Moving Object Segmentation Using Optical Flow and Depth Information

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

[1] Environment Perception Group, Daimler AG, Sindelfingen, Germany
[2] *.enpeda..* Project, The University of Auckland, New Zealand
jens.klappstein@daimler.com, t.vaudrey@auckland.ac.nz, clemens.rabe@daimler.com, andreas.wedel@daimler.com

Abstract. This paper discusses the detection of moving objects (being a crucial part of driver assistance systems) using monocular or stereoscopic computer vision. In both cases, object detection is based on motion analysis of individually tracked image points (optical flow), providing a motion metric which corresponds to the likelihood that the tracked point is moving. Based on this metric, points are segmented into objects by employing a globally optimal graph-cut algorithm. Both approaches are comparatively evaluated using real-world vehicle image sequences.

Keywords: Motion detection, optical flow, stereo, segmentation.

1 Introduction

Kinesthesia, the sensation or perception of motion, is an 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 model such perception of motion using computer vision.

Detecting moving objects is a major issue for driver assistance and road safety. The detection of moving traffic participants is an important step toward attention-based environment perception. In this paper, we investigate 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 the stereo system. However, a stereo system yields direct range measurement estimates, 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 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.

T. Wada, F. Huang, and S. Lin (Eds.): PSIVT 2009, LNCS 5414, pp. 611–623, 2009.

The motion of the ego-vehicle greatly complicates the problem of motion detection because simple background subtraction of successive images yields no result. This paper presents and investigates techniques to distinguish between stationary and non-stationary points. They are based on tracking feature points in sequential images. As a result, feature points on independently moving objects are detected as moving. These features, however, are sparse and do not characterize the whole image. 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 [6] to reject outliers and to find image regions with an accumulation of image features lying on moving objects. The proposed algorithm is able to find both rigid objects such as cars and non-rigid objects such as moving pedestrians.

The paper is organized as follows. Section 2 investigates the motion analysis techniques. Section 3 deals with the segmentation of the objects. In the result of Section 4 different scenarios are presented, 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. A concluding section on future work and obtained insights closes this paper.

2 Motion Analysis

The detection of moving objects is based on motion analysis of individual tracked image features, using the KLT tracker [20]. 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, 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) [7] or based on optical flow [2,14].

There is a fundamental difference between the monocular and the stereoscopic reconstruction. Moving points cannot be correctly reconstructed by monocular vision, except in special situations, such as using trajectory triangulation [3]. The erroneous reconstruction of moving points can be identified as erroneous if the constraints for a static 3D point are violated. The monocular detection of moving points relies on this fact. In Section 2.2 the constraints for static 3D points are defined and an algorithm, evaluating them, is discussed.

In the case of stereoscopic vision, moving points are reconstructed correctly for every stereo pair using [4,19], by considering reconstructed 3D points over time and integrating the results. This allows to calculate 3D velocity as well, referred to as 6D-Vision in [9]. (The 3D velocity of a point indicates whether the point is moving or not.) The 6D-Vision approach is discussed further in Section 2.1.

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. See Figure 1.

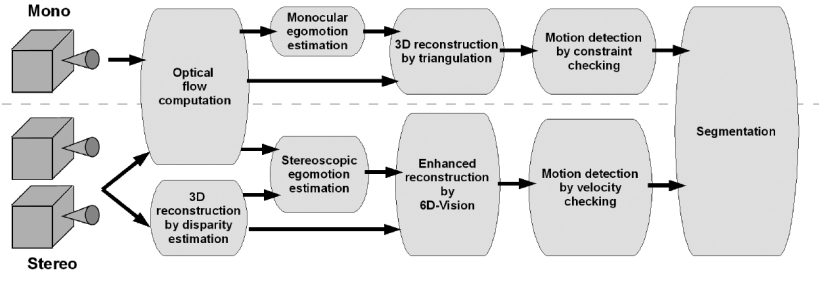

Fig. 1. Work flow for monocular or binocular motion segmentation

2.1 Stereo Vision

We start with the stereo case. The displacement of image features between the left and the right image (the disparity) is inversely related to the depth of the corresponding 3D point. This information is accumulated in an evidence-grid, similar to approaches such as in [18]. We refer to it as the *bird-view map*. This map is usually segmented, and detected objects are tracked over time in order to obtain their motion. The major disadvantage of this standard approach is that the performance of the detection depends highly on the correctness of the segmentation. Especially moving objects in front of stationary ones are often merged and therefore not detected. This causes dangerous misinterpretations and requires more powerful solutions.

In order to obtain motion information directly from the images, the optical flow has to be analysed. It gives the displacement of image features in two consecutive images of one camera, and depends on the motion of the observer as well as the motion of the corresponding 3D point. By combining the left and right optical flow fields [22], or the optical flow field of one camera with the stereo information [1,13], the 3D scene motion relative to the observer is reconstructed. Inconsistencies in scene motion fields are then detected as independently moving objects.

Direct optical flow analysis provides fast detection results, but is limited with respect to robustness and accuracy due to the immanent measurement noise. To get more reliable results, an integration of the observations over time is necessary. The Kalman filter solves this in an elegant manner. Each measurement is used to improve the current estimate of the systems state. In addition, the Kalman filter propagates the covariances of the estimated state over time, which allows the application of stochastical methods.

The core algorithm of the stereo vision system presented here follows the principle of fusing optical flow and stereo information given in [9]. The basic

idea is to track points with depth estimated from stereo vision over two or more consecutive frames and to fuse the spatial and temporal information using Kalman filters. The result is an improved accuracy of the 3D-position and an estimation of the 3D-motion of the considered point at the same time. Taking into account the motion information, the above mentioned segmentation problem can be solved much more easily and robustly. In addition, using the 3D-motion information a prediction of the objects movement is possible. This allows a driver assistance system to warn and react to potential collisions in time.

The fusion implies the knowledge of the ego-motion. In our system we compute it from image points found to be stationary using a Kalman filter based approach described in [19]. This allows a fast calculation using all information already acquired by the system including inertial sensor data. We briefly discuss the proposed Kalman filter-based fusion of optical flow and stereo information.

System Model. We use a left handed coordinate system with the origin on the road. This coordinate system is fixed to the car, so that all estimated positions are given in the coordinate system of the moving observer. The lateral x-axis points to the left, the height axis y points upwards and the z-axis represents the distance of a point straight ahead. The camera is at $(x, y, z)^T = (0, height, 0)^T$ looking along the positive z-direction.

Let $\mathbf{p}_k = (x, y, z)^T$ be an observed 3D point and $\mathbf{v}_k = (\dot{x}, \dot{y}, \dot{z})^T$ its associated velocity vector at the time step k. Assuming a constant motion during the time interval Δt the 3D position at the time step $k + 1$ is given by

$$\mathbf{p}_{k+1} = \mathbf{R}\mathbf{p}_k + \mathbf{t} + \Delta t \mathbf{R}\mathbf{v}_k \tag{1}$$

Here the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ give the motion of the scene, that is the inverse ego-motion. The new velocity vector of the observed point is described by

$$\mathbf{v}_{k+1} = \mathbf{R}\mathbf{v}_k \tag{2}$$

Combining the location $\mathbf{p}_k$ and the velocity $\mathbf{v}_k$ in the 6D state vector $\mathbf{s}_k = (x, y, z, \dot{x}, \dot{y}, \dot{z})^T$, the time-discrete linear system model is given by

$$\mathbf{s}_k = \mathbf{A}_k \mathbf{s}_{k-1} + \mathbf{b}_k + \omega \tag{3}$$

with state transition matrix

$$\mathbf{A}_k = \begin{bmatrix} \mathbf{R}_k & \mathbf{R}_k \Delta t \\ 0 & \mathbf{R}_k \end{bmatrix} \tag{4}$$

control vector $\mathbf{b}_k = [\mathbf{t}_k \ , \ 0 \ , \ 0 \ , \ 0]^\top$ and noise term ω (assumed to be Gaussian white noise with covariance matrix $\mathbf{Q}$).

Measurement Model. We measure image coordinates u and v of a tracked feature and the disparity d delivered by stereo vision, working on rectified images.

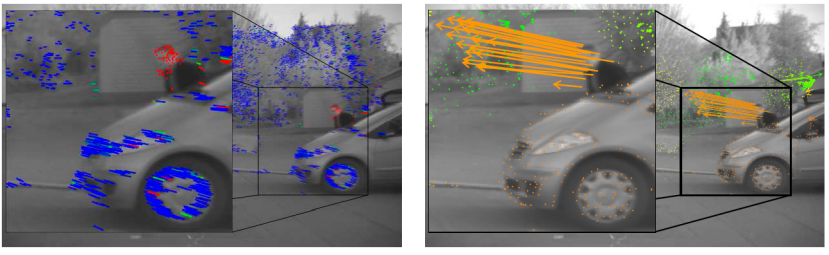

Fig. 2. Monocular (left) and stereo (right) motion analysis for a moving pedestrian appearing behind a stationary vehicle

Assuming a pinhole-type camera, the non-linear measurement equation for a point given in the camera coordinate system is as follows:

$$\mathbf{z} = \begin{bmatrix} u \\ v \\ d \end{bmatrix} = \frac{1}{z} \begin{bmatrix} x f_u \\ y f_v \\ b f_u \end{bmatrix} + \nu \tag{5}$$

with focal lengths f_u and f_v (in pixels), and baseline b (in metres). The noise term ν is assumed to be Gaussian white noise with covariance matrix $\mathbf{S}$.

As the measurement equations are non-linear, we have to apply the Extended Kalman Filter (EKF), which is known to be sensitive to wrong initializations. To improve the Kalman filter's rate of convergence, a multi-filter system is used. It consists of multiple differently initialized and parameterized Kalman filters running in parallel. By analysing the innovation of each filter the best matching estimation is chosen. A detailed description of this approach is given in [9].

The result of a 6D-vision algorithm is illustrated in Figure 2. Images are 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. [The colour encoding on the left corresponds to the motion metric (blue: 0 px, red: 2 px); the arrows in the 6-D vision image on the right point to the estimated 3D position in 0.5 s, reprojected into the current image, where the colour encoding corresponds to estimated depth (close = red, far = green).]

Scalar Motion Metric for Moving Object Detection. The monocular or binocular algorithm estimates the position and velocity of independent image features. Due to systematical measurement errors, induced for example by occlusion effects or repetitive patterns, single points may be incorrect and a driver assistance system has to deal with them accordingly. This is accomplished by combining the estimates of multiple image features belonging to the same object, which in turn requires an object segmentation.

In order to obtain the boundaries of all moving objects, we are first interested in the question whether a 3D point is static or moving. As 6D vision estimates the 3D velocity vector, we reduce this information to absolute velocity.

2.2 Monocular Vision

In this case we forbear from the usage of the second camera. This affects the approach for the detection of moving points, since moving 3D points cannot be reconstructed with one camera only. A reconstruction of a moving 3D point is erroneous. The point is detected as moving if its reconstruction is identified as erroneous. To this end, one checks whether the reconstructed 3D point fulfills the constraints of a static 3D point. These constraints are as follows:

Epipolar Constraint: This constraint expresses that viewing rays of a static 3D point (lines joining 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 knowledge about normal vectors of the road surface and the camera distance to the road surface. These entities are estimated exploiting the optical flow on the road [16].

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*.

Existing motion detection schemes exploit a subset of the above constraints either directly or indirectly. A popular scheme is the angle criterion [8,24] which uses the direction of optical flow vectors. When moving purely translational toward the scene, all flow vectors are parallel to the corresponding epipolar lines and point away from the epipole (focus of expansion). This holds true for the entire static scene. If a measured optical flow vector deviate from this expected flow direction (i.e., if the angle between measured and expected direction is not zero), the corresponding 3D point is moving. This angle criterion indirectly exploits the epipolar and the positive depth constraint.

Another popular scheme is the planar motion parallax. It is defined as the deviation of the measured optical flow from the expected flow on the road plane. For correspondences violating the positive height constraint, the parallax vector points toward the epipole since the measured flow is shorter than expected. [5,10] evaluate the planar motion parallax. A scheme exploiting the trifocal constraint is presented in [11]. It not only detects moving points but also clusters them. However, the computational burden is high.

We now develop an algorithm evaluating all available constraints quantitatively. In the work flow diagram (Figure 1), reconstruction and detection are 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.

The algorithm provides a motion metric measuring to which extent the constraints are violated. It is correlated to the likelihood that the point is moving (i.e., higher values indicate a higher probability).

The motion metric is developed in two steps. First, the two-view constraints are evaluated taking view one and two into account. Afterward, the trifocal constraint is evaluated using the third view.

Two-View Constraints. A motion metric combining the two-view constraints has been introduced in [15]. It measures the distance of a given image point in the first view to the closest point fulfilling all constraints (epipolar, positive depth, and positive height constraint). For the ease of computational complexity image points in the second view are considered noise free. We use this metric but swap the roles of the views [i.e., we compute the error (distance) in the second view].

This is illustrated in Figure 3. We first consider the correspondence $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$ in views one and two. The closest point to $\mathbf{x}_2$, fulfilling the two-view constraints, is $\mathbf{x}_{f2}$. It lies on the epipolar line $\mathbf{l}_2 = \mathbf{F}\mathbf{x}_1$ with $\mathbf{F}$ as the fundamental matrix. Note that the vector from $\mathbf{x}_{f2}$ to $\mathbf{x}_2$ is not necessarily perpendicular to $\mathbf{l}_2$. The distance d_2 between $\mathbf{x}_{f2}$ and $\mathbf{x}_2$ is the error arising from the first two views. For the computation of d_2 see [15].

Three-View Constraint. We now add the third view and consider the correspondence $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2 \leftrightarrow \mathbf{x}_3$. As the point $\mathbf{x}_{f2}$ is defined such that it fulfills the two-view constraints, the reconstructed 3D point, arising from the triangulation of the points $\mathbf{F}\mathbf{x}_1$ and $\mathbf{x}_{f2}$, constitutes 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

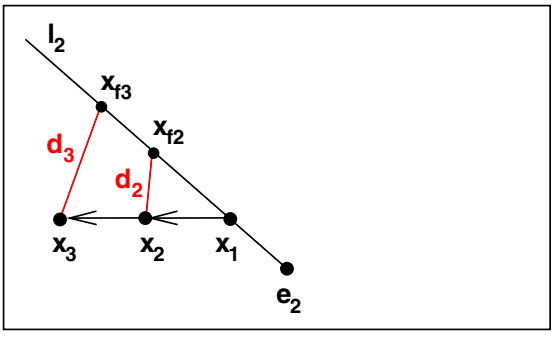

Fig. 3. Monocular motion metric. The image of the second view is shown. The camera moves along its optical axis observing a lateral moving point $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2 \leftrightarrow \mathbf{x}_3$. The closest point to $\mathbf{x}_2$ fulfilling the two-view constraints is $\mathbf{x}_{f2}$. 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. The overall error is $d = d_2 + d_3$. Note: in general, $\mathbf{x}_1$ and $\mathbf{x}_{f3}$ do not lie on the epipolar line $\mathbf{l}_2$.

is a distance d_3 (Figure 3) between them which we call *trifocal error*. $\mathbf{x}_{f3}$ is computed via the point-point-point transfer using the trifocal tensor [12]. This approach avoids the explicit triangulation of image points $\mathbf{F}\mathbf{x}_1$ and $\mathbf{x}_2$.

The final motion metric, combining the two-view constraints and the three-view constraint, is $d = d_2 + d_3$. It measures the minimal required displacement in pixel necessary to change a given correspondence into a correspondence belonging to a valid static 3D point. See Figure 3 for an example of the final motion metric. To be exact, d is a pseudo-metric only since we may have $d_2(\mathbf{x}_2, \mathbf{x}_{f2}) = 0$ for distinct points $\mathbf{x}_2 \neq \mathbf{x}_{f2}$.

3 Segmentation

In order to derive objects from individual 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. Objects could be found by calculating a dense flow field in which each image pixel yields an error value. A subsequent connected components analysis yields objects. Such dense flow calculation is computationally expensive and the result needs to be post-processed to distinguish between noise and moving objects.

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 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 a global optimal graph-cut algorithm [6]. Further speed up techniques for flow vector segmentation can be achieved using a Multi-Resolution Graph Cut [21].

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

$$d(\mathbf{x}) < \sigma \Rightarrow e(s, \mathbf{x}) = \sigma - d(\mathbf{x}) \tag{6}$$

$$d(\mathbf{x}) > \sigma \Rightarrow e(\mathbf{x}, t) = \min(d(\mathbf{x}) - \sigma, C_{max}) \ . \tag{7}$$

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, adjacent image pixels (here only 4-adjacency is taken into account) are connected by

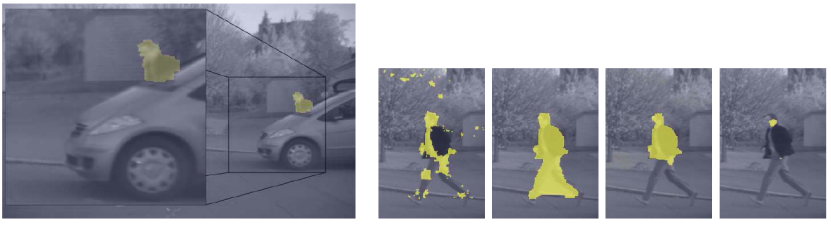

Fig. 4. The images on the left show the segmentation for a moving pedestrian appearing behind a stationary vehicle. Outliers are rejected and the segmentation border is accurate. The four images on the right show the influence of the edge costs on the segmentation result (later in the sequence). While small edge costs result in segments with only a few pixels (left), high edge costs result in small regions (such that the number of cut edges is minimized, right). From left to right: $C_e = \{1.5, 50, 500, 1000\}$.

edges. The costs of these edges depend on the grey-value difference of its two end points. The cost values are defined by

$$e(\mathbf{x}, \mathbf{y}) = \frac{C_e}{\|I(\mathbf{x}) - I(\mathbf{y})\| + \varepsilon} \tag{8}$$

where C_e is a constant scaling 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 (8) 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 4. If the sum of all edges of a pixel is larger than C_{max}, the pixel will not be cut. Therefore we set $e(\mathbf{x}, \mathbf{y}) = 0.5\ C_{max}$, for all tracked points $\mathbf{x}$.

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 $e(s, \mathbf{x}) = C_{BG}$, from every node $\mathbf{x}$ to the source s.

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 [6,23]. 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; see, for example, [21] for a diagram.

4 Experimental Results

This section applies our motion analysis and segmentation to real imagery. We use the same set of features for monocular and binocular motion analysis. The first example in Figure 4 (left) 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 (compared with Figure 2, right and left). The monocular and the stereo approach yield exactly the same segmentation result for the lateral moving pedestrian.

Figure 5 shows a traffic scene with a crossing car and a preceding car in 31 m distance. 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, see [17].

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

(a) (b) (c) (d)

Fig. 5. Detection and segmentation results of a crossing and a preceding object. Monocular vision (a,b) performs similar to stereoscopic vision (c,d). Tracked image features are shown on the left of each pair; they are color encoded according to the corresponding motion metric. For monocular vision (a), the range is from 0 px (blue) to 7 px (red); for stereoscopic vision (b), the range is from 0 m/s (blue) to 7 m/s (red).

(a) (b) (c) (d)

Fig. 6. Detection and segmentation results of preceding and oncoming objects. Monocular vision (a,b) is only able to detect the lower parts of the preceding objects; the oncoming object is not detected at all. Stereoscopic vision (c,d) does not suffer from these limitations. (Color encoding as in Figure 5.)

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 6. 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.

5 Conclusion

This paper investigates a monocular and a stereo 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 motion metrics to traffic scenes captured by a camera installed in a vehicle. Using image segmentation based on the investigated motion metrics we were able to detect and segment other moving traffic participants. On average, the stereo approach outperforms the monocular approach in terms of accuracy. However, there is a higher computational cost for the computation of both stereo and KLT tracks. Image sequences on highways and urban scenarios using the same parameter sets demonstrate the practicality of this novel approach to machine sensing of motion.

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. Argyros, A.A., Lourakis, M.I., Trahanias, P.E., Orphanoudakis, S.C.: Qualitative detection of 3d motion discontinuities. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems, vol. 3, pp. 1630–1637 (1996)

2. Armangué, X., Araújo, H., Salvi, J.: Differential epipolar constraint in mobile robot egomotion estimation. In: Proc. IEEE Int. Conf. Pattern Recognition, pp. 599–602 (2002)
3. Avidan, S., Shashua, A.: Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. IEEE Trans. Pattern Analysis Machine Intelligence 22, 348–357 (2000)
4. Badino, H.: A robust approach for ego-motion estimation using a mobile stereo platform. In: Proc. Int. Workshop Complex Motion (2004)
5. Baehring, D., Simon, S., Niehsen, W., Stiller, C.: Detection of close cut-in and overtaking vehicles for driver assistance based on planar parallax. In: Proc. IEEE Intelligent Vehicles Symposium (2005)
6. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/Max-flow algorithms for energy minimization in vision. In: Figueiredo, M., Zerubia, J., Jain, A.K. (eds.) EMMCVPR 2001. LNCS, vol. 2134, pp. 359–374. Springer, Heidelberg (2001)
7. Chalimbaud, P., Berry, F., Marmoiton, F., Alizon, S.: Design of a hybrid visuo-inertial smart sensor. In: Proc. Workshop Integration Vision Inertial Sensors (in conjunction with IEEE Int. Conf. Robotics Automation) (2005)
8. Clauss, M., Bayerl, P., Neumann, H.: Segmentation of independently moving objects using a maximum-likelihood principle. In: Proc. Autonome Mobile Systeme (2005)
9. Franke, U., Rabe, C., Badino, H., Gehrig, S.: 6D-vision: Fusion of stereo and motion for robust environment perception. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 216–223. Springer, Heidelberg (2005)
10. Giachetti, A., Campani, M., Torre, V.: The use of optical flow for road navigation. IEEE Trans. Robotics and Automation 14, 34–48 (1998)
11. Hartley, R., Vidal, R.: The multibody trifocal tensor: Motion segmentation from 3 perspective views. In: Proc. IEEE Int. Conf. Computer Vision Pattern Recognition (2004)
12. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2003)
13. Heinrich, S.: Fast obstacle detection using flow/depth constraint. In: Proc. IEEE Intelligent Vehicles Symposium, vol. 2, pp. 658–665 (2002)
14. Ke, Q., Kanade, T.: Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. I–390– I–397 (2003)
15. Klappstein, J., Stein, F., Franke, U.: Monocular motion detection using spatial constraints in a unified manner. In: IEEE Intelligent Vehicles Symposium, IV (2006)
16. Klappstein, J., Stein, F., Franke, U.: Applying Kalman filtering to road homography estimation. In: Proc. Workshop Planning Perception Navigation Intelligent Vehicles (in conjunction with IEEE Int. Conf. Robotics Automation) (2007)
17. Klappstein, J., Stein, F., Franke, U.: Detectability of moving objects using correspondences over two and three frames. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 112–121. Springer, Heidelberg (2007)
18. Martin, M.C., Moravec, H.: Robot evidence grids. Technical Report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University (1996)
19. Rabe, C., Franke, U., Gehrig, S.: Fast detection of moving objects in complex scenarios. In: Proc. IEEE Intelligent Vehicles Symposium, pp. 398–403 (2007)
20. Tomasi, C., Kanade, T.: Detection and tracking of point features. Carnegie Mellon University, Technical Report CMU-CS-91-132 (1991)

21. Vaudrey, T., Gruber, D., Wedel, A., Klappstein, J.: Space-time multi-resolution banded graph-cut for fast segmentation. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 203–213. Springer, Heidelberg (2008)
22. Waxman, A.M., Duncan, J.H.: Binocular image flows: steps toward stereo-motion fusion. IEEE Trans. Pattern Analysis Machine Intelligence 8, 715–729 (1986)
23. Wedel, A., Schoenemann, T., Brox, T., Cremers, D.: Warpcut - fast obstacle segmentation in monocular video. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 264–273. Springer, Heidelberg (2007)
24. Woelk, F., Koch, R.: Fast monocular bayesian detection of independently moving objects by a moving observer. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 27–35. Springer, Heidelberg (2004)