Proceedings of the World Congress on Engineering 2011 Vol II
WCE 2011, July 6 - 8, 2011, London, U.K.

An Obstacle Detection System for Blind People

Nazli Mohajeri, Roozbeh Raste, Sabalan Daneshvar*, member, *IAENG*

Abstract— **Blind people face several problems in their life, one of these problems that is the most important one is detection the obstacles when they are walking. In this research, we suggested a system with two cameras placed on blind person's glasses that their duty is taking images from different sides. By comparing these two images, we will be able to find the obstacles. In this method, first we investigate the probability of existence an object by use of special points that then we will call them "Equivalent points", then we utilize binary method, standardize and normalized cross-correlation for verifying this probability. This system was tested under three different conditions and the estimated error is in acceptable range.**

***Index Terms*— Obstacle detection, Blind people, Equivalent points.**

I. Introduction

EYES play a vital role in our life. All of us have seen the blind people and know the problems that they face in their life. In order to detection the obstacles blind people use stick when they are walking but this instrument just can help them find objects on the ground. Obstacle detection is a field of effort that has led to vast progress in primary safety systems and in primary–secondary safety systems interaction. To detect obstacles at a medium to long distance, either static or mobile, different technologies have been used, like laser scanners Solutions based on the sensory fusion of lasers canner, radar and computer vision have been used with the purpose of obtaining additional information for a better interpretation of the environment, as well as for mitigating the deficiencies of each sensor [1-5].

Our suggested system uses two cameras placed on blind person's glasses that take online images and within image processing detection the obstacles, and finally notifies blind person. In this system there is no need to recognize all the obstacles but we just require those that are in a specific distance from the cameras. As shown in figure (1) we have considered a virtual plane with specific dimensions, any kind of objects that enter this plane must be recognized by system. Figure (2) shows the thorough algorithm of the system.

Nazli Mohajeri (BSc) is with the Electrical Engineering Faculty, Sahand University of Technology, Tabriz, Iran (e-mail: nazi_mhjr@yahoo.com).
Roozbeh Raste (BSc) is with the Electrical Engineering Faculty, Sahand University of Technology, Tabriz, Iran (e-mail: roozbeh.raste@gmail.com).
Sabalan Daneshvar (PhD) is with the Electrical Engineering Faculty, Sahand University of Technology, Tabriz, Iran (corresponding author to provide phone: +98912-2258523; fax: +98412-3459342; e-mail: daneshvar@sut.ac.ir).

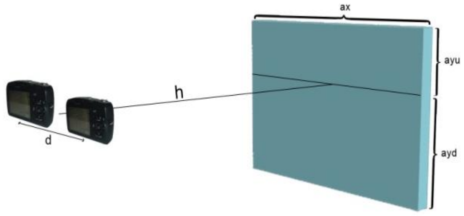


Fig.1 Position of the camera 1(right), 2 (left) and the virtual plane

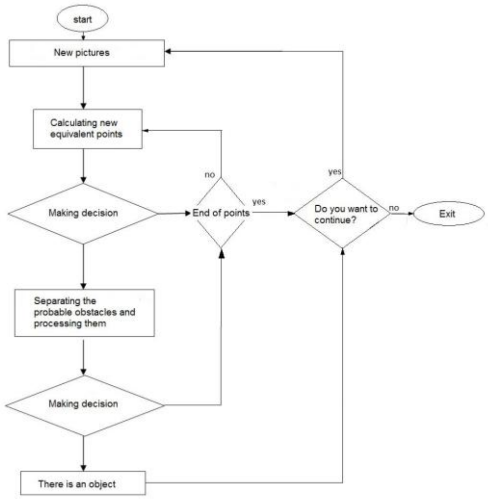


Fig.2 The thorough algorithm

II. Materials and Methods

A. *Equivalent Points*

In this research, there is an object 70 cm far from the cameras, this object will specify a particular place for itself in the pictures taken by cameras (Fig.3). We call these places (two points in pictures) "Equivalent Points". These points are unique (i.e. those pixels that are equivalent for 70 Cm distance are not equivalent for 75Cm), additionally computing these points are completely parametric and you can change the initial conditions (initializing depends on field of vision of cameras).

As initializing, 50 points on x axis and 40 points on y axis (one point per cm) have been considered (i.e. 2000 pair of equivalent points), all of these points will be probed on virtual plane (Fig.4). Exploring the points is from the top of the plane to bottom and from right to left (Fig.5 right). The place of our points is expressed in centimeter and we need to transform them into pixels. As shown in figure (5 left) we use simple calculations for transforming. For an example, if we multiple the division of *dby* (the height of recent point) and *day* (the whole height of taken picture), to *yr/2* (half of the height of picture) and subtract the result of that from *yr/2*, we will obtain the height of the expressed point in pixel.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

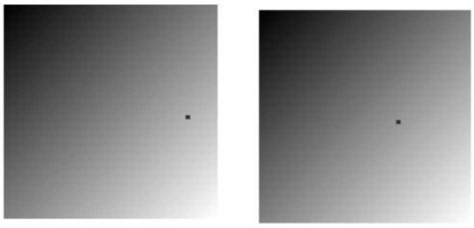


Fig.3 Simulation of a point in vision of camera 1(right) and 2(left)

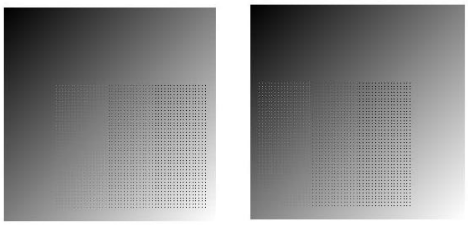


Fig.4 All equivalent points in picture of camera 1(right) and 2(left)

$$ypix = \frac{yr}{2} + \frac{-dby}{\mathrm{day}} * \frac{\mathrm{yr}}{2}$$

Where, *day= h.tg(za)* and *Za* is vertical field of vision of cameras. After calculations of height, all 50 points in this height must be explored; in this stage we define three regions (Fig.6 right) and behave in same way (Fig.6 left). By using this method the equivalent pixels will be detected. If gray scale of 2 equivalent pixels has differences less than 5 in value, being an object in that place will be probable. There must be no difference between two pixels if they are from same object but because of existing shadows, we apply this tolerance, if this probability is verified by the system, more difficult decisions will be made for later sections, if not the system will ensure that the taken pictures are from different objects and there is no object in that distance and will calculate other points.

B. *Obstacle detection*

This section is the step of making more difficult decisions for detection obstacle. For excel describing, we show processing on one of the pictures step by step (Fig.7). First we convert our pictures to binary mode by changing the value of pixels to 0 and 1, if they are in range of equivalent pixels will get 1 if not will be 0, (Fig.8). Then we separate the regions that involve equivalent points and blacken the rest of the image (Fig.9).

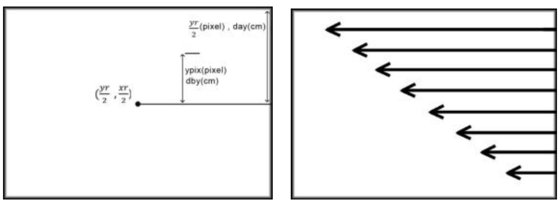


Fig.5 Height of points in Cm and pixel mode (left) and method of exploring the points (right)

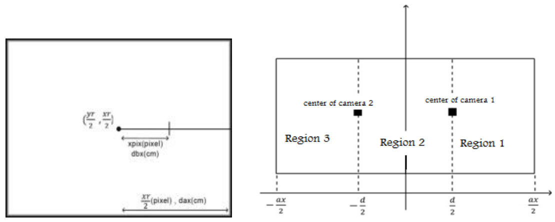


Fig .6 Widths of points in cm and pixel mode (left) and regions on virtual plane (right)

Fig.7 Pictures taken by camera 1(right) and 2(left)

Fig.8 Binary pictures

Fig.9 Separated regions

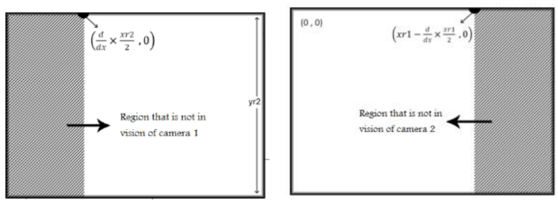


Fig.10 Regions that are not in vision of each camera

In order to find out whether or not these obtained regions refer to one object, we draw a bonding box around them. Since taken images by each camera include some regions that are not exist in second one, we must cut these regions out (Fig.10), this should carry out specially when a part of object is in vision of one of the cameras and not in vision of other one. Now the center of two separated regions must be calculated so we should change the pixels to Cm, for avoiding having error we accept 3 cm tolerance and ignore regions that are less than 10 pixels.

When an object is in front of a camera, it looks bigger in vision of that camera than another one, for solving this problem we have performed a method that stretches the picture of objects and makes them standardized and comparable. Fig.11 shows this method.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

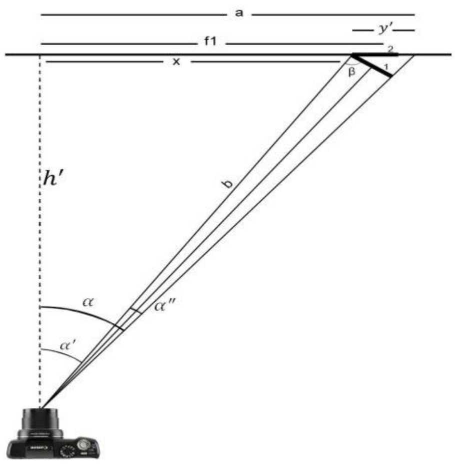


Fig.11 Width of object in picture and its real width

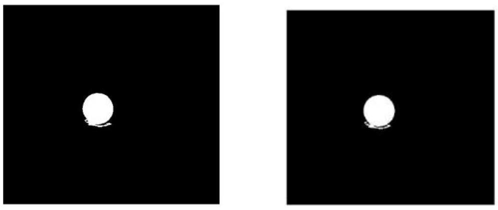


Fig.12 Standardized pictures of camera 1(right) and 2(left)

$$f = \left(\frac{xr}{2} - CO\right) \;;\quad \acute{h} = \frac{\frac{xr}{2}}{\tan(za)}$$

$$\acute{\alpha} = \operatorname{atan}\left(\frac{f - HW}{\acute{h}}\right)$$

$$b = \frac{\acute{h}}{\cos(\acute{\alpha})} \quad => \quad \beta = \operatorname{acos}\left(\frac{HW}{b}\right)$$

$$\gamma = \pi - 2\beta \quad => \quad \alpha = \acute{\alpha} + \gamma$$

$$y = (h.\tan(\alpha)) - (f - HW)$$

$$x = (f - HW)$$

$$a = (h(\tan(\alpha)))$$

Where, *CO* is center of object and *HW* is half of the width of region. f is distance between the center of object and center of camera, y is the real width of object and *za* is horizontal field of vision of camera.

By means of these calculations, we defined a coefficient that can express the real size of object in picture versus its distance from the center of camera. This coefficient can help to make two images comparable and stretching them by a "projective transformation" (Fig.12).

$$k = \frac{y'}{\text{width of the region}}$$

We employed normalized cross-correlation for discovering the similarity of images, if the value of cross-correlation of images is more than 0.9, we will consider them similar (Fig.13). In ideal form the value must be 1 but considering it 0.9 doesn't make mistake, on the other hand since we have employed lots of sections for finding object, it looks impossible that this insignificant difference could make error.

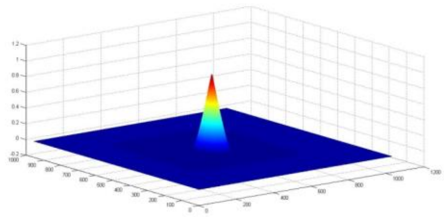


Fig.13 Result of normalized cross-correlation

After this step the existence of obstacle is confirmed and the place of that will be determined by its position in that three regions. Finally the system will notify the user as these sentences "there is an object in front of you" for region number 2," there is an object on your right" for region number 1 and " there is an object on your left" for region number 3.

III. Results and Discussions

This system was tested under three conditions and for three objects. For taking images we used CANON SX120 IS, 0.3 mega pixels, without using any automatic forms. The program was processed by DELL Inspiron 1545 notebook with processor: Intel(R) Celeron(R) CPU @ 2.20 GHz and installed memory (RAM) 2.00 GB.

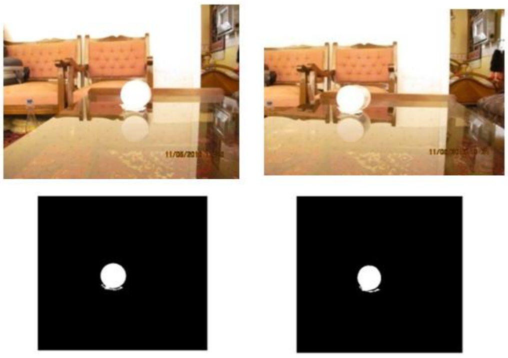
First we tested the system for a circle object under lamp light and the camera was fixed 55cm height from ground level (Fig.14-a). In Figure (14-b) square object was 70 cm far from camera but circle one was nearer, we observe that system doesn't detect the circle object. In next one square object was in same place and circle object was farer from camera, system doesn't detect the circle one like the former picture (Fig.14-c). Figure (14-d) describes the result of our experiment under sunlight for a black object and camera was fixed 70 cm height from ground level. In figure (14-e), object was in same place but camera was in 81cm height with 20 degrees of depression angle. Duration of processing was 33.7969 seconds.

In this research, our purpose is detection of those obstacles that their distance from cameras is 70Cm. Undoubtedly system has error, for estimating this error we tested the system for 4 distances 70 cm, 71 cm, 72 cm and 69 cm, the results show that system detected an obstacle for 70 and 71 cm (Fig. 14-f and 14-g) but not for 72 and 69 cm, thus if the center of coordinates is at 70cm the maximum error of system is: (72-70, 69-70) => (2, -1) cm.

IV. Conclusion

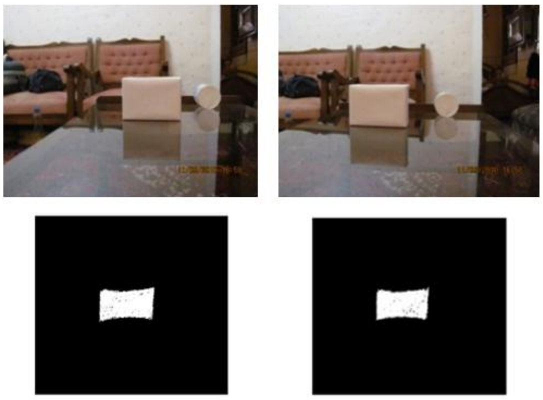
Our suggested system has high sensitivity for the height of cameras and distance between them. In this research, we designed a system with two cameras placed on blind person's glasses that their duty is taking images from different sides. By comparing these two images, we could be able to find the obstacles. In the proposed method, first we investigated the probability of existence an object by use of special points that then we called them "Equivalent points", then we utilized binary method, standardized and normalized cross-correlation for verifying this probability. This system was tested under three different conditions and the estimated error is in acceptable range.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

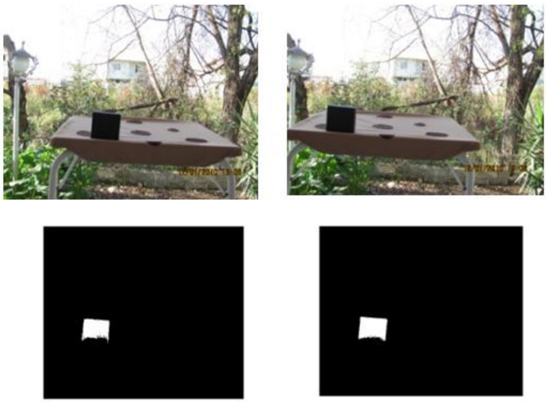


(a)

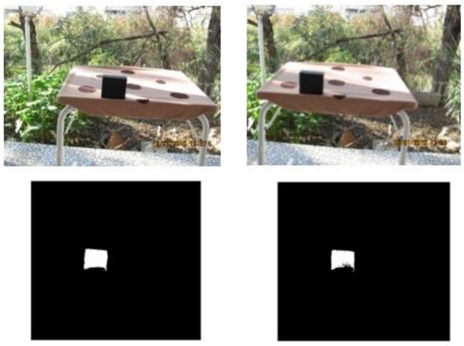
(b)



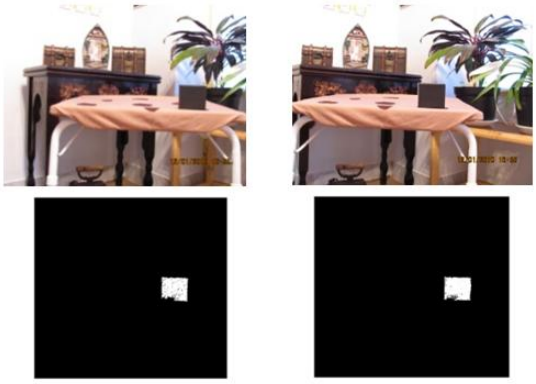
(c)



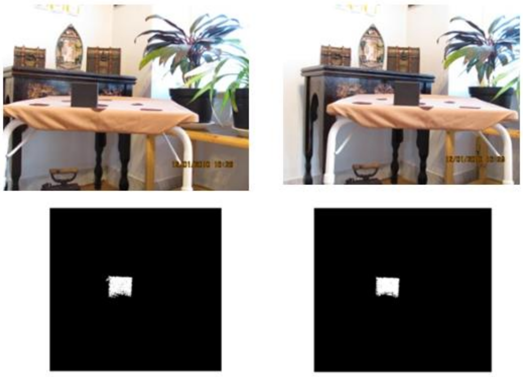
(d)



(e)



(f)



(g)

Fig.14 Results of tested system

REFERENCES

[1] Felipe Jiménez, José Eugenio Naranjo, "Improving the obstacle detection and identification algorithms of a laserscanner-based collision avoidance system," *Transportation Research Part C, Article in press.*

[2] Kunsoo Huh, Jaehak Park, Junyeon Hwang, Daegun Hong, "A stereo vision-based obstacle detection system in vehicles," *Optics and Lasers in Engineering*, Vol. 46, pp. 168–178, 2008.

[3] J. Jesus Garcia, Jesus Urena, Manuel Mazo, Felipe Espinosa, Alvaro Hernandez, Cristina Losada, Ana Jimenez, Carlos De Marziani, Fernando Alvarez, Enrique Garcia, "Sensory system for obstacle detection on high-speed lines," *Transportation Research Part C* , Vol.18, pp. 536–553, 2010.

[4] Iwan Ulrich and Illah Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision," *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, 2000.

[5] Joshua Redding, Jayesh N. Amin, Jovan D. Boskovic, Yeonsik Kang, Karl Hedrick, Jason Howlett, Scott Poll, "A Real-time Obstacle Detection and Reactive Path Planning System for Autonomous Small-Scale Helicopters", *AIAA Guidance, Navigation and Control Conference*, Carolina, USA, Aug. 2007.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)