Bendjaballah *et al. EURASIP Journal on Image and Video Processing* (2016) 2016:41
DOI 10.1186/s13640-016-0147-0

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

A classification of on-road obstacles according to their relative velocities

Mourad Bendjaballah[1*], Stevica Graovac[2] and Mohammed Amine Boulahlib[1]

Abstract

The systems based on image processing have numerous applications in the domain of motion control of robots and autonomous vehicles. The current paper is oriented to the solution of the problem that precedes the implementation of automatic avoidance of the on-road obstacles—how to detect them, to track in the sequence of images, and to recognize which of them are stationary, incoming, or outgoing from the camera. The overall algorithm of obstacle classification presented in this paper consists of three basic phases: (1) image segmentation in order to extract the pixels belonging to the image of a road and the objects over it; (2) extraction of characteristic points inside the area of the obstacle, their description and tracking in following frames; and (3) estimation of distances between the camera, the obstacles and their rates of change (relative velocities). The verifications of particular steps of the proposed algorithm are illustrated using real road-traffic images, while the overall algorithm is tested using both synthesized sequences of images and the ones acquired in real driving.

Keywords: Machine vision, Image processing, Image segmentation, Pattern recognition, Support vector machines, Feature extraction, Tracking, SURF, Pose estimation

1 Introduction

In recent years, self-anti-collision systems have been developed for preventing traffic accidents and achieving safe driving. This system should alert drivers of the presence of obstacles and help them to react in advance. In these systems, the ability to detect obstacles is essential. The safe operation of a vehicle depends heavily on the vision. The vision of a driver can be improved by systems that provide information about the environment around the vehicle that cannot be seen or barely seen by human eyes. Therefore, an obstacle detection system based on machine vision is the subject of current research in smart vehicle technology.

The particular aim of this work is to enable the classification of the on-road obstacles according to their relative velocities, onto the categories of incoming, outgoing, and stationary, as a prerequisite for their avoidance in the context of autonomously guided cars.

The existing techniques used in the on-road obstacle detection may vary according to the definition of the obstacles. They might be classified into two categories [1]. The first one is related to the obstacles reduced to a specific object (vehicle, pedestrian, etc). In this case, the detection can be based on search for specific patterns, possibly supported by features such as texture, shape [2, 3], symmetry [4, 5], or the use of an approximate contour. The second category is used when the definition of the obstacles is more general. In this case, two methods are generally used. (1) The usage of a monocular camera based on an analysis of optical flow [6–9]. This method requires rather huge calculation, and it is sensitive to vehicle movement. Also, it detects only the moving obstacles and fails when obstacle has small or null speed (static ones). (2) The method based on stereo vision [10–13]. Images are captured using two or more cameras at the same time from different angles, and then obstacles are detected by matching. This method generally requires more time to do the necessary calculations, and it is sensitive to the local motion of each camera caused by the vehicle movement.

Generally, a method for detecting both moving and static objects simultaneously is required because the static objects such as boxes can fall on the road in front of a car and they are dangerous too.

*Correspondence: mourad.bendjaballa@hotmail.com
[1]Military Academy, University of Defence, Belgrade, Pavla Jurisica Sturma 33, Belgrade, Serbia
Full list of author information is available at the end of the article

Actually, the algorithm of on-road obstacles detection should provide that:

1. The objects that are outside the road are eliminated.
2. Irregularities on the road surface that are not affecting the driving are not considered.
3. Static obstacles on the road are properly recognized in order to be avoided.
4. Vehicles on the road are detected in order to adjust own motion according to their relative distances and velocities.

We propose in this paper an obstacle detection method using a monocular camera mounted on a vehicle receiving the light variations in the scene on the road ahead and analyzing the captured images to carry out the obstacle detection. The output of the proposed algorithm is a classification onto classes of moving and static obstacles. After getting the obstacle information, drivers can react quickly and precisely to take corresponding actions to prevent car accidents. Moreover, the system of autonomous car driving may react appropriately in order to keep the motion of a car along the nominal trajectory relative to the road borders, simultaneously avoiding incoming and outgoing cars. Here, obstacles are defined as actual arbitrary objects protruding from the ground plane in the road area, both static and moving ones. Road markers in the road area (e.g., pedestrian crossings) as well as a number of objects outside the road region are considered as the obstacles of no interest.

The current paper is organized as follows. Section 2 presents the methods used in the diffrents steps of the proposed algorithm. Section 2.1, for the given sequence of video images, synthetic or taken by a mobile camera, the road region is detected using support vector machine method. Section 2.2 describes how the on-road obstacles are detected and extracted over the road region. From this point on, the obstacle is represented by rectangular area around the detected object on the road. Description of the characteristic points inside these areas and their tracking from frame to frame is done using the SURF (Speeded-Up Robust Features) algorithm in Section 2.3. Section 2.4 deals with the calculation of the position and the relative velocity of each obstacle to classify the static and dynamic (incoming, outgoing) ones. Illustrations based on experimental results are given throughout Section 3. The paper is concluded in Section 4, with some comments regarding the actual limits of the application and suggestions for the future work.

2 The methods

In order to detect the road obstacles, to track them, and to determine their positions and relative velocities, the following operations are employed (as illustrated on Fig. 1).

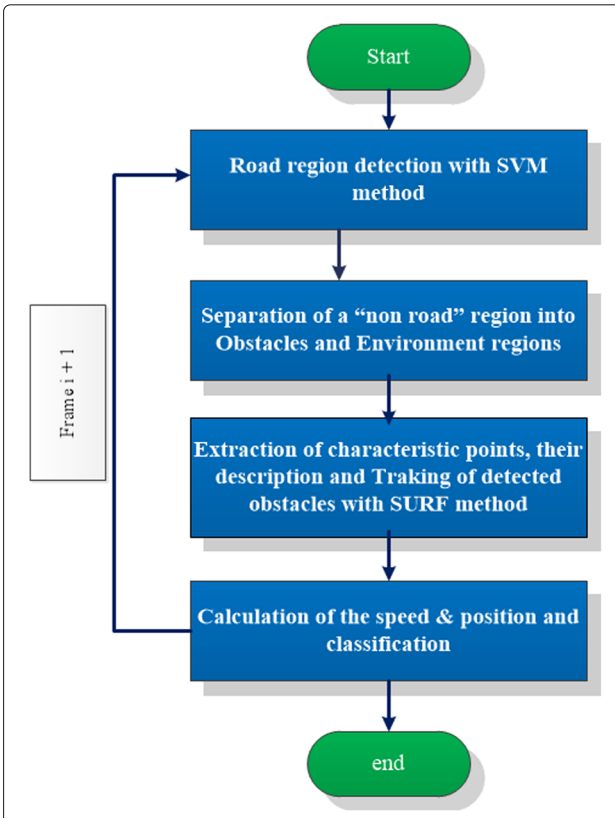

Fig. 1 Flowchart of the proposed method

First, the road region is detected using the SVM (support vector machine) classification method in order to distinguish class "road" from the class "non-road". Second, the non-road region as the result of this detection is classified into two areas: "obstacles" and "road environment." After the latter classification, one has three types of regions: environmental area, road region, and obstacles. The real obstacles on the road like cars, pedestrians, boxes, etc. are belonging to the class "obstacles." Monitoring each of these obstacles is done by using the SURF matching algorithm. The final step consists in calculating the obstacles' positions in the field of view and the calculation of their relative velocities in order to distinguish the static and dynamic obstacles (inside the range of 200 m ahead).

2.1 Road region extraction

The first step in the algorithm consists in segmentation of an image into the road region and the other region that includes the remaining part of image ("non-road"). In order to classify one pixel as a member of a class "road," there are a number of possible segmentation methods based on color, texture descriptors based on statistic parameters, structure, or frequency spectrum, etc. While some acceptable results have been obtained when the color components have been used only, even three

decades ago [14] or by use of the best candidates among texture statistic and structure descriptors [15], our reasoning here was oriented toward a more complex approach where the color and texture are simultaneously considered [16].

The proposed algorithm is composed of five components. In the first feature extraction component, a feature vector is extracted from each pixel of the input image. Second, the component of dynamic training database (DTD) is filled with training set labeled by a human supervisor in initialization and updated by the new training set online. Third, the component of Classifier Parameters Computing is used to estimate the parameters in SVM classifier. The fourth SVM classifier component is in charge of training and classification which takes the training data and classifier parameters to train the SVM classifier and use the trained SVM classifier to classify image into road/non-road classes. The last component contains two stages: morphological operation and online learning operation. The former implements connected region growing and hole filling on the classification result to determine the road region. The latter compares morphological result and classification result to evaluate the quality of the current classifier, then select new training set from that comparison and update the DTD. The flowchart shown on Fig. 2 illustrates this algorithm.

As an initial operation, the populations of "road" and "non-road" pixels are indicated by an operator (driver) action, via marking the appropriate rectangular regions on the image as shown in Fig. 3. The same initialization can be made by automatic designation of a rectangular window in the central lower part of image, a priory guaranteeing that the contents is typical for the road area. This way, an initial content of a Dynamic Training Database (DTD) is specified.

In order to reduce the calculations, the number of pixels inside the rectangle is limited to 1000. If the total of encompassed pixels is greater, one thousands of them will be chosen in a random manner. This DTD is going to be continually updated in order to follow the changes in the road scene. The selected set of classifying parameters is calculated for every subsequent image. The process of segmentation is based on the SVM method. The final step in the classification consists in morphological processing of the binarized image. After the final segmentation is done, the upgrade consisting in online updating of DTD is the finishing step before the acquisition of a new image.

Feature vector is eight-dimensional:

$$F_{i,j} = [f_{t1(i,j)}, f_{t2(i,j)}, f_{t3(i,j)}, f_{t4(i,j)}, f_{t5(i,j)}, f_{c1(i,j)}, f_{c2(i,j)}, f_{c3(i,j)}] \quad (1)$$

where $i = 1..H$ and $j = 1..W$

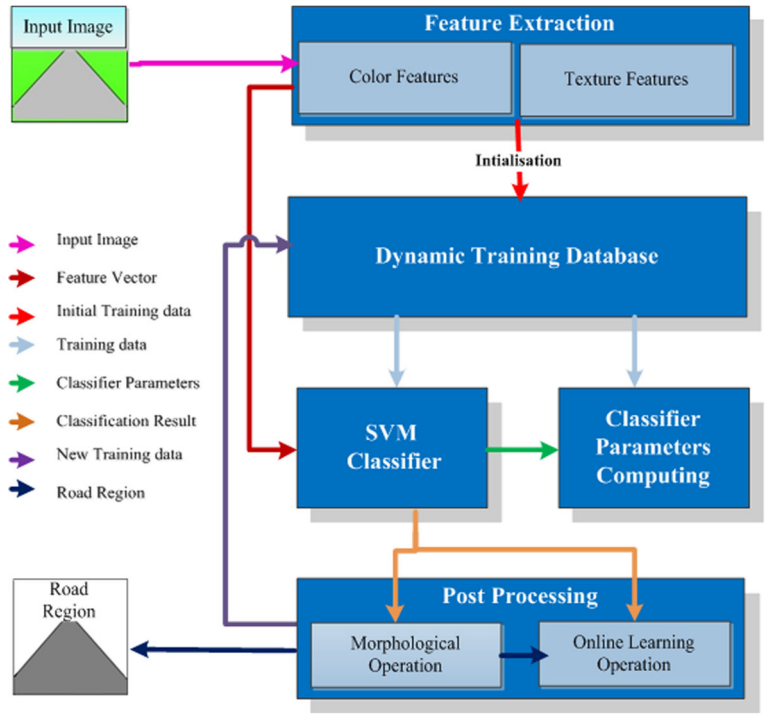

Fig. 2 Flowchart of the road region extraction algorithm

Fig. 3 Initialization of DTD. *Red rectangles* are used for "positive" training (road class), and *blue* ones for "negative" training (non-road class)

The first five elements are Haralick's statistical features:

$$\text{Energy} = \sum_u \sum_v \{p(u,v)\}^2 \tag{2}$$

$$\text{Enthropy} = \sum_u \sum_v p(u,v) \log\{p(u,v)\} \tag{3}$$

$$\text{Contrast} = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{u=1}^{N_g} \sum_{v=1}^{N_g} p(u,v) \right\} \tag{4}$$

$$\text{IMD} = \sum_u \sum_v \frac{1}{1+(u-v)^2} p(u,v) \tag{5}$$

$$\text{Correlation} = \frac{\sum_u \sum_v (u.v) p(u,v) - \mu_x \mu_y}{\sigma_x \sigma_y} \tag{6}$$

where IMD is the inverse moment of differences, $p(u,v)$ is an element of gray level co-occurrence matrix (GLCM), and (μ_x, μ_y) and (σ_x, σ_y) are the mean values and covariances calculated using this matrix.

The remaining three elements of feature vector are pixels, USV color, and components.

It is natural to suppose that the features space of "road" and "non-road" classes are in nonlinear relation and that it is not expected to obtain some linear hyper-plane which is distinguishing these two classes in original feature space. Following the results given in [17], a Gaussian radial basis function (RBF) kernel is used as the SVM kernel function. There are two classifying parameters: complexity parameter C and γ parameter. It should be found which one is more appropriate for this discrimination. In order to do so, the parallel validation relative to these two parameters is done on the image belonging to DTD.

Due to the continuous dynamic changes of the road contents as a result of camera motion, DTD should be updated from time to time. It was chosen that after each ten frames, the training databases for both classes are refreshed by replacing a hundred of stochastically chosen old members by a hundred of new ones, among the population of pixels already classified in the particular class. The larger numbers of updated elements leads to excessive impact of incorrectly classified pixels, while for too low numbers of replaced sampled pixels, one can expect low adaptation abilities.

After this step of classification, it is usual that there would be a number of small unconnected groups of pixels around the road, classified as the "road", as well as the number of "holes" over the road region. In order to eliminate such small aggregations of pixels, the algorithm includes morphological operations "opening" and "filling the holes".

Online training upgrade of SVM method [16] is optional but is very useful in the context of this application. Besides the already mentioned updating the DTD, it includes the evaluation of the performance of current classification. This process is based on the basic assumption that the road region consists from connected pixels. As a result of this, "road" pixels detected outside the main region of road as well as the "non-road" pixels located over the road region are the sources of information on how the classifier should be modified.

2.2 On-road obstacles extraction

2.2.1 Classification inside the "non-road" region

To extract the on-road obstacles, one has to remove two kinds of image objects: the marks on the road and the environment outside the road. It is supposed that the markers on the road are going to be associated to the road region in the previous process of road detection. The environment around the road has been already classified as "non-road" region in the first step of classification. This step of algorithm is oriented toward the separation of the whole "non-road" region into two subclasses: the "obstacles on the road" and "everything else existing outside the road".

Figure 4 shows the result of detection of a road region (a) and the template image of the road region (b) where the black pixels are representing the road. After analyzing of a particular row in the image, one obtains a profile as shown in Fig. 5. Based on this line profile, white line segments that have two adjacent black segments on both the left and right sides are the line segments belonging to the on-road obstacles. By checking each row in the template image of the road, this classification can be done. Figure 4c shows the results of this classification. The overall class of obstacles on the road is represented by gray pixels on (d).

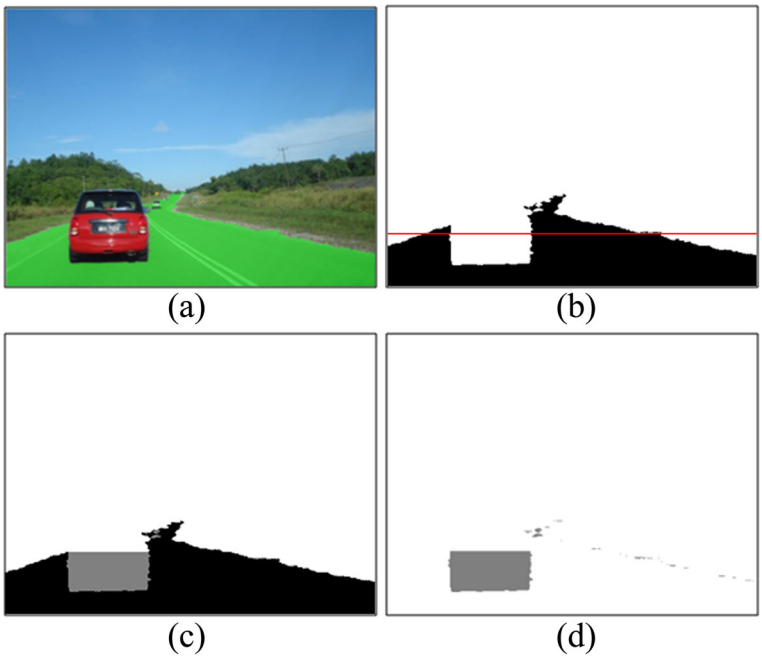

Fig. 4 Phases of classification inside the "non-road" region. **a** The result of road region detection. **b** Road region template image. **c** The result of region classification. **d** Obstacle region

2.2.2 Obstacles' detection and marking

After the latter classification phase, three regions (classes of pixels) are obtained: road region, obstacles on the road, and environment region, while the on-road obstacle class is important only. This region contains multiple objects of different sizes (Fig. 4d). As a first step, the small objects (less than 50 pixels) are eliminated because they are considered as the false obstacles.

The extracted obstacles should be tracked continuously in the sequence of incoming frames. In order to prepare this tracking phase, some area of interest should be specified—the detected obstacles should be marked by specifying some tracking window encompassing each of them. Even the last step in the relative velocity estimation is strictly affected by the choice of this regular geometrical shape corresponding to the particular obstacle. Figure 6 shows the different steps of marking the obstacles on the road. Figure 6a shows the region of the real obstacle after filtering unwanted objects. The red rectangle around this region is shown in b which will be replaced in the next step by the green square of the width equal to the base of red rectangle as in c. The final representation of a search area superimposed to the original image is shown in d.

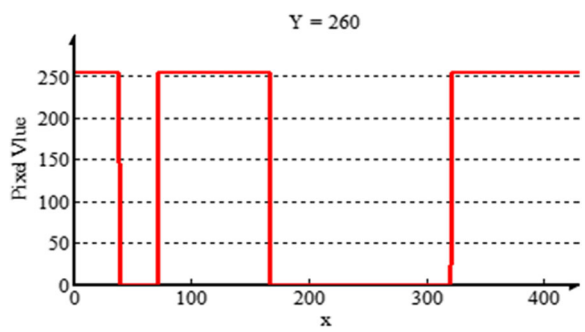

Fig. 5 Line profile of pixel intensity values (260th row in the road region image)

2.3 Principle of tracking of the on-road obstacles

In order to estimate the relative velocities of the obstacles, they should be tracked in sequence of frames. If the camera is stationary, the difference between two consecutive images would be used as a natural source of information which part of the image is belonging to the stationary background and which part is candidate to be associated to the moving object. In our particular case, the camera itself is a moving object and tracking principle cannot be based on this reasoning. The previous step in the proposed algorithm was ended by the extraction of a rectangular area around the detected road obstacle, and the focus of attention should be directed toward these regions in the sequence of incoming images.

The very first idea could be to detect in the next frame what is the position where the overall content of rectangular window around the obstacle can be found, based on some correlation measure. This principle would be obviously a time-consuming one and, moreover, sensitive to

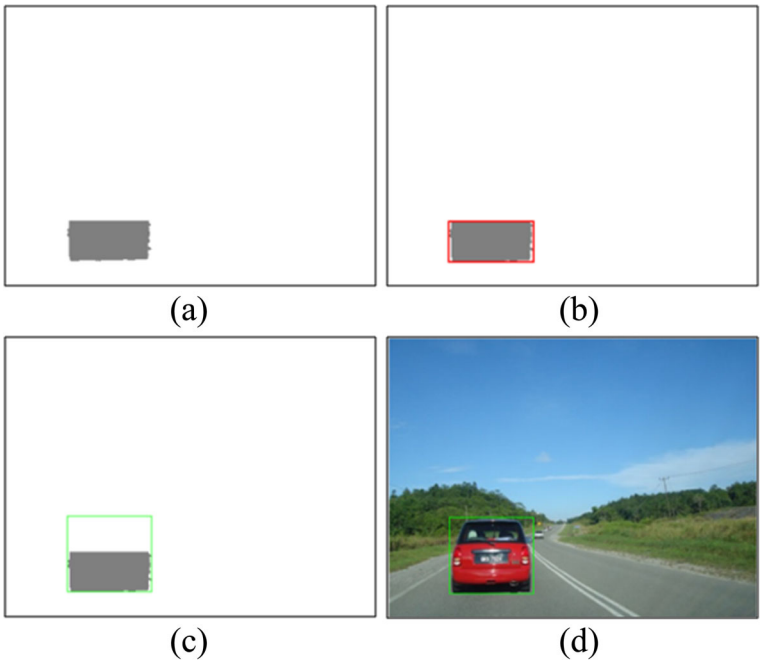

Fig. 6 a–d Phases of on-road obstacle marking

the expected scale and rotation transformations in both cases of incoming and outgoing obstacles.

As a result of this, it is more appropriate to select among all pixels inside the tracking window a subset of points that are the characteristics according to some pre-specified criterion and to track them from frame to frame. These characteristic points (key-points) may vary a lot, based on the principle of extraction and capabilities to preserve the stability of the object features based on their choice. Reference [18] was the source of exhaustive survey of methods related to the subject of specifying the points of interest as well as to the descriptors associated to these points.

Basically, our choice was oriented toward the points characterized by high value of local gradients. The illumination conditions between two consecutive frames are not going to be changed in some appreciable amount, while the local gradients would keep almost the constant values in the presence of affine transformations.

Corners are usually used as the characteristic points, since they can be used to compute an angular orientation for the feature. Sometimes, it is highly suggested to apply some sort of low-pass filtration as a first step in order to reduce the influence of the noise. This way, some combinations of LP and HP filtration, (Laplacian of Gaussian—LOG—is the typical example) are used to extract the corners as points of interest.

The well-known Harris detector [19] was analyzed as a first among the appropriate candidates for characteristic points extraction and description. While some very good tracking results have been obtained using Harris detector in a number of typical road scenarios, it was decided that some more complex description of the neighbourhood around the "high gradient" points is preferable in order to overcome the problems when the obstacle changes its size and orientation rapidly, which is typical when the incoming vehicle closely approaching the camera, or the other vehicle is just over-passing the vehicle where the camera is mounted.

The next choice in this direction was the choice of SIFT (Scale Invariant Feature Transform) algorithm. SIFT belongs to spectra descriptors, typically involving more intense computations in floating point. It is developed by Lowe [20, 21] and provides the way of finding interest points and feature descriptors, invariant to scale, rotation, illumination, affine distortion, perspective and similarity transforms, and noise. SIFT includes stages for selecting center-surrounding circular weighted difference of Gaussian (DoG) maxima interest points in scale space to create scale-invariant key-point. While the SIFT algorithm can be considered as the most powerful for this purpose, it is a rather complex for any kind of real-time applications.

As a natural candidate based on SIFT principles, we considered the SURF algorithm [22] as a version reducing the computing time. The SURF algorithm is composed of three consecutive steps [22, 23]. The first step is the detection of interest points, and the second step is building the descriptor associated with each of the interest points. The last step is descriptor matching.

2.3.1 SURF algorithm

SURF was developed to improve the speed of interest point detector, descriptor generation, and matching. SURF uses a very basic Hessian matrix approximation for feature point detection. At a point $p(x, y)$ in an image I, Hessian matrix in$H(p, \sigma)$ at scale is defined as follows:

$$H(p,\sigma) = \begin{bmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{bmatrix} \quad (7)$$

where $L_{xx}(p, \sigma)$ is the convolution of the Gaussian second-order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image I in point x, and similarly for $L_{xy}(p, \sigma)$ and $L_{yy}(p, \sigma)$. The authors approximate the Hessian matrix with box filters approximating second-order Gaussian derivatives and the filtering can be performed using integral images with a very low computational complexity while the calculation time is independent of the filter size. Let D_{xx}, D_{xy}, and D_{yy} be the approximations of L_{xx}, L_{xy}, and L_{yy} respectively.

The filter responses are further normalized with respect to their size, which guarantees a constant Frobenius norm for any filter size. With the Frobenius norm remaining constant for the box filters at any size, the filter responses are scale normalized and require no further weighting. The construction of the scale space starts with the 9×9 filter. Then, filters with sizes 15×15, 21×21, and 27×27 are applied (Fig. 7).

The dominant orientation assignment for the local set of HAAR features is found using a sliding sector window of size $\frac{\Pi}{3}$. This sliding sector window is rotated around the interest point at intervals. Within the sliding sector region, all HAAR features are summed. This includes both the horizontal and vertical responses, which yield a set of orientation vectors. The largest vector is chosen to represent dominant feature orientation. By way of comparison, SURF integrates gradients to find the dominant direction.

To create the SURF descriptor vector, a rectangular grid of 4×4regions is established surrounding the point of interest, and each region of this grid is split into 4×4 sub-regions. Within each sub-region, the HAAR wavelet response is computed over 5×5 sample points. The final descriptor vector is of dimension 64: 4×4 regions with four parts per region.

2.3.2 Proposed tracking algorithm

Algorithm 1 On-road obstacles tracking

Begin
Extraction of obstacles from the frame N°1;
for $i = 2 \rightarrow$ *Number of frames* **do**
  Extraction of obstacles from the frame N°i;
  for $j = 1 \rightarrow$ *Number of obstacles in frame N°i − 1;* **do**
    Test = false;
    for $k = 1 \rightarrow$ *Number of obstacles in frame N°i;* **do**
      Test = SURF(frame i − 1, frame i, obstacle j, obstacle k); ▷ *search if the obstacle j in the frame i-1 matches obstacle k in the frame i.*
      if *Test* **then**
        obstacle k = obstacle j;
      end if
    end for
    if *Test == false* **then**
      lost obstacle;
    end if
  end for
  The remaining obstacles are the new ones;
end for
end

The key-points are extracted from the rectangular regions detected after the step described in Section 3.2. These key-points are described as vectors in the description step. The next step is the matching. Several vectors from a database are matched against new vectors from a new input image by calculating the Euclidian distance between these vectors. This way the objects can be recognized in a new frame. When the sufficient number of matched points is found, particular obstacle is marked as recognized and it is not tested more. Some new obstacles would appear in this new frame, and they will be considered in the next one, while there would be the cases that some of previously existing obstacles are now vanishing or are not recognized. Unmatched obstacles

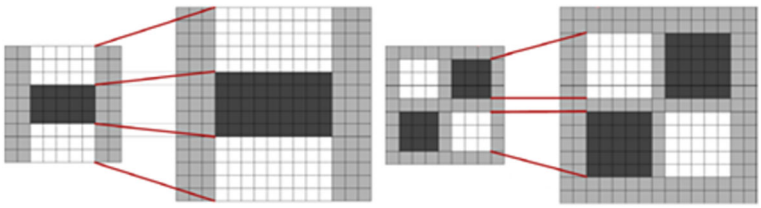

Fig. 7 Filters D_{xx} (*left*) and D_{xy} (*right*) for two successive scale levels (9×9 and 15×15)

would be treated in the next frames as the new ones. The pseudo-code illustrating this part of algorithm.

2.4 Calculation of relative positions, velocities, and classification of the on-road obstacles

2.4.1 General case of a reconstruction of camera's spatial and angular position

The spatial and angular position of a camera [24] (moving coordinate frame $O_c x_c y_c z_c$ (CCF)) can be calculated relative to the outer stationary coordinate frame ($O_I x_I y_I z_I$ (ICF)) according to the general relationships illustrated in Fig. 8. Transformation matrix between these two coordinate frames is defined as $T_O = T_1(\phi)T_2(\vartheta)T_3(\psi)$, where the angular orientation of a camera relative to ICF is defined via a set of Euler angles of jaw, pitch, and roll (ψ, ϑ, ϕ), and $T_i i = 1, 2, 3$ are the elementary matrix transformations. This rotational transformation is followed by translation specified by position vector $\vec{R}$:

$$\vec{e}_C = T_O \vec{e}_I;\ \vec{R} = x_O \vec{e}_{I1} + y_O \vec{e}_{I2} + z_O \vec{e}_{I3} \tag{8}$$

In order to reconstruct the scene depth, $|\vec{R}|$, one should know some a priory information about the distance $|O_I M|$ between two points inside the scene (e.g., the distance between the point M in horizontal plane of ICF and coordinate origin O_I as is illustrated in Fig. 8).

In practice, position reconstruction is based on detection of a rectangle $ABCD$, in the horizontal plane of ICF with coordinate origin O_I at the cross-section of diagonals, the axis $O_I x_I$ parallel with AB (in direction of a vanishing point P), and the axis $O_I y_I$ parallel with BC (in direction of a vanishing point Q), (Fig. 9).

For this particular application, parallel road lane borders are used to specify vanishing point P (direction of $O_I x_I$), and the direction of $O_I y_I$ is perpendicular to it (according to the condition $\vec{m}_P \vec{m}_Q = 0$), while the lane-width is used as a priory known distance in ICF.

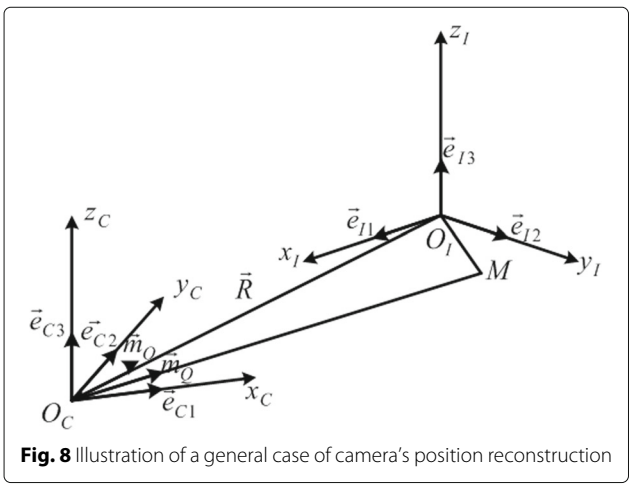

Fig. 8 Illustration of a general case of camera's position reconstruction

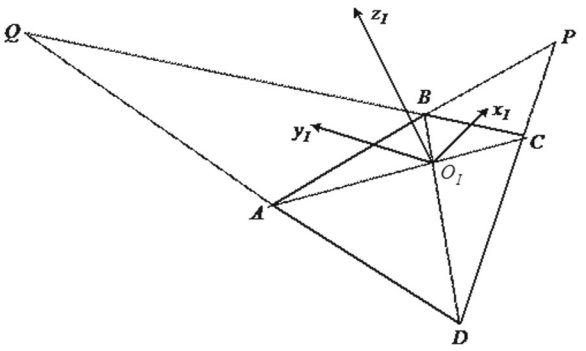

Fig. 9 Position reconstruction procedure

"m-vector" for any image point is generally defined as:

$$\vec{m} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = f \begin{bmatrix} \frac{x_c}{x_c} \\ \frac{y_c}{x_c} \\ \frac{y_c}{x_c} \end{bmatrix} = \begin{bmatrix} f \\ x_L \\ y_L \end{bmatrix};\ x_c \neq 0 \tag{9}$$

where f is a focal distance, (x_c, y_c, z_c) are the coordinates of a point represented in CCF, and (x_L, y_L) are its coordinates in the focal plane.

2.4.2 Position of an obstacle relative to the camera

To calculate the position of an on-road obstacle relative to a camera mounted on the moving vehicle, one should apply the principle illustrated in Fig. 9. This principle is illustrated here on an example of synthesized sequence of images, assuming the angular orientation of camera $(0°, 0°, -5°)$, field of view $\pm 15°$, and focal distance $f = 5$ mm. As a first step, one has to extract the borders of the lane in the lower part of image, where they are parallel. The reference rectangle $ABCD$ now is as shown in Fig. 10a with a priory known the information: $(AD)//(BC)$, $(AB)//(CD)$ and the width of the lane (3 m in this example). Vanishing point P is the intersection of AD and BC, Q is the intersection of AB and DC, and O is the center of the rectangle $ABCD$. Figure 10b shows the result of calculating the scene depth using the information about distance $|OE|$ which is equal to one half of the lane-width and the position of the camera relative to the virtual point O ($\vec{R}$ vector). The next step is calculating the distance to each of the obstacles on the road by providing the virtual point O using the scene depth, but here, the unknown quantity becomes $|O_I M| = |OO_1|$. Point O_1 is the center of the base of the green square (the result of algorithm described in Section 3). Figure 10c illustrates this last step in calculation of a scene depth to the point O_1, while in Fig. 10d, the camera

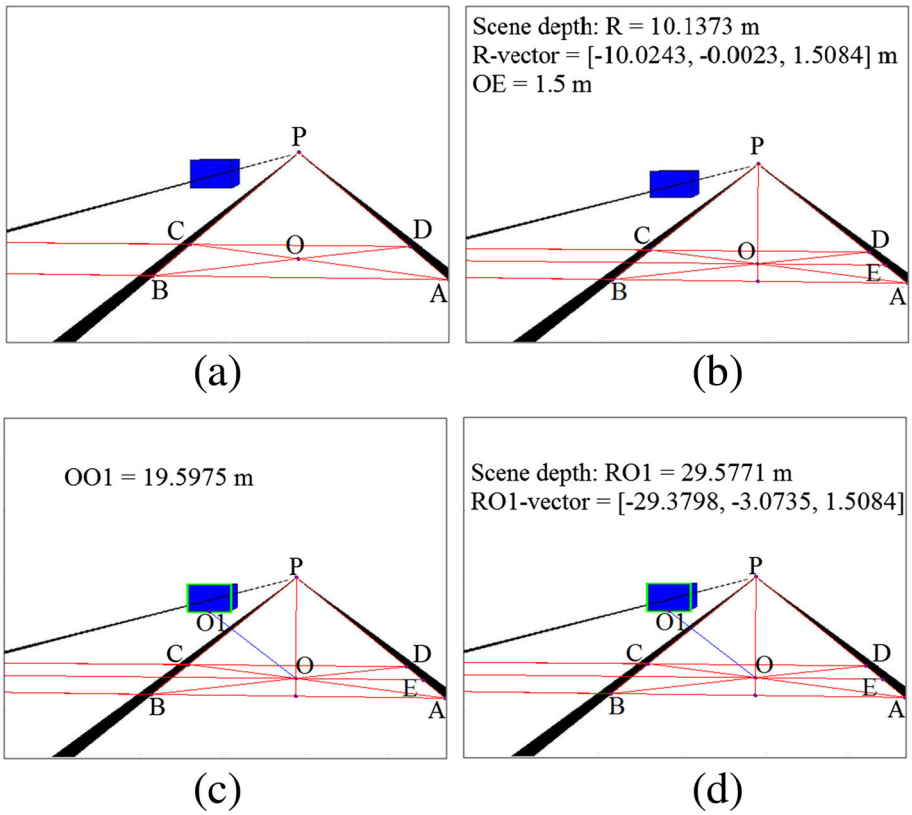

Fig. 10 a–d Illustration of a camera's position reconstruction

distance from the obstacle is shown. The estimated relative position is $(-29.38, -3.07, 1.51)$[m], while the real was $(-30, -3.15, 1.5)$[m], introducing the relative error of $(2.1, 2.5, 0.67)$%.

2.4.3 Relative velocities

To calculate the relative velocity of each of the obstacles on the road, one should first calculate the relative positions in consecutive time instants t and $t+1$, as shown in Fig. 11, using the relation:

$$V_{ob} = \frac{R_{ob}^{t+1} - R_{ob}^{t}}{\triangle t} \tag{10}$$

2.4.4 Classification

$$V_{\text{Relative}} = V_{\text{Obstacle}} - V_{\text{Camera}}$$

$$\Rightarrow V_{\text{Obstacle}} = V_{\text{Relative}} + V_{\text{Camera}} \tag{11}$$

The classification is done according to:

$$\begin{aligned} V_{\text{Obstacle}} = 0 &\Rightarrow \text{Stationary} \\ V_{\text{Obstacle}} < 0 &\Rightarrow \text{incoming} \\ V_{\text{Obstacle}} > 0 &\Rightarrow \text{outgoing} \end{aligned} \tag{12}$$

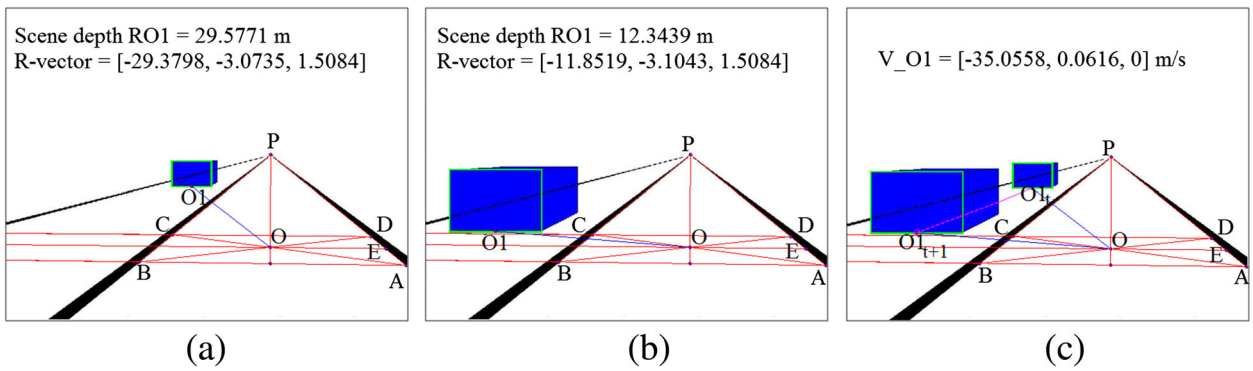

Fig. 11 a–c Illustration of a relative velocity calculation

2.4.5 Potential obstacles

Besides the obstacles that are already existing on the road that should be classified according to their velocities relative to the moving camera, there are some objects potentially suddenly incoming onto the road from the lateral sides. These are pedestrians who are irregularly crossing the street or vehicles making some parking maneuver nearby the street car track, animals on the road, etc. While it was shown that the initial classification of the road surface is possible to make by proper characterization based on color and texture features, the same approach is practically not possible to apply when one considers the neighboring non-road region, due to a huge diversity of all possible continuously changing "out-of-the-road" backgrounds. In order to take into account these potential obstacles, they should be detected also and characterized among themselves relative to their velocity component perpendicular to the road orientation, in the following way.

Firstly, the vanishing point P should be determined as the intersection of the lane and road border lines. Then, the neighboring strip close to the road on the right side (3 m in width and 30 m in length) could be extracted and considered from its light intensity distribution point of view. All pixels with a light intensity outside the region specified as $\pm\sigma$ (standard deviation) around the mean value are the candidates to represent the potential objects. The realistic images require some morphological filtration of erosion type in order to eliminate small groups of pixels after this segmentation. After that, the relevant groups of pixels are aggregated and their centroids are representing the potential objects' positions. Their positions and velocities in direction perpendicular to the road orientation are calculated in the same manner as for the obstacles already appearing on the road. Objects that are moving toward the road are classified as potential suddenly incoming obstacles. Their detection on the road (if they appear) is going to be done in the regular way as for the other road obstacles, but their existence is a type of alarm because their appearance might require some immediate reaction (fast stopping, sharp maneuver,...).

3 Results and discussion

3.1 Experimental results of the road region extraction

3.1.1 Comparison of results obtained by SVM method and color only method

The results of the road region extraction are shown in Fig. 12. The typical images representing the highway scenario, country road, and a street scene in an urban environment are exploited.

The first row on Fig. 12 shows that the initialization of DTD should select possible road markers (lane separators) as a part of road region. Similarly, when some areas in the image are initially designated as the road environment, the representative pixels of near and far environment should be included. These were the criteria for positioning the selecting windows on the images, while their shapes and sizes are of no particular importance since the selection of 1000 points inside each rectangle is used. The second row in Fig. 12 shows the results of SVM classification made on these test pictures. It is obvious that some morphological erosion procedures are needed in all cases, while the need for filling of holes on the road was not frequently present. The third row illustrates the situation when the final classification results are superimposed onto the original image. In order to show the superiority of the used classification method, the fourth row represents the final classification results if the color feature is used only (initialization and morphological processing have been the same).

3.1.2 Comparison of results between offline learning and online learning

Figure 13 shows three illustrations of advantages of online training upgrade.

In the first example (a), the initial sampling acquired the shadowed part of road but not in a sufficient way. The second example (b) represents the case where the background was not encompassing the sky region. In the third example (c), one can see that the surrounding buildings are misclassified as a road because of the lack of learning. Results obtained via offline learning are shown in the second row, while the online upgrade resulted in the binary images shown in the third one. After the morphological processing, the resultant classification of offline learning are superimposed in the original images in the fourth row. The results of the online learning, clearly showing superiority, are illustrated in the fifth row.

3.2 Experimental results of the on-road obstacles extraction

Figure 14 shows the different steps of the on-road obstacles detection in a sequence of digital images. the first row illustrates the final classification results of the road region using SVM method. The second row shows the result of the real obstacle extraction. The third row represents the final representation of a search area superimposed onto the original image. Some of the tall vehicles, like the trucks, are not going to be fully encompassed by this type of tracking window, some low-profile cars would not fill the tracking window completely, but the choice of a square-shaped tracking window seemed as a reasonable compromise.

3.3 Experimental results of the on-raod obstacles tracking

3.3.1 Comparison between Harris, SIFT, and SURF

Running time

The average time spent on key-point extraction, description, and matching, normalized by the total

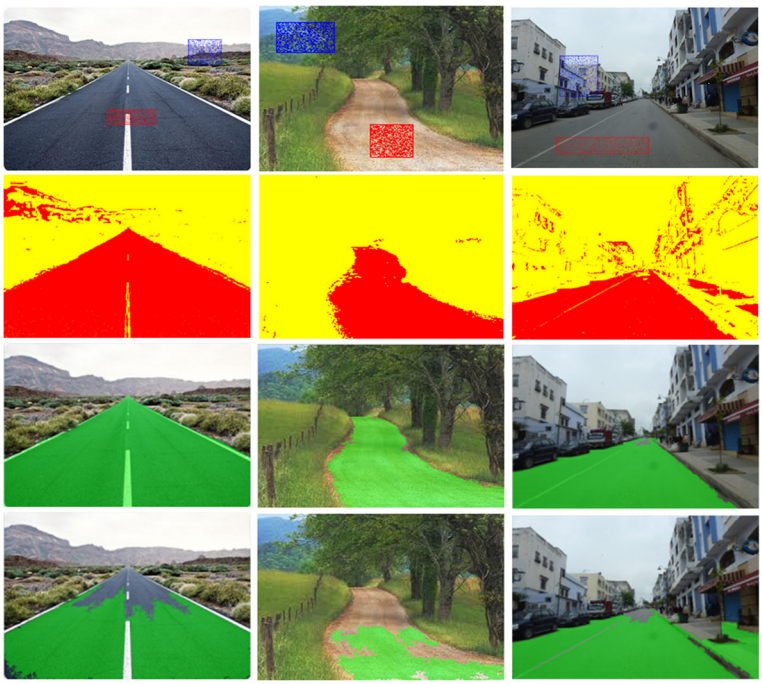

Fig. 12 The process of basic road detection algorithm and comparison of results obtained by SVM method and color only method. *1st row*: the original image and sampling windows. *2nd row*: the classification results (*Red* is road class. *Yellow* is non-road class). *3rd row*: the results of morphological filtration of SVM method. *4th row*: the results obtained by the color only method

number of key-points processed are compared. The results were obtained on an Intel(R) Core(TM) i5-4210 CPU @ 1.70 GHz, 1.70 GHz, and 8 GB of RAM using the Matlab library implementations for each detector, descriptor, and matching scheme. The results in Fig. 15a clearly show that Harris detector spends approximately 0.75 ms on each feature, while SIFT spends 3 ms and SURF spends approximately 1 ms. SIFT and SURF compute at all scales, but SURF is most likely because $t_{\text{SIFT}} \simeq 3 \times t_{\text{SURF}}$.

Number of matches

Figure 15b shows the number of matching points obtained by Harris detector, SIFT, and SURF algorithm. Harris detector generates approximately 15 matches, and in other case, using the SIFT algorithm, it generates approximately 53 matches. The accelerated variant of SIFT is SURF, and it generates approximately 31 matches which is very accepted if the running time of each algorithm is taken into consideration.

3.3.2 Experimental results of the on-road obstacles tracking

As it is illustrated in Fig. 16, two outgoing vehicles are considered as obstacles in series of pairs of two consecutive frames, on 40 [ms] interval. The number of matches of characteristic points between the pairs of tracked obstacles is enough sufficient, especially having in minds the fact that there were no false matches between the different obstacles at all. As it was expected, the number of characteristic points and matched ones is obviously greater when the obstacle is closer to the camera. The second row in Fig. 16 illustrates the final result of the tracking.

3.4 Results of calculation of relative positions, velocities, and classification of the on-road obstacles

3.4.1 Simulation results of relative position calculation

The above described algorithm of relative position calculation was first tested through a simulation of the tracked vehicle, assumed to perform a uniform movement. Although the image of the obstacle seen by the camera is "calculated" by the simulator, the processing chain implemented (incorporating the detection phases/tracking and extraction of the primitive of interest) is exactly the one to be used in real cases of images. The tracked vehicle undergoes a uniform speed defined by the vector $(12.5, 0, 0)^T$[m/s] and an initial position defined by the vector $(70.5, 0, 0)^T$[m]. It is important to note in this test, the vehicle carrying the camera undergoes a uniform speed defined by the vector $(21.25, 0, 0)^T$ [m/s] and an initial position defined by the vector $(0, 0, 0)^T$[m].

The results obtained at the end of the simulation are shown in Fig. 17. These show the estimates of the coordinates of the reference point in comparison to the actual data, imposed by the simulated spatial relationships between the obstacle/vehicle and the vehicle carrying the camera. It is obvious that the error increases with the

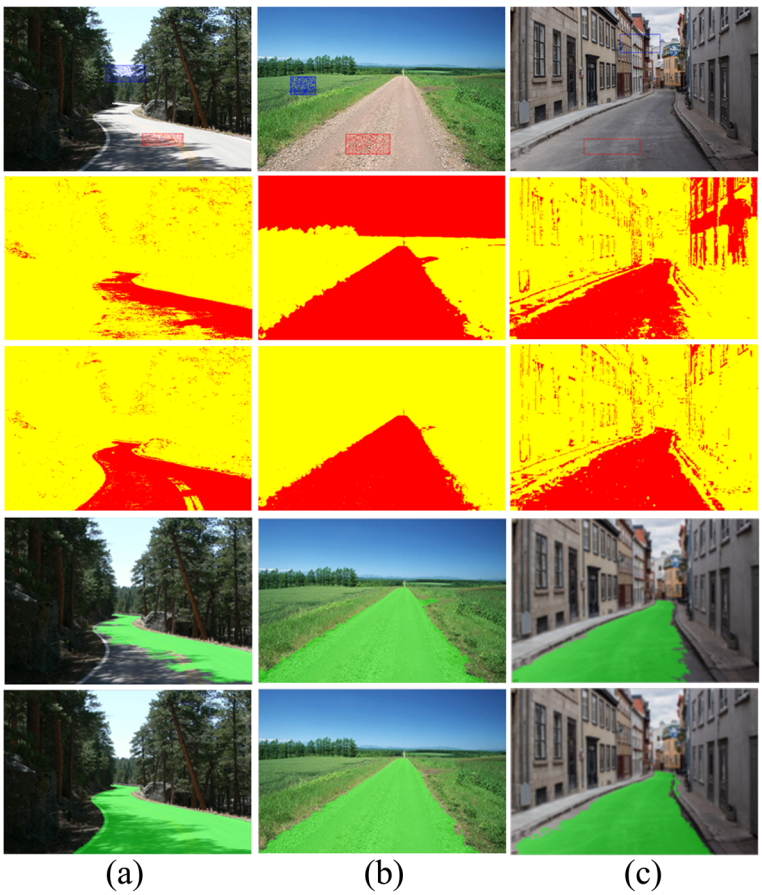

Fig. 13 a–c Comparison of results between offline learning and online learning

distance between the two vehicles. This is basically due to the fact that as the vehicle moves away, its relative size in the image decreases, affecting the image finite resolution that becomes more important. In order to improve the estimates of the relative position for further objects, one can consider the usage of additional camera with a narrow field of view, mounted above this one. The overall task of tracking and position estimation could be done by proper fusion of data collecting from these two cameras.

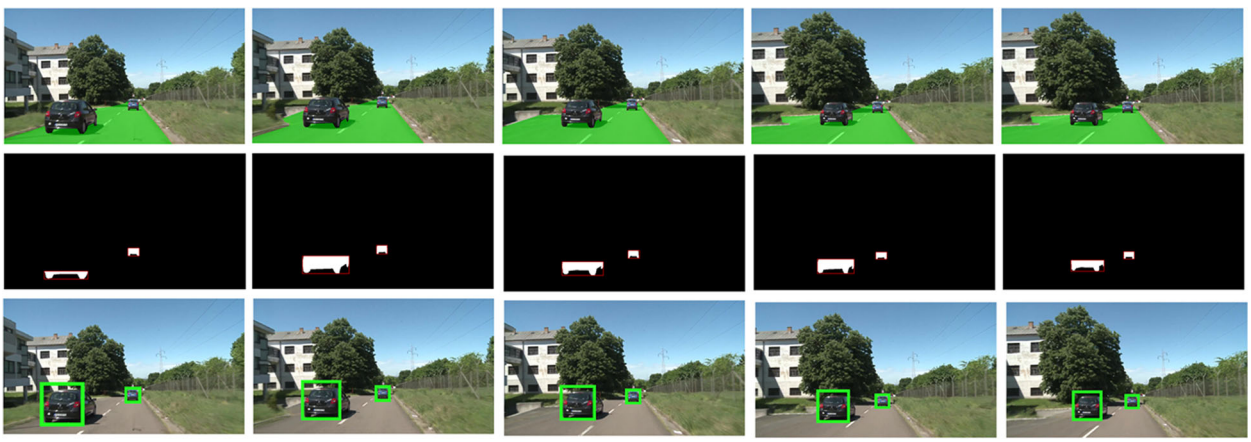

Fig. 14 On-road obstacles extraction

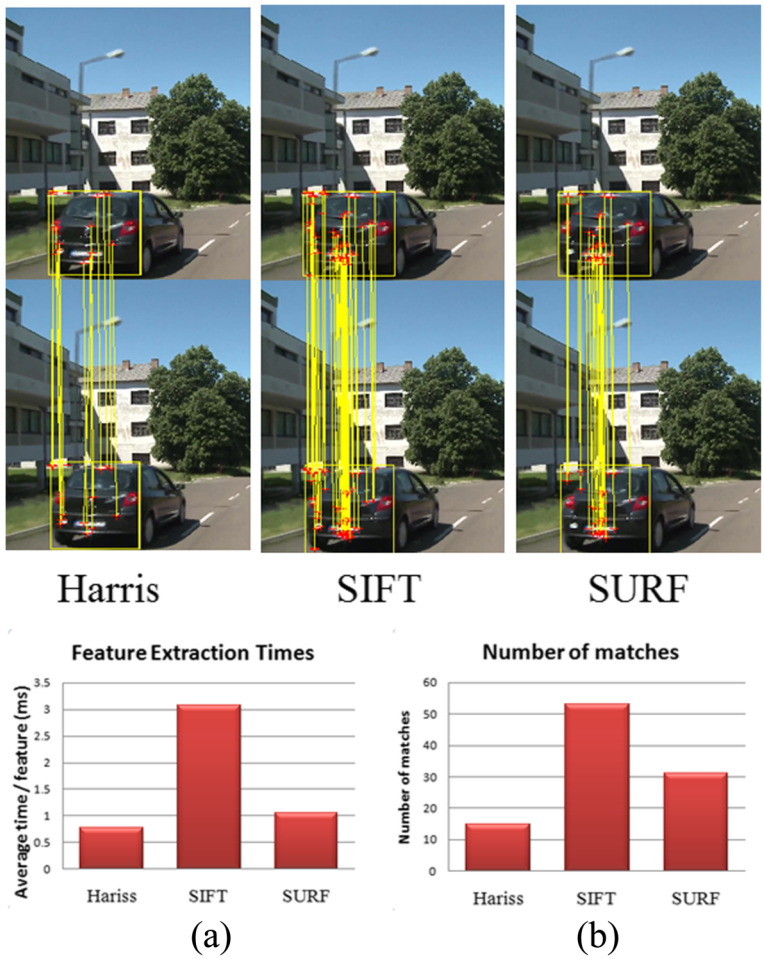

Fig. 15 a, **b** Comparison between three different tracking methods: Harris detector, SIFT, and SURF

3.4.2 Simulation results of relative velocities calculation

Figure 18 shows a comparison between the real and the estimated relative positions and velocities of a simulated moving obstacle on the road. It may be noted that the actual relative velocity was uniform ($V = -8.75$ m/s). The estimated relative velocity oscillates around this value. The error is increasing with the distance between the two vehicles.

A simple moving average filter is applied to solve this problem while its dimension (window size) has been

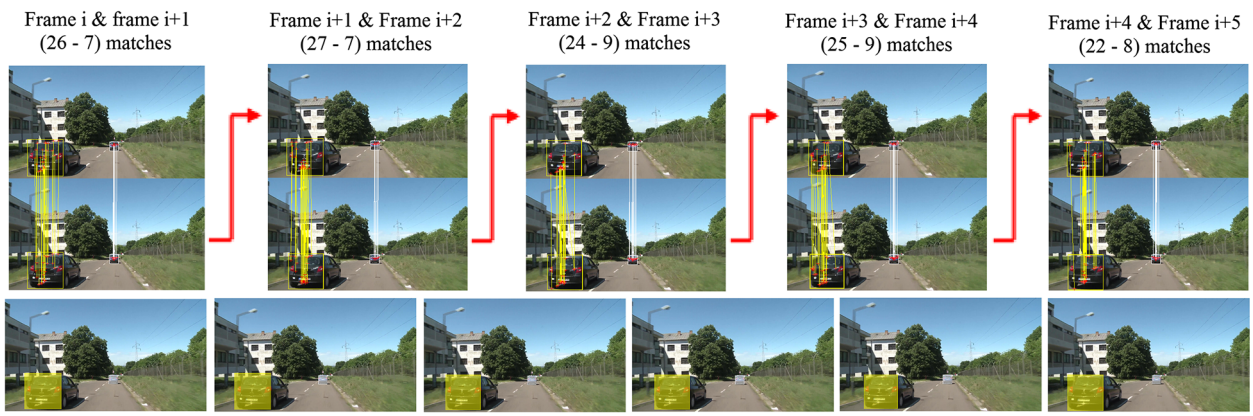

Fig. 16 Experimental results of the on-road obstacles tracking

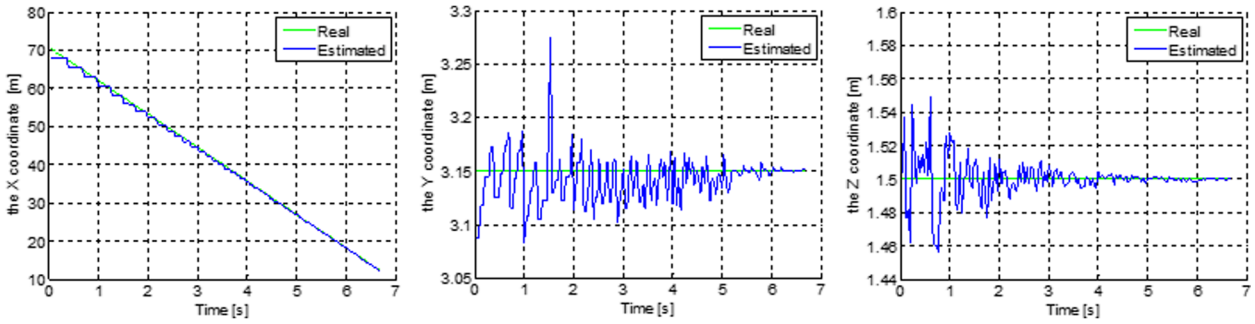

Fig. 17 Relative position estimation

dynamically adapted according to the estimated distance as it is shown in Fig. 19.

3.4.3 Simulation results of the classification

For the specified scenario including four different vehicles as shown in Fig. 20. In the sequence of frames with 400 [ms] inter-frame interval, the comparison results between the estimated and real classification are shown in f and g, respectively. Three areas can be distinguished: the red zone for incoming vehicles, green zone for outgoing ones, and the white area for stationary obstacles/vehicles. Colors of curves correspond to the color of the vehicles. The estimated results are consistent with actual data.

The illustration of detecting potentially incoming obstacles coming from lateral direction is given on Fig. 21. The first row represents the sequence of four images illustrating the potential obstacles in the strip aside the road. Some of them are stationary while the others are moving toward the road or out of the road. The result of detection of groups of pixels representing the potential obstacles is given in the second row. The third row of Fig. 21 is consisting from the original sequence where the object's labels and the calculated data about their positions and lateral velocities are superimposed. The fourth row distinguishes the "dangerous" objects moving toward the road, as it is also visible from the fifth row where the lateral velocities of all objects are represented, while two of them are moving in direction toward the road.

3.4.4 Experimental results of the classification

Figure 22 shows a real scenario of a road traffic, presented by a sequence of digital images captured by a monocular camera mounted on a mobile vehicle. In this sequence, the inter-frame interval is 400 [ms]. Each frame contains two vehicles (black and blue) which are moving with different speeds. Figure 22f represents the estimation results of the relative distance for each vehicle. The vehicles velocities

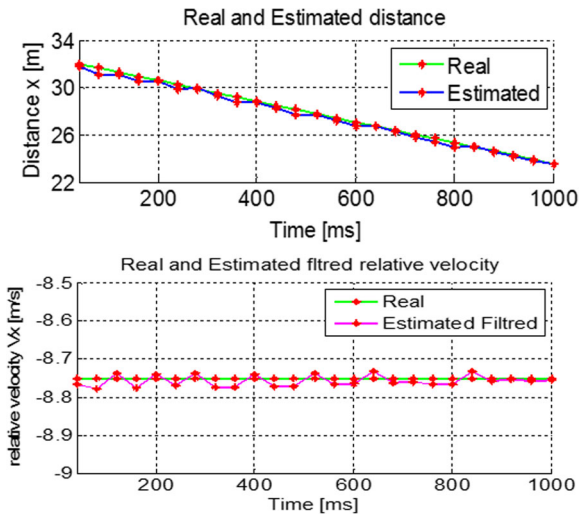

Fig. 18 Comparison between the real and estimated relative positions and velocities

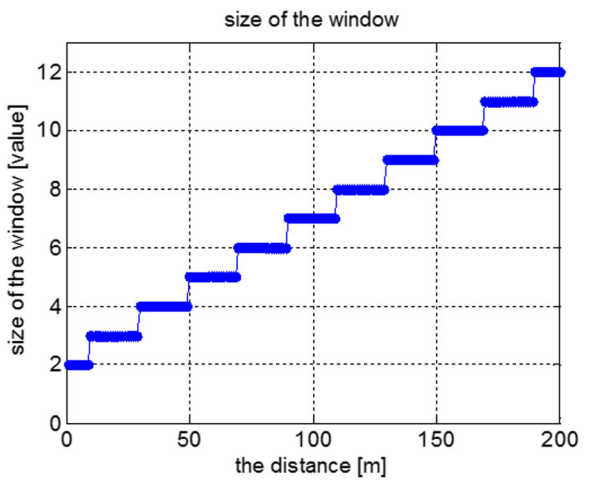

Fig. 19 Size of averaging filter in dependence to distance from the obstacle

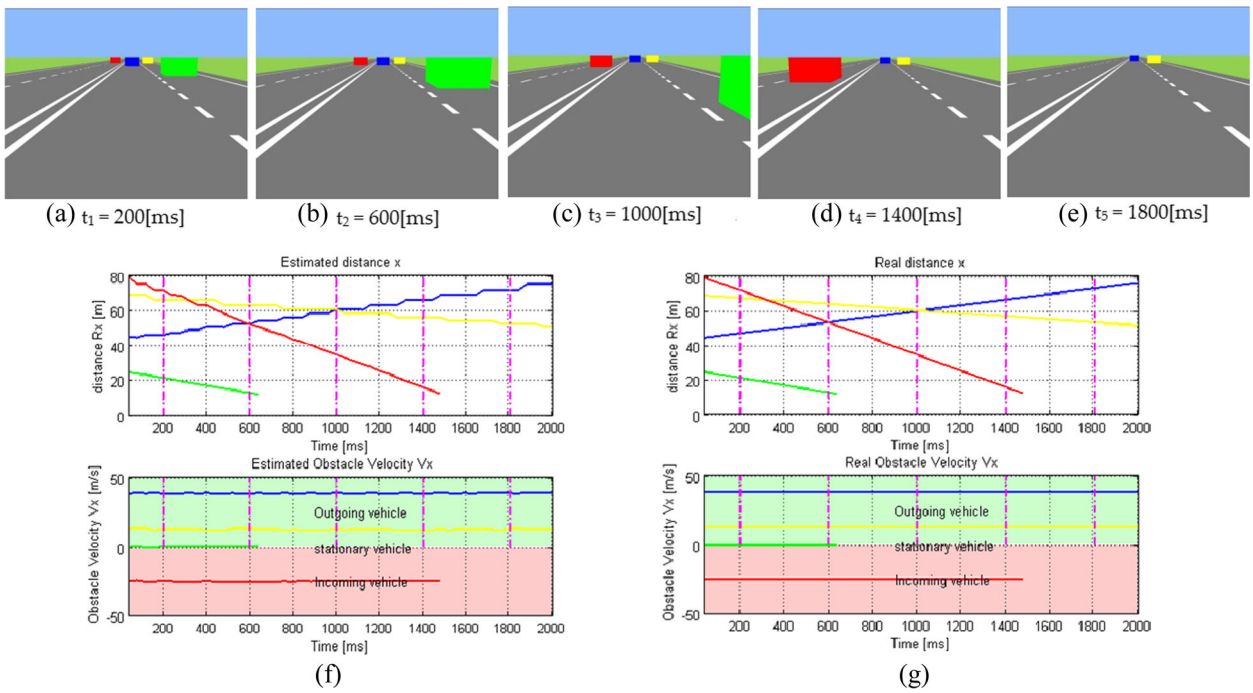

Fig. 20 a–g Classification of the on-road obstacles (synthesized sequence of frames)

and their classifications are shown in Fig. 22g where it can be seen that both of them represent an outgoing obstacle.

4 Conclusions

The algorithm of an automatic classification of the on-road obstacles according to their relative velocities is presented here as a prerequisite for an application of the automatic control system for obstacle avoidance. The on-road obstacles have to be detected firstly, then described properly in order to enable their tracking from frame to frame. Our choice on these two steps have been oriented toward rather complex methods: SVM based on eight component vector (color + texture) for the recognition of a road area and SURF based on 64 component vector for the description and tracking of characteristic points inside the tracking windows. These steps are verified using the realistic road-traffic images. The effects of choice of these complex methods onto the accuracy of detection and tracking have been shown partially, comparing them with the simpler approaches in road detection and obstacle's characterization, and showing the superiority. Consequently, the higher computational cost must be paid and the ability to implement the algorithm in real time might be compromised. Our further research will be oriented more toward the algorithm implementation based on recent results in parallel processing and new types of image coding given in [25–28].

The final step of verification that was related to the estimation of distances to the obstacles and their rates of change was made using the synthesized sequences representing the simulated motion of the camera and multiple vehicles as well as on the sequences from the real driving.

These results have shown highly acceptable accuracy of estimated relative velocities of obstacles. The automatism of this algorithm is reduced by the very first requirement that the operator should point onto the regions in the image which are typical representatives of road and non-road, but from that point on, nothing is required as a human intervention. A priory knowledge of some road measures, as lane-width, could be easily provided from the vehicle global positioning system and digital map of the road. A number of practically important algorithm parameters have been analyzed and specified. They are regarding to the part of the field of view usable for basic orientation on the road, minimal size of tracked vehicles, minimal correspondence of the characteristic points required for the reliable tracking of obstacles, the size of an averaging filter used in estimation of relative velocity, etc. The future work will be oriented toward further verification of the algorithm using controlled experiments in real road-traffic situations.

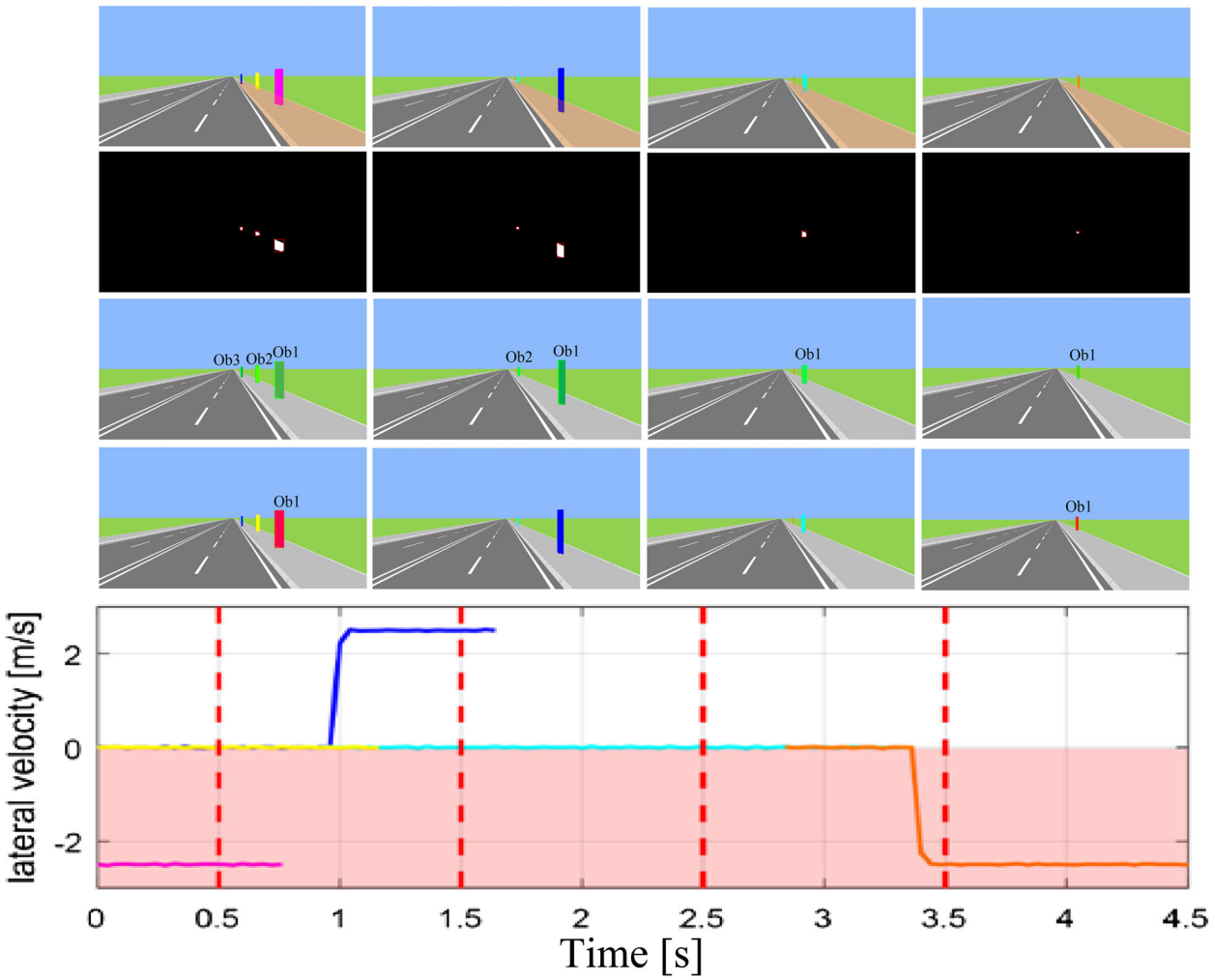

Fig. 21 Classification of potential obstacles (synthesized sequence of frames)

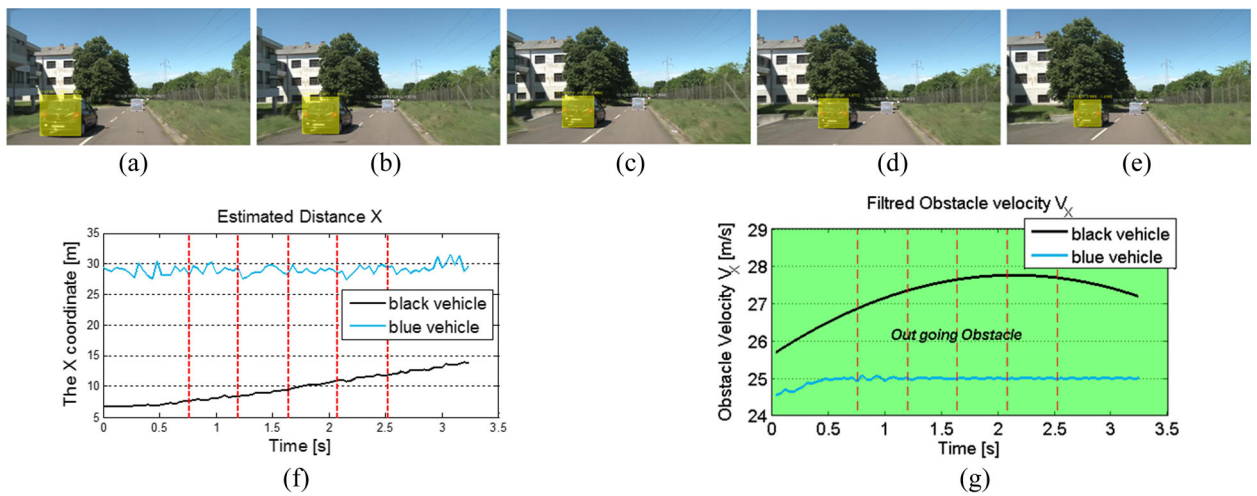

Fig. 22 a–**g** Classification of the on-road obstacles (real scenario)

Acknowledgements
The work is a part of research related to the application of inverse proportional navigation in scenario of avoiding of the on-road obstacles, leading toward Ph.D. thesis of the first author, on Military Academy, University of Defence, Belgrade, Serbia.

Authors' contributions
MB is a leading author for all parts of paper, SG is a supervisor of research, and MAB has assisted in experimental part of work. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Military Academy, University of Defence, Belgrade, Pavla Jurisica Sturma 33, Belgrade, Serbia. [2]School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, Belgrade, 11020 Belgrade, Serbia.

Received: 15 September 2016 Accepted: 19 November 2016
Published online: 05 December 2016

References

1. C Demonceaux, D Kachi-Akkouche, in *Proceedings of the IEEE Intelligent Vehicles Symposium 2004*. Robust obstacle detection with monocular vision based on motion analysis (IEEE, Parme, 2004), pp. 527–532. doi:10.1109/IVS.2004.1336439
2. A Broggi, M Bertozzi, A Fascioli, M Sechi, in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511)*. Shape-based pedestrian detection (IEEE, Dearbon, 2000), pp. 215–220. doi:10.1109/IVS.2000.898344
3. M Lützeler, E Dickmanns, in *Proceedings of the IEEE Intelligent Vehicles Symposium 98*. Robust road recognition with marveye (IEEE, Stuttgart, 1998), pp. 341–346
4. A Kuehnle, in *Proc. SPIE, Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS*. Symmetry-based vehicle location for AHS, vol. 2902 (SPIE, Orlando, 1997), pp. 19–27. doi:10.1117/12.267158
5. SS Teoh, T Bräunl, Symmetry-based monocular vehicle detection system. Mach. Vis. Appl. **23**(5), 831–842 (2012)
6. W Kruger, W Enkelmann, S Rossle, in *Proceedings of the Intelligent Vehicles' 95 Symposium*. Real-time estimation and tracking of optical flow vectors for obstacle detection (IEEE, Detroit, 1995), pp. 304–309. doi:10.1109/IVS.1995.528298
7. C Braillon, C Pradalier, JL Crowley, C Laugier, in *IEEE Intelligent Vehicles Symposium 2006*. Real-time moving obstacle detection using optical flow models (IEEE, Tokyo, 2006), pp. 466–471. doi:10.1109/IVS.2006.1689672
8. SA Mahmoudi, M Kierzynka, P Manneback, K Kurowski, Real-time motion tracking using optical flow on multiple gpus. Bull. Pol. Acad. Sci. Tech. Sci. **62**(1), 139–150 (2014)
9. G Lefaix, T Marchand, P Bouthemy, in *Object recognition supported by user interaction for service robots*. Motion-based obstacle detection and tracking for car driving assistance, vol. 4 (IEEE, Quebec, 2002), pp. 74–77. doi:10.1109/ICPR.2002.1047403
10. M Bertozzi, A Broggi, Gold: A parallel real-time stereo vision system for generic obstacle and lane detection. IEEE Trans. Image Process. **7**(1), 62–81 (1998)
11. Q Yu, H Araújo, H Wang, A stereovision method for obstacle detection and tracking in non-flat urban environments. Auton. Robot. **19**(2), 141–157 (2005)
12. N Bernini, M Bertozzi, L Castangia, M Patander, M Sabbatelli, in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. Real-time obstacle detection using stereo vision for autonomous ground vehicles: a survey (IEEE, Qingdao, 2014), pp. 873–878. doi:10.1109/ITSC.2014.6957799
13. R Labayrade, D Aubert, Robust and fast stereovision based obstacles detection for driving safety assistance. IEICE Trans. Inf. Syst. **87**(1), 80–88 (2004)
14. D Kuan, G Phipps, A-C Hsueh, Autonomous robotic vehicle road following. IEEE Trans. Pattern Anal. Mach. Intell. **10**(5), 648–658 (1988)
15. S Graovac, A Goma, Detection of road image borders based on texture classification. Int. J. Adv. Robot. Syst. **9**, 1–12 (2012)
16. S Zhou, J Gong, G Xiong, H Chen, K Iagnemma, in *Intelligent Vehicles Symposium (IV), 2010 IEEE*. Road detection using support vector machine based on online learning and evaluation (IEEE, San Diego, 2010), pp. 256–261. doi:10.1109/IVS.2010.5548086
17. C-C Chang, C-J Lin, Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST). **2**(3), 27 (2011). doi:10.1145/1961189.1961199
18. S Krig, in *Computer Vision Metrics: Survey, Taxonomy, and Analysis*. Interest point detector and feature descriptor survey (Apress, Berkeley, 2014), pp. 217–282. doi:10.1007/978-1-4302-5930-5_6
19. C Harris, M Stephens, in *Proceedings of the Alvey Vision Conference*. A combined corner and edge detector (CiteSeerX, Manchester, 1988), pp. 147–152. doi:10.1.1.231.1604
20. DG Lowe, in *The proceedings of the seventh IEEE international conference on Computer vision, 1999*. Object recognition from local scale-invariant features, vol. 2 (IEEE, Kerkyra, 1999), pp. 1150–1157. doi:10.1109/ICCV.1999.790410
21. DG Lowe, Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
22. H Bay, A Ess, T Tuytelaars, L Van Gool, Speeded-up robust features (surf). Comp. Vision Image Underst. **110**(3), 346–359 (2008)
23. D Helly M, G Vaibhav, Real-time moving object detection using surf. IOSR Journal of Computer Engineering (IOSR-JCE). **17**(3), 75–78 (2015)
24. K Kanatani, *Geometric computation for machine vision*. (Oxford University Press, Inc, New York, 1993)
25. C Yan, Y Zhang, J Xu, F Dai, L Li, Q Dai, F Wu, A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors. IEEE Signal Proc. Lett. **21**(5), 573–576 (2014)
26. C Yan, Y Zhang, J Xu, F Dai, J Zhang, Q Dai, F Wu, Efficient parallel framework for HEVC motion estimation on many-core processors. IEEE Trans. Circ. Syst. Video Technol. **24**(12), 2077–2089 (2014)
27. C Yan, Y Zhang, F Dai, X Wang, L Li, Q Dai, Parallel deblocking filter for HEVC on many-core processor. Electron. Lett. **50**(5), 367–368 (2014)
28. C Yan, Y Zhang, F Dai, J Zhang, L Li, Q Dai, Efficient parallel HEVC intra-prediction on many-core processor. Electron. Lett. **50**(11), 805–806 (2014)