. Belief-propagation on edge images for stereo analysis of image sequences. CITR-TR-208, Computer Science, The University of Auckland, Auckland, 2007.
8. M. H. Hayes. *Statistical Digital Signal Processing and Modeling.* John Wiley & Sons Inc., Hoboken, New Jersey, 1996.
9. B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, **17**:185–203, 1981.
10. J. Klappstein, F. Stein, and U. Franke. Detectability of moving objects using correspondences over two and three frames. In Proc. *DAGM*, pages 112–121, 2007.
11. R. Klette, K. Schlüns, and A. Koschan. *Computer Vision.* Springer, Singapore, 1998.
12. R. Klette, S. Stiehl, M. Viergever, and V. Vincken (editors). *Performance Evaluation of Computer Vision Algorithms.* Kluwer Academic Publishers, Amsterdam, 2000.
13. B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proc. *Int. Joint Conf. Artificial Intelligence*, pages 674–679, 1981.
14. N. Ohnishi, A. Imiya, L. Dorst, and R. Klette. Zooming optical flow computation. CITR-TR-197, Computer Science, The University of Auckland, Auckland, 2007.
15. J. M. Sanchiz, A. Broggi, and F. Pla. Stereo vision-based obstacle and free space detection in mobile robotics. In Proc. *Int. Conf. Industrial Engineering Appl. Artificial Intelligence Expert Systems* Volume 2, pages 280–289, 1998.
16. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, **47**:7–42, 2002.
17. G. Welch and G. Bishop. An Introduction to the Kalman Filter. `www.cs.unc.edu/~welch/kalman/index.html`.