Human-Aware Navigation in Crowded Environments using Adaptive Proxemic Area and Group Detection

Carlos Medina-Sánchez[1*], Simon Janzon[1*], Matteo Zella[2], Jesús Capitán[3] and Pedro J. Marrón[1]

***Abstract*— Navigation is an essential task for social robots. However, certain rules must be followed to allow them to move without causing distraction or discomfort to people. Considering that the context surrounding robots and persons affects the expected behavior, this work defines a social area around a person that adapts to the real situation. In addition, the social context of a person is extended to identify groups of people, which the robot should take into account while navigating. With this understanding of the surrounding of the robot together with the ability to predict the trajectory of individuals as well as groups, the proposed solution not only effectively addresses collision avoidance while promoting socially acceptable behavior but also outperforms the majority of recent works in terms of accuracy. Furthermore, a dedicated policy is introduced to react to social navigation conflicts. The evaluation performed in a simulated environment shows that the computation of our proposed solution is at least 8 times faster than the best state-of-the-art approach while preserving comparable social conduct. Also, the results of realistic experiments performed using Gazebo and a real robot are reported.**

I. INTRODUCTION

The presence of robots in social environments is becoming a reality. Nowadays we can see robots being deployed in a wide variety of environments, from large automotive industries [1], delivery centers [2], airports [3], museums [4] or even in our homes [5]. However, enabling a natural and comfortable co-existence of robots and people is still an open challenge. One of the most addressed aspects is how the robot should react to the presence of individuals or groups of people while moving in an environment [6]. For this task, the robot can exploit perceived contextual information to perform the most suitable action according to commonly accepted social rules. The problem is an interdisciplinary one, involving disciplines like psychology and sociology. For example, the theory of Proxemics [7] defines certain areas around people that indicate how comfortable an individual feels when another person or object enters these zones. This theory, when applied to robot navigation in social environments, allows us to define how close the robot can be to a person without creating discomfort.

In this paper, we address social navigation by defining an *Adaptive Proxemic Area* (APA), whose shape changes depending on contextual information perceived by the robot. To achieve this goal, we introduce a mathematical model that differentiates between individuals and groups of people, while taking into account the presence of obstacles in the environment. By considering the predicted movements of the persons in the surrounding, the APA is used to compute possible evasion points in an effective manner. The selection of the exact path to follow weights both the persons' as well as the robot's perspective and selects the most comfortable route. In doing this, we differentiate from the state of the art (presented in Section II) by exploiting the APA model in order to decrease the computational effort without affecting the robot's social consciousness.

In conclusion, the main contributions of this work are: (1) the definition of an accurate model for group detection (Section III-A); (2) the mathematical formulation of an Adaptive Proxemic Area used for social navigation (Section III-B); (3) a conflict resolution that allows the robot to effectively react to collisions with agents in the scene (Section III-C). We have evaluated our work (Section IV) in comparison to state-of-the-art navigation approaches using the SocNavBench [8] simulator, which bases on real-world datasets from UCY and ETH [9]. After discussing the effectiveness of our solution as well as its limitations, Section V provides an overview of our contributions in the context of future work.

II. RELATED WORK

Social navigation is an essential element in applications where robots share their environment with people. For this reason, it is necessary that robots behave in an acceptable way without disturbing people, while still being efficient in the development of their tasks. With this in mind, multiple papers on this subject have been published in recent years. For instance, Kruse et al. [10] define the important aspects to consider while navigating. These elements include the *comfort* of the people in the presence of the robot, the *naturalness* with which the robot moves and how the robot interacts with the people around it (*sociability*).

In contrast to Kruse et al., Charalampous et al. [11], discuss the topic of navigation from a mapping perspective. The authors start with the definition of metric maps, where it is only distinguished whether or not a section can be traversed by the robot. On top of this map, a semantic layer is defined in which multiple objects including people are

*This work was not supported by any organization. The first two authors contributed equally to this work. Asterisk indicates corresponding authors

[1]C. Medina-Sánchez, S. Janzon, and P.J. Marrón are with the Institute for Computer Science and Business Information Systems (ICB) of the Faculty of Business Administration and Economics, University of Duisburg-Essen, 45127 Essen, Germany carlos.medina-sanchez@uni-due.de, simon.janzon@stud.uni-due.de and pjmarron@uni-due.de

[2]Matteo Zella is with the Faculty of Electrical Engineering and Computer Science, University of Applied Sciences Niederrhein, 47805 Krefeld, Germany matteo.zella@hs-niederrhein.de

[3]Jesús Capitán is with the Department of System Engineering and Automation, University of Seville, 41092 Seville, Spain jcapitan@us.es

identified and classified. Finally, a social layer is added, in which multiple parameters of the agents themselves are taken into account, as well as their relationship with other agents and with the environment. Based on this information, it is possible to identify different zones according to the *Proxemics* theory [12]. The invasion of such zones by external persons (or robots) creates discomfort and should, therefore, be avoided by a robot during navigation.

Rios et al. [13] also show how proxemics can be applied in social navigation by defining different areas around individuals and groups of people that may or may not be invaded. Navigation is then classified in 3 possible groups: (1) robots and people negotiate space (*Unfocused interaction*); (2) the robot adapts to the changes of people (*Focused Interaction*); (3) a combination of the two previous ones (*Focused and Unfocused interaction*). The work also proposes several variations to the original concentric model defined by the proxemics theory. Among these variations is the "egg shape", which maintains a concentric shape in the direction opposite to the orientation of the agent's head and an elliptical shape in the direction of the head. A second adaptation is made by a completely elliptical shape that deforms equally in the direction of the head as well as in the opposite direction. Finally, a variation is presented that reduces the space on the dominant side of the person (right or left).

In the same way, Truong et al. introduce the *Dynamic Social Zone* (DSZ) [14], which takes into account personal space, human states, and human social signals for human safety in robot navigation tasks. The first aspect is related to the proxemics areas, the second one includes people's position and velocity, and the last one includes contextual information like hand detection. All this information allows the author to create a DSZ that changes its shape depending on these variables. This work only considers the current position of the person and not future states. Also, they evade people without considering the possibility that the robot may enter these zones, something that could happen in closed environments with people crossing doors.

As presented by Rios [13], the detection of groups of people plays an important role in social navigation, given that proxemics areas change. For example, a robot should not navigate traversing a group of two people in conversation. This is why some work has focused on determining when an agent is considered part of a group. In this area, one of the first published works was that of Ciolek and Kendon [15], who proposed a series of conditions for two individuals to be considered as part of a group, using their position and orientation. Also, Kendon [16] expanded his research with an exhaustive study on the behavior of multiple people in groups, concluding that their formation tends to be concentric, and that parameters such as the people position and orientation need to be taken into account. Following this line, Truong et al. [17] included interactions between multiple people modifying the Graph Cuts of F-Formation algorithm, adding the motion information to the model [18].

Both proxemics theory and group detection are essential for robot navigation in a social environment. However, it is necessary to apply this information to the navigation planners. This is done in works like the one presented by Zhou et al. [19], where the On-line Collision Avoidance for Dynamic Vehicles is presented, which uses Buffered Voronoi Cells to derive an optimal avoidance model. Moreover, Vega-Magro [20] proposes a solution that integrates multiple layers of information. In particular, a social layer is defined, where the detection of people is performed. By using the proxemics theory, the authors define an adapted proxemics area based on a Gaussian function. Once these areas are defined, the proposed solution checks for overlapping in order to determine whether there is interaction between people and thus create a group. Based on this information, they create a navigation layer with the areas that are traversable by the robot and apply the Dijkstra algorithm to obtain the robot's trajectory.

Additionally, Singamaneni et al. [21] introduce the Human Safety and Human Visibility layers, where Gaussian areas of 3-meter radius are defined. The authors also make use of a state prediction module for agents, based on the goal that the human wants to reach. Then a local planner uses the proposed layers and human trajectory prediction to compute the robot's trajectory. The work defines four navigation methods: the first one considers that there are no agents in the environment; the second one only considers the information of the two closest agents; the third one considers the information of all agents; and the last one considers the case where there is no possible solution for the robot to reach its goal.

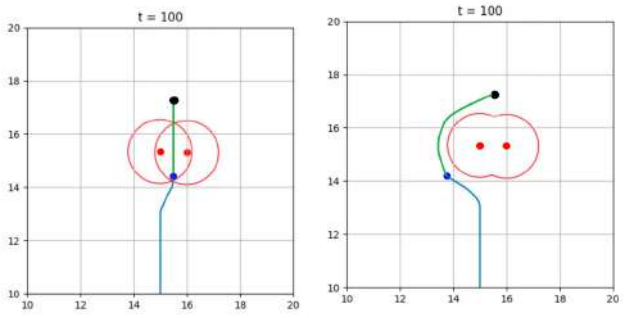
Our work focuses on the use of contextual information during the computation of an adaptive proxemic area, which adjusts to the current situation. Differently from the state-of-the-art approaches, we integrate all the modules for a successful social navigation with a reduced computational effort: group detection, trajectory prediction, adaptive proxemic area, evasion policies, and motion controller.

III. PROPOSED METHOD

The goal of the proposed method is to provide robots with the ability to efficiently navigate in social environments. In this section, we describe the three components building up our solution. First, considering the influence that group detection has on social navigation, i.e. avoiding navigating in the middle of two people talking (see Figure 1a without group detection and Figure 1b with group detection), we present the model formulated to determine whether the agents moving in an environment are part of a group. Afterward, we introduce the concept of APA, which is a variation of the original proxemics theory that adjusts according to the current context of the person. Also, we present how we calculate the trajectory that the robot should follow taking into account the APA, the predicted trajectory of the people, and the local robot information. Finally, we present our motion controller.

A. Group Detection

We assume that a group is a set of people who are somehow interacting with each other and have certain similar



(a) (b)

Fig. 1: Social navigation (a) without group detection and (b) with group detection with two static persons.

characteristics in terms of orientation, position and speed. We detect groups by computing the *Position Similarity* (σ_d), the *Orientation Similarity* (σ_h) and the *Speed Similarity* (σ_s) among the different individuals present in the environment. At the beginning, each individual builds up a group composed only by one person. The three aforementioned parameters are then computed and compared between all the possible group pairs. If the resulting similarity is above a given threshold (τ), the groups are merged into one. Algorithm 1 presents the general process of group classification, where $w_{distance}$, $w_{heading}$ and w_{speed} are constants and the remaining parameters are discussed in detail in the following.

Algorithm 1 for group detection is executed with the same frequency as the person detection algorithm. This allows that at each time instant people are grouped depending on the similarity results, avoiding the need to ungroup agents that have formed a group in the previous samples and that may take different paths at a specific point.

Algorithm 1 Group detection algorithm

$i \leftarrow 0$
for i < groups.length - 1 **do**
  $j \leftarrow 0$
  for j < groups.length **do**
    if $i \neq j$ **then**
      $\sigma_{distance} \leftarrow w_{distance} * \sigma_d$
      $\sigma_{heading} \leftarrow w_{heading} * \sigma_h$
      $\sigma_{speed} \leftarrow w_{speed} * \sigma_s$
      $\sigma_{p_j} \leftarrow \sigma_{distance} + \sigma_{heading} + \sigma_{speed}$
      if $\sigma_{p_j} > \tau$ **then**
        $g_i.add(g_j)$
        $delete(g_j)$
        $j \leftarrow i - 1$
      end if
    end if
  end for
end for

The *Position Similarity* is calculated by taking the Euclidean distance (Δ_d) between two groups, which could be formed by one or more individuals. In the case of a group i, we denote the position of the group as g_{i_p}. This position is either the position of the individual for groups with a single person or the average position among all the members. Equations 1 and 2 are used for the calculation of the Euclidean distance (Δ_d) and the final similarity score (σ_d), where v_d is a constant factor. Since the distance between agents affects more than the speed and orientation, this relationship is exponential and not linear as in the other cases. Moreover, this exponential should be negative, since the greater the distance, the smaller the result of Equation 2 and therefore the lower the probability of group formation.

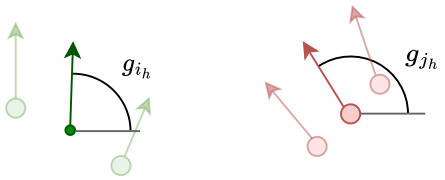


Fig. 2: Variables for the calculation of *Orientation Similarity*. In light green the position of the persons in the group i, in dark green the average position of the persons in the group i and its average orientation (g_{i_h}), in light red the position of the persons in the group j and in dark red the average position of the persons in the group j and its orientation (g_{j_h}).

$$\Delta_d = \sqrt{(g_{i_p}.x - g_{j_p}.x)^2 + (g_{i_p}.y - g_{j_p}.y)^2} \quad (1)$$

$$\sigma_d = e^{-v_d * \Delta_d} \quad (2)$$

The *Orientation Similarity* is obtained by comparing the average orientation of all group members (g_{i_h}) in relation to the orientation of a second group (g_{j_h}). Figure 2 shows the variables used for the *Orientation Similarity* calculation. To obtain the value of this similarity score we use the absolute angle difference that could take values from 0 to π and then scale the result to obtain a value between 0 and 1, as Equation 3 shows.

$$\sigma_h = 1 - \frac{||g_{i_h}| - |g_{j_h}||}{\pi} \quad (3)$$

Finally, the *Speed Similarity* between each two groups is calculated. For this, the average speeds of all members of both groups are calculated (g_{i_s} and g_{j_s}) and compared (see Figure 3). As in the previous parameter, the absolute speed difference is calculated and assigned a score (Equation 4) between 0 and 1 to be used for group detection.

$$\sigma_s = 1 - \frac{|(g_{i_s} - g_{j_s})|}{max(g_{i_s}, g_{j_s})} \quad (4)$$

B. Adaptive Proxemic Area

As mentioned above, our proposal makes use of the theory of proxemics by defining an area around the groups of agents that the robot should not intrude. Due to the people dynamics in the environment, we consider that a static area is not enough. For example, the speed of a person directly affects

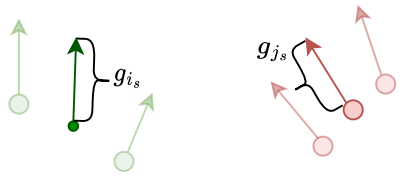
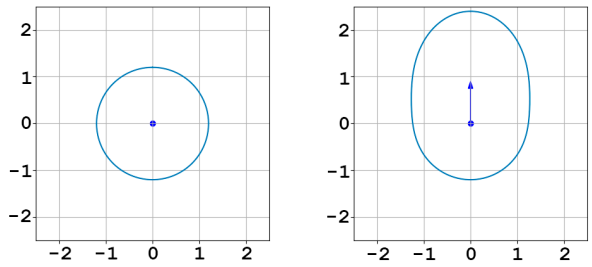


Fig. 3: Variables for the calculation of *Speed Similarity*. In light green position of the persons in the group i, in dark green the average position of the persons in the group i and its average speed vector (g_{i_s}), in light red the position of the persons in the group j and in dark red the average position of the persons in group j and its speed vector (g_{j_s}).



(a) (b)

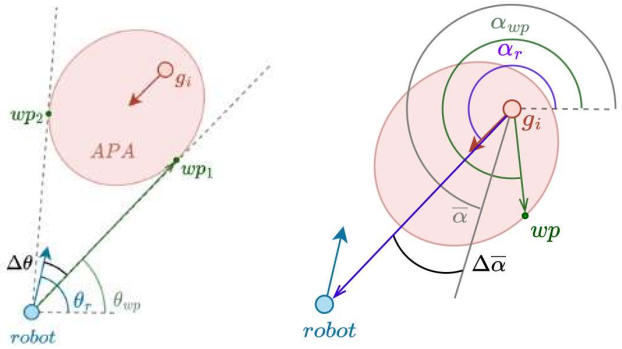
Fig. 4: Example of an *Adaptive Proxemic Area* with $\sigma = 0.6$, $d = 1.2$ and $g_{i_h} = 90°$ and different speed factors: in (a) $a = 0.0$ and in (b) $a = 1.80$.

robot evasion tasks, since the higher the speed, the more space is needed for evasion while maintaining the safety of the person. Thus, we introduce the concept of an Adaptive Proxemic Area. This area extends the traditional circular shape in the same direction of the moving group with a half ellipse, whose semi-major axis is aligned with the group heading and has a length proportional to the group speed. The resulting area is calculated by Equation 5, where a is defined as $w_v g_{i_s}$, with g_{i_s} the speed of the group and w_v its weight; σ is the deformation factor; and d is the minimum radius of the area. An example of the resulting area is shown in Figure 4 where $d = 1.2m$ which is the external limit of the personal area in the proxemics.

$$P(x) = \begin{cases} \frac{a}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x}{\sigma})^2} + d & \text{if } x \in [g_{i_h} - \frac{\pi}{2}; g_{i_h} + \frac{\pi}{2}], \\ \frac{a}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{2\pi - x}{\sigma})^2} + d & \text{otherwise} \end{cases} \tag{5}$$

C. Human-Aware Navigation

For navigating the robot, we consider two relevant scenarios, the first one in which the robot is outside the APA and must avoid entering this area (*Out-of-Area Evasion*). The second one, where the robot is already inside the area and must leave it as soon as possible (*In-Area Evasion*) in a socially acceptable way. Since the focus of this work is not people trajectory prediction, human motion modeling or human motion learning, we have considered the *Constant Turn Rate and Acceleration Motion Model* (CTRA) defined



(a) (b)

Fig. 5: Out-of-Area Evasion for Human-aware navigation. Definition of parameters for (a) *Similarity of Robot's Orientation* and (b) *In-The-Wayness* score calculation.

in [22] for human motion prediction, which provides sufficient prediction capabilities.

1) Out-of-Area Evasion: When a future robot-human collision is detected at the Prediction Window (p) given in seconds used as parameter in the CTRA model, two waypoints are defined demarcated by the tangents of the APA in relation to the position of the robot (wp_1 and wp_2 in Figure 5). Once these points are defined, two features are used for waypoint selection, the *Similarity of Robot's Orientation* and *In-The-Wayness* of a waypoint in relation to the group.

The *Similarity of Robot's Orientation* score is obtained by Equation 6, which uses the absolute value of the angular difference ($\Delta\theta$) between the orientation of the robot (θ_r) and the angle of the waypoint (θ_{wp}). Additionally, the factor ω_θ is calculated based on the relationship between the absolute agent's speed (v_p) and the absolute robot's speed (v_r) using Equation 7.

$$\sigma_\theta = 1 - \frac{\Delta\theta}{\pi} \tag{6}$$

$$\omega_\theta = \begin{cases} 1 - |\frac{v_r}{v_p}| & \text{if } |\frac{v_p}{v_r}| \in [0; 1], \\ 0 & \text{if } |\frac{v_p}{v_r}| \in (1; +\infty], \end{cases} \tag{7}$$

If we define $\overline{\alpha}$ as the median angle between the angle to the robot position (α_r) and the angle to the waypoint (α_{wp}) (see Figure 5b), the *In-The-Wayness* score of a waypoint in relation to a group is calculated through Equation 8, in which $\Delta\overline{\alpha}$ is the angular difference between $\overline{\alpha}$ and the angle to the robot position (α_r). Finally, the factor w_α is calculated to weight this score using Equation 9, where v_p and v_r are absolute values of the person and robot speeds.

$$\sigma_\alpha = 1 - \frac{\Delta\overline{\alpha}}{\pi} \tag{8}$$

$$\omega_\alpha = \begin{cases} \frac{v_r}{v_p} & \text{if } \frac{v_p}{v_r} \in [0; 1], \\ 1 & \text{if } \frac{v_p}{v_r} \in (1; +\infty], \end{cases} \tag{9}$$

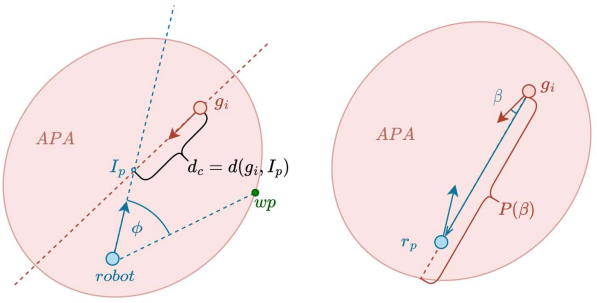


Fig. 6: In-Area Evasion for human-aware navigation. Definition of parameters for (a) *Intersection Score* and (b) *Intrusion Score* calculation.

Using Equation 10, the final score for the waypoints is defined and the one with the best score is chosen as the next waypoint for the robot toward its final destination.

$$\sigma_{wp} = \omega_\theta \sigma_\theta + \omega_\alpha \sigma_\alpha \tag{10}$$

2) In-Area Evasion: This evasion method is executed once the robot is inside an APA. This method is designed to make the robot leave the area as soon as possible in a natural way. In this case, an intermediate waypoint for the robot is defined by its current orientation with an offset (ϕ). To obtain this angle, we use two features, the *Intersection Score* and the *Intrusion Score*.

To obtain the *Intersection Score*, first, it is necessary to calculate the collision distance (d_c) between the group's position (g_i) and the point of intersection (I_p) with the robot (Figure 6a). Then the score of this feature is obtained by using Equation 11, where $P(0)$ is calculated by Equation 5.

$$\sigma_{is} = 1 - \frac{d_c}{P(0)} \tag{11}$$

On the other hand, the *Intrusion Score* is obtained by Equation 12, where β is the relative angle between the group's orientation and the robot's position, r_p is the robot position, g_i the group position and n is the number of groups (Figure 6b).

$$\sigma_{it} = 1 - \frac{\sqrt{\sum_{i=0}^{n}(r_{p_i} - g_i)^2}}{P(\beta)} \tag{12}$$

Using the *Intersection Score* and the *Intrusion Score*, the final score (σ_{wp}) for the waypoint is calculated using Equation 13. Finally, the angle offset is defined by Equation 14, when the evasion happens in the front of the group and the robot is not facing away, otherwise, ϕ is equal to 0.

$$\sigma_{wp} = \omega_{is} \sigma_{is} + \omega_{it} \sigma_{it} \tag{13}$$

$$\phi = \begin{cases} \sigma_{wp} * \frac{\pi}{2} & \text{if } \theta_{wp} > \theta_r, \\ -\sigma_{wp} * \frac{\pi}{2} & \text{otherwise} \end{cases} \tag{14}$$

3) Special considerations: In social environments with few people, waypoint selection does not represent any particular challenge, since in most cases it is possible to find an accessible point for the robot. However, in crowded environments, waypoint selection can be more complex. One of the problems identified is the probability that a calculated waypoint belongs to another group or individual. In this case, we create a waypoint list of all the robot's neighbors, then we check if the waypoints of an APA are inside the APA of another person, and if so, we eliminate them from the list. Finally, the waypoint that remains in the list and that needs less path to be reached is selected.

The second identified problem is when none of the waypoints calculated and added to the list can be reached by the robot. This case is the well-known "freezing robot problem" [23]. To handle this situation, we took some measurements. First, once it is detected that no waypoint can be reached, the speed of the robot is slowed down, to a stop if necessary, before invading the APA of any individual, and the humans can avoid the robot as another object in the environment. After that, with each sensor sample, we evaluate new waypoints until there is a free one and the navigation is re-started.

D. Motion Controller

We have implemented a motion controller that takes care of the speed control of the robot. For this, after defining the goal or waypoint to reach, we calculate the angle difference between the current robot orientation and the relative orientation between the robot and the goal in both directions of rotation. Once the side where the difference is smaller is identified, we increase the angular velocity of the robot to turn in that same direction. Once the angle difference is less than a pre-defined threshold, the angular velocity is decreased and the linear velocity is activated. Both velocities remain active until the orientation of the robot is aligned with the goal, at which point the angular velocity is deactivated and only the linear velocity remains active until the goal is reached.

IV. EVALUATION

We evaluate the proposed approach (Ours) first by optimizing the parameters involved in the group detection. Then we validate the behavior of the robot navigating in social environments.

A. Parameter Optimization

Our solution is based on different heuristics with several parameters. In order to identify their optimal value, we perform an optimization step using available datasets.

1) Group Detection: We use the ETH Hotel dataset [24] introduced by Ess et al. [25] to optimize the group detection parameters. For this, we take the first half of the video where people and groups are labeled. In this video, the approximate position of each person is calculated taking into account that 55 pixels of the recorded image correspond to 1m in real life. Also, the speed of the persons is calculated through the change of position between two frames, and their orientation

TABLE I: Resulting parameters from the optimization of the group detection procedure.

Parameter	Value
v_d	0.3
$w_{distance}$	0.74
$w_{heading}$	0.16
w_{speed}	0.1
τ	0.78

TABLE II: Resulting parameters from the optimization of the human-aware navigation.

Parameter	Value
τ_w	0.4 meters
p	4.0 seconds
w_v	2.0
σ	1.0
w_{is}	0.5
w_{it}	0.5

is computed by comparing the position of the person in two different frames of the video. In total, 10 different scenes are extracted, including 92 pedestrians and 17 groups.

To obtain the best combination of parameters, we ran Algorithm 1 by changing the values of the different variables. In the case of the threshold (τ), from 0.1 to 0.9 with steps of 0.01, and the rest of them with values from 0.1 and 0.8 with steps of 0.1. In total, more than 29,000 combinations were run. To determine whether a result was acceptable or not, we compared the results of these experiments with the manually labeled groups in the first part of the ETH Hotel dataset video. Using the combinations with more true positives group detections, we redefined smaller ranges for the variables changing the steps to all of them to 0.01. Finally, the combination of variables that reached the best true positive accuracy with around 94% are presented in Table I.

To evaluate the accuracy of the group detection algorithm, we labeled the groups in the second half of the ETH Hotel dataset. Using 10 scenes as samples, we compared the results of the group detection with the labeled groups and we found that our solution was able to detect correctly 16 of the 17 groups, reaching a true positive rate of 94%.

2) Navigation: Regarding the optimization of the parameters corresponding to the navigation procedure, we used 25 atomic scenarios similar to those described by Khambhaita [26]. These scenarios include examples of collisions with a person or group of persons approaching the robot from different directions, the robot overtaking a person or a group, the evasion of one or multiple stationary persons in the environment, the collision in a corridor, and the crossing of a doorway. The values obtained for the navigation parameters are presented in Table II. The variable d in Equation 5 takes the value of 1.2 meters which is the outer limit of the personal area in the Proxemics Theory.

B. Social Navigation

1) Simulation in SocNavBench: To evaluate the resulting social behavior of the robot, we conducted experiments using 33 episodes designed by the authors of SocNavBench [8]. These episodes are built using pre-sourced real data from the UCY and ETH [9] datasets, where each scenario contains between 24 and 72 people, with an average of 44. Also, we used an Alienware Area-51 R5 (836) with 32 GB of RAM and a Core i7-7800X CPU @ 3.50GHz processor. The speed of the robot in the simulation was set in $1m/s$. Although this simulator does not take into account the dynamics of the robot, the comparison is performed under the same circumstances for all approaches.

To verify the accuracy of our solution, we compare our method with 4 available solutions: *Pedestrian-Unaware Algorithm* (PUA) [8], which directs the robot towards the goal without taking collisions into account (without people or obstacle detection); *Optimal Reciprocal Collision Avoidance* (ORCA) [27], which makes the assumption that all agents in an environment move according to the same planning strategy, assuming reciprocal reactions by pedestrians and the robot; *Socially Aware Collision Avoidance with Deep Reinforcement Learning* (DRL) [28], which utilizes deep reinforcement learning to formulate solutions to navigational conflicts (rules that describe social awareness are also used); and *Social Forces* (SF) [29], which introduces different forces that act on the robot attracting it to its goal and repealing it from pedestrians and obstacles.

The metrics that we take into account are: the *Success Rate*, which is the fraction of episodes that are completed successfully, without collisions or timeouts; the number of *Pedestrians Collisions*; the average *Planning Wait Time*, defined as the average time in seconds the simulator waits for the algorithm to finish its planning step; the *Path Length*, defined as the distance in meters traveled by the robot to reach its goal; the *Path Irregularity*, defined as the average of the angle between the robot's orientation and the local angle between the robot and the goal; the *Closest Pedestrian Distance* in meters [6]; the *Time in Personal Area* in seconds, in which the personal area around a person is considered to have a radius of 1.2 meters.

The results are presented in Table III and IV. In terms of accuracy, our method proved to be the second best, completing 24 of the 28 scenarios without any type of collision (person or object) or timeouts. Of the four scenarios not completed by our algorithm, 3 of them were due to collisions with people and one due to collision with objects. The collisions with pedestrians were caused by the sensor detecting people late due to the obstruction of the field of view by another person or object. Only the pure SF performed better with 25 of the 28 scenarios without collisions. SF, instead, obtained one collision with people, one with an object and one timeout. In the case of DRL, it failed 10 of the 28 scenarios with 19 pedestrian collisions and one with objects. Finally, ORCA experienced 4 pedestrian collision and 3 timeouts with a total of 7 failed scenarios out of 28.

The most significant advantage of our method is a considerable decrease in the Planning Wait Time. In fact, the processing time is one-eighth of the second most efficient method (PUA) and 96 times more efficient than the slowest

TABLE III: Success Rate and Pedestrian Collisions.

Metric	PUA	ORCA	DRL	SF	Ours
Succ. Rate (%)	25	75	64.28	89.2	85.7
Pedestrian Col.	38	6	19	1	3
Object Col.	2	0	1	1	1
Timeout	0	3	0	1	0

TABLE IV: Median ($\overline{x}$) and Interquartile (IQR) for *Planning Wait times* (PWT) in seconds, Path Length (PL) in meters, Path Irregularities (PI) in radians, Closest Distance to a Pedestrian (CD) in meters and Time in Personal Area (TPA) in seconds.

Metric		PUA	ORCA	DRL	SF	Ours
PWT	$\overline{x}$	19.13	232.11	108.95	20.67	2.4
	IQR	6.82	175.97	54.22	29.67	1.68
PL	$\overline{x}$	18.29	16.36	16.57	18.62	17.72
	IQR	4.94	6.93	5.65	7.22	4.27
PI	$\overline{x}$	1.34	1.4	1.75	1.39	1.77
	IQR	1.6	1.72	1.8	1.67	1.71
CD	$\overline{x}$	0.11	0.42	0.43	0.86	0.58
	IQR	0.33	0.11	0.48	0.71	0.41
TPA	$\overline{x}$	3	3.75	2.95	1.43	2.6
	IQR	3.4	6.5	2.97	3.12	3.84

method ($ORCA$). This result highlights the practicality of our approach and its low computational requirements. In relation to the Path Length and Path Irregularity, our algorithm obtained the third and last place, respectively. However, observing the data it is clear that a shorter or less irregular path in social robotics does not mean to be the best because one of the most important aspects is safety, that we keep with one of the approaches with less pedestrian collisions. As we can see in Table III, $ORCA$ and DRL had shorter paths at the cost of higher collisions with pedestrians.

One of the most important metrics in social navigation is how close to the person the robot navigates. The more distant the robot is from a person, the more socially acceptable its behavior is considered. In this case, SF obtained the first place, keeping the robot at a median distance of $0.86\ m$, while ours obtained a result of $0.58\ m$, both distances remaining in the social zone defined by the theory of proxemics. While $ORCA$ and DRL, with medians of $0.42\ m$ and $0.11\ m$, are inside the personal area. Also related to the discomfort of the persons in the environment is the time that the robot remains in the personal area. In this aspect, SF obtained a result of $1.43\ s$, while our method obtained $2.6\ s$. This is due to the way in which the robot tries to leave the area. While SF tries to move away from the person in a faster way, our method tries to avoid the person without abrupt changes that could make people uncomfortable.

Finally, one aspect observed during the simulations was that a purely SF-based method leads to reactive navigation which does not predict future states and, therefore, the robot behavior may not be perceived as *natural*. Methods such as DRL or ours take into account the prediction of people's trajectories, resulting in predictive navigation that is perceived more *natural*. A demonstration video of our method can be found at https://youtu.be/3Uf2nvLK-KA.

TABLE V: Comparison results between a *simulated* framework and emulated environment with Turtlebot 2 in Gazebo. The minimum distance to pedestrian (d_{min}) and the path smoothness Δ_θ are presented.

Scene	d_{min} (meters)			Δ_θ (radians)		
	Python	Gazebo	Real	Python	Gazebo	Real
Front	1.79	0,91	1.75	0.02	0.1	0.16
Side	1.92	1.06	1.19	0.01	1.7	0.18
Back	1.43	0.3	1.27	0.02	0.6	0.12
Still	1.69	1.70	1.16	0.1	0.8	0.1

2) Turtlebot 2 experiments: Finally, we implemented our approach in a ROS node to navigate with a Turtlebot 2. We used angular and linear speeds at the same time to have more efficient movements using the motion controller proposed in the previous section. To evaluate our approach with the Turtlebot 2, we test it in the same 25 atomic scenarios that we used for parameter optimization in navigation. We select 4 of these scenarios and the results are presented in Tables V and VI. In these tables, we present the comparison of the robot behavior using a pure simulated environment developed using Python tools that do not take into account the dynamics of the robot (Python). Also, we present the results for a simulated environment using the Turtlebot 2 and Gazebo (Gazebo) and finally the results for a real environment using a real Turtlebot 2 (Real).

In the first scenario, a person approaches the robot from the front (Front). In the second scenario, a person moves perpendicularly at 90 degrees to the robot (Side). The third scenario consists of a person approaching the robot from the rear (Back), and in the last scenario, there is a static person in the path of the robot toward the goal (Still). The maximal speed of the robot was set to $0.5\ m/s$, taking into account the dynamics of the Turtlebot 2.

In the results, the influence of the robot dynamics on the behavior of our method is evident. For example, due to the acceleration and velocities (v) reached by the Turtlebot 2, no collisions with people are evidenced. However, in all cases where the person moves, the distance between the robot and the person (d_{min}) is reduced. The dynamics also influence the smoothness of the path (Δ_θ), as the robot tries to rotate more to avoid people. However, the length (l) is decreased, as the speed of the person influences the amount of adjustments to be made to the path and how the robot reacts to the evasion. However, the results obtained through the *Real* experiment showed that the robot behaves acceptably, staying within the social area defined by the p Proxemics in the worst case. The difference in the results between *Gazebo* and *Real*, is due to the fact that in the first case, the speed of the people was constant at 1 m/s while in the second case, there was an average speed of 0.75 m/s. The complete video with the behavior of the Turtlebot 2 in Gazebo and real-world experiments can be seen at https://youtu.be/UDUtqZ-En-s.

V. CONCLUSIONS

In this paper, we have presented a method for efficient social navigation that exploits group detection and trajectory

TABLE VI: Comparison results between a *simulated* framework and emulated environment with Turtlebot 2 in Gazebo. The path length (l) and the average velocity of the robot v are presented.

Scene	l (meters)			v ($v_{max} = 0.5m/s$)		
	Python	Gazebo	Real	Python	Gazebo	Real
Front	11.1	10.25	10.61	0.49	0.38	0.33
Side	11.44	10.51	12.44	0.48	0.16	0.31
Back	12.2	10.04	11.2	0.48	0.43	0.31
Still	10.65	10.01	11.12	0.49	0.4	0.33

prediction, resulting in a path with a lower number of collisions that keeps the robot outside the personal area defined by the proxemics theory. We introduce the concept of *Adaptive Proxemic Area*, which can be easily extended to integrate more information about the person's context. We show that our method performed well compared to other solutions, in particular for what concerns the required computational effort. Pure SF offers a better performance in several aspects at the cost of a reactive response, which can cause discomfort to people. We deem the provide comparative experimentation to be helpful, in particular in highlighting both the different trade-offs and the existing space for optimizations, e.g., in resource utilization.

While focusing on the latter, the evaluation of our approach shows also the need to extend the Prediction Window to the cases in which persons cannot be perceived by the sensor because they are hidden behind other persons or obstacles. To address these cases, we plan as next steps the extension of our formulation through the integration of additional context information. Also, the integration of external sensors that perceive the complete scenario could help to avoid collisions with persons that are in the shadow of the robot's sensors. Finally, it is necessary to extend the solution for the *robot freezing problem* [23], in order to improve the behavior in very crowded environments.

REFERENCES

[1] C. Garriz and R. Domingo, "Development of trajectories through the kalman algorithm and application to an industrial robot in the automotive industry," *IEEE Access*, vol. 7, pp. 23 570–23 578, 2019.

[2] J. Hooks, M. S. Ahn, J. Yu, X. Zhang, T. Zhu, H. Chae, and D. Hong, "Alphred: A multi-modal operations quadruped robot for package delivery applications," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5409–5416, 2020.

[3] "Spencer project," http://www.spencer.eu/, accessed: 2022-08-30.

[4] A. Moualla, A. Karaouzene, S. Boucenna, D. Vidal, and P. Gaussier, "Readability of the gaze and expressions of a robot museum visitor: Impact of the low level sensory-motor control," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017, pp. 712–719.

[5] "Irobot roomba," https://www.irobot.com/en_US/roomba.html, accessed: 2022-08-30.

[6] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, 2023.

[7] S. M. Bhagya, P. Samarakoon, M. A. Viraj, J. Muthugala, A. G. Buddhika, P. Jayasekara, and M. R. Elara, "An exploratory study on proxemics preferences of humans in accordance with attributes of service robots," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2019, pp. 1–7.

[8] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni, "Socnavbench: A grounded simulation testing framework for evaluating social navigation," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 11, no. 3, pp. 1–24, 2022.

[9] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 261–268.

[10] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[11] K. Charalampous, I. Kostavelis, and A. Gasteratos, "Recent trends in social aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 93, pp. 85–104, 2017.

[12] E. T. Hall, R. L. Birdwhistell, B. Bock, P. Bohannan, A. R. Diebold Jr, M. Durbin, M. S. Edmonson, J. Fischer, D. Hymes, S. T. Kimball, *et al.*, "Proxemics [and comments and replies]," *Current anthropology*, vol. 9, no. 2/3, pp. 83–108, 1968.

[13] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 137–153, 2015.

[14] X.-T. Truong, V. N. Yoong, and T.-D. Ngo, "Dynamic social zone for human safety in human-robot shared workspaces," in *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, 2014, pp. 391–396.

[15] T. M. Ciolek and A. Kendon, "Environment and the spatial arrangement of conversational encounters," *Sociological Inquiry*, vol. 50, no. 3-4, pp. 237–271, 1980.

[16] A. Kendon, "Spacing and orientation in co-present interaction," in *Development of multimodal interfaces: Active listening and synchrony*. Springer, 2010, pp. 1–15.

[17] X.-T. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 4, pp. 1743–1760, 2017.

[18] F. Setti, C. Russell, C. Bassetti, and M. Cristani, "F-formation detection: Individuating free-standing conversational groups in images," *PloS one*, vol. 10, no. 5, p. e0123783, 2015.

[19] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, online collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1047–1054, 2017.

[20] A. Vega-Magro, R. Gondkar, L. Manso, and P. Núñez, "Towards efficient human-robot cooperation for socially-aware robot navigation in human-populated environments: the snape framework," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3169–3174.

[21] P. Teja 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)*, 2021, pp. 5817–5824.

[22] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *2008 11th international conference on information fusion*. IEEE, 2008, pp. 1–6.

[23] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 797–803.

[24] "BIWI walking pedestrians dataset," https://icu.ee.ethz.ch/research/datsets.html, accessed: 2023-07-04.

[25] A. Ess, B. Leibe, and L. Van Gool, "Depth and appearance for mobile scene analysis," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[26] H. Khambhaita and R. Alami, "Assessing the social criteria for human-robot collaborative navigation: A comparison of human-aware navigation planners," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017, pp. 1140–1145.

[27] J. v. d. Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.

[28] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350.

[29] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.