

Aerial obstacle detection with 3D mobile devices

J.M. Sáez, F. Escolano, M.A. Lozano

Abstract—In this paper, we present a novel approach for aerial obstacle detection (e.g. branches or awnings) using a 3D smartphone in the context of the visually impaired (VI) people assistance. This kind of obstacles are especially challenging because they cannot be detected by the walking stick or the guide dog. The algorithm captures the 3D data of the scene through stereo vision. To our knowledge, this is the first work that presents a technology able to obtain real 3D measures with smartphones in real time. The orientation sensors of the device (magnetometer and accelerometer) are used to approximate the walking direction of the user, in order to look for the obstacles only in such direction. The obtained 3D data are compressed and then linearized for detecting the potential obstacles. Potential obstacles are tracked in order to accumulate enough evidence to alert the user only when a real obstacle is found. In the experimental section, we show the results of the algorithm in several situations using real data and helped by VI users.

Index Terms—Computer Vision, Visually Impaired, Mobile Vision.

I. INTRODUCTION

A. Contextualization

BLINDNESS is considered the major sensory disability (it is estimated that 80% of the human sensorial information is provided by sight), which determines to a large extent the life of a person, the interaction with the environment and with the society, and so on. A report of the WHO [1] indicates that there were 285 million VI people in the world in 2010. These amount includes different scales of visual impairment, where the severe is blindness (visual acuity below 5%). This group represents a 13.6% of the VI (39 million people in the world).

One of the daily challenges faced by a blind person is the autonomous movement. Regarding global orientation, there are different GPS-based systems available in the market with specific cartographies and a voice interface that solve this problem (e.g. the Kapten system [16]). As for the detection and obstacle avoidance, classic systems such as the walking stick and the guide dog are the most used.

Despite there exist technological advances in this field [2] [4], they have not become daily use tools for this community. This is due to the fact that the classic systems achieve their goals successfully and the new developments are bulky and uncomfortable, hindering the social integration of the user. In addition, these devices often send acoustic signals via ear-phones, which deprives the blind user of his main information source: the sound.

In recent years, the development of efficient computer-vision algorithms for solving specific tasks for the blind or the VI, including low vision, has emerged as a challenging field of

The authors are with the MVR Lab (Mobile Vision Research Lab) of the Department of Computer Science and Artificial Intelligence, 03690, University of Alicante, Spain.

scientific development. Fortunately the fast rate of appearance of novel and helpful devices (e.g. smart phones) has opened a new technological landscape: some researchers are intensely working towards complementing the traditional cane, which is usually considered the most practical tool for this persons, with additional wearable devices based on computer vision.

B. Mobility in open spaces

Large open spaces are a challenging context for the VI. They are low-structured environments such as parks where VIs have a limited number of structured references. In these environments the traditional cane is also of limited help, and most of the sensorial references are auditive (traffic on the left/right, child playing, people chatting).

In the literature, we found some notable examples of mobility developments for the VI. Some of them refer to text reading in the street (identifying street names and/or bus lines). There are two main approaches to identify patches of the image containing text: learning-based [8] and grouping-based [9]. The latter method has been recently extended for dealing with severe blur [11]. Factor graphs are also applied to another important topic in mobility: crosswalks protocols. In [10], for finding the best alignment between the user and the crosswalks, audio feedback is exploited to align the VI properly. In [12] 360° panoramas have been incorporated and converted to an aerial view of the nearby intersection for a later integration with Google Maps satellite imagery. Since, in general, GPS has a limited reliability because of the potential proximity of buildings, images become the most reliable source of information. For instance, in [13] vision is used for guiding VI to a target.



Fig. 1. Examples of aerial obstacles.

Another application for VIs in the context of mobility deals with aerial obstacle avoidance. These obstacles have no projection on the floor (typically tree branches, awnings, or similar elements). Some examples of this kind of obstacles are shown in Fig. 1. In [14], our experience in stereo-based SLAM

has provided a method for finding stereo maps with a stereo camera carried by a human user [15]. Having a short-term map computed on-the-fly we are able to classify obstacles in front of the user as aerial or not-aerial. In this paper we propose to adapt this kind of application to mobile devices (smartphones). In this regard, the main limitation to overcome is that SLAM-based short-term maps are too computationally demanding for a practical use, especially when real-time constrains arise. The structure of the environment could be also estimated through a monocular-based approach (see for example [3] where a monocular SLAM system is integrated into a smartphone). These approaches are suitable because all the smartphones integrate a camera. However, the range information extracted with this kind of algorithms is up to scale. In other words, the relative scale of the data depends on the nature of the environment. Then, the scale of the data changes as the environment changes. In practice, this kind of algorithms only works in limited space environments.

C. Goals

The main goal of our proposal is to develop a mobile application that acts as a walking stick or a guide dog complement. It does not replace these elements, but it solves their main problem, that is, their inability to detect aerial obstacles. In the case of walking sticks, this limitation is obvious. Dogs cannot be trained to detect these obstacles, because they are not aware of the height difference between them and their owners.

One of its main advantages is that the application is embedded into a smartphone, obtaining a comfortable and discreet system that favors the user social integration. Furthermore, the smartphone is also able to notify the presence of an obstacle by means of acoustic signals (through the phone speaker, not earphones) or vibrations. The latter option makes the system less noticeable and does not deprive the user of the sense of hearing.

Our approach is based on distance measures taken from the environment within a range of several meters. These measures are obtained from a stereo pair of images. Hence this software requires a hardware capable of obtaining the scene in stereo. Within these devices, we find the 3D phones that are endowed with a parallax-barrier glasses-free 3D screen and a double back camera (see Fig. 2). The purpose of these cameras is merely multimedia, but this equipment brings the opportunity of applying stereo vision on mobile devices (see Fig. 3). From the observation of the pair of images provided by the double camera, the scene can be partially reconstructed in 3D. This reconstruction includes the obstacles in front of the user and their distances.

In addition to the observation of the stereo pair of images, the application uses data from different sensors, such as magnetometers and accelerometers. These sensors provide the global orientation of the device, which is key to solve the direction in which the user is walking. With this information at hand we estimate the volume in which the obstacles should be detected.

This system has been developed for the Android platform, because other platforms (like iOS) do not have currently



Fig. 2. Smartphones endowed with 3D camera. HTC EVO 3D (left) and LG Optimus 3D (right).



Fig. 3. Reference and depth images (top left), and some views of the resulting 3D scene (bottom right).

available 3D devices. Nevertheless, it could be ported to any other platform whenever the required hardware is available.

II. AERIAL OBSTACLE DETECTION

The pipeline of our obstacle detection approach consists of four phases: (i) capture a stereo pair of images, (ii) obtain a set of 3D points using a dense stereo algorithm, (iii) build a histogram of 3D points in the direction in which the user is walking, and (iv) check for obstacles in the histogram.

A. Scene reconstruction

Let (I_t^L, I_t^R) be the stereo pair of images provided by the camera at instant t . Our goal is to obtain a set of 3D points $P_t = \{p_1, p_2, \dots, p_N\}$, where $p_i = (x_i, y_i, z_i)$ in metric coordinates with respect to the optical center of I_t^L .

Mobile devices equipped with a 3D camera provide a pair of rectified and pre-aligned images, so that the epipolar line of every pixel in the left image correspond to the same row in the right one. This fact allows us to apply a dense stereo algorithm [5] to obtain a disparity map D_t from the pair of images.

The device also provides the extrinsic data from its stereo camera: focal distance f (in pixels) and baseline B (in meters). The 3D scene can be reconstructed combining this information with the disparity map D_t . For each pixel i in the disparity image whose value is not unknown, a 3D point $p_i = (x_i, y_i, z_i)$ can be obtained as follows:

$$z_i = \frac{fB}{D_t(u_i, v_i)}, x_i = \frac{u_i z_i}{f}, y_i = \frac{v_i z_i}{f}, \quad (1)$$

being u_i, v_i the coordinates of the pixel in the 2D disparity image (with the origin of coordinates in the image center).

B. Distance histogram from 3D data

Let \vec{V}_t be the direction in which the user is walking at instant t . Only the obstacles found in this direction should be considered, and therefore 3D data obtained in the previous step should be filtered to remove side obstacles. Unfortunately, \vec{V}_t is not always the direction the smartphone is pointing to. On the one hand, we have to consider that the device lays on the user's chest, so that it has a pitch angle α_t that differs between users. On the other hand, a swing movement is produced as the user walks. This produces a yaw angle β_t that is always changing. Therefore, the direction \vec{V}_t is not constant with respect to the device and should be estimated.

Vector \vec{V}_t can be estimated from (α_t, β_t) . The global orientation of the device $(\alpha_t^g, \beta_t^g, \gamma_t^g)$ (pitch, yaw, roll) is obtained from the coordinate system defined by the accelerometer/gravity reading \vec{G}_t , the magnetometer reading \vec{M}_t and the cross product $\vec{G}_t \times \vec{M}_t$ (\vec{G}_t and \vec{M}_t are approximately orthogonal). The value of α_t can be determined directly by the device sensors. To make \vec{V}_t parallel to the floor, global pitch should be set to: $\alpha_t \leftarrow \alpha_t^g$.

If the movement of the user were straight (e.g. the movement in a vehicle) α_t would be enough to obtain \vec{V}_t . However, the swinging movement of the user causes β_t^g to change constantly. Thus, β_t has to be estimated from a set of N previous readings of $\beta_{t-1}^g, \beta_{t-2}^g, \dots, \beta_{t-N}^g$. The estimation of β_t is the difference between β_t^g and the expected value of the set of previous readings. Considering that this distribution is bimodal, a k -mean algorithm [6] with $k = 2$ is applied to separate them into the subsets β_A and β_B , having $\beta_A \cup \beta_B = \beta_{t-1}^g, \beta_{t-2}^g, \dots, \beta_{t-N}^g$. The estimation of β_t is then obtained as $\beta_t = \beta_t^g - \{E(\beta^A) + E(\beta^B)\}/2$.

Given the estimations of (α_t, β_t) the walking direction vector \vec{V}_t can be built. Around the axis determined by the center of our reference system and \vec{V}_t we place a parallelepiped of size $1m \times 1m \times 4m$ corresponding to the extrusion of the user's torso in the walking direction (see Fig. 4). This parallelepiped is used to register the subset of 3D points $P_t^* \in P_t$ that will intersect with the user's torso if the movement continues in the estimated direction. These points represent the possible obstacles for the user.

To interpret the obstacles, the parallelepiped is quantized in different bins, representing a discrete set of distances from the user position. We divide the parallelepiped in sections of s metres in depth ($s = 0.05m$ in our setup), and count how many 3D points belong to each block. This is represented by a histogram H_t . Each bin $H_t[i]$ represents the fraction of 3D points contained between the planes $s(i)\vec{V}_t$ and $s(i+1)\vec{V}_t$ of the parallelepiped. H_t represents a one-dimensional distribution of obstacles in the walking direction.

It is worth to remark that P_t has a projective nature, given that it is provided by a stereoscopic system. The higher the distance of observation, the higher the point sparseness. The trend of the degree of sparseness follows an exponential increase with respect to distance. This implies that cells $H_t[i]$ will present a decreasing density as i increases, which is due to the anisotropic error distribution but not to the obstacles. To deal with this problem, a unitary square C_i is created for

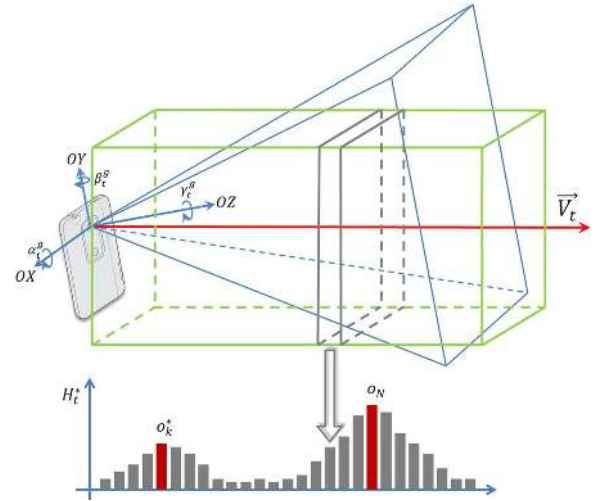


Fig. 4. We need to know the direction in which the user is walking to detect the obstacles, that may not match with the pointing direction of the camera.

each bin $H_t[i]$ at distance $s(i)\vec{V}_t$. The square is projected on the reference image, and we take the size S_i of the projection. These sizes have the same projective nature than $H_t[i]$, but in inverse order. Thus, we can obtain a linearized version of the histogram as follows: $H_t^*[i] \leftarrow H_t[i]/S_i$. The values of the histogram are also affected by the 3D occlusions of the points (each point of $H_t[i]$ projects a 3D shadow over the following bins that decreases their densities). However, in our problem the key obstacles are the closest ones, that are the least affected by this fact.

C. Obstacle detection from distance histogram

Each cell in H_t^* represents a possible obstacle. A single observation may present obstacles at different distances. Hence it follows that H_t^* is multimodal. Mean-Shift [7] is then used to separate it into different distributions, by using a uniform K -unit kernel. From the set of obtained centers, we keep the most significant ones at instant t , that is $O_t = o_1, o_2, \dots, o_N$.

The initial set of potential obstacles O_t may contain some *phantom* data due to the noise in the 3D reconstruction step. A robust set of obstacles O_t^* is obtained by considering only the obstacles detected in the last M observations $O_{t-M+1}, O_{t-M+2}, \dots, O_t$. An obstacle $o_i \in O_u$ matches an obstacle $o_j \in O_v$ if the distance between them in the histogram is less than K units, in consonance with the size of the Mean-Shift kernels. This guarantees that pairs of centers close enough will be discarded.

Given the set of obstacles O_t^* , the one o_n^* with the lowest index n (the nearest one to the user) is selected, whose distance is $d(o_n^*) = n \cdot s$. If this distance is below a given threshold (in our case 2 meters) then it is considered a potential threat and an alert signal (sound or vibration) is generated with a frequency inversely proportional to the distance $d(o_n^*)$. Closer obstacles cause a higher alert frequency.

III. APPLICATION INTERFACE

The usability of this application is directly related to its portability, because the device must hang from the user's

neck with the camera facing forward and the screen on the chest to activate the obstacle detection mode (see Fig. 11 left). Once the proximity sensor detects the device is in this position, the screen is locked and the obstacle detection begins. The detection finishes by flipping the device, or simply by separating it from the chest.

The obstacle detection is performed up to four meters forward, within the space corresponding to the user's torso (a volume of $1.0 \times 1.0 \times 4.0$ meters is estimated), correcting the swing movement produced when the user walks. The user receives obstacle alerts when they are closer than two meters in the walking direction.

The application presents an accessible interface, designed for blind users and it is based on three gestures: vertical swipe to change the menu item, horizontal swipe to explore the different values for the current item, and touch to select the current value. After each gesture, the device pronounces the current selection by voice synthesis, to let the user know the actions that have been executed. For example, in Fig. 5(left) the setting *Mode* is currently selected. An horizontal swipe changes this setting from *pause* to *obstacles* or *telemeter*. A vertical swipe moves to the *Warnings* setting.



Fig. 5. Look of the user interface.

The interface allows us to configure different features: **Mode**, that could be *obstacles* (for walking assistance), *telemeter* (for free environment exploration) or *pause*; **Alerts**, that may be *beep* (acoustic signal) or *vibration*; **Volume**, that sets the volume of the system; **Voice**, that sets the speech velocity; **Language**, that sets the language of the application (*English*, *German*, *French* or *Spanish*); **About** and **Exit**.

IV. EXPERIMENTS

In this section we present some tests about the most critical aspects of our proposal.

A. Implementation details

Besides the drastic changes that we have performed in the approach, the implementation has also suffered big changes with respect to [14], according with the new platform. Both 3D smartphones (see Fig. 2) are based on Android, whose principal language is Java. Nevertheless, we have used Qt¹ for Android (also known as Necessitas), a C++ based SDK that generates the code directly on Android native, which

¹qt-project.org

is more suitable for real-time applications. Also, we have used OpenCV4Android², the well known Computer Vision library [17], for image manipulation. In order to speed up some parts of the algorithm we have used parallelization strategies (through threading) that exploit the device Dual Core processor, as well as vectorization strategies with Neon Intrinsics (a set of instructions similar to Intel SSE integrated with the ARM architectures). These tools are justified by the computational requirements of the problem and the limitations of the platform.

B. Measure accuracy

In this first experiment we evaluate the accuracy of the distance estimation in our proposal. We have taken 19 3D images of a wall perpendicular to the focal axis of the camera. These images have been taken at distances from 0.35m to 3.95m, every 20cm. For each image a set of 3D points is obtained. Given that the only element of the image is a wall, all the 3D points should be placed at the same z coordinate, corresponding to the distance from the camera to the wall.

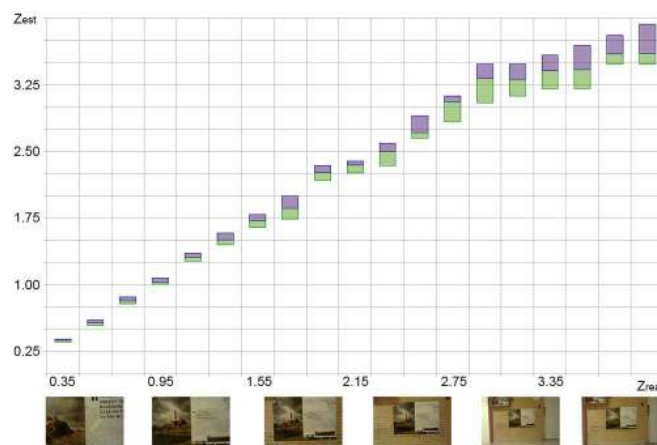


Fig. 6. Accuracy of the scene reconstruction. Top: First, second and third quartile of the 1D depth distribution of each image. Bottom: Some examples of wall images at different distances.

The results are shown in Fig. 6. Columns represent the images taken at each distance. For each image a 1D depth distribution has been obtained, and its first, second (median) and third quartiles are displayed in the figure.

It can be seen that as depth increases, the distribution becomes sparser and more noise is introduced. Our system discards measures larger than 4m due to the exponential growth of noise with the distance. In case of short distances, below 2m, it provides accurate measures (with a low error). We can also see that the standard deviation of the set of points increases as we increase the distance to the obstacles.

C. Histogram linearization

In this experiment we explore the projective nature of the histogram. We have taken 3 observations of a single object (a fire extinguisher) at different distances. For each observation

²opencv.org

we obtained its distance histogram P_t . Fig. 7(top) shows the raw histogram of each observation represented in different colors, and Fig. 7(bottom) shows the linearized histograms P_t^* . The horizontal axis represents the histogram bins (we consider a total distance of 4 meters, and each bin is taken every 5cm, so that we have 80 bins). The vertical axis represents the number of points in each bin (or the result of the linearization, in the linearized version).

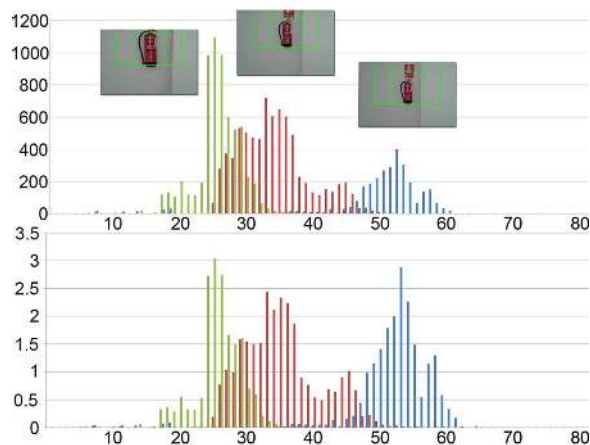


Fig. 7. Raw P_t (top) and linearized P_t^* (bottom) version of distance histograms obtained from 3 observations of a single object at different distances.

We can see that the raw histogram presents a variable density depending on the distance to the object, due to the effect of the projective geometry (the closer is the object, the wider is its area). Therefore, the value of the histogram bins can not be directly compared, which makes Mean-Shift not applicable. In the linearized version the densities of different observations of a single object achieve (approximately) a balance.

D. Obstacle tracking

In this experiment we evaluate the robustness of the obstacle detection over time in two different environments: a park (Fig. 8) and a corridor (Fig. 9). In these figures the horizontal axis represents the time, and the vertical axis is the distance histogram. That is, each column represents the distance histogram of each sequence frame (processed at 9fps approximately), so that we can observe the evolution of the histogram over time.

The blue line represents the threshold we use to notify the user about the presence of an obstacle (2m in our setting). The histogram represents 4m in total. The red points represent the obstacles that have been detected as real, that is, means obtained by Mean-Shift at a lower distance that the specified threshold and with tracking information enough to be considered a real obstacle and not a phantom.

In the first environment (Fig. 8) a tree is avoided. Note that once the tree has been avoided, it stops detecting this obstacle. In the second environment we first get close to a wall, and then we move away from it. We can see this reflected in the shape of the plot.

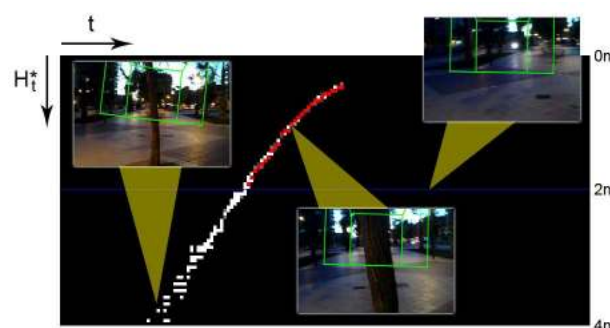


Fig. 8. Obstacle tracking in a park environment. The horizontal axis represents time. Each column displays the distance histogram for each frame.

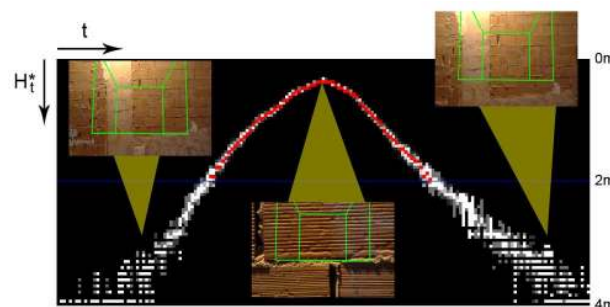


Fig. 9. Obstacle tracking in a corridor environment. The horizontal axis represents the time. Each column displays the distance histogram for each frame.

E. Multimodal histograms

Our obstacle detection approach uses Mean-Shift because the distance histogram is multimodal and may contain different distributions. This experiment aims to test the robustness of our approach and it consists of analyzing a sequence where a shelving is always observed at the back of the scene (see Fig. 10).

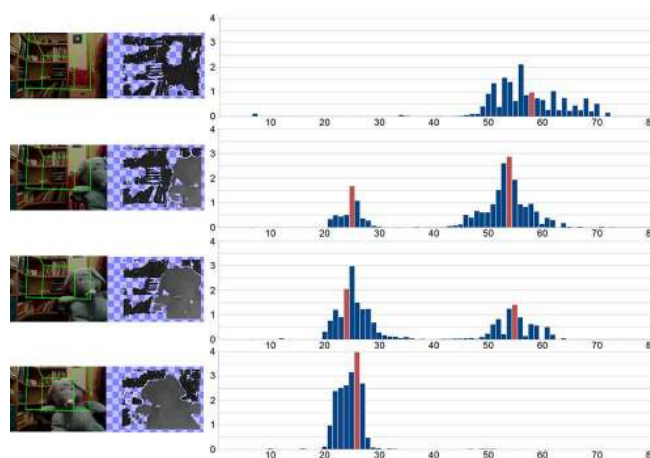


Fig. 10. Mean-Shift results. First and fourth rows: Only one obstacle is detected (either shelving or stuffed elephant). Second and third rows: Two obstacles are detected (shelving and stuffed elephant).

In the first frame, only the shelving is observed, so that it is represented by a single distribution in the histogram

(the result of Mean-Shift is displayed in red). In the second and third frames a stuffed elephant appears at the front of the scene. It shares the scene with the shelving in the back, which yields a second distribution. We can see that Mean-Shift correctly detects both distributions. In the last frame, the stuffed elephant covers all the projection (we only consider the parallelepiped corresponding to the user's torso extrusion). Therefore, only a single distribution is obtained.

F. Tests with VI users

The last experiment consists of several tests with blind users. In Fig. 11 we can see the people that have collaborated in this experiment: Maria Dolores (left) and Yolanda (right). Maria Dolores works in ONCE foundation as a psychologist. She is blind since she was 20 years old. She has an almost null residual vision (between 2% and 3%). She is only able to perceive light or darkness. Yolanda works as a counselor in a secondary school. She is psychologist too. She was born blind and does not have any residual vision.



Fig. 11. Test users: Maria Dolores (left) and Yolanda (right)

Fig. 12 shows a test with Maria Dolores. There is a palm tree leaf within the path. She walks slowly because she is not following a margin (she is walking in an open space). Some pictures of the scene taken from outside are shown in the first row of the figure, and the application visual-log is shown in the second row. In the visual-log we can see the distance histogram over the image. In the left row the obstacle has been detected. In the central row a notification is sent, because the obstacle is closer than 2 meters. In the right image the obstacle has been avoided.

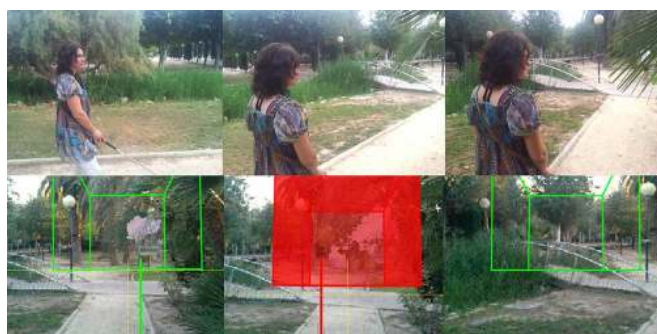


Fig. 12. First test with Maria Dolores. See the text for details.

A second test with Maria Dolores is shown in Fig. 13. In this case is following the curb with the cane, hence she walks faster. In the path there is a fuzzy object: a bush. This kind



Fig. 13. Second test with Maria Dolores. See the text for details.

of obstacles could not be detected by other sensors like sonar-based ones. The figure has the same format than the previous experiment: scene from outside (top) and visual-log (bottom), before (left) and after (right) avoiding the obstacle.

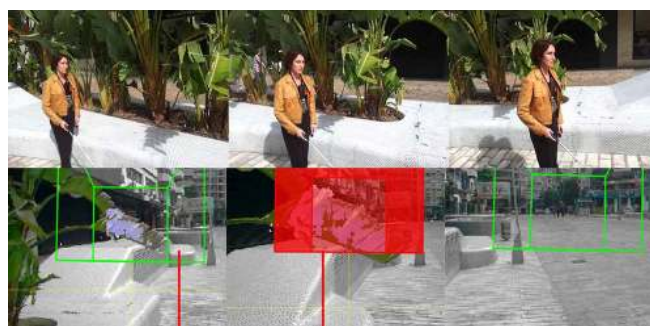


Fig. 14. Test with Yolanda. See the text for details.

Fig. 14 shows a test with Yolanda similar to the previous one. She is walking following the curb with her cane, and the application detects a the branch of palm tree. The displayed data follows the format described above.

All the experiments have been executed in a LG Optimus 3D Max smartphone (the results obtained with HTC EVO 3D are similar), which is endowed with a 1.2GHz Dual Core processor. The resolution of the captured images is 360×240 (we need a pair of images). The 1520mAh battery provides an operation time of about 132 minutes with the application running. Therefore, the application should be used only sporadically, in unknown environments.

The application is able to process an average of 9.17fps. Thus, the average lag of an obstacle alert is about 109ms. The walking velocity of a person is usually in the range between 4km/h (slow) and 6km/h (very fast). A blind person is usually slower than the lower limit. If we suppose a velocity of 4km/h (1.11m/s), with a processing time of 109ms per frame, and taking into account that the average reaction time of a person is 750ms, then the elapsed time since the obstacle appears until the user reacts is 859 ms. In the worst case, the user walks 0.95m from the instant in which the obstacle comes into the field of vision. Therefore, there is a margin of 1.05m to avoid the obstacle. For this reason the alert threshold is set to 2m, but the obstacle tracking is performed from 4m.

Dolores and Yolanda are our usual collaborators, but we have tested the approach with many other volunteers of the

blind community. Here we summarize the feedback that we recovered from 9 users that have tested the prototype. All of them consider that the problem we are facing represents a handicap in their lives, and a solution like this proposal could improve their quality of live. Seven of them agree with using a smartphone, that could be reused for other useful tasks, while 2 of them prefer an ad-hoc cheaper platform. With respect to the interface and the accessibility of the application, most of them agree that it is easy to use (8 of 9). We have observed that all users get a full control of the application in a guided session of around 10 minutes. Finally, the best result that we have observed (that we cannot show with data) is the great sensation that they experience in the first use, when they can sense the distances to the objects without touching them.

V. CONCLUSIONS AND FUTURE WORK

It is worth to highlight that the technology presented in this paper is new for this kind of devices. Until now, smartphones were not able to extract real measures from the environment. This application extracts about 30,000 real environment measures per frame at 9fps in commercial devices.

The major limitation of this technology is the dependency on a hardware that must incorporate a 3D camera. Our future work includes adapting this application to monocular devices. A way to do this is to incorporate a catadioptric device that splits a single camera observation into two separated ones. Another alternative consists on rethinking the algorithm with a Structure From Motion (SFM) approach instead of the Stereo one. This change could affect many parts of the approach, because the 3D results of the SFM algorithms are up to scale, that is, we only know the relative scale (depth) of a point with respect to the other points in the image, but the absolute scale is unknown and continuously changing.

ACKNOWLEDGMENT

J.M. Sáez and M.A. Lozano are supported by the University of Alicante research grant GRE10-21. F. Escolano is supported by the project TIN2012-32839 of the Spanish Government. The authors would like to thank ONCE, the Spanish Association for Blind People, for kindly testing the approach presented in this paper. This work was awarded with the VII Vodafone Foundation Prizes, in the "Mobile for Good" category in 2013.

REFERENCES

- [1] S. P. Mariotti, *Global data on visual impairments 2010*. World Health Organization, January 2012.
- [2] J. L. Finkel et al., *Ultrasonic path guidance for visually impaired*. United States Patent, December 2003.
- [3] G. Klein, D. Murray, *Parallel Tracking and Mapping on a Camera Phone*. Proc. of the 8th IEEE Int. Symposium on Mixed and Augmented Reality (ISMAR 2009).
- [4] S. Shoval, L. Ulrich and J. Borenstein, *Computerized Obstacle Avoidance Systems for the Blind and Visually Impaired*. Intelligent Systems and Technologies in Rehabilitation Engineering, December 2000.
- [5] K. Konolige, *Small vision systems: hardware and implementation*. International Symposium on Robotics Research, 1997.
- [6] J. A. Hartiga, M. A. Wong, *Algorithm AS 136: A K-Means Clustering Algorithm*. Journal of the Royal Statistical Society, 1979.
- [7] Y. Cheng, *Mean Shift, Mode Seeking, and Clustering*. IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE), August 1995.

- [8] X. Chen and A. L. Yuille, *Detecting and Reading Text in Natural Scenes*. Proceedings CVPR 2004.
- [9] H. Shen, J. Coughlan and V. Ivanchenko, *Figure-Ground Segmentation Using Factor Graphs*. Special Issue of Image and Vision Computing, Vol. 27, Issue 7. 2009.
- [10] H. Shen, K. Y. Chan, J. Coughlan and J. Brabyn, *A Mobile Phone System to Find Crosswalks for Visually Impaired Pedestrians*. Technology and Disability, Vol. 20, Number 3, pp. 217-224. 2008.
- [11] P. Sanketi, H. Shen and J. Coughlan, *Localizing Blurry and Low-Resolution Text in Natural Images*. Proceedings 2011 IEEE Workshop on Applications of Computer Vision (WACV 2011), January 2011.
- [12] J. Coughlan and H. Shen, *Crosswatch: a System for Providing Guidance to Visually Impaired Travelers at Traffic Intersections*. Special Issue of the Journal of Assistive Technologies (JAT), Vol. 7, Issue 2. 2013.
- [13] R. Manduchi and J. Coughlan, *The Last Meter: Blind Visual Guidance to a Target*. To appear in ACM CHI Conference on Human Factors in Computing Systems (CHI 2014).
- [14] J. M. Sáez and F. Escolano, *Stereo-based Aerial Obstacle Detection for the Visually Impaired*. Proceedings CVAVI 2008.
- [15] J. M. Sáez and F. Escolano, *Entropy Minimization SLAM for Autonomous Vehicles and Wearable Devices*. Computer Vision and Image Understanding. 2011.
- [16] J. Denham, *Oh Kapten! My Kapten! Where am I?: A Review of the Kapten PLUS Personal Navigation Device*. AFB AccessWorld Magazine, Vol. 12, Number 7, July 2011.
- [17] G. Bradski, *The OpenCV library*. Dr. Dobb's Journal of Software Tools, Vol. 25, Issue 11. 2000.



J.M. Sáez Juan Manuel Sáez obtained his Bachelor and Ph Degree in Computer Science at the University of Alicante in 1999 and 2005 respectively. He is currently Associate Professor at the University of Alicante. His research interests are focused on the development of vision-based algorithms combined with 3D range data for robot applications, smartphones, wearable devices and 3D modeling. He is a member of Mobile Vision Research Lab.



F. Escolano Francisco Escolano obtained his Bachelors Degree in Computer Science at the Polytechnical University of Valencia in 1992 and his PhD Degree in Computer Science at the University of Alicante in 1997. Since 1998, he has been an Associate Professor at the University of Alicante. He has been post-doctoral fellow at the Biomedical Engineering Department at the University of South California. His research interests are focused on the development of efficient and reliable pattern recognition and computer vision algorithms. In 2001

he co-founded the Robot Vision Group from which emerged the Mobile Vision Research Lab.



M.A. Lozano Miguel Angel Lozano obtained his Bachelor and Ph Degree in Computer Science at the University of Alicante in 2001 and 2008 respectively. Since 2007 he has been a Lecturer in the Department of Computer Science and Artificial Intelligence of the University of Alicante. He has visited Computer Vision & Pattern Recognition Lab at the University of York, and Bioinformatics Lab at the University of Helsinki. He is the head of the Mobile Vision Research Lab. His research interests include pattern recognition and computer vision.