Robot Guide : Social Robot-based Intelligent Navigation of a Visually Impaired Person in Human Aware Indoor Environment

Vishnu T P[1], *Student Member, IEEE* and Abhra Roy Chowdhury*[2], *Senior Member, IEEE*

***Abstract*— This research aims to equip a social robot to assist visually impaired individuals to navigate and better perceive their surroundings to socially engage with people. The work focuses on enabling the visually impaired user to identify one's destination in a social scenario and move independently towards it. A novel Group Aware Pose Estimation(GAPE) algorithm, to identify a position for the new group member to join is implemented. A modified RRT* for path planning, an obstacle avoidance framework and follower aware controller is used to guide the user towards the goal. Multimodal feedback is provided through audio and custom-designed hand-held direction indicator "NAVI-Stick". The results obtained for the proposed GAPE algorithm works in real-time with pose metric(based on Field of view) evaluated to be over 90% for groups less than 6 members. Trials were conducted for guiding a blindfolded person towards the desired human group. The maximum Euclidean distance between user's final position and final goal position for N=15 trials is 0.98m. The average Euclidean distance-based positional error of the follower aware controller is 0.0943m.**

***Index Terms*— Social robot, Intelligent guidance, Social navigation, Human-Robot Interaction, Human Pose Estimation,**

I. INTRODUCTION

As humans continue to explore the extent to which robotics can influence human lives, it is essential that all sections of society benefit from its applications. Globally, around 1 billion people have moderate to severe vision impairment[1]. It is extremely unfortunate that the visually impaired individuals continue to face concerns with independent navigation and accurate perception of their surroundings[2]. Many such individuals struggle to independently decide their desired goal position and navigate in an unfamiliar indoor environment. Moreover, in a social setup, a visually impaired person lacks the awareness to identify and join a group of people. The main motivation of our work is to assist a visually impaired person to engage in social interactions effectively.

Robotic canes and robotic walkers [3]–[5] are major types of navigation aids that are prevalent and are becoming popular. [6] improves the paradigm of a walker by better perception of the environment and using haptic and audio feedback to guide the user. [7] uses leash tension to describe hybrid human robot interaction to implement a guide dog like robot for guiding the visually challenged. In a social scenario, the navigation needs to incorporate spatial affordances and human groups as highlighted in [8]. However, most of these works fail to use social cues from the environment to make the user more independent. Moreover, the haptic devices used are expensive. Guiding a person in a social scenario requires safety and social interactions to be considered.The work in [9] modifies the motion primitives of RRT for social aware navigation. For effective guidance robots, the notion of a user following the robot needs to be incorporated during trajectory planning and obstacle avoidance. [10] tracks the human effectively and plans using a human-robot model.

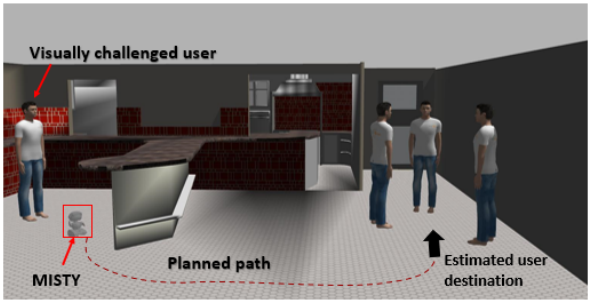


Fig. 1: Gazebo simulation environment depicting the targeted scenario

Following are the main contributions of this paper:

- An algorithm for identifying a goal pose for the visually impaired user, to join the group of people standing at a distance is implemented.
- Path planning using RRT* with modifications is used. An obstacle avoidance framework is implemented with trajectory optimization based on guidance scenarios.
- Follower aware controller is used for guiding the user to the goal.
- A custom-made handheld direction indicator "NAVI-Stick" was designed and fabricated to assist the visually impaired person to follow the robot.
- Implementation and experimental validation of the proposed system for the visually impaired in social scenarios.

II. METHODOLOGY

The intended human user of the social robot is a visually impaired person in an indoor hall in a social gathering like a conference. There are groups of people interacting with each other. One person from a group calls the user to join his group by calling out his name and raising his hand to produce a calling gesture. This work implements necessary components on a social robot to perceive and guide the visually impaired person in this scenario(Fig. 1).

[1]Vishnu T P, vishnu096@gmail.com

[2]Prof. Abhra Roy Chowdhury (corresponding author), Centre for Product Development and Manufacturing, Indian Institute of Science, Bangalore abhra@iisc.ac.in

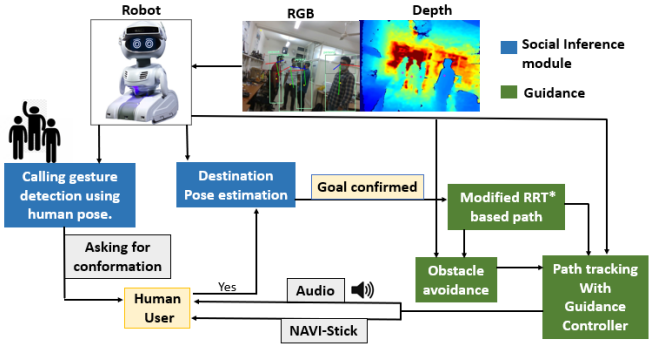


Fig. 2: Process flow of the implemented system.

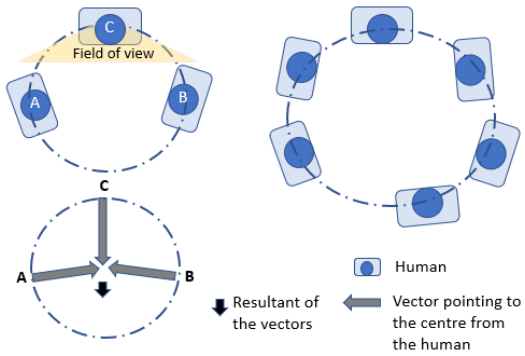


Fig. 3: Different configurations in which people group themselves. Circle is fitted to use symmetry in the arrangement. The vector representation is used to infer from the assumed symmetry.

Algorithm 1 Group Aware Pose Estimation(GAPE)

Require: $X_i =< x_i, y_i >$, N= Number of humans in the group.

1: $X_c \leftarrow \frac{(\sum_i^N X_i)}{N}$
2: **while** $change > threshold$ **do**
3: $\min_{X_c} (\sum_{i=0}^{m} pf_i(x)^2)$
4: $f_i = (\frac{\sum_{j=0}^{N} \sqrt{(x_j - x_c)^2 + (y_j - y_c)^2}}{N})$
5: p = scalar valued function
6: $change = current - previous\ estimate\ of\ X_c$
7: **end while**

Require: $\vec{X_i}$ is vector pointing from X_i to X_c *and* $R = Radius\ of\ circle$

8: $\vec{S} = \sum_{i=0}^{N} \vec{X_i}$
9: $\theta = \arctan \frac{\vec{S}}{\|\vec{S}\|}$
10: $\bar{X_n} = X_c + R * [cos(\theta)\ sin(\theta)]^T$
11: **for** points on the circle Y_i **do**
12: $X_n \ni C\ is\ minimum$
13: $C = d(\bar{X_n}, Y_i) + (d(X_i, Yj) - d_{min})$
14: $d(X, Y)$ is the euclidean distance between X and Y
15: d_{min} = minimum euclidean distance X_i
16: **end for**
17: $\hat{X_n}$ = -$\hat{S}$

A. *Social Inference*

According to the targeted social scenario, the robot must identify the person calling the user. With consensus from the user, the robot estimates the socially acceptable goal position based on the group of people([Fig. 2]).

1) Calling person identification:: Speech recognition module based on DeepSpeech [12] initially identifies the name of the user, when called. The microphone array in the robot detects the direction of the voice. Based on the received information, the robot turns its head to the detected direction and look for hand raised gesture. Human detection is performed using the Yolov3 model [13]. Human skeletal detection using Google media-pipe [14] is used to identify the person with the raised hands(indicating calling gesture).

2) Group Aware Pose Estimation(GAPE): : After identification of the person calling the user, the user is notified through audio. Once the user confirms to go towards the person, the next stage of processing starts. The group of people containing the target person needs to be differentiated from the scene. The spatial information and orientation between different humans are used to obtain a social graph with the target person. Grouping is done using a probabilistic Support Vector Machine (SVM) classifier trained on coherent motion and pose indicators as in [15]. Position, where the user can join, is determined based on the group identified to contain the target person. Humans, when interacting in a group, arrange themselves symmetrically to face each other effectively. This symmetry is assumed for the proposed algorithm. As a prepossessing step, the 3D coordinates of people in the scene are converted to a top-view 2D frame (as shown in Fig. 3). Our region of interest is only the group containing the target person. Let x,y,z be the coordinates of a human in the group. According to the requirements, a scaling ratio k and translation vector T=$[a\ b]^T$ can be used for transformation as shown in (1).

$$\begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix} = \begin{bmatrix} k & 0 & 0 \\ 0 & 0 & k \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \quad (1)$$

The newly transformed coordinates X_i are used to fit a circle over the human group using the method of least square(lines 1-5 in Algorithm 1). A vector pointing from each human to the center of the circle is added up to form a resultant vector as shown in Fig. 3. This provides a measure of how the new person must be aligned to ensure symmetry in the group. Lines 8 to 10 calculates an initial estimate position on the circle using the direction of resultant vector. This is further refined by putting one more constraint. The person joining is expected to maintain a minimum distance d_{min} based on the proxemics [16] involved in social interaction. Lines 11 to 13 select the best position on the circle that satisfies the proximity constraint and the symmetry-based initially computed estimate. The final estimated position is

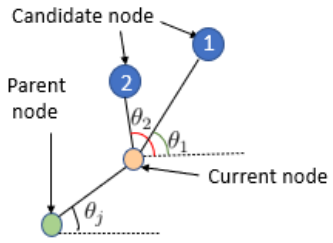


Fig. 4: Wiring step in RRT*.

X_n. The final orientation $\hat{X_n}$ is estimated to be in the opposite direction of the resultant vector to preserve the group symmetry.

B. Robot Navigation

1) Path planning: Path planning is done using a modified RRT* [17] on the occupancy grid map generated using RTAB SLAM. According to the addressed scenario, it is necessary to avoid sharp sudden turns such that the user can easily follow the robot. If the global path planner already considers it, there is less burden on the local planner. Therefore a cost function that accounts for the sudden change in angle is considered for path planning. The cost we used in the $ChooseParent$ and $Rewire$ step in RRT* is defined as:

$$Cost = w1 * dist(i,j) + w2 * (\theta_j \ - \ \theta_i) \tag{2}$$

Where, $dist(i,j)$ is the euclidean distance between current node and candidate node. θ_i for candidate node i represents the angle made by the line joining node and current node with the x-axis.(θ_1,θ_2 indicated in Fig. 4). Similarly, θ_j is the angle made by the line joining current node and the parent node with the x-axis. The term $\theta_j - \theta_i$ in cost function emphasizes wiring of nodes that align in the same direction. The weights $w1$ and $w2$ are finalized using Twiddle algorithm [18], from multiple experiments with different scenarios where the position of obstacles are varied. The planned path is further smoothed out using Cubic Splines.

2) Obstacle avoidance: Dynamic obstacle avoidance is not favored as the robot is guiding the user. Therefore slowing down and stopping in the vicinity of moving obstacles is implemented (Fig.5). The robot slows down based on the estimated obstacle position using a simple positional controller. For static obstacle avoidance a trajectory around the obstacle is planned. $P = [p_1, p_2, ...]$ be the list of poses in the trajectory.Given the start pose $p_s = [x_s, y_s, \theta_s]$ and the target pose in the global path after the obstacle be $p_g = [x_g, y_g, \theta_g]$. The trajectory planning can be formulated as a non-linear program :

$$\begin{aligned} &\min_{z \in P} f(z) \quad subject\ to \\ &p_1 = p_s, p_n = p_g \\ &g(z) = 0 \ \text{ linear constraints} \\ &k(z) = 0 \ \text{ kinematics} \end{aligned} \tag{3}$$

The kinematics for the differential drive robot is represented using $k(z) = 0$. Other constraints in velocity or rate of change in the heading can be represented using linear

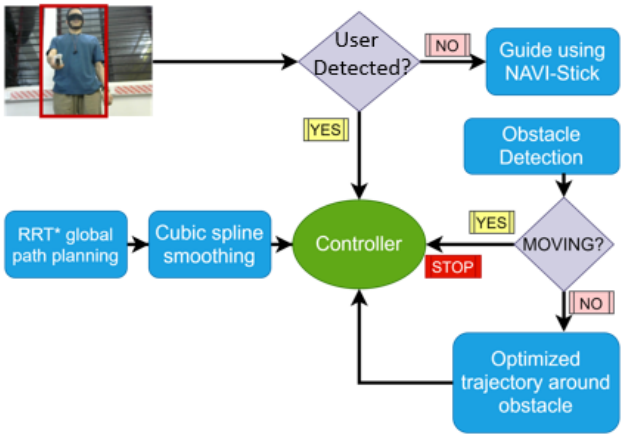


Fig. 5: The framework used for navigation of guidance robot.

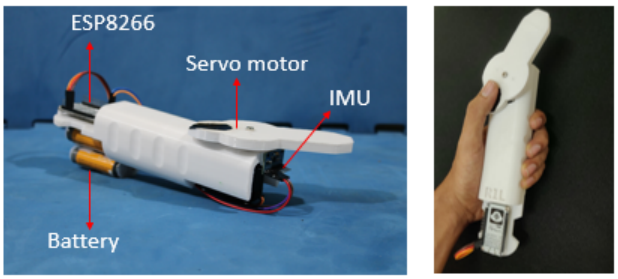


Fig. 6: NAVI-Stick used for navigational assistance.

constraints $g(z)$. The objective function that is minimized takes care of avoiding the obstacle based on the constraints set up for the problem. Obstacle is represented using circle in 2D by center (x_c, y_c) and Radius R.The objective function $f(z)$ is defined as:

$$f(x,y) = (\frac{\partial O}{\partial x}dx + \frac{\partial O}{\partial y}dy)^2 \tag{4}$$

Where $O(x,y)$ describes the obstacles in 2D.The constraint that is necessary to produce an admissible optimal path for the targeted scenario is the rate of change in the heading. It is restricted to an upper threshold so that the robot does not take quick and sharp turns.

C. Guidance

1) Controller: The differential drive model [19] is used for designing the controller. Apart from path tracking, controller also tries to maintain an optimum distance with the user following the robot. An optimum distance is fixed at d_{optim}. The real-time distance between robot and user is received from depth camera at the back of robot. Using the error in heading e_θ and error in distance to user e_d linear velocity and angular velocity v and w are calculated as:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} v_{desired} - K_{p1}e_\theta + K_{p2}e_d + K_{d1}\dot{e_\theta} K_{d1}\dot{e_d} \\ K_{p3}e_\theta + K_{d3}\dot{e_\theta} + K_i \int e_\theta \end{bmatrix} \tag{5}$$

The gains K_{p1},K_{p2} , K_{p3} , K_{d1} , K_{d2} K_{d3} are calculated based on experiments. The Ziegler Nichols tuning method [20] was used to come up with initial values. The terms e_d

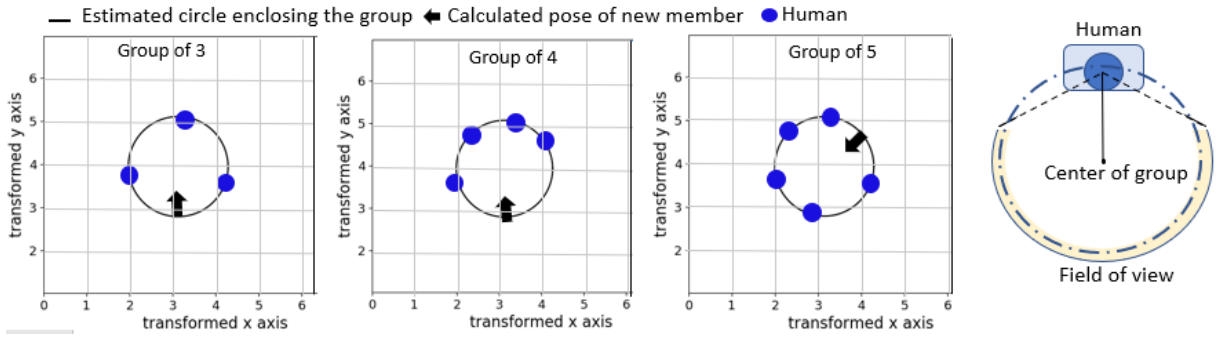


Fig. 7: Results of GAPE algorithm with groups of people in different orientations and numbers [Top View].

and $\dot{e_d}$ helps in controlling the velocity of the robot based on the error in maintaining optimum distance with the user.

2) NAVI-Stick: The user is guided along the path by providing multi-modal feedback. In addition to the voice input, the user receives a better sense of direction using NAVI-Stick shown in Fig. 6. The device has a direction pointer in the front that indicates the direction to move. It has finger shaped concavity where the user can place his finger for a more intuitive sense of direction as the pointer rotates. The calculation of direction also incorporates the orientation in which the device is held by the user using an IMU(MPU9250). It is interfaced with Esp8266 NodeMCU which receives the data sent through WiFi and rotates the pointer using a servo motor. As depicted in Fig. 5, the user is tracked using a camera at the back of the robot. If the user is not detected, stop signal is sent to the controller. The robot searches for the user by changing its heading and once found, a path is planned from the user's current position to the robot. The depth camera is used to compute the euclidean distance of user from the Robot. Thus the user is localized on the map based on which the path is planned. The user is then guided to the robot using NAVI-Stick.

III. EXPERIMENTS AND RESULTS

The experiments were conducted using the MISTY II Robot. It runs on Snapdragon 410 and Snapdragon 820 mobile processors, Windows IoT Core, and Android 8. Jetson Nano was used for parallel processing.Occipital Structure Core depth sensor in MISTY II enables SLAM capabilities. Intel Realsense camera D435i is mounted at the back of MISTY's head to monitor the visually impaired user following the robot. 3 far-field microphones using Qualcomm® FluenceTM PRO is used for sound localization. The overall implementation was tested in Gazebo simulation [21] in an environment as shown in Fig. 1. Python and MATLAB were used to simulate and validate the individual components of the system. An indoor environment with humans and obstacles as described in Target Scenario was set up. Real-time experiments were conducted with a blindfolded person.

A. Group Aware Pose Estimation

The algorithm was validated by using groups of people with varying orientations and numbers. The first set of experiments evaluated the estimated pose by mere inspection. Fig. 7 shows the results obtained for experiments with people arranged in groups in different formations. The top-view of the scene and estimated joining position are displayed in the Figure with the transformed axes. A pose metric based on the Field Of View(FOV) (as indicated in Fig. 7) was used to evaluate the results.

GAPE algorithm		
Number of people in group	FOV metric	Distance to nearest member[m]
3	97.8	1.1
4	94.4	0.72
5	96.7	0.88
6	77	0.34
7	75.3	0.49

TABLE I: GAPE algorithm analysis.

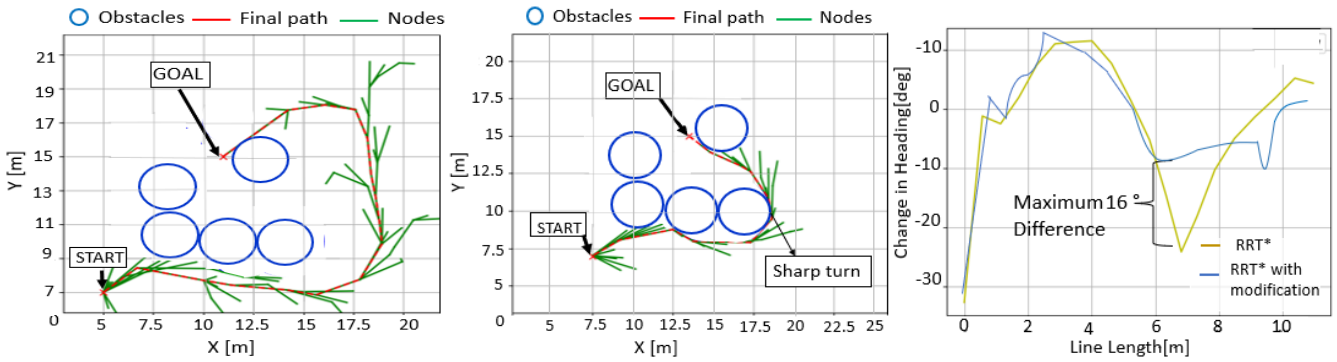
$$F = \frac{\sum_{i=0}^{N} b_i}{N} \tag{6}$$

$$b_i = \begin{cases} 1, & \text{if estimated position is in the FOV of member i} \\ 0, & \text{otherwise} \end{cases}$$

Given the number of people in the group is N, a metric F defined in (6) was used. bi is a binary number, which is 1 if the estimated pose for the new person falls in the FOV of the person i. The visual field of the human eye spans approximately 120° of arc**??**. Thus an arc of 120° is used for checking if the estimated pose is in the FOV of each group member. The results were also evaluated based on the minimum distance the new person maintains with a group member. A distance in the range of 0.5-2.5 m is considered based on proxemics. The results obtained in Table (1) show that the algorithm does well when the number of people in the group is less than 6. The proximity also decreases as the number of people in the group increases. Real-time experiments also reveal that the algorithm doesn't perform well when people in the group are standing in a linear fashion.

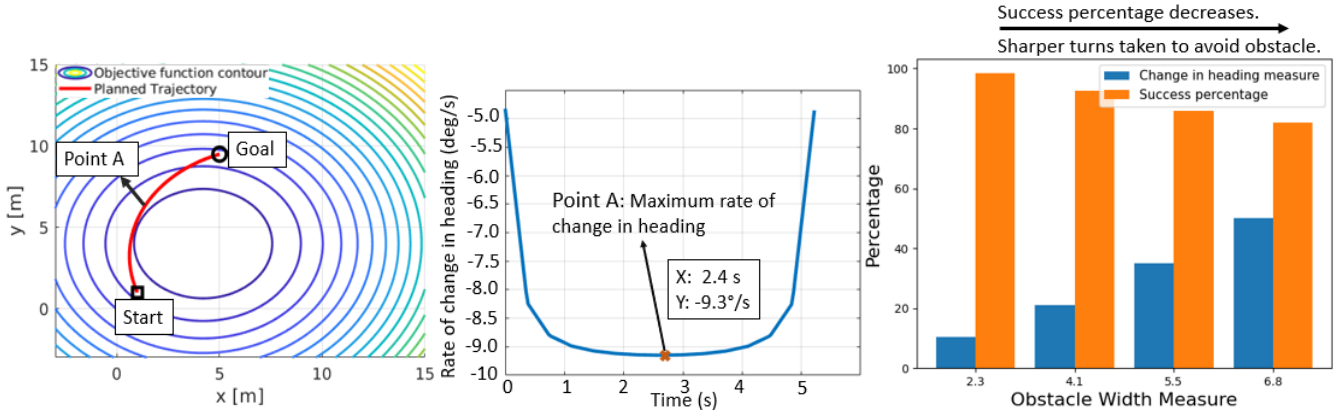
B. Robot Navigation

The global path planning was tested in various scenes with obstacles(humans) represented using circles of diameter



(a) Planned path using Modified RRT*. (b) Planned path using RRT* (c) Change in heading in final path, compared between RRT* and modified RRT*.

Fig. 8: Global path planner



(a) Trajectory planned around obstacle (b) Rate of change in heading vs time (c) Analysis

Fig. 9: Optimal static obstacle avoidance

2.7m placed at different positions. The modification in cost function in RRT* is evaluated based on the change in angle from one node to another.Maximum length of a node is set to be 1.2m. Path planning based on both methods is performed multiple times in the same scene and averaged to compare results. For the scenario in Fig. 8, the change in heading is compared between the two algorithms in Fig. 8c. The RRT* has a steep turn as indicated in Fig. 8b which corresponds to the maximum difference of change in heading for the two planners(Fig. 8c). The path produced by RRT* is optimal based on distance. The modified planner tries to go around the object and results in a longer path when compared to RRT* but results in less sharp turns as shown in Fig. 8(a) and 8(b).Fig. 9a shows the static obstacle avoidance in action. The magnitude of rate of change in heading was constrained to be less than 12 degrees/s. Optimal trajectory to the goal pose is planned without colliding with the static obstacle using Eq. 3. The Fig. 9b shows the rate of change of heading angle of robot with time. Point A in the Fig. 9a has a maximum change in heading in a sec. The same point shown in Fig. 9b validates that the maximum rate of change of heading is within the defined constraint. The static obstacle avoidance was tested in different scenarios. The velocity is fixed at 0.25m/s. Objects with varying radius are used as obstacles and the start point was fixed. Obstacle width measure O defined as:

$$O = \frac{w_o}{w_r} \tag{7}$$

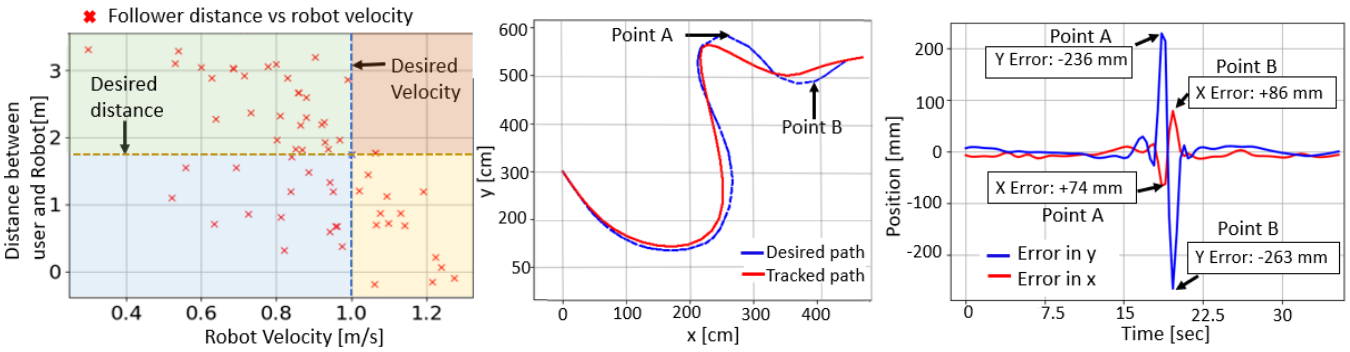
where w_o width of an obstacle and $w_r$width of the robot is used to analyze the success percentage (Fig. 9c). The obstacle avoidance has success percentage of 98.4% when the obstacle width measure is 2.3. The percentage measure of maximum heading change is defined as:

$$h = \frac{c}{90}\% \tag{8}$$

The maximum change in heading in 10cm of path, c is normalized by a fixed 90° change in heading for the same distance to get a percentage measure. The results highlight that as the obstacle size increase the avoidance of the obstacle by satisfying the constraints become difficult.

C. Controller

The controller used is analyzed in Fig. 10a. The variation of velocity of MISTY based on the distance to the following user over the trial is illustrated in Fig. 10a. The points in the green region show that the velocity is decreasing when



(a) Variation of robot velocity with distance to following user.

(b) Path tracking trial performed with robot for user guidance.

(c) Error in x and y when path tracking.

Fig. 10: Results of the designed controller for a trial.

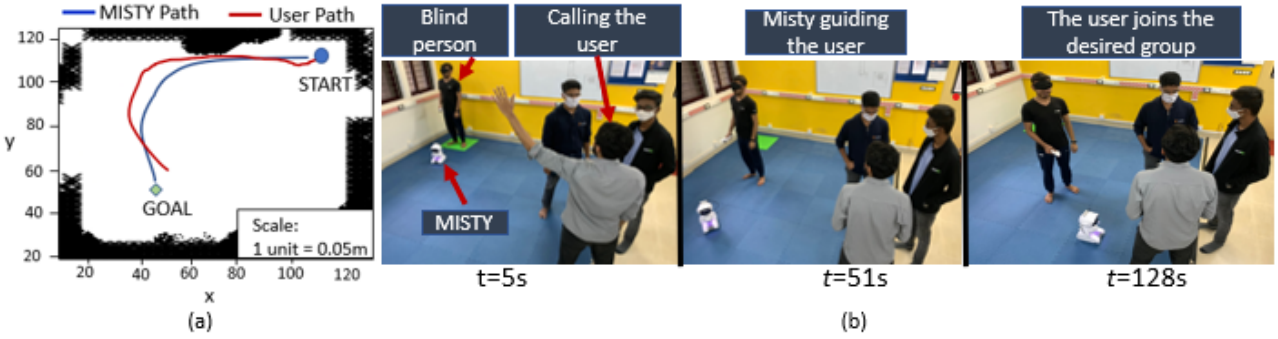


Fig. 11: (a) Path followed by MISTY and the user in an indoor trial depicted on the grid map. (b)Snapshot of the indoor trial. MISTY identifies the person calling the user, estimates the destination and guides the user towards the goal.

the distance to the user increases. As the distance to the user decreases less than the desired distance, the velocity increases as shown by points in yellow region. There are few cases where velocity decreases even though the distance is less than the desired value. These occur because the velocity also depend upon error in heading Eq. 5, i.e the velocity decreases when there is a sharp turn in the trajectory. The absence of points in the red region highlights that the controller reduces velocity as distance is more and thus effectively tries to maintain an optimum distance of 1.75m. The path tracking and the positional error of the controller are shown in Fig. 10b and Fig. 10c respectively. The maximum error in y-direction on the map occurs at point A and the maximum error in x occurs at point B. Given the current position of the robot X=$[x\ y]^T$ and desired position Xd = $[x_d\ y_d]^T$, the positional error based on euclidean distance over N time steps of one trial can be defined as:

$$e_p = \frac{\sum_{j=0}^{N} \sqrt{(x - x_d)^2 + (y - y_d)^2}}{N} \tag{9}$$

e_p computed for each trial, averaged over 15 trials is 0.0943m. The complete experiment of the system was carried out in a social indoor environment with a blindfolded person (Fig. 11(b)). The Navi-stick was used along with the audio input from MISTY by the user to navigate. The path followed by the person is plotted against the desired trajectory in Figure 11(a). The goal reached by user has a Euclidean distance error of 0.96 m with respect to the desired goal.

IV. CONCLUSION

The work implemented a guidance social robot for intelligent navigation with valuable social inference skills for the visually impaired. The GAPE algorithm works effectively in real time when social groups contain less than 6 people. Based on the defined pose metric, groups with 6 people show a performance of 77% and average proximity of 0.34m with the nearest person. This algorithm works better in cases where there is symmetry in the human group. The path planning using modified RRT* minimize the change in heading in the path planned. Proposed static obstacle avoidance works with a 92.6% success rate when the obstacle width measure is 4.1. The guidance framework employed used Navi-Stick and audio feedback from robot to guide the user. The follower aware controller, has an average positional error of 0.0943m from 15 trials. Based on the conducted trials, user is able to follow the robot using NAVI-Stick and the maximum distance error between user's final position and goal position is 0.98m.

REFERENCES

[1] "Vision impairment and blindness," World Health Organization, 26-Feb-2021. [Online]. Available: https://www.who.int/news-

room/fact-sheets/detail/blindness-and-visual-impairment. [Accessed: 12-Sep-2021].
[2] S.E. Hassan, J.E. Lovie-Kitchin and R.L. Woods Optom Vis Sci, 79 (Nov 2002), pp. 697-707
[3] C. Ye, S. Hong, et al., “Co-robotic cane: A new robotic navigation aid for the visually impaired,” IEEE Trans. Syst., Man, Cybern., vol. 2(2), pp. 33–42, 2016.
[4] A. K. Srinivasan, S. Sridharan and R. Sridhar, ”Object Localization and Navigation Assistant for the Visually challenged,” 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 324-328, doi: 10.1109/ICCMC48092.2020.ICCMC-00061.
[5] S. Krishnakumar, B. Mridha, M. J. N. Naves and K. Kowsalya, ”Intelligent walker with obstacle detection technology for visually challenged people,” 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017, pp. 2863-2867, doi: 10.1109/ICPCSI.2017.8392244.//
[6] Palopoli, L., Argyros, A., Birchbauer, J., Colombo, A., Fontanelli, D., Legay, A., ... Sedwards, S. (2015). Navigation assistance and guidance of older adults across complex public spaces: the DALi approach. Intelligent Service Robotics, 8(2), 77-92.
[7] Xiao, A., Tong, W., Yang, L., Zeng, J., Li, Z., Sreenath, K. (2021). Robotic Guide Dog: Leading a Human with Leash-Guided Hybrid Physical Interaction. arXiv preprint arXiv:2103.14300.
[8] Vega, A., Manso, L. J., Macharet, D. G., Bustos, P., Núñez, P. (2019). Socially aware robot navigation system in human-populated and interactive environments based on an adaptive spatial density function and space affordances. Pattern Recognition Letters, 118, 72-84.
[9] Henderson, M., Ngo, T. D. (2021, August). RRT-SMP: Socially-encoded Motion Primitives for Sampling-based Path Planning. In 2021 30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN) (pp. 330-336). IEEE.
[10] Ferrer, G., Sanfeliu, A. (2019). Anticipative kinodynamic planning: multi-objective robot navigation in urban and dynamic environments. Autonomous Robots, 43(6), 1473-1488.
[11] Li, Z., Hollis, R. (2019, November). Toward a ballbot for physically leading people: A human-centered approach. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 4827-4833). IEEE.
[12] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, et al., ”Deep Speech: Scaling up end-to-end speech recognition”, Http://arxiv.org/abs/1412.5567, 2014.
[13] Redmon, J., Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
[14] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... Grundmann, M. (2019). Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172.
[15] T. Linder and K. O. Arras, ”Multi-model hypothesis tracking of groups of people in RGB-D data,” 17th International Conference on Information Fusion (FUSION), 2014, pp. 1-7.
[16] Hall, E. T. (1966). The hidden dimension (Vol. 609). Garden City, NY: Doubleday.
[17] S. Karaman and E. Frazzoli, ”Sampling-based algorithms for optimal motion planning”, Int. J. Robot. Res., vol. 30, pp. 846-894, 2011.
[18] The Twiddle algorithm is attributed here to Sebastian Thrun, in lectures through the Udacity programme, [Online], Available: https://www.youtube.com/watch?v=2uQ2BSzDvXs
[19] Dhaouadi, R., Hatab, A. A. (2013). Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework. Advances in Robotics Automation, 2(2), 1-7.
[20] Hang, C. C., Åström, K. J., Ho, W. K. (1991, March). Refinements of the Ziegler–Nichols tuning formula. In IEE Proceedings D (Control Theory and Applications) (Vol. 138, No. 2, pp. 111-118). IET Digital Library.
[21] N. Koenig and A. Howard, ”Design and use paradigms for Gazebo, an open-source multi-robot simulator,” 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), 2004, pp. 2149-2154 vol.3, doi: 10.1109/IROS.2004.1389727.