

Watch out! There may be a Human.

Addressing Invisible Humans in Social Navigation

Phani Teja Singamaneni¹, Anthony Favier^{1,2}, Rachid Alami^{1,2}

Abstract—Current approaches in human-aware or social robot navigation address the humans that are visible to the robot. However, it is also important to address the possible emergences of humans to avoid shocks or surprises to humans and erratic behavior of the robot planner. In this paper, we propose a novel approach to detect and address these human emergences called ‘invisible humans’. We determine the places from which a human, currently not visible to the robot, can appear suddenly and then adapt the path and speed of the robot with the anticipation of potential collisions. This is done while still considering and adapting humans present in the robot’s field of view. We also show how this detection can be exploited to identify and address the doorways or narrow passages. Finally, the effectiveness of the proposed methodology is shown through several simulated and real-world experiments.

I. INTRODUCTION

Human-aware or social robot navigation is rapidly advancing, and new frameworks [1], [2], [3], [4] are required to address the intricate social navigation scenarios. However, most of these frameworks [5], [6] address only the visible humans and do not take into account the possible emergence of humans that are not visible currently. We believe that such invisible humans should be considered while developing a human-aware navigation framework to avoid any erratic behaviors of the robot planner when a human suddenly appears. Therefore in this work, we try to address these invisible humans in social navigation settings.

There are not many works that address this problem in the field of human-aware navigation. However, there are some existing works in classical robot navigation that address similar issues. Particularly, this work is inspired by the pioneering work of M. Krishna concerning the ability of a mobile robot, based on the model of its perception functions, to assess from where in the close environment of the robot a human can emerge and prepare to react to ensure no-collision by adapting its path and velocity [7], [8], [9]. Some recent works like [10], [11] address the issues of robot navigation in occluded or unknown regions with a limited field of view. The work presented in [12] talks about the adaptive speed control of the robot in unknown environments and also talks about the occluded regions. The authors of [13] propose a methodology to mitigate or avoid collisions while navigating. In our case, we are trying to mitigate possible future collisions with a human. The motion planning problem in the presence of dynamic

obstacles can be solved in several ways. Some works use grid-based planning [1], [14] while some others [15], [16] use optimizations like model predictive control. We use sparse graph optimization proposed in [17] and add human-aware constraints as proposed in our previous works [18], [19], [20].

As it is evident that the unknown or occluded region could cause issues with classical navigation, the same applies to social navigation. Hence, we propose the concept of ‘invisible humans’ to human-aware navigation planning through this paper. As per the knowledge of the authors, there is no other work that addresses this problem in human-aware navigation. The major contributions of this paper are as follows. Firstly, we formulate the problem of invisible humans detection and propose an algorithm to determine the locations from which humans who are not currently visible can emerge suddenly. These invisible humans are then integrated into our human-aware navigation framework, CoHAN [20], by introducing a new human-aware constraint into our optimization scheme. This constraint modifies the path and speed of the robot taking into account the anticipation of potential human appearances to avoid collisions and surprises. We further show how the detected invisible humans can be exploited to identify some interesting scenarios (like doors/passages) and address these by adding new modalities to CoHAN. The implementation and code can be found at https://github.com/sphanit/cohan_planner_multi/tree/model.

The rest of the paper is organized as follows. Section II presents the formulation and an algorithm to detect invisible humans. Section III shows how the invisible humans are integrated into CoHAN and talks about the issues that arise. It also presents a simple formulation to identify narrow passages. In Section IV, various experiments to evaluate the proposed approach are presented, followed by the real-world experiments in Section V and a discussion on the limitations in Section VI. Finally, Section VII concludes the work.

II. INVISIBLE HUMANS DETECTION

The invisible humans are detected using an emulated laser scan on a 2D map in ROS. A custom laser scan is attached to the robot’s base and it is continuously updated as the robot moves on a given map. The entire system is implemented in ROS [21] and requires the map that is published by the ROS Navigation stack. In order to avoid too many detections, we limit the invisible humans detection to a radius of 5 m in front of the robot. The detection of invisible humans is a two-step process involving corner detection and locating invisible humans. Each step is explained in detail below.

¹Authors are with LAAS-CNRS, Universite de Toulouse, CNRS, Toulouse, France, {ptsingaman, anthony.favier, rachid.alami}@laas.fr

²This work has been partially funded by the Agence Nationale de la Recherche through the ANITI ANR-19-PI3A-0004 grant

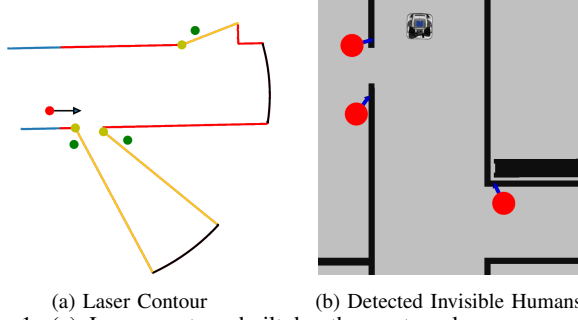


Fig. 1: (a) Laser contour built by the custom laser sensor. The red and blue lines are the actual walls or obstacles in the front and back of the robot respectively. The black lines are the laser range boundaries and the yellow lines are interpolated lines between two gaps of laser data. The robot is shown as a red dot with an arrow. The detected corners are shown in yellow, while the detected invisible humans are shown in green. (b) The detected invisible humans on the map for the contour shown. The red circles shows the location and the blue arrow shows the assumed direction, which is always oriented towards the robot.

A. Laser Contour and Corner Detection

A custom laser scan sensor that is attached to the robot's base, scans the given 2D map to get the visible contour of the map. The laser data consists of a list of values showing the scan ranges in the field of vision of the sensor. This custom sensor data is used in the place of the real laser scan data to ensure uniformity across different robots and sensors. An example laser contour built using this is shown in Fig. 1. Different parts of these contour lines are shown in different colors for ease of explanation. The red and the blue lines together constitute the regions on the real map where the laser has hit a wall or an obstacle. The black lines represent the laser data that did not hit anything and reached the end of their range (in our case, range of the laser is 7 m). Lastly, the yellow lines are interpolated rays joining large separations between consecutive laser values and play a major role in our algorithm. The red circle and the arrow represent the robot's position and its direction, while the green circles represent the estimated invisible humans. The corner detection is relatively easy once the laser contour is available. Firstly, all the pairs of consecutive laser ranges that are separated by more than 0.5 m are determined and stored in a set $\{V\}$. The threshold of 0.5 m is chosen to filter out small gaps from where a human will not emerge. After this, the values that are closest to the robot in each of the above pairs are identified to be corners and stored in a set $\{c\}$. These are shown as yellow circles in Fig. 1.

B. Estimation of Invisible Humans

The estimation of possible locations for the invisible humans is not very straightforward. The laser contour forms a complex non-convex polygon, and we are searching for circles whose centers are outside this polygon and do not intersect the contour. We solve this problem using a combination of ray tracing and vector algebra. Consider a non-convex polygon as shown in Fig. 2. The vertices are numbered in the anti-clockwise direction. Consider a point P_1 that lies

Algorithm 1 Locate Invisible Humans

```

1: Determine the vertex pairs set  $\{V\}$  using laser contour
2: Determine the corners set  $\{c\}$  from  $\{V\}$ 
3: for each  $c$  do
4:    $V_1 = c = (x_1, y_1)$ 
5:    $V_2 = (x_2, y_2)$   $\triangleright$  Corresponding pair from  $\{V\}$ 
6:    $u_x = \frac{(x_2 - x_1)}{\|V_1 V_2\|}$ ,  $u_y = \frac{(y_2 - y_1)}{\|V_1 V_2\|}$ 
7:   Set  $P = (x_p, y_p) = (x_1, y_1)$ 
8:   while True do
9:     Calculate  $H_{inv}$  using Eq. (4)
10:    Calculate  $\vec{r}$  and  $\beta$ 
11:    if  $\|\vec{r}\| < LaserData(\beta)$  then
12:       $x_p = x_p + \alpha u_x$ 
13:       $y_p = y_p + \alpha u_y$ 
14:      continue
15:    end if
16:    Check for overlap on the Map
17:    if no overlap then
18:       $advance = False$ 
19:      for  $i = 1$  to  $k$  do
20:         $d = \frac{i}{k} * (h_{rad} + \epsilon)$ 
21:        Calculate  $P_r$  and  $P_l$  using Eqs. (4), (5)
22:        Check  $P_r$ ,  $P_l$  for the overlap on Map
23:        if no overlap then
24:          continue
25:        else if overlap then
26:           $advance = True$ 
27:          break
28:        end if
29:      end for
30:    end if
31:    if  $advance == True$  then
32:       $x_p = x_p + \alpha u_x$ 
33:       $y_p = y_p + \alpha u_y$ 
34:    else if  $advance == False$  then
35:      break
36:    end if
37:  end while
38:  Add  $H_{inv}$  to the set of invisible humans,  $\{H_{inv}\}$ 
39: end for
40: return  $\{H_{inv}\}$ 

```

between the vertices V_1 and V_2 . If P_1 is outside the polygon it should lie to the right of the vector $\vec{V_1 V_2}$. Similarly, a point P_2 lying outside the polygon between V_2 and V_3 lies to the right of $\vec{V_2 V_3}$ and so on. It holds true irrespective of the number of sides of the polygon. We exploit this property to determine the positions of the invisible humans. Note that, a point lying outside the polygon will always be on the right side of the vectors, but not every point on the right of the vectors lies outside the polygon. This is because this methodology uses only a single side and does not consider the other sides. To handle this, we use the fact that the polygon in our case is determined by the laser contour and any point outside this polygon is not visible to the laser.

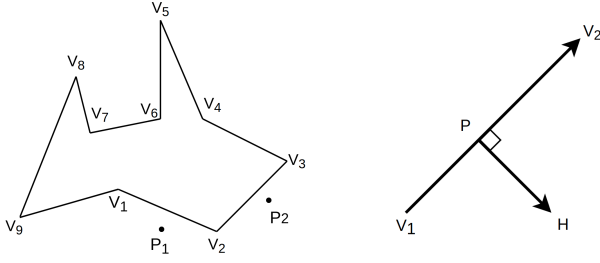


Fig. 2: Non-convex polygon and vector formulation to determine invisible humans. We try to find a point H that lies between two vertices V_1 and V_2 and lies to the right of $\vec{V_1V_2}$. Note that the perpendicular distance of this point should be greater than the assumed human radius.

In Fig. 1, the yellow lines correspond to the edges of interest in the polygon. The numbering of the vertices is determined based on the indices of the laser scan data which (the rays) move from right to left in the anti-clockwise direction. In order to determine a possible invisible human, we need to determine a point H that is to the right of $\vec{V_1V_2}$ and whose perpendicular distance is greater than an average human radius, h_{rad} as shown in Fig. 2. Consider a point P that lies on line segment $\vec{V_1V_2}$ such that \vec{PH} is perpendicular to $\vec{V_1V_2}$. If we know the point P , then H can be determined using the following equations:

$$\text{sign}(\vec{V_1V_2} \times \vec{PH}) = -1 \quad (1)$$

$$\vec{V_1V_2} \cdot \vec{PH} = 0 \quad (2)$$

$$\|\vec{PH}\| = h_{rad} + \epsilon \quad (3)$$

where (\times) is the cross product, (\cdot) is the dot product and $(\|\cdot\|)$ is the euclidean norm respectively. ϵ is the off-set on the human radius to avoid unrealistic detections. In this work, we chose ϵ to be $\frac{h_{rad}}{2}$. Let $V_1 = (x_1, y_1)$, $V_2 = (x_2, y_2)$ and $P = (x_p, y_p)$, then by solving the above equations, $H = (x_h, y_h)$ is given by:

$$\begin{aligned} x_h &= x_p + \frac{d(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\ y_h &= y_p - \frac{d(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \end{aligned} \quad (4)$$

where $d = (h_{rad} + \epsilon)$. Similarly, the point on the left can be obtained by reversing the sign in Eq. (1), which yields:

$$\begin{aligned} x_h &= x_p - \frac{d(y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\ y_h &= y_p + \frac{d(x_2 - x_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \end{aligned} \quad (5)$$

From the Eq. (4), we can see that (x_p, y_p) is required to determine (x_h, y_h) and it is still unknown as we cannot solve for four variables using only three equations (Eq. (1)-(3)). As it is already known that P lies on the line segment joining $\vec{V_1V_2}$, it can be determined by performing a search on this line segment, starting at one end and moving towards the other in small increments. The set of detected corners,

$\{c\}$ are taken as the starting points of this search. In each iteration, a possible invisible human position is estimated using Eq. (4) and then projected onto the map to see if there is any overlap with an obstacle or wall. As mentioned before, we need another check to ensure that the point is outside the laser contour. Suppose, the vector joining the robot and the point H is \vec{r} and it subtends an angle β with the positive x -axis of the base frame of the robot. As the custom laser is also attached to the base frame of the robot, there should be laser scan data corresponding to this angle β . Hence, when the H is outside, the following condition is satisfied:

$$\|\vec{r}\| > \rho(\beta) \quad (6)$$

where $\beta = \text{atan2}(x_h - x_{rb}, y_h - y_{rb})$, (x_{rb}, y_{rb}) is the robot base frame's position and $\rho(\beta)$ is the laser scan reading at angle β . To refine this search further, two points, one on left, P_l , and the other on the right, P_r , of the H are considered with incremental distances until $h_{rad} + \epsilon$ and checked for overlap using the map.

The entire procedure is shown below in Algorithm 1 where u_x and u_y (line 6) are the unit vectors along the direction of $\vec{V_1V_2}$ and α is a scalar determining (lines 26, 27) the step size or increment. The invisible humans detected using the above-mentioned algorithm are shown in Fig. 1 (b). The red circles are the detected location while the blue arrows show the direction. We assume that the humans are always coming towards the robot and hence, the direction is always oriented towards the robot. In the next section, we explain how this is integrated into the human-aware planning framework for social robot navigation.

III. INTEGRATION WITH A HUMAN-AWARE PLANNER

In the previous version of CoHAN, we address different types of visible humans by introducing new modalities and human-aware constraints [20]. In this work, we extend it further to address the invisible humans. The invisible humans are detected as explained above and then they are published on a ROS topic. CoHAN subscribes to this topic and adds a new constraint to its optimization that is specifically designed for invisible humans. Further, using these invisible human detections, we propose a methodology to identify doors and narrow passages. For more details on the modality switching and constraint implementation readers are advised to refer to our previous works [18], [19], [20].

A. Invisible Humans Constraint

The invisible humans constraint takes into account the human reaction time, walking speed, and deceleration and aims to make the robot cautious about the sudden human emergence. The cost added by this constraint for the n^{th} pose of the robot's trajectory is given as:

$$\begin{aligned} \text{cost}_{inv_human} &= \max\left(\frac{V - a\Delta t_n}{d}, 0\right) \quad \text{if } \Delta t_n > 0.5s \\ &= \frac{V}{d} \quad \text{otherwise} \end{aligned} \quad (7)$$

where d is the distance between the invisible human and the robot, V is the average human walking speed, 1.3 m/s [22], a is the deceleration of the human and Δt_n is the time difference between the n^{th} pose and the starting pose of the planned trajectory of the robot. The value of the deceleration, a , can vary and can be up to a maximum of 2.94 m/s² (0.3 g) [23]. In this work we take a reaction time of 0.5 s as discussed in [23], [24]. Hence the constraint adds the maximum possible cost until 0.5 s. Then we assume that the human will continuously decelerate to avoid collision with the robot over time and eventually stops, which is reflected in the upper part of Eq. (7). The time (Δt) and human detections are reset after every control cycle.

1) *Issue with the constraint:* The main objective of the constraint is to push the robot away from the opening, anticipating the emergence of invisible humans. However, when the robot needs to pass through this opening and if the passage is narrow (door or narrow corridor), the constraint pushes the robot away and makes it impossible to enter the passage. To mitigate this, we devise a simple formulation that detects such scenarios. Once a narrow passage is detected, the invisible humans constraint is switched off, and the maximum robot's velocity is reduced until it passes through. The passage detection process is explained in detail below.

B. Doorway or Narrow Passage Detection

The detection of narrow passages or doorways not only allows us to overcome the issue of the invisible humans constraint but also allows us to define a new modality of planning that needs to be handled separately. In this work, we try to address three different scenarios, as shown in Fig. 3. The first scenario, Fig. 3 (a), occurs in the case of a doorway or the openings and closings of a narrow passage. In such scenarios, the invisible humans exert equal forces from two different directions, which align the robot at the center of the passage, but it cannot pass through until the invisible constraint is switched off. However, the threat of invisible humans still exists, therefore, we make the robot act cautiously and move it with a lower velocity (≤ 0.3 m/s) until it passes through the passage. To detect this scenario, we use the positions of the invisible humans and the robot to check whether an *isosceles triangle* is formed with the three vertices. The robot lies on the vertex, which connects the approximately equal sides, and humans are present at the base vertices. In order to limit false detections, we set some numerical limits on the lengths of the equal sides and the base. Assuming that a human has 0.3 m radius and the robot has 0.5 m, the length of the base should be ≥ 1.6 m. When the clearance from obstacles or walls is taken into account it increases further. In this work, we set the limit on base length as 3 m. Similarly, for the equal sides, there should be a minimum length of 0.8 m, and we chose the limit to be 2 m. These values are chosen empirically based on the tests in several situations. If the above conditions are satisfied, a passage is detected, and CoHAN switches to a new modality called *Passing Through*, which sets the conditions mentioned above. The situation shown in Fig. 3 (c) is almost the same as

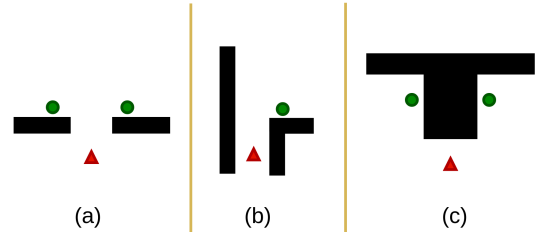


Fig. 3: Different types of passages that are detected using invisible humans. (a) Doorways or openings and endings of the corridors (b) a narrow passage with opening on one side and a wall on the other side (c) a large pillar or obstacle where robot cannot see on either side. The green circles are the possible locations of the invisible humans and the red triangle shows the robot pointed towards the direction of its motion.

the doorway and we differentiate these scenarios by reading the center value of the laser scan data. If the value in the data is less than the length of the perpendicular bisector of the triangle's base, then it is identified as a pillar or large obstacle approach. CoHAN identifies this as a new modality, but for now, we handle it the same we handle the doorway.

The situation shown in Fig. 3 (b) is different from the other two as the robot's passage is blocked by an invisible human on one side and obstacle clearance on the other side. As the robot, may or may not align in this case, it is handled differently. We check the angle of the laser scan corresponding to the detected corner and read the value of the data that is symmetrical to this angle along the direction of the robot. Finally, if the difference between the distance of this laser scan data and the invisible human from the robot's position is < 1 m, we identify it as a wall passage and set CoHAN to *Pass Through* mode. The threshold of 1 m is chosen empirically here.

IV. EXPERIMENTS

The proposed approach, after being completely integrated with CoHAN is tested in several settings. In this section, we show four interesting scenarios and present a detailed analysis. In all these experiments, we assume $h_{rad} = 0.3$ m and set $k = 10$ and $\alpha = 0.2$. We use ROS-melodic with Ubuntu 18.04, and all the scenarios are simulated using MORSE [25] simulator. The simulated human agents used in the experiments are controlled using InHuS [26], a human simulator developed in our lab.

A. The effect of the Invisible Humans constraint

To show the effect of introducing the invisible humans constraint into CoHAN, we present the robot with a door crossing scenario as shown in Fig. 4. We test scenarios without and with the invisible humans constraint and the corresponding paths of the robot are presented in Fig. 5 (a) and Fig. 5 (b) respectively. The paths are colored, and the color moves from blue to red as the robot moves from start to goal. It can be clearly seen from these paths that the inclusion of the constraint made the robot more cautious as it takes a larger distance and aligns its path earlier to pass through the doorway. The corresponding speed plots are shown in

Fig. 5 (c) and (d). Comparing the plot in Fig. 5 (d) with the profiles in Fig. 5 (c), it can be clearly seen that the robot slows down significantly before passing through the passage. The cautious behavior of the robot is again reflected in these speed profiles.

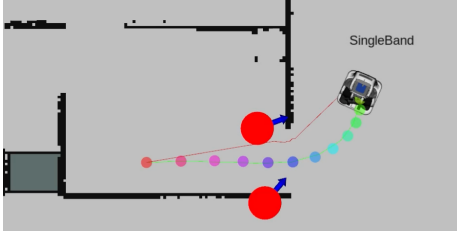


Fig. 4: The robot passing through the door under the presence of invisible humans. The colored circles represent the poses of the robot and different color corresponds to different time instance.

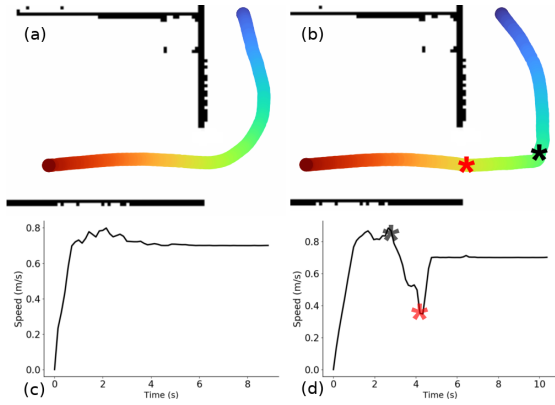


Fig. 5: Paths and speed profiles of the robot passing the door without (a, c) and with (b, d) the Invisible Humans constraint. The color of the paths indicates the time and progress of the robot, from blue to red (start to goal). In (a) the robot crosses the door “full speed”. In (b) it decelerates before entering the door (black star) and has the lowest speed at the entrance to the door (red star) around 4.2s corresponding to the shortest distance to invisible humans.

B. Navigation in the presence of visible and invisible humans

The inclusion of the invisible humans into human-aware navigation planning should not cause discomfort to the visible humans that are moving around the robot. To show that CoHAN finds a fine balance between the invisible and visible humans, we present two corridor scenarios, one with many doors (or openings) as shown in Fig. 6 (a) and the other with pillars as shown in Fig. 6 (b). In both of these scenarios, the robot faces complex situations where it has to find a balance between the visible and the invisible humans.

In the case of the corridor with many openings, the robot anticipates that a human might emerge anytime and tries to move away from the openings. However, when it sees a human passing through the corridor, it tries to provide more space to the human by moving to one side. At the same instance, it faces the forces from the invisible humans and tries to find a balance between these and the visible human. By observing the path and speed profile of this scenario from Fig. 7 (a) and (c), we can see that the robot moves away as

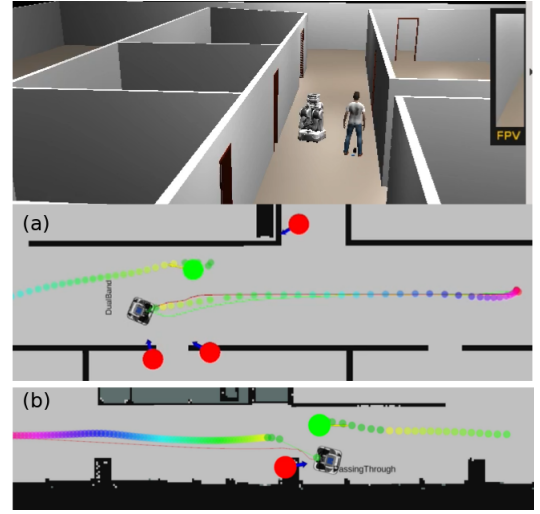


Fig. 6: Corridor scenarios used for testing CoHAN. (a) Corridor with many openings where the robot continuously anticipates the emergence of humans. (b) Corridor with pillars between passage that creates complex navigation scenarios. In both these settings, the robot tries to find a balance between visible and invisible humans. The green circle is the visible human interacting with the robot, while the red circles are estimated invisible humans. The colored path with circles is the planned trajectory of the robot.

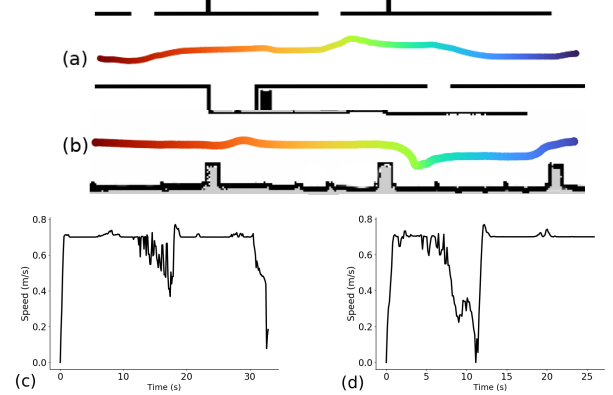


Fig. 7: Paths and speeds profiles of the robot in corridor scenarios. (a), (c) correspond to the corridor with many openings. (b), (d) correspond to the pillar corridor. The color of the paths indicates the time and progress of the robot, from blue (start) to red (goal).

well as reduces its speed rapidly to accommodate the visible human. Nonetheless, it does not move very close to the wall as it anticipates a human emergence.

In the corridor with pillars, the robot faces another complex situation where it has to pass through a very narrow opening and has to let the human pass through the same as shown in Fig. 6 (b). From the path and speed profiles of this scenario from Fig. 7 (b) and (d), we can see that the robot slows down rapidly while moving to a side and momentarily stops before moving forward again. Here, the robot stops and lets the visible human pass before it can continue its navigation. Further, it detects the narrow passage scenario discussed in Section III and changes to *Pass Through* mode. We can, therefore, infer that CoHAN always tries to find a fine balance between visible and invisible humans and can mitigate very complex situations.

C. Sudden emergence of a human

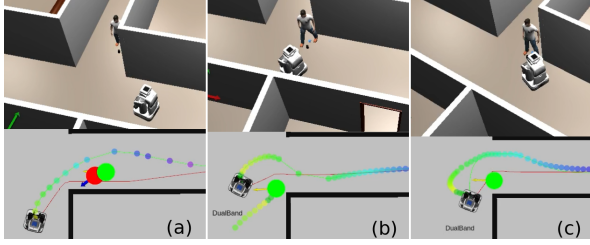


Fig. 8: Sudden emergence scenario. The colored paths with circles is the planned trajectory of the robot. (a) Shows the anticipated invisible human in red and the real human in green. The robot starts moving away from the corner. (b) The robot has seen the human and adjusted its trajectory to provide more space to the human. (c) The scenario without the invisible humans constraint. The robot moves very near to the wall blocks the human momentarily before adapting its path.

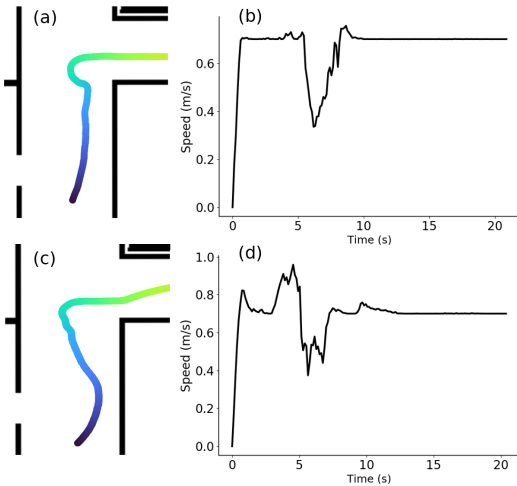


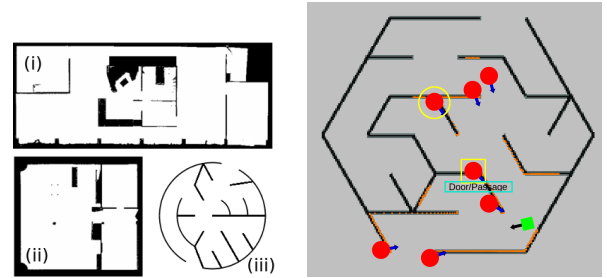
Fig. 9: Path and speed profiles of the sudden emergence scenario without (a, b) and with (c, d) the invisible humans constraint. The color of the path indicates the time and progress of the robot, from blue to green (start to goal).

The final scenario we discuss in this section shows a situation where a human emerges suddenly from an occluded region. The snapshots of this scenario before and after the emergence are shown in Fig. 8. The added invisible humans detection predicts a possible position of the human as shown in Fig. 8 (a), which approximately overlaps with the real human. The robot starts moving away from the wall slowly because of this anticipation, and suddenly a real human appears in front of it (Fig. 8 (b)). The robot quickly adapts its trajectory and moves away from the human, slowing down a little before continuing to its goal. However, without this detection, as shown in Fig. 8 (c), the robot moves close to the wall and blocks the human's way for a moment before adapting its path. The paths taken by the robot without and with the addition of invisible humans constraint to CoHAN are shown in Fig. 9 (a) and Fig. 9 (c) respectively. It is clear from these plots that the proposed constraint makes the robot move cautiously and lessens the surprise to humans. Further, the path of the robot is smoother in Fig. 9 (c) when compared

to the path in Fig. 9 (a) as there are no sudden path changes.

The speed profile in Fig. 9 (b) shows a sudden drop in the velocity of the robot. This occurs because CoHAN adapts its speed to prevent a possible collision and shock to the human. Then, the robot slowly moves away and plans a new path to the goal. The speed profile in Fig. 9 (d) is completely different compared to the last one. The robot starts drifting away from the wall (both x and y velocities) before seeing the human and this shows the increase in the velocity. When the human appears, it slows down and then changes its direction before continuing to the goal with almost a constant speed. This is the sharp change (slowdown) we see in Fig. 9 (d).

D. Quantitative Analysis



(a) Maps used for testing (i) LAAS (ii) Bremen (iii) Random Maze

(b) Failed detections

Fig. 10: (a) Different maps used for testing the robustness and accuracy of the approach. The map in (iii) shows an example of randomly generated maze. The other maps are standard ones. (b) Different types of failed detections. The detection in yellow circle is a false positive whereas that in yellow square is an overlap (possibly true). The passage detection is also wrong as the wall is detected as passage. The green cube with black arrow shows the robot and its direction. The red cylinders are the invisible humans.

For testing the proposed algorithm, we have designed randomized experiments. We either generate a random map using the Maze Generator¹ or randomize the position of the robot in the known map. The maps used for these experiments are shown in Fig. 10 (a). The LAAS and Bremen maps are collected from the models of the real spaces. Fig. 10 (a) (iii) shows a random map generated using the Maze Generator. Fig. 10 (b) shows some failed detections using the proposed algorithm that are taken into account while calculating the accuracy. The red circles with blue arrows are the predicted invisible humans. The invisible human in the yellow circle is classified as a false positive as no human could be located inside the wall. The detection in the yellow square is similar, but it is not completely inside the wall. We call this case an 'overlap' and classify this also as a false positive. The door/passage detected (cyan rectangle) in this picture is wrong, and we classify this as a false positive while calculating accuracy for passage detection. The green square with black arrow is the robot in the figure.

1) *Robustness and accuracy* : To test the robustness of the proposed algorithm, we perform several randomized experiments in different settings. In the LAAS and Bremen maps, we randomize the position of the robot and evaluate

¹<https://github.com/razimantv/mazegenerator>

the detections manually. We did 50 such evaluations for each of the above two maps. In the next set of experiments, we generate a random map and place the robot in a random pose and then evaluate the detections. In this case, we have done 100 evaluations using 100 randomly generated maps. The calculated accuracy of our invisible humans detection algorithm based on these 200 experiments is 76.85%. However, if we include the overlaps are true detections, it increases by over 12% to 89.16%. These overlaps could be reduced by improving the filtering. Table I shows the list of experiments and the accuracy in each case.

Experiment	Invisible Humans Detection	
	Accuracy (%)	Accuracy with overlap (%)
LAAS	91.82	94.55
Bremen	65.28	81.94
Random	76.90	90.42
Overall	76.85	89.16

TABLE I: Accuracy calculated based on 200 experiments in 3 different environments. By correcting the overlapping detections, the accuracy could be increased by over 12%.

Experiment	Passage Detection	
	Accuracy (%)	Misses due to limits (%)
LAAS	66.00	30.00
Bremen	54.00	36.00
Random	65.00	14.0
Overall	62.50	23.50

TABLE II: Accuracy calculated based on 200 experiments in 3 different environments. Increasing the limits of detection could increase the accuracy by over 23%. These limits could be decided based on the requirement.

For calculating the accuracy of passage detection, we have performed similar experiments as above and evaluated the detections in 200 experiments. Here, we classify the detection simply either as true or false. There are also cases where there are no detections. In such cases, no detection within limits is classified as false, and out of the limits is classified as a miss. Table II shows the accuracy of passage detection in different settings. The overall accuracy based on these experiments is around 62.50%. Note that the percentage of misses is around 23.50%. This can be improved by having higher or adaptive detection limits. They have to be set based on the requirement.

2) *Comparison with other planners*: To test the advantage of the proposed approach, we compare the sudden emergence scenario using three different planners. The first one is Simple Move Base (SMB), where humans are added using the laser scan. Then we use CoHAN with and without the proposed constraint as the other two planners. As the

Planners	Min. Human-Robot Distance (m)
SMB	0.584
CoHAN	0.922
CoHAN with constraint	1.247

TABLE III: Minimum human-robot distance in the sudden emergence scenario using 3 different planners. The results are the average over 5 runs.

proposed work makes the robot cautious and tries to reduce the surprise to humans, we check the minimum human-robot

distance while navigating using these planners. We have performed 5 runs of the same scenario with each planner, and the results are presented in Table III. From Table III, we can see that adding the invisible humans constraint to CoHAN makes the robot maintain a larger distance from the human around a corner. Keeping distance from humans avoids surprise to humans and provides time for the robot planner to adapt slowly.

V. REAL WORLD RESULTS

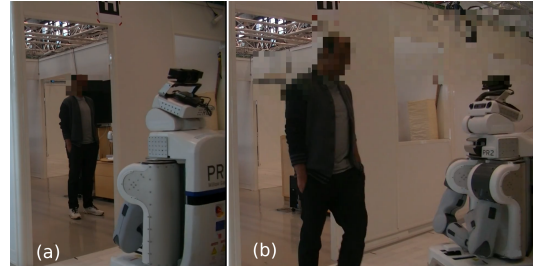


Fig. 11: Real world experiment setting (a) Human is inside the room and does not move. (b) The human starts coming out of the door as the robot approaches the door. In both scenarios, the robot tries to pass through the door.

The CoHAN system is installed on the PR2 robot in our lab and then tested in the doorway scenario discussed above. We performed two kinds of experiments, as shown in Fig. 11. In the first situation, shown in Fig. 11 (a), the human remains stationary, whereas in the second situation, shown in Fig. 11 (b), he comes out of the door as the robot approaches the door. The first situation is tested, with and without the invisible humans constraint, and the results are shown in Fig. 12. We can see from the figure that the real-world results match the results of the simulation approximately both in the paths and the speed profiles. Note that in Fig. 12 (c), the robot halts momentarily. This may be because of a sudden human appearance or moving very close to the wall. The

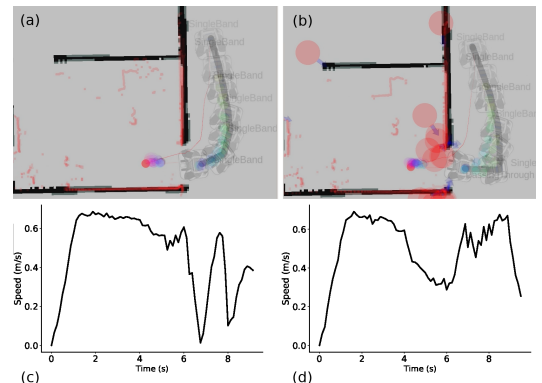


Fig. 12: Paths and speed profiles without (a, c) and with (b, d) invisible humans constraint. The addition of the constraints makes robot takes larger turn (b).

second situation is similar to the sudden emergence scenario, and the results of this run are shown in Fig. 13. The robot takes a larger turn and slows down twice, once to align itself for the door and then when the human emerges. The video

showing some more experiments and the tested scenarios can be found at <https://youtu.be/cbeFRkEdGgA>.

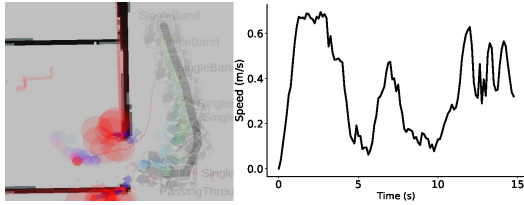


Fig. 13: Path and speed profile in the sudden emergence scenario.

VI. DISCUSSION AND LIMITATIONS

The introduction of invisible humans into human-aware navigation planning is relatively new and requires further research. We present one possible approach to address this issue. What is particularly interesting here is that our approach is modeled as a situation assessment and prediction ability to integrate into a mobile robot human-aware navigation. Having this, we will have a robot that can interact, using several modalities, with humans present in its field of view while making provisions to adapt to humans that are not yet seen. One difficulty we faced is the integration all these features without being “too conservative” and avoiding another case of the “freezing” robot. However, there are still some limitations to this approach. Since the approach is based on a 2D map, we can have false detections in the regions visible through the head of the robot but not through the base. There may also be some false detections in complex maps. Both these can be mitigated by augmenting the current approach with new sensor data and filtering further.

VII. CONCLUSION

We have proposed an approach to estimate the locations of invisible humans on a 2D map. These invisible humans are then integrated into our human-aware navigation planner via a new constraint. We also show how narrow passages can be identified by exploiting the detected invisible humans. We have presented the qualitative analysis of several simulated scenarios, followed by the results of accuracy and comparisons with some planners. Finally, we have shown the real-world experiments and presented some discussion. In the future, we plan to refine this approach further and address the different modalities identified in a better manner. We also aim to build a complete human-aware navigation system that can address very intricate human-robot interactions.

REFERENCES

- [1] M. Kollmitz, K. Hsiao, J. Gaa, and W. Burgard, “Time dependent planning on a layered social cost map for human-aware robot navigation,” in *2015 European Conference on Mobile Robots (ECMR)*, pp. 1–6, IEEE, 2015.
- [2] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, *et al.*, “Spencer: A socially aware service robot for passenger guidance and help in busy airports,” in *Field and service robotics*, pp. 607–622, Springer, 2016.
- [3] E. Repiso, G. Ferrer, and A. Sanfeliu, “On-line adaptive side-by-side human robot companion in dynamic urban environments,” in *IEEE/RSJ IROS*, 2017.
- [4] R. Guldensing, M. Görner, N. Hendrich, *et al.*, “Learning local planners for human-aware navigation in indoor environments,” in *IEEE/RSJ IROS*, 2020.
- [5] R. Möller, A. Furnari, S. Battiato, A. Härmä, and G. M. Farinella, “A survey on human-aware robot navigation,” *Robotics and Autonomous Systems*, vol. 145, p. 103837, 2021.
- [6] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, “Human-aware robot navigation: A survey,” *Robotics and Autonomous Systems*, vol. 61, pp. 1726–1743, Dec. 2013.
- [7] K. Madhava Krishna, R. Alami, and T. Simeon, “Safe proactive plans and their execution,” *Robotics and Autonomous Systems*, vol. 54, pp. 244–255, Mar. 2006.
- [8] R. Alami, K. M. Krishna, and T. Simeon, “Provably Safe Motions Strategies for Mobile Robots in Dynamic Domains,” in *Autonomous Navigation in Dynamic Environments*, vol. 35 of *Springer Tracts in Advanced Robotics*, pp. 85–106, Springer Berlin Heidelberg, 2007.
- [9] K. Madhava Krishna, R. Alami, and T. Simeon, “Moving Safely but not Slowly - reactively Adapting Paths for Better Trajectory Times,” in *1th International Conference on Advanced Robotics*, (Coimbra, Portugal), June 2003.
- [10] W. Chung, S. Kim, M. Choi, J. Choi, H. Kim, C.-b. Moon, and J.-B. Song, “Safe navigation of a mobile robot considering visibility of environment,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3941–3950, 2009.
- [11] S. Bouraine, T. Fraichard, and H. Salhi, “Provably safe navigation for mobile robots with limited field-of-views in unknown dynamic environments,” in *2012 IEEE International Conference on Robotics and Automation*, pp. 174–179, 2012.
- [12] J. Miura, Y. Negishi, and Y. Shirai, “Adaptive robot speed control by considering map and motion uncertainty,” *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 110–117, 2006.
- [13] A. Lambert, D. Gruyer, G. Saint Pierre, and A. N. Ndjeng, “Collision probability assessment for speed control,” in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp. 1043–1048, IEEE, 2008.
- [14] M. Phillips and M. Likhachev, “Sipp: Safe interval path planning for dynamic environments,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 5628–5635, IEEE, 2011.
- [15] T. Schoels, L. Palmieri, K. O. Arras, and M. Diehl, “An nmmpc approach using convex inner approximations for online motion planning with guaranteed collision avoidance,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3574–3580, IEEE, 2020.
- [16] C. Rösmann, A. Makarow, and T. Bertram, “Online motion planning based on nonlinear model predictive control with non-euclidean rotation groups,” in *2021 European Control Conference (ECC)*, pp. 1583–1590, IEEE, 2021.
- [17] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, “Efficient trajectory optimization using a sparse model,” in *2013 European Conference on Mobile Robots*, pp. 138–143, IEEE, 2013.
- [18] H. Khambhaita and R. Alami, “Viewing robot navigation in human environment as a cooperative activity,” in *International Symposium on Robotics Research (ISSR 2017)*, p. 18p, 2017.
- [19] P. Teja S. and R. Alami, “Hateb-2: Reactive planning and decision making in human-robot co-navigation,” in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 179–186, 2020.
- [20] P. T. Singamaneni, A. Favier, and R. Alami, “Human-aware navigation planner for diverse human-robot interaction contexts,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5817–5824, IEEE.
- [21] M. Quigley, K. Conley, *et al.*, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, 2009.
- [22] K. Teknomo, *Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model*. PhD thesis, 03 2002.
- [23] T. I. Lakoba, D. J. Kaup, and N. M. Finkelstein, “Modifications of the helbing-molnar-farkas-vicsek social force model for pedestrian evolution,” *Simulation*, vol. 81, no. 5, pp. 339–352, 2005.
- [24] D. Helbing, I. Farkas, and T. Vicsek, “Simulating dynamical features of escape panic,” *Nature*, vol. 407, no. 6803, pp. 487–490, 2000.
- [25] G. Echeverria, N. Lassabe, *et al.*, “Modular open robots simulation engine: Morse,” in *IEEE ICRA*, 2011.
- [26] A. Favier, P. Singamaneni, and R. Alami, “An intelligent human simulation (inhus) for developing and experimenting human-aware and interactive robot abilities,” 2021.