2013 IEEE Intelligent Vehicles Symposium (IV)
June 23-26, 2013, Gold Coast, Australia

Spatio-Temporal Prediction of Collision Candidates for Static and Dynamic Objects in Monocular Image Sequences

Alexander Schaub and Darius Burschka

Abstract— **This paper presents a novel approach for reactive obstacle avoidance for static and dynamic objects using monocular image sequences. A sparse motion field is calculated by tracking point features using the Kanade-Lucas-Tomasi method. The rotational component of this sparse optical flow due to ego motion of the camera is compensated using motion parameters estimated directly from the images. A robust method for detection of static and dynamic objects in the scene is applied to identify collision candidates. The approach operates entirely in the image space of a monocular camera and does not require any extrinsic information about the configuration of the sensor or speed of the camera. The system prioritizes the detected collision candidates by their time to collision. Additionally, the spatial distribution of the candidates is calculated for non-degenerated conditions.**

We present the mathematical framework and the experimental validation of the suggested approach on simulated and real-world data.

I. INTRODUCTION

An elementary task in autonomous vehicles from small flying robots to large robot cars [1] is collision avoidance. This task requires a fast response time and robustness in detection even in cases where calibration of the system changed due to external factors like vibrations. Because of these requirements, this task usually does not rely on data abstraction but tries to couple the response directly to a sensor input like, e.g., a bumper response. It is desirable to detect collision ahead of time using the existing sensors on the system. The sensors can be subdivided in active and passive sensors. Active sensors rely on active illumination of the scene and are usually able to detect directly the 3D position and motion parameters of obstacles. The active illumination results in a higher power consumption of the sensor, limited sensing range, and a possible cross-talk between multiple sensors operating in the same environment. Examples of such active sensors in automotive area are radar and lidar systems [2], [3]. On the other hand, passive sensors rely on the ambient light of the scene. They require some additional processing of the sensed light to recover the 3D information. A typical sensor system in this domain is a binocular camera system [4], in which the information of two cameras is combined to 3D data using extrinsic calibration parameters of the system. This additional step makes the system usually more computationally expensive and sensitive to possible changes in the calibration.

A. Schaub is with the Institute of System Dynamics and Control at the Robotics and Mechatronics Center, German Aerospace Center, 82234 Wessling, Germany alexander.schaub@dlr.de

D. Burschka is with the Department of Computer Science, Technische Universität München, 85748 Garching bei München, Germany burschka@tum.de

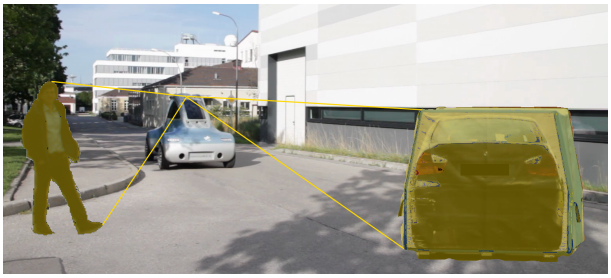

Fig. 1. Optical flow in monocular image sequences is used to detect and avoid obstacles on the RoMo vehicle.

Brooks introduced a task based hierarchical scheme for autonomous robots in [5] in 1986. The sensor information can be coupled to the actuators at different abstaction levels. While for planning tasks an abstraction of the raw sensor information to three-dimensional information with object labels is desireable, the collision task at hand needs to react fast and work reliable even under difficult conditions. We decided to derive the collision information directly from the raw sensor data. We address the problem of the robustness and complexity of passive approaches by computing the collision data directly from the image data of a monocular sequence. This significantly reduces the dependency on the failure-prone extrinsic calibration of the system, which may result wrong collision estimates. The proposed system (Fig. 1) is used on the robot car RoMo from the German Aerospace Center.

Algorithms interpreting visual motion for obstacle avoidance are definitely settled at the lowest level of the autonomy scheme as sensor data leads directly to actions. A comparison of different methods for measuring the time to contact can be found in [6]. All algorithms have in common that a single camera is the only required sensor.

Optical flow is the motion of image points over succeeding images. Popular gradient based techniques were introduced by Lucas and Kanade [7] and Horn and Schunck [8] in 1981. A first comparison and performance evaluation of those early optical flow techniques can be found in [9]. In recent years there have been a number of improvements to the known optical flow algorithm in order to deal with problems like the preservation of discontinuities [10], occlusions [11], and respecting motion boundaries [12]. For an increased robustness in urban areas the fusion of the optical flow from a standard camera with the flow from an infrared camera is suggested in [13]. An alternative to dense optical flow are feature based techniques. They need less computational time

978-1-4673-2754-1/13/$31.00 ©2013 IEEE

and have no aperture problem [14].

Nelson proposed a method in [15] and verified it with basic obstacle detection tests to make use of the flow field divergence for obstacle avoidance. Since then, different techniques have been developed to interpret the optical flow to provide collision free movements of autonomous mobile robots on a low-level.

There exist multiple approaches how to calculate the depth information from the optical flow. The results are similar to stereo vision, but with the use of only one camera. The scheme described in [16] determines the direction of movement based on a depth histogram. In [17], a depth map generated by optical flow is compared to maps from sonar and laser data. The depths from motion algorithm perform slightly worse than the lidar results. In [18] the idea of estimating obstacles from the current flow field by a neural network is proposed.

Early real time approaches were balancing peripheral flows, while perceiving centrally located obstacles by calculating the time to contact for image motions around the center. A mobile robot in [19] was able to move in a static environment collision free for up to 20 minutes with an average speed of 30 cm/s. Simple strategies were necessary due to very limited computational power. Some approaches use a higher resolution at the center region than at peripheral regions [20][21]. Hence, the input data is reduced without the loss of necessary information.

Our approach is developed for ground based vehicles equipped with standard PCs. Hence, more information from the optical flow can be extracted to execute new reactive strategies.

Typical problems when using optical flow for collision avoidance are for example camera rotations between successive frames [19], large objects like walls, and dynamic obstacles. One possible solution for the rotation compensation is the use of an additional sensor like an inertial measurement unit [22]. The method proposed in this paper eliminates the rotation of the camera by a Zinf ego-motion estimation [23]. Large objects do not have to be treated specially by our segmentation and we explicitly consider moving obstacles comparing their measured flow to the flow expected from ego-motion.

A remaining shortcoming of the optical flow is the requirement of a minimum motion. The signal to noise ratio gets useless at slow motions. The combination of the optical flow with stereo vision [24] solves this problem. While a binocular system due to its limited distance between the cameras is not able to reconstruct correctly information at larger distances, it is a perfect system to navigate in parking scenarios. At higher speeds, when larger sensing distance is required, optical flow approaches can seemlessly provide the necessary information. This requires a pair of calibrated cameras. Very good results in detecting obstacles and their respective speeds can be achieved in this way [4].

The remainder of this paper is organized as follows: In Section 2, the calculation of the used optical flow and the representation of the gained information is described. Section 3 shows the calculation of the time to collision and proposes a reactive strategy based on that data. First results derived from simulation and testing are reported in Section 4. We conclude in Section 5 with a summary of the achieved results and sketch our future work in that field.

II. Clustering of Object Candidates in the Optical Flow

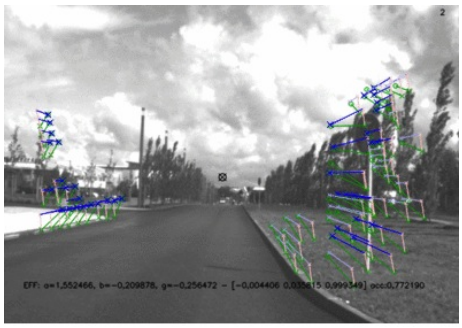

Fig. 2. The green lines in the image show the combined motion of the points in the image due to rotation and translation of the camera, the blue lines is the resulting image motion, after the rotation of the camera is compensated in the vector field.

Calculation of a dense optical flow results in a large computational effort as the relative motion of every pixel in the image is estimated. This calculation is based on uniqueness of the pixel neighborhood in a local area that allows a correct estimation of its image motion. The flow determination in parts of the image with weak or no trackable features is not unique and results often in errors. Therefore, we focus on sparse optical flow in this work. Only points with multiple gradient directions in the local area are tracked by the Kanade-Lucas-Tomasi method [25]. We are interested in invariant points representing object boundaries or texture on the surfaces. Shadow casts are also useful candidates since they are static in the time span of the measurement. The possibility of omitting poorly textured objects exists, but most of the objects in the real world have sufficient contrasts. The resulting motion in the image position is a result of a rotation and translation of the camera relative to the imaged surface (Fig. 2).

After the flow vectors are calculated a rotation compensation is executed. The flow vectors should point now to the epipole E, which is the direction of the camera's motion (see Figure 2 and 3). Image points of objects that are located in the real world near to the camera have longer vectors than distant points. In case of dynamic scenes, we observe multiple relative camera to surface motions with different epipoles. Static obstacles are characterized by vectors pointing to the epipole of the background but having a larger magnitude than expected. An object above the ground is closer to the camera than the ground plane itself and therefore creates a longer flow vector.

In order to outline regions with similar attributes a clustering can be run on the flow field [26]. In this work regions of distorted vectors are clustered by a local search. An

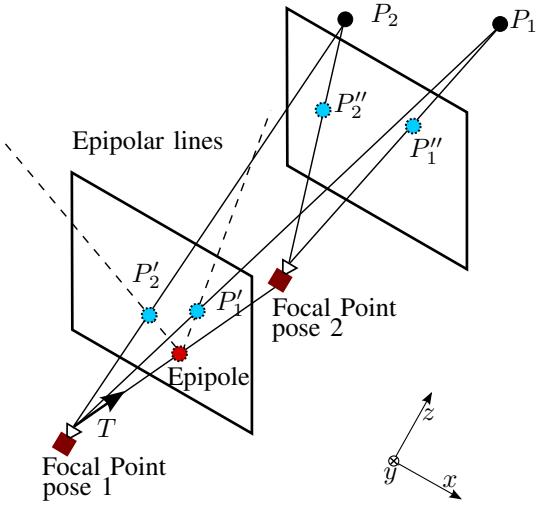

Fig. 3. The intersection point E of the translational optical flow defines the direction of the camera motion.

object is characterized by vectors with a common intersection respectively the object's epipole E_o. A description of the calculation can be found in [23]. The difference here is that the epipole determination is executed iteratively until the clustering of the entire vector field is finished.

The distinction if an object is dynamic and on a collision course can be made from the relative position of the epipole as will be explained in the next section. A bounding box is laid around the feature points of a cluster. If the epipole of that cluster lies inside the bounding box, a collision is imminent.

At least 8 vectors are necessary to calculate a rotation of an object using the standard 8-point algorithm based on the decomposition of the Essential Matrix, but only two vectors are sufficient to determine a relative translation of the object. For objects having less than 8 corresponding points, we neglect the rotation (which cannot be estimated correctly for such small objects anyway) and calculate just the translation parameters.

A special case occurs if the obstacle is moving in the same direction as the camera so that the epipole of the obstacle E_o and the global epipole E are not distinguishable from each other. Flow vectors belonging to the dynamic obstacle can only be specified by their length.

By utilizing the epipole based clustering it is now possible to calculate the time to collisions for the different obstacles. In contrast to former methods [6] [17], we make no assumptions regarding the motion or depth. In some cases, it is also possible to determine the three-dimensional motion vector (from one camera) by utilizing structure from motion principles.

III. Estimation of the Time to Collision

After objects are extracted from the optical flow and their epipoles are determined, the respective time to collision for every object is calculated. Here we distinguish between objects with an arbitrary trajectory and a special case where an object has a crossing trajectory. A general discussion of the detection limits of moving objects using correspondences over succeeding frames can be found in [27].

A. Objects with a crossing trajectory

If the epipole of an optical flow object is not in a defined region around the global epipole, the time to collision can be calculated concurrently with additional information about the depths in the scene. For a colliding object, its epipole is seen under a constant angle α in all images. An analogy from sailing is used here. If a point on the other ship is seen under a constant angle, a collision is imminent. Figure 4 illustrates this situation, whereas α is the constant angle to the Epipole E_o of a dynamic obstacle. The real motion direction $\vec{v_o}$ cannot be observed directly, as only the motion direction parallel to the image plane $\vec{v_o^x}$ is observable. Using the geometric principle of the constant angle α the following formula can be derived:

$$\vec{v_o^x} = \tan\alpha \cdot \vec{v_c} \tag{1}$$

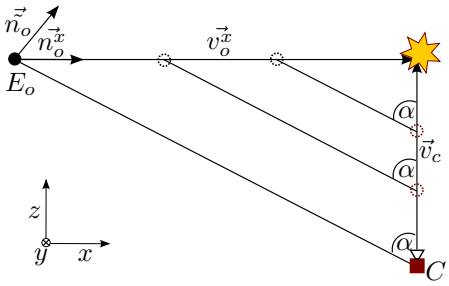

Fig. 4. Epipole seen under a constant angle

$\vec{v_o^x}$ is the horizontal motion between two sequential images and $\vec{v_c}$ is the velocity of the camera respectively of the vehicle.
If the vehicle's speed is unknown, we can set $\vec{v_c} = 1$. This artifical unit represents 'camera speed'. In that case, all speeds and, therefore, all distances will be known relatively up to a scale. This is valid because we calculate the velocity profiles for just 2 image frames and do not propagate the information over the sequence.

Since the camera and the colliding obstacle are both moving, a relative motion $\vec{v_o}$ is observed by the camera:

$$\vec{T} = \vec{v_o} = -\vec{v_c} + \vec{v_o} \tag{2}$$

Considering the camera as stationary, the epipole will approach the camera with $\vec{v_o}$ along a vector with the same direction as E_o. The length λ is the distance to the object in the world (Fig. 5). The magnitude of $\vec{T}$ is the distance covered between two sequential images and hence:

$$\Delta\lambda = |\vec{T}|$$

The angle α between the optical center of the camera and the line of epipoles can be estimated from the image projection

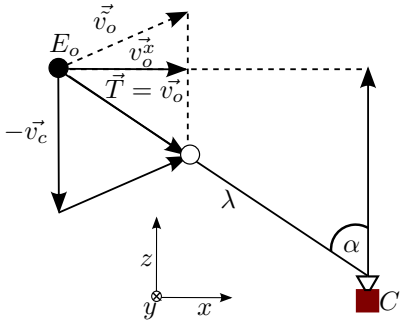

Fig. 5. Epipole seen under a constant angle

of the epipole. Hence, the following equation can be solved directly:

$$\Delta\lambda = \frac{\vec{v_o^x}}{\sin\alpha} \tag{3}$$

At this point, the absolute distance λ to the epipole E_o is unknown yet. This information can be derived by applying a structure from motion approach to points of the moving object in a distance from the epipole. The movement of all features is known from the optical flow. Their position in at least two sequential scenes can be used to estimate the λ. The translation $\vec{T_2^1}$ of a feature from point P_1 to point P_2 is known from (2), as we assume rigid bodies. The direction vectors $\vec{n_1}$ and $\vec{n_2}$ can be extracted from the projections in respective images. Combining this information, the structure from motion equation (4) can be solved to get the distance $\lambda_{1,2}$ to a point in images 1 and 2.

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \vec{n_2} & -\vec{n_1} \end{pmatrix}^+ \cdot \vec{T_2^1} \tag{4}$$

The $^+$ stands for the Moore–Penrose pseudoinverse.
This equation degenerates for the epipole, as E_o remains stationary in the image plane and n_1 and n_2 are identical.Hence, the distance to the collision point can not be calculated directly. In order to get a more robust result the depth of all features of a cluster should be calculated and averaged to receive λ. This is a good estimation but the quality decreases the nearer an object gets.
The time to collision can now be calculated as follows:

$$T_{TC} = \frac{\lambda}{\Delta\lambda} \cdot \Delta t \tag{5}$$

If the time between two images is known, T_{TC} is expressible in seconds. If the frame-rate is unknown, Δ can be set to 1 and T_{TC} is denoted in frames. The described method has the neat advantage that it provides additional depth information of the scene. The time to collision of all crossing objects is calculated concurrently with information about their distances. Those values can be meters if the camera speed is known, otherwise they are expressed up to scale.

B. Objects with an arbitrary trajectory

The method described above only holds for obstacles on a crossing collision course. It can be seen in (3) that the T_{TC} of an object moving in the same direction as the camera cannot be calculated.
Figure 6 depicts an obstacle moving towards the camera. This is again a relative motion as in Figure 5. The epipole E_o is always seen under a constant angle since it moves along a single ray. Other feature points, e.g. in distance h to E_o, change their angles over different images.
Figure 7 examines this scene from a different perspective.

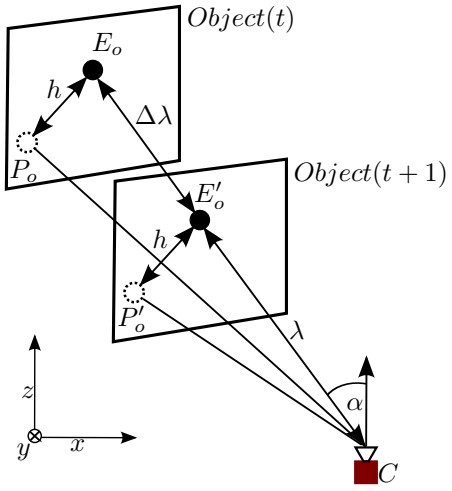

Fig. 6. Relative obstacle-point motion

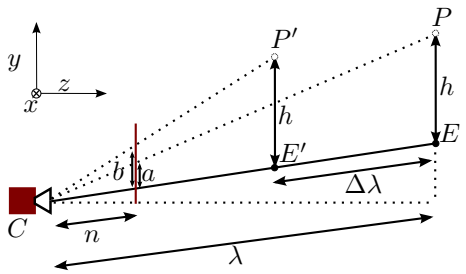

Fig. 7. TTC calculation by observing the distance $\bar{PE}$

The following relations are obtained from similar triangles relations in Fig. 7:

$$\frac{a}{n} = \frac{h}{\lambda} \tag{6}$$

$$\frac{b}{n} = \frac{h}{\lambda - \Delta\lambda} \tag{7}$$

The projection of the distance h from a feature P to the epipole E is called a for the distance λ to object. After the object moved by the distance $\Delta\lambda$, h is mapped to b, whereas n is the distance from the camera center to the projection of the epipole. Both equations can be combined to (8) with the nice effect that n and h cancel out. Hence, the equation does not require neither knowledge of intrinsic camera parameters nor any metric information.

$$\frac{\lambda}{\Delta\lambda} = \frac{b}{b-a} \tag{8}$$

If the frame-rate is known then $T_{TC} = \frac{\lambda}{\Delta\lambda} \cdot \Delta t$ can be expressed in seconds.

This approach is similar to the method proposed in [6], but due to the clustering of the flow vectors no size and shape estimation of the objects is necessary.
Compared to the method for crossing obstacles the epipole-distance-approach has the advantages of a simple calculation and general validity, but lacks information about the distances to the obstacles.

The knowledge of the T_{TC} of all obstacles and their respective motion directions $\vec{n_o}$ could be used as base for a reactive obstacle avoidance strategy for ground-based mobile robots.

IV. Experimental Results

The collision detection system was verified on a Pentium i5 computer system running Linux OS. The system was used with two types of features as input data for the collision detection. It was able to run with 30fps using a Kanade-Lukas-Tracker (KLT) for ego-motion estimation and 8fps using SURF features to establish correspondences between images of the monocular sequence.

A. Evaluation of the Collision Detection

First, we test how different collision scenarios are represented in the sparse optical flow data. We simulated in Fig. 8 three situations, where the possible collision candidate is too slow (green), colliding (red), or too fast in the intersection scenario. We see that for each of the cases there exists a point, where the flow vectors of each of the obstacles intersect. It is important to observe that for the case that the camera reaches the collision point faster than the vehicle (green case), the intersection point (epipole) of the object is outside of the object boundaries on the same side as the the motion epipole of the background (in this case, the motion of the camera is along the optical axis and the motion epipole of the background is the image center). The intuitive explanation is that the epipole is the position of the camera in the plane parallel to the image plane, when it reaches the plane containing the corresponding physical point on the colliding object. For the case that the object reaches the intersection point first (blue case) the epipole is outside of the object on the opposite side to the motion epipole of the camera. The camera will pass the plane of the object trajectory "behind" the object. The crucial case is the red configuration, where the optical flow vectors of the object intersect within the surface of the object. In that case, the camera will collide with the surface of the object. The collision happens if the optical flow vectors point outwards from the epipole and intersect within the object. All three cases start at the same object-to-camera position end expand from there.

A result of our rotation compensation on the optical flow field is depicted in Fig. 2. The sparse optical flow being a result from a KLT tracker here (green lines) is compensated for rotation in this image. The resulting flow vectors (blue) intersect all in one point as predicted in Fig. 3.

The clustering and the evaluation of the time to collision is based entirely on image information without any additional information about calibration parameters or motion of the camera. This allows to evaluate arbitrary monocular sequences (even from the Internet) for collisions if the motion of the camera was a pure translation in this sequence. We tested the system on sequences of collision movies. A segmentation result for an exemplary collision scene is depicted in Fig. 9.

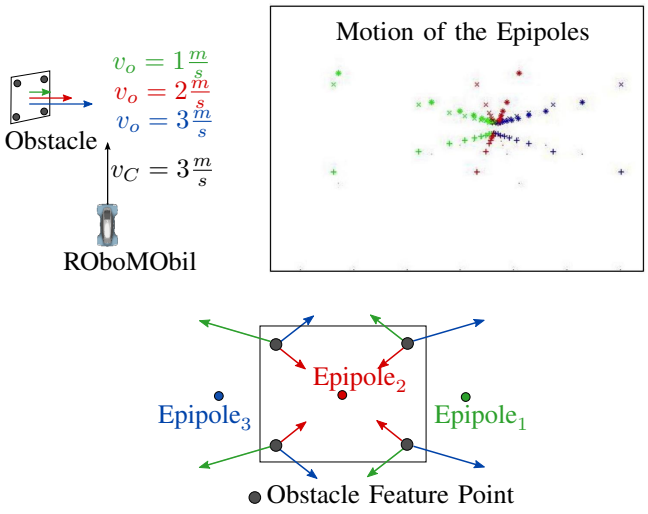

Fig. 8. Simulation of the feature motion for three intersecting trajectories.

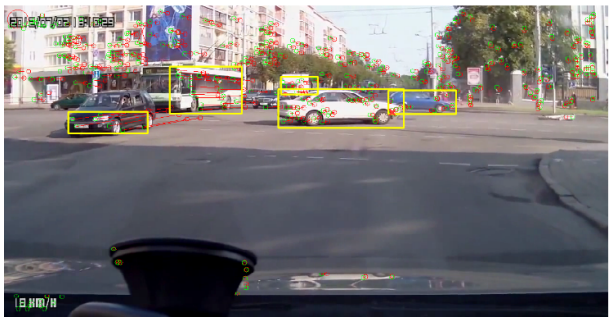

Fig. 9. Clusters in the collision scene grouped based on common intersection point (epipole)

We see an urban intersection scene where just a collision between the white and the blue car happened. The system segmented 5 independent motion components with detectable changes compared to the background. It can be seen in Fig. 9 that only the white car in front of the own vehicle has an epipole within the point cluster defining the vector field. The detected vector field is magnified in Fig. 10.

We can see in Fig. 10 that the flow defined from red to green circles in the image is moving away from the computed epipole (the red "x" in the image). This is the required indication for a collision for the case that the epipole is within the cluster of points from which it was calculated.

We see that the explosion field in Fig. 10 along the vehicle has a with the distance increasing length. Here we used SURF [28] features for feature matching. Some of the features were not matched correctly. These outliers can be

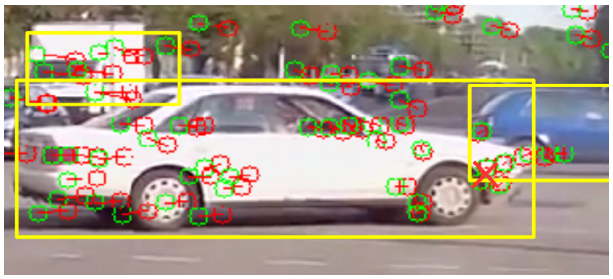

Fig. 10. Magnified distribution of optical flow vectors on a collision candidate.

Fig. 11. Easy detection of independent motion clusters.

filtered based on the distance of their supporting line to the estimated epipole. From the motion of the features (distance between the red and the green circle) and its distance to the epipole, we have estimated the TTC to 15 steps. We calculated the vectors between every 6th image to obtain longer vectors. This means that after 90 frames our camera would collide with the object in the intersection.

Fig. 11 depicts a detection of independent motion clusters in the scene represented by the three cars in front of the camera. The camera attached to a vehicle was moving towards the center line at this point.

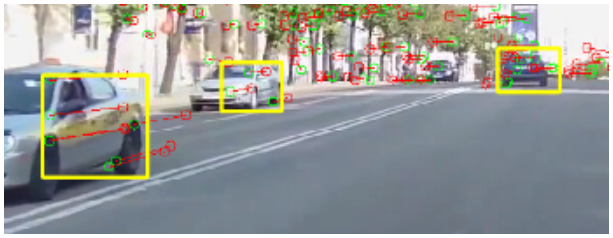

Fig. 12. Closeup depicting the direction of the clustered flow vectors (motion red to green).

We see in Fig. 12 that although the motion components in optical flow (especially for the left most car) are significant, no danger for the own vehicle exists. All the epipoles are outside of the clustered points and the obstacles move way from the motion epipole of the own car which is around the 4th car in the scene in the center of the image. This simple processing provides an easy way to deal with complex dynamic scenes, where many possible collisions may occur. Just objects with epipoles within the cluster have to be prioritized for planning based on their TTC value. The objects with shorter times represent a higher danger in the scene.

B. Evaluation of Time to Collision Estimation - TTC

We have already shown in case of Fig. 10 how the TTC estimation works. For the case that the epipole is withing the cluster, we need to estimate the ratio of the change in the radial length $\Delta\lambda$ to the absolute distance to the epipole λ to estimate the time. We validate the estimation of the time-to-collision based on (8) on all features in Fig. 8. All of them result in an identical plot depicted in Fig. 13.

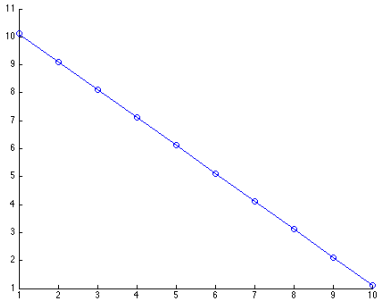

Fig. 13. Estimates of time-to-collision using (8) for any of the features (red case) in Fig. 8. We plotted the expected number of frames until collision along the y-axis and the frame index along x.

It is an interesting special case that if we follow another vehicle and travel into exactly the same direction and are able to track on optical flow, the denominator of (8) will go to zero indicating an infinite time until collision.

V. CONCLUSIONS AND FUTURE WORK

This paper has introduced a novel approach for the prediction of collision candidates by interpreting objects derived from optical flow. Dynamic obstacles are explicitly considered. Neither assumptions about the movement of dynamic objects are made nor their shape and size has to be estimated. For crossing objects the motion vector is calculated not in the two-dimensional image space but in three-dimensional Cartesian coordinates. This is possible by utilizing structure from motion methods. Additionally, the computational effort is limited through clustering disturbed regions in the optical flow field. Uncertainties and possible error sources are mutually eliminated over different processing steps. Only information derived from images is used. An intrinsic calibration for the camera is only necessary for estimating rotations.
An open issue is the clustering of objects that move in the same direction as the camera. A solution may be to analyze the length of flow vectors pointing to the global epipole.

The next step will be the implementation and evaluation of the reactive collision scheme on our robotic electric vehicle ROMO [1]. Additionally, the integration in a complex scheme for autonomous mobile robots as introduced by

Brooks [5] is an open topic. Since the proposed algorithm is settled on the lowest level of such an autonomy architecture, it will have to run concurrently and supportively with higher path planning algorithms.

ACKNOWLEDGMENT

The authors would like to thank the German Aerospace Center (DLR) for funding this project, Johann Bals, Jonathan Brembeck, and the ROboMObil team, who built the test platform to validate the results. We would like to thank Prof. Hirzinger for his continuous support and new ideas that made this work possible.

REFERENCES

[1] J. Brembeck, L. M. Ho, A. Schaub, C. Satzger, and G. Hirzinger, "Romo - the robotic electric vehicle," in *22nd International Symposium on Dynamics of Vehicle on Roads and Tracks*. IAVSD, 2011. [Online]. Available: http://www.iavsd2011.org/

[2] J. Silvious and D. Tahmoush, "Automotive gmti radar for object and human avoidance," in *Radar Conference (RADAR), 2011 IEEE*, may 2011, pp. 375 –377.

[3] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, june 2011, pp. 163 –168.

[4] C. Rabe, T. Müller, A. Wedel, and U. Franke, "Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time," in *Proceedings of the 11th European Conference on Computer Vision*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6314. Springer, September 2010, pp. 582–595.

[5] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, no. 1, pp. 14 – 23, Mar. 1986.

[6] G. Alenya, A. Negre, and J. Crowley, "A comparison of three methods for measure of time to contact," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 4565 –4570.

[7] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (ijcai)," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, April 1981, pp. 674–679.

[8] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *ARTIFICAL INTELLIGENCE*, vol. 17, pp. 185–203, 1981.

[9] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *INTERNATIONAL JOURNAL OF COMPUTER VISION*, vol. 12, pp. 43–77, 1994.

[10] M. Heindlmaier, L. Yu, and K. Diepold, "The impact of nonlinear filtering and confidence information on optical flow estimation in a lucas 00026; kanade framework," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, nov. 2009, pp. 1593 –1596.

[11] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sanchez, "Symmetrical dense optical flow estimation with occlusion detection," in *In ECCV*. Springer, 2002, pp. 721–735.

[12] X. Ren, "Local grouping for optical flow." in *CVPR'08*, 2008.

[13] F. Obermeier, S. Hawe, M. Zwick, and K. Diepold, "Optical flow reliability measures and their applicability for multi-sensor fusion," in *Intelligent Vehicles Symposium, 2009 IEEE*, june 2009, pp. 527 –531.

[14] A. Chavez and D. Gustafson, "Vision-based obstacle avoidance using sift features," in *International Symposium on Visual Computing*, 2009, pp. 550–557.

[15] R. Nelson and J. Aloimonos, "Obstacle avoidance using flow field divergence," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 10, pp. 1102 –1106, oct 1989.

[16] K.-T. Song and J.-H. Huang, "Fast optical flow estimation and its application to real-time obstacle avoidance," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3, 2001, pp. 2891 – 2896 vol.3.

[17] T. Low and G. Wyeth, "Obstacle detection using optical flow," in *Proceedings of the 2005 Australasian Conference on Robotics & Automation*, C. Sammut, Ed., 2005.

[18] E. Dur, "Optical flow-based obstacle detection and avoidance behaviors for mobile robots used in unmaned planetary exploration," in *Recent Advances in Space Technologies, 2009. RAST '09. 4th International Conference on*, june 2009, pp. 638 –647.

[19] D. Coombs, M. Herman, T.-H. Hong, and M. Nashman, "Real-time obstacle avoidance using central flow divergence, and peripheral flow," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 1, pp. 49 –59, feb 1998.

[20] G. Baratoff, C. Toepfer, M. Wende, and H. Neumann, "Real-time navigation and obstacle avoidance from optical flow on a space-variant map," in *Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings*, sep 1998, pp. 289 –294.

[21] M. Tistarelli and G. Sandini, "On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 4, pp. 401 –410, apr 1993.

[22] B. Cohen and J. Byrne, "Inertial aided sift for time to collision estimation," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, may 2009, pp. 1613 –1614.

[23] E. Mair and D. Burschka, *Mobile Robots Navigation, chapter Zinf - Monocular Localization Algorithm with Uncertainty Analysis for Outdoor Applications*, 2010.

[24] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 328 –341, 2008.

[25] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, pp. 137–154, 1992, 10.1007/BF00129684. [Online]. Available: http://dx.doi.org/10.1007/BF00129684

[26] J. Byrne and C. J. Taylor, "Expansion segmentation for visual collision detection and estimation," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, may 2009, pp. 875 –882.

[27] J. Klappstein, F. Stein, and U. Franke, "Detectability of moving objects using correspondences over two and three frames," in *Proceedings of the 29th DAGM conference on Pattern recognition*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 112–121.

[28] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.