Realtime Depth Estimation and Obstacle Detection from Monocular Video

Andreas Wedel[1,2], Uwe Franke[1], Jens Klappstein[1],
Thomas Brox[2], and Daniel Cremers[2]

[1] DaimlerChrysler Research and Technology, REI/AI,
71059 Sindelfingen, Germany
[2] Computer Vision and Pattern Recognition Group,
Rheinische Friedrich-Wilhelms Univeristät, 53117 Bonn, Germany

Abstract. This paper deals with the detection of arbitrary static objects in traffic scenes from monocular video using structure from motion. A camera in a moving vehicle observes the road course ahead. The camera translation in depth is known. Many structure from motion algorithms were proposed for detecting moving or nearby objects. However, detecting stationary distant obstacles in the focus of expansion remains quite challenging due to very small subpixel motion between frames. In this work the scene depth is estimated from the scaling of supervised image regions. We generate obstacle hypotheses from these depth estimates in image space. A second step then performs testing of these by comparing with the counter hypothesis of a free driveway. The approach can detect obstacles already at distances of 50m and more with a standard focal length. This early detection allows driver warning and safety precaution in good time.

1 Introduction

Automatic detection and verification of objects in images is a central challenge in computer vision and pattern analysis research. An important application is robustly hypothesizing and verifying obstacles for safety applications in intelligent vehicles. The practical value of such systems becomes evident as obstacle

Fig. 1. Six out of ten front–end crashes could be prevented if safety systems reacted a split second earlier than the driver. Detecting arbitrary obstacles from monocular video in the road course ahead, however, is quite challenging.

K. Franke et al. (Eds.): DAGM 2006, LNCS 4174, pp. 475–484, 2006.

detection is a prerequisite to warn the driver of approaching hazards (see Fig. 1). Commonly used radar sensors lack detecting static objects therefore we tackle this problem using computer vision.

For a camera mounted in a vehicle with a given camera translation in depth, detecting obstacles in traffic scenes has three major challenges:

1. The algorithm has to run in real time with minimum delay in reaction time.
2. Obstacles have to be detected at large distances to route the vehicle or warn the driver as early as possible.
3. The position and horizontal dimension of an obstacle have to be estimated precisely to safely guide the vehicle in case an emergency brake is insufficient.

The first two challenges demand an algorithm able to detect obstacles in the focus of expansion where optical flow displacement vectors between consecutive frames are extremely small. Overcoming this by skipping frames violates the first constraint. The last challenge requires robust verification of obstacle boundaries.

Traditional vision based obstacle detection relies on depth estimation from stereo systems [7]. Such systems work well. However, single cameras are already available in series production performing numerous vision based driver assistance algorithms such as intelligent headlight control and night view. Obstacle detection from a single camera is, hence, a desirable alternative.

According to [1,14] obstacle detection in monocular vision can be split into methods employing a–priori knowledge and others based on the relative image motion. Former algorithms need to employ strict assumptions regarding the appearance of observed objects. Since we are interested in a model free approach, we have to use latter methods. Proposed realtime optical flow algorithms [3,11,13] and obstacle detection based on those [12] calculate the displacement between consecutive frames of an image sequence. In such a basic approach, integrating flow vectors over successive image pairs is subject to drifts and therefore these algorithms are not suitable for the posed problem. Moreover, these methods detect obstacles in two steps firstly calculating flow vectors for every pixel and secondly analyzing those flow vectors. Working directly in image space is more desirable as all the information available is accessed directly.

We propose an obstacle detection algorithm in the two standard steps:

1. **Hypothesis generation from** estimating **scene depth** in image space.
2. **Candidate testing by analyzing** perspective **transformation** over time.

In the first step, conclusions about scene depth are drawn from the scaling factor of image regions, which is determined using region tracking. We use the tracking algorithm described in [8] which is consistent over multiple frames of an image sequence and directly estimates scale and translation in image space. For an evaluation of different tracking algorithms we refer to [5]. If distance measurements fall below a given threshold, obstacle hypotheses are generated. Due to the restricted reliability of depth from region scaling, such an approach can result in false hypotheses which have to be dismissed.

The testing of generated hypotheses is performed in the second step. We check whether the observed perspective distortion over time corresponds to an obstacle

with distance given from hypothesis generation or to a free driveway. In such an approach we are able to detect arbitrary obstacles directly in image space. The two steps will be investigated separately in Sects. 2 and 3. Experimental results on real image data can be found in Sect. 4. A final conclusion and motivation for further work will be given in Sect. 5.

2 Depth Tracking from Monocular Video

This section investigates the mathematical foundation for reconstructing scene depth from monocular vision. First we describe the underlying perspective projection and the model used for depth computation. Then we describe how depth information can be computed from scaling of image regions and how this fact can be used to efficiently detect stationary obstacles. Finally we investigate the error in depth estimation.

We use a monocular camera mounted on a vehicle such that the camera's optical axis e_3 coincides with the vehicle translation direction. The reference system is the left–handed camera coordinate system $(0, e_1, e_2, e_3)$ with the e_2 unit vector being the ground plane normal. In particular, we assume a flat ground and a straight road to travel on. The image plane has equation $Z = f$, where f is the focal length of the camera. The ground plane is $Y = -Y_0$ with the camera height Y_0. For a point $\boldsymbol{X} = (X, Y, Z)^\top$ in 3–D space we obtain the corresponding image point $\boldsymbol{x} = (x, y)^\top$ by a perspective projection:

$$\boldsymbol{x} = \frac{f}{Z} \begin{pmatrix} X \\ -Y \end{pmatrix} . \tag{1}$$

In practice the camera coordinate system e_3 axis usually is not parallel to the ground plane. Camera rotation can be compensated transforming the camera to a virtual forward looking camera in a similar way as described in [9]. The camera translation in depth between consecutive frames is known from inertial sensors.

Obstacles are assumed to be axis parallel bounded boxes. This states that the Z coordinate of the obstacle plane facing the camera is constant. In practice the relative depths on obstacle surfaces are small compared to the distance between obstacle and camera such that this assumption is a good approximation.

Let $\boldsymbol{X}(t) = (X(t), Y(t), Z(t))^\top$ be a point at time t and $\boldsymbol{x}(t)$ its projected image point. The camera translation in depth between time t and $t+\tau$ is $\boldsymbol{T}(t, \tau)$ leading to $\boldsymbol{X}(t+\tau) = \boldsymbol{X}(t) + \boldsymbol{T}(t, \tau)$. The camera translational and rotational velocity is $\dot{\boldsymbol{T}}(t)$ and $\dot{\boldsymbol{\Omega}}(t)$ respectively. Particular coordinates are represented by subscripted characters. Traditional structure from motion algorithms based on optical flow involve using the image velocity field mentioned by Longuet-Higgins and Prazdny in [10] (the time argument is dropped due to better readability):

$$\dot{\boldsymbol{x}} = \frac{1}{Z} \begin{pmatrix} x\dot{T}_Z - f\dot{T}_X \\ y\dot{T}_Z - f\dot{T}_X \end{pmatrix} - \begin{pmatrix} \dot{\Omega}_X \frac{xy}{f} + \dot{\Omega}_Y \left(f + \frac{x^2}{f} \right) + \dot{\Omega}_Z y \\ \dot{\Omega}_X \left(f + \frac{y^2}{f} \right) + \dot{\Omega}_Y \frac{xy}{f} + \dot{\Omega}_Z x \end{pmatrix} . \tag{2}$$

Such algorithms are exact for time instances where image velocities are measurable. However, flow vectors measure the displacement of image points between frames. Therefore resolving (2) using an explicit or implicit integration method induces drifts by adding up errors in inter–frame motions. We divide the motion of image regions into two parts and show that under the given conditions the scene depth can be estimated solely by estimating the scaling factor of image regions. The transformation of an image point for a pure translation using (1) becomes

$$\boldsymbol{x}(t+\tau) = \frac{f}{Z(t+\tau)} \begin{pmatrix} X(t+\tau) \\ Y(t+\tau) \end{pmatrix} = \frac{f}{Z(t)+T_Z(t,\tau)} \begin{pmatrix} X(t)+T_X(t,\tau) \\ Y(t)+T_Y(t,\tau) \end{pmatrix} \quad (3)$$

$$= \underbrace{\frac{Z(t)}{Z(t)+T_Z(t,\tau)}}_{s(t,\tau)} \underbrace{\frac{f}{Z(t)} \begin{pmatrix} X(t) \\ Y(t) \end{pmatrix}}_{\boldsymbol{x}(t)} + \frac{f}{Z(t)+T_Z(t,\tau)} \begin{pmatrix} T_X(t,\tau) \\ T_Y(t,\tau) \end{pmatrix} . \quad (4)$$

It should be pointed out, that we use absolute image coordinates and not velocities for computation. With a correctly given vehicle translation and displacement of image points, scene depth can be directly calculated over large time scales. As only the translation in depth $T_Z(t,\tau)$ is known, a single observation is not sufficient to determine scene depth. With the assumed model though, front faces of obstacles have equal Z coordinates and therefore multiple observations in an image region can be used to solve an over–determined equation system for the scaling factor and the translation.

The key for depth reconstruction is to use the scale $s(t,\tau)$ directly obtained by the used region tracking over multiple frames to calculate scene depth:

$$d \equiv Z(t) = \frac{s(t,\tau)}{1-s(t,\tau)} T_Z(t,\tau) . \quad (5)$$

Distance histogram. For reconstructing scene depth we observe the image region to which obstacles in 30 m distance with 0.9 m height are mapped (compare Fig. 4). This region is divided up into n overlapping image regions $\{R_i\}_{i=0}^{n}$, which are individually tracked until their correlation coefficient surpasses a fixed threshold. The estimated distances of the tracked regions are projected onto the x–axis in image space (which can be regarded as a discrete resolution of the viewing angle) to receive a distance histogram. Projected obstacle distances are weighted by distance from region center. An appropriate weighting function is the triangular hat function Δ_{R_i} defined on the region width. With the image region distance $d(R_i)$ and the characteristic function $\chi_{R_i}(x) = 1 \Leftrightarrow x \in R$ this results in the following distance histogram (see Fig. 5):

$$d(x) = \frac{1}{\sum_i \chi_{R_i}(x)\Delta_{R_i}(x)} \sum_i \chi_{R_i}(x)\Delta_{R_i}(x)d(R_i) . \quad (6)$$

Error in depth estimation. In this paragraph we will show how depth variance can be calculated by error propagation taking into account errors due to rotation as well. In real case scenarios rotational effects occur as steer angle, shocks and vibrations experienced by the vehicle can introduce rapid and large transients in image space.

Recalling that (2) involves velocities, we use explicit Euler integration for modeling the incremental rotational transformation. For the analysis of rotational errors the translational part can be set to zero leading to:

$$\boldsymbol{x}(t+\tau) = \boldsymbol{x}(t) + \tau \begin{pmatrix} \dot{\Omega}_X(t)\frac{x(t)y(t)}{f} + \dot{\Omega}_Y(t)\left(f + \frac{x(t)^2}{f}\right) + \dot{\Omega}_Z(t)y(t) \\ \dot{\Omega}_X(t)\left(f + \frac{y(t)^2}{f}\right) + \dot{\Omega}_Y(t)\frac{x(t)y(t)}{f} + \dot{\Omega}_Z(t)x(t) \end{pmatrix} . \quad (7)$$

The constant terms in (7) will influence only the translation estimation and therefore keep the scaling factor unchanged. The influence of the roll rate ($\dot{\Omega}_Z$) when looking at each image coordinate equation by itself is constant, too. The yaw rate ($\dot{\Omega}_Y$) and pitch rate ($\dot{\Omega}_X$) are linear and quadratic in the image coordinates and therefore will influence the scale factor. Let s be the estimated scale factor, e_s the error in scale estimation, and $\hat{s}$ the true scale with zero rotation. From (7) it follows that

$$s = \hat{s} + e_s + cx + cy \; . \quad (8)$$

However, assuming the yaw and pitch angle to be bounded by $\pm 10°$ and the focal length to be greater than 800 px leads to

$$c \equiv \frac{\tau\dot{\Omega}}{f} \in \left[-2.2 \cdot 10^{-4},\; 2.2 \cdot 10^{-4}\right] \; . \quad (9)$$

The limited image size (of 640×480 pixel) and the bounded values of the rotation parameters therefore limit the effect on estimation of region scale. With known scale variance from tracking σ_s^2 and variance in translation $\sigma_{T_Z}^2$ the depth variance can be calculated by error propagation from (5) and (8) via:

$$\sigma_d^2 = \frac{1}{(1-s)^2}\; T_Z\; (\sigma_s + x\sigma_c + y\sigma_c)^2 + \frac{s}{1-s}\; \sigma_{T_Z}^2 \; . \quad (10)$$

It has to be pointed out that the relative error in scale estimation becomes smaller as the scale factor increases, such that the influence of rotation on the scaling factor becomes negligible over large time scales (see Fig. 3).

The next section deals with obstacle detection based on the distance histogram from (6). The separation between depth estimation and obstacle detection allows for usage of distance histograms generated by alternative sensors (e.g. a scanning radar) for a sensor–fusion. Results obtained from depth estimation can be found in Sect. 4.

3 Obstacle Detection by Hypothesis Verification

The distance histogram from the previous section can serve to find potential obstacles. If any entry in the image distance histogram falls below a fixed distance threshold, an obstacle hypothesis is created. As pointed out in [6], robust computer vision algorithms should provide not only parameter estimates but also quantify their accuracy. Although we get the distance accuracy of an obstacle hypothesis by error propagation from tracking, this does not evaluate the probability of an obstacle's pure existence. This section describes obstacle detection by hypothesis testing resulting in a quality specified output.

Let d be the distance of an obstacle hypothesis drawn from the distance histogram. With the known camera translation in depth T_Z the transformation of the obstacle in image space using (5) is

$$\boldsymbol{x'} = V(\boldsymbol{x}) = \begin{pmatrix} \frac{d-T_Z}{d} & 0 \\ 0 & \frac{d-T_Z}{d} \end{pmatrix} \boldsymbol{x} \ . \tag{11}$$

The counter hypothesis of a free driveway with plane equation $e_2 = -Y_0$ will be transformed in image space using homogeneous coordinates according to

$$\boldsymbol{x'} = Q(\boldsymbol{x}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{T_Z}{Y_0} & 1 \end{bmatrix} \boldsymbol{x} \ . \tag{12}$$

Obviously this is only true for the ground plane up to the projected horizon. The hypothesis *no obstacle* above the horizon is set to be the identity (as this is equivalent with obstacles being infinitely distant).

Hypothesis testing. Let $F(\boldsymbol{x})$ and $G(\boldsymbol{x})$ be the intensity value for the initial image and the image after vehicle translation respectively. We assume a Gaussian distribution of the intensity values and fixed standard deviation, thus for an image transformation function f corresponding to an image region R we get

$$p_R(f) \propto e^{-|G-F|^2} \tag{13}$$

with $|G-F|^2$ being the *sum of squared differences* defined as

$$-\log(p_R(f)) = \sum_{\boldsymbol{x} \in R} \left(G(\boldsymbol{x'}) - F(\boldsymbol{x})\right)^2 \ . \tag{14}$$

p is maximal if the intensity value differences are minimal and vice versa. The scope of hypotheses verification is finding the transformation with higher probability. Let $p_1 = p_R(V)$ and $p_2 = p_R(Q)$, it then follows

$$p_1 > p_2 \quad \Leftrightarrow \quad \log p_1 > \log p_2 \quad \Leftrightarrow \quad \log p_1 - \log p_2 > 0 \ . \tag{15}$$

Therefore hypothesis testing boils down to calculating the SSD–difference for the two transformation assumptions. The absolute distance from zero represents the reliability of the result.

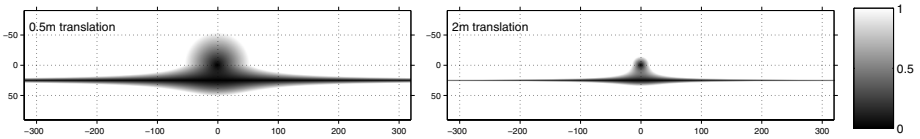

Fig. 2. Difference between flow field from planar ground (no flow above horizon) and obstacle in 30 m distance (planar motion parallax) for different camera translation in depth. The brightness corresponds to the length of the flow difference vector. Clearly, the challenge to distinguish between an obstacle and the planar ground near the focus of expansion by relative image motion becomes visible (camera focal length 689 pixel).

In practice, vehicle translation is not solely restricted to translation in T_Z. However, the motion parameters not included in the model can be compensated for the most part by estimating an extra region shift. Nevertheless, over larger time scales, hypothesis verification becomes more and more prone to errors due to lighting changes and the unmodelled motion.

Therefore, in the verification case, we restrict ourselves to time scales of 20 frames (in practice this corresponds to camera translations of more than 2 m). As indicated in Fig. 2 and by our experimental results, such a translation provides a sufficient difference between the two transformation assumptions and allows for reliable hypothesis testing.

4 Experimental Results

The proposed algorithm has been tested on real roads. The results are given in the following.

Comparison with distance from radar. A textured wall with a corner reflector behind the wall represents the obstacle. Due to the breadboard construction the distance measurement from radar is taken as the reference value and compared to distance from depth tracking. The results in Fig. 3 show, that distance measurement by scale is error-prune around the initial frame. This is not surprising as the scale factor is close to 1 and therefore division by $1 - s$ in (5) for distance computation leads to high inaccuracies. However, distance computation becomes quickly stable with greater vehicle translation. This clearly shows that distance estimation over large time scales is indispensable.

Obstacle detection performance. In the remaining part of this section we show three exemplary sequences from our test series on real roads to demonstrate hypotheses generation and testing. Figure 4 shows the first frame for each of these sequences. Notice that obstacle edges are present close to the focus of expansion what makes detection quite challenging.

The sequences are taken from a camera with 8.4 mm focal length (8.4 mm corresponds to 840 pixel) and 1.1 m camera height. The correlation threshold for replacing a depth tracker is set to 0.8. The threshold for hypothesis verification in the distance histogram is set to 70 m and restricted to the driving corridor.

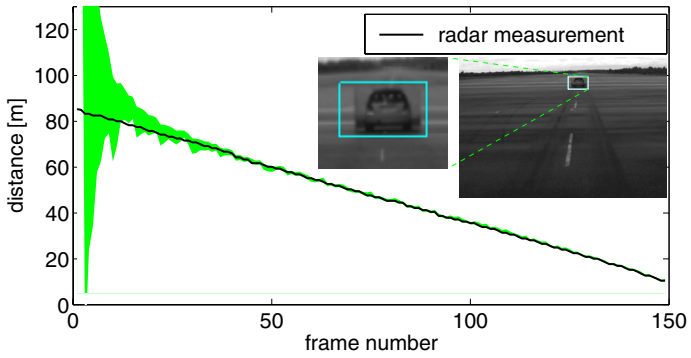

Fig. 3. Distance from radar compared to distance from region scale. Distance from scale plus and minus its standard deviation is represented by the gray area. The *thumbnail image was taken at 50 m obstacle distance*. The plot shows, that depth tracking allows to accurately estimate depth for distant obstacles.

These settings have been fixed in all three experiments showing the robustness of such parameters.

The first sequence shows an approach to a crash scene more than 100 m away. The vehicle speed is approximately 10 m/sec. The algorithm detects the stationary obstacle already at 69 m distance. Obstacle verification is error prone at such distances leading to a low value for the SSD difference. At 45 m distance (see Fig. 5) the obstacle is verified and horizontal obstacle boundaries are successfully detected such that a driver assistance system can safely evade this hazardous situation.

The middle set of images proves the reliable testing of obstacle hypotheses. The two trucks influence depth tracking and generate an obstacle hypothesis in the distance histogram for the free region amongst them (see Fig. 5 black line). Obstacle verification clearly rejects this hypothesis verifying a free corridor. As the vehicle approaches closer to the hazardous scene, the distance histogram adopts to the true observations picking up the bushes in the background as obstacles.

The right example deals with an obstacle boundary close to the focus of expansion. Note that the truck trailer has no texture making it hard for structure from motion algorithms to detect the vehicle in general. Nevertheless, the truck is detected and verified successfully at 67 m. Obstacle boundaries are close to ground truth. At such a large distance, the two trucks on the right influence hypothesis verification and lead to obstacle assumptions. Obviously the verification is correct but not for the given hypothesis distance. As the vehicle approaches the hazardous object in Fig. 5, the obstacle boundary is estimated precisely although it runs next to the focus of expansion. The image shows the truck still 50 m away. Experiments on several test sequences show, that robust object detection and verification can be reached with the proposed basic approach. Further quantitative studies on larger test data bases are the focus of ongoing research.

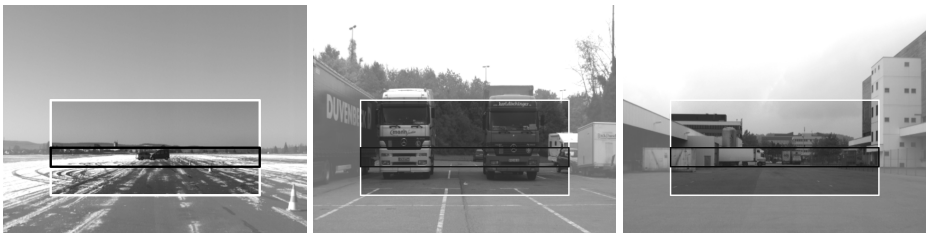

Fig. 4. Initial frames. The **white box** indicates the cropped image size shown in Fig. 5. The **black box** marks the area used for depth tracking.

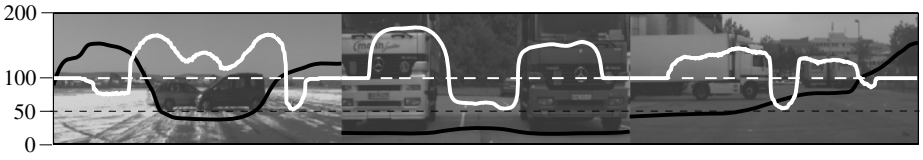

Fig. 5. Obstacle detection with **distance histogram** (*black, scale on the left*) and **hypotheses verification** (*white, logarithmic scale, obstacle verified if above dashed line*). The images show, that robust obstacle detection and verification is reached.

5 Conclusions

We have presented an algorithm for static obstacle detection in monocular image sequences. The scene depth is estimated by the change of region scale in image space; obstacle hypotheses are generated if depth estimation falls below a fixed threshold. To verify these hypotheses we check whether the observed transformation in image space is more likely to be generated by a static object or by the flat ground.

We implemented the algorithm on a Pentium IV with 3.2 GHz and achieved a framerate of 23 frames per second for the distance histogram calculation. The distance histogram and verification computation together run at approximately 13 frames per second. To the authors' knowledge, this is the fastest monocular motion–base obstacle detection algorithm in literature for obstacles close to the focus of expansion. The approach is easily applicable to other motion based distance measurements for obstacle detection and verification.

Further research will concentrate on speed gain. A wide range of algorithms in literature was proposed to speed up and stabilize tracking in image space. To name one possibility, pixel selection can be used to reduce computation time in region tracking. It is in the focus of ongoing studies to intelligently distribute the single regions used for depth tracking in image space. Although the described system works well in unknown environments we believe that optimizing the distribution and number of the tracked regions with respect to the currently observed scene will lead to even better results and less computation time. Moreover, we will investigate means to improve obstacle detection by method

of segmentation [4] and globally optimized optic flow estimation [2] forced into distinction of vertical and horizontal planes.

It also remains an open problem to detect moving obstacles in a monocular scenario. However, to pick up the threads given in the introduction, moving objects are well detected by common radar sensors therefore a sensor fusion combining measurements from an active radar and passive visual sensor is a promising field for further research.

References

1. M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta. Artificial vision in road vehicles. In *Proceedings of the IEEE*, volume 90, pages 1258–1271, 2002.
2. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3024 of *LNCS*, pages 25–36. Springer, May 2004.
3. A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 63(3):211–231, 2005.
4. D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005.
5. B. Deutsch, C. Gräßl, F. Bajramovic, and J. Denzler. A comparative evaluation of template and histogram based 2D tracking algorithms. In *DAGM-Symposium 2005*, pages 269–276, 2005.
6. W. Förstner. 10 pros and cons against performance characterization of vision algorithms. *Performance Characteristics of Vision Algorithms*, 1996.
7. U. Franke and A. Joos. Real-time stereo vision for urban traffic scene understanding. In *Proc. IEEE Conference on Intelligent Vehicles*, pages 273–278, 2000.
8. G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1025–1039, 1998.
9. Q. Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 390–397, 2003.
10. H. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. In *Proceedings of the Royal Society of London*, volume 208 of *Series B, Biological Sciences*, pages 385–397, July 1980.
11. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
12. C. Rabe, C. Volmer, and U. Franke. Kalman filter based detection of obstacles and lane boundary. In *Autonome Mobile Systeme*, volume 19, pages 51–58, 2005.
13. F. Stein. Efficient computation of optical flow using the census transform. In *DAGM04*, pages 79–86, 2004.
14. Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection using optical sensors: A review. In *IEEE International Conference on Intelligent Transportation Systems*, volume 6, pages 125 – 137, 2004.