Socially Reactive Navigation Models for Mobile Robots

Francisco Melo
Instituto Superior Técnico, Lisboa, Portugal
francisco.rapos.melo@tecnico.ulisboa.pt

Abstract—This work considers socially acceptable behaviors in traditional reactive navigation systems, allowing a robot to approach a group of humans in a socially acceptable manner by considering the personal space and the group space. In contrast to the fixed parameters of social distancing, this work presents an adaptive model; that is, the parameters of the personal and group space's cost functions adapt according to the arrangement of the group and space constraints, avoiding the choice of initial parameters. A socially-aware navigation system capable of approaching groups is implemented for a general-purpose mobile robot. The adaptive personal and group space algorithm is integrated with the standard navigation system of ROS, representing their information in a costmap layer. The adaptation of spaces is tested using fixed and adaptive parameters for different groups provided by three datasets. The navigation system is evaluated through simulation experiments, demonstrating that the robot is capable of approaching groups and, at the same time, provides a more realistic space modeling adapted to the context.

Index Terms—Human-robot interaction, social robot, socially-aware navigation, proxemics, approaching humans, adaptive space.

I. INTRODUCTION

With the evolution of robotics, robots have expanded from controlled environments to dynamic social spaces shared with humans. As robots have improved perceptual and social skills, it is easier for humans to have contact and interact with them. An essential social ability for a robot is to be proactive and approach a person or group of people with socially acceptable behavior and respecting people who are interacting. Approaching a group adds extra difficulty since it is also necessary to understand how humans behave in a group and share space. Thus, besides modeling humans’ personal space, the robot also needs to model the group space. This work aims to consider socially acceptable behaviors in traditional reactive navigation systems, allowing a robot to approach humans or groups of humans in a socially acceptable manner.

State of the art methods use fixed parameters to model the personal and group space’s shape and size. However, one must take into account that people in specific situations do not respect the personal space of other individuals. Personal space and group space parameters depend on the group’s members’ culture and relationship, influencing the distance of group members from each other, resulting in groups with members closer or further apart. Personal space and group space should be evaluated with the obstacles near the group to ensure these spaces do not overlap them. Thus, the personal space and group space parameters should adapt depending on the group arrangement and space constraints to make it feasible and natural for a robot to approach a group of humans and navigate around them, rather than being fixed. The socially reactive navigation system should detect the situations mentioned above and modify the personal and group space shape and size by adjusting the cost function that models them, avoiding the exclusive dependence on the choice of initial parameters in the modeling of spaces. To perform this detection, it must evaluate the individuals’ space by identifying obstacles near them and considering the group’s people’s pose (position and orientation).

The main contributions of this work are: (i) A novel adaptive modeling of space for individuals and groups of humans, that work for groups of large size while considering other humans and obstacles, (ii) Estimation of the approaching pose to a group using the adaptive spaces, (iii) Implementation of the adaptive modeling of spaces through costmap layers and implementation group approach algorithms in a navigation system using ROS, allowing the robot to navigate in an environment with groups and approach them.

The remainder of the document is structured as follows. In Section II, we present the related work. Section III presents the methodology and some preliminary results. Section IV defines how our approach is integrated into the socially reactive navigation system. It also describes the simulation experiments performed. Finally, Section V presents the conclusions and proposes future work.

II. RELATED WORK

A. Proxemics

In 1963, Edward T. Hall, in his first study on proxemics, introduced the concept of proxemics as the study of how humans use and manage space on interpersonal communication.

1) Personal Space: Hall proposed the division of the space around a human into four spaces: (i) Intimate Space (<0.45m), (ii) Personal Space (0.45 - 1.2m), (iii) Social Space (1.2 - 3.6m) and (iv) Public Space (>3.6m).

Subsequent studies proposed more complex shapes of personal space. noticed that people give more importance to their frontal space. Thus, personal space is then modeled as an egg shape, where the area is larger on the frontal side. models the personal space by a monotonic decreasing function with equipotential lines having the form of an ellipse directed in the direction of motion. defined an asymmetric personal space model.
space and divided it into two zones around a human: (i) In reach of the hand of the dominant side and (ii) Out of reach of the hand. The personal space is smaller on the pedestrian’s dominant side. [7] demonstrated through experiments that personal space is dynamic and situation-dependent, and it is a momentary spatial preference.

In this work, the concept of personal space will be based on a combination of the egg shape proposed in [4] and the idea of an adaptive Personal Space proposed in [7] to make it more realistic and adaptable.

2) Group of Individuals Space: A group can be defined as two or more individuals who are connected by a social relationship [8]. During many different social occasions, a type of group called free-standing conversational group (FCG) emerges when people spontaneously decide to be in each other’s immediate presence to interact with each other [9]. [10]. The spatial position and orientation of people describe an FCG; most know as Adam Kendon’s Facing Formation (F-Formation). In an F-Formation, people establish and maintain a convex space, where everybody in the formation has equal and direct access. In this work, when we refer to groups, we refer to the type of groups F-Formations.

An F-Formation is organized in three social spaces [9]: (i) The o-space is space surrounded by the group’s individuals, where they are oriented, and no external people are allowed to enter it, (ii) The p-space is the ring of space evolving the o-space where individuals of the F-Formations are placed and (iv) The r-space is the space that surrounds both the o-space and p-space, and the F-Formation members monitor it. It separates the group from the outside world and is where people leave or try to join the F-Formation.

B. Social Robots and Proxemics

When social robots plan to navigate, they need to consider humans’ comfort, preferences and needs [11] and social aspects of interaction with people [12]. Assuming that people will keep the same social space management conventions when they interact with robots as they do when interacting with each other, we can use the proxemics model presented in Section II-A, personal space and o-space, to improve robot’s sociality by incorporating proxemic models in their navigation system.

C. Socially Aware Robot Navigation - Literature Review

Socially reactive navigation approaches can be divided into: model-based and learning-based [13]. Model-based approaches extend traditional reactive navigation models taking into account humans and groups of humans proxemics. The tuning of the parameters is a problem because of variations from person to person, regarding culture, social situation, gender, age, personality, and physical appearance [14]. Nevertheless, there are already some works that explore the idea of an adaptive personal space for individuals. On the other hand, learning-based approaches emulate human behaviors, providing more realistic data. However, the drawback of these approaches is that they are data-dependent.

[15]–[18] proposed methods for a robot to approach groups. However, they only focused and presented results for small groups, two to three elements.

[19] proposed a unified framework for approaching pose prediction and socially aware robot navigation. This framework allows a mobile robot to safely and socially approach both stationary and dynamic humans and groups of interacting humans, including sitting or moving humans. That work is an extension of their previous work proposed in [14] and is the primary inspiration of our work. They define a dynamic social zone that results from the maximum of the extended personal space (EPS) and social interaction space (SIS). The EPS models the personal space, and the SIS models the interaction space. Both spaces are modeled using Gaussian functions. This framework can drive a robot to approach and avoid stationary and moving person/group of persons in a safe and socially acceptable way while keeping comfort to humans in both simulation and real world-experiments. Nevertheless, this framework has only been tested on small groups, up to 4 individuals in laboratory scenarios (i.e., groups were created from a predefined script and not in realistic scenarios). They also do not evaluate whether people do not follow proxemic rules and adjust their space due to space constraints.

[20] proposed a data-driven model that estimates the approaching pose a robot should use to join a group in a more human-like way. This estimation is based on real humans approaching others in real-world scenarios. The algorithm can estimate approaching poses for groups from 2 to 6 persons in simulations, taking into account the personal space and the o-space. In the real-world, a group of only two persons was tested. Nevertheless, this model is data-dependent as it depends on the number and the variety of samples in the training set and depends on a match on the annotated data. A navigation system that implements this model is also missing.

D. Adaptive Space

Some work has appeared in the literature that uses this concept of adaptive proxemic spaces.

Regarding context adaptation, [21] proposed a flexible spatial density model to automatically adapt the personal space to spatial context and human intention. They use a narrow corridor as a simulation environment where the personal space has to be adapted so that the robot can navigate. [22] presented an automated system that generates the most suitable personal space for any environmental condition. It generates personal space by considering the height, appearance and familiarity of an individual. More recently, [23] presented a proxemic framework to represent, with adaptive proxemics shapes, human activities, location, culture, or specific situations. They also introduced a new proxemic shape called the cooperation zone. This zone is located outside the intimate zone and inside the personal zone, allowing fluid and natural cooperation between humans and robots in navigation and interaction tasks.

Nevertheless, these works only consider the adaptation of a human’s space. There is a lack of research on adaptive proxemics when we have a group of people.

III. Adaptive Space

Our approach will be based on the work presented in [14], [19], but with a more comprehensive approach regarding space modeling.
A. Modeling of Space

The personal and group space will be represented as cost functions in a cost map, and one will use the Asymmetric Gaussian function proposed in [24] to model the personal space and a 2D Gaussian function to model the group space.

1) Personal Space and Group Space: A 2D Asymmetric Gaussian function is proposed in [24] by composing two 2D Gaussian functions with the same \( \sigma_x \) but different \( \sigma_y \) values. Thus, reducing the symmetry of the functions to only the x-axis. This function allows a high adaptability capacity, as it can vary its size and shape. The parameters and notation of the function are:

\[ f_{\text{AsymmetricGaussian}}(x,y) = \frac{1}{2\pi \sigma_x \sigma_y} \exp \left( -\frac{1}{2} \left( \frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2} \right) \right) \]

\( \theta \) orientation of the function;
\( \sigma_h \) standard deviation in the \( \theta_0 \) direction (front);
\( \sigma_s \) standard deviation in the \( \theta_0 \pm \pi/2 \) direction (sides);
\( \sigma_r \) standard deviation in the \( -\theta_0 \) direction (rear).

Using Algorithm 1, it is possible to compute the value of the Asymmetric Gaussian function for a given coordinate of a cell \((x,y)\). In line 10 of the algorithm a 2D Gaussian function with arbitrary rotation \( \theta_0 \) can be defined for a coordinate of a cell \((x,y)\) in the grid map \( M_{n,m} \) by composing two 2D Gaussian functions with the same \( \sigma_x \) but different \( \sigma_y \) values. Thus, reducing the symmetry of the functions to only the x-axis. This function allows a high adaptability capacity, as it can vary its size and shape. The parameters and notation of the function are:

\[ f_{\text{Gaussian}}(x,y) = \frac{1}{2\pi \sigma_x \sigma_y} \exp \left( -\frac{1}{2} \left( \frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2} \right) \right) \]

\( \theta \) orientation of the function;
\( \sigma_h \) standard deviation in the \( \theta_0 \) direction (front);
\( \sigma_s \) standard deviation in the \( \theta_0 \pm \pi/2 \) direction (sides);
\( \theta \) standard deviation in the \( -\theta_0 \) direction (rear).

Using Algorithm 1, it is possible to compute the value of the Asymmetric Gaussian function for a given cell coordinate \((x,y)\).

Algorithm 1: Algorithm to compute the Asymmetric Gaussian function at a given cell coordinate \((x,y)\) [24].

```python
function AsymmetricGaussian(x, y, x0, y0, \theta_0, A, \sigma_r, \sigma_h, \sigma_s) :
    \theta \leftarrow \text{atan} 2(y - y_0, x - x_0);
    \alpha \leftarrow \theta + \pi/2 - \theta_0;
    if \alpha < 0 then
        \sigma \leftarrow \sigma_r;
    else
        \sigma \leftarrow \sigma_h;
    end
    d \leftarrow \sqrt{(x - x_0)^2 + (y - y_0)^2};
    cost = A \exp \left(-\left(\frac{(d \cos(\theta - \theta_0))}{\sqrt{2\sigma}}\right)^2 + \left(\frac{d \sin(\theta - \theta_0)}{\sqrt{2\sigma}}\right)^2\right);
    return cost
end function
```

Considering a person \( p_i \) with position \((x_i^p, y_i^p)\) and orientation \( \theta_i^p \), the Asymmetric Gaussian function \( f_{\text{Gaussian}}^p(x,y) \) that models the personal space, with the maximum at the center \((x_i^p, y_i^p)\), can be defined for a coordinate of a cell \((x,y)\) in the grid map \( M_{n,m} \). The parameter set \([A^p, \sigma_r^p, \sigma_h^p, \sigma_s^p]\) are the adjusting parameters of the function \( f_{\text{Gaussian}}^p(x,y) \), where \( A^p \) is the selected amplitude, and \( \sigma_h^p, \sigma_s^p \) and \( \sigma_r^p \) are the respective standard deviations of the Asymmetric Gaussian function. The values of these parameters are going to affect the size and shape of personal space. In this work, the parameters are going to adapt according to the situation rather than fixed.

Considering a set of \( N \) humans detected in the scenario, the function \( F^p(x,y) \) that represents all the personal spaces of the persons \((f_1^p(x,y), \ldots, f_N^p(x,y))\) is computed as:

\[ F^p(x,y) = \max (f_1^p(x,y), \ldots, f_N^p(x,y)). \] (1)

In this work, the group space is represented by the o-space. One assumes that the human body dimensions can be represented by an \( x \) and \( y \) components. \( \text{human}_x \) and \( \text{human}_y \). Before describing the function that models the o-space, we clarify the three different types of radii considered in this work for groups: (i) The group radius is given by the average Euclidean distance of the group members to the center, (ii) The o-space radius is the distance of the person closest to the center of the group minus half of the human dimension in \( x \) and (iii) The p-space radius is the farthest person’s distance to the center of the group plus half of the human dimension in \( x \).

Thus, the limits of the social spaces can be defined in terms of radii as: (i) The o-space limit is defined from the center of the group to the o-space radius, (ii) The p-space limit is defined from the o-space radius to the p-space radius and (iii) The r-space limit is defined from the p-space radius.

The modeling of the group space will be based on the 2D Gaussian function. A group \( g_k \) can be defined by the set of parameters \( g_k = (x_k^g, y_k^g, r_k^g) \), where \( x_k^g, y_k^g \) is the center point of the o-space and \( r_k^g \) is the radius of the group. This interaction is modeled by a 2D Gaussian function \( f_{\text{Gaussian}}^g \), which is centered at the center of the o-space of the group \((x_k^g, y_k^g)\), and is represented by the parameters \([A^g, \sigma_r^g, \sigma_h^g, \sigma_s^g]\).

Considering a set of \( N \) groups detected in the scenario, the function \( F^g(x,y) \) that represents all the o-spaces of the groups \((f_1^g(x,y), \ldots, f_N^g(x,y))\) is computed as:

\[ F^g(x,y) = \max (f_1^g(x,y), \ldots, f_N^g(x,y)). \] (2)

Finally, the function that incorporates both the functions that represent all the personal spaces \( (F^p(x,y)) \) and all the o-spaces \( (F^g(x,y)) \) detected is computed as:

\[ F(x,y) = \max (F^p(x,y), F^g(x,y)). \] (3)

An example of the representation of the personal space using an Asymmetric Gaussian, and the functions defined in [1], [2] and [3] for a group of four individuals is represented in Figure 1.

B. Space Adaptation

Here, we present the main contribution of this work, how each individual’s personal space and the group space adapt depending on the spatial context, i.e., how we choose the parameters that model the spaces’ functions (standard deviations of the Gaussian functions). The adaptation is divided into two types: (i) Distance Adaptation and (ii) Obstacles Adaptation.

1) Distance Adaptation: This adaptation adjust the parameters based on the distance between the group members.

First, we model the personal spaces as ellipses centered at the person position and rotated with the person orientation, where the major axis is defined by a variable \( S_x \) and the minor
axis by a variable \( S_y \). Since our algorithm adapts the space regarding the initialization, one can use a maximum value of personal space dimensions as the initial value to cover all contexts and cultures. The initial parameters are represented by two constants called \( \text{pspace}_x \) and \( \text{pspace}_y \). Our assumption is that humans maintain the same shape of personal space, where the shape of the Gaussian is represented by the aspect ratio between \( S_x \) and \( S_y \). The parameters \( S_x \) and \( S_y \) are related by a factor \( \text{pfactor} \), to ensure the personal space keeps the elliptical shape when adapting. Their relation is computed by:

\[
S_y = \frac{S_x}{\text{pfactor}},
\]

where the \( \text{pfactor} \) is the relation between the personal space initial parameters:

\[
\text{pfactor} = \frac{\text{pspace}_x}{\text{pspace}_y}.
\]

Then, we adapt the parameters depending on the group configuration. If the group has two members with a \( \text{vis-a-vis} \) configuration, that is, the difference between the orientation of the members is approximately \( \pi \) the \( S_x \) parameters adapt to be half the Euclidean distance between the persons, and \( S_y \) is equal to \( S_x \) divided \( \text{pfactor} \). If they are in a \( \text{side-by-side} \) configuration, they have approximately the same orientation, \( S_y \) is the half Euclidean distance between the persons, and \( S_x \) is equal to \( S_y \) multiplied by \( \text{pfactor} \). If the group is not in one of these configurations or the groups have more than two members, we start by computing the overlapping area of the personal spaces ellipses between all group members. While the overlapping area is higher than zero, we reduce the parameters \( S_x \) and \( S_y \) proportionally using until the area is zero; that is, there is no overlapping.

Next, we assure that the parameters are at least the human size by verifying if the values obtained for \( S_x \) and \( S_y \) are at least higher or equal than the human dimensions \( \text{human}_x \) and \( \text{human}_y \), respectively. If one of the parameters is smaller, we set its value to the respective human dimension.

Finally, the parameters \( S_x \) and \( S_y \) are mapped to the desired parameters of the Asymmetric Gaussian:

\[
\sigma^p_x = S_x, \quad \sigma^p_y = S_y, \quad \sigma^p_z = S_x/\text{back\_factor},
\]

where the \( \text{back\_factor} \) represents the relation between personal frontal space and back space.

In this adaptation all group members have the same personal space dimensions, as they share the space equally when engaged in an F-Formation, and it only considers the distance between the group members.

Regarding group space, for this type of adaptation all the standard deviations of the 2D Gaussian (\( \sigma^p_x, \sigma^p_y \)) share the same value and are equal to the o-space radius.

2) \textbf{Obstacles Adaptation:} In this adaptation, personal space and group space adapt depending on the spatial context, specifically on the distance to obstacles, where these can be walls or objects. It aims to ensure that the personal space and group space do not overlap obstacles and make it easier for the robot to navigate in tight spaces, such as corridors, without invading personal spaces and group space. This adaptation extends with some adaptations the work proposed in [21], by considering the group space’s adaptation in addition to the personal space adaptation. Also, we modify their approach by searching for obstacles using the costmap information instead of using a list of identified obstacles.

The adaptation is performed after the Distance Adaptation, using as initial parameters (standard deviations of the Gaussian functions) obtained from the Distance Adaptation algorithm. For this adaptation, since obstacles affect persons individually, each individual will adapt its space, which means that individuals in the same group can have different personal space parameters.

The Obstacle Adaptation for the personal space is applied to all individuals detected. We start by initializing the personal space functions of each individual using the parameters provided by Distance Adaptation \( \sigma^p_x, \sigma^p_y, \sigma^p_z \). Then, we compute the adaptation for the personal space for individual \( p_j \). The algorithm receives as input the pose of the individual \((x^j_p, y^j_p, \theta^j_p)\), the parameters provided by the Distance Adaptation algorithm \((\sigma^p_x, \sigma^p_y, \sigma^p_z)\), human dimensions \((\text{human}_x, \text{human}_y)\), robot width \( \text{robot\_dim} \), the costmap and information related to it \((\text{resolution}, \text{costmap}, \text{origin}, \text{width})\) and finally the threshold costmap cell value to consider as object \( \text{cm\_obs\_threshold} \).

The first step of the algorithm is to find the obstacles in the costmap. For that, we start by drawing straight lines from the position of the person \((x^j_p, y^j_p)\) to a point at a distance \( d_j \), in four directions, \([\theta^j_p, \theta^j_p + \pi/2, \theta^j_p + \pi, \theta^j_p + 3\pi/2]\), where \( \theta^j_p \) is the orientation of the person. The maximum distance to search \( d_j \), i.e., the line length in \( j \) direction, is computed as:

\[
d_j = \sigma_j + \text{robot\_dim},
\]

where \( \sigma_j \) is the correspondent parameter of the Asymmetric Gaussian function in direction \( j \), and \( \text{robot\_dim} \) is the robot’s
width. We define this line length search since our objective is to find obstacles that impact robot navigation and space invasion. For obstacles farthest than the sum of the parameters of personal space and the robot's width, the robot will always be able to navigate without invading the space, and also, there will be no space overlap. Thus, there is no need to search further than this distance. The function parameter $\sigma_j$ can be associated with each search direction as:

$$\theta^p \rightarrow \sigma_1 = \sigma^p, \quad \theta^p + \frac{\pi}{2} \rightarrow \sigma_2 = \sigma^p, \quad \theta^p + \frac{3\pi}{2} \rightarrow \sigma_3 = \sigma^p, \quad \theta^p + \pi \rightarrow \sigma_4 = \sigma^p. \quad (8)$$

The next step passes by converting the start and endpoints of the line from meters to indexes to search for obstacles in the costmap. To compute the indexes of the costmap that compose the line, that is, the points representing a line in the costmap, we use the Bresenham's line algorithm [25]. This algorithm returns a list of indexes that create a line between the two provided points. Using the list of points of the line obtained, we can determine if there exists an obstacle in that direction. For that, we iterate the costmap using the line indexes until a cell’s cost is higher than a predefined threshold. If this happens, an obstacle is found, and one can determine the direction and distance to the obstacle.

For the directions without obstacles, the parameters related to that direction remain unchanged. For the directions with obstacles, we adapt the parameters.

The explanation of the adaptation will be given for a single direction; however, the reasoning applies to all directions. First, we start by verifying if the difference between the person’s distance to the obstacle and the parameter is less than the robot dimensions. If this difference is higher, there is enough room for the robot to navigate without adapting and invading personal space. However, if the difference is less than the human dimensions, we verify if the distance to the obstacle is less or equal to the parameter. If yes, this means that the personal space overlaps the obstacles, and we reduce personal space in that direction to be at least the distance to the obstacle. Then we verify if the parameter is smaller than humans’ dimensions. If not, we ensure that the parameters are at least the size of the human. If the distance to the obstacle is higher than the parameter, we verify if the distance to the obstacle minus the robot width is higher than the human dimensions. If the distance is higher, it is possible to adapt the parameter while keeping the parameter higher than human dimensions, making it possible for the robot to navigate without invading the personal space. The new parameter is computed as the distance to the obstacle minus the robot dimension.

Finally, we obtain for the individual $p_i$ its own parameters represented by $\sigma^p_{hi}, \sigma^p_{si}$ and $\sigma^p_{pi}$.

Regarding the group space adaptation, the algorithm is quite similar but does not have the human dimensions constraints. One can compute the obstacles adaptation for a group $g_k$ centered at $(x^g_k, y^g_k)$ using the group space parameters provided by the Distance Adaptation $(\sigma^g_x, \sigma^g_y)$. Since the group does not have orientation, the lines are also drawn in four directions, but with zero orientation, $[0, \pi/2, \pi, 3\pi/2]$. Since the group space is modeled by a 2D Gaussian function with only two parameters, now the relation of lines direction and parameters is given by:

$$0 \rightarrow \sigma_1 = \sigma^g_x, \quad \frac{\pi}{2} \rightarrow \sigma_2 = \sigma^g_y, \quad \frac{3\pi}{2} \rightarrow \sigma_3 = \sigma^g_y, \quad \pi \rightarrow \sigma_4 = \sigma^g_x. \quad (9)$$

Regarding the adaptation of the parameters, here we only verify if the difference between the wall and the parameter is less than the robot width. If it is, we adapt the parameter to be equal to the difference between the distance to the obstacle and the robot width.

As a result, if there is an obstacle for any direction, a new pair of parameters for the group space of a group $g_k$, $\sigma^g_{sx}, \sigma^g_{sy}$ are obtained.

C. Approaching Pose Estimation

After modeling the personal space and group space, it is possible to estimate the position and orientation to approach the groups. The approaching pose estimation will be based on the algorithm presented in [14]. Here, the field of view filtration will not be considered, the initial circle is created using the group radius instead of a predefined radius, and the obstacles are also considered so that there are no approaching areas on top of obstacles.

To guarantee the constraints of the F-Formations introduced in [9], and [26], the approaching pose of a group of humans must be outside all personal spaces and the group space. However, it must be inside the previously defined p-space, between the o-space radius and the p-space radius.

The algorithm is divided into two steps: 1) approaching area estimation and 2) approaching pose estimation. An example of the algorithm steps is presented in Figure 2. It starts by estimating the approaching area. First, a circle with a radius equal to the group radius is created around a group, as shown in Figure 2(a) represented by a thick blue line. Then, a filter that considers the obstacles and the maximum personal space of all group members and the group space is applied, as shown in Figure 2(b). If the filter’s output is empty, the circle’s radius increases by a preset value, only if it still lies in the F-Formation’s p-space. As mentioned in Section II-A, the group members position themselves in the p-space; this condition assures that the approaching pose is inside the p-space. The next step passes by estimating the approaching pose given the approaching areas. The potential approaching poses are determined as the central point of each approaching area, as represented in Figure 2(d) by red dots and black arrows. The final approaching pose will be selected by computing the robot’s distances to these points and choosing the closest to the robot’s current position.

D. Results

We present some preliminary results regarding the adaptive space algorithms and approaching pose estimation. For these results, the groups’ information (poses of the group members and group center) are provided as input files.

All the algorithms, code developed for this results and datasets used are available in the GitHub repository: https://github.com/franciscormelo/Adaptive-Space. The algorithms...
were developed in Python, and the results were obtained using a computer running macOS Big Sur with a 2.8 GHz Dual-Core Intel Core i7 processor and 16 GB 1600 MHz DDR3 of RAM.

For all the experiments of this section we set the parameters $A = 1$, $human_x = 20 \text{cm}$ and $human_y = 45 \text{cm}$.

1) Distance Adaptation: The Distance Adaptation is evaluated through different experiments with three different datasets: Groups Data, Synthetic Data \cite{10} and IDIAP Poster Data \cite{27}.

2) Groups Data: The Groups Data is a small dataset composed of nine synthetic groups with two to six elements. This dataset was created to visually test the adaptation algorithm by evaluating personal space adaptation. We apply the Distance Adaptation algorithm to the different groups of this dataset with the parameter $pspace_x = 80 \text{cm}$, $pspace_y = 60 \text{cm}$, $pfactor = 1.33$ and $back\_factor = 1.3$.

Thus, after analyzing the results for all groups of this dataset, for eight out of nine groups, the approaching area’s perimeter increases with adaptive parameters, and even in some groups, they have approaching zones while before they did not have them. The only exception was the group with the members farthest from each other, resulting in no overlapping of personal space; this means that the parameters are not adapted. The algorithm has more impact in groups with circular arrangement, as can be seen from the group 8 of the dataset represented Figure 3. In these cases, the use of fixed parameters returned empty approaching areas. On the other hand, the use of adaptive parameters returned possible approaching poses. This may be explained because when people are organized in a circular arrangement, they tend to be closer. Even though the number of approaching areas increases in this type of configuration, their dimension also decreases.

3) Synthetic Data and IDIAP Poster Data: We additionally evaluate the algorithm with two larger datasets with a larger number of groups: (i) Synthetic Data \cite{10} and (ii) IDIAP Poster Data \cite{27}. Table I shows the number of groups extracted from each dataset categorized by the different group sizes (number of individuals).

![Fig. 2: Approaching Pose estimation steps.](image1)

![Fig. 3: Personal Space and Group Space representation using adaptive and fixed parameters for group 8 of the Groups Data dataset and, respectively, approaching pose estimation.](image2)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Group Size</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic</td>
<td>180</td>
<td>260</td>
</tr>
<tr>
<td>IDIAP Poster</td>
<td>152 106 56</td>
<td>345</td>
</tr>
</tbody>
</table>

4) Parameters Distribution: In this experiment, we evaluate the frequency distribution after adapting the personal space dimensions $S_x$ and $S_y$ represented by $x$ and $y$, respectively. This evaluation is performed for the pair of parameters represented by a 2D Histogram. The results obtained for the Synthetic dataset can be seen in Figure 3 (a) and for the IDIAP Poster dataset in Figure 3 (b). The parameters used for this experiment are $pspace_x = 120 \text{cm}$, $pspace_y = 100 \text{cm}$, $pfactor = 1.2$ and $back\_factor = 1.3$. Note that the personal space parameters are initialized with the maximum values, to evaluate the distribution for the possible parameters.

By observing the figures from both datasets, one can immediately observe a high personal space dimension standard deviation. For the Synthetic dataset, the most frequent pair of values $(x, y)$ only has a percentage of 23.1%. For the IDIAP Poster dataset, the more frequent pair of values $(x, y)$ only has a percentage of 14.5%. For the Synthetic dataset, the percentages are higher because the Synthetic dataset is that the groups are more alike and with fewer individuals. It is also possible to notice that the most frequent dimensions are between 20 cm and 50 cm. Another conclusion that can be extracted from the 2D histograms is the most likely aspect ratio (the line that crosses most of the points).

This experiment shows that even in the same datasets, in the same conditions, the people are organized in different ways; thus, with fixed parameters, one can not have parameters adapted to each group.

5) Approaching Area Perimeter: This experiment starts by comparing the perimeter sum of approaching area and then compares the mean and standard deviation using fixed and adaptive parameters. The parameters used for the experiments with both datasets are $pspace_x = 55 \text{cm}$, $pspace_y = 45 \text{cm}$, $pfactor = 1.22$ and $back\_factor = 1.3$. We used as initial values of the personal space similar to the ones used in \cite{19}.

\url{http://profs.sci.univr.it/~cristianm/datasets.html}

\url{https://www.idiap.ch/dataset/idiap-poster-data}
Therefore it is possible to compare our approach for the average personal space parameters.

First, one compares the perimeter sum of approaching areas for each group size for both datasets and the sum for all groups. The results obtained for the Synthetic dataset can be seen in Figure 5 and for the IDIAP Poster data set in Figure 6. The sum of perimeters is higher for the approach with adaptive parameters for all group sizes in both datasets, and consequently higher for all groups. From these results, one obtains an increase of about 45% from the fixed parameters for the Synthetic dataset and about 107% for the IDIAP Poster dataset.

Next, the approaching area perimeters mean and standard deviation are measured for the different group sizes and both datasets. These results are described in Table II for the datasets. There is an increase of the mean and the standard deviation for all group sizes and both datasets with adaptive parameters compared to the fixed parameters approach.

IV. Socially Reactive Navigation System

We implemented a socially reactive navigation system to approach humans in social environments by incorporating the modeling of space and approaching pose estimation methods presented in the previous chapter into the system architecture proposed. The system is presented in Figure 7 and it comprises two main parts: a classic robot navigation scheme and a socially aware navigation framework [3], [19]. The socially aware navigation framework will be the main focus of this work, and it intends to distinguish humans from obstacles by extracting human’s socio-spatial-temporal characteristics within the proximity of the robot [19]. This part will be divided into four major functional blocks: human detection and tracking, group detection, modeling of space, and approaching pose estimation. The socially aware robot navigation system is implemented using ROS.
A. Group Detection

Having multiple people’s position and orientation, the following step passes by recognizing groups. [10] proposed a method called Graph-Cuts for F-Formations (GCFF) for group detection. This method detects an arbitrary number of F-Formations in still images, using as input the individuals’ pose, and it provides as output the center of the formation’s o-space.

B. Modeling of Space

The costmap_2d package [3] provides an implementation of a 2D costmap based on the layered costmap method proposed by [28]. The package provides standard layers, and for this work, the following were considered: (i) Static Layer, (ii) Obstacles Layer and (iii) Inflation Layer.

We propose two new layers for the costmap to implement the functions and algorithms that model the adaptive spaces: (i) Clean People Layer and (ii) Adaptive Layer. These new layers receive as input the groups detected by the robot provided by the GCFF algorithm with the respective adaptive parameters of the functions and those not engaged in a group. The Clean People Layer receives a list of poses of individuals detected by the robot and removes them from the costmap, as they were considered as obstacles. The removal is done by clearing the cells around an individual position, marking them as free. The clearing radius is equal to an average human footprint. The Adaptive Layer implements the personal space and group space cost functions with adaptive parameters proposed in Section III-A for groups. If an individual is not engaged in the group, it represents only the individual’s personal space. It receives the individuals’ poses and the parameters of each personal space, and the group’s information (center and radius). It also marks the cells of the zone of the body of the individuals with the maximum cost. The order in the layered costmap approach needs to be taken into account. The order of the layers used in this work is: (i) Master, (ii) Adaptive Layer, (iii) Inflation Layer, (iv) Clean People Layer, (v) Obstacles Layer and (vi) Static Layer. The clean people layer is between the inflation layer and obstacles layer, so that inflation only applies to obstacles and static layer. The adaptive layer is on top, meaning that it will be the last layer to update the costmap. Thus, no inflation will be applied to the information represented on this layer.

C. Approaching Pose Estimation

To estimate the approaching pose we use the algorithm presented in Section III-C. The filter that considers the maximum of personal spaces and group spaces and obstacles is made by checking the respective cells’ cost at each point of the circle with a radius equal to the group radius. If the point in a cell is marked as free, the respective point is added to the approaching area.

D. Results

We evaluate the socially reactive navigation system through experiments in simulation scenarios to demonstrate that the robot can simultaneously model the adaptive spaces for multiples groups, estimate the possible approaching poses, approach groups, and navigate around them without invading personal and group spaces.

Three new open-source ROS packages were developed to implement the system. The first package provides a new set of ROS messages that contain group information. The second represents the adaptive spaces in a costmap, and the third allows the robot to approach groups.

1) Simulation Setup: The socially reactive navigation system was tested in simulation using the Gazebo simulator. The platform used for the implementation was Vizzy [29], a humanoid on wheels robot. The simulation environment considered for the experiments is the 7th floor of the North tower at IST Lisbon. The different experiments are created by placing individuals in Gazebo with different 2D poses to generate different group arrangements. For the simulation experiments, the individuals’ 2D poses were obtained directly from Gazebo, as it keeps track of a list of person models in Gazebo.

The simulation experiments were obtained on a computer with a 2.8 GHz Dual-Core Intel Core i7 processor and 16 GB 1600 MHz DDR3 of RAM running Ubuntu 16.04 and using ROS Kinetic as middleware. For the simulations, Gazebo 9 was used as a simulator. The costmap layers were implemented using C++ and the ROS Nodes using Python.

The personal and group space functions of spaces are no longer normalized to 1 since the maximum value possible to represent in the costmap is 255. Thus, the function amplitude parameter A is set to 255. The robot width robot_dim considered was set to 100 cm and the parameters for the modeling of space were pspace_x = 120 cm, pspace_y = 100 cm, pfactor = 1.3 and back_factor = 1.3. The costmap obstacle threshold cm_obs_threshold is set to 254 since the obstacles are marked in the costmap with this value. The robot width robot_dim considered was set to 100 cm and the parameters for the modeling of space were pspace_x = 120 cm, pspace_y = 100 cm, pfactor = 1.3 and back_factor = 1.3.

We present the respective scenario in Gazebo for each experiment, and the costmap obtained is represented using RViz.

2) Multiple Groups Experiment: This experiment evaluates if the system can detect multiple groups and model the respective spaces. As shown in Figure 8, the system can detect the three groups and single individuals. For the single individuals, the system only models the personal space as expected using the initial parameters since no adaptation is applied to these individuals. For the groups, personal space and group space with adaptive parameters is represented. Through a visual inspection, it is possible to observe that the groups with individuals closer have smaller parameters while the ones with individuals farthest have higher parameters.

3) Approaching Pose Experiments: In these experiments, we pretend to demonstrate that the robot can estimate the
approaching area and possible approaching poses and then approach the group using adaptive parameters to model the spaces. For better visualization, the approaching areas and poses are represented in RViz using markers.

Since our approach impacts groups where individuals are close to each other and for bigger groups, we present two more experiments, one with five individuals (Figure 9) and the other with seven individuals (Figure 10).

For the tested groups, the robot could estimate the possible approaching areas and poses and approach the group by sending the approaching goal pose as a move_base goal. It is also essential to refer that the approaching poses estimation algorithm considers the obstacles by removing the approaching zones occupied by them, as seen from the experiment with five individuals.

The second scenario consists of a group placed between a wall and cylindrical objects. A goal pose is sent to the robot, in which the only way to achieve is by passing in front of the group. The results obtained for the experiment can be seen from Figure 12. The group space was adapted so that the robot could pass in front of the group without invading its group space.

Fig. 11: Obstacles adaptation for a group surrounded by obstacles.

4) Obstacles Adaptation Experiments: With the obstacles adaptation experiment, we aim to demonstrate the obstacles adaptation algorithm’s functioning and how the algorithm’s implementation allows the robot to adapt the spaces using the ROS costmap. To better visualize the obstacle adaptation result, the inflation layer was removed from the costmap layers to visualize the obstacles without inflation.

The first scenario consists of an individual placed in a corridor next to the wall, and a robot needs to pass through the human to pursue its path. The result obtained is shown in Figure 11. As can be seen from the costmap, the back of the individual’s personal space adapted so that the personal space does not overlap the obstacle. Also, there is enough distance from the wall to the human to pass without invading the personal space. Thus, the frontal space also adapts for the robot to pass by the human.

Fig. 12: Obstacle adaptation for a group surrounded by obstacles.
V. Conclusions and Future Work

This work’s objective was to implement a socially reactive navigation system that allows a robot to approach a group of humans in a socially acceptable manner where the personal space and group space adapt depending on the group arrangement and space constraints, avoiding the dependency on the choice of initial parameters. The distance and obstacle adaptation algorithms experiments using the datasets demonstrated that initial parameters’ choice impacts the approaching area’s dimensions and the importance of obstacle adaptation on robot navigation. By using adaptive parameters, the spaces are suitable for any context. Thus, the combination of algorithms combination provides a more realistic approach, making the robot more human-like. The socially reactive navigation system was tested in challenging scenarios for conventional navigation approaches on simulation, and the results demonstrated that the robot could detect and approach groups of different sizes while using adaptive personal and group spaces. It also demonstrated that the robot could adapt the spaces based on the distance to obstacles providing more realistic navigation in tight spaces.

However, the system presents some limitations. The first is that the approaching pose estimation algorithm does not remove the approaching areas where the robot does not fit. Another limitation is the obstacles adaptation does not consider human-object interactions. The robot should detect the interactions and not adapt the spaces in these cases. Finally, a more flexible group space function should have been considered since the function that represents the group space only has two parameters. Thus, when there is an adaption in one direction, the function also adapts in the opposite direction. In the future, we also intend to test and make experiments of the system in the real world and evaluate the safety and comfort through surveys with different humans.

References


