Traffic road obstacles detection based on analysis of relative motion vectors

Ion Giosan, Emőke Olti, Sergiu Nedevschi
Computer Science Department
Technical University
Cluj-Napoca, Romania
ion.giosan@cs.utcluj.ro, emokeolti91@gmail.com, sergiu.nedevschi@cs.utcluj.ro

***Abstract*—Computer vision obstacle detection on either road lanes or sidewalks is very important for traffic participants. This paper presents an approach for obstacle detection from sequences of consecutive monocular color image frames. Key-points are uniformly distributed in a grid structure on each input image. A Lucas-Kanade optical flow algorithm is performed between each pair of consecutive frames, on the considered key-points, in order to compute the relative motion vectors. Background movement estimation is computed across frames using a RANSAC procedure. Optical flow vectors that are belonging to the background are filtered out. The others are considered to be within the obstacles and are grouped by a hierarchical clustering algorithm in separate obstacles by analyzing their locations, angles and magnitudes. Spurious clusters with low number of motion vectors are filtered out. Finally, an imminent collision warning is issued both visually and acoustically when an obstacle is detected to be too close and it is about to crash with the ego-camera.**

Keywords-obstacle detection; motion vectors; hierarchical clustering; background movement estimation; collision alert.

I. INTRODUCTION

Nowadays, the number of intelligent vehicles and smart devices is growing rapidly due to the technological possibilities that are into a continuous development process. In case of an intelligent vehicle, the driver is alerted by a driving assistance system when there appear potentially dangerous situations. Usually it includes many safety functions like obstacle collision warning, lane departure warning, lane keeping assistance, speed keeping assistance, etc. There are also other traffic participants like vision deficiency persons [1] who can be guided or warned on the sidewalks by a similar system. This could be possible if they carry on a smart device with a specific application installed which performs environment understanding.

Here resides the motivation for building high accuracy obstacle detection module that can help either drivers or persons with vision deficiency or even blind people. An obstacle detection module must determine the regions of interest from traffic scene where exist obstacles. It may also provide this information to a subsequent module which may classify it into a specific obstacle class like pedestrians, poles, trees, vehicles, wall etc. Another important aspect for the detected obstacles is to find their exact location within the traffic scene or at least to infer if there may be an imminent collision with the ego-vehicle or with the person who carries the smart-device.

There are many different technologies like LASER-scanners, RADAR, infrared sensors, ultrasound sensors and video cameras that can be used for obtaining the scene information. However, video cameras are capable of acquiring visual information that can be further processed for environment understanding. This way of getting the traffic scene information is no pollutant for the environment, being also similar to the human eyes vision system. Depth computation of scene elements is also very important. This can be done basically by using a stereo-cameras setup. We don't use a stereo-setup due to the fact that it implies higher costs and usually smart-devices don't have integrated stereo cameras. We use a single color camera setup, with low costs, which offers us sufficient information for implementing all the processing operations.

We propose an approach for detecting the obstacles either on lanes or sidewalks by analyzing the relative motion vectors between frames. Optical flow is computed on uniformly distributed grid points by Lucas-Kanade algorithm. The result defines the relative motion vectors which are then used for both background movement estimation and obstacles segmentation. A RANSAC algorithm is used for determining the background movement in order to be filtered out from the motion vectors field. The obstacles are defined by clusters of motion vectors obtained after a two-step hierarchical clustering procedure considering their specific features: location, magnitude, and angle. Noisy clusters are finally removed obtaining the valid obstacles. The possibility of imminent collision of each obstacle with the ego-camera is evaluated and signalized. We also present the evaluation of the entire obstacle detection system.

II. RELATED WORK

Obstacle detection from a video sequence is one of the most important task used in many real time automotive applications. The researchers carry out a lot of work for developing very good solutions for obstacle detection from both monocular and stereo vision images.

In case of stereo vision systems the task is easier than in monocular systems. Stereo systems acquire much more traffic scene information using at least two cameras. This allows the obstacle detection [2] to be done by analyzing both

color/intensity and depth information [3] which considerably reduces the amount of noisy information. The approach is continued with a module that gathers the reconstructed points into obstacles by using a paradigm of points grouping [4] and density maps [5], followed by optical flow and motion computation for obstacle tracking [6].

In our approach we use a single color camera. In this case of monocular vision, common features like color or gray intensities [7], symmetry [8], edges [9], shadows [10] and textures [11] are widely used for obstacle detection. Motion from optical flow [12] may also be computed for detecting moving obstacles by subtracting the ego motion of the vehicle.

Detection and segmentation of moving objects can be achieved in several ways. There are region-based approaches, boundary based approaches and combinations of them. In the region based approach, the image is partitioned into connected regions by grouping pixels of similar intensity levels situated in the same neighborhood. Adjacent regions are merged under some constraints involving perhaps homogeneity or sharpness of region boundaries. The region based approach includes the traditional background subtraction and optical flow methods. In [13-15] region based techniques for are successfully used for moving objects detection. The methods that are used in the region based approach are background subtraction [13], Markov random field (MRF) [16], change detection mask (CDM) [17], clustering [18] and statistical methods [15].

In [19] the idea of multiscale regions is used for better obstacle detection. It takes a pixel neighborhood for filtering out the possible perturbations of few pixels in that vicinity. An iterative background subtraction method is also proposed, starting from large rectangular regions and reducing them to narrower regions by means of color histogram analysis. These histograms are robust to presence of local noise and they could be built recursively: a large region histogram from combining the component small regions histograms. The motion is detected by comparing the corresponding rectangular regions from the largest scale to the smallest one. At smaller scales, the comparisons are made just in case when there exists a detected motion at a higher scale. The advantage consist in reducing the number of false detections. The background is multiscale modeled by color histograms of each rectangular region. Its model is periodically updated and a similarity measure is used for comparing the current frame with the background.

Boundary based methods deals with the discontinuities in the images. These approaches use active contour, edge based and optical flow methods. Different algorithms and methods are used for the detection of moving objects. Gradient based optical flow together with an edge detector represent a good choice for a precise speed computation, although they are not so frequently used in systems that detect and track the moving obstacles. In [20], a boundary based detection method which uses the optical flow computation is presented. The solution is based on lines computation through gradient-based optical flow and edge detection. First, all edges are extrapolated into lines and those which are belonging to the background are then removed. The obstacles contours are extracted. Each obstacle is detected and tracked in order to make the approach robust to occlusions and interferences.

A similar method that uses boundary as a feature, but finds a probabilistic model it is proposed in [21]. A Bayesian classifier is used for image local motion representation and classification. It uses two basic models: translation motion and motion contours. The motion contours are defined by a non-linear generative model which contains the contour orientation, the speed on both sides, the movement of occlusive edge in time and the appearance/disappearance of the contour pixels. The a-posteriori probability density function over the model parameters is computed and propagated in time with a specific filtering algorithm.

A combination of region-based and boundary based methods it is also possible. In [22, 23], [16] the ideas are evenly merged taking the advantages from both and trying to remove the disadvantages as much as possible in order to obtain a robust solution. The obstacle detection approaches can also be classified in techniques for detecting fast obstacles and techniques for detecting slow obstacles. It is proven that the slower obstacles' segmentation is more difficult.

An object is moving if its scene position is changing in time. Given a traffic scene, an object is moving if it changes its relative position in relation with other objects between two different well defined timestamps. Based on these assumptions, we have to find the image points that are moving between two consecutive frames. Every video sequence contains multiple image frames, each frame with its recorded timestamp. In order to detect the obstacles that have a relative motion, we can compare the two successive frames and analyze the difference between them. Our proposed solution is region based and uses the clustering of moving adjacent points by their characteristics for detecting the moving obstacles.

III. System Architecture

Road obstacle detection system architecture with all its component modules is depicted in Figure 1.

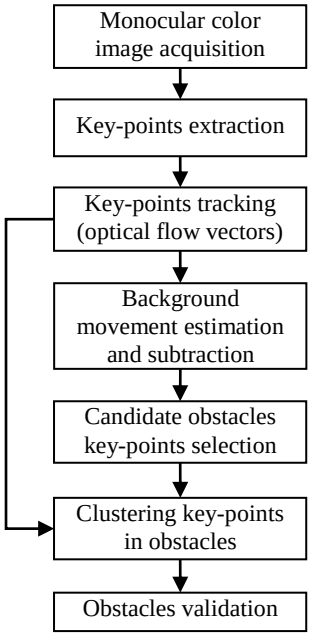


Figure 1. Obstacle detection system architecture based on analysis of relative motion vectors

IV. Key-Points Extraction

The first step of the obstacle detection approach consists in key-points' extraction and selection. An image contains a way too large number of pixels that could not be all considered for computing and analyzing the relative motion across frames. Due to this fact we have to make a selection of pixels that will be considered in further processing. Our

objective is to compute first the background movement between each two consecutive frames.

The key-points may be extracted by selecting those pixels with specific robust features that can be easily tracked across frames (e.g. SIFT features). There will be many key-points on textured surfaces and almost no point on uniform color surfaces (which usually appears on both background composition and people clothes). This may lead to a very difficult background movement estimation due to the lack of key-points.

The idea for selecting the key-points is the usage of a uniform grid which doesn't take into account the local features. Its granularity is τ described by the following equation:

$$\tau = \frac{W * H}{N} \tag{1}$$

where W and H are the image width and height and N is a parameter that specifies the total number of key-points that is considered.

In Figure 2 the difference between considering the key-points based on SIFT features and the uniform grid key-points is shown. A remark is that the number of SIFT based key-points is very low on the uniform textured areas.

a)

b)

Figure 2. Key-points' selection – with red dots: a) SIFT features based; b) uniform grid based.

The number of considered key-points is very important. If we choose a very high number then the further processing time could be too high and inacceptable. In the opposite situation, when the number of key-points is very low, the obstacles detection by means of relative motion could be very difficult. A solution of compromise between performance and execution time is to set about N=1500 key-points for low-medium resolution video images (352x288 pixels).

V. KEY-POINTS TRACKING

The next step in our algorithm consists in the computation of each key-point movement in sequences of consecutive frames. It means that knowing the position of each key-point in the current frame, we would like to find the position of the same key-point in the next frame. This could be achieved by computing the optical flow which represents the pattern of apparent motion of scene elements caused by the relative motion between the camera and the scene. It is computed by considering that the intensity of a scene point stays constant even if its position is changed from one frame to another:

$$I(x,y,t) = I(x + v_x(x,y), y + v_y(x,y), t + 1) \tag{2}$$

Using a first order Taylor approximation we obtain:

$$\begin{aligned} I(x,y,t+1) &= I'_x(x,y,t)v_x(x,y,t) + \\ &+ I'_y(x,y,t)v_y(x,y,t) + I_t(x,y,t) \end{aligned} \tag{3}$$

The solution of the equation above represents the optical flow. Having a single equation with two unknown variables, supplementary constraints must be added. Lucas-Kanade algorithm assumes that the optical flow and the illumination are constant in each key-point and also in a window around that point. The solution of the equation is found by using the least squares method:

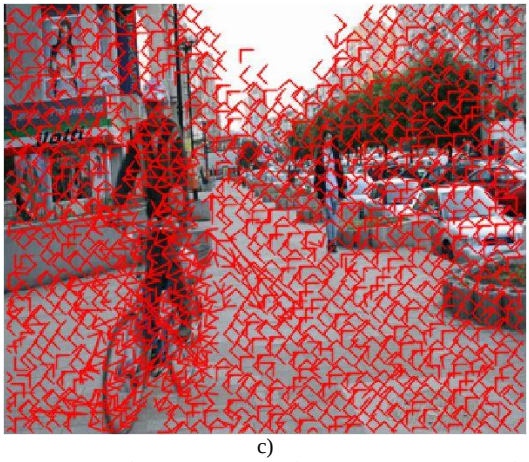
$$\begin{aligned} &I'^{1}_x v_x + I'^{1}_y v_y = -I'^{1}_t \\ &I'^{2}_x v_x + I'^{2}_y v_y = -I'^{2}_t \\ &\dots \\ &I'^{n}_x v_x + I'^{n}_y v_y = -I'^{m}_t \end{aligned} \tag{4}$$

The stability of this method is given by the matrix A^TA (where A is the system matrix of the above equations set). If it doesn't have two eigenvalues high enough, the system could not be solved. This condition is equivalent with the existence of some edges on two orthogonal directions in the considered neighborhood window.

Optical flow vectors computation using Lucas-Kanade algorithm is depicted in Figure 3. They are characterized by their magnitude and motion angle. These two values are further used for finding and grouping the obstacles' points and for removing the noise. If $P_1(x_{1i},y_{1i})$ is a point in the previous frame and $P_2(x_{2i},y_{2i})$ is the correspondent in the current frame, then the optical flow features are computed using the following formulas:

a)

b)



c)

Figure 3. Sample of computed optical flow vectors: a) previous frame; b) current frame; c) motion vectors on the current frame

$$v_{ix} = x_{2i} - x_{1i}, v_{iy} = y_{2i} - y_{1i}$$
$$D_{i(1,2)} = \sqrt{v_{ix}^{\ 2} + v_{iy}^{\ 2}}, i = \overline{1,n} \quad (5)$$
$$\theta_{i(1,2)} = \arccos \frac{v_{ix}}{D_{i(1,2)}}, \theta \in [0, 2\pi]$$

where $D_{i(1,2)}$ and $\theta_{i(1,2)}$ are the motion magnitude and respectively the motion angle of the optical flow vector.

VI. Background Movement Estimation

In the previous section we described the individual motion of every key-point. Now, we have to extract only those points that moved more or less than the general scene background movement. These key-points that have a different motion than the background are considered to be obstacle point candidates.

General background movement refers to all the pixels motion that are not obstacle pixels. Considering a point P_1 in the previous frame and applying the general background movement rule H we can estimate the location in the current frame as being the point P_{2est} (assuming that it is a background pixel). A transformation rule H is correct if the location difference between the estimated location P_{2est} and the real location P_2 is as small as possible.

We have to compute the transformation matrix H that is used for estimating the background pixels locations in the current frame knowing the previous frame:

$$\begin{bmatrix} x_{2i}^{\ est} \\ y_{2i}^{\ est} \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \end{bmatrix} * \begin{bmatrix} x_{1i} \\ y_{1i} \\ 1 \end{bmatrix} \quad (6)$$

The location estimation error is computed with the following equation (using Manhattan distance):

$$E = \left| p_{2i}^{\ est} - p_{2i} \right| = \left| x_{2i}^{\ est} - x_{2i} \right| + \left| y_{2i}^{\ est} - y_{2i} \right| \quad (7)$$

If the error E is greater than an empirically chosen threshold then we assume that the point doesn't respect the general background movement meaning that it has its own movement and being considered as an obstacle point candidate.

We use a RANSAC approach for computing the transformation matrix parameters from equation (8). The input data set consists in pairs of key-points locations from previous frame and the corresponding locations in the current frame computed with the optical flow.

$$H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \end{bmatrix} \quad (8)$$

The algorithm has several iterations. In every iteration it randomly selects pairs of corresponding points, computes the matrix H and the associated total error E. After N iterations we select the matrix H with the lowest error E. The matrix H has six parameters. With a pair of corresponding points, a set of two equations can be written:

$$\begin{aligned} x_{2i}^{\ est} &= h_{00} * x_{1i} + h_{01} * y_{1i} + h_{02} \\ y_{2i}^{\ est} &= h_{10} * x_{1i} + h_{11} * y_{1i} + h_{12} \end{aligned} \quad (9)$$

At each iteration, three random pairs of corresponding points must be taken in order to deterministically find the parameters of matrix H that defines the general background movement. According to equation (7) the error is computed for each point. We set two thresholds: a low one T_l and a high one T_h. If the error is below the low threshold then the point belongs to background. If the error is above the high threshold then it is considered as being noise. An obstacle point is found if the corresponding movement error is between T_l and T_h:

$$T_l \le E \le T_h \quad (10)$$

We empirically set the values T_l=4 and T_h=30 pixels. A sample of optical flow vectors that are assigned to obstacles is depicted in Figure 4.

Figure 4. Sample of optical flow vectors assigned to obstacles – with red arrows

VII. Candidate Motion Vectors Clustering in Valid Obstacles

In this section, we present our proposed method for clustering the motion vectors in separate obstacles. After removing the background key-points, the remaining ones are clustered together starting from the minimum feature distance to the maximum feature distance between them. We use a hierarchical clustering algorithm which merges elements based on a specific feature similarity. This is an iterative approach. Initially, each point is assigned to a different class. At each iteration, two or more classes are merged together in a new class. The merging process will continue until the minimum distance between every two classes is greater than a predefined threshold.

Feature similarity computation is relatively easy. It will be computed between every two candidate obstacle points, taking into account the location, the magnitude and angle of their optical flow vectors. The first iteration is trivially and consists in merging every pair of two points which are the closest in a new class. After this step, we have to re-compute the distance from the other classes to this new class and so on. There are several possibilities for defining the distance between classes. We define the distance between two classes as being the minimum distance between any pair of points (one from the first class and the other from the second class).

We propose a two-step clustering: in the first one we cluster the motion vectors based only on their location. The distance between a pair of motion vectors is defined as the Euclidean distance between their origins. A threshold must be set in order to obtain good obstacle clusters. In the case of having N=1500 key-points, in an image with medium resolution (352x288 pixels), the threshold T that defines the maximum allowable distance between two classes that could be merged is set to T=20 pixels.

It isn't enough for a good clustering to take into account only the location feature. There may appear many kind of errors both in the optical flow computation and background subtraction. In order to remove these errors we have to extract the common member features within each cluster and to remove those instances that differ. A common feature for all the points belonging to the same object is the angle of the motion vectors. Another common feature is the magnitude of the motion vectors. Within each class, the points should have approximately the same motion angle and magnitude. We propose a histogram based approach in order to filter out the spurious points in each obstacle class.

We compute a bi-dimensional histogram: magnitude-angle histogram. The maximum value from this histogram is then extracted and the pair (magnitude, angle) corresponding to this maximum is considered as being the reference ($Magn_{class}$, $Angle_{class}$) for that obstacle motion vectors cluster. All the other points i, with corresponding motion vectors that differ in magnitude or angle than the reference are filtered out. The differences ($E_{magn,i}$ and $E_{angle,i}$) are computed with the Manhattan distance:

$$\begin{aligned} E_{magn,i} &= | Magn_{class} - Magn_i | \\ E_{angle,i} &= | Angle_{class} - Angle_i | \end{aligned} \tag{11}$$

We set two empirical thresholds T_{magn}=5 pixels and T_{angle}=45º on the previously computed errors. A point is filtered out if at least one of the following conditions is true:

$$\begin{aligned} E_{magn,i} &> T_{magn} \\ E_{angle,i} &> T_{angle} \end{aligned} \tag{12}$$

In the case of the motion vectors angle, a higher value than 45º may also be set for the threshold T_{angle} due to the fact that if an obstacle is closer to the camera and moving with high speed, there may appear motion vectors with a slight different angle on its surface (see Figure 5).

Figure 5. Differences in motion vectors angles (highlighted with blue arrows) of the same obstacle

After removing the spurious points, we re-apply the clustering procedure. This is necessarily for ensuring that there aren't clusters having the within members distance greater than the maximum admitted distance. Such a case is depicted in Figure 6.

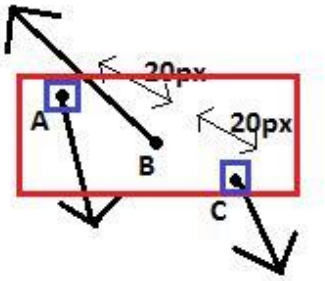


Figure 6. Three motion vectors with their magnitudes and angles corresponding to the points *A*, *B* and *C*

In the first iteration, the points *A*, *B* and *C* are grouped together in the same cluster. Suppose that in the filtering procedure, the point *B* is removed. The minimum distance between *A* and *C* is now greater than the allowed distance. The re-clustering will solve this problem and split the points *A* and *C* in two separate classes.

The resulting obstacle key-points cluster set may contain instances with very small number of points. These are considered to be noise and must be removed. We set a threshold T_{points}=10 representing the minimum number of key-points that defines a valid obstacle *i*, according to the following condition:

$$\left\| KeypointsSet \right\|_i > T_{points} \tag{13}$$

VIII. Imminent Collsion Obstacle Classification

The resulting valid obstacles are going to be classified by their possible imminent collision degree. It is difficult to define a general rule. We take into account the obstacle position relative to the camera, defining the following cases:

- The obstacle is approaching the camera with a high risk of imminent collision
- The obstacle is either approaching to or departing from the camera with low risk of imminent collision

First, we have to define the obstacle position. Basically it may be considered as being its center of mass (x_{center}, y_{center}). We define the obstacle area as a bounding rectangle (x_{min}, x_{max}, y_{min}, y_{max}) circumscribed to its key-points. In our monocular vision, taking into account the perspective view, the lower part of the obstacle that appears closer to the camera has to be considered for evaluating the imminent collision risk. The obstacle center is defined as follows:

$$\begin{aligned} x_{center} &= (x_{\min} + x_{\max}) * \frac{1}{2} \\ y_{center} &= (y_{\min} + y_{\max}) * \frac{2}{3} \end{aligned} \tag{14}$$

Scene elements that are relatively further away from the camera don't represent danger, the imminent collision risk being minimal. The scene image is divided in three equal sections, the lowest image part (see Figure 7) being the region with the highest risk of imminent collision. An obstacle is considered to be in this part if the following inequality is true:

$$\begin{aligned} & y_{center} < y_{image}, \\ & y_{image} = height_{image} * \frac{2}{3} \end{aligned} \tag{15}$$

Figure 7. High risk of imminent collision area – with red dots

The obstacle may enter in the high risk collision area but there might be no imminent collision. The most critical point is the one marked in Figure 7 with an "x" situated on the last line in the middle of the image. We use the motion vector of the entire object to determine if it is or not a dangerous object. We also compute the reference critical direction for each obstacle as being the segment that connects the most critical point to the center of the obstacle. If the obstacle's motion vector has a small angular deviation from the reference critical direction then it is consider a critical one. An angular deviation threshold T_{angdev}=15º is set for taking this decision. An obstacle *i* is inside the critical angle section if the following condition is accomplished:

$$\left| Angle(Obstacle_i) - Angle(CriticalDir) \right| < T_{angdev} \tag{16}$$

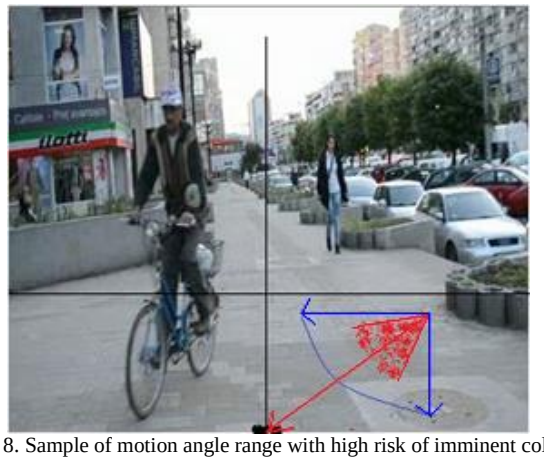


Figure 8. Sample of motion angle range with high risk of imminent collision (red dots area), around the reference critical direction (red vector)

IX. Experimental Results

In this section we present the results of moving obstacles detection based on the relative motion analysis. There are two major situations:

- The camera is fixed and the obstacles are moving
- The camera is moving and the obstacles are either moving or fixed

We evaluated the system on two video sequences containing more than one thousand frames from different traffic scenarios. The images resolution is 352x288 pixels and the parameters of the algorithms are those specified in previous sections. The entire system is performing real-time at about 20fps on an Intel® Core™ i5-3210M processor (2.5GHz frequency). The obstacle detection results were evaluated in terms of true positive rate and false positive rate and are presented in Table I.

TABLE I. Relative Moving Obstacle Detection Accuracy

	True positive rate	**False positive rate**
Sequence 1 (1550 frames)	76%	0%
Sequence 2 (1550 frames)	87%	2%
OVERALL	**81.5%**	**1%**

In the first video sequence, the camera was fixed and the scene obstacles were moving. In the second sequence the camera was either fixed or moving and the obstacles were either stationary or moving. Samples of obstacle detection results are shown in Figure 9 and Figure 10.

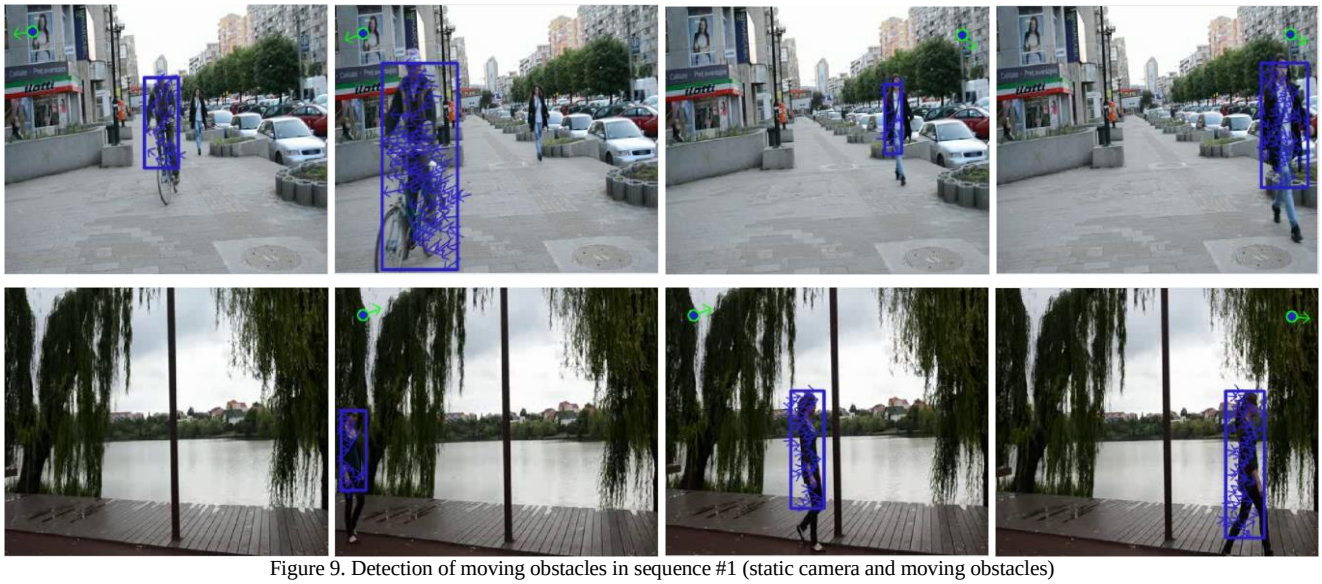


Figure 9. Detection of moving obstacles in sequence #1 (static camera and moving obstacles)

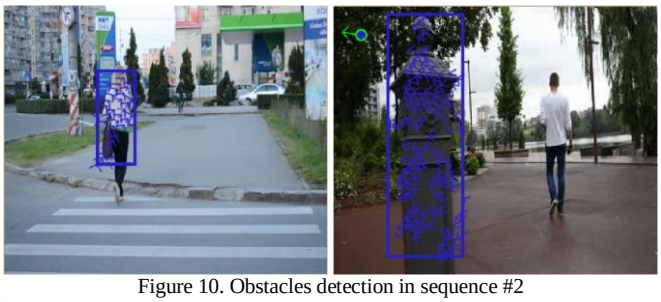


Figure 10. Obstacles detection in sequence #2
(left – moving camera, moving obstacle; right – moving camera, stationary obstacle)

X. Conclusions

We proposed and implemented a novel approach for road obstacle detection based on their relative motion vectors analysis. The color image sequences offered us the possibility of obtaining accurate motion vectors between frames using the Lucas-Kanade optical flow algorithm. They were computed in key-points locations uniformly distributed in a grid-like structure. This configuration, in comparison with SIFT points, has the advantage that the key-points exist even on the uniform colored and textured areas with the possibility of computing the motion vectors.

We determined the background general movement by proposing a robust RANSAC approach on the previously computed motion vectors. The motion vectors that belong to the background are then filtered out. With the remaining ones, a novel two-step clustering first on location and then on magnitude and angle are applied for grouping them in separate obstacles. A filtering is also used for removing spurious clusters (obstacles). Finally, we also defined a procedure for determining if there is an imminent collision with a detected obstacle and issue a warning in this situation.

The system is performing real-time and the results show good true positive rate (of finding the real scene obstacles) and very low false positive rate (of finding obstacles in locations where there is no real obstacle).

References

[1] R. Tapu, B. Mocanu, A. Bursuc, and T. Zaharia, "A Smartphone-Based Obstacle Detection and Classification System for Assisting Visually Impaired People," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 444-451.

[2] D. F. Llorca, M. A. Sotelo, A. M. Hellín, A. Orellana, M. Gavilan, I. G. Daza, *et al.*, "Stereo regions-of-interest selection for pedestrian protection: A survey," *Transportation research part C: emerging technologies*, vol. 25, pp. 226-237, 2012.

[3] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, *et al.*, "A Sensor for Urban Driving Assistance Systems Based on Dense Stereovision," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 276-283.

[4] C. Pocol, S. Nedevschi, and M. A. Obojski, "Obstacle Detection for Mobile Robots, Using Dense Stereo Reconstruction," in *IEEE International Conference on Intelligent Computer Communication and Processing*, 2007, pp. 127-132.

[5] S. Nedevschi, S. Bota, and C. Tomiuc, "Stereo-Based Pedestrian Detection for Collision-Avoidance Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 380-391, 2009.

[6] R. Danescu, S. Nedevschi, M. M. Meinecke, and T. Graf, "Stereovision Based Vehicle Tracking in Urban Traffic Environments," in *Intelligent Transportation Systems Conference*, 2007, pp. 400-404.

[7] G. Dong, T. Fraichard, X. Ming, and C. Laugier, "Color modeling by spherical influence field in sensing driving environment," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2000, pp. 249-254.

[8] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, "Stereo vision-based vehicle detection," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2000, pp. 39-44.

[9] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, pp. 62-81, 1998.

[10] H. Mori and N. M. Charkari, "Shadow and rhythm as sign patterns of obstacle detection," in *IEEE International Symposium on Industrial Electronics*, 1993, pp. 271-277.

[11] M. Heikkila and M. Pietikainen, "A Texture-Based Method for Modeling the Background and Detecting Moving Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 657-662, 2006.

[12] N. S. Boroujeni, S. A. Etemad, and A. Whitehead, "Fast obstacle detection using targeted optical flow," in *IEEE International Conference on Image Processing (ICIP)*, 2012, pp. 65-68.

[13] B. S. M. Madhavi and M. V. G. Rao, "A fast and reliable motion human detection and tracking based on background subtraction," *IOSR Journal of Electronics and Communication Engineering*, vol. 1, pp. 29-34, 2012.

[14] Z. Zhu and Y. Wang, "A hybrid algorithm for automatic segmentation of slowly moving objects," *International Journal of Electronics and Communications*, pp. 249-254, 2012.

[15] S. Vahora, N. Chauhan, and N. Prajapati, "A Robust Method for Moving Object Detection Using Modified Statistical Mean Method," *International Journal of Advanced Information Technology*, vol. 2, 2012.

[16] Z. Wei and G. Wen, "Accurate moving object segmentation by a hierarchical region labeling approach," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004*, 2004, pp. iii-637-40 vol.3.

[17] B. N. Subudhi and P. Kumar Nanda, "Detection of slow moving video objects using compound Markov random Field model," in *IEEE Region 10 Conference TENCON*, 2008, pp. 1-6.

[18] L. Qing-Zhong, D.-X. He, and W. Bing, "Effective Moving Objects Detection Based on Clustering Background Model for Video Surveillance," in *Congress on Image and Signal Processing*, 2008, pp. 656-660.

[19] P. D. Z. Varcheie, M. Sills-Lavoie, and G.-A. Bilodeau, "A Multiscale Region-Based Motion Detection and Background Subtraction Algorithm," *Sensors*, vol. 10, pp. 1041-1061, 2010.

[20] M. Yokoyama and T. Poggio, "A contour-based moving object detection and tracking," in *Proceedings of the 14th International Conference on Computer Communications and Networking*, 2005, pp. 271-276.

[21] M. Black and D. Fleet, "Probabilistic Detection and Tracking of Motion Boundaries," *International Journal of Computer Vision*, vol. 38, pp. 231-245, 2000.

[22] N. Thakoor and J. X. Gao, "Automatic Video Object Extraction With Camera in Motion," *International Journal of Image and Graphics*, vol. 08, pp. 573-600, 2008.

[23] A. Roy, S. Shinde, and K.-D. Kang, "An Approach for Efficient Real Time Moving Object Detection," in *International Conference on Embedded Systems & Applications*, Las Vegas, 2010, pp. 157-162.