UV disparity based obstacle detection and pedestrian classification in urban traffic scenarios

Alexandru Iloie, Ion Giosan, Sergiu Nedevschi
Computer Science Department
Technical University of Cluj-Napoca, Romania
alexiloie@gmail.com,{ion.giosan, sergiu.nedevschi}@cs.utcluj.ro

***Abstract*—High accuracy pedestrian detection plays an important role in all intelligent vehicles. This paper describes a system for detecting the obstacles in front of the vehicle and classifying them in pedestrians and non-pedestrians. It acquires the traffic scenes using a low-cost pair of gray intensities stereo cameras. A SORT-SGM stereo-reconstruction technique is used in order to obtain high density and accuracy in stereo-reconstructed points. First, the road plane is computed using the V disparity map and then the obstacles are determined by analyzing the U disparity map. Size related and histogram of oriented gradient based on gray levels features are used for describing each pedestrian hypothesis. A principle component analysis on the features is used for their selection and projection in a relevant space. Different SVM classifiers are trained considering the relevant features on large pedestrian and non-pedestrian image sets. A comparison between them is finally performed for selecting the one that achieves the best classification score.**

Keywords—UV-disparity; road plane detection; obstacle detection; feature extraction; feature selection; pedestrian classification

I. INTRODUCTION

Nowadays, the number of intelligent vehicles is growing rapidly due to the technological possibilities that are into a continuous upgrading process. Each intelligent vehicle is equipped with an intelligent driving assistance system. Generally, it includes many safety functions like obstacle collision warning, lane departure warning, lane keeping assistance, speed keeping assistance, etc.

Specifically, there exists a special category of intelligent vehicles in which the manufacturers built some protection parts that are automatically triggered in case of an imminent collision with a pedestrian in order to reduce the risk of fatal injuries. There comes the researchers' motivation for building very good obstacle detection and classification modules that could be successfully integrated in each driving assistance system for assisting the driver and providing alerts. A pedestrian detection module with high accuracy results is also crucial to be integrated in order to trigger the vehicle's protection parts in case of an unavoidable collision with a pedestrian. It must have a very low false positive detection rate in order not to falsely trigger the protection parts and also a sufficient true positive rate not to miss detect some pedestrians. Usually, a pedestrian tracking method is finally used for improving the overall accuracy.

This is a real challenge especially in difficult traffic scenarios like urban traffic with high cluttered scenes mainly with vehicles and pedestrians. The environmental conditions, the high frequency of traffic scene background variation, the variety of scene obstacles and their different locations in respect with the vehicle, and scene cluttering are making both the obstacles detection and pedestrian classification modules much more complex with many issues that must be solved in order to obtain accurate results. Otherwise, these modules cannot be used in a real-world driving assistance system.

Many different technologies like LASER-scanners, RADAR, LIDAR, ultrasound sensors, piezoelectric sensors and video cameras are used for acquisitioning the scene information. These may be spitted in two main categories: active and passive sensors. The video cameras are passive sensors that could acquire a large amount of information useful for further processing steps. They offers clean, passive, with no pollution for environment, scene image acquisition being the most similar with the human eyes-vision system.

In computer vision, the depth computation of each scene element is very important. This could be achieved by using stereo-cameras for images acquisition. We use a pair of two gray level cameras due to their low cost and their specifications that are high enough for achieving our goal. The stereo-reconstruction procedure offers the possibility of assigning to each pixel from intensity images, a corresponding depth value to the ego-vehicle. It also provides the disparity value for each reconstructed point.

In this paper we take the advantages of a dense and accurate SORT-SGM stereo-reconstruction. We assume a planar road surface for the urban traffic scenarios and present a method for road plane estimation from V disparity map. This is further used for separating the obstacles' points from the road surface points. The U disparity map is used for grouping the reconstructed points in different obstacles. The result is a list of precisely defined disjoint obstacles and well positioned in traffic scene. A size based filtering procedure is applied on this list in order to obtain the hypotheses that will be used in the pedestrian classification. Histogram of oriented gradients (HOG) descriptors are extracted from intensity bounding box of each hypothesis and used as preliminary features. Principal

This work was supported by the Romanian Agency of Scientific Research (UEFISCDI) under Contract PN-II-ID-PCE-2011-3-1086, MultiSens

component analysis (PCA) is applied for feature selection and projection on relevant vectors. Many SVM powerful classifiers are trained and tested on different sets containing pedestrians and non-pedestrians intensity images from our own database. We select one SVM classifier that achieves the best performances in terms of accuracy and speed to be used in pedestrian classification of the hypotheses.

The main contributions of this paper rely on the UV disparity based obstacle detection, obstacles filtering and hypotheses generation, feature extraction and selection, SVM classifiers training and finally obstacles classification in pedestrians and non-pedestrians.

II. RELATED WORK

A lot of research activity is carried out by the engineers for implementing better solutions for obstacle detection and pedestrian classification both from monocular vision and stereo vision setup cameras. In the case of monocular vision, common features like color or gray intensities [1], symmetry [2], edges [3], shadows [4] and textures [5] are widely used for obstacle detection. Optical flow [6] may also be computed for detecting moving obstacles by subtracting the ego motion of the vehicle.

Stereo-vision based systems acquire much more traffic scene information using at least two cameras. This allows the obstacle detection [7] to be done by analyzing both color/intensity and depth information [8] 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 [9] and density maps [10], followed by optical flow and motion computation for obstacle tracking [11] and finally obstacle classification [12], [13].

An approach based on inverse perspective transformation is proposed in [14]. It uses this process for transforming the left image for predicting the right image. Making a planar road assumption, the obstacles are determined by difference computation between the real right image and the predicted one. Methods that consist in finding the obstacles boundaries as parallel lines are used especially for vehicle detection. They are combined with the Hough transform in order to obtain a better matching with the corresponding edges in [15].

A method for locally estimating the road parameters is presented in [16]. It assumes that the road is locally planar. This offers the possibility of dealing also with non-flat roads by splitting the geometry in a sequence of quasi-planar geometry parts. The method does not need lane markings extraction and takes into account the other relevant road information like texture, shadows, edges etc. Using this estimation, the system achieves a robust and accurate detection of the on road obstacles.

The obstacle detection module has a primary role in a driving assistance system and its results will be used for further processing steps. It must achieve very good results, otherwise, if it miss-detect some obstacles then they won't enter into the classification module, so there may appear a high risk of not seeing a pedestrian. In order to obtain very good performance it must have as an input high quality stereo-images with high quality stereo-reconstruction.

In this work we use the SORT-SGM stereo reconstruction [17] implemented on a fast GPU, having the advantage of providing a high accurate dense stereo depth map in a short processing time. The depth map is denser than the one obtained using a local matching technique implemented on a hardware stereo-machine [18]. This is an advantage that offers us the possibility of building high quality UV disparity maps.

In case of crowded traffic scenarios, a simple obstacle detection procedure applied in every frame isn't enough. There are many obstacle occlusions that can affect the detection accuracy. Some obstacles are detected in several frames but lost in few other frames. In this kind of scenario an obstacle tracking module is mandatory. Generally, a tracker uses robust invariant features like SIFT or SURF or Harris corners [19] for obstacle description. The tracker is computed with a classifier that uses the corner position and other attributes in the feature set. In case when the obstacle class is known e.g. it is a pedestrian, then contour models could be used [20]. There exists other probabilistic shape based models that could be successfully used for distinguishing individual pedestrians within a group [21], [22], [23].

Obstacle classification is the next task that usually follows the obstacles detection. It has the main objective of assigning a different class to each detected obstacle. In literature exists a large number of different approaches achieving moderate to high accuracies. However the issue of obtaining a very high accuracy is far from being solved. Neural networks may be used for pedestrians, vehicles and other background clutters classification like in [24]. In [25], SVM is used for pedestrians, animals and vehicles classification. A similar approach but classifying bikes, pedestrians, trucks and cars using appearance features is described by the authors in [26].

In case of pedestrian classification, pattern matching techniques may be used with success. Such an approach based only on gray levels information is presented in [27]. Usually the depth information is used as a validation step after the obstacles were preliminary detected [28]. In [29], simple features extracted from many training instances define the input for a boosting technique. The result is a fast and robust Adaboost classifier. HOG features are also very widely used for pedestrian detection. The scene gray level image is basically spitted in non-overlapping cells distributed evenly. Inside each cell a weighted HOG is computed. Then the cells are grouped in overlapping blocks and the values of the histograms are being normalized within each block [30]. In [31] multiple discriminant obstacle features are extracted for each detected obstacle . A random forest classifier is applied on each obstacle in order to classify it in a set of different classes: pedestrians, cars, poles and other obstacles. Then a tracker is used for keeping the classified obstacle across frames.

In the next sections we present our pedestrian classification system. It consist of several modules. We describe our obstacle detection module based on UV disparity maps analysis, HOG feature extraction module from intensity information for each obstacle (pedestrian hypothesis), feature selection module using PCA and obstacle classification module (pedestrian/non-

pedestrian) using a robust SVM classifier trained on principal components projected features extracted from many input instances both pedestrians and non-pedestrians images.

III. System Overview

The obstacle detection and pedestrian classification system architecture with all its modules is depicted in Figure 1.

Gray levels stereo traffic images with resolution of 512x383 pixels are acquired. Stereo-reconstruction is performed using a semi-global optimized algorithm (SORT-SGM) on GPU having as input the two undistorted and rectified intensity images. This results in an accurate and dense depth map that is essential in further processing. The depth map stores, for each reconstructed point, the distance from the stereo cameras. The main parts of our system are described in details in the next sections.

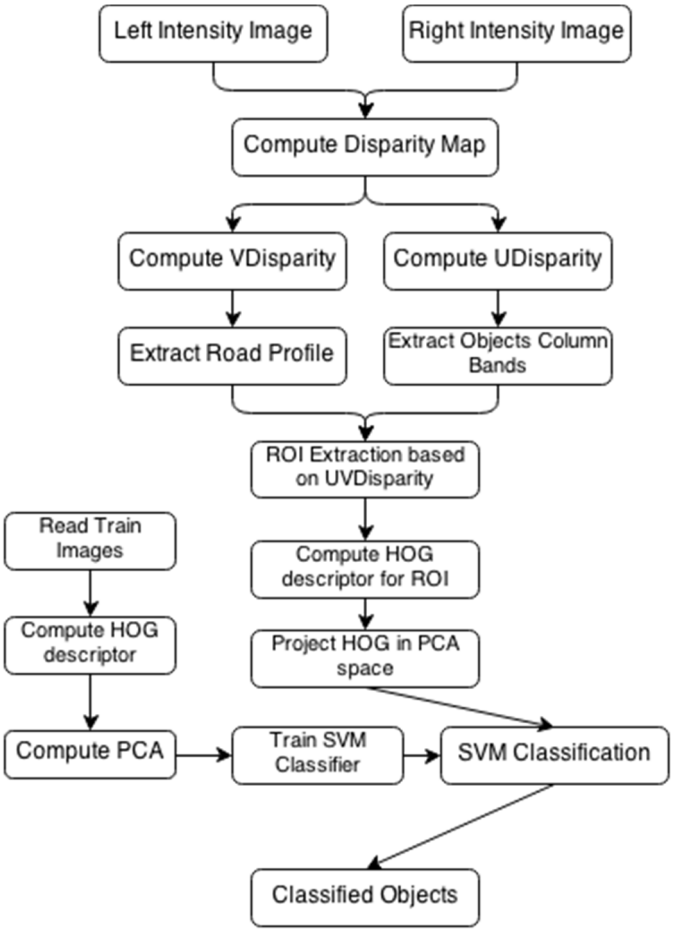

Figure 1. Obstacle classification system architecture

IV. ROI Extraction Based on UV-Disparity

In order to extract the regions of interest (obstacle regions) we used UV-disparity space. UV-disparity space is computed from disparity map, which is the first step of the presented algorithm.

A. Sterovision model

In order to compute the disparity map, we use the reconstructed 3D space. The 3D information allows us to compute the disparity [32] values using the following expression:

$$d = x_l - x_r = f\frac{b}{Z} \quad , \tag{1}$$

where d represents disparity value, x_l and x_r are coordinate values of the corresponding pixel, f is the focal length, b is baseline distance (distance between the two cameras of stereovision system) and Z is the depth coordinate of the 3D point. In Figure 2, a basic stereovision system is presented.

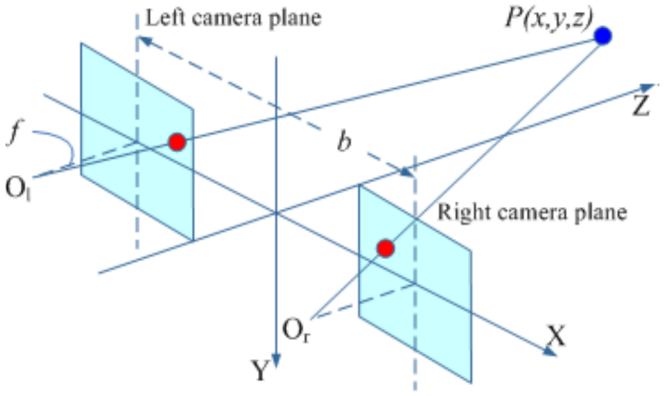

Figure 2. A canonical stereovision system

B. UV-disparity space

After the disparity map is computed, we have to compute the UV-disparity space.

U-disparity space is a column based matrix which stores the same disparity values for every column from disparity map.

$$u_{id} = \sum_{j=0}^{rows} \Delta_{ij}, \ \Delta_{ij} = \begin{cases} 1, & disp_{ij} = d \\ 0, & otherwise \end{cases} \quad , \tag{2}$$

where u_{id} represents the value from *U-disparity* space which cumulates the number of pixels with disparity d from column i in the disparity map.

Same as U-disparity space, V-disparity space is a row based matrix which stores the disparity values for every column from disparity map.

$$v_{dj} = \sum_{i=0}^{cols} \Delta_{ij}, \ \Delta_{ij} = \begin{cases} 1, & disp_{ij} = d \\ 0, & otherwise \end{cases} \quad , \tag{3}$$

where v_{dj} represents the value from *V-disparity* space which cumulates the number of pixels with disparity d from row j in the disparity map.

In Figure 3 we present the left image captured using the stereovision system, the computed V-disparity space to the right and below the computed U-disparity space.

Figure 3. UV-disparity space example

Next step of our algorithm is represented by the extraction of the road surface from V-disparity space and extracting the objects column bands from U-disparity space.

The road surface determines a straight line in V disparity space. It is found by using Hough transform. Basically the algorithm convert all the points from the input image from Cartesian space to Polar space. In the Polar space, every point is represented by a curve. The intersection between two or more curves gives us the Polar coordinates of a line in the image. We can define a minimum number of intersections (threshold) which will define a line and extract all the Polar coordinates of the intersections that are above that threshold. For the road surface extraction we only need the best line (see Figure 4) that results from the Hough algorithm (the line that has the largest number of intersections in the Polar space).

Figure 4. Extracted line (right image) from V-disparity space (left image) using Hough algorithm

We use the horizontal coordinates from U-disparity space in order to detect all the region of interests (obstacles). As we can see in Figure 7a, the U-disparity space has noise which makes difficult to detect any line. In order to remove the noise caused by road surface or by other small object we determined a threshold using object height, and after this threshold is applied (Figure 7b) we extract the lines using a labeling algorithm.

The labeling algorithm uses a classical approach which means that we start to search from the first row and first column and when we find a non-background point we save it and start to search for all his neighbors. Every neighbor is labeled with a unique class id to maintain the constraint that a point belongs to a single class. After we find all the neighbors of all points for one class we continue to search for new classes. The algorithm stops when we reach at the right corner of the input image and we have no points in the labeling list that have no class id defined. In order to define the best boundary between the lines (implicitly between detected objects) the search for the neighbors is based on the disparity value (given by the row that we are in the U-disparity space in that moment of search). The disparity value is inversely with the distance to the object. In case of objects that have a small value of the disparity (long distance) we have to search for a smaller range on the vertical and horizontal axis and opposite, for the high values of the disparity we have to increase the range of the search on both horizontal and vertical axis. We did this based on the fact that the objects which are closer to the camera can have bigger fluctuation of the disparity and horizontal coordinate than the objects that are at a long distance (see Figure 5).

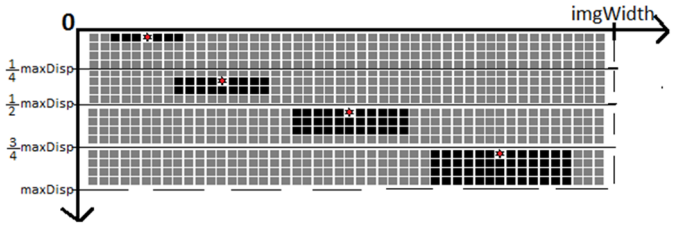

Figure 5. Neighbors of a pixel in the U-disparity space based on the disparity

The result of the labeling algorithm gives us a list of objects (every object is formed by a list of points that are in a neighbor relation) and for every object we have four important points: the vertical coordinates of the lower (point 1 in Figure 6) and upper (point 4 in Figure 6) point and the horizontal coordinates of the leftmost (point 2 in Figure 6) and rightmost (point 3 in figure 6) points. We use this four points in order to establish the bounding areas of the ROIs (obstacles) that will be used as input for the classification algorithm.

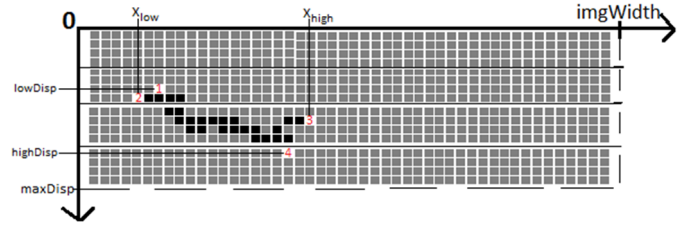

Figure 6. Labeled object in U-disparity space

In order to detect the regions of interest we traverse the list of segments and for each one we go through the respective column band (the column band is defined by x_{low} and x_{high} of the labeled object in the U-disparity space) and detect the pixels which have disparities in the same range (the range is defined by [*lowDisp*; *highDisp*]) as the respective object.

We can see in Figure 7 the column bands (B1, B2, …, B8) that are computed using the detected objects from U-disparity space. In order to reduce the number of ROIs we applied a threshold on the band width based on the pedestrian minimum

width (we see the threshold is applied because not all detected objects from Figure 7b give us a column band).

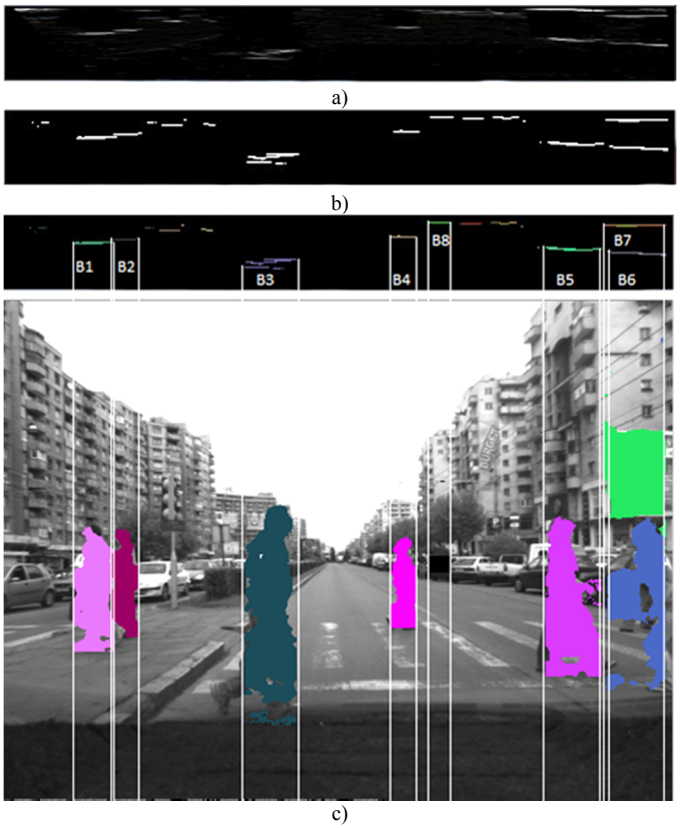

Figure 7. U-disparity space based processing: a) U-disparity space accumulators; b) thresholded U-disparity space; c) objects horizontal limits

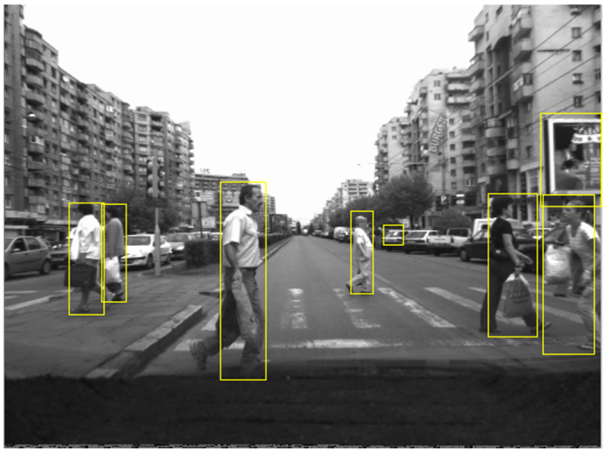

Figure 8. Detected ROIs from UV-disparity space

Every ROI is defined by four lines: two vertical lines (x_{low} and x_{high}) and two horizontal lines.

Using the column band disparity range (Figure 7c) and the parametric line computed using Hough algorithm in the V-disparity space we can compute the lower horizontal line of the object (boundary between the object and the road surface). Because we have a range of disparities, we will use only the upper limit of that interval in order to find the lowest line that will represent the ROI bottom line.

$$ahigh_{Disp} - by_{max} + c = 0 \quad , \tag{4}$$

where a, b, c were calculated using the Hough algorithm, $high_{Disp}$ represents the upper boundary of the disparities interval and y_{max} is the bottom line of the ROI.

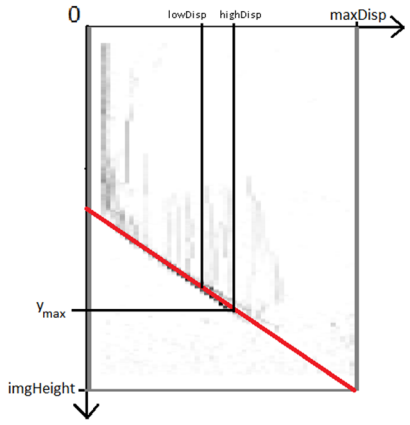

Figure 9. Computing y_{max} using the disparity range and the detected line from V-Disparity space

The next step consists in filtering the erroneous ROIs. In order to do this, we focused on one aspect: the ROIs that have the number of pixels lower than a threshold are removed. This threshold is computed using the boundary values of the ROI. In Figure 10 we can see the final detected regions of interest.

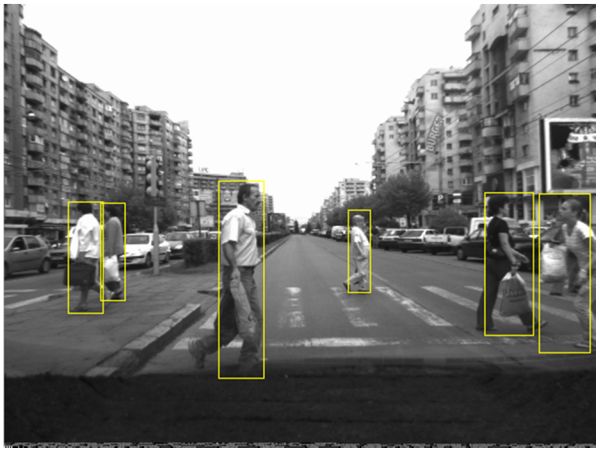

Figure 10. Result after filtering ROIs

V. OBSTACLE CLASSIFICATION

A. Features used

In order to classify an obstacle, relevant features have to be extracted. In this work we propose the use of histogram of oriented gradients features (HOG). The HOG descriptor is based on the oriented gradients in each pixel neighborhood. Initially, it was used for constructing descriptor vectors for sliding windows in the context of pedestrian detection [30]. In our experiments we computed the HOG descriptors for 64x128 window size, a block size of 16x16, cell size of 8x8 and 9 gradients bins per cell. A 3780 dimensional descriptor vector is then obtained.

B. *Principal component analysis*

In order to reduce the feature space, we used the PCA algorithm. The input data consist of a matrix with *nrOfTrainImages* lines X 3780 columns. PCA algorithm is applied on the input matrix for projecting it into the PCA space. The result is a matrix of *nrOfTrainImages* X *nrOfPCA* size. The *nrOfPCA* parameter is given as input data to PCA algorithm and cannot be higher than *nrOfTrainImages* (number of images used for training). Basically, the Principal Component Analysis examines relationships of variables and finds a linear projection of high dimensional data into a lower dimensional subspace. Each feature obtained after PCA is a linear combination of the input features.

C. *Classification method*

The last step of the presented algorithm represents the classification process. We used a Support Vector Machine classifier which was trained using the principal components of the HOG descriptor values of training set. We achieved the best results using a polynomial kernel for SVM. The parameters set for the kernel are: degree=3; gamma=1; coefficient=0.01; k=200 iterations.

We have two classes of objects, pedestrian and other objects. In the next paragraph, it is explained how we obtain the final result for a detected ROI.

All the regions of interest extracted from UV-disparity space will be resized to 64x128 pixels. Over this image the HOG algorithm is applied and the output represent a 3780 feature vector. This vector will be projected into PCA space computed in the train phase of the classifier. The feature vector resulted from PCA will become the input for the classifier which predicts the class of the candidate ROI.

VI. EXPERIMENTAL RESULTS

In this section we present our experimental results obtained with the proposed detection and classification system.

We trained the classifier using 10000 positive images (pedestrians) and 10000 negative class images (non-pedestrians). We used as test images a number of 3000 images for the positive class and 3000 images for negative class. We achieved the best statistical results when considering 1000 principal components. All the results can be found in the TABLE I.

TABLE I. THE PERFORMANCE PARAMETERS USING DIFFERENT NUMBER OF FEATURES FOR CLASSIFICATION SYSTEM

Without PCA		**PCA (500*)**		**PCA (1000*)**		**PCA (1890*)**	
TP Rate	**FP Rate**	**TP Rate**	**FP Rate**	**TP Rate**	**FP Rate**	**TP Rate**	**FP Rate**
0.903	0.011	0.946	0.048	**0.931**	**0.040**	0.982	0.229

*number of principal components

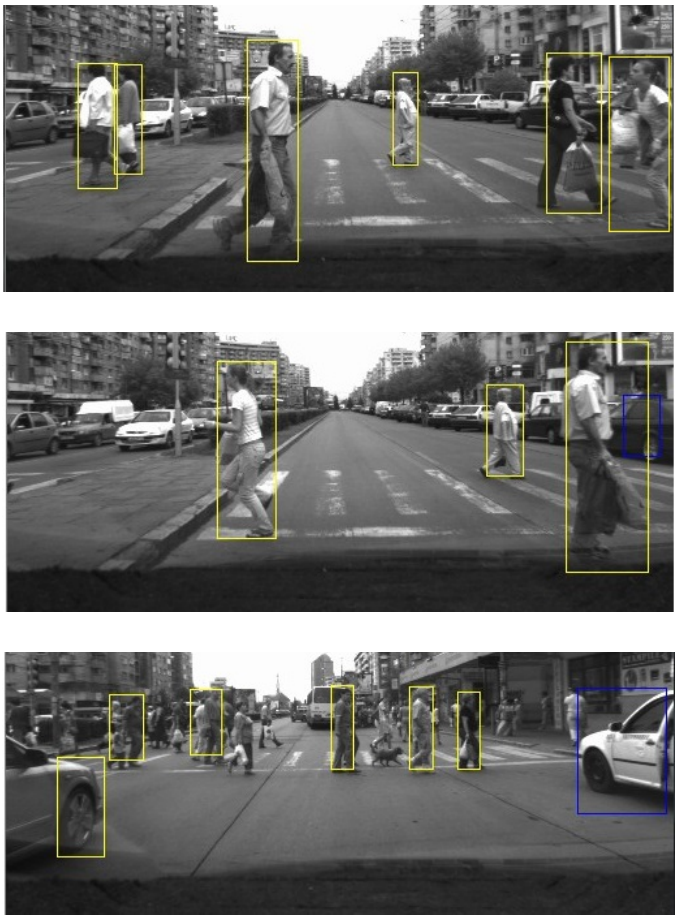

Figure 12. Obstacle classification: pedestrian (yellow color); other objects (blue color)

Regarding the U-disparity objects detection, we tried two algorithms, first one was the labeling algorithm presented in the paper and the second one was the Hough algorithm. We choose the labeling algorithm over the Hough because the last one can detect as a line two or more different objects that are on the same row or it is unable to detect multiple intersected lines as an object and we have to add another step to compose object between two or more intersected lines (see Figure 11 as a comparison between the detected lines of the two algorithms). We can see that the labeling algorithm (upper image) detects the composite lines as a single object while the Hough algorithm (lower image) detects them as being separate multiple obstacles.

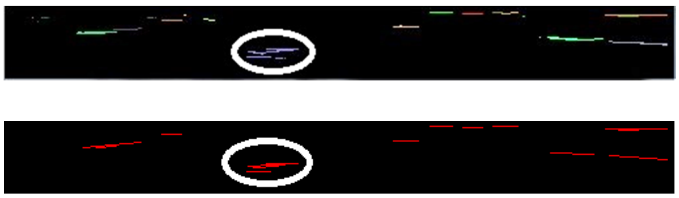

Figure 11. Difference between detected lines using labeling (upper image) and Hough algorithm (lower image)

VII. CONCLUSIONS

We achieved our purpose of creating a real time system that extract regions of interest using UV-disparity space and classify detected ROIs using a SVM classifier trained using HOG features. In order to extract important features from the UV-disparity space we used a combination of Hough algorithm for the V-disparity space and a labeling algorithm for the U-disparity space. Another important aspect, when extracting features from the UV-disparity space, several adaptive threshold values were used in order to cut off the noise. As a part of the system, we have successfully integrated a principal component analysis algorithm which reduced the number of used features, speeded up and improved the classification step. The system achieves real-time execution at about 24 fps on a PC with an Intel Core i3 processor at 2.39 GHz frequency.

A future work would be of improving the detected ROIs using the UV-disparity space and also improving the classification step, either by using other features or by mixing different classifiers.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] S. Bota, S. Nedevschi, and M. Konig, "A framework for object detection, tracking and classification in urban traffic scenarios using stereovision," in *IEEE 5th International Conference on Intelligent Computer Communication and Processing*, 2009, pp. 153-156.

[13] I. Giosan and S. Nedevschi, "A solution for probabilistic inference and tracking of obstacles classification in urban traffic scenarios," in *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2012, pp. 221-227.

[14] Z. Guo-Wei and S. Yuta, "Obstacle Detection By Vision System For An Autonomous Vehicle," in *Intelligent Vehicles Symposium*, 1993, pp. 31-36.

[15] H. Moon, R. Chellappa, and A. Rosenfeld, "Performance analysis of a simple vehicle detection algorithm," *Image and Vision Computing,* vol. 20, pp. 1-13, 2002.

[16] R. Labayrade, D. Aubert, and J. P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," in *IEEE Intelligent Vehicle Symposium*, 2002, pp. 646-651 vol.2.

[17] C. D. Pantilie and S. Nedevschi, "SORT-SGM: Subpixel Optimized Real-Time Semiglobal Matching for Intelligent Vehicles," *IEEE Transactions on Vehicular Technology,* vol. 61, pp. 1032-1042, 2012.

[18] J. I. Woodlill, G. Gordon, and R. Buck, "Tyzx DeepSea High Speed Stereo Vision System," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 41-41.

[19] B. Mccane, B. Galvin, and K. Novins, "Algorithmic Fusion for More Robust Feature Tracking," *Internation Journal Computer Vision,* vol. 49, pp. 79-89, 2002.

[20] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, pp. 809-830, 2000.

[21] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE,* vol. 90, pp. 1151-1163, 2002.

[22] H. K. Galoogahi, "Tracking Groups of People in Presence of Occlusion," in *Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, 2010, pp. 438-443.

[23] B. Lau, K. O. Arras, and W. Burgard, "Tracking groups of people with a multi-model hypothesis tracker," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 3487-3492.

[24] D. Toth and T. Aach, "Detection and recognition of moving objects using statistical motion detection and Fourier descriptors," in *12th International Conference on Image Analysis and Processing*, 2003, pp. 430-435.

[25] E. Rivlin, M. Rudzsky, R. Goldenberg, U. Bogomolov, and S. Lepchev, "A real-time system for classification of moving objects," in *16th International Conference on Pattern Recognition*, 2002, pp. 688-691 vol.3.

[26] Z. Lun, S. Z. Li, Y. Xiaotong, and X. Shiming, "Real-time Object Classification in Video Surveillance Based on Appearance Learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8.

[27] D. Gavrila, "Pedestrian Detection from a Moving Vehicle," *Proceedings of the 6th European Conference on Computer Vision-Part II,* pp. 37-49, 2000.

[28] D. M. Gavrila, J. Giebel, and S. Munder, "Vision-based pedestrian detection: the PROTECTOR system," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 13-18.

[29] A. Khammari, F. Nashashibi, Y. Abramson, and C. Laurgeau, "Vehicle detection combining gradient analysis and AdaBoost classification," in *Proceedings of Intelligent Transportation Systems*, 2005, pp. 66-71.

[30] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886-893 vol. 1.

[31] I. Giosan and S. Nedevschi, "Multi-feature Real Time Pedestrian Detection from Dense Stereo SORT-SGM Reconstructed Urban Traffic Scenarios," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, Lisbon, 2014.

[32] H. Zhencheng and K. Uchimura, "U-V-disparity: an efficient algorithm for stereovision based scene analysis," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2005, pp. 48-54.