2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Modeling a Social Placement Cost to Extend Navigation Among Movable Obstacles (NAMO) Algorithms

Benoit Renault[a], Jacques Saraydaryan[b] and Olivier Simonin[a]

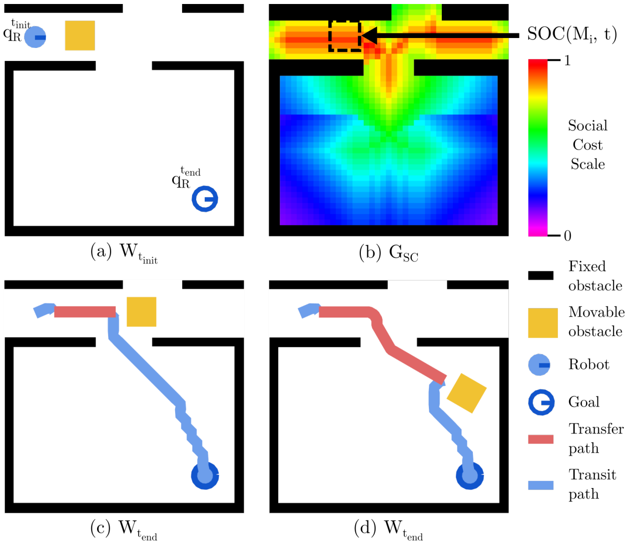
Abstract— **Current Navigation Among Movable Obstacles (NAMO) algorithms focus on finding a path for the robot that only optimizes the displacement cost of navigating and moving obstacles out of its way. However, in a human environment, this focus may lead the robot to leave the space in a socially inappropriate state that may hamper human activity (i.e. by blocking access to doors, corridors, rooms or objects of interest). In this paper, we tackle this problem of "Social Placement Choice" by building a social occupation costmap, built using only geometrical information. We present how existing NAMO algorithms can be extended by exploiting this new cost map. Then, we show the effectiveness of this approach with simulations, and provide additional evaluation criteria to assess the social acceptability of plans.**

I. INTRODUCTION

In 2005, Stilman et al. [1] formulated the field of Navigation Among Movable Obstacles (NAMO). The NAMO problem consists in planning a path from a start to a goal position, while moving obstacles if necessary. It extends the well known Piano Mover's Problem by differentiating static and movable obstacles, and allowing the manipulation of the latter. to find a path or to minimize a displacement cost function (eg. travel distance, time, energy). Service robotics, in particular, would definitely benefit from algorithms capable of dealing with manipulable clutter, doors or objects.

Actually, service robotics also imply robot navigation in human environments. Research in Socially-Aware Navigation (SAN) [2], [3], [4] has shown that minimizing disturbance[1] to humans is paramount in order to raise their acceptability of robots in such environments. However, until now, to the best of our knowledge, this necessity of minimizing disturbance to humans (or any other autonomous agents types) has never been considered in NAMO problems. This may lead to cases such as in Fig.1, where current algorithms would even deteriorate free space connectivity; forcing other agents to either move the object again or take detours to reach their own goal. This is why we propose to define Social NAMO (S-NAMO) problems and algorithms to avoid such situations.

When modifying the environment configuration, the most obvious way for a robot to minimize disturbance to humans is to avoid invading space that humans need to keep free. Determining these semantic spaces could be done by in situ human behavior observation [5], but the robot may not have the opportunity to observe long enough (due to timing requirements, privacy concerns, or task complexity). Human activity could otherwise be inferred from the objects layout, through the understanding of their respective affordances and related affordance spaces [6]. However, this would require both an extensive prior knowledge of human environments and good enough sensing capabilities to leverage it.



(a) $W_{t_{init}}$ (b) G_{SC} (c) $W_{t_{end}}$ (d) $W_{t_{end}}$

Fig. 1: Basic pathological case (see notations Sec. III). (a) The robot's goal is in the adjacent room but unreachable due to a movable obstacle. (b) Our social occupation costmap. (c) Classic NAMO solution. (d) Our S-NAMO approach's solution.

Our claim is that the significance of space (or rather, the importance of leaving it free) can be at least partially inferred without such strong pre-requisites. Indeed, as the field of interior design shows, the layout of human space is very much deliberate: *"Providing adequate space for movement is essential and requires careful consideration of ergonomics, scale, and organizational flow"* [7]. Free space thus has, in and of itself, a strong semantic meaning we can use of accessibility and circulation.

We build upon this observation to compute a social occupation costmap to improve the social acceptability of NAMO plans. For this, we model the environment as passageways corresponding to the skeleton, then associate to each skeleton cell their corresponding space allowance, and convert it into a social cost that we then propagate in a decreasing wave. We evaluate this model by applying it to a reference NAMO algorithm [1] in several experimental scenarios. We provide new quantitative performance criteria to evaluate the social acceptability of plans, and study the computational and displacement cost overhead of our approach.

[a] Benoit Renault and Olivier Simonin are with INSA Lyon, CITI Lab and INRIA Chroma Team `benoit.renault@insa-lyon.fr`, `olivier.simonin@insa-lyon.fr`

[b] Jacques Saraydaryan is with CPE Lyon, CITI Lab and INRIA Chroma Team `jacques.saraydaryan@cpe.fr`

[1] In Socially-Aware Navigation, disturbance is used as a synonym for 'discomfort', the feeling/state of being unsafe [3].

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

This paper is organized as follows. Section II provides an overview of related work. Section III introduces the Social Placement Choice problem and Section IV details the proposed costmap computation. Section V presents how to use the costmap to extend existing NAMO algorithms, and we evaluate the approach with simulated scenarios in Section VI. We conclude and discuss future work in Section VII.

II. RELATED WORK

In a previous state-of-the-art on NAMO algorithms [8], we have shown that existing approaches have never considered the consequences of the robot's choices, beyond its interest to reach its goal. All existing NAMO works only consider the minimization of the robot's displacement cost. Only Stilman et al [1] & Kakiuchi et al [9] suggested the idea of taking object fragility into account, but have not applied it.

The Social/Human-aware variations of problems most related to NAMO mainly focus on direct human interaction. In Manipulation Planning [10], [11] , Assembly Planning [12] as well as in Combined Task/Motion Planning [13], the focus of research is collaborative manipulation in close proximity and/or handover manipulations. In the end, to the best of our knowledge, there is no existing work on the social acceptability estimation of object placement choice in the NAMO-related motion planning literature either.

We have found that the Social Placement Choice problem we investigate in this paper shares objectives with realistic indoor scene synthesis [14], in that it also tries to find implicit or explicit rules to help decide of socially acceptable object placement For example, in [15], Yu et al. used a set of rules, automatically learned from human-designed indoor environments (e.g. distance/orientation to walls/specific objects, pathways between doors,...) to arrange furniture in a convincing-enough way for humans. However, there are two major differences with our problem: in realistic indoor scene synthesis, the aim is not to *preserve* an existing human-made arrangement of space designed by humans, and since there is no robot, the optimization of displacement cost for an agent actually moving the objects is not addressed. Also, it is assumed that any required semantic data is available, but acquiring such knowledge in situ is no easy task for a robot, as the active work in semantic mapping capabilities shows [4]. This motivates our choice to build a new model requiring only a binary occupancy grid of the fixed obstacles, such a grid already being a pre-requisite of commonly-used navigation frameworks like the ROS Navigation Stack [16].

Finally, in a survey on human-aware robot navigation, Kruse et al. [2] show that a common social navigation approach is to build a costmap representing zones to avoid or favour for the robot's trajectory. While they recognize it may oversimplify the problem, it allows for simple search, representation, and combination of many types of social considerations. That is why we decide to build our model as a social occupation costmap.

III. SOCIAL PLACEMENT CHOICE PROBLEM

Following the quote of [7] in Sec. I, "Socially-acceptable object placements" can be defined as object configurations that best preserve the adequacy of space for movement in regards to ergonomics, scale and organizational flow. We reformulate this notion into the S-NAMO problem of Social Placement Choice hereafter (illustrated in Fig.1).

Definition 1: SPC problem In S-NAMO, the Social Placement Choice problem is to determine for a robot R the plan P that brings it from initial configuration $q_R^{t_{init}}$ to a goal configuration $q_R^{t_{end}}$ while minimizing both the resulting displacement cost[2] $DC(P)$ and total social occupation cost $SOC(P)$ caused by the occupation of obstacles M_i it moves.

Let us define notations and structures of the problem:

- a mobile robot R
- a set of fixed undeformable obstacles that cannot be moved by the robot $F =< F_1, ..., F_m >$
- a set of movable undeformable obstacles that can be moved by the robot $M =< M_1, ..., M_n >$, $F \cap M = \emptyset$
- a physical representation of the world W, with $R, F, M \subset W$. For the sake of simplicity, elements $E_i \in W$ are considered undeformable polygons in a 2D plane, allowing us to define a world state at time t as a set of element configurations $W_t =< q^t_{E_1}, ..., q^t_{E_p} >$ with $q^t_{E_i} = (x^t_{E_i} \in \mathbb{R}, y^t_{E_i} \in \mathbb{R}, \theta^t_{E_i} \in [0, 2\pi])$.
- a discrete representation $G(W_t)$ of W_t as a 2D binary occupancy grid without loss of generality. Cells are addressed by tuples of coordinates $(x, y) \in \mathbb{N}^2$.
- another 2D grid G_{SC} built from $G(W_t)$ (thus sharing the same dimensions, position, orientation and resolution), containing in each cell a social occupation cost $SC(x, y) \in \mathbb{R}^+$.

We write $Cells(E_i, t)$ the cells occupied by an element E_i at time t. The social occupation cost $SOC(M_i, t)$ is written as the sum SC(x, y) of each cell occupied by M_i at t:

$$SOC(M_i, t) = \sum_{\forall (x,y) \in Cells(M_i,t)} SC(x, y) \qquad (1)$$

The navigation plan P followed by the robot is made of path components. A path component is a sequence of robot configurations q_R^i where the robot moves without manipulating the object (*transit path*) $p_{transit}$, or where the robot moves with the object (*transfer path*) $p_{transfer}$.

The displacement cost $DC(P)$ of the plan P is expressed as the sum of the estimated displacement cost $DC(p)$ for each path component p:

$$DC(P) = \sum_{\forall p \in P} DC(p) \qquad (2)$$

The displacement cost $DC(p)$ is the sum between each successive robot pose $q_R^i \in p$ of an estimation function $f(q_i, q_{i+1})$ of distance, time or energy, that depends on hypotheses as to the robot capabilities and the physical characteristics of obstacles:

$$DC_f(p) = \sum_{i=1}^{|p|-1} f(q_i, q_{i+1}) \qquad (3)$$

[2] typically energy, time, or distance

IV. SOCIAL OCCUPATION COSTMAP COMPUTATION

A. Heuristic hypotheses

Following our argumentation in Section I, let us assume that all free space in human environments can be considered as passageways that are *designed* for a certain number of people to walk abreast (which holds especially true in public buildings [17]). Since we only assume to have the binary occupancy grid of static obstacles, our focus can only be the preservation of the environment's original topology and space allowances. Thus, we propose the following **two heuristic hypotheses** to ensure this:

1) AVOID-MIDDLE: The original free space allowance of a passageway must be preserved as much as possible, so it must be the least divided (see Fig. 2): thus, the cells in the "middle of the space" must have the highest local costs;
2) AVOID-NARROW: The narrower a passageway, the more risk there is of blocking it with an object: thus, narrow spaces must have higher costs than wide spaces.

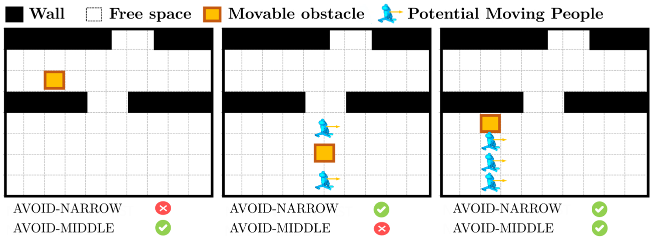


Fig. 2: Illustrations of the AVOID-MIDDLE and AVOID-NARROW heuristic hypotheses. In the first figure, we can see that leaving an obstacle in a narrow space is more likely to end up reducing connectivity. In the last two figures, we can see that leaving an obstacle furthest from the middle of space least divides it, maximizing the number of people that can access it at once.

B. Computation steps overview

The above heuristic hypotheses are translated into a four step algorithm to build the social occupation costmap from the binary occupancy grid of fixed obstacles $G(F)$:

1) **Skeletonize**: Find the set of free cells $SK(G(F))$ that represents the middle of all passageways[3] (Fig.3.A.);
2) **Compute the minimal distance** for each cell in $SK(G(F))$ to the fixed obstacles $\in F$, i.e. measure the original space allowance of passageways (Fig.3.A.);
3) **Associate** a social cost to each cell in $SK(G(F))$ by relating the previously determined minimal distance from obstacles to appropriate human-sized constants using f_{conv} (Fig.3.B);
4) **Propagate** the social cost values from the skeleton cells in $SK(G(F))$ in a decreasing wave defined by f_{prop} (Fig.3.C.).

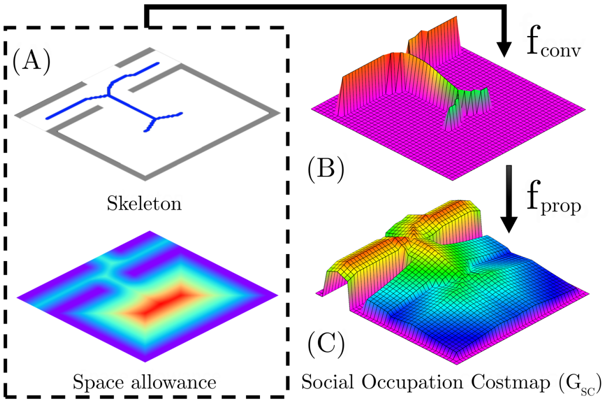


Fig. 3: Illustration of the 4 step computation: skeletonization & distance transform (A), conversion of space allowance into Social Cost (B) and decreasing wave propagation (C)

C. Skeleton & Space Allowance

Determining the skeleton of a binary occupancy grid and the minimal distance to obstacle cells are well-known problems in the literature, and can respectively be done using skeletonization [18] and distance transform [19] algorithms.

Given the tremendous amount of choice of skeletonization algorithms in the literature, we focused our choice on three of the most popular: Zhang-Suen [20], Guo-Hall [21] and Medial-Axis [22]. We ended up choosing the Guo-Hall algorithm, because, it has a very good contour noise immunity, and does not produce end-of-edge artifacts like Zhang-Suen. Just like the Medial-Axis method, it guarantees the preservation of connected components, which is paramount to properly represent passageways. An example of such skeletonization is available on Fig. 3A.

To express the original space allowance of passageways (i.e. their "broadness"/"narrowness"), we must measure the minimum **euclidean** distance from fixed obstacles (in contrast to taxicab or chessboard distance), since passageways do not necessarily follow perfect diagonals or straight lines in the grid. We do that using the Euclidean Distance Transform (Fig. 3 A.). Since the minimum distance to obstacles is only half of the space allowance between the two nearest obstacles, we double it to get the space allowance $d_{allow}(x, y)$.

D. From Space Allowance to Social Occupation Cost

To transform the previously computed space allowance in each cell $\in SK(G(F))$ into a meaningful social cost that respects the AVOID-NARROW rule, we must use real-world measurements that define what is narrow or wide **for humans**. Basically, we want the cost to be maximal for any space allowance that is below a reference human diameter $\boldsymbol{d_h}$, then decrease as the space allowance increases.

To find a convincing d_h value, we turn to anthropometrics, and use the DINED [23] database. We derive the value of d_h from standing breadth over the elbows, which is the widest measurement in a standing posture with arms at rest. According to the most recent 2004 data for Dutch adults between the age of 20 to 60, we can reasonably cover about 95% of the population by setting $d_h = 0.55m$. This

[3] That is, the thinnest discrete version (one cell wide) of the environment's shape equidistant to the obstacles boundaries.

value is coherent with the average adult space clearance for ambulatory human movement of 0.559m provided by [7].

Then, to determine a relevant decrease profile, we turn to a commonly used law-defined measurement in french construction [17]: "UP" or "Unités de Passage" (litt. "Passage Units"). UPs describe the minimal space allowance between obstacles/walls for the passageway to allow proper emergency evacuation for n persons: 1UP $= 0.90m$ (meant to comply with standard wheelchair width), 2UP $= 1.40m$, and for three and beyond nUP $= n * 0.60m$. An indoor passageway is rarely more than 5UP $= 3m$ large because the same law recommends to increase the number of passageways rather than create ones that are more than 5UPs large: therefore, we choose to keep the social cost constant beyond 5UP.

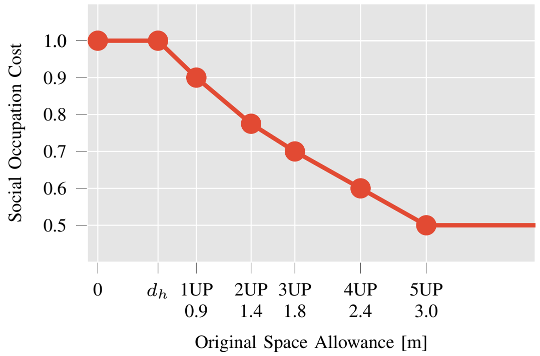


Fig. 4: Social Occupation Cost as a function of the Original Space Allowance: $SC(x, y) = f_{conv}(d_{allow}(x, y))$

For the sake of keeping social occupation cost values comparable, we reduce the interval of acceptable values to $[0, 1]$. The cost values associated with each UP between 1 to 5 can be adjusted depending on the relative importance users give to preserving space allowance. Fig. 4 shows a set of values we have found to yield a reasonably convincing social occupation cost representation in our experiments of Section VI. Intermediate values are determined by simple linear interpolation.

E. Propagation

As explained in the introduction of Section IV, we need a propagation procedure that guarantees a decreasing cost from the skeleton values, in order to guarantee the benefit of getting the obstacle away from the "middle of space".

For this, we use a variant of the Wavefront Propagation algorithm [24] where we start from the skeleton's set of cells SK. We then iteratively mark all unlabeled 4-neighbors cells with the λ-decayed minimum value of their own neighbor cells from the previous iteration, with $\lambda \in]0, 1[$, as described in equation 4. Since the choice of λ value depends on the grid resolution, we will discuss it in Section VI.

$$f_{prop} \colon W \to \mathbb{R}[0, 1]$$

$$f_{prop}(x, y) = \begin{cases} f_{conv}(x, y) & \text{if } (x, y) \in SK \\ \lambda \cdot min(Neigh(x, y)) & \text{if } (x, y) \notin SK \end{cases} \quad (4)$$

The resulting costmap is illustrated in 3D in Figure 3.C. As expected, the middle of space is populated by ridges created by local social occupation cost maxima, and the narrowest spaces are associated with the highest costs, encouraging placement choices outside the corridor and away from the room's center, especially in front of the opening.

V. EXTENSION OF NAMO ALGORITHMS

A. NAMO Algorithms structure

From our previous state-of-the-art work [8], we draw the conclusion that NAMO algorithms share a common structure. Basically, as long as no satisfying[4] plan to reach the goal has been/can be found, these algorithms iteratively or recursively loop through known movable obstacles, each using a different way to choose the order in which to do so.

Once an obstacle is selected for manipulation search, a reachable non-colliding pose to leave it at needs to be found so that it creates an opening for the robot. While APPROACH 1 finds the relocation configuration as the robot explores its action space, APPROACH 2 first samples configurations, regularly or randomly, within a set of constraints (Alg.1). Usual constraints for pose sampling are: distance from initial obstacle pose, and a discrete set of accepted rotations. Each NAMO approach explores the action space with different algorithms, generally based on graph-search like Dijkstra, A* or RRT (see NAMO algorithms comparison table in [8]). Computation of transit paths to and from obstacles and action space exhaustion exit conditions are not shown for the sake of brevity.

Algorithm 1: NAMO manip. search macro algorithm

```
Data: World state $W^t$ and most relevant obstacle $M_i$
Result: A transfer path $p_{transfer}$
while no satisfying transfer path $p_{transfer}$ found do
    if APPROACH 1 then
        Explore robot action space without focus;
    else if APPROACH 2 then
        Sample obstacle poses within constraints;
        forall sampled poses do
            Explore robot action space with pose
              focus;
        Relax constraints one step;
return transfer path $p_{transfer}$;
```

B. Social occupation cost integration

Minimizing both the displacement cost of existing NAMO algorithms and our social occupation cost is a typical multi-objective optimization problem. Such a problem is generally solved by comparing the cost of all available solutions. In our case, this would mean exploring the entire robot action space to find all reachable non-colliding poses for the obstacle and

[4] "satisfying" meaning displacement cost optimal or not, depending on the algorithm

computing the associated social occupation costs. Exploring such a high-dimensional problem is however intractable, even in small spaces like in Fig.1.

That is why we propose to compute a heuristic compromise cost $CC(x,y)$ for the set of potentially reachable cells to move the selected obstacle at. We do that with the following process, illustrated in Fig. 5:

1) Create an occupancy grid taking into account all obstacles except the robot R and selected obstacle, M_i.
2) Inflate the grid by the inscribed circle radius of M_i. This operation allows to find a subset of potentially acceptable cells for M_i.
3) Compute a goal-less Dijkstra search from M_i's center cell. The set of potentially reachable cells for moving M_i is thus obtained, with an underestimate of the distance $d_{M_i}(x,y)$ to reach them[5].
4) Compute the compromise cost CC for each potentially reachable cell with a weighted sum of the following normalized costs: occupation cost ($SC(x,y)$), distance $d_{M_i}(x,y)$ and a heuristic euclidean distance from cell to NAMO goal $d_{goal}(x,y)$:

$$CC(x,y) = \frac{w_1 * d\prime_{M_i} + w_2 * d\prime_{goal} + w_3 * SC\prime}{w_1 + w_2 + w_3} \quad (5)$$

where $SC\prime$, $d\prime_{M_i}$, and $d\prime_{goal}$ are respectively normalized versions of SC, d_{M_i}, and d_{goal} in $[0,1]$.

Finally, we use this knowledge in the manipulation search subprocedure of NAMO algorithms by:

- bounding the action space exploration by stopping when a solution leaves M_i's center in one of the cells with the top n percent best CC cost,
- sampling a goal obstacle pose among the cells with the top n percent best CC cost to focus the search.

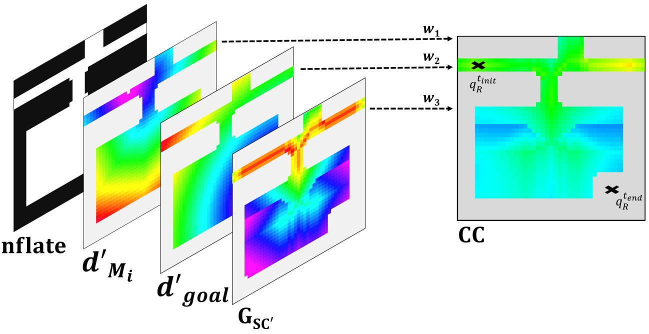


Fig. 5: Compromise cost (CC) computation steps

VI. EXPERIMENTAL EVALUATION

A. Algorithm choice

We choose to implement our method upon the founding algorithm of Stilman et al. [1] since all following works on NAMO have been inspired by it, and because it focuses more on the method to choose which obstacles to evaluate and their order than the actual manipulation search method. Indeed, in [1], this search is mentioned to be a bounded Breadth-First Search over the robot action space (APPROACH 1 in Alg. 1), and we implemented it as is for the baseline NAMO to compare against. However, to take full advantage of our computations from Section V, we changed this subroutine for an A* search that focuses on the best compromise cell for the S-NAMO variant (APPROACH 2)[6].

B. Simulator & simulation parameters

We implemented our approach in a custom open-source simulator[7] that only considers object geometry for interaction (no kinematics nor dynamics) akin to the one used in [1], and provide an integration layer with ROS for visualization in Rviz. Fig. 1 and 6 illustrate the simulator. It is also presented in the video provided with the paper.

All grids are set to a resolution of 0.1m/cell and the robot action space allows for backward/forward unit translations of 0.1m, and unit rotations of 30°. We assume that the robot can be reduced to a circle (of radius 0.2m) to accelerate collision checks. Full geometry and movability-related world knowledge is provided to the robot, as required by the baseline algorithm [1]. Such knowledge could be provided by an external motion capture system and markers on movable obstacles like the baseline authors did in [25]. Displacements with an obstacle are penalized by a factor of 2 to reflect the extra effort (additional semantic data like obstacle weigth could result in a finer approach). All experiments are run with $\lambda = 0.97$ (so that social costs decrease fast enough given the chosen resolution), $w_1 = 10$, $w_2 = 2$ and $w_3 = 15$, and robot action space exploration stops when a solution is at a cell with a compromise cost within the best 1%. The choice of weights here is manually tuned over the given scenarios, but could be optimized through machine-learning over a wider range of real-world scenarios.

C. Comparison criteria

We now need criteria for measuring both the displacement cost efficiency and social acceptability of solutions. For displacement cost efficiency, as in all of NAMO literature, we use the total euclidean distance for transfer and transit paths, respectively $L_{transit}$ and $L_{transfer}$. As for social acceptability, we define the following criteria:

- **Total Social Occupation Cost**: Derived from our model, it is the sum of SC for all movable obstacles. It represents the global space accessibility disturbance for humans.

$$ST(W_t) = \sum_{\forall M_i \in M} S_{(M_i,t)} \quad (6)$$

- **Number of connected components**: In the grid representation of W_t inflated by half the reference human

[5] The goal cells are filtered out of this set afterwards so that the robot can't put the obstacle on it.

[6] The accompanying video shows how both algorithms unfold: https://team.inria.fr/chroma/files/2020/07/iros_2020_Renault.mp4

[7] Provided with all simulation code and data, available at: https://gitlab.inria.fr/brenault/s-namo-sim

diameter d_h defined in Section IV, it is the number of interconnected sets of cells $C_h^i(W_t)$ that make up the global set of accessible cells for humans. It allows to measure the topology variation.

$$Ncc(W_t) = |\{C_h^i(W_t)\}| \tag{7}$$

- **Space fragmentation percentage**: In the same grid as before, it relates the size of the biggest free space component $C_h^{max}(W_t)$ to the total size of free space. It allows to observe the variation in space allowance.

$$frag(W_t) = 1 - \frac{|C_h^{max}(W_t)|}{|C_h^{acc}(W_t)|} \tag{8}$$

A plan improves social acceptability by lowering any of these criteria between $W_{t_{init}}$ to $W_{t_{end}}$. Finally, we also provide the total planning time $T_{planning}$ to represent the computational cost of each plan.

D. Experimentations

	$L_{transit}$ / $L_{transfer}$ [m] / [m]	$ST(W_{tinit})$ / $ST(W_{tend})$	$Ncc(W_{tinit})$ / $Ncc_(W_{tend})$	$frag(W_{tinit})$ / $frag(W_{tend})$ [% / %]	$T_{planning}$ [s]
[A]					
Baseline	4. / 1.1	46.8 / 43.0	2 / 3	0.4 / 12.6	0.5 ± 0.2
S-NAMO	2.5 / 2.8	46.8 / **15.0**	2 / **1**	0.4 / **0**	1.1 ± 0.2
[B]					
Baseline	33.0 / 1.2	61.0 / 44.9	3 / 4	23.6 / 74.3	13.8 ± 1.6
S-NAMO	35.7 / 3.4	61.0 / **23.7**	3 / **1**	23.6 / **0.**	**2.0** ± 0.3
[C]					
Baseline	11.2 / 0.6	559.0 / 556.5	4 / 3	45.5 / 28.5	1.3 ± 0.2
S-NAMO	13.0 / 2.5	559.0 / **545.9**	4 / 3	45.5 / 28.6	3.1 ± 0.4
[D]					
Baseline	31.8 / 1.0	105.4 / 116.7	14 / 15	47.2 / 45.5	28.1 ± 1.8
S-NAMO	35.4 / 4.5	105.4 / **79.3**	14 / **13**	47.2 / **5.0**	26.5 ± 1.7

TABLE I: Performance criteria comparison table. Displacement and computation costs for [B] and [D] are cumulated over goals. ST and Ncc have no units. $T_{planning}$ is given as average with standard deviation over 50 iterations.

We evaluated our approach with four scenarios of increasing complexity, called [A] "corridor-to-room" (Fig.1), [B] "crossing", [C] "after-the-feast" and [D] "CITI-lab" (Fig. 6). In scenarios [A] and [C], the robot is only required to reach one goal, however in [B] and [D], it has to sequentially reach several. Table I compares results obtained in these 4 scenarios between the baseline NAMO algorithm and its S-NAMO counterpart. We comment these results hereafter.

Scenario [A] shows the worst-case scenario that can happen with current NAMO approaches: unintentional connectivity loss. At the cost of little extra computation and reasonable further displacement of the obstacle (the total distance is almost the same), our approach results in complete defragmentation. The low initial fragmentation is due to lack of knowledge beyond the corridor.

Scenario [B] allows to highlight an interesting phenomenon: NAMO algorithms, when faced with a movable obstacle blocking a tight intersection branch they need to pass will naturally tend to block another branch because of the sole focus on displacement cost. As the robot needs to get from one room to the next, it repeatedly blocks another corridor and systematically induces an additional computation cost, that after 4 iterations is already 6 times greater than our S-NAMO approach. After the 4th iteration the chosen obstacle placement even blocks all branches for a human being of diameter d_h.

Scenario [C] shows that, of the three criteria we propose to evaluate social acceptability, only the one based on our social cost computation ST allows us to differentiate the quality of solutions that produce the same relative change in number of connected components and space fragmentation.

Finally, Scenario [D] shows that our approach is capable of growing up to scale in big realistic environments. Indeed, despite the environment size and complex topology, Fig.6 corridors around the middle obstacle and the tight passages between desks in the surrounding the offices are properly detected as less appropriate than the wider areas like office corners, the entry hall on the left and the big intersection on the right. The results in Table I show that despite this increase in complexity compared to the other scenarios, our approach still succeeds in achieving more socially acceptable solutions than the baseline approach, for almost the same computational cost[8].

While the S-NAMO solutions presented here have better social acceptability than their baseline NAMO counterparts, some of them could still be improved. For example, in case [C], if the moved obstacle were a stool, it would have been better to keep it closer and aligned to the tables than the wall. This could only be solved better if this extra semantic information were available. A potential approach would be to apply conditional cost reductions and augmentations around points of interest, but this would be the object of future work.

VII. CONCLUSION

This paper first introduced a social placement cost model based on two heuristic hypotheses as to the significance of free space in human environments: avoiding narrow spaces, and not leaving objects in the middle of space. This model relies solely on the analysis of the environment's binary occupancy grid of fixed obstacles. Then, we showed how to extend NAMO to S-NAMO algorithms that result in plans with better social acceptability. Experimental results on scenarios of increasing complexity show the scaling capability of our method. To the best of our knowledge, this is the first approach to propose an answer to social concerns in NAMO problems.

Our future works will focus on incorporating more complex semantic information to further improve social acceptability. We will also apply our model to other NAMO algorithms (e.g. [26], [27]) in order to manage partial and uncertain knowledge of the environment, thus bringing us closer to real-world conditions and experiments. Finally, we plan to study the implications of other autonomous agents actual presence such as humans.

[8] The high amount of detected connected components is due to the chosen reference human diameter d_h: in the offices, space is considered fragmented between some desks because of this. This does however not make the comparison any less relevant.

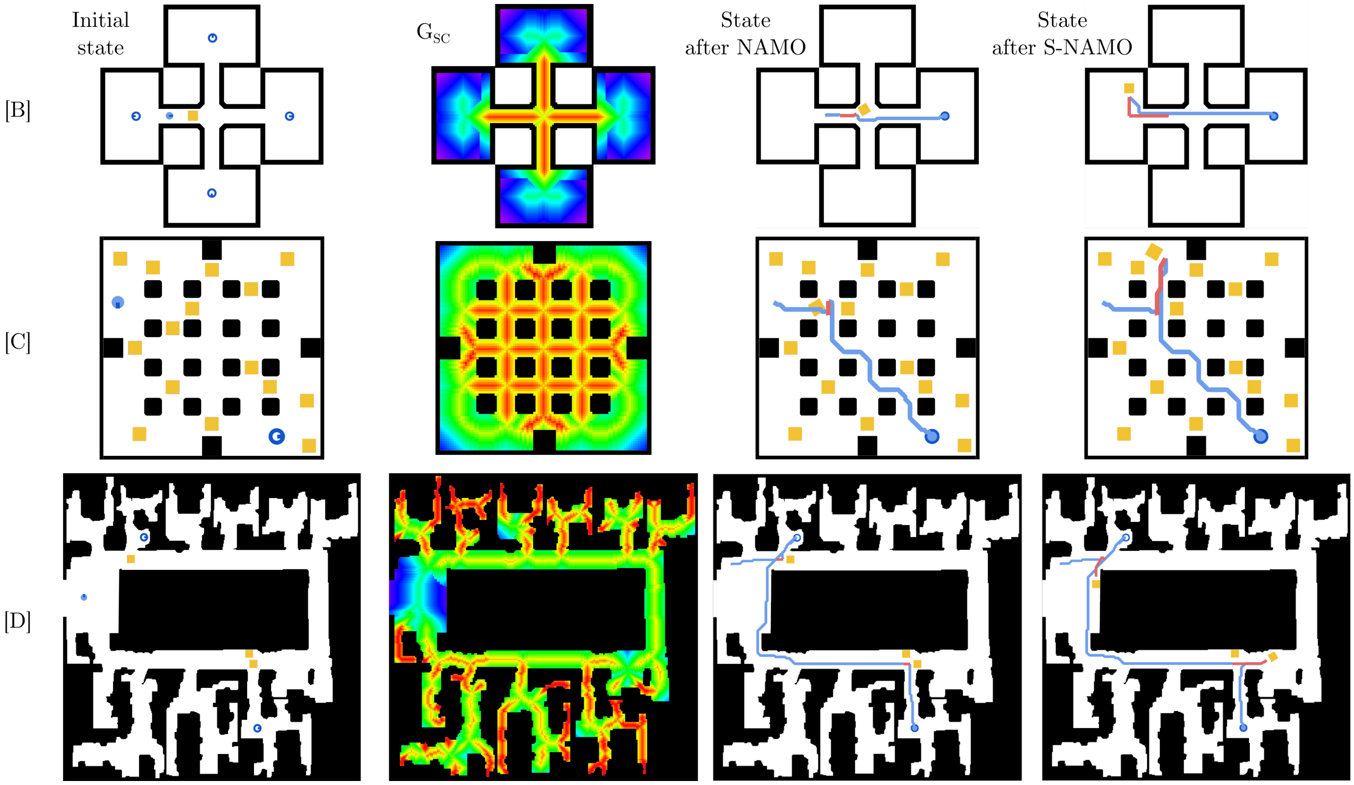


Fig. 6: Experimental scenarios & NAMO vs. S-NAMO results. [B] is 4 rooms joined by corridors, each with a goal (executed counter-clockwise from the right). Path to 1st goal is the only one shown for the sake of readability. [C] is a room filled with unmovable tables (black squares) and miscellaneous movable obstacles "after a party". [D] is a grid representing our lab's 2nd floor obtained from a 3D scan. First goal is in 2nd room from the left at the top, second one is in the first room from the right at the bottom.

References

[1] M. Stilman and J. J. Kuffner, "Navigation among movable obstacles: real-time reasoning in complex environments," *International Journal of Humanoid Robotics*, vol. 02, no. 04, pp. 479–503, 2005.

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

[3] 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.

[4] 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.

[5] R. Limosani, L. Y. Morales, *et al.*, "Long-term human affordance maps," in *IEEE/RSJ-IROS*, 2015, pp. 5748–5754.

[6] F. Lindner and C. Eschenbach, "Towards a Formalization of Social Spaces for Socially Aware Robots," in *Spatial Information Theory*, ser. Lecture Notes in Computer Science, M. Egenhofer, N. Giudice, *et al.*, Eds. Springer Berlin Heidelberg, 2011, pp. 283–303.

[7] M. Mitton and C. Nystuen, *Residential Interior Design: A Guide to Planning Spaces*, 3rd ed. New York, NY: Wiley, 2016.

[8] Benoit Renault, Jacques Saraydaryan, and Olivier Simonin, "Towards S-NAMO: Socially-aware Navigation Among Movable Obstacles," in *RoboCup 2019: Robot World Cup XXIII*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2019.

[9] Y. Kakiuchi, R. Ueda, *et al.*, "Working with movable obstacles using on-line environment perception reconstruction using active sensing and color range sensor," in *IEEE/RSJ-IROS*, 2010, pp. 1696–1701.

[10] S. Pellegrinelli, A. Orlandini, *et al.*, "Motion planning and scheduling for human and industrial-robot collaboration," *CIRP Annals*, vol. 66, no. 1, pp. 1–4, 2017.

[11] E. A. Sisbot and R. Alami, "A Human-Aware Manipulation Planner," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1045–1057, 2012.

[12] V. V. Unhelkar, P. A. Lasota, *et al.*, "Human-Aware Robotic Assistant for Collaborative Assembly: Integrating Human Motion Prediction With Planning in Time," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2394–2401, 2018.

[13] R. Alami, M. Gharbi, *et al.*, "On human-aware task and motion planning abilities for a teammate robot," in *Human-Robot Collaboration for Industrial Manufacturing Workshop, RSS*, 2014.

[14] S.-H. Zhang, S.-K. Zhang, *et al.*, "A Survey of 3d Indoor Scene Synthesis," *Journal of Computer Science and Technology*, vol. 34, no. 3, pp. 594–608, 2019.

[15] L.-F. Yu, S. K. Yeung, *et al.*, "Make it home: automatic optimization of furniture arrangement," *ACM Trans. Graph.*, vol. 30, p. 86, 2011.

[16] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *IEEE/RSJ-IROS*, 2014, pp. 709–715.

[17] French Interior Ministry, "Arrete du 25 juin 1980," livre II, Section 9, Sous-section 1, Article CO 36.

[18] P. K. Saha, G. Borgefors, and G. S. d. Baja, *Skeletonization: Theory, Methods and Applications*. Academic Press, 2017.

[19] R. Fabbri, L. D. F. Costa, *et al.*, "2d Euclidean Distance Transform Algorithms: A Comparative Survey," *ACM Comput. Surv.*, vol. 40, no. 1, pp. 2:1–2:44, 2008.

[20] T. Y. Zhang and C. Y. Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[21] Z. Guo and R. W. Hall, "Parallel thinning with two-subiteration algorithms," *Communications of the ACM*, vol. 32, no. 3, pp. 359–373, 1989.

[22] H. Blum, "A Transformation for Extracting New Descriptors of Shape," in *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn, Ed. Cambridge: MIT Press, 1967, pp. 362–380.

[23] M. J.F.M. (Johan), "DINED - anthropometric database," 2018.

[24] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[25] M. Stilman, K. Nishiwaki, *et al.*, "Planning and executing navigation among movable obstacles," *Advanced Robotics*, vol. 21, no. 14, pp. 1617–1634, 2007.

[26] M. Levihn, M. Stilman, and H. Christensen, "Locally optimal navigation among movable obstacles in unknown environments," in *IEEE-RAS*, 2014, pp. 86–91.

[27] M. Levihn, J. Scholz, and M. Stilman, "Planning with movable obstacles in continuous environments with uncertain dynamics," in *IEEE-ICRA*, 2013, pp. 3832–3838.