

Online Social Robot Navigation in Indoor, Large and Crowded Environments

Steven Silva¹, Nervo Verdezoto¹, Dennys Paillacho², Samuel Millan-Norman¹, and Juan David Hernández¹

Abstract—New robotics applications require robots to complete tasks in social spaces (i.e. environments shared with people), thus arising the necessity of enabling robots to operate in a socially acceptable manner. Some social spaces tend to be large and crowded (e.g. museums, shopping malls), which require robots to move around while showing appropriate social behaviors (e.g. not interfering with human’s comfortable areas). Moving under such conditions is generally called social robot navigation, and there are different approaches to do so. Nonetheless, current approaches are mostly limited to navigate large and outdoor spaces, where both robots and people can easily avoid each other. Other approaches have been tested in indoor environments, however, the test environments tend to be small and largely empty. In this paper, we present an online social robot navigation framework, which allow robots to navigate indoor, large and crowded environments, while showing social behaviors. Our framework consists of 3 modules: 1) world modeling that incorporates a novel Social Heatmap (SH) to represent crowded areas, 2) multilayered path planning that uses sampling-based approaches, and 3) path following control. We extensively benchmark our approach against state-of-the-art approaches in challenging simulated scenarios, and we also demonstrate its feasibility with the Pepper robot in real-world trials.

I. INTRODUCTION

In the last decade there has been a significant increase in the number of applications in which robots should operate in environments shared with people (i.e. social spaces) [1], [2]. Application examples include transporting medications around hospitals [3] and guiding people with visual impairments [4]. While navigating social spaces, it is important for robots not only to preserve human physical safety, but also to move in a socially acceptable manner, e.g. respecting the human’s comfortable spaces [5], [6]. Social spaces are highly dynamic, thus requiring robots to be robust and adaptable.

In contrast to robot navigation in non-social settings (e.g. underwater environments [7], warehouses [8]), Social Robot Navigation (SRN) considers both non-social obstacles, and social agents (i.e. people) and their comfort, naturalness and sociability [9]. There are several approaches to solve the SRN problem that are based on: Artificial Potential Field (APF) [10], [11], Deep Reinforcement Learning (DRL) [12],

The authors would like to acknowledge the support provided by Cardiff University’s Centre for Artificial Intelligence, Robotics and Human-Machine Systems (IROHMS) operation C82092, part-funded by the European Regional Development Fund (ERDF) through the Welsh Government.

¹S. Silva, N. Verdezoto, S. Millan-Norman, and J.D. Hernández are with IROHMS in the School of Computer Science & Informatics at Cardiff University, UK. {SilvaS1, VerdezotoDiasN, Millan-NormanS, HernandezVegaJ}@cardiff.ac.uk

²D. Paillacho is with the Research Development and Innovation Center of Computer Systems - CIDIS at ESPOL Polytechnic University, Ecuador. dpaillac@espol.edu.ec

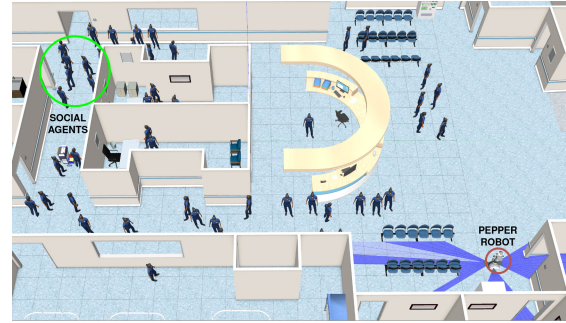


Fig. 1: The Pepper robot navigates a crowded hospital-like environment simulated with Gazebo and PedsimROS.

[13], as well as search-based and sampling-based techniques [14], [15], [16].

Some of these techniques present limitations in indoor crowded environments. APF and DRL based approaches could lead to situations in which the robot can get stuck. In the case of DRL, this issue occurs when a social agent is in the middle of the robot path to the destination [17]. In APF, this situation is generated when the force gradient that guides the robot results in local minima that do not correspond to the desired destination [18], [19]. Search and sampling-based approaches have proved to be effective in SRN, however they have only been tested in small, static, and relatively empty indoor environments [20], [21].

To address the aforementioned limitations, we proposed an online social robot navigation framework for indoor and office-like scenarios [16]. However our previous work was limited to small environments and with a low number of social agents. In order to overcome these limitations, we present an enhanced framework that is capable of solving start-to-goal robot navigation queries in large environments with a high number of dynamic and static social agents. The enhanced framework is composed of three modules: 1) a world modeling module that provides information of non-social obstacles and social agents, while also keeping record of crowded regions, 2) a multilayered path planning module that generates collision-free and socially admissible paths, and 3) a path following control module.

Our work has multiple contributions. First, we present an online social robot navigation framework with a multilayered planning strategy capable of solving start-to-goal queries in large and highly dynamic social spaces. Second, we propose the use of a Social Heatmap (SH) that represents people density in the surroundings, thus leading the robot through less crowded areas. Lastly, we extensively benchmark our enhanced framework with other state-of-the-art approaches, and also demonstrate its feasibility with the Pepper robot.

II. RELATED WORK

This section presents an overview of both the most relevant approaches for SRN, and the different metrics that are commonly used to evaluate such approaches.

A. Social Robot Navigation Approaches

1) *Artificial Potential Field*: a widely known and commonly extended approach is the Social Force Model (SFM) [22]. This model aims to approximate human movement behavior by calculating repulsive forces from undesired elements (e.g. non-social obstacles and people) and attractive forces to desired elements (e.g. goal destination) [23]. SFM is widely used because it can be extended to different scenarios by adding forces that enable additional robot behaviors (e.g. following a person) [24].

Kivrak et al. proposed an Extended Social Force Model (ESFM), in which an additional repulsive force is computed based on a collision prediction with social agents to increase people's safety [25]. Their approach is combined along with a multilevel mapping technique, which reduces the noise and resolution of the grid map, and increases the performance in calculating the repulsive forces from obstacles.

Ferrer et al. also proposed an ESFM that includes an additional force so that the robot is able to move next to a person in an outdoor environment [26]. This additional force attempts collision prediction by considering the movement of the person to be followed and people in the surroundings.

Although the aforementioned approaches have been tested in simulation and real life, most of the tests were limited to outdoor environments. Additionally, APF-based approaches require heuristic parameter tuning, which in some cases may lead to undesired robot behavior (e.g. jerky movement) [24]. Moreover, these methods can also suffer from local minima and robot stuck situations [27].

2) *Deep Reinforcement Learning*: Chen et al. presented DRL-based approaches [13], [28], which consider socially aware behaviors. The DRL policy is trained to minimize the time to move the robot to the desired goal, while avoiding collisions with nearby people in the environment.

Pérez-D'Arpino et al. proposed a DRL approach, which trains a policy that is capable of moving a robot in indoor constrained environments with the presence of dynamic social agents [29]. They combine their approach with a sampling-based global planner that guides the trained robot to the destination. While their approach proved to be effective, they do not consider the effect of static social agents in the global planner, which can lead to get the robot stuck. In fact, DRL as well as APF approaches' effectiveness is highly dependent on the specific global planner.

3) *Search-based and Sampling-based*: Patompak et al. presented an approach that utilizes a Transition-based RRT (T-RRT) [30], in which a Social Relationship Model (SRM) is used as a cost function to generate socially acceptable paths [20]. The SRM models the social agents' personal comfortable distance by using a bivariate Gaussian distribution function, which depends on gender and social agents' kinship. Whilst their approach generates feasible and collision-

free paths, they do not compare against other approaches and their test environments only include static social agents.

Ngo et al. use Dynamic Social Zone (DSZ) [31] and implement a human interaction detection module together with a personal space module [15]. Their approach generates socially acceptable paths by using Dynamic Window Approach (DWA) [32] with a personal space model as a cost optimization function. In spite of showing positive results, they do not compare against other approaches, and the test environment is static and with a low number of social agents.

To overcome some of the mentioned SRN challenges, we presented an approach based on an RRT* [33] that included several strategies that help a robot navigate dynamic and partially known social spaces [16]. We proposed a new strategy called *social relevance validity checking* that helps the robot focus on the social agents that might affect its near future movement. Although, our approach showed to be effective, it struggles to find solution paths with limited time-budget in larger social spaces with high crowd density.

B. Social Robot Navigation Metrics

There are several metrics that can be used to estimate human acceptance, this can be related to different aspects that go from path quality to human psychological safety [34], [2]. Below we present some commonly used metrics:

- *Time to solve query* is the time that takes for a robot to move from a starting point to a final point [35].
- *Total number of collisions* represents the total amount of collisions that a robot has with other people or objects. It can be represented as the amount of times a robot has collisions in a single navigation query [36].
- *Social Individual Index (SII)* expresses the proxemic physical safety from a robot to people in the environment (i.e. how much a robot invades the personal space of a person) [37]. The higher the value, the more uncomfortable a person is, after a certain threshold collision is assured.
- *Relative Motion Index (RMI)* expresses physical safety and probability of collision according to the velocities of both the person and the robot [37]. The higher the value, higher is the probability that a robot will collide with a person.

III. PROBLEM DEFINITION

Robot navigation refers to the capability of a robot to safely move through its surroundings, i.e. without colliding with obstacles. To do so, a robot must be able to map its surroundings, localize itself in the generated map, while finding safe and feasible paths to the desired destinations [38].

SRN is therefore a more specific case in which a robot moves in a social space [39]. In terms of mapping, SRN requires the robot to consider nearby people as social entities that interact with each other and with the environment. In terms of robot path planning, SRN requires the robot to consider social agents and social interactions to generate paths, which do not affect or interfere human social behaviors.

A. Definitions

Definition 1: As it was discussed in our previous work [16], a social agent is a social entity (i.e. a person) capable of expressing social behaviors and social cues, e.g. discomfort against objects, and social interactions.

Definition 2: the configuration space (\mathcal{C}) corresponds to the set of all the possible robot configurations (i.e. position and orientations). \mathcal{C} is therefore divided into the space free of obstacles (\mathcal{C}_{free}) and obstacle space (\mathcal{C}_{obs}), i.e. $\mathcal{C} = \mathcal{C}_{free} \cup \mathcal{C}_{obs}$ [40]. For start-to-goal social robot path planning problem, we extend the \mathcal{C}_{obs} to consider social agents, i.e. $\mathcal{C}_{obs} = \mathcal{C}_{common_obs} \cup \mathcal{C}_{social_agents}$ [16].

B. Social Robot Path Planning

In social robot path planning the objective is to find a collision-free, feasible and socially acceptable path that connects a start configuration (q_{start}) and a goal configuration (q_{goal}). While there are different aspects to consider when planning socially acceptable paths, this work considers a discomfort function $U(q_{start}, q_{goal}) \rightarrow \mathfrak{R}$ that represents the discomfort of the nearby agents [16]. Then the optimal solution path P^* is given by:

$$P^*(q_{start}, q_{goal}) = \underset{P(q_{start}, q_{goal}) \in \mathcal{C}_{free}}{\arg \min} U(q_{start}, q_{goal}) \quad (1)$$

IV. ONLINE SOCIAL ROBOT NAVIGATION FRAMEWORK

In this work, we aim to enable robots to navigate large and crowded social spaces (e.g. hospitals, museums, airports). We propose an enhanced framework that consists of three functional modules (see Fig. 2): 1) world modeling, 2) multilayered path planning, and 3) path following control.

A. World Modeling

This module is in charge of building a representation of the robot's surroundings. In SRN scenarios, such a representation needs to consider non-social obstacles (e.g. walls, desks, chairs, etc.) and social agents separately. In the case of non-social obstacles, the world modeling module generates a volumetric representation that is used for collision-checking purposes. In this work, such a volumetric representation is obtained with an Octomap [41], which is constructed with range data that can be provided by different perception sensors such as LiDARs and depth cameras.

Social agents, on the other hand, are considered both individually and collectively. In both cases, the state (i.e. position, orientation and velocity) of each agent is required. Such a state information can be obtained from a people-detector-tracker system, e.g. computer-vision-based systems that recognize and track people. Section V will provide implementation details on the specific approach used for the experimental validation in this work.

When social agents are considered individually, the state of each agent is directly used for collision-checking purposes. This module also considers social agents collectively by providing an estimation of those regions in the surroundings that are crowded (see Fig. 3). Such an estimation is represented

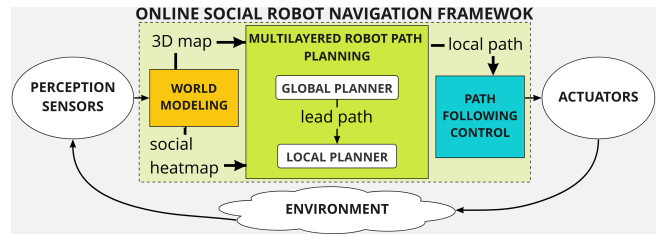


Fig. 2: Proposed framework for online social robot navigation and its three functional modules: world modeling, multilayered path planning and path following control.

by a Social Heatmap (SH), which is an occupancy grid map [42], in which each cell of the map has a value that indicates the density of people, and high values correspond to crowded areas. Such values are calculated as:

$$cell_{val} = \sum_{i=0}^L persistence_i * EPSM(state_i) \leq 100 \quad (2)$$

where L would correspond to the number of social agents that would have the cell within their comfort personal space [43], $state_i$ is the i_{th} social agent's state (position, orientation and velocity) that is used to calculate the Extended Personal Space Model (EPSM) [37], and $persistence_i$ relates to the last time an agent has been observed, and it calculated as:

$$persistence_i = 100 * exp^{-A*(t_i - t_0)} \quad (3)$$

where t_i is the current time, t_0 is the first time in which the social agent was observed and A is a time decay factor.

To create the SH (see Alg. 1), this module uses a list of *social_agents* that is periodically updated with the list of *reported_agents* provided by a people-detector-tracker (line 3). Every reported agent's time is reset (line 4) and is assigned with a *persistence* value of 100 (line 5). If such an agent had not been observed before, it is added to the list of *social_agents* (line 6). The updated list of *social_agents* is then used to populate the SH (lines 8-14). In doing so, if an existing social agent is not reported, their persistence is reduced as shown in (3) (line 12). It is worth noting that if the agent's persistence is equivalent to zero, the agent is removed from the list of *social_agents* (line 9).

B. Multilayered Robot Path Planning

We propose a multilayered approach (see Alg. 2) that includes: 1) an upper layer that plans a lead (*global*) path from the robot's current position to a final destination; and 2) a lower layer that plans a *local* path to an intermediate goal along the lead path. For both layers, we propose to use sampling-based planners that explore the environment by extending a tree of valid robot configurations.

1) *The global planner*: uses a tree-based strategy that builds on the RRT* planner [33] to rapidly generate a lead path by using a 10% of the total allocated planning time. To do so, this planner only checks non-social obstacles for collision. While generating a first approximation of the lead path, i.e. one that is close to be socially acceptable, the global

Algorithm 1: GenerateSocialHeatmap(Input)

Input:
reported_agents[]): List of detected social agents
social_agents[]): List of social agents
Output:
SH: Grid map with social agents occupancy.

```
1 begin
2   SH ← 2DGridMap()
3   for agent in reported_agents do
4     agent.resetTime()
5     agent.setPersistence(100)
6     if agent not in social_agents then
7       social_agents.addAgent(agent)
8   for agent in social_agents do
9     if agent.getPersistence() ≤ 0 then
10      social_agents.pop(agent)
11      continue
12    if agent not in reported_agents then
13      agent.reducePersistence()
14  SH.computeGrid(agent)
15  return SH
```

planner uses an optimization cost function that combines the path length and people density as follows:

$$GlobalCost(q) = \int_0^q SH(q) dq \quad (4)$$

where $GlobalCost(q)$ corresponds to the cost of a robot configuration q when searching for a global path, and it is calculated as the integral of the people density $SH(q)$ with respect to the traveled distance. By using (4), the calculated lead path will try to avoid crowded areas whenever possible, thus minimizing social discomfort.

The global planner uses two strategies to improve its performance for online incremental planning in partially known environments. The *opportunistic collision checking*, which allows the planner to assume configurations within unexplored regions to be collision-free. The *reuse of last best known solution*, which allows the planner to take the previous known solution as a starting point of each planning cycle. We had proved the utility of both strategies in robot navigation in non-social settings [7], and in this work we further demonstrate their use in SRN.

2) *The local planner*: also uses a tree-based strategy that builds on the Informed RRT* [44], and employs the remaining 90% of the allocated planning time to find collision-free and socially acceptable local paths within a shorter planning horizon. To do so, the local planner considers both non-social obstacles and social agents, while planning paths to intermediate goals, which are extracted from the lead path generated by the global planner.

To generate socially acceptable paths, the local planner uses a social cost optimization function that is defined as

Algorithm 2: Start2GoalOverallPathPlanning(q_{start}, q_{goal})

Input:
 q_{start} : Start configuration.
 q_{goal} : Goal configuration.
Output:
 $local_p^*$: An array of configuration waypoints.

```
1 begin
2   gp ← GlobalPlanner(SE2StateSpace)
3   lp ← LocalPlanner(SE2StateSpace)
4   last_best_known_solution ← {}
5   q_new_start ← q_start
6   while not stop_condition do
7     wm ← reqUpdatedWorldModel()
8     gp.updateWorldModel(world_model)
9     lp.updateWorldModel(world_model)
10    gp.startFrom(last_best_known_solution)
11    gp.solve(q_new_start q_goal)
12    if global_solution_not_found then
13      local_p* ← getPartialSolution()
14    else
15      g_p* ← gp.getSolution()
16      last_best_known_solution ← g_p*
17      l_qgoals ← getIntermediateG(g_p*)
18      lp.solve(l_qgoals, g_p*)
19      if local_solution_not_found then
20        local_p* ←
21          getPartialSolution()
22      else
23        local_p* ← lp.getSolution()
24  return local_p*
```

follows:

$$LocalCost(q) = \int_0^q (SH(q) + U(q)) dq \quad (5)$$

where $LocalCost(q)$ corresponds to the cost of a configuration q when searching for a local path, and it is calculated as an integral of the sum of the people density $SH(q)$ and the social discomfort $U(q)$ with respect to the traveled distance. In this work, we calculate the social discomfort as the maximum discomfort value obtained from each social agent using EPSM, as it is shown in (6).

$$U(q) = \max(EP\text{SM}_i(q)) \quad (6)$$

The local planner also *opportunistically checks* for collision, and *reuses* the lead path as the last best known solution. Furthermore, this planner also uses the *social relevance validity checking* strategy to identify those social agents that are relevant to the robot's near future movement.

3) *Solving social start-to-goal robot navigation queries*: is done as shown in Alg. 2. The framework first updates

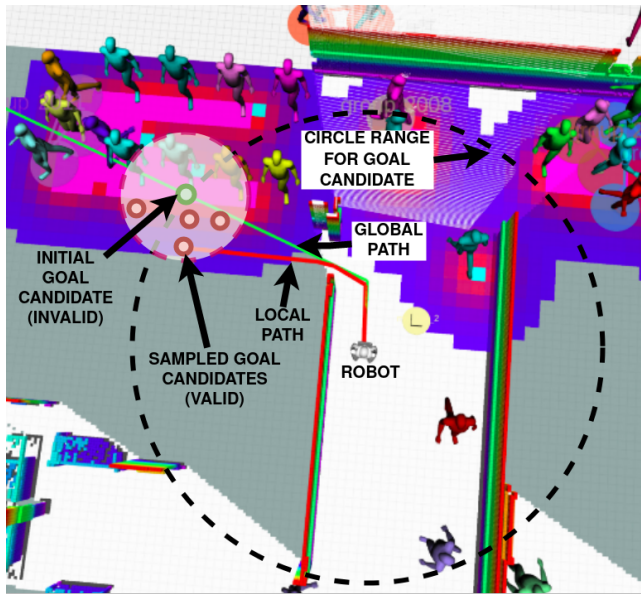


Fig. 3: Sampling valid intermediate goal candidates. Social Heatmap is also observed in blue and red colored voxels.

the world model (line 7-9). The global planner reuses the last best known solution (line 10) when searching a new solution path (line 11). If a new lead path is not found, the planning module sends a partial solution that is based on the last local path (line 12). This partial path is the section of the last local path that is still valid (i.e. is not under collision and is socially acceptable). Once a new lead path is found, it is used to define the intermediate goal of the local planner (line 17). This intermediate goal is extracted from the lead path at a predefined distance (e.g. the robot’s maximum perception range) (see Fig. 3). If such an intermediate goal is not valid (i.e. in collision with a social agent), a set of alternative valid goals are obtained by sampling the vicinity of the original intermediate goal. To find the local path, the local planner receives a list of intermediate goals and the lead path (line 18). As it occurs with the global planner, if a local path is not found (line 20), the path planning module sends a partial solution that is based on the last local path.

C. Path Following Control

This module receives and follows the path from the local planner. While alternatives exist to implement such a controller, this work discretizes the local path and uses a waypoint follower controller.

V. EXPERIMENTS AND RESULTS

To validate our proposed framework, we tested and compared it with other state-of-the-art approaches.

A. Experimental Setup

We conducted an extensive simulation-based evaluation and benchmarking of our framework using Gazebo [45] and the Robot Operating System (ROS) [46]. Our simulation test environment consisted of a hospital-like setting with an approximate area of $1375m^2$, in which a simulated Pepper robot [47] moves around a high number of static

and dynamic social agents (> 70) (see Fig. 1). The Pepper robot was equipped with a depth camera to detect non-social obstacles. To simulate the behavior of social agents, we used *pedsimROS* [48], which is a ROS package to interface with the *pedsim* library that models walking pedestrians by using the SFM. As this work is mostly focused on the planning aspects of SRN, we made different assumptions and simplifications such as: 1) a perfect robot localization, and 2) the use of *pedsimROS* output and an indoor positioning system as the people-detector-tracker for simulation and real-world tests, respectively.

B. Benchmark Scenarios

We designed two different test scenarios in which the Pepper robot had to move from the rear entrance of the hospital to the opposite side of the hospital, while navigating around non-social obstacles, and static and dynamic social agents. Those dynamic social agents follow several random waypoints. It is important to note that social agents react to the presence of the robot similar to another social agent.

We compare our enhanced framework with our previous work [16], the Proactive Social Motion Model (PSMM) approach [37], the SFM [22] and socially aware collision avoidance with deep reinforcement learning (SA-CADRL) [13]. Since PSMM, SFM and SA-CADRL apply short planning horizons, they all require collision-free waypoints as an input. As this work aims to enable robots to navigate large environments, we manually defined such waypoints to reach the predefined distant destinations.

As our proposed approach has a stochastic behavior, a single execution of a given test does not provide a statistical valid conclusion. Therefore, we conducted 100 tests for each of the compared methods in each of the test scenarios. Furthermore, in each of those attempts, some behavioural aspects of the social agents were randomly varied, e.g. the location of static agents and the order of the waypoints that dynamic agents followed. All the experiments were assessed with different metrics (e.g. SII and RMI) providing insights in terms of both the performance and the social safety capabilities of the tested approaches. Finally, to better understand the experiments, the reader is referred to: <https://youtu.be/Gq149wFVj7A>.

1) *Scenario 1: Large and Crowded Indoor Environments.* In this scenario, the Pepper robot had to navigate across the hospital, in which 78 social agents (63 dynamic and 15 static) were located in different areas of the hospital. As it can be observed in Table I, our enhanced framework shows the highest success rate, particularly when using the proposed Social Heatmap (SH). Although the averages of SII and RMI for our approach are higher (worse) than the ones obtained with PSMM, such a comparison has to be analyzed carefully, since PSMM also had the lowest success rate (around 9% out of 100 runs). This latter probably means that PSMM is a very conservative approach that tries to enforce a distance to any obstacle, thus preventing the robot to reach its destination when navigating in crowded areas.

TABLE I: Benchmark of proposed framework and other approaches for Scenarios 1 and 2 (SD = Standard Deviation).

| | Scenario 1: Large and Crowded Indoor Environments | | | | | | Scenario 2: Queues of Social Agents | | | | | |
|------------------------|---|---------------|---------------|----------------------|-------------------|-------------------|-------------------------------------|---------------|---------------|----------------------|-------------------|-------------------|
| | Success Rate | SII Mean (SD) | RMI Mean (SD) | Total Time Mean (SD) | # Coll. Mean (SD) | # Nodes Mean (SD) | Success Rate | SII Mean (SD) | RMI Mean (SD) | Total Time Mean (SD) | # Coll. Mean (SD) | # Nodes Mean (SD) |
| New Fram. (with SH) | 92% | 0.074 (0.025) | 1.182 (0.119) | 342.1 (45.4) | 0.696 (1.066) | 1419 (201) | 91% | 0.095 (0.020) | 1.297 (0.095) | 355.4 (58.7) | 0.714 (1.068) | 1010 (119) |
| New Fram. (without SH) | 83% | 0.065 (0.020) | 1.150 (0.098) | 309.5 (41.3) | 0.566 (0.736) | 1073 (145) | 70% | 0.094 (0.021) | 1.293 (0.095) | 304.5 (44.1) | 0.643 (0.762) | 873 (123) |
| Prev. Fram. | 73% | 0.076 (0.012) | 1.197 (0.055) | 317.5 (37.5) | 1.192 (1.243) | 945 (77) | 81% | 0.114 (0.017) | 1.400 (0.072) | 355.4 (61.5) | 1.148 (1.205) | 873 (73) |
| PSMM | 9% | 0.033 (0.035) | 0.589 (0.523) | 357.0 (157.6) | 1.250 (1.165) | - | 2% | 0.070 (0.028) | 1.140 (0.198) | 601.5 (12.0) | 1.000 (1.414) | - |
| SFM | 47% | 0.096 (0.040) | 1.277 (0.206) | 354.0 (71.2) | 2.848 (2.231) | - | 26% | 0.081 (0.025) | 1.155 (0.145) | 419.9 (113.4) | 1.808 (1.201) | - |
| SA-CADRL | 40% | 0.132 (0.092) | 1.400 (0.408) | 345.3 (76.1) | 2.650 (2.032) | - | 44% | 0.123 (0.072) | 1.364 (0.317) | 326.7 (62.2) | 2.419 (1.577) | - |

In terms of the total time required to complete the navigation task, it can be observed that our previous work and our new framework can complete the task in less time than the other approaches. Likewise, our approaches, particularly the one presented in this work, has a very low average of number of collisions (see Table I).

Another relevant analysis can be done when comparing our enhanced framework with our previous framework. In such a comparison our enhanced framework has similar values for SII and RMI, but the success rate has an important improvement, passing from 73% to 92%. An explanation to this improvement can be found in the proposed multilayered planning strategy, which allows the enhanced framework to dedicate more resources when planning local paths around the lead path. This can be clearly observed in the number of nodes of the tree-based planners, which is higher for the enhanced framework (1419 vs. 945) and means the planner was able to better explore the environment while calculating a solution path (see Table I).

2) *Scenario 2: Queues of Social Agents*. In this scenario, the Pepper robot had to navigate the same hospital environment, but this time 35 out of 96 social agents were static and divided in 6 groups that form queues in different corridors of the environment. This scenario attempts to emulate common situations in social spaces, where robots might have to navigate across queues of social agents, thus leading to challenging situations that resemble narrow passages.

As it can be observed in Table I, in this scenario our enhanced framework also has the highest success rate (91%), and the SII and RMI values are lower than the ones obtained with our previous work. Similar to Scenario 1, PSMM shows better SRN metrics, but with a very low success rate. Finally, apart from corroborating the results obtained in Scenario 1, the results of Scenario 2 demonstrates the ability to navigate challenging situations, such as queues of social agents.

C. Real-World Trials

We tested our proposed framework with the real-world Pepper robot in an indoor laboratory setting, in which the robot was equipped with a LiDAR to self-localize by using Cartographer [49], an Intel Realsense D435i to detect non-social obstacles, and a Pozyx indoor positioning system [50]

to track social agents. The robot was able to navigate the environment while avoiding social agents, hence proving the feasibility of our approach in a real-world system.

VI. DISCUSSION AND CONCLUSIONS

This paper presented an enhanced online social robot navigation framework, which enables a robot to navigate large and crowded spaces in a socially acceptable manner. The framework is composed of three modules: world modeling, multilayered path planning, and path following control. The world modeling module includes a novel Social Heatmap (SH), which attempts to represent the crowded areas in the environment. The SH proved to be instrumental to allow the multilayered planning to better dedicate its computation resources. In the multilayered planning module, the local planner implements a social cost optimization function that models the comfort space around the social agents. That module also combines different strategies like the *opportunistic validity checking*, *reuse of last best known solution* and *social relevance validity checking* to help the robot navigate in large crowded environments.

We tested and compared our enhanced framework against other state-of-the-art approaches such as SA-CADRL, SFM and PSMM. The results proved that our new approach has the highest success rate. Although, some of the tested approaches showed better SRN metrics, the success rate for such approaches is significantly lower. In contrast to our enhanced framework, the APF-based approaches (e.g. SFM) require tuning heuristic parameters, while DRL-based approaches are highly limited to be effective in those situations that are similar to the training sets [6]. We also demonstrated our framework's feasibility with the Pepper robot.

In the future, we plan to explore hybrid approaches for path planning that combine learning-based methods and sampling-based planners. Such a hybrid approach will aim to learn and help overcome situations that are commonly difficult in SRN. Furthermore, we would like to add complex human-group and human-object interactions to further test our new framework. Last, we would like to conduct real-world experiments in large and crowded environments, while also conducting the corresponding user studies to analyze the social acceptability of the robot in human-shared spaces.

REFERENCES

- [1] A. Vega, R. Cintas, L. J. Manso, P. Bustos, and P. Núñez, "Socially-Accepted Path Planning for Robot Navigation Based on Social Interaction Spaces," in *Robot 2019: Fourth Iberian Robotics Conference*, M. F. Silva, J. Luís Lima, L. P. Reis, A. Sanfeliu, and D. Tardioli, Eds. Cham: Springer International Publishing, 2020, pp. 644–655.
- [2] Y. Gao and C.-M. Huang, "Evaluation of Socially-Aware Robot Navigation," *Frontiers in Robotics and AI*, vol. 8, no. October 2020, pp. 1–21, 2022.
- [3] R. Bloss, "Mobile hospital robots cure numerous logistic needs," *Industrial Robot*, vol. 38, pp. 567–571, 2011.
- [4] J. Guerreiro, D. Sato, S. Asakawa, H. Dong, K. M. Kitani, and C. Asakawa, "Cabot: Designing and evaluating an autonomous navigation robot for blind people." Association for Computing Machinery, Inc, 10 2019, pp. 68–82.
- [5] A. Bera, T. Randhavane, R. Prinja, and D. Manocha, "SocioSense: Robot navigation amongst pedestrians with social and psychological constraints," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-September, no. c, pp. 7018–7025, 2017.
- [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. 1, no. 1, 2021. [Online]. Available: <http://arxiv.org/abs/2103.05668>
- [7] J. D. Hernández, M. Moll, E. Vidal, M. Carreras, and L. E. Kavraki, "Planning feasible and safe paths online for autonomous underwater vehicles in unknown environments," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 1313–1320, 2016.
- [8] M. Gadd and P. Newman, "A framework for infrastructure-free warehouse navigation," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 3271–3278, 2015.
- [9] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, 2013.
- [10] M. Prédhumeau, J. Dugdale, and A. Spalanzani, "Adapting the social force model for low density crowds in open environments," *Julie Dugdale*, vol. 1, pp. 1–14, 2019. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-02194735>
- [11] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505.
- [12] R. Guldenring, M. Gerner, N. Hendrich, N. J. Jacobsen, and J. Zhang, "Learning local planners for human-aware navigation in indoor environments," *IEEE International Conference on Intelligent Robots and Systems*, pp. 6053–6060, 2020.
- [13] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 1343–1350, 2017.
- [14] V. H. Nguyen, V. B. Hoang, C. A. My, L. M. Kien, and X. T. Truong, "Toward socially aware trajectory planning system for autonomous mobile robots in complex environments," *Proceedings - 2020 7th NAFOSTED Conference on Information and Computer Science, NICS 2020*, pp. 90–95, 2020.
- [15] H. Q. T. Ngo, V. N. Le, V. D. N. Thien, T. P. Nguyen, and H. Nguyen, "Develop the socially human-aware navigation system using dynamic window approach and optimize cost function for autonomous medical robot," *Advances in Mechanical Engineering*, vol. 12, pp. 1–17, 2020.
- [16] S. Silva, D. Paillacho, N. Verdezoto, and J. D. Hernández, "Towards online socially acceptable robot navigation," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022, pp. 707–714.
- [17] L. Kastner, T. Buiyan, L. Jiao, T. A. Le, X. Zhao, Z. Shen, and J. Lambrecht, "Arena-rosvan: Towards deployment of deep-reinforcement-learning-based obstacle avoidance into conventional autonomous navigation systems," *IEEE International Conference on Intelligent Robots and Systems*, pp. 6456–6463, 2021.
- [18] J. Barraquand and J. C. Latombe, "Robot Motion Planning: A Distributed Representation Approach," *The International Journal of Robotics Research*, vol. 10, no. 6, pp. 628–649, 1991.
- [19] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
- [20] P. Patompak, S. Jeong, N. Y. Chong, and I. Nilkhamhang, "Mobile robot navigation for human-robot social interaction," *International Conference on Control, Automation and Systems*, vol. 0, pp. 1298–1303, 1 2016.
- [21] P. T. Wu, C. A. Yu, S. H. Chan, M. L. Chiang, and L. C. Fu, "Multi-layer environmental affordance map for robust indoor localization, event detection and social friendly navigation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2945–2950, 2019.
- [22] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, pp. 4282–4286, 1995.
- [23] H. Kivrak, F. Cakmak, H. Kose, and S. Yavuz, "Waypoint based path planner for socially aware robot navigation," *Cluster Computing*, 2022. [Online]. Available: <https://doi.org/10.1007/s10586-021-03479-x>
- [24] Y. Chen, C. Liu, B. E. Shi, and M. Liu, "Robot Navigation in Crowds by Graph Convolutional Networks with Attention Learned from Human Gaze," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2754–2761, 2020.
- [25] H. Kivrak, F. Cakmak, H. Kose, and S. Yavuz, "Social navigation framework for assistive robots in human inhabited unknown environments," *Engineering Science and Technology, an International Journal*, no. xxxx, 2020. [Online]. Available: <https://doi.org/10.1016/j.jjestch.2020.08.008>
- [26] G. Ferrer, A. G. Zulueta, F. H. Cotarelo, and A. Sanfeliu, "Robot social-aware navigation framework to accompany people walking side-by-side," *Autonomous Robots*, 2017.
- [27] H. T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia, "Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 2347–2354, 2015.
- [28] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 285–292, 2017.
- [29] C. Pérez-D'Arpino, C. Liu, P. Goebel, R. Martín-Martín, and S. Savarese, "Robot Navigation in Constrained Pedestrian Environments using Reinforcement Learning," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2021-May, no. Iera, pp. 1140–1146, 2021.
- [30] L. Jaillet, J. Cortés, and T. Siméon, "Transition-based RRT for path planning in continuous cost spaces," *International Symposium on Combinatorial Search, SoCS 2008*, no. Section V, pp. 22–26, 2008.
- [31] X. T. Truong and T. D. Ngo, "Dynamic social zone based mobile robot navigation for human comfortable safety in social environments," *International Journal of Social Robotics*, vol. 8, pp. 663–684, 2016.
- [32] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [33] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [34] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni, "Soc-NavBench: A Grounded Simulation Testing Framework for Evaluating Social Navigation," *ACM Transactions on Human-Robot Interaction*, 2022.
- [35] A. K. Reddy, V. Malviya, and R. Kala, "Social cues in the autonomous navigation of indoor mobile robots," *International Journal of Social Robotics*, vol. 13, pp. 1335–1358, 9 2021. [Online]. Available: <http://link.springer.com/10.1007/s12369-020-00721-1https://link.springer.com/10.1007/s12369-020-00721-1>
- [36] J. Guzzi, A. Giusti, L. M. Gambardella, G. Theraulaz, and G. A. Di Caro, "Human-friendly robot navigation in dynamic environments," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2013, pp. 423–430. [Online]. Available: <http://ieeexplore.ieee.org/document/6630610/>
- [37] 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, pp. 1743–1760, 10 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/8011466/>
- [38] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, and C. C. Peng, "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," 9 2018.
- [39] 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, apr 2015. [Online]. Available: <http://link.springer.com/10.1007/s12369-014-0251-1>
- [40] T. Lozano-Pérez, “Spatial Planning: A Configuration Space Approach,” *IEEE Transactions on Computers*, vol. C-32, no. 2, pp. 108–120, 1983.
- [41] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [42] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 116–121.
- [43] E. Hall, *The hidden dimension: man’s use of space in public and private*. London, UK: The Bodley Head Ltd, 1966.
- [44] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Informed rrt: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2997–3004.
- [45] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.
- [46] Stanford Artificial Intelligence Laboratory et al., “Robotic operating system.” [Online]. Available: <https://www.ros.org>
- [47] SoftBank. Softbank mobile and aldebaran unveil “pepper” – the world’s first personal robot that reads emotions. [Online]. Available: https://www.softbank.jp/en/corp/group/sbm/news/press/2014/20140605_01/
- [48] B. Oka and K. O. Arras, “Towards group-level social activity recognition for mobile robots,” in *Assistance Serv. Robot. Hum. Environ. Workshop at IROS*, 2014.
- [49] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2d lidar slam,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1271–1278.
- [50] Pozyx. Enterprise kit. [Online]. Available: <https://store.pozyx.io/shop/enterprise-kit-238#attr=93,103,156,180>