Smartphone based Guidance System for Visually Impaired Person

Muhammad Asad
Department of Electrical and Electronics Engineering
University of Sheffield
UK
masad.801@gmail.com

Waseem Ikram
Department of Electrical Engineering
National University of Computer and Emerging Sciences
Pakistan
waseem.ikram@nu.edu.pk

***Abstract*— In order to facilitate the visually impaired person in navigation, we have developed a prototype guidance system. The main assumption of this guidance system is that there are many straight paths in different real world scenarios. These straight paths have parallel edges, which when captured as an image seem to converge to a single point called the vanishing point. Proper feature extraction and mathematical modelling of the captured frame leads to the detection of these parallel edges. The vanishing point is then calculated and a decision system is formed which notifies the blind person about his/her deviation from a straight path. The scope of this system is limited to a straight path and has been tested in different lighting conditions and with different level of occlusion. A laptop mounted on a 2D robotic platform is used to develop and verify the robustness of the algorithm. Finally, a smartphone based real-time application has been implemented for this visual guidance system, in which the decision system returns an audio output to guide the visually impaired person. This application has an average execution rate of 20 frames per second, with each frame being of 320 by 240 pixel size. The system has an accuracy of 84.1% in a scenario with pedestrians and other objects, while without pedestrians it produces an accuracy of over 90%.**

***Index Terms*— Guidance System, Real-time System, Visually Impaired, Hough Transform, Vanishing Point, Navigation**

I. Introduction

Recent developments in the field of computer vision and the emergence of different compact and powerful smart phones with high resolution cameras, made it possible for the development of different computer vision applications on these devices. This is evident by the recent porting of many real-time computer vision libraries to different mobile phone platforms. The most prominent and powerful of all is the implementation of OpenCV libraries for android based smart phones. This has given a whole new platform for research in the field of computer vision applications on mobile phones and has already resulted in many smart phones based applications which use different complex computer vision techniques [1][2][3].

Today, about 4% (285 million of the world's total population) is visually impaired. From these 13% are blind, while the rest 87% have low vision [4]. One of the biggest problem faced by these visually impaired persons is to perform everyday task. For this purpose, designers, both visually impaired and sighted, have developed a number of tools for use by the blind people.

The major mobility tools used include white canes and guide dogs. White canes are claimed to be not safe enough to cross streets and go to other insecure places. There has also been some reluctance to use these white canes, particularly by those people who are not totally blind. The much bigger problem is that these canes are not reliable as they are not giving sufficient information of the surroundings to the blind person. A small number of people employ guide dogs to assist in mobility, as it is mostly done in western countries and can be expensive or unavailable to many. Also in some cultures, dogs are socially unacceptable.

In literature, there has been a lot of research in this topic. This includes mobile phone based systems to find cross walks and paths [5][6][7]. Research has also been done using stereo vision cameras and mobility aids such as power wheelchairs [8]. Work has also been done for correcting the orientation of a blind person [9][10][11]. Most of these approaches were aimed at small devices and, hence, were computationally challenged, because of the limited hardware available. Therefore these applications were either not real-time or had very limited functionality. Some of these methods used colour targets for easy recognition of location. Most of the research was aimed at indoor or controlled environment, with less variations in illumination and was sensitive to occlusion. Similar work has also been reported for robot navigation in indoor environment [12].

In this paper, we propose a guidance system which makes use of the fact that due to the structural symmetries in all man made structures, there exist a lot of straight paths in different real world scenarios. These straight paths have many parallel edges, which humans use everyday to perceive the 3D picture. This picture however is not available to both completely blind and partially blind persons. These edges, when looked upon through a camera, converge to a vanishing point. We compute the deviation of a visually impaired person according to the location of this vanishing point in the image, which is then used to guide the visually impaired person in the straight path.

The organization of this paper is as follows: Section II explains the method used to develop the guidance for a visually impaired person. The results are presented in Sec. III followed by conclusion in Sec. IV.

II. Methodology

We propose a guidance system based on the fact that in almost every man made structure, there exists a geometrical symmetry. Human visual system uses these structures to perceive the location and assist the person in navigating in a path. However this is not the case with computer vision programs where this is viewed as a 2D image with parallel edges converging to a point [13]. Our proposed approach simulates a system which is similar to human visual system and provides the visually impaired person with the information about their deviation from the straight path.

Fig. 1: Block diagram of the proposed method, depicting the three main blocks involved in designing the guidance system

Our proposed guidance system consists of three major blocks, which are shown in Fig. 1. These blocks are discussed in detail in subsequent sections.

A. Feature Extraction

One of the most important step in any computer vision application is the extraction of relevant features. This extraction step should focus on both extracting relevant information and minimizing out all the outliers. This process can then be followed by much more complex operations only on the relevant and small set of features. An efficient feature extraction step is essential for the implementation of real-time systems. These systems have limited time to process and present results, and with an proper feature extraction technique they can be more robust and efficient. The proposed system uses several feature extraction techniques which are discussed below.

In the proposed system the features extracted are edge features of the parallel lines in straight paths. It was observed from the results of testing phase of this system, that from all the edges detected only a few prominent edges actually contributed towards the guidance decision which will be presented in Section II-B. Since the system is aimed at real-time implementation, therefore a balance between efficiency and accuracy was required. For this reason, a statistical comparison between different existing edge detectors was done, comparing both efficiency and accuracy of each operator with different variation in illumination, occlusion and different level of prominent paths. Each operator was run on different set of images of size 640 by 480 pixels and the average execution time was calculated, which is presented in Table I. Canny's edge detector being the most accurate edge detector was also computationally the most expensive one. For this prototype system, only the dominant edges were required but canny's edge detector detects edges of even small details in the image. Therefore, these small details needs to be filtered out if canny's edge detector is used, which lower the efficiency of the system. Other edge detection techniques like perwitts, sobel, LoG, Roberts and Zero Cross are very efficient, however they are not very accurate as they filter out a lot of required details. Using the edge detection technique from [14] efficient results were achieved with very less compromise over accuracy. This technique is based on the morphological operators which makes it efficient as only the dominant edges are detected with high accuracy. From Table I, it is seen that this morphological edge detection technique is almost as efficient as perwitts, however it produces much better result than any other operators with similar execution time [14]. It was also observed that taking average of the frame before applying edge detection technique improved the detection of the prominent edges. A 3x3 averaging filter was used to perform this operation. The size of the image was chosen to be 320 by 240 pixels as this size was neither too small to diminish the effect of edge detection techniques, nor it was too large to have an effect on the execution time. The images were directly acquired in grayscale format to lower the execution time of the system.

Edge Detection Techniques	Average Execution Time (ms)
Canny	905.9
LoG	346.5
Prewitts	293.2
Roberts	322.3
Sobel	341.7
Zero Cross	332.8
Morphological Edge Detection	306.6

TABLE I: Average Execution Time of different edge detection operators on images of size 640 by 480 pixels

B. Feature Selection

Feature selection process is also a trivial part of a real-time system. This process makes use of the observations and selects the best possible features for further processing. It uses the edge detected output of the feature extraction step, and selects the relevant lines in order to facilitate the guidance system with required information. This step is further divided into two steps. First step deals with mapping prominent lines into mathematical equations and the second step involves selecting lines which are relevant to the system and also connecting disconnected lines due to occlusion and illumination changes.

(a) Orignal Image

(b) Edge detection using canny's edge detector

(c) Edge detection using edge detection method from [14]

Fig. 2: Output comparison of canny's and morphological edge detection technique from [14]

The mapping of lines into mathematical equations is achieved using a feature extraction technique called Hough Transform. Hough transform is used to extract features such as lines or curves in an image [15]. This is achieved using the basic idea that if two or more set of points are collinear, then they will have a unique line associated with them. This line can be found by fitting all possible number of lines and finding the one which fits all these points. Using Cartesian coordinate system, the computation required for this is huge and in some cases the number of lines are infinitely large and becomes difficult to model. On the contrary, using hough transform reduces the computational complexity of modelling mathematical equations for these lines. In this transform, each point in Cartesian coordinate system is mapped to a parameter space defined by sinusoidal curves, which is given by:

$$\rho = x\cos\theta + y\sin\theta \tag{1}$$

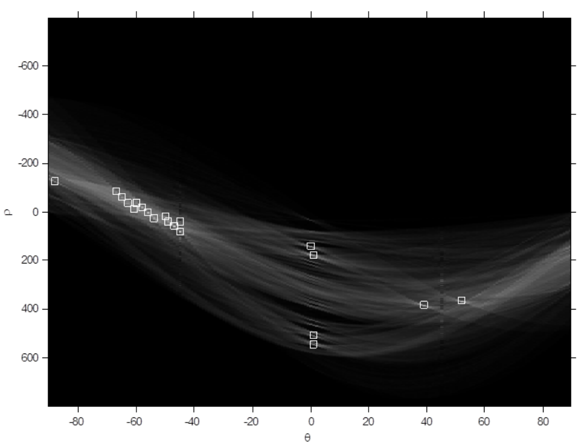

Fig. 3: Hough transform of Fig. 2(c) with lines represented by maximum number of curve intersections (shown here using small white squares)

The collinear points correspond to sinusoidal curves which intersect at some arbitrary value (ρ, θ). Computing the points where maximum number of curves intersect, gives the (ρ_i, θ_i) parameters of a line. Fig. 3 shows the results obtained by applying hough transform on the binary image shown in Fig. 2(c) with the points where maximum number of curves intersect. These parameters can then be used to calculate gradient and y-intercept of line using eq. (2) and (3).

$$m_i = \tan\theta \tag{2}$$

$$c_i = y - m_i x \tag{3}$$

From the above equations it can also be noted that θ represents the orientation of the line with ρ being the perpendicular distance of the given line to the origin. Further explanation of this can be found in [14].

From Fig. 3, it is observed that maximum number of lines have a θ between -70 to -40 or 40 to 70 degrees. This is due to the detection of prominent edge in the straight path in Fig. 2. In addition to these lines, a noticeable amount of lines are seen at θ equal to 0 and 90 degrees. These lines are due to the horizontal and vertical edges detected as a result of object which do not contribute towards a continuous straight path, for example doors and windows in this case. These lines can be discarded, and the former set of lines is selected for computing the guidance decision in the next step of this system. A major problem faced in this step is the identification of occluded edges and the interpolation of many disjoint parts of a line into one single line. This problem is solved by using hough transform as even if the line is made up of many disjoint lines, hough transform always has single pair of (ρ_i, θ_i) corresponding to one line in Cartesian coordinate system.

C. Guidance Decision

The vanishing point is a point to which all the edges of a straight path converge. In reality, these edges are parallel to each other, however when looked at from a specific point, they all seem to converge to one point. Using the mathematically mapped equations from previous step, this imaginary vanishing point can be found easily [16]. The gradient and y intercept corresponding to each point (ρ_i, θ_i) in hough transform is used to solve equations

Fig. 4: Extraction of vanishing point from intersection of lines. Points in blue are the intersection points, while point in magenta color is the vanishing point calculated by taking median of all the intersection points

of different lines simultaneously with each other. These points are not always inside the range of the frame as vanishing point itself is imaginary point, therefore these intersection points are also imaginary and sometimes are even out of the range of the image. This results in a cluster of a lot of intersection points with many points near the actual vanishing point and some outliers as a result of edges which do not converge to the vanishing point. The effect of these outliers can be minimized while finding vanishing point by taking median of all these points. Median filter ranks all the points and uses the region where many intersection points exist to define vanishing point. Figure 4 shows the calculation of the vanishing point on the frame from Fig. 2(a). The points in blue show the intersection points of the lines and the magenta coloured point is the vanishing point which arises by taking median of all these intersection points.

The vanishing point is then used along with two vertical markers, to determine three different decisions. These decisions are based on the horizontal position of the vanishing point in one of the three regions defined by these two markers. This process of decision making is based on the fact that vanishing point always stays in the middle of the path, as all the lines in a path converge to this point. In case the visually impaired person, with camera facing directly in front of him, deviates from the straight path, the frame from his camera shifts to either side depending on the direction of his deviation. Since vanishing point arises from converging edges, it always remains in the centre of the path. Hence with a deviation in frame, this point also shifts towards either side. This deviation of the vanishing point makes it move in different regions as defined by vertical markers. Based on the vanishing point's location in either of these regions, the decision about the heading of the person is made and accordingly one of the three instructions are given to the visually impaired person to correct his path, i.e. go straight, go left and go right. The number of deviation markers can be raised to make more decision regions and specific and accurate guidance according to transition of the vanishing point in either of these regions. This decision making process is explained in Figure 5 with all three possible situations.

Sequence	Accuracy
Mobile Scene Analysis	84.1%
Hall	90.4%
Highway	95.5%

TABLE II: System accuracy on different sequences

III. Results

The algorithm was tested in different lighting conditions both outdoors and indoors. A laptop mounted on a 2-D robotic platform was used for testing and development of this algorithm. Later, a smart phone application was developed using OpenCV® libraries for android and the algorithm was tested. For testing, the smart phone was mounted onto the chest of a user, so that it was facing directly into the direction the person was moving in. The guidance decision was then given to the user using an audio signal. This algorithm was tested in many straight paths, with and without other pedestrians. It was observed that the algorithm correctly guided the person even if there were other pedestrians occluding the path partially. This algorithm produced an average execution rate of 20 frames per second on Galaxy® Nexus S smart phone.

To test the accuracy of this algorithm in real world environment, image sequence # 0 for [17] was used, which contained a total of 499 RGB frames from left camera. These frames were first classified into one of the three possible decision categories using the actual location of vanishing point. The algorithm was then run with maximum number of lines selection limit of 20 lines from hough transform. Out of 499 frames, 420 frames output had correct decisions. This made the overall system accuracy of 84.1% in real world path with pedestrians. The algorithm was also tested on Hall sequence and Highway sequence [18]. Table II presents the system accuracy for each of these sequences. The proposed approach produced 90.4% accuracy on Hall Sequence and 95.5% accuracy on Highway Sequence. The results from some of the frames are shown in Fig. 6, 7, 8 and 9 at the end.

IV. Conclusion

A guidance system for visually impaired person is proposed. This system makes use of the vanishing point concept to make a guidance decision for the visually impaired person. A real-time application on an android smart phone was developed for this guidance system,

(a) Decision -> Go Straight (b) Decision -> Turn Left (c) Decision -> Turn Right

Fig. 5: Guidance decision making process with two vertical decision markers (shown in black) and relative position of vanishing point(shown by red markers)

giving an average execution of 20 frames per second and an accuracy of 84.1% with pedestrians and above 90% without pedestrians. This difference in accuracies is due to the fact that some sequences have a lot of occlusion due to pedestrians and the background is also cluttered. This system uses the edges from different surrounding structures, hence if these edges are not present it loses its accuracy.

This algorithm can also be used in many more applications involving navigation of different vehicles. Future work may look into paths which are not straight, in more complex environments and with a lot of variations in occlusion. The concept of interpolation of lines can be used to cater for many real world scenarios where a large part of the edge information of paths is occluded.

There exist many reliable and robust methods for calculating vanishing points in a scene [19], [20] and [21]. We plan to use a similar method to further improve the results of this algorithm, making it more reliable and robust.

References

[1] W. Oui, E. Ng, and R. Khan, "An augmented reality's framework for mobile," in *Information Technology and Multimedia (ICIM), 2011 International Conference on*. IEEE, 2011, pp. 1–4.

[2] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *Proceeding of the 1st ACM international conference on Multimedia information retrieval*. ACM, 2008, pp. 427–434.

[3] V. Paelke and C. Reimann, "Vision-based interaction-a first glance at playing mr games in the real-world around us," in *Proceedings of the 2nd International Workshop on Pervasive Gaming Applications (PerGames) at ERVASIVE*, vol. 2005, 2005.

[4] http://www.who.int/mediacentre/factsheets/fs282/en/, "Statistics about blindness and eye disease," vol. 2011, October 2011.

[5] H. Shen, K. Chan, J. Coughlan, and J. Brabyn, "A mobile phone system to find crosswalks for visually impaired pedestrians," *Technology and disability*, vol. 20, no. 3, pp. 217–224, 2008.

[6] V. Ivanchenko, J. Coughlan, and H. Shen, "Staying in the crosswalk: A system for guiding visually impaired pedestrians at traffic intersections," *Assistive technology research series*, vol. 25, no. 2009, p. 69, 2009.

[7] J. Coughlan, R. Manduchi, and H. Shen, "Cell phone-based wayfinding for the visually impaired," in *1st International Workshop on Mobile Vision*, 2006.

[8] V. Ivanchenko, J. Coughlan, W. Gerrey, and H. Shen, "Computer vision-based clear path guidance for blind wheelchair users," in *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 2008, pp. 291–292.

[9] V. Ivanchenko, J. Coughlan, and H. Shen, "Crosswatch: a camera phone system for orienting visually impaired pedestrians at traffic intersections," *Computers Helping People with Special Needs*, pp. 1122–1128, 2008.

[10] S. Se, "Zebra-crossing detection for the partially sighted," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2. IEEE, 2000, pp. 211–217.

[11] V. Ivanchenko, J. Coughlan, and H. Shen, "Detecting and locating crosswalks using a camera phone," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE, 2008, pp. 1–8.

[12] E. Bayramo, N. Andersen, N. Poulsen, J. Andersen, and O. Ravn, "Mobile robot navigation in a corridor using visual odometry," *Control*, 2009.

[13] L. Quan and R. Mohr, "Determining perspective structures using hierarchical hough transform," *Pattern Recognition Letters*, vol. 9, no. 4, pp. 279–286, 1989.

[14] Y. Zhao, W. Gui, and Z. Chen, "Edge detection based on multi-structure elements morphology," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 2. IEEE, 2006, pp. 9795–9798.

[15] R. Duda and P. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[16] S. Barnard, "Interpreting perspective images," *Artificial intelligence*, vol. 21, no. 4, pp. 435–462, 1983.

[17] A. Ess, B. Leibe, and L. V. Gool, "Depth and appearance for mobile scene analysis," in *International Conference on Computer Vision (ICCV'07)*, October 2007.

[18] http://trace.eas.asu.edu/yuv/, "Yuv video sequences."

[19] J. Choi, W. Kim, H. Kong, and C. Kim, "Real-time vanishing point detection using the local dominant orientation signature," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*. IEEE, 2011, pp. 1–4.

[20] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I–470.

[21] J. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1250–1257.

(a) Original image (b) Edge detection output (c) Guidance decision (vanishing point in magenta color)

Fig. 6: Result: Pedestrian Sequence (Decision: Go Straight)

(a) Original image (b) Edge detection output (c) Guidance decision (vanishing point in magenta color)

Fig. 7: Result: Pedestrian Sequence (Decision: Go Right)

(a) Original image (b) Edge detection output (c) Guidance decision (vanishing point in magenta color)

Fig. 8: Result: Hall Sequence (Decision: Go Straight)

(a) Original image (b) Edge detection output (c) Guidance decision (vanishing point in magenta color)

Fig. 9: Result: Highway Sequence (Decision: Go Straight)