Evaluation of Moving Object Segmentation Comparing 6D-Vision and Monocular Motion Constraints

Tobi Vaudrey[1], Andreas Wedel[2], Clemens Rabe[2], Jens Klappstein[2], Reinhard Klette[1]

[1] The *.enpeda..* Project, The University of Auckland, New Zealand
[2] Environment Perception Group, Daimler AG, Germany

Abstract

Detecting moving objects is a very important aspect of driver assistance systems (DAS). This paper handles this issue by using a vision based system mounted within the vehicle. The pipeline for both a stereoscopic and monocular approach are covered. Both approaches use image sequences and compare moving feature points over time. This sparse information is then segmented using the optimal graph-cut algorithm, by also incorporating the grey-scale images. This paper then evaluates and contrasts the two approaches to identify the accuracy and robustness of each approach. The two methods both work in real-time on normal PC hardware (Quad Core CPU).

Keywords: driver assistance systems (DAS), motion detection, stereo analysis, segmentation, evaluations

1 Motion Detection and DAS

Motion is one of the major cues for human perception. Detecting moving objects is also a major issue for driver assistance systems (DAS) and road safety. The authors consider the detection of moving traffic participants to be an important step toward attention-based environment perception.

This paper investigates methods and limitations of both monocular and binocular camera systems for motion detectability. It is evident that a monocular system is cheaper, uses less installation space, and suffers less decalibration issues, compared to a stereo system. However, a stereo system yields direct range measurement estimates (e.g. [5]), but the orientation between the two cameras needs to be known accurately, and decalibration can cause major issues. This paper provides insight into the difference between monocular and stereo camera performance.

The motion of the ego-vehicle greatly complicates the problem of motion detection because simple background subtraction of successive images yields no result. The key idea behind our approach of detecting independently moving objects is to distinguish between motion in the images caused by the ego-motion of the ego-vehicle (static objects) and motion caused by dynamic objects in the scene. This paper presents and investigates techniques to distinguish between stationary and non-stationary points. They are based on tracking feature points in sequential images.

Section 2 presents our algorithm, starting with an investigation of motion analysis techniques, and followed by presenting segmentation of the moving objects from the static scene. In Section 3, different scenarios are presented and analysed, confirming the practicality of computer vision for the sensation and perception of motion. Differences between monocular and binocular motion detection are discussed and segmentation results for moving objects are presented. The concluding section is on future work and obtained insights.

2 Our Algorithm

The proposed algorithm is able to find both rigid objects such as cars and non-rigid objects such as moving pedestrians, and it is subdivided into two main steps:

Step 1. As a first result, feature points on independently moving objects are detected as moving. These features, however, are sparse and do not characterize the whole image.

Step 2. In a second step, moving objects are segmented in the images using these sparse features as seeds for segmentation. We make use of the globally optimal graph-cut segmentation algorithm [2] to reject outliers and to find image regions with an accumulation of image features lying on moving objects.

978-1-4244-2582-2/08/$25.00

2.1 Tracking of Image Features

The detection of moving objects is based on a motion analysis of individual tracked image features. The tracking is done employing the optical flow method KLT, developed by [10]. The tracked features are then reconstructed into 3D coordinates. The stereoscopic approach accomplishes this using a pair of stereo images by estimating the disparity and using triangulation (e.g. [4]), where as the monocular approach accomplishes this using sequential images and evaluating the optical flow. The monocular approach additionally requires the knowledge about the ego-motion of the camera which can be obtained either by an inertial measurement unit (IMU) or by the optical flow itself [1].

Both approaches, monocular and stereoscopic, provide a motion metric which is correlated to the likelihood that the point is moving. This motion metric serves as input for the segmentation. The sequence of operations is summarized in Figure 1.

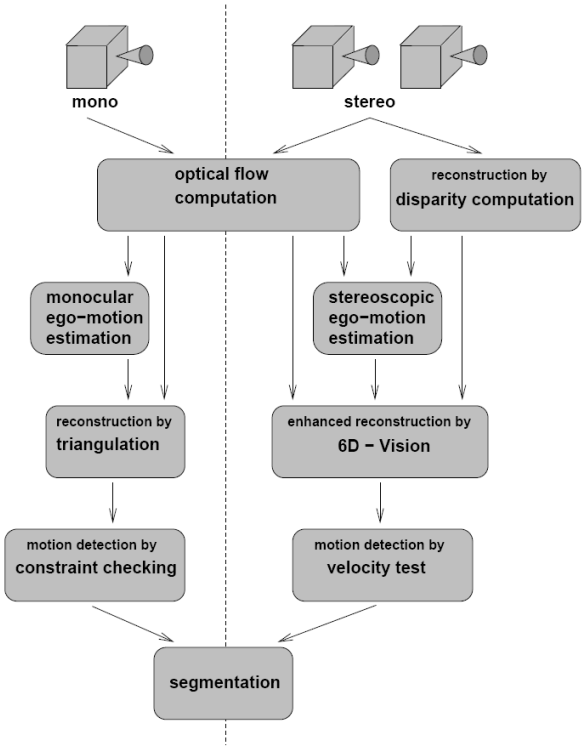


Figure 1: Sequence of operations. Both approaches, monocular and stereoscopic, have the computation of the optical flow and the segmentation in common. The greatest difference is the way in which the motion is detected.

2.1.1 Monocular Motion

A monocular vision system can not reconstruct the 3D position of a scene point in one time frame. It relies on structure from motion, to gather 3D data over time. Points that do not fulfill the constraints of a static 3D point are classified as moving. These constraints are as follows:

Epipolar Constraint: The epipolar constraint expresses that the viewing rays of a static 3D point (the lines joining the projection centres and the 3D point) must meet. A moving 3D point in general induces skew viewing rays violating the constraint.

Positive Depth Constraint: The fact that all points seen by the camera must lie in front of it is known as the positive depth constraint. It is also called cheirality constraint. If viewing rays intersect behind the camera the actual 3D point must be moving.

Positive Height Constraint: All 3D points must lie above the road plane. If viewing rays intersect underneath the road the actual 3D point must be moving. This constraint requires the knowledge about the normal vector of the road plane and the camera distance to the road plane. These entities are estimated exploiting the optical flow on the road [7].

Trifocal Constraint: A triangulated 3D point utilizing the first two views must triangulate to the same 3D point when the third view comes into consideration. This constraint is also called trilinear constraint.

An algorithm evaluating all available constraints quantitatively is outlined in the block diagram (Figure 1), with the reconstruction and detection shown as two separate steps. However, the actual algorithm avoids the explicit reconstruction in favour of a reduced computational complexity and a better statistical manageability.

Motion Metric

The motion metric is a combination of two approaches; the two-view and the trifocal constraint.

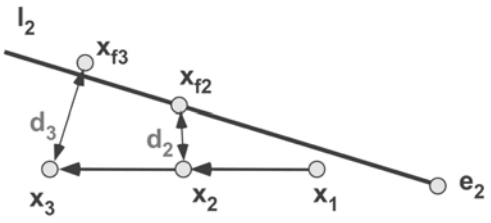


Figure 2: Monocular motion metric. Point $\mathbf{e_2}$ is in the center of the image (an example of a point moving from the focus of expansion). The camera moves forward along its optical axis observing a lateral moving point $\mathbf{x_1} \leftrightarrow \mathbf{x_2} \leftrightarrow \mathbf{x_3}$. The closest point to $\mathbf{x_2}$ fulfilling the two-view constraints is $\mathbf{x_{f2}}$. The error arising from two-views is the distance d_2. Transferring the points $\mathbf{x_1}$ and $\mathbf{x_{f2}}$ into the third view yields $\mathbf{x_{f3}}$. If the observed 3D point was actually static, $\mathbf{x_3}$ would coincide with $\mathbf{x_{f3}}$. However, the 3D point is moving which causes the trifocal error d_3. Note: in general, $\mathbf{x_1}$ and $\mathbf{x_{f3}}$ do not lie on the epipolar line $\mathbf{l_2}$.

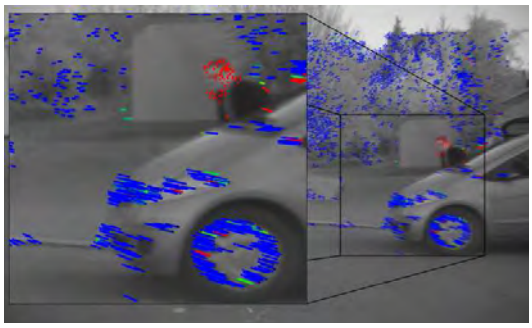


Figure 3: The figure shows the monocular motion analysis for a moving pedestrian appearing behind a stationary vehicle. The colour encoding corresponds to the motion metric (blue: 0 px, red: 2 px).

Full details of the two-view constraint can be found in [6]. Basically, the approach incorporates the epipolar, positive depth, and positive height constraints. This yields a distance error $d_2(\mathbf{x})$ (in pixels) from the expected motion, see Figure 2 for a brief summary.

For the trifocal constrain consider the correspondence $\mathbf{x_1} \leftrightarrow \mathbf{x_2} \leftrightarrow \mathbf{x_3}$. $\mathbf{x_{f2}}$ is defined such that it fulfills the two-view constraint (i.e., $\mathbf{x_1}$ and $\mathbf{x_{f2}}$ constitute a valid 3D point). This 3D point is projected into the third view yielding $\mathbf{x_{f3}}$. The measured image point $\mathbf{x_3}$ will coincide with $\mathbf{x_{f3}}$ if the observed 3D point is actually static. Otherwise there is a distance d_3 (see Figure 2) between them which we call trifocal error. $\mathbf{x_{f3}}$ is computed via the point-point-point transfer using the trifocal tensor [4].

The overall error metric used is (see Figure 3 for an example of results):

$$d(\mathbf{x}) = d_2(\mathbf{x}) + d_3(\mathbf{x}) \tag{1}$$

2.1.2 Stereo Motion (6D-Vision)

The core algorithm of the stereo vision system presented here (6-D vision) follows the principle of fusing optical flow and stereo information given in [3]. The basic idea of 6-D vision is to track points in the image (using KLT), with depth estimated from stereo vision (e.g., [5]); these points are tracked over two or more consecutive frames and the spatial and temporal information is fused using Kalman filters [12]. The result is a relatively accurate estimation of the 3D-position and 3D-motion of feature points, thus 6-D vision.

The fusion implies the knowledge of the ego-motion. In our system we compute ego-motion from image points found to be stationary using a Kalman filter based approach described in [9]. This allows a fast calculation using all information already acquired by the system including inertial sensor data.

The result of the 6D-Vision algorithm are illustrated in Figure 4, showing a pedestrian appearing behind a stationary vehicle. The image is taken from a moving vehicle, driving at about 30 km/h . We see, that 160 ms after the pedestrian's head was first visible, an estimation of its motion is already available, which allows analysis for the risk of collision. Full details of the 6D vision approach, including the *system model* and *measurement model* of the Kalman filter can be found in [3].

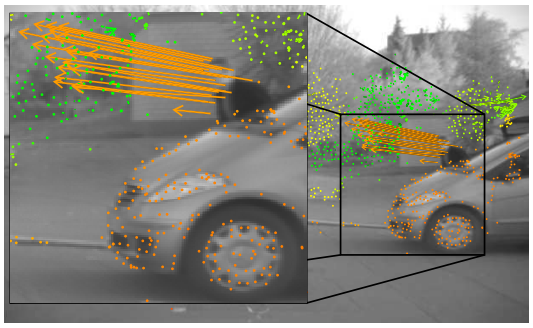


Figure 4: The figure shows the stereo motion analysis for a moving pedestrian appearing behind a stationary vehicle. The arrows point to the estimated 3D position in 0.5 s, reprojected into the image. The colour encoding corresponds to the estimated depth (close = red, far = green).

Motion Metric

The motion metric $d(x)$ used for this paper is simply the absolute velocity (i.e., speed) of the ego-motion compensated 6D-Vision's 3D velocity vector:

$$d(\mathbf{x}) = \left| \left[\dot{X}(\mathbf{x}), \dot{Y}(\mathbf{x}), \dot{Z}(\mathbf{x}) \right]^\top \right| \tag{2}$$

where $\mathbf{x} = (u, v)^\top$ is the pixel position and $\dot{X}$,$\dot{Y}$ and $\dot{Z}$ are the Cartesian velocities (in m/s) for the lateral, vertical and depth direction respectively. $|\cdot|$ is the ℓ^2 norm.

2.2 Segmentation - Graph Cut

In order to derive objects from individually tracked image features, the features have to be clustered into coherent objects. Image features are usually sparse and appropriate for ego-motion estimation, however, they are not sufficient to describe whole objects or object boundaries.

We therefore find objects by segmenting the image into foreground (moving objects) and background (stationary world) taking the motion metric values as probabilities for the tracked image features. Image features with values above a noise threshold vote for foreground, all other features below the threshold vote for background. The noise in the motion metric is mainly due to the tracking and disparity measuring inaccuracies. For monocular motion analysis we assume an inaccuracy of $\sigma = 0.1$ px, for the stereo approach the threshold is

set at $\sigma = 1.0$ m/s. Accumulations of such foreground seeds denote an object. Single features with a high error metric value need to be rejected as outliers. We define an energy which penalizes boundary length of object segments. The energy is then minimized using the global optimal graph-cut algorithm [2]. Further speed up techniques for flow vector segmentation can be achieved using the multi-resolution graph-cut [11].

In a first step, every image pixel $\mathbf{x}$ corresponds to a node in a graph with a source node s representing the background and a sink node t for the foreground. Pixels voting for background are connected via an (undirected) edge to the source node, those voting for foreground to the sink node vice versa. The cost of an edge is defined as

$$d(\mathbf{x}) < \sigma \quad \Rightarrow \quad e(s, \mathbf{x}) = \sigma - d(\mathbf{x}) \tag{3}$$

$$d(\mathbf{x}) > \sigma \quad \Rightarrow \quad e(\mathbf{x}, t) = \min(d(\mathbf{x}) - \sigma, C_{max}) \tag{4}$$

where C_{max} is a threshold to limit outliers. The minimum function is necessary to limit the influence of wrong tracks (outliers) on the result. Additionally, neighbouring image pixels (here only the 4 next neighbours are taken into account) are connected by edges. The costs of these edges depend on the grey value difference of its two end points. The cost values are defined by

$$e(\mathbf{x}, \mathbf{y}) = \frac{C_e}{\|I(\mathbf{x}) - I(\mathbf{y})\| + \varepsilon} \tag{5}$$

where C_e is a constant scale factor, used to regularize the influence of edge costs (boundary length), and ε is a small value to prevent numerical instability. $I(\mathbf{x})$ is the grey value of $\mathbf{x}$, in our case a scalar value between 0 and 4095, as we use 12 bit images. Equation 5 is designed such that segmentation boundaries along high image gradients are more likely than in homogeneous regions.

Clearly, the result depends on the costs of the edges, especially on the constant C_e. If C_e is too low, the segmentation only contains single pixels whereas a high value of C_e results in only one small segment (or no segment at all) because removing edges to the source or the sink becomes less costly than removing those edges connecting image pixels. Both situations can be seen in Figure 5(b). If the sum of all edges of a pixel is larger than C_{max}, the pixel will not be cut. Therefore we set:

$$\forall \text{ tracked } \mathbf{x}: \; e(\mathbf{x}, \mathbf{y}) = 0.5 \; C_{max} \tag{6}$$

To regularize the size of the segments, especially in low-contrast regions such as the road surface, the number of foreground pixels is penalized. This is done by adding additional edges with constant cost from every node to the source

$$\forall \, \mathbf{x}: \; e(s, \mathbf{x}) = C_{BG} \tag{7}$$

(a)

(b)

Figure 5: The main figure (a) shows the segmentation for a moving pedestrian appearing behind a stationary vehicle. (b) shows the influence of the edge costs on the segmentation result. Edge cost from left to right: $C_e = \{1.5, 50, 500, 1000\}$.

This is equivalent to adding a background prior for every pixel in the image. In the following results section we use constant values for the determinable parameters of the algorithm demonstrating the adaptability of the algorithm for different scenarios:

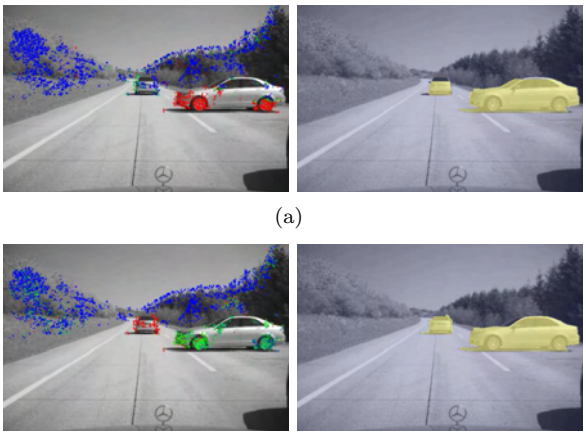
$$C_{max} = 6 \qquad C_e = 150 \qquad C_{BG} = 0.01$$

This is a usual mapping of image pixels onto a graph representation as done in [2]. A cut in a graph is found by removing edges such that no more connections between source and sink exist. The cost of a cut is the sum of its comprised edges. The minimal cut is defined as the cut with the minimal cost out of all possible cuts in the graph. A good overview and diagram of graph-cut can be seen in [11].

3 Experimental Results

In this section we apply our motion analysis and segmentation to real imagery. We use the same set of features (KLT tracks) for the monocular and binocular motion analysis. The chosen parameters for the segmentation are given in Section 2.2.

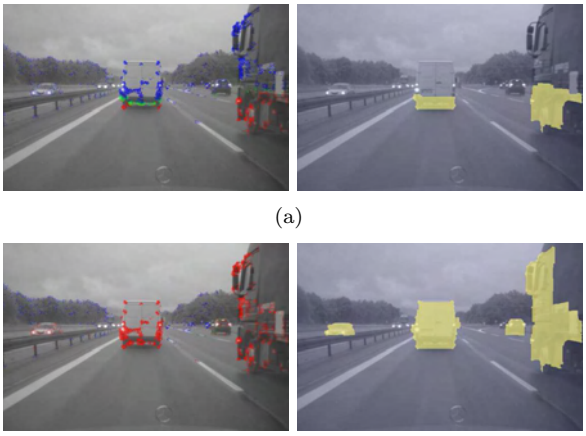
The first example in Figure 5(a) shows the segmentation of the pedestrian appearing behind a stationary vehicle. The segmentation boundary proves to be accurate keeping in mind that features are sparse in the image (see Figures 4 and 3). The monocular and the stereo approach yield exactly the same segmentation result for the lateral moving



(a)

(b)

Figure 6: Tracked image features (left) and segmentation (right) results of a crossing and a preceding object. The monocular approach (a) performs similar to the stereoscopic approach (b). The tracked image features are color encoded according to the corresponding motion metric. (a) ranges from 0 px (blue) to 7 px (red). (b) ranges from 0 m/s (blue) to 7 m/s (red).



(a)

(b)

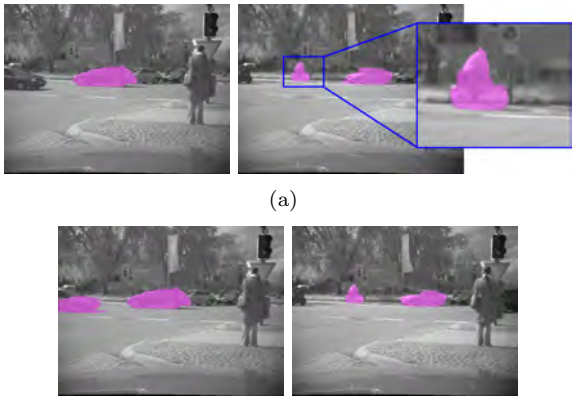
Figure 7: Detection and segmentation results of preceding and oncoming objects. The monocular approach (a) only detects the lower parts of the preceding objects. The oncoming object is not detected at all. The stereoscopic approach (b) does not suffer from these limitations (color encoding as in Figure 6).

pedestrian. This shows that for even small image regions (50×50 pixels), the sparse features allow a good segmentation.

Figure 6 shows a traffic scene with a crossing car and a preceding car at a distance of 31 m. The speed of both cars is approximately 36 km/h. Both approaches, monocular and stereo motion analysis, yield similar segmentation results. Looking at the motion metric values, which are the driving energies for the graph-cut segmentation, the difference between both approaches becomes visible. In the monocular case, the energy values of features located on the preceding car are small. This is due to the fact that the car moves longitudinal at a high distance and the corresponding flow vectors do not differ much from those generated by stationary objects. On the other hand, most flow vectors induced by the crossing car deviate from any flow vectors of stationary objects, which fulfill the monocular motion constraints. However, the flow vectors in the vicinity of the horizon are similar to those generated by stationary objects. The segmentation result still is accurate and both moving vehicles are detected. For a more detailed investigation of these phenomena refer to [8]. The stereo approach measures the absolute 3D velocities of tracked features. The preceding car is moving at a relatively high speed of 36 km/h while the crossing car is moving at lower speed. This is clearly represented by the motion metric. In contrast to the monocular approach, all features on both cars yield correct results as the stereo approach does not suffer from the motion ambiguity between features on moving and stationary objects. The preceding car is therefore fully segmented.

This situation becomes even more evident when looking at the autobahn sequence in Figure 7. The vehicles move with a speed of 84 km/h. The monocular approach is able to detect the car driving ahead, and the truck, being overtaken, on the right side. But only the lower parts of the vehicles are detected, resulting in an incomplete segmentation of the vehicles. The stereo approach not only detects the vehicles completely, it is also able to detect oncoming traffic (car to the left) and longitudinally moving traffic outside of the vicinity of the focus of expansion (car to the right).

The final example is an intersection with cars and a cyclist (Figure 8). The left images in this figure show that the stereo approach can detect vehicles even near the image borders, where as the monocular approach fails to detect the vehicle to the left. The right hand images shows that both ap-



(a)

(b)

Figure 8: This figure shows the segmentation results from an two images in an intersection sequence that contains a cyclist. (a) shows the results from the monocular approach and (b) the stereo approach.

proaches can detect different moving objects with much different sizes. The cyclist is detected and segmented as well as the moving vehicles in the scene. The pedestrian is stationary (as can be seen by no movement between the two images) so there is no segmentation.

4 Conclusions

Kinesthesia, the sensation or perception of motion, is one important part of human perception. It encompasses both the perception of motion of one's own body and a spectators perception of the motion of a scene. In vehicle applications these two steps refer to ego-motion and the detection of other moving traffic participants. Visual kinesthesia is done by using the sense of sight to observe the effect of scene motion. In this paper, we modeled such perception of motion using computer vision.

We investigated a monocular and a stereoscopic approach to perceive motion in image sequences. For each approach a motion metric was introduced measuring the likelihood that a tracked image feature corresponds to a moving 3D point. We applied the motion metrics and segmentation approach to traffic scenes captured by a camera installed in a moving vehicle. Using image segmentation based on the investigated motion metrics we were able to detect and segment other moving traffic participants.

Image sequences on highways and urban roads using the same parameters demonstrate the practicality of this robust novel approach for motion segmentation. We showed that both approaches are able to detect; cars, trucks, cyclists, and pedestrians, high-lighting that this segmentation is a good initialisation for other classification tools.

On average, the stereo approach outperforms the monocular approach in terms of accuracy. However, with stereo there is a higher computational cost as stereo disparity needs to be estimated. In saying that, both approaches do run in real-time (20 Hz) on standard off the shelf PC hardware (Pentium Quad Core).

Future work in this area may consist of integrating the tracking of features in the monocular approach for a temporal integration of information. Also the extension of the segmentation algorithm to distinguish between different motion directions is in the scope of future work, to be able to determine different objects and obstacles.

References

[1] X. Armangué, H. Araújo, and J. Salvi. Differential epipolar constraint in mobile robot egomotion estimation. In Proc. *IEEE Int. Conf. Pattern Recognition*, pages 599–602, 2002.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 359–374, 2001.

[3] U. Franke, C. Rabe, H. Badino, and S. Gehrig. 6d-vision: Fusion of stereo and motion for robust environment perception. In Proc. *DAGM Symposium*, pages 216–223, 2005.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Press, second edition, 2003.

[5] H. Hirschmüller. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In Proc. *IEEE Conf. Computer Vision Pattern Recognition*, volume 2, pages 807–814, 2005.

[6] J. Klappstein, F. Stein, and U. Franke. Monocular motion detection using spatial constraints in a unified manner. In Proc. *IEEE Intelligent Vehicles Symposium*, 2006.

[7] J. Klappstein, F. Stein, and U. Franke. Applying kalman filtering to road homography estimation. In Proc. *Workshop Planning Perception Navigation Intelligent Vehicles* (in conjunction with IEEE Int. Conf. Robotics Automation), 2007.

[8] J. Klappstein, F. Stein, and U. Franke. Detectability of moving objects using correspondences over two and three frames. In *Pattern Recognition (Proc. DAGM)*, pages 112–121. Springer, 2007.

[9] C. Rabe, U. Franke, and S. Gehrig. Fast detection of moving objects in complex scenarios. In Proc. *Proc. IEEE Intelligent Vehicles Symposium*, pages 398–403, 2007.

[10] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report CMU-CS-91-132, Carnegie Mellon University, 1991.

[11] T. Vaudrey, D. Gruber, A. Wedel, and J. Klappstein. Space-time multi-resolution banded graph-cut for fast segmentation. In Proc. *DAGM Symposium*, pages 203–213. Springer, 2008.

[12] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report TR 95-041, University of North Carolina at Chapel Hill, Department of Computer Science, 1995.