Navigation Assistance for Individuals with Visual Impairments in Indoor Environments

Rupam Kundu[1], Gopi Krishna Tummala[1] and Prasun Sinha[1]

Department of Computer Science and Engineering, The Ohio State University
kundu.24@osu.edu, tummala.10@osu.edu, sinha.43@osu.edu

Abstract. Canes or service dogs in indoor environments are unable to provide spatial information to the Individuals with Visual Impairments (IVIs) to make them independent. An indoor navigation assistance system can provide information on the presence of any obstacles in their vicinity, the distance of separation and their direction of motion (in case of mobile objects) w.r.t the IVIs. In this paper, we attempt to address the above objective by designing a novel time-efficient algorithm where a smart-glass is employed to spot an obstacle (stationary or mobile) in indoor environment using the inbuilt camera and inertial sensors. The system is implemented and tested extensively in indoor settings.

1 Introduction

Obtaining the spatial information around us is inevitably necessary to perform our day-to-day activities and to move around safely without colliding with other objects. However, Individuals with Visual Impairments (IVIs) are devoid of this crucial human functionality. According to WHO's announcement in 2014 [5], around 285 million people are victims of visual impairments worldwide out of which 39 million are blind and the rest suffer from low vision. Out of these 39 million (30% of world blind population), 12 million blind people in India alone [6]. So the urge to restore the vision functionality in IVIs to ensure an independent and a comfortable life, has drawn the attention of many researchers from diverse domains.

Recently, various wearable frameworks have been devised for detecting obstacles in the user's vicinity. They belong to one of the four categories stated below:

Ultrasonic Sensors: Ifukube et al.[12] used two ultrasonic sensors which gather spatial information based on reflections of the transmitted wave from various objects. Ultrasonic sensors are highly directional (resolution of 1 mm) and require mechanical movements to pinpoint obstacles in different directions [4]. Shovel et al [17] introduced a belt, a portable computer and an array of ultrasonic sensors that scan all the signals arriving at the sensors. The hardware is bulky, and the technique requires training.

Infrared: An Infrared sensor can be used to compute the depth of an object by detecting the phase shift of the modulated light reflected from the target. The range of detection is 10 centimeters to 1.5 meters with 95% accuracy [8]. However, the depth maps become noisy in presence of sunlight (since it contains infrared light) [22]. The Infrared sensors are also highly directional and require mechanical panning to detect objects in different directions.

Radar: RADAR operates similar to ultrasound navigation. Instead of acoustic waves, radio waves are employed to measure the Time-of-Flight as the signals bounce back from nearby obstacles. However, due to the high directionality of RADAR, either a mechanical movement is required or an array of such sensors need to be employed resulting in a bulky design. Moreover, commercially available RADARS and LIDARS consumes high power (nearly 8 watts) [7].

Stereo: "Smart-Vision" introduced by Fernandes et al. [11] used several independent modules like GPS, RFID, Vision and Wi-Fi but it can detect only specific landmarks. Balakrishnan et al. [9] and Vimal et al. [16] proposed to identify obstacles in the user's vicinity based on stereo disparity. As the disparity is measured using pixel based operations, these processes are computationally heavy. Moreover, small depth variations for far away objects are hard to interpret using stereo disparity techniques. For a baseline of 28 mm, image sensor pixel size of 17um, focal length of 2.8mm and a disparity range of 5-35, Khaleghi et al [13] showed that if an object is detected at 1 m, the error is around 20 cm. The error increases non-linearly for higher ranges. Also, commercially available long range stereo cameras consume high power (nearly 4 watts) [1], while the small ones [2] have very short range (2.5 meters).

RFID: Ding et al. [10] proposed the use of RFID readers on canes and RFID tags along navigation paths so that the reader can read pre-installed information from the tags. The tag installations require modification to the infrastructure and also information about moving objects cannot be captured using this technique.

Advanced positioning and tracking services can be used along with smart-glass to emulate stereo across the motion of the smart-glass. Smart-glasses, such as Google-glass and Microsoft HoloLens[19], are integrated with sensors such as camera, accelerometer, magnetometer and gyroscope. Also, they are capable of communicating with the infrastructure or a paired mobile phone using wireless technologies. However, unlike stereo, the smart-glasses are usually equipped with a single camera. Recent advancements in wireless localization and tracking techniques using wireless radio [23, 14], RF-ID's [24] and camera [15, 21] can achieve sub-meter level accuracies. These location-tracking techniques can be used to relate multiple frames captured at different spatial points to estimate the depth of different objects. So the key question we raise in this context is: *Can we estimate the depth of obstacles in an indoor environment using a smart-glass to provide a vision based navigation aid for IVIs?*

Fusing multiple frames across the motion trajectory of a smart-glass is challenging due to - *location errors in the trajectory*, *lack of information about objects in the environment*, *dynamic objects*, *limited-range* and *run-time complexity* of different multi-frame fusion algorithms. In this paper, we design a novel and time-efficient algorithm for assisting IVIs for enabling them to navigate independently in indoor settings using vision in a smart-glass framework that can be implemented in real time. The following are the contributions of this paper:

- Novel algorithms to spot an obstacle in the IVI's path (stationary or dynamic) and also to figure out the direction of motion of the dynamic object w.r.t. IVI.
- Proof of concept implementation, experiments and evaluation of modules for static object sensing and dynamic object sensing.

2 Challenges

- **Lack of depth information:** Video feed captured by commodity smart glass lacks - **(a)** depth information of the captured objects, **(b)** association of the observed objects with their real entities.
- **Object Uncertainty:** Neither the object's shape nor its location or mobility information (stationary or dynamic) is known. Vision-based modules are not suited for real time implementation when there are many of objects in the scene.

3 Vision based Tracking for IVIs

3.1 Background

The *Pinhole camera model* [20] describes the geometric relationship between the 2D image-plane (i.e, pixel positions in a camera capture) and the 3D ground coordinate system. Let the image plane be the UV-plane and the camera coordinate system be the (XYZ) space. Let us assume that the perpendicular ray emanating from the center of the camera frame is along the Z-axis, V-axis is parallel to Y-axis and U-axis is parallel to X-axis (see Figure 1). The geometrical relationship is given by,

$$\begin{aligned}\frac{u_1}{f_u} &= \frac{x_1}{z_1}, \\ \frac{v_1}{f_v} &= \frac{y_1}{z_1},\end{aligned} \tag{1}$$

where f_u and f_v are the focal lengths of the camera.

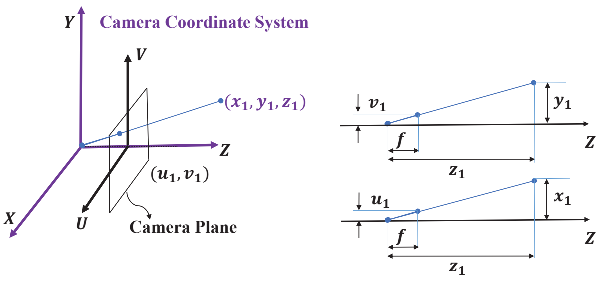

Fig. 1: *Geometric relationship between image plane and the camera coordinate system. Illustration assumes* $f = f_u = f_v$

3.2 Design

The system is implemented as a smart-glass app that uses a camera in a Google-glass framework and identifies feature points in the environment. Feature points are distinguishable points such as corners or the edges of a shirt etc., which can be tracked across consecutive image frames using optical-flow based techniques [21]. The user wearing a smart-glass walks in an indoor environment. The camera records a short video which is processed to identify feature points corresponding to the objects across multiple video frames. Using the coordinates of the observed feature points in camera frame along with the location of the user, the location of the object can be determined. The location of the user can be obtained from any localization services such as WiFi[14], Visual Light Communication [15] or Fingerprinting[18].

To spot an obstacle in an IVI's path, we need to consider two scenarios (a) When the user is moving towards a static object, e.g., a wall, door or tables; and, (b) When the user is approaching a moving object, e.g., a human passing by. We discuss the above cases in the subsequent sections.

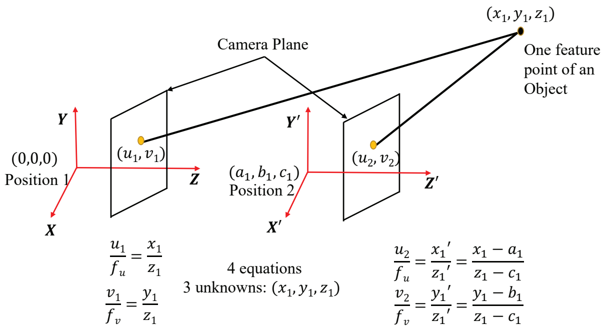
(x_1, y_1, z_1)
One feature point of an Object
Camera Plane
Y
Y'
$(0,0,0)$
Position 1
(u_1, v_1)
Z
(a_1, b_1, c_1)
Position 2
(u_2, v_2)
Z'
X
X'
$\frac{u_1}{f_u} = \frac{x_1}{z_1}$
$\frac{v_1}{f_v} = \frac{y_1}{z_1}$
4 equations
3 unknowns: (x_1, y_1, z_1)
$\frac{u_2}{f_u} = \frac{x_1'}{z_1'} = \frac{x_1 - a_1}{z_1 - c_1}$
$\frac{v_2}{f_v} = \frac{y_1'}{z_1'} = \frac{y_1 - b_1}{z_1 - c_1}$

Fig. 2: *Static Case: Set of equations over-constrained*

3.3 Static Case

We consider the static case when the user is moving while the object is static as illustrated in Figure 2. Let us assume that the user moves from Position 1 (P_1) - $(0,0,0)$ to Position 2 (P_2) - (a_1, b_1, c_1) while the feature point associated with an object (x_1, y_1, z_1) is static. For sake of simplicity, let us assume the orientation of the camera is not changing from P_1 to P_2 (later this assumption is relaxed). *All the coordinates in the rest of the paper are measured according to camera coordinate system when it is at P_1*. In particular, this coordinate system has its XY-Plane aligned with camera plane, and the Z-axis is perpendicular to the camera plane. Let P_1 and P_2 correspond to two frames in a video feed which are analyzed to sense object depths. Due to the translation, the new coordinate system at P_2 is a translated version of the coordinate system at P_1. Therefore, the coordinates of the point located at (x_1, y_1, z_1) will be $(x_1', y_1', z_1)'$ in coordinate system at position 2, where $x_1' = x_1 - a_1$, $y_1' = y_1 - b_1$ and $z_1' = z_1 - c_1$.

Following the Pinhole Camera Model, two equations can be written for each position similar to Equation (1) as shown in Figure 2. Analyzing a set of two frames, the change in the relative location of the object w.r.t. the user can be determined. Geometric representation of a feature point with pixel positions (u, v) represents a ray emanating from the camera. The translation of the user - (a_1, b_1, c_1) can be obtained from standard localization services. Therefore, using four equations along with the user's location, the object's position (x_1, y_1, z_1) can be determined. The intersection of two rays from P_1 and P_2 gives the location of the feature point. Essentially, this resembles stereo to compute the depth of an object.

XYZ to X'Y'Z' coordinate system: For simplicity, we have shown in Figure 2 that the camera plane at P_1 and P_2 are aligned. In reality, the camera orientation might change from P_1 to P_2 and the user might be moving in an arbitrary direction. We use translation of camera from P_1 to P_2 w.r.t camera coordinate system at P_1, $\mathbf{T_{12}}$ (3×1) obtained using a localization technique. The Rotation matrix from the coordinate system at P_1 to P_2, $\mathbf{R_{12}}$ (3×3) is obtained from MEMS gyroscope present in smart-glass. Using the translation and rotation, the position of the feature point in the two coordinate systems can be related as,

$$\begin{bmatrix} x'_1 \\ y'_1 \\ z'_1 \end{bmatrix} = \mathbf{R_{12}} \times \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \mathbf{T_{12}}. \tag{2}$$

Gyroscopes are known to accumulate error, but since we are using orientation information for two closely spaced frames, the error accumulated is negligible.

3.4 Dynamic Case

We now consider a scenario in which the feature point corresponding to the object is also moving as the user holding the camera moves. In Figure 3, the feature point of an object moves from O-(x_1, y_1, z_1) to P-(x_2, y_2, z_2) as the user carrying the camera moves from P_1 - $(0, 0, 0)$ to P_2 - (a_1, b_1, c_1). For the sake of simplicity, we assume that the orientation of the camera is same at P_1 and P_2. In Figure 3 the two rays intersect at a point 'A' which is not the true current location of the obstacle. This provides an incorrect estimation to the IVI. Figure 3 shows the entire range of possible wrong solutions for different movements of the feature point.

Exploring multiple points belonging to dynamic object: To overcome the above problem, we make use of multiple feature points. Suppose the camera observes n feature points from a particular object at both P_1-$(0, 0, 0)$ and P_2-(a_1, b_1, c_1). P_1 and P_2 correspond to two different frames from the video feed. The location coordinate of the n feature points - $(x_i^j, y_i^j, z_i^j), 1 \leq i \leq n, \forall j \epsilon [1, 2]$ at two positions have $6n$ unknowns. Using the Pinhole Camera Model, for each feature point (x_i^j, y_i^j, z_i^j) at each position, two equations can be written similar to Equation (1). This generates $4n$ equations for two positions. Additionally, the velocity vector along x, y and z axes for all the feature points are equal since they belong to the same object providing $3.(n-1)$ equations. These equations can be written as $V_i^x = V_j^x$, $V_i^y = V_j^y$ and $V_i^z = V_j^z$ where $i, j \in [1, n]$. Moreover, since we are considering closely spaced frames, we can assume the height has not changed from P_1 to P_2 which provides one more equation since the velocity along y axis will be zero $V_i^y = V_j^y = 0$ where $i, j \in [1, n]$. Therefore, there are $6n$ unknowns and $4n + 3(n-1) + 1$ equations. For solving these sets of equations, the following relation needs to hold:

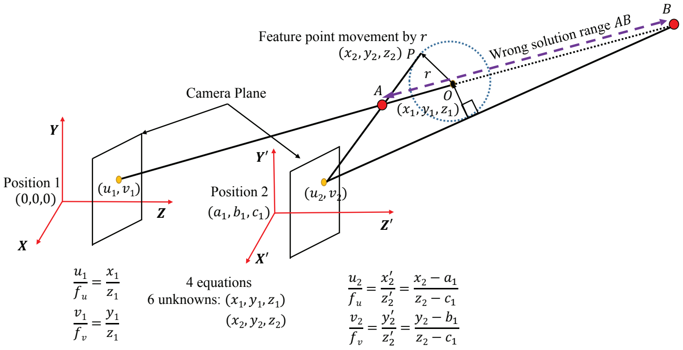
Feature point movement by r
(x_2, y_2, z_2) P
Wrong solution range AB
B
A
r
O
(x_1, y_1, z_1)
Camera Plane
Y
Position 1
(0,0,0)
(u_1, v_1)
Z
X
Y'
Position 2
(a_1, b_1, c_1)
(u_2, v_2)
Z'
X'
$\frac{u_1}{f_u} = \frac{x_1}{z_1}$
$\frac{v_1}{f_v} = \frac{y_1}{z_1}$
4 equations
6 unknowns: (x_1, y_1, z_1)
(x_2, y_2, z_2)
$\frac{u_2}{f_u} = \frac{x_2'}{z_2'} = \frac{x_2 - a_1}{z_2 - c_1}$
$\frac{v_2}{f_v} = \frac{y_2'}{z_2'} = \frac{y_2 - b_1}{z_2 - c_1}$

Fig. 3: *Dynamic Case: Both the user and the object are moving*

$$\begin{aligned} 4n + 3(n-1) + 1 &\geq 6n \\ \Rightarrow n &\geq 2 \end{aligned} \tag{3}$$

For sensing depth of dynamic objects at least two feature points from it must be identified and tracked across two closely spaced frames.

Object detection modules to group feature points: To identify feature points from the same object, object boundaries need to be determined. Object detection modules like '*face detection*' are already available on Android [3]. Once the object is detected, its boundaries can be identified and the corresponding feature points within that object contour can be deduced. Here also, for simplicity, we have shown in Figure 3 that the camera plane is aligned with the ground coordinate system and the user moves along the Z-axis. We have used the same technique as mentioned in §3.3.

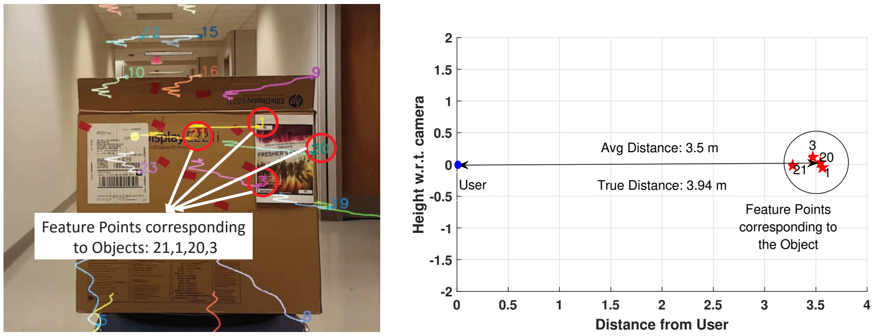

Fig. 4: *Depth estimation at 3.94 m*

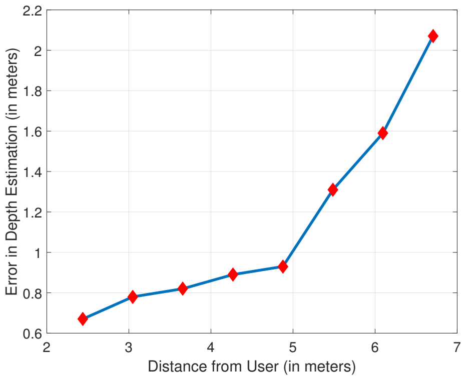

Fig. 5: *Error in Depth estimation in Static Case*

4 Experiments

The system is implemented in Python which uses open source computer vision library (OpenCV) to track different future points and the static and dynamic object sensing modules are implemented in Matlab. The video is recorded using a Samsung Galaxy S6 phone and the videos are analyzed offline. With this implementation, we have performed some *Proof of Concept* experiments to test the feasibility of our design in sensing both the *Static* and *Dynamic* objects. We have also assumed perfect knowledge about the absolute location of a user. We plan to work in more real settings using an accurate indoor localization service in future extensions.

Static Case: In this case, we placed an object at different distances along a straight line, starting from 3.95 meters w.r.t the user. A camera mounted on a moving cart is moved following the same straight line for a distance of 1.22 m for all the sets of experiments. The ground truths are marked with red colored markers and were documented using a different camera. A sample error measurement in sensing depth information for one of the experiments is shown in Figure 4. The average depth estimation error vs Object depth is shown in Figure 5.

Dynamic Case: In this case, a person is asked to take a step within a circle of 0.6096 m (2 feet) circle at known locations, starting from 3.95 meters w.r.t. the object. A camera mounted on a moving cart is moved in a straight line for a distance of 1.22 m for all the sets of experiments. The average depth estimation error for movement at various separation distances from the user is shown in Figure 6.

Analyzing Figure 5 and Figure 6, we observe for closer separation distances (within 5 meters), the *Static* and *Dynamic* cases provide comparable errors close to 1 meter. However, as the distance increases beyond 5 meters, the error shoots up due to:

- Reduction in the number of feature points.
- Feature points belonging to the edges get merged with other background objects as the feature point moves.

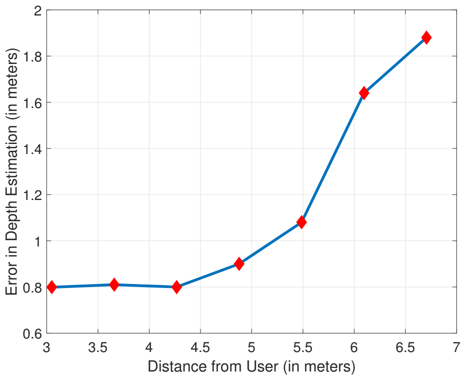

Fig. 6: *Error in Depth estimation in Dynamic Case*

5 Conclusion and Future Work

We proposed a novel time-efficient solution for assisting IVIs in navigating independently in indoor scenarios. The proposed system is built on a smart-glass framework and uses it's camera to capture video of the scene in front of an IVI. Both static and dynamic obstacles can be pointed out using our technique.

Bibliography

[1] Bumblebee XP3 3D. https://www.ptgrey.com/bumblebee-xb3-1394b-stereo-vision-camera-systems-2

[2] Duo 3D Camera. https://duo3d.com/

[3] Face Detection. https://developer.android.com/reference/android/media/FaceDetector.html

[4] Ultrasonic sensors Find Direction And distance. http://www.maxbotix.com/documents/MaxBotix_Ultrasonic_Sensors_Find_Direction_and_Distance.pdf

[5] Visual impairment and blindness. http://www.who.int/mediacentre/factsheets/fs282/en/

[6] Visual Impairment and Blindness statistics India. http://www.sightsaversindia.in/

[7] VLP 16 Specs. http://velodynelidar.com/docs/manuals/VLP-16%20User%20Manual%20and%20Programming%20Guide%2063-9243%20Rev%20A.pdf

[8] Al-Fahoum, A.S., Al-Hmoud, H.B., Al-Fraihat, A.A.: A smart infrared microcontroller-based blind guidance system. Active and Passive Electronic Components 2013 (2013)

[9] Balakrishnan, G., Sainarayanan, G., Nagarajan, R., Yaacob, S.: A stereo image processing system for visually impaired. International Journal of Information and Communication Engineering 2(3), 136–145 (2006)

[10] Ding, B., Yuan, H., Jiang, L., Zang, X.: The research on blind navigation system based on rfid. In: 2007 International Conference on Wireless Communications, Networking and Mobile Computing. pp. 2058–2061. IEEE

[11] Fernandes, H., Costa, P., Filipe, V., Hadjileontiadis, L., Barroso, J.: Stereo vision in blind navigation assistance. In: World Automation Congress (WAC), 2010. pp. 1–6. IEEE (2010)

[12] Ifukube, T., Sasaki, T., Peng, C.: A blind mobility aid modeled after echolocation of bats. IEEE Transactions on biomedical engineering 38(5), 461–465 (1991)

[13] Khaleghi, B., Ahuja, S., Wu, Q.J.: A new miniaturized embedded stereo-vision system (MESVS-I). In: Computer and Robot Vision, 2008. CRV'08. Canadian Conference on. pp. 26–33. IEEE

[14] Kumar, S., Gil, S., Katabi, D., Rus, D.: Accurate indoor localization with zero start-up cost. In: Proc of ACM MobiCom 2014. pp. 483–494

[15] Kuo, Y.S., Pannuto, P., Hsiao, K.J., Dutta, P.: Luxapose: Indoor positioning with mobile phones and visible light. In: Proc of ACM MobiCom 2014. pp. 447–458

[16] Mohandas, V., Paily, R.: Stereo disparity estimation algorithm for blind assisting system. CSI Transactions on ICT 1(1), 3–8 (2013)

[17] Shoval, S., Borenstein, J., Koren, Y.: The Navbelt-A computerized travel aid for the blind based on mobile robotics technology. IEEE Transactions on Biomedical Engineering 45(11), 1376–1386 (1998)

[18] Shu, Y., Shin, K.G., He, T., Chen, J.: Last-Mile Navigation Using Smartphones. In: Proc of ACM MobiCom 2015. pp. 512–524

[19] Statt, N.: Microsofts HoloLens explained: How it works and why its different (2015)

[20] Szeliski, R.: Computer vision: algorithms and applications. Springer Science & Business Media (2010)

[21] Tummala, G.K., Kundu, R., Sinha, P., Ramnath, R.: Vision-track: Vision Based Indoor Tracking in Anchor-free Regions. In: Proc of the ACM HotWireless 2016

[22] Viswanathan, P., Boger, J., Hoey, J., Mihailidis, A.: A comparison of stereovision and infrared as sensors for an anti-collision powered wheelchair for older adults with cognitive impairments. In: Proc of the 2nd International Conference on Technology and Aging. Citeseer (2007)

[23] Xiong, J., Jamieson, K.: ArrayTrack: a fine-grained indoor location system. In: Presented as part of USENIX NSDI 13. pp. 71–84

[24] Yang, L., Chen, Y., Li, X.Y., Xiao, C., Li, M., Liu, Y.: Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. In: Proc of the ACM MobiCom 2014. pp. 237–248