Monocular Rear Approach Indicator for Motorcycles

Joerg Deigmoeller[1], Herbert Janssen[1], Oliver Fuchs[2] and Julian Eggert[1]
[1]*Honda Research Institute Europe, Carl-Legien-Strasse 30, 63073 Offenbach, Germany*
[2]*Honda R&D Europe (Germany), Carl-Legien-Strasse 30, 63073 Offenbach, Germany*

Keywords: Driver Assistance System, Monocular Camera, Optical Flow.

Abstract: Conventional rear-view mirrors on motorcycles only allow a limited visibility as they are shaky and cover a small field of view. Especially at high speeds with strong headwind, it is difficult for the rider to turn his head to observe blind spots. To support the rider in observing the rear and blind-spots, a monocular system that indicates approaching vehicles is proposed in this paper. The vision based indication relies on sparse optical flow estimation. In a first step, a rough separation of background and approaching object pixel motion is done in an efficient and computationally cheap way. In a post-processing step, pixel motion information is further checked on geometric meaningful transformations and continuity over time. As a prototype, the system has been mounted on a Honda Pan-European motorcycle plus monitor in the dashboard that shows the rear-view image to the rider. If an approaching object is detected, the rider gets an indication on the monitor. The rear-view on the monitor not only acts as HMI (Human Machine Interface) for the indication, but also significantly extends the visibility compared to mirrors. The algorithm has been extensively evaluated for relative speeds from 20 km/h to 100 km/h (speed differences between motorcycle and approaching vehicle), at normal, rainy and night conditions. Results show that the approach offers a sensing range from 20 m at low speed up to 60 m at night.

1 INTRODUCTION

On motorcycles, the visibility to all sides is significantly worse compared to cars. The rear-view is only possible by means of two mirrors, instead of three, whereas the field of vision is partially hidden by the riders body. Additionally, rear-view mirrors have a tendency to tremble at high velocity or on uneven ground. Therefore, improved rear-view and rider assistance are of great importance in the motorcycle domain to reduce the number of accidents and fatalities.

So far, there exist only special rear view systems (Quick, 2012) and blind spot assistance systems for cars. Latter mainly make use of radar ((Audi, 2013),(Mazda, 2011),(Hella, 2012)) or sonar (Bosch, 2012) sensors. Both sensors are not well suited for motorcycles as they require further treatment or even additional sensors like gyroscopes to deal with leaning conditions. Additionally, radar is quite heavy and expensive compared to the overall costs of a motorcycle. Sonar, in turn, only has a range of up to 6 m (at least for an acceptable sensor size). Therefore, cameras represent a good alternative with respect to costs, size and sensing range. Despite that, cameras can cope with leaning conditions and provide a rear-view image that can be displayed on a dashboard monitor.

Existing work on camera-based assistance systems are unfortunately only available for cars (Nissan, 2012). Commonly, the scaling factor of detected objects in consecutive video images, i.e. its change of size over time is used to decide whether it is approaching or not. For the detection of objects, (Stierlin and Dietmayer, 2012) and (Mueller et al., 2008) use optical flow information, whereas (Stein et al., 2003)) apply an appearance based method.

More complex approaches using pixel motion information from monocular images requires an ego-motion-compensation at first to detect other moving road users (Ma et al., 2004). This can either be done by feeding data from an IMU (Inertial Measurement Unit) and refining the ego-motion based on visual motion information (Rabe et al., 2007) or relying on vision information only ((Klappstein et al., 2006), (Scaramuzza and Siegwart, 2008)). Those methods are mainly used for front-facing cameras.

As soon as the camera is mounted to the rear, motion segmentation becomes much easier, as background motion caused by the ego-vehicle and object motion caused by approaching objects significantly differ in their scaling factor (see discussion in follow-

Deigmoeller J., Janssen H., Fuchs O. and Eggert J..
Monocular Rear Approach Indicator for Motorcycles.
DOI: 10.5220/0004668304740480
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 474-480
ISBN: 978-989-758-009-3

ing chapter), because the former is contracting while the latter is expanding.

Therefore, the method developed within this work is also based on monocular pixel motion. The decision was against a stereo system to save one camera. Additionally, the baseline of a stereo system would be quite small because of the limited space on a motorcycle. This in turn restricts the sensing range significantly. Another advantage of using motion features is the independence of an objects appearance, e.g. at night the appearance changes significantly compared to day-time as a vehicle can only be identified by its front-lights.

To the knowledge of the authors, there exists no such vision-based system for motorcycles yet. As the main difference in vehicle dynamics between car and motorcycle is the ability to ride in leaning position, (Schlipsing et al., 2012) proposed a method to estimate the roll angle of a motorcycle. The idea is to transfer existing assistance systems from the car domain, like lane detection or obstacle detections which requires such a roll-angle compensation. Obviously, this represents a convenient solution to make use of already available technologies. The disadvantage is that all post-processing depends on the reliability of the roll-angle compensation, which might not be desirable in sense of error propagation and independent running applications.

In the remainder of this paper, the approaching vehicle indication is described at first in Section 2. The implementation of the system on a motorcycle and experimental results under rainy, dark and high speed conditions are discussed in Section 3. Finally, the discussion and conclusion section summarizes the outcomes and explains remaining challenges.

2 APPROACHING VEHICLE DETECTION

Mounting a camera to the rear of a vehicle causes a contracting pixel motion in the image sequence if the vehicle moves forward. This means that all pixels move towards a focus of contraction if the scene is static. The magnitude of each motion vector in the image mainly depends on the corresponding depth of a pixel in real world coordinates. If an object is moving in the scene, the measured pixel motion of the object is a combination of the ego-motion and the object motion. This results in zero motion if the object is moving with the same speed in the same direction as the ego-vehicle.

As soon as the object velocity is larger than the ego-vehicle velocity, the pixel motion pattern of the object becomes upscaling with a flow field moving away from a focus of expansion. This means, during ego-vehicle movement, an approaching object causes an expanding flow field while static background causes a contracting flow field. This makes both patterns distinguishable by their scaling factor (greater or lower than 1). In turn, if the object drives with lower speed as the ego-vehicle, background and object motion are both contracting, i.e. both scaling factors are lower than 1.

If the ego-vehicle additionally undergoes a rotational motion, the projected motion pattern on the image plane is overlayed with a vertically translating component in case of pitching or a rotational component in case of rolling, whereas the magnitude of a motion vector is independent of the corresponding depth of a pixel. However, scaling factors are not influenced by rotational motion. The rolling component is of special interest for this application, as such a motion occurs only for motorcycles and not for cars.

In the following, potentially approaching object motion is detected by simply checking the scale factor in a local neighborhood. This is an efficient preprocessing step to reduce the number of non-relevant motion vectors for fitting a geometric motion model for approaching vehicles in a second step. The main advantage of this approach is that no ego-motion compensation needs to be done at all, which avoids the influence of errors from an additional pre-processing step.

2.1 Pre-selection of Motion Information

For motion estimation, a sparse pixel motion estimation method has been applied. Sparse means that motion vectors $\vec{v}_i = (u_i, v_i)^T$ with corresponding homogeneous pixel coordinates $\mathrm{x}_i = (x_i, y_i, 1)^T$ are only computed at well structured regions. Compared to dense motion estimations, which compute motion vectors for every pixel, such methods can save significant computational effort. The method used here is the pyramid implementation of the Lucas and Kanade optical flow estimation (Bouguet, 2000), available in the OpenCV library (OpenCV, 2013). The big advantage of the pyramid implementation compared to the standard Lucas and Kanade approach is the ability to cover large pixel displacements by propagating over different image resolutions.

To decide whether a motion vector corresponds to an approaching vehicle or to the background, at least two neighboring motion vectors are required to compute their scaling factor. As opposed to dense motion vector fields, the neighborhood within a sparse motion vector field is not clearly defined.

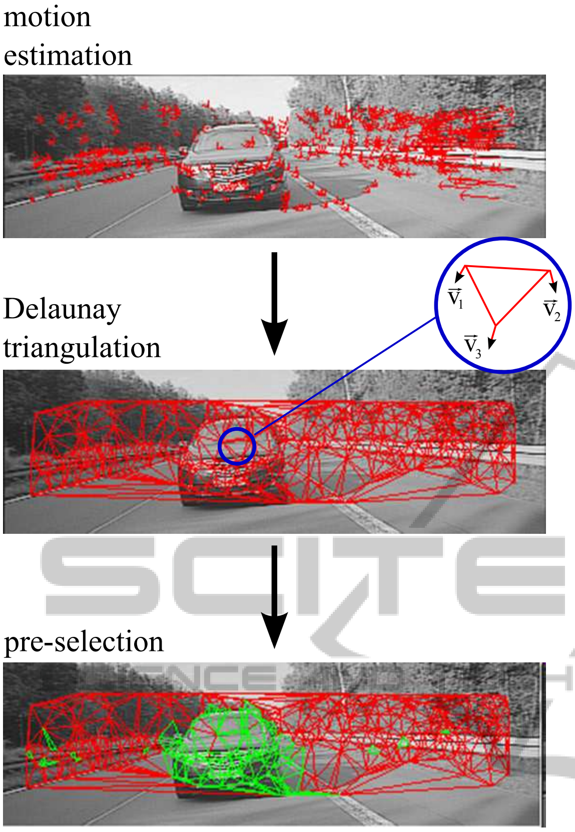

Figure 1: Pre-selection of motion vectors that correspond to an approaching object, starting with motion estimation, followed by triangulating the coordinates of motion vectors and, finally, keeping only those vector triples (green edges) where all possible combinations of vectors fulfill $s_x > 1$ and $s_y > 1$.

Therefore, a Delaunay triangulation (Shewchuk, 2002) is applied to create neighborly relations between all x_i in a mesh. For triangulation, the software Triangle by Jonathan Shewchuk is used (Shewchuk, 1996).

The Delaunay triangulation has the specific property that within a circle that is drawn around three coordinates, a triangle does not contain any other coordinate of the complete mesh (see middle image in Fig. 1). Such a network allows to compare each motion vector with its closest neighbors within a triangle. To make a decision whether a triple of motion vectors may correspond to an approaching vehicle or not, the scaling factor of two motion vectors at each edge of a triangle is computed.

It is assumed that the motion in a close neighborhood mainly consists of a translation T and scaling S, whereas rotational and shearing as well as perspective transformations can be neglected. A motion vector $\vec{\mathrm{v}}_i$ at the homogeneous coordinate x_i can be expressed as follows:

$$\begin{aligned}
\vec{\mathrm{v}}_i &= \underbrace{(A_s - E')}_{A'_s}\mathrm{x}_i, \quad \text{where} && (1)\\
A_s &= \begin{pmatrix} S & T \end{pmatrix}, \; S = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \\
T &= \begin{pmatrix} t_x \\ t_y \end{pmatrix} \text{ and } E' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}
\end{aligned}$$

E' has been subtracted here to allow a direct mapping between homogeneous coordinates x_i and 2D-vectors $\vec{\mathrm{v}}_i$. To estimate the scaling factors s_x and s_y, two motion vectors within a triple are subtracted to get rid off the translation T:

$$\vec{\mathrm{v}}_i - \vec{\mathrm{v}}_j = A'_s\mathrm{x}_i - A'_s\mathrm{x}_j = A'_s \begin{pmatrix} x_i - x_j \\ y_i - y_j \\ 0 \end{pmatrix} \qquad (2)$$

Rearranging the equation above and solving for s_x and s_y yields:

$$\begin{pmatrix} s_x \\ s_y \end{pmatrix} = \begin{pmatrix} \frac{u_i - u_j}{x_i - x_j} + 1 \\ \frac{v_i - v_j}{y_i - y_j} + 1 \end{pmatrix} \qquad (3)$$

Three motion vectors within a triangle are only considered for further processing if all possible combinations of vector pairs fulfill the constraint to be upscale, i.e. $s_x > 1, s_y > 1$. Fig. 1 illustrates all steps for pre-selecting motion vectors.

2.2 Geometric Model Fitting

After the pre-selection step, it is assumed that if any motion information remains, it mainly corresponds to approaching vehicles. Due to wrong or imprecise measurement it is still possible that some motion vectors may fulfill the constraint to be upscaling even if they do not follow a meaningful motion.

Therefore, the widely used robust regression method RANSAC (RANdom SAmple Consensus, (Fischler and Bolles, 1981), (Ma et al., 2004)) is applied to fit a geometric model into the remaining motion vectors to further check for a meaningful transformation. The chosen model is an affine transformation A, which is a good approximation when real world coordinates lie on a plane parallel to the image sensor and move towards the camera:

$$\vec{\mathrm{v}}'_i = (A - E')\mathrm{x}_i \qquad (4)$$

Again, E' has been subtracted to allow an affine mapping between x_i and $\vec{\mathrm{v}}'_i$. The RANSAC method finds a model that supports as many motion vectors as possible, which is defined by the number of motion vectors in the inlier set or also called consensus set $C(A)$:

$$C(A) = \{\vec{v}_i \,\varepsilon\, V : \min_{\vec{v}_i^{'} \varepsilon M(A)} \text{dist}(\vec{v}_i^{'}, \vec{v}_i) \leq 1\text{px}\}, \qquad (5)$$

where V is the whole data set of motion vectors, $M(A)$ is the manifold of the model A and $\text{dist}(\cdot,\cdot)$ is the Euclidean distance between measured vector $\vec{v}_i$ and model vector $\vec{v}_i^{'}$. Here, the Euclidean distance is fixed to an accuracy of 1 pixel.

After fitting the parameters of the affine model, A is decomposed into its components to identify whether the motion pattern is upscaling or not:

$$\begin{aligned} A &= \begin{pmatrix} K & T \end{pmatrix}, \text{ where} \qquad (6) \\ K &= \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix}, \text{ and } T = \begin{pmatrix} t_x \\ t_y \end{pmatrix} \end{aligned}$$

The translational component T can be read off directly. In turn, scaling components can be separated from rotational components, if shearing components are comparably small within K, by taking the square root of the diagonal entries of $K^T K$:

$$\begin{aligned} K^T K &\approx (RS)^T RS = S^T R^T RS \\ &= S^T ES = \begin{pmatrix} s_x^2 & 0 \\ 0 & s_y^2 \end{pmatrix} \qquad (7) \end{aligned}$$

where S, R and E are 2 by 2 scaling, rotation and identity matrices, with $R^T R = E$. Only affine motion patterns that are upscaling are considered for post-processing.

In this application, there remain two possible motion patterns after the pre-selection step. First, approaching vehicles, i.e. motion patterns that are of interest. Second, if the motorcycle is standing still, e.g. at a traffic light. This type remains because the scaling factor of the background motion hovers around one ($s_x \approx 1$, $s_y \approx 1$) due to noise. Therefore, this motion information can not be identified in the pre-selection step based on local relations only. Instead, the global affine transformation clarifies this circumstance by its averaged scaling parameters. It has to be mentioned, that in case of a non-moving motorcycle, the static scene becomes planar and can be expressed by an affine transformation. Only if the scaling factors of the global transformation are below a certain threshold t_s, it is assumed that the motorcycle is standing still.

The RANSAC method is iteratively applied two times to cover both possible motion patterns of an approaching vehicle and the non-moving motorcycle. If the first fit describes the situation of a motorcycle which is standing still or s_x and s_y is even downscaling, corresponding motion vectors are removed and a second model is computed. In turn, if the first fit contains an upscaling model, no more iteration is done. The following pseudo-code illustrates this procedure:

```
FOR i = 0 to 1
  doRANSAC()          // compute affine model
  IF (($s_x > t_s$)  &&  ($s_y > t_s$))
    removeOutlier()   // remove motion vectors
                      // that do not fit to model
    break             // break loop and do
                      // post-processing
  ELSE
    removeInlier()    // remove motion vectors
                      // that fit to model
    continue          // do second fitting
ENDFOR
```

2.3 Post-processing

To make the detection of approaching vehicles more robust, a temporal filtering is applied. To do so, all values of a 2-dimensional grid $I_C(x_i, y_i, t)$, with size of the input image, are initialized with zero values. Values at coordinates $(x_i, y_i)^T$ are set to 1, if they relate to a motion vector in the current consensus set at time t (see Fig. 2):

$$I_C(x_i, y_i, t) = \{ \begin{array}{l} 1 \text{ if } \vec{v}_i^{'} \,\varepsilon\, M(A) \\ 0 \text{ else} \end{array} \qquad (8)$$

$I_C(x_i, y_i, t)$ is then combined with values $I(x_i^{'}, y_i^{'}, t)$ which have been predicted from previous time steps:

$$I(x_i, y_i, t) = \alpha\, I_C(x_i, y_i, t) + (1 - \alpha)\, I(x_i^{'}, y_i^{'}, t), \qquad (9)$$

where α describes the decay over time.

In this application, α is set to 0.1 to make the filtering robust against outliers. To also allow a variance in position, each coordinate value is blurred by a 3 by 3 Gaussian kernel followed by a median filter of size 3 by 3 to remove noisy areas.

An indication is finally given to the rider if the sum of values in the filtered 2-dimensional grid $I(x_i, y_i, t)$ exceeds a certain threshold t_a, i.e.

$$t_a > \sum_i med[G * I(x_i, y_i, t)], \qquad (10)$$

where G is the Gaussian kernel and med is the median filtering.

The prediction of coordinate values $I(x_i, y_i, t)$ to the next time step $I(x_i^{'}, y_i^{'}, t+1)$ is done with the aid of the affine transformation A from Equation 6:

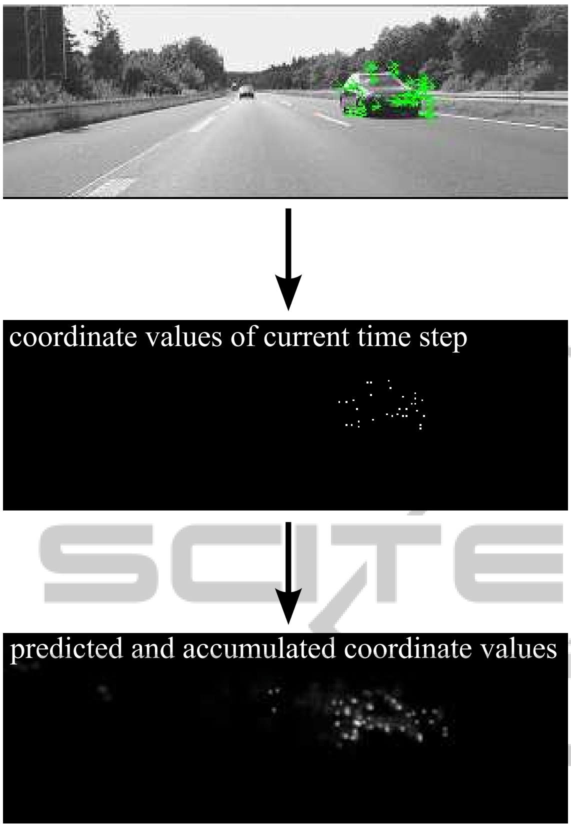
coordinate values of current time step
predicted and accumulated coordinate values

Figure 2: Illustration of coordinate values that correspond to the detected motion vectors (top and middle image) and prediction plus accumulation over time (bottom image).

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = K \begin{pmatrix} x_i \\ y_i \end{pmatrix} + T \qquad (11)$$

3 PROTOTYPE EVALUATION

The prototype motorcycle is a Honda Pan European with a PlayStation Eye camera (75° diagonal field of view, 640x480 at 15 frames per second) mounted to the rear. For the image processing part, the image is scaled down to 320x240 and is cropped afterwards to a resolution of 320x108 (mainly sky and lower part of the road are removed). The algorithm is running on a Core2-Duo PC (each core running at 1.86 GHz), that is stored in the side-bag of the motorcycle. The average computation time at day is 28 ms and 6 ms at night (almost dark). A display is connected to the PC which shows the full rear-view image plus the indication in the upper image part (s. Fig. 3).

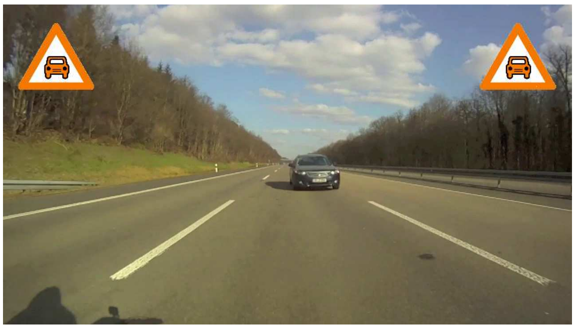

Figure 3: Overlay of triangle icon to indicate approaching vehicle to the rider.

3.1 Recording Set-up

For recording, the prototype and an additional car were equipped with GPS-sensors to get ground truth data for relative speed and distance between motorcycle and approaching vehicle. The GPS-data has been synchronized with the video stream, which has been captured frame-wise (uncompressed) by the PlayStation Eye camera.

The test-rides include high-speed conditions (car is overtaking with up to 200 km/h while the motorcycle is at 100 km/h), bad-weather and night conditions. The overall recording times were 26 min of maneuvers with approaching cars and 1 hour 45 min without any car in the video stream as well as sequences where the motorcycle is standing still.

3.2 Data Evaluation

The ROC curves below (Receiver Operator Curve) show the correct warnings (true positive rate, TPR) against the false warnings (false positives per hour, FP/h) for all recorded conditions. The TPR is event based, which means that as soon as the algorithm detects an approaching vehicle within a positive labeled sequence, the detection is correct. Each positive labeled sequence has a fixed length of 15s (time until the approaching vehicle reaches the motorcycle). Finally, the TPR is the ratio between the number of correctly detected vehicles divided by the total number of positive labeled sequences. The FP/h is estimated by counting false warning events, i.e. consecutive frames of false warnings are interpreted as single event.

The ROC curves are estimated by increasing the threshold t_a (cf. Equation 10) by 0.01 for a range of $t_a = [0.0 \ldots 10.0]$ (see Fig. 4). It can be directly perceived, that the red ROC-curve is a bit jagged instead of monotonically decreasing. This effect is because of clustering consecutive false positives to a single event.

In special situations, the system still returns false

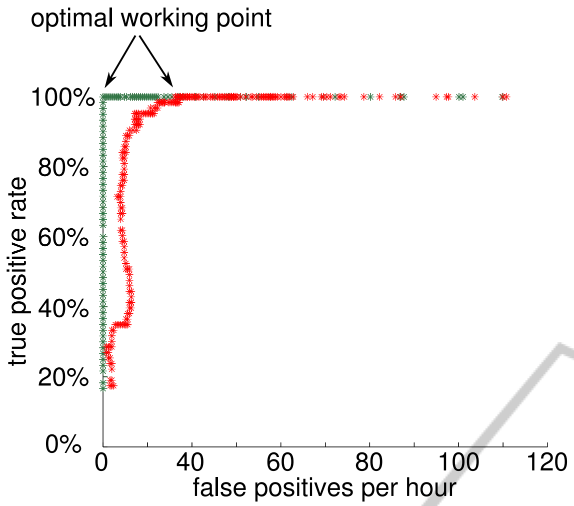

Figure 4: ROC curves of the system for values of t_a ranging from 0.0 to 10.0. The red curve represents the performance of the system for recordings including the stroboscopic effect. The green curve shows the performance for recordings without stroboscopic effect. The two arrows indicate the same optimal working point for each ROC curve.

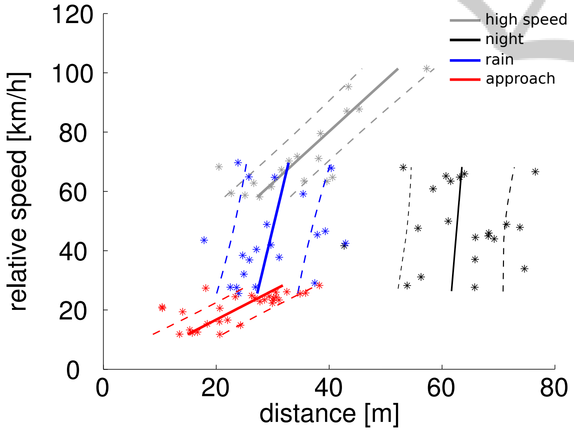

Figure 5: Scatter plot of detection distance (distance where the vehicle has been detected) against relative speed (speed differences between motorcycle and approaching vehicle) for the optimal working point chosen from the ROC curves above.

warnings (see red ROC curve in Fig. 4). Those are due to the so called stroboscopic effect which humans also experience, e.g. on television (tyres of vehicles are moving backwards while they are driving forwards). Objects seem to move in different directions as in the real world, because of temporal aliasing effects caused by periodic motion at a rate close to the frame rate of the camera. The proposed system interprets this motion as true motion and warns the rider in case that the motion is upscaling. The recordings contain such effects at an approximate speed of 120 km/h of the motorcycle while it passes a specific bridge railing with periodic pattern. These bridges appear several times in the recordings, also because the route has been driven multiple times.

If sequences including the stroboscopic effect are removed from the data, the algorithm has an optimal working range for values of $t_a = [1.7 \ldots 1.74]$, i.e. TPR = 100% and FP/h = 0 (see green ROC curve). The remaining recording times are 25 min of maneuvers with approaching cars and 1 hour 37 min without any car in the video stream. Choosing the same working point for the red ROC-curve gives TPR = 100% and FP/h = 32.

For the optimal working point in the ROC curve, which corresponds to $t_a = 1.7$, an additional scatter plot is drawn (see Fig. 5). It depicts the distance and relative speed between vehicle and motorcycle when the vehicle has been detected for the first time. Each dot in the plot represents one driven maneuver. The scatter plot shows four types of maneuvers: overtaking at high speed (up to 100 km/h relative speed), overtaking at night, overtaking in rainy conditions and approaching (car approaches on same lane as motorcycle).

For each maneuver, a line has been fit into the data set (solid line) with standard deviation (dashed lines). As can be seen, the detection distances increase with higher relative speed. Rainy conditions obviously do not significantly worsen the performance of the system. This is due to the fact that the lens kept clean for the whole ride because of the airstream. Surprisingly, the detection distance at night is nearly constant at approximately 60 m. The contrast at night around the vehicle spotlights and the light cone in front of the vehicle allow a very good motion estimation in the images. As soon as the vehicle is at a certain distance, so that pixel movement is measurable, the algorithm is able to immediately identify the moving front-lights.

4 CONCLUSIONS AND OUTLOOK

Supporting the rider in observing blind spots and improving the surround view compared to mirrors is an important task to reduce accidents involving motorcycles.

In this paper, a very robust and simple monocular approaching vehicle detection has been presented, so that the rider is aware of other traffic participants behind or in blind spots. The system presented in this paper relies on pixel motion information only and hence is independent of the object appearance.

Extensive tests have been carried out under different conditions including bad weather and rain. The sensing distances span a range from 20 m at low relative speed (20 km/h relative speed) up to 60 m at night conditions. The quantitative results are very promising so that the presented approach provides a cheap and easy to implement support feature for motorcycles.

Next steps will be undertaken to tackle the problem of the stroboscopic effect. Additionally, the viewing conditions concerning display size, display resolution and camera field of view will be optimized to further increase the riding comfort.

ACKNOWLEDGEMENTS

The authors thank Achim Bendig for the productive teamwork and setting up the prototype. Special thanks also to Nils Einecke for fruitful discussions, and, last but not least, many thanks to Kazuyuki Maruyama and Mutsumi Katayama for supporting the project.

REFERENCES

Audi (2013). Audi side assist - innovative driver assistance system. http://www.gizmag.com/audi-digital-rear-view-mirror-production/23681/.

Bosch (2012). Side view assist. http://www.bosch-automotivetechnology.com/en/de/homepage/homepage_1.html.

Bouguet, J.-Y. (2000). Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. http://robots.stanford.edu/cs223b04/algo_tracking.pdf.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*.

Hella (2012). Driver assistance systems. http://www.hella.com/hella-com/502.html?rdeLocaleAttr=en.

Klappstein, J., Stein, F., and Franke, U. (2006). Monocular motion detection using spatial constraints in a unified manner. *IEEE Intelligent Vehicles Symposium*.

Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. S. (2004). *An Invitation to 3-D Vision*. Springer-Verlag, New York, 2nd edition.

Mazda (2011). Mazda's rear vehicle monitoring system to receive euro ncap advanced award. http://www.mazda.com/publicity/release/2011/201108/110825a.html.

Mueller, D., Meuter, M., and Park, S.-B. (2008). Motion segmentation using interest points. *IEEE Intelligent Vehicles Symposium*.

Nissan (2012). Multi-sensing system with rear camera. http://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/rear_camera.html.

OpenCV (2013). Open source computer vision library. http://opencv.willowgarage.com/wiki/.

Quick, D. (2012). Audi's digital rear-view mirror moves from racetrack to r8 e-tron production vehicle. http://www.gizmag.com/audi-digital-rear-view-mirror-production/23681/.

Rabe, C., Franke, U., and Gehrig, S. (2007). Fast detection of moving objects in complex scenarios. *IEEE Intelligent Vehicles Symposium*.

Scaramuzza, D. and Siegwart, R. (2008). Monocular omnidirectional visual odometry for outdoor ground vehicles. *Computer Vision Systems, Springer Lecture Notes in Computer Science*.

Schlipsing, M., Salmen, J., Lattke, B., Schroeter, K. G., and Winner, H. (2012). Roll angle estimation for motorcycles: Comparing video and inertial sensor approaches. *IEEE Intelligent Vehicles Symposium*.

Shewchuk, J. R. (1996). Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. *Applied Computational Geometry: Towards Geometric Engineering*.

Shewchuk, J. R. (2002). Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22:1–3.

Stein, G., Mano, O., and Shashua, A. (2003). Vision-based acc with a single camera: bounds on range and range rate accuracy. *IEEE Intelligent Vehicles Symposium*.

Stierlin, S. and Dietmayer, K. (2012). Scale change and ttc filter for longitudinal vehicle control based on monocular video. *IEEE Intelligent Transportation Systems*.