Negative Obstacle Detection for Wearable Assistive Devices for Visually Impaired

Paul Herghelegiu, Adrian Burlacu and Simona Caraiman
Faculty of Automatic Control and Computer Engineering
Technical University "Gheorghe Asachi"
Iasi, Romania
{pherghelegiu | aburlacu | sarustei}@tuiasi.ro

***Abstract*— The research on developing assistive devices for visually impaired people greatly relies on image processing techniques. Extensive effort is made on detecting the obstacles in front of the user. Beside the normal obstacles that the user can bump into, another category of obstacles, negative obstacles must also be detected. Negative obstacles are usually represented by holes in the ground or regions that lay bellow the ground surface. Such obstacles represent a potential danger to the visually impaired user and could lead to serious injury if not detected. In this paper, we introduce a stereo vision system to identify and track negative obstacles located in front of the user. The identification is performed in the disparity image based on an estimation of the ground surface in the stereo images. The tracking relies on a camera motion estimation approach. The algorithm we introduce in this paper was integrated and tested with a wearable assistive device to prove its efficiency. We evaluate the accuracy of our solution using various real-life scenarios.**

Keywords*— visually impaired users; assistive devices; negative obstacles

I. INTRODUCTION

Visually impaired people commonly use a white cane to help them navigate. They use the cane to scan the environment in front of them. The cane is usually less than 2 meters long. In some cases, if there is a negative obstacle (e.g. a hole in the ground) in front of the user, the scanning technique the visually impaired use might fail to detect it. This happens when the white cane does not touch the ground, but its end is held at a low distance above the ground. Another case is when the negative obstacle is not large enough so the user might simply not detect its presence. Such situations are extremely dangerous to the users as they might fall and injure themselves.

In current days, a very important research area focuses in developing electronic assistive devices for the visually impaired people. Such devices aim to help the user navigate in an unknown environment. They also help the users have a more independent life, not having to depend on others in their everyday tasks. The tasks that the assistive devices must meet mainly focus on detecting obstacles in front of the user, detecting potentially dangerous situations or detecting some entities of interest like bus stops, crosswalks, markets, drug stores or others. The added value of using such assistive devices compared to the various uses of the white cane mainly comes from the covered distance ranges, the possibility to foresee the position of objects in the environment and to evaluate their size and type. This is done without the need of direct contact with them. Another added value is the early detection of elevated objects (e.g., at head height) as well as negative obstacles (e.g., missing sewer caps, inappropriately signaled construction work, etc.).

In this paper we introduce an algorithm to detect the negative obstacles in front of the user, in outdoor environments. The algorithm is integrated in an assistive system for visually impaired, the Sound of Vision system (SoV). It is a non-invasive, wearable sensory substitution device that assists visually impaired people by creating and conveying an auditory and tactile (haptic) representation of the surrounding environment. This representation is created based on computer vision techniques, updated and conveyed to the blind users continuously and in real time. The objective of the SoV system is to aid both the perception and the navigation of visually impaired users in unknown environments.

The proposed algorithm for negative obstacles detection is tailored for running on mobile devices. It relies on a disparity image computed using the stereo vision system integrated in the SoV device. The proposed approach aims to minimize the required computational time. Thus, we are able to enhance the assistive system with this functionality and still avoid adding significant delays to its main processing pipeline.

The paper is organized as follows: Section 2 describes the related work. In Section 3 the context and details of our algorithm are presented. The results of an experimental evaluation are discussed in Section 4. The paper is concluded in Section 5.

II. RELATED WORK

A large part of the research on the development of assistive devices for the visually impaired has been dedicated to obstacles detection. Existing obstacles detection systems for the blind can be divided in two classes, based on the computer-vision technique exploited to extract this information from the input images. One class of methods relies on information about the planar surfaces in the image, particularly the ground plane [1]-[6]. A grouping or a classification of the (remaining) pixels/points is used to identify potential obstacles. The methods of the second class directly extract information about the obstacles from the images, without the need for detecting any planar surfaces. These methods either rely on edge

information in the images [7]-[10], or infer information on potential obstacles directly in a 3D representation of the environment [11]-[13].

In the last decade, research on detecting negative obstacles was mainly reported in robotics, especially in autonomous driving of mobile robots. Coughlan and Shen [14] present some results on negative obstacles detection for a wheelchair equipped with a stereo vision acquisition system. The proposed approach is based on the analysis of the disparity map, with some hard constraints such as "the ground plane is then determined by finding the plane that best fits as much of the disparity map as possible". Regarding evaluation, the authors do not present any data on how their algorithm performs in different scenarios.

Murarka, et. al. [15], use stereo vision and motion cues to detect drop-offs for driving mobile robots with wheels in urban environments. The underlying principle for drop-off detection, for an occluding edge, new regions come into view as the robot moves towards the edge. For an occluding edge, features above the edge will hence appear to move "faster" relative to the edge than features below the edge, as compared to a non-occluding edge. This approach can detect only frontal drop-offs. The work is further extended by Murarka and Kulpers [16] where a more extended evaluation is performed.

Assistive devices based on stereo vision are more complex systems and deal with the same goals as in mobile robotics. One key aspect is the higher number of degrees of freedom for the motion of the stereo system. To the best of our knowledge, very few research works have specifically tackled this problem. Fazli, et. al. [17], proposed such a method for negative obstacle detection using seed-growing and dynamic programming. The method is built on multiple stages that include edge extraction, disparity map correction and noise removing. The solution mostly targets the detection of drop-offs between sidewalks and road.

The solution we propose in this paper is integrated in an assistive system that relies on ground information to detect obstacles in the environment. The system uses stereovision to infer depth information in outdoor environments. To reliably exploit this information inherently affected by noise, we introduce a temporal tracking and validation mechanism for selected candidates. Our method is able to detect various types of negative obstacles, both of hazardous (e.g., holes in the ground, potholes) and non-hazardous (e.g., difference of ground level) nature.

III. METHOD DESCRIPTION

A. Context

The workflow of the SoV system consists in four main steps: (1) acquisition of 3D information from the environment, (2) 3D reconstruction of the sensed environment and segmentation into objects of interest, (3) audio and haptic modeling of the processed 3D scene, (4) rendering of audio and haptic information to the user.

The acquisition of 3D information from the environment is performed using two types of depth sensors placed onto a rigid structure, which can be easily attached to various headgear designs. The 3D processing module exploits different combinations of sensor data to maximize the system usability in different situations (e.g. indoor/outdoor, different lighting conditions) and still provide environmental information in conditions atypical to standalone sensors. The acquisition system also integrates an IMU device that allows recovering the orientation of the head and cameras.

The 3D processing step performs a 3D reconstruction of the environment and has the role of identifying the elements of interest, such as ground, walls, ceiling, generic obstacles, negative obstacles, doors, stairs, signs and texts.

The detected objects are further encoded using custom audio and haptic models. Rendering the audio information can be performed by means of several types of headphones: regular on/over headphones, in-ear headphones, and custom design multi-speaker headphones. The main requirement for the audio rendering unit is to be either open or hear-through, such that the visually impaired user is still able to perceive the natural sounds in the environment. Haptic information is conveyed to the user by means of a custom made belt, placed on the user's abdomen.

Although the 3D acquisition system of the SoV device integrates both stereo and structured-light based depth sensing, we only refer to its stereo-based outdoor functioning. In this mode, the 3D processing component iteratively accumulates a 3D point cloud representation of the environment, based on stereo matching and camera motion estimation. A segmentation step, considering both 3D and image features, attempts to first differentiate between ground, walls and generic obstacles (above ground). Various heuristics are included in this step to also detect the presence of doors and stairs. The proposed algorithm for negative obstacle detection is integrated in the ground estimation step.

Once detected, the negative obstacle entities are passed to the encoding module of the SoV system, responsible for producing the user output. To this end, we provide the position in 3D space of the point on the contour closest to the camera (in camera coordinate space).

B. Hypothesis

By negative obstacle we understand a region of the environment that is located below the ground level. These regions can come up in the environment in two types:

- *Hazardous* negative obstacles – areas below ground that emerge due to natural or human factors, e.g., holes in the ground due to cracks in the pavement/asphalt, missing sewer cap.
- *Man-made* negative obstacles – areas below ground purposely existing in the environment, e.g., edge of a railway platform, stairs down.

The visually impaired user should definitely avoid the first type of obstacles, while the second type can purposely come up on their navigation path. Nevertheless, signaling their presence is of same utmost importance, both for safe navigation and for early environment mapping. In this paper, we address the detection of negative obstacles present on

horizontal ground, with a negative elevation with respect to the ground surface of at least 15 cm.

We performed an extensive analysis of the disparity maps computed for sequences of stereo images containing the specified types of negative obstacles, acquired in various illumination conditions. We conclude that the areas in the image corresponding to negative obstacles can have two appearances, depending on their size and illumination conditions (Figure 1):

Type1 - Disparity values for the corresponding pixels in the region indicate a negative elevation with respect to the adjacent ground area;

Type2 - Disparity values for the corresponding pixels in the region cannot be computed.

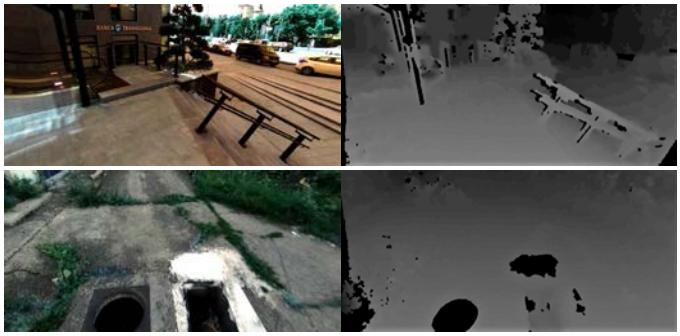

Figure 1. Images of negative obstacle and computed disparity maps. Top: Negative obstacle of Type 1, in the form of stairs down. Bottom: Negative obstacles of Type 2, in the form of missing sewer caps. (Note: the disparity images have adjusted brightness for presentation purposes)

The latter situation usually corresponds to small sized, deep holes in the ground (e.g., missing sewer caps). Due to lack of light reflection, they are captured by the cameras as black, uniform regions for which, the matching between pixels in the left and in the right image fails. The disparity computation can fail also for other uniformly colored areas. The proposed algorithm distinguishes them as described in Section III.C.3))

C. Proposed algorithm

The technique we introduce uses a disparity image as input data. In this image, we try to identify regions corresponding to negative obstacle. The errors and noise in the disparity image computation can often lead to false positive detections. To cope with such inconsistencies, we rely on a tracking mechanism for the validation of selected areas. If such an area in the image has consistent features over a number of frames then we mark the region as negative obstacle.

The main steps of our algorithm are (i) computation of the disparity map from the stereo vision system, (ii) detection of the ground surface, (iii) selection of negative obstacle candidates, (iv) tracking and validation of negative obstacle candidates.

Although the proposed solution relies on steps already integrated in the SoV system, we provide the standalone algorithm that can be reproduced outside of the SoV environment.

1) Disparity map computation

The established features of negative obstacles are valid for disparity maps computed with the ELAS algorithm [18]. Experiments with several other stereo matching schemes have been conducted. However, a set of stable features for negative obstacles could not be established, due either to the sparseness of the computed map (in the case of Block-Matching (BM) and Semi-Global Block Matching (SGBM)), or to the high amount of smoothing (BM and SGBM with post-filtering). Moreover, we only refer to stereo matching solutions that can work in real time.

2) Detection of ground surface

Detection of the ground surface is an important step of our algorithm. It is exploited for detecting regions with negative elevation, but also for the selection of negative obstacle candidates. This ensures that, irrespective of their features, the algorithm only considers the candidates connected to ground areas (regions with sparse or invalid disparity can also appear on walls or other surfaces).

The detection of the ground plane in stereo images can be achieved using different techniques. There are basically two approaches to tackle this task: processing the point-cloud associated to the disparity/depth map [2], [3], [5] or detecting the ground plane directly in the disparity domain [4], [19]-[21]. While the point-cloud processing solution is completely invariant to camera tilting, the disparity domain approach is less computationally demanding. The camera tilt can be managed by means of an IMU and/or a specific computer vision algorithm.

We employ a custom ground detection algorithm, working in image space, based on the method in [22].

3) Selection of candidates

A negative obstacle candidate is considered to be a region in the disparity map with the following properties, depending on its type (as defined in Section III.B):

Type 1:
(1) all the pixels inside the region have valid disparity values
(2) the elevation in 3D space indicates that the pixels of the region are positioned below the computed ground, with a difference of at least T_{e1} cm
(3) the region has an area in image space larger than a threshold, T_{a1}
(4) the region is connected with ground pixels

Type 2:
(1) all the pixels inside the region have invalid disparity values
(2) the region is connected with pixels having valid disparity values
(3) the region is connected with ground pixels or with pixels below ground
(4) the region has an area in image space larger than a threshold, T_{a2}
(5) the mean color of the pixels in the region is black

The contours of Type 1 candidates are detected in an image representing an elevation mask with respect to the computed ground level. In this image, the value of a pixel indicates its height above ground, compared to the T_{e1} threshold (Figure 2).

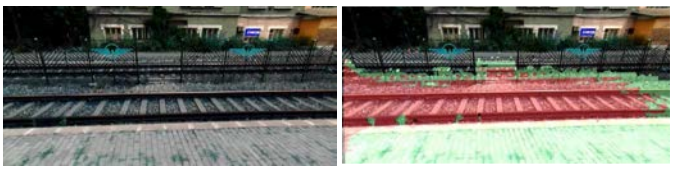

Figure 2. Type 1 candidate. (left) Original left image; (right) Elevation mask overlaid on the original image (green mask indicates detected ground surface, red mask corresponds to regions below ground, regions above ground have their originl color)

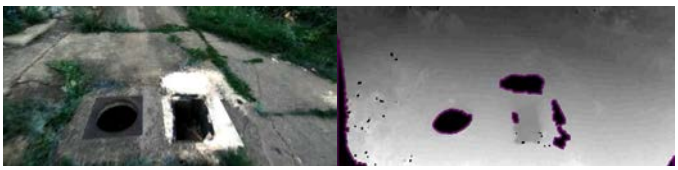

Figure 3. Type 2 candidate. (left) Original left image; (right) Detected contours superimposed on the disparity image, for regions larger than T_{a2} pixels. Contours of corresponding non-black regions in the original color image will be discarded.

Property (3) for Type 2 candidates ensures that the contours in 3D space can be computed for the region. The contours are required in the last step of the algorithm, which performs the tracking and validation of candidates. The Type 2 candidates are detected on an inverted thresholded disparity map of regions classified as ground and below ground. The threshold value is 1 as we need to select the regions with invalid disparity. On the resulting image we apply a dilation operation to slightly enlarge the areas with no disparity. The dilation is performed using a 3x3 kernel and one iteration. After these preprocessing steps, we apply the contour detection algorithm. This will ensure the fact that we can compute a 3D contour for the region (Figure 3). We only consider the contours of invalid disparity areas connected to ground regions (property 4). Moreover, we perform an additional check in color space to exclude non-black regions and thus filter out false positive detections (property 5).

4) Tracking and validation of candidates

Tracking is required to filter out the false positive detections. These can appear due to (i) erorrs in ground estimation, generating false regions below ground, and (ii) errors in the stereo matching step, generating areas with invalid disparity or false regions below ground.

The input to the tracking step is represented by the 3D contours of the candidate regions. Tracking all the points of the contour is not feasible as the shape of the candidate regions can highly differ from one frame to another. To simplify the tracking algorithm, we average the position of all the 3D points corresponding to one contour. This way we obtain the 3D centroid of the area that is bounded by the contour.

To track a point from one frame to the previous frame, we employ a camera motion estimation approach. Having the homogenous camera transformation between consecutive frames allows us to register the 3D points measured by the stereo vision system in the previous frame with the camera coordinate system of the current one.

However, the motion of a head mounted stereo camera used by a walking visually impaired person is highly complex. Regarding internal structure, this type of motion is a combination of 6-Dof head motion and body motion, the latter being dominated only by the forward velocity and yaw angle. In addition, visually impaired applications require a higher rate of visual odometry update. Hence, the dead reckoning error accumulated over time may grow faster than the moving speed of the camera. The performance of several camera motion estimation methods were analyzed with respect to: robustness to motion-blur caused by body motion, high accuracy in relation with ground truth. Most of the existing methods make use of a sparse set of local visual features, such as feature points, in order to estimate the camera motion. We selected the camera motion estimation approach by Geiger in the libviso2 framework [23]. Having a pair of stereo frames as input, it outputs an estimation for the (R, t) motion parameterization. The detection of feature points is performed using a peak threshold technique based on finding local maxima to reduce the amount of features. The set of detected feature points is then matched between four images, namely the left and right images of two consecutive stereo frames. This allows for the rejection of outliers due to false correspondences or moving objects.

Given all 'circular' feature matches found, the feature points from the previous frame are projected in 3D space via triangulation using the calibration parameters of the stereo camera rig. Assuming square pixels and zero skew, instead of minimizing the residuals in Euclidean space, the libviso2 framework makes use of the intrinsic parameters of stereo camera to minimize the residuals in the image space, where the noise level is similar for all components of the measurement vector:

$$\sum_1^N \left\| p_i^{(l)} - \pi^{(l)}(P_i, R, t) \right\|^2 + \left\| p_i^{(r)} - \pi^{(r)}(P_i, R, t) \right\|^2, \quad (1)$$

where $\pi^{(l)}$ denote the projection which takes a 3D point P and maps it to a pixel p. Using Gauss-Newton optimization the (R,t) motion estimation parameterization is recovered. Also, the framework refines the obtained velocity estimates by means of a Kalman filter assuming constant acceleration.

The motion estimation technique previously described is usually employed in automotive and robotics applications. For visually impaired assistive devices applications the internal parameters need to be adapted. A series of 3D virtual environments were designed to generate benchmark testing stereo sequences for human assistive devices. Since virtual scenes can provide ground truth information, testing using such synthetic data can offer valuable information about the efficiency of the algorithms and acknowledge worst case scenarios. Moreover, different environment scenarios can be tested without the need to physically find these locations or recreate some special situations in real life environments.

The evaluation of the stereo motion estimation accuracy was performed using the mean and standard deviation of two

errors: one for rotation (a Frobenius norm that measures the differences between the ground truth rotation matrix and the estimated one) and one for translation (a Euclidean norm that measures the differences between the ground truth translation vector and the estimated one). Tests conducted in the virtual scenes (with a scene consisting in 376 frames) revealed the following results:

Rotation: mean err 0.1027; std err 0.0736

Translation: mean err: 0.1719 [m]; std err 0.0813

To apply the transformation matrix on a pixel of the disparity image, we first compute the 3D position of the pixel. This is performed using:

$$\boldsymbol{z} = \frac{\boldsymbol{f}*\boldsymbol{b}}{\boldsymbol{d}}\ ,\boldsymbol{x} = \frac{(\boldsymbol{u}-\boldsymbol{c_x})*\boldsymbol{z}}{\boldsymbol{f}}\ ,\boldsymbol{y} = \frac{(\boldsymbol{u}-\boldsymbol{c_x})*\boldsymbol{z}}{\boldsymbol{f}}, \qquad (2)$$

where x, y, z are the 3D coordinates, (c_x, c_y) represent the center of projection, d is the disparity value of pixel(u, v), b is the baseline of the stereo camera, f is the focal length of the camera.

At this point of the algorithm, we have computed for each candidate region a 3D centroid of the points that bound the area. Each frame will have a set of computed 3D centroids. Next, we track these centroids over consecutive frames. For this, we store for each frame all the 3D centroids and the frame number they were computed in.

For each frame, the tracking mechanism is implemented as follows:

- Transform the 3D centroids computed in the last frame by applying the camera transformation matrix computed between the previous and the current frame
- Compute the distance between all the 3D centroids found for the current frame and the transformed 3D centroids from the previous frame
- If the distance between two such 3D centroids is lower than a threshold, update the frame number of the 3D centroid from the current frame with the frame number of the transformed 3D centroid
- If the difference between the updated frame numbers and the current frame number is higher than a threshold, consider the corresponding centroid as belonging to a region in the image that has to be avoided

A candidate becomes valid and signaled to the user as negative obstacle after continuously tracking it over a minimum number of frames (Figure 4).

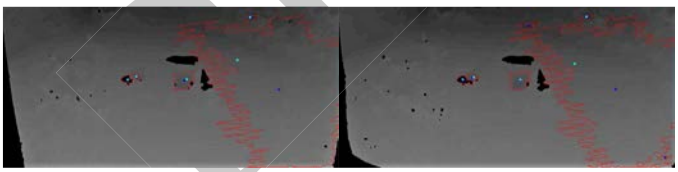

Figure 4. Contours of selected candidates tracked between two consecutive frames. The centroids of the candidates detected in the current frame are marked with blue, while the reprojected centroids of candidates from previous frame are marked with cyan.

IV. Experimental results

A. Implementation details

The proposed algorithm was implemented using C++ and OpenCV [24].

The threshold for negative elevation, T_{e1}, in case of Type 1 candidates was set to 15cm. The value for the area thresholds, T_{a1} and T_{a2}, was empirically selected to be 150 pixels.

For Type 2 candidates, we only consider black regions on the ground, with invalid disparity values. We compute the mean color of the candidate pixels in the left camera image, in RGB space. We use the maximum norm to impose a color difference threshold of 20.

For all candidates the 3D centroid is computed and compared from the Euclidean distance point of view with the 3D centroids of the candidates from the previous frame. If the distance in 3D space is lower than a threshold, the track of the candidate is incremented. If the 3D centroid on a contour cannot be associated with a previously computed 3D centroid, it follows that the candidate is new and it will only be considered when processing the subsequent frames. The minimum length of the track, before a candidate is validated depends on the camera acquisition rate. A value of 5 frames was empirically set in our experiments with a 15fps stereo acquisition rate. The threshold for the distance between 3D centroids of tracked candidates was considered 50cm, for both types.

B. Computing time

Stereo sequences containing various types of negative obstacles were acquired using a LI-OV580 stereo system from Leopard Imaging. It provides images with a resolution of 2560 x 720 (2 x 1280 x 720) at 15 fps. The stereo camera was mounted on a custom made headgear with a baseline of 15cm.

We evaluated the computing time on two mobile platforms, an Nvidia Tegra X1 (quad-core ARM Cortex-A57, Ubuntu L4T) and an Ultrabook (Intel Core i7-4720HQ, Windows 10). The average execution time for the detection and tracking of negative obstacle candidates, computed over several sequences, with a total number of 790 frames, was 23.85ms on the TX1 platform and 11ms on the Ultrabook. Additionally, the execution times for disparity computation using ELAS with the Robotics setup and an OpenMP implementation, estimation of the ground surface based on the v-Disparity approach and the camera motion estimation step are detailed in TABLE I. . We reach an overall performance of the entire algorithm of 7.8 fps and 3 fps for the two platforms, respectively.

C. Accuracy of detection

For evaluation purposes, we recorded several sequences of images, containing the following types of negative obstacles: stairs down, edge of railway platform, missing sewer caps (TABLE II.). We captured the images in two situations, walking by and towards the negative obstacles. S1 was recorded while walking along a set of stairs going down. A handrail divides the stairs down area in two negative obstacles. S2 contains images with a single negative obstacle instance. S3 captures a complex and difficult scenario of

rough terrain, with the presence of three simultaneous instances of negative obstacles (two missing sewer caps and one area below ground). For all sequences, we performed the evaluation of our algorithm with respect to an environment reconstruction up to 5m distance in front of the camera/user. This is the normal usage scenario of the SoV system. To this end, the total number of ground truth instances were counted within this distance (estimated in the images based on the computed disparity).

TABLE I. COMPUTATIONAL PERFORMANCE OF THE PROPOSED NEGATIVE OBSTACLE SOLUTION

Computing Platform	**Operation**	**Execution time (ms)**
Nvidia TX1	Stereo correspondence	162.5
	Camera motion estimation	103
	Ground estimation	33.5
	Negative obstacles – Type 1	22
	Negative obstacles – Type 2	1.8
	TOTAL	**322.8**
Ultrabook	Stereo correspondence	68
	Camera motion estimation	42
	Ground estimation	8.4
	Negative obstacles – Type 1	9
	Negative obstacles – Type 2	1
	TOTAL	**128.4**

TABLE II. DESCRIPTION OF TEST DATASETS

Sequence ID	**Total Frames**	**Negative obstacles**	
		type	**instances**
S1	213	stairs down (Type 1)	253
S2	300	railway (Type 1)	170
S3	277	missing sewer caps (Type 1,2) region below ground (Type 1)	501

TABLE III. RESULTS OF EVALUATION

Sequence	**TP**	**FP**	**TPR**	**PPV**
S1	230	56	**0.82**	**0.87**
S2	160	20	**0.94**	**0.89**
S3	446	39	**0.89**	**0.92**
		AVERAGE	**0.90**	**0.87**

The accuracy evaluation was performed using standard metrics, i.e., True Positive Rate (TPR) and Positive Predictive Value (PPV):

$$TPR = \frac{TP}{TP+FN}, \quad (3)$$

$$PPV = \frac{TP}{TP+FP}, \quad (4)$$

where TP – true positives, FP – false positives, FN – false negatives. The results of our evaluation are presented in TABLE III.

The high number of false positive detections in the stairs sequence (S1) is almost exclusively due to the presence of black obstacles connected to the ground (e.g., pillars of the handrail), represented by areas with invalid disparity (Figure 5 – middle row). While they are not negative obstacles per se, they still represent entities that need to be avoided by the user. The negative obstacles in sequence S2 did not pose any problems to our algorithm (Figure 5 – top row). In this case, the false negative detections are very few (the track of the single negative obstacle is lost in only one frame), while the false positive ones are caused by errors in disparity estimation leading to imperfect ground surface fitting. This causes that small regions of the railway platform to be incorrectly considered below ground. S3 contains two areas with missing sewer caps, one of them represented by a Type 1 region, while the other is represented by a Type 2 region (Figure 5 – bottom row). As the camera approaches, it gets divided in two separate regions, of both types. Neither the algorithm, nor the evaluation considered merging two regions of different types in a single negative obstacle. The implementation of our tracking mechanism allows filtering most of the false positive detections. However, this is achieved at the cost of increasing the false negative rate when the track is lost. It takes 5 frames to re-validate a candidate after its track is lost in one frame.

On average, we miss the negative obstacles present in the scene in less than 10% of the time.

V. CONCLUSIONS

In this paper we present a novel algorithm to detect and track negative obstacles in stereo sequences. The method uses stereo correspondence and ground plane estimation to identify such potentially dangerous situations. The validation of negative obstacle areas is performed using tracking based on camera motion estimation. The algorithm was integrated with a wearable assistive device for visually impaired users. As the processing unit is not very powerful in terms of performance, all computations must be kept as efficient as possible. Also, such algorithms must run in real time so the user has time to react to the feedback provided by the system regarding potentially dangerous situations. The proposed algorithm was tested using real world data and multiple scenarios providing promising results. It provides results in real time and with high accuracy, on portable computing platforms.

The features of the selected negative obstacle candidates were defined based on the intended use of the assistive system. Thus, we only consider regions with invalid depth information or at more than 15cm below ground. The detection of curbs is also possible with our algorithm. However, due to the quality of the depth maps that can be computed in real time based on wearable stereo vision systems, curbs detection cannot be performed with the same reliability.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 643636 "Sound of Vision".

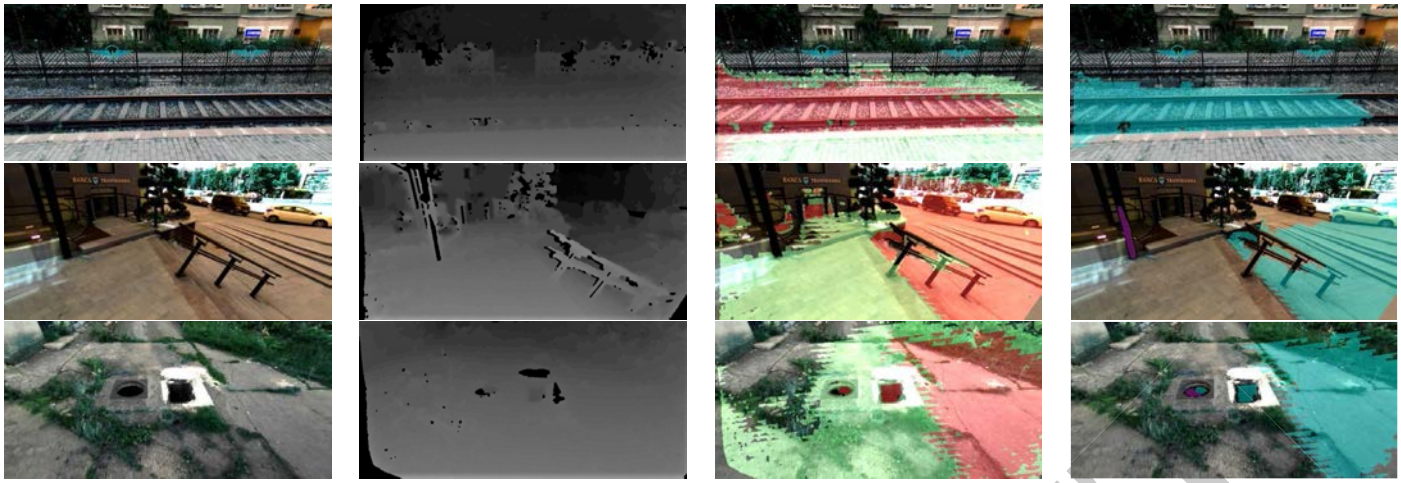

Figure 5. Results of negative obstacles detection. From left to right: Original left image; Disparity map computed with Elas; Elevation mask overlaid on the original image (green mask indicates detected ground surface, red mask corresponds to regions below ground, regions above ground have the originl color); Negative obstacles tracked over 5 consecutive frames (magenta – areas with invalid disparity, cyan – areas below ground).

REFERENCES

[1] J. M. Saez Martinez and F. Escolano Ruiz. Stereo-based Aerial Obstacle Detection for the Visually Impaired. In Workshop on Computer Vision Applications for the Visually Impaired, Marseille, France, Oct. 2008. James Coughlan and Roberto Manduchi.

[2] M. Bujacz. Representing 3d scenes through spatial audio in an electronic travel aid for the blind, 2010. PhD Thesis, Technical University of Lodz

[3] A. Rodriguez, J. Yebes, P. Alcantarilla, L. Bergasa, J. Almazan, and A. Cela. Obstacle avoidance system for assisting visually impaired people. In Proc. of the 2012 IEEE Intelligent Vehicles Symposium Workshops, 2014

[4] S. Mattoccia and P. Macri. 3d glasses as mobility aid for visually impaired people. In Proc. of the ECCV2014 Workshop, 2014

[5] F. Ribeiro, D. Florencio, P. Chou, and Z. Zhang. Auditory augmented reality: Object sonification for the visually impaired. In Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop on, pages 319–324, Sept 2012

[6] A. Burlacu, S. Bostaca, I. Hector, P. Herghelegiu, G. Ivanica, A. Moldoveanu, and S. Caraiman. Obstacle detection in stereo sequences using multiple representations of the disparity map. In 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), pages 854–859, Oct 2016.

[7] R. Nagarajan, G. Sainarayanan, S. Yaacob, and R. Porle, "Fuzzy-rule-based object identification methodology for navi system," EURASIP Journal on Applied Signal Processing, vol. 14, pp. 2260 – 2267, 2005.

[8] G. Balakrishnan, G. Sainarayanan, R. Nagarajan, and S. Yaacob. A stereo image processing system for visually impaired. International Journal of Signal Processing, 2(3):136, 2008.

[9] E. Peng, P. Peursum, L. Li, and S. Venkatesh. A smartphonebased obstacle sensor for the visually impaired. In Z. Yu, R. Liscano, G. Chen, D. Zhang, and X. Zhou, editors, Ubiquitous Intelligence and Computing, volume 6406 of Lecture Notes in Computer Science, pages 590–604. Springer Berlin Heidelberg, 2010.

[10] J. Jose, M. Farrajota, J. M. Rodrigues, and J. H. du Buf. The smartvision local navigation aid for blind and visually impaired persons. JDCTA: International Journal of Digital Content Technology and its Applications, 5(5):362 – 375, 2011.

[11] L. Chen, B.-L. Guo, and W. Sun. Obstacle detection system for visually impaired people based on stereo vision. In Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference on, pages 723–726, Dec 2010.

[12] J. Saez, F. Escolano, and M. Lozano. Aerial obstacle detection with 3d mobile devices. IEEE J Biomed Health Inform, 19:74 – 80, 2015.

[13] R. Tapu, B. Mocanu, A. Bursuc, and T. Zaharia. A smartphone-based obstacle detection and classification system for assisting visually impaired people. In Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, pages 444–451, 2013

[14] Coughlan J. and Shen H., Terrain analysis for blind wheelchair users: computer vision algorithms for finding curbs and other negative obstacles, Conference & Workshop on Assistive Technologies for People with Vision & Hearing Impairments CVHI 2007, pp.1-6

[15] Murarka A., Sridharan M. and Kuipers B., Detecting Obstacles and Drop-offs using Stereo and Motion Cues for Safe Local Motion, IEEE/IRJ International Conference on Intelligent Robots and Systems (IROS), 2008

[16] Murarka A. and Kuipers B., A Stereo Vision Based Mapping Algorithm for Detecting Inclines, Drop-offs, and Obstacles for Safe Local Navigation, IEEE/IRJ International Conference on Intelligent Robots and Systems (IROS), 2009

[17] Fazli S., Dehnavi H-M, Moallem P., A Robust Negative Obstacle Detection Method Using Seed-Growing and Dynamic Programming for Visually-Impaired/Blind Persons, OPTICAL REVIEW Vol. 18, No. 6, pp. 415–422, 2011

[18] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In Asian Conference on Computer Vision (ACCV), 2010.

[19] J. Zhao, M. Whitty and J. Katupitiya, "Detection of non-flat ground surfaces using V-Disparity images," 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, 2009, pp. 4584-4589

[20] D. Yiruo, W. Wenjia and K. Yukihiro, "Complex ground plane detection based on V-disparity map in off-road environment," Intelligent Vehicles Symposium (IV), 2013 IEEE, Gold Coast, QLD, 2013, pp. 1137-1142

[21] A. Rodriguez, J.J. Yebes, P.F. Alcantarilla, Bergasa, M. Luis, J. Almazan, A. Cela, Assisting the Visually Impaired: Obstacle Detection and Warning System by Acoustic Feedback, Sensors 12(12), pp. 17476-17496, 2012.

[22] P. Herghelegiu, A. Burlacu and S. Caraiman, "Robust ground plane detection and tracking in stereo sequences using camera orientation," 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, 2016, pp. 514-519. doi: 10.1109/ICSTCC.2016.7790717

[23] A. Geiger, J. Ziegler, C. Stiller, StereoScan: Dense 3D Reconstruction in Real-time, Intelligent Vehicles Symposium (IV), 2011.

[24] Open Source Computer Vision Library, Itseez, 2015.