Robotics and Computer Integrated Manufacturing 63 (2020) 101913

Contents lists available at ScienceDirect

Robotics and Computer Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Handling and pushing objects using unmanned guided vehicles

M.B. Quemelli[a], A.S. Brandão[*,a,b]

[a] Post-graduate Program in Computer Science, Department of Informatics, Universidade Federal de Viçosa, MG, Brazil
[b] Nucleus of Specialization in Robotics, Department of Electrical Engineering, Universidade Federal de Viçosa, MG, Brazil

ARTICLE INFO

Keywords:
Bézier curves
Dijkstra's algorithm
Path planning
Laser scanner data

ABSTRACT

This paper aims handling box-shape objects combining mapping, searching, and path planning techniques. The proposal enables a mobile robot to push objects autonomously from random positions to a final destination. Laser scanner data are used to build up a 2D map, which aids the objects' identification in the scene. Next, a topological map is created and Bézier curves provide suitable paths taking into account the position of the robot, objects and final destination. Then, Dijkstra's algorithm finds the optimal route. Finally, simulations are run in V-REP + Matlab, and real experiments validate the proposal, which demonstrates quite efficient for environments without occlusion of the objects to be transported.

1. Introduction

Autonomous systems are capable of transporting objects and it can be extremely efficient in multiple application with high economic and social impact; as examples, waste recovery and disposal, demining or operations requiring object handling in environments where direct human intervention is impossible or impractical.

In such a context, the task of object handling is sometimes not trivial and it becomes unenviable to design analytical models capable of observing the full complexity of interactions between environment, objects, and robot. Pushing is one of the many handling alternatives that might be the most appropriate depending on the constraints of the task, whether the physical properties of the object or even the robot itself (i.e. the lack of a grip) [14]. Besides, pushing-only tasks require a considerable amount of action coordination to sustainable transportation. In other words, the robots must manage variables such as friction and dynamics to adjust the direction of movement, defining a suitable route and then execute the task [24].

Many approaches in literature use Unmanned Guided Vehicles (UGVs) to perform pushing and dislocating tasks, such as (i) a robot group use a behavioral approach to handle a single object [7], (ii) a single robot performs the handling of a complex shaped material, to drive a rolling ball along with a given path [10], (iii) three disc-shaped robots execute a manipulation of a polygonal object [21], (iv) a gripping and tilting robots are used to transport objects which are not grasped, by loading objects onto hand cart, turning the carrying task less friction [18], (v) multiple mobile robots using implicit communication to coordinate a box-carrying task [16], (vi) a group of mobile robots called m-bots use strategy based on tightening a payload [4].

To achieve successfully a handling task, a UGV should know at least one safe route before executing any maneuver. In other words, we commonly require a map representation and path planning strategy to provide deliberative navigation for a UGV. For instance, in [3] multiple topological representation stores the reachable relevant places in order to generate task planning. Besides, Fuzzy grid maps assist a safe navigation while avoiding obstacles [12]. Further, in [19] a piecewise Bézier curve provides a collision-free path. As well, in [22] Ant system and Dijkstra's algorithms result in an optimal path for deliberative navigation. Following, in [25] Dijkstra and A* algorithms introduce a cost-to-go function, providing flexibility through choosing the optimal path, resulting in improved performance.

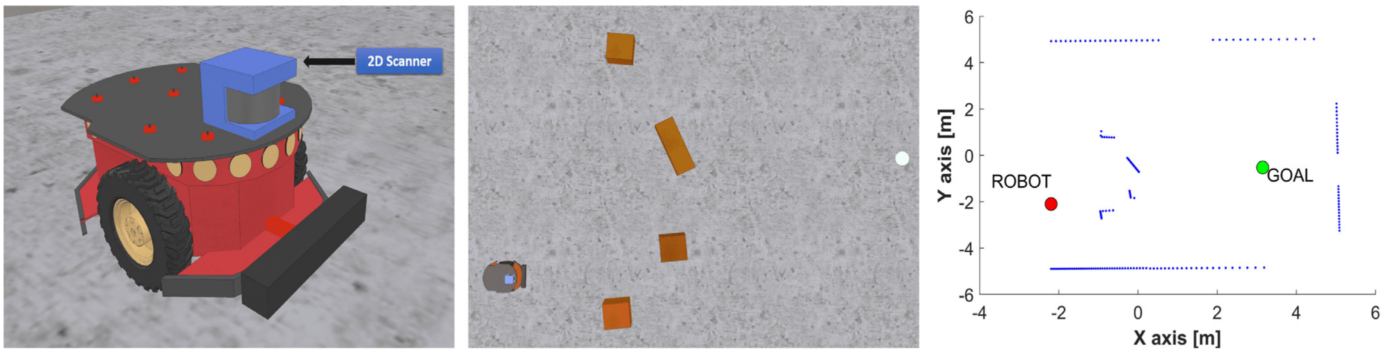
This proposal was motivated by some broad applications that require object transport in scenarios such as.

- In agricultural environments, considering a structured scenario, where an autonomous agent is responsible to find objects in the farm field, and then leading it to a warehouse;
- In companies' warehouse that require logistics activities, such as post offices, supermarkets, and general department stores that receive their products in boxes or pallets;
- In processes that require product separation, where it is mandatory to detect objects and distinguish them from its size and shape.

All the techniques presented in this research are already consolidated in the current literature. Meanwhile, from the best of our knowledge it is the first paper to deal with scene representation using

* Corresponding author at: Post-graduate Program in Computer Science, Department of Informatics, Universidade Federal de Viçosa, MG, Brazil.
E-mail addresses: matheus.quemelli@gmail.com (M.B. Quemelli), alexandre.brandao@ufv.br (A.S. Brandão).

https://doi.org/10.1016/j.rcim.2019.101913
Received 25 April 2019; Received in revised form 14 November 2019; Accepted 22 November 2019
Available online 02 December 2019
0736-5845/



(a) Laser scanner mounted on the robot. (b) Top view of V-REP environment. (c) Geometric map in the Cartesian plane.

Fig. 1. V-REP environment and the geometric map in Cartesian coordinates created in Matlab platform.

laser scanner data, Dijkstra's algorithm, and Bézier curves in order to achieve the goal of handling and pushing objects.

This work presents an approach to detect objects using laser scanner data, besides moving them using a mobile robot. The proposed method is similar to the Sokoban Game, however, in our work the agent can freely move to all directions instead of only four movements (Manhattan displacement). It is worth mentioning that this paper will not concern about the robot dynamic model, neither the control law and its stability proof. We consider them as a solved problem and we adopted the proposal presented in [13]. Furthermore, the paper contribution is twofold: a strategy to determine the posture of objects in the environment, and an optimized path planning strategy based on Bézier curves and Dijkstra's algorithm for handling and pushing objects.

Our strategy runs whenever there is no object occlusion, i.e., our constraint considers the robot might "see" all distinct objects in the structured scene at once. In addition, it is assumed that there is sufficient space for approaching and pushing maneuvers. All the irregularities on the floor are minimal in such a way it does not trap the objects' movement.

The rest of the manuscript is organized as follows. Section II presents the theoretical content that allows the environment representation in geometric form, and thereafter to the topological form. Section III describes the methodology stressing the algorithm to find the shortest path. Section IV exhibits the simulations and real experiments to validate the proposed methodology in two scenarios, emphasizing how the position, orientation, and size of the objects modify the definition of the optimal path to accomplish the pushing task. Finally, Section V presents the concluding remarks and some suggestions for future works.

2. Scene representation: geometric and topological maps

The task of moving objects commonly requires a description of the environment, which raises the definition of the most relevant features in the workspace, such as walls, areas, doors, wait-points, docking-points for loading or unloading, and dangerous zones. Generally, two-dimensional maps provide enough information for most applications involving UGVs, although some descriptors are required to effectively accomplish handling and carrying tasks, such as localization and shape of the objects.

The main problem discussed in this section is the identification and characterization of boxed objects in a structured environment without occlusion among them.

In the literature, there are some well-known techniques to describe a workspace, such as grid-based [6,8,9,20] and topological approach [1,11,15,23]. In the first case, the environment is represented by a matrix, whose cells are region spots in the real scenario, and its value indicates its occupancy probability (or even the presence or absence of an obstacle). Nevertheless, this technique presents drawbacks regarding storage space required for big scenarios or small grid size and computational time to update new information. In contrast, topological maps use graphs to represent an environment in a compact way, where each node can indicate a situation, place, or landmarks. Such an approach is more recommended for high-level planners, which is the case of this work. Sometimes, it is possible to apply a hybrid methodology relying on an original multi-layer environment model containing geometrical, topological and semantic layers as it is demonstrated in [2].

The steps to create a topological map commonly are: (i) get the sensor data, (ii) identify and feature the objects in the scenario, and (iii) represent this information in a graph.

The stage of object detection and recognition is commonly performed for computer vision methods, whether using CCD or depth cameras. In [17], a 2-D laser scanner was attached to the end-effector of a robot manipulator in an eye-in-hand setup, in order to create an image representation of the scenario. In our case, the laser scanner is mounted on a Pioneer 3-DX mobile robot and provides 181 distance measurements of its frontal view (as shown in Fig. 1(a)).

In general, a geometric map is the Cartesian representation of the polar laser data measurements, taking into account the robot pose. Fig. 1(b) illustrates the robot, the sensor and four box-shaped objects in V-REP simulator environment, while Fig. 1(c) shows the robot view in the Cartesian plane.

Assuming that the posture of any entity is described by $\mathbf{x} = (x\ [\mathrm{m}], y\ [\mathrm{m}], \psi\ [^\circ])$, where ψ is its heading with respect to the $x-$axis, from Fig. 1(c), we notice the robot is at $\mathbf{x}_r = (-2, -2, 0)$, the cloud point between $(-2, -4, 0)$ and $(0, 2, 0)$ represents the box-shape objects, and other points are associated with the walls that limit the workspace.

Once having the geometric map, a procedure searches patterns to identify the objects. First, we get the sensor data in polar coordinates (Fig. 2 top), and then calculate the difference between two consecutive measurements (Fig. 2 bottom). One can observe that the discontinuities in the laser information are highlighted on its "derivative". Thus, we assume that a negative peak followed by a positive one represents an object in the environment. In summary, whenever two consecutive angular measurements are lower (or greater) than a threshold, there is an object closer (or farther) from the robot.

In some cases, the scanner laser data provides information on two faces from the same object; thus, we need to check which is the appropriate between them to perform the handling and pushing task. In summary, we look for the face that results a smoother path from the robot to the object, and then from the robot-object to the destination.

Once finding an object, we also have its beginning and ending angles at the laser scanner data. Observe on the top of Fig. 2, the angular intervals $O_1 \in [-26^\circ, -10^\circ]$, $O_2 \in [6^\circ, 15^\circ]$, $O_3 \in [31^\circ, 45^\circ]$ and $O_4 \in [60^\circ, 68^\circ]$ contain the four objects.

If only one object's face is detected, the angular range describes a monotone function. However, if the function in this interval has a local

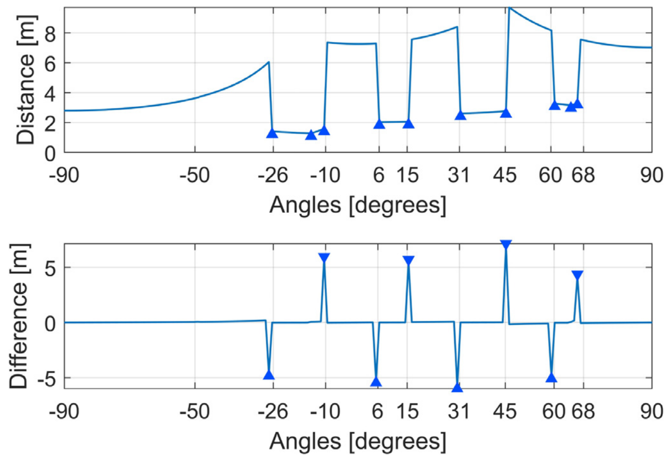


Fig. 2. Searching for objects through the laser measurement data.

minimum, then two faces are being observed. Notice that only one face is displayed in O_2 and O_3, while two faces are observed in O_1 and O_4. Examining each object, we find that O_1 and O_4 have a local minimum located at -13° and 66°, respectively. In contrast, monotonous functions describe O_2 and O_3.

As a result, O_1 presents the faces $F_{1a} \in [-26^\circ, -13^\circ]$ and $F_{1b} \in [-13^\circ, -10^\circ]$, while O_4 presents the faces $F_{4a} \in [60^\circ, 66^\circ]$ and $F_{4b} \in [66^\circ, 68^\circ]$. Thereby, in case of two faces, it is necessary to adopt that one closest to orthogonality in relation to the straight segment from the robot to the destination. In the sequence, we determine the face midpoint, from now on labeled the robot-box contact point.

Finally, after obtaining the midpoints of each object face from the laser scanner data, it is necessary to create a list of object face midpoints including their respective positions. Summarizing, the list represents strategic points for the topological map formulation, where each item in the list indicates a node in the graph.

3. The path planning strategy

The next step after getting the list of interest points is to create a graph connecting them. First, we define the robot's home position and the box delivery position, as start and end nodes, respectively. Fig. 3(a) illustrates a scenario with four box-shaped objects, as well as the four possible paths connecting the robot to the destination, passing by each object. In such a case, the solid red line indicates the shortest path.

Notice any path connecting the robot to an object or an object to its destination is not always a straight line; instead of it, Bézier curves are used to connect two nodes smoothly. It is worth mentioning that such smoothness is required because the robot does not have a gripper to attach an object to itself. In turn, the robot manipulates the object just by controlling its point of contact. A similar approach can be observed, for instance, in maneuvers performed by tractors or bulldozers during garbage or rubbish removal tasks [5].

In this work we adopt a quadratic Bézier curves given by

$$B(t) = (1-t)^2\mathbf{p}_0 + 2t(1-t)\mathbf{p}_1 + t^2\mathbf{p}_2, \text{with} 0 \le t \le 1, \tag{1}$$

and $\mathbf{p}_i$ being the control points. The edge weights consequently are not the Euclidean distance between two nodes, but the arc length of each Bézier curve, given by

$$D = \int_0^1 \sqrt{\left(\frac{d}{dx}B(t)\right)^2 + \left(\frac{d}{dy}B(t)\right)^2}\quad dt, \tag{2}$$

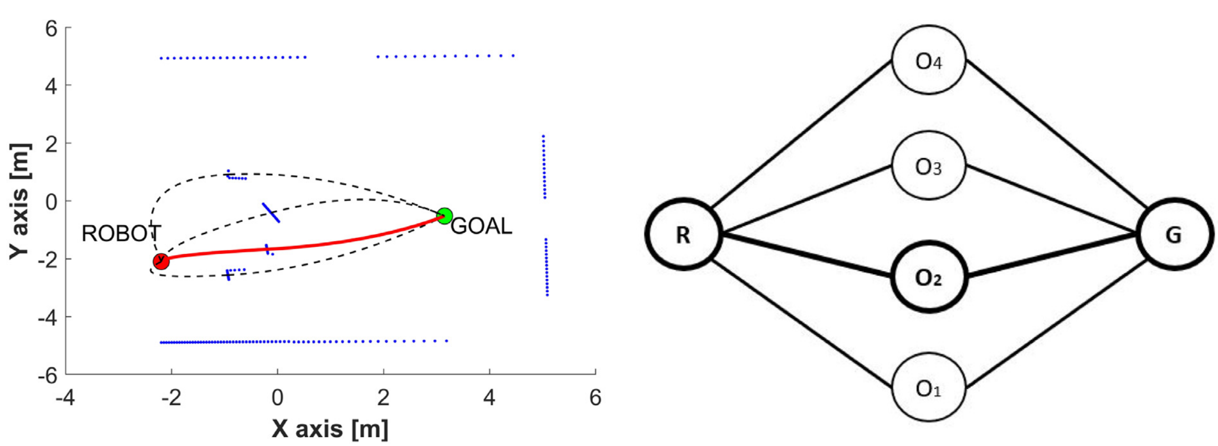
where D is the arc length of each parametric curve.

Besides, each maneuver requires handling and pushing stages. In other words, the robot has to travel towards the object and touch it, then guarantee it will not miss robot-box contact during their displacement towards the destination. Mathematically, two Bézier curves are required. The first one indicates the robot-object path, where the start node $\mathbf{p}_0$ is the robot position, the auxiliary point $\mathbf{p}_1$ is in some place of an orthogonal line that intersects the opposite box surface (see Fig. 4), and finally, $\mathbf{p}_2$ is the midpoint at the opposite box surface. Further, the second curve starts at the last point of the first curve, it has the mirrored $\mathbf{p}_1$ as its auxiliary point, and it ends at the goal point.

Once completing the graph representation, we should find the shortest path to handle and push each object to its destination. Thus, Dijkstra's algorithm calculates the best route taking into account the robot's home position, the goal localization, and the position of all objects as they are "seen" by the robot.

Figs. 6 and 8 show a sequence of all the pushing task, as well as all possible paths for each situation. Notice that initially the robot, the boxes and the destination are at the same position in both cases; but the boxes have different orientations. The solid red line highlights the shortest path calculated by Dijkstra's algorithm.

Algorithm 1 describes and solves the proposed handling and pushing task. In summary, each sub-task has its priority and the robot has to accomplish them according to a priority queue defined by the Dijkstra's algorithm. For instance, taking a scenario with four objects as shown in Fig. 5, the initial step is to compute all possible routes to the destination. Then, after selecting the best one, the chosen box associated to it is delivered. Notice here that the spot where the box was is



(a) Bézier curves connecting the robot, objects, and goal.
(b) Graph representation for four objects.

Fig. 3. Geometric map and its graph representation.

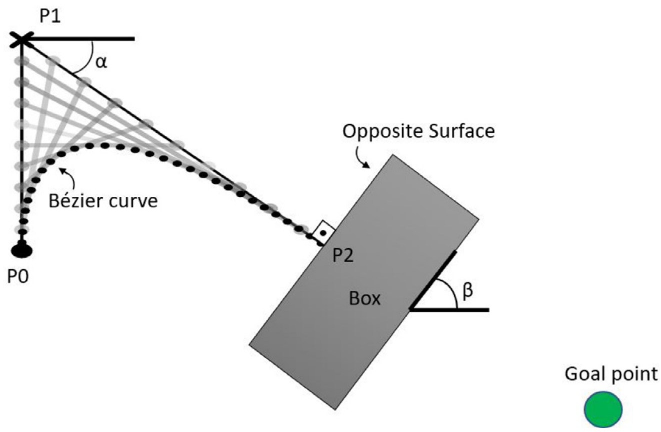


Fig. 4. Quadratic Bézier Curve.

now an empty and free-collision region; so, the algorithm can choose it as the shortest and the more efficient route, in such a case, a straight line. Thus, after pushing the first box, the algorithm computes a new set of possible routes and searches for the shortest one. The routine repeats until the last path is planned and all objects from the list is transported.

For a quantitative comparison, we define our proposal as the optimized strategy (POS) and the rest of the possible solutions as a non-optimized strategy (NOS). Any solution set can be described as pushing and returning maneuvers with $n.(n-1).\cdots.2.1 = n!$ and $1.2.\cdots.(n-1) = (n-1)!$ possibilities, respectively, being n the number of objects. Our proposed algorithm finds the optimal solution in the set of all possible paths. To contrast it, we choose one of all NOS possibilities, where the robot picks up the rightmost box and then moves one by one to their destination as shown in Fig. 7.

Finally, it is important to point out that the whole task of searching for objects and computing paths is performed before starting navigation, so the robot has free processing time to calculate the optimal path to be followed.

4. Results and discussion

This section presents the results of numerical experiments ran in V-REP + Matlab environment and practical tests performed for the method evaluation, applying to a concrete use case as separation of objects. The UGV used to perform the handling and pushing tasks is the Pioneer 3-DX, a differential drive robot, whose dynamics model and control law to guide it are described in [13]. Knowing the robot limitation, we adopt 0.4 m s^{-1} as a maximum suitable linear velocity to follow the Bézier curves.

Two scenarios shown in Figs. 9 and 10 are discussed here. In both of them, the robot and the destinations are initially at the same position, but the boxes have different positions and sizes. The purpose here is to separate small and large boxes to two different destinations. Imagine that in a post office, small boxes are destined for a different place than large boxes, so the robot can assist with the sorting task. Figs. 9 and 10 illustrate the steps to accomplish the pushing and sorting task on V-REP simulator. The black trace represents the route traveled by the robot, which is the shortest one computed by Dijkstra's algorithm in each scenario.

To clarify even more the proposed strategy, a video of the numerical experiment is available on NERO UFV Channel on YouTube through the link: https://youtu.be/6NTg9uzCNek. Also, to validate the method presented, we successfully performed real experiments which can be check on the link: https://youtu.be/t2eYq1xdoBc.

It is important mentioning our strategy has been validated in several scenarios with many objects since they are no occluded. We decide to present an environment with only four objects, because it is more didactic and makes easy explanation of each stage of our proposal.

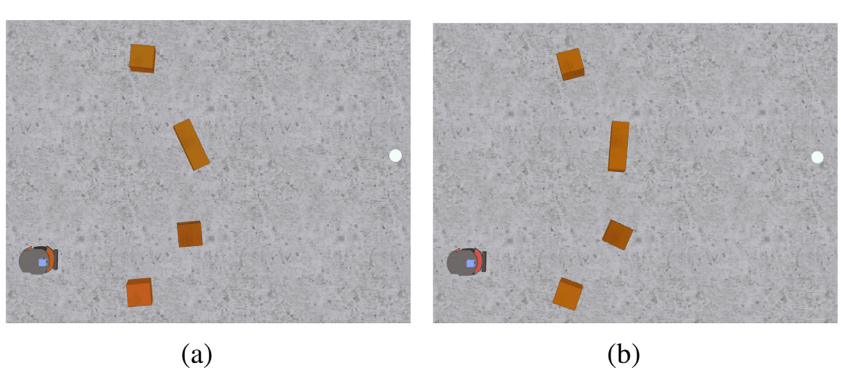
Furthermore, if NOS is considered, we can assume the objects are transported according to Fig. 7. In other words, the first object to be pushed is the rightmost, and the last one is the leftmost. Then, contrasting it against the POS applied on Scenarios 1 and 2 from Figs. 6 and 8, respectively. Comparatively, we verify our approach presents a shorter traveled distance and lower time spent, as stressed in Table 1. The savings in distance and time increase as the number of objects in the scenario also increases. At the same time, we say that the problem of performing the shortest path to the object, and then to the final destination, is similar to the traveling salesman problem. The robot needs to achieve the task by spending less energy as possible going through the optimized path.

Besides, from a qualitative perspective, it is possible to check that once changing the box orientations, the distance and time spent also alter. Thus, the optimal path depends on the box orientation. Nonetheless, it is also worthwhile to stress that the posture of the objects is defined when the robot "sees" them through the laser scanner data at the first time, before starting moving.

5. Conclusion remarks and future works

This manuscript presents techniques of mapping, searching and path planning, which enabled the mobile robot to accomplish box-shaped objects handling and pushing tasks. First, a UGV explores the environment where it is and builds a geometric 2D map, then a topological map is created, finally, Dijkstra's algorithm searches for the best robot-object-destination path. In summary, our strategy provides an optimal path through the graph's theory, where the edge weights are given by the arc lengths of the Bézier curves.

As an additional contribution, we also formulated a strategy to find and characterize box-shaped objects using laser scanner data, which proved feasible in scenarios without object occlusion; although we believe it works whenever information about objects in the workspace is continuously updated.



(a) (b)

Fig. 5. Simulation scenarios.

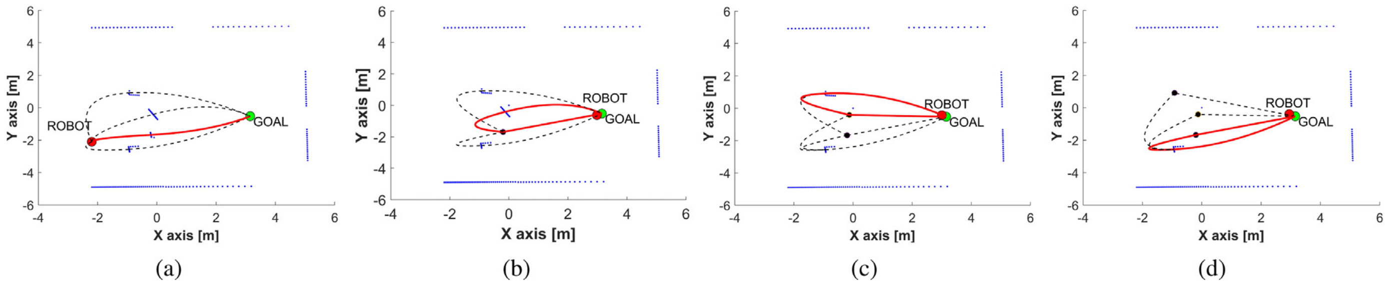


Fig. 6. Dijkstra's algorithm working in the Scenario 1.

```
P ← Pioneer3DX                                        ▹ Pioneer Robot class constructor
Connect(Matlab, VRep)                                 ▹ Connect Matlab to V-Rep

Map ← P.GetLaserData                                  ▹ Get Laser Data to build the Map
ObjectList ← ObjectSearching(Map)                     ▹ Find the objects vertices through the Map

while ObjectList do                                   ▹ While Object List is not empty
    Paths ← BézierPath(Robot position,Object position,Goal position)
    OptimalPath ← Dijkstra(Paths)
    Queue ← PriorityQueue.Insert(OptimalPath)
end while

while Queue do                                        ▹ While Queue is not empty
    Path ← PriorityQueue.Pop(Queue)
    Navigation(Path)                                  ▹ Navigation Task
end while
```

Algorithm 1. Material Detection and Transport.

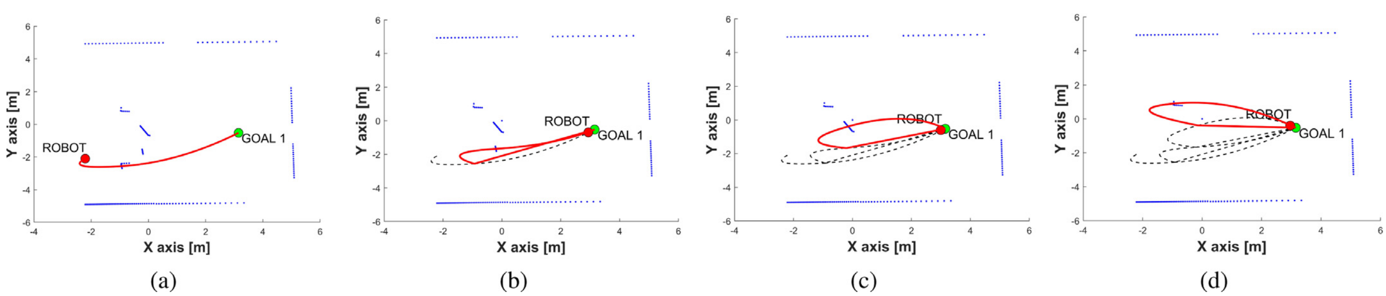


Fig. 7. Non-Optimized Strategy working in the Scenario 1.

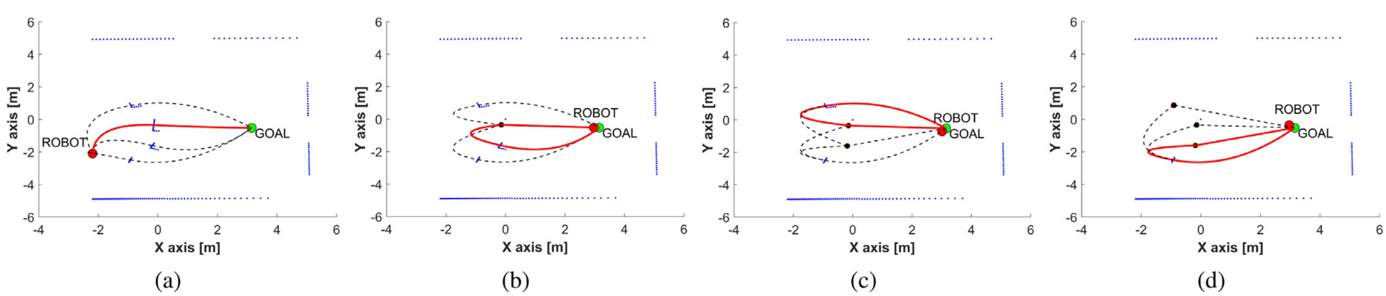


Fig. 8. Dijkstra's algorithm working in the Scenario 2.

Finally, the next steps intend to add more agents in such a manner they can move several small objects faster or help one another to move large materials collaboratively. It is important to remark that our strategy works on these situations if a centralized unit manages the queue priorities of the agents to prevent that two or more UGVs are sent to the same object unnecessarily.

CRediT authorship contribution statement

M.B. Quemelli: Data curation, Software, Validation, Visualization, Formal analysis, Writing - original draft, Writing - review & editing. **A.S. Brandão:** Conceptualization, Methodology, Supervision, Writing - review & editing, Funding acquisition.

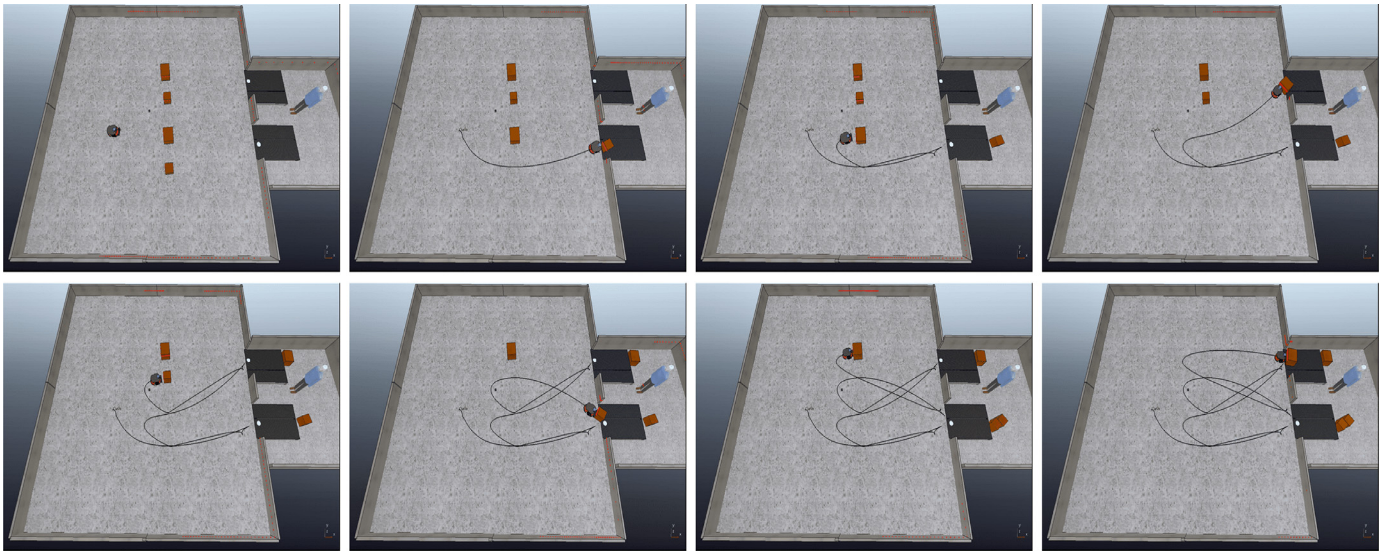


Fig. 9. Snapshots of the steps in Scenario 1.

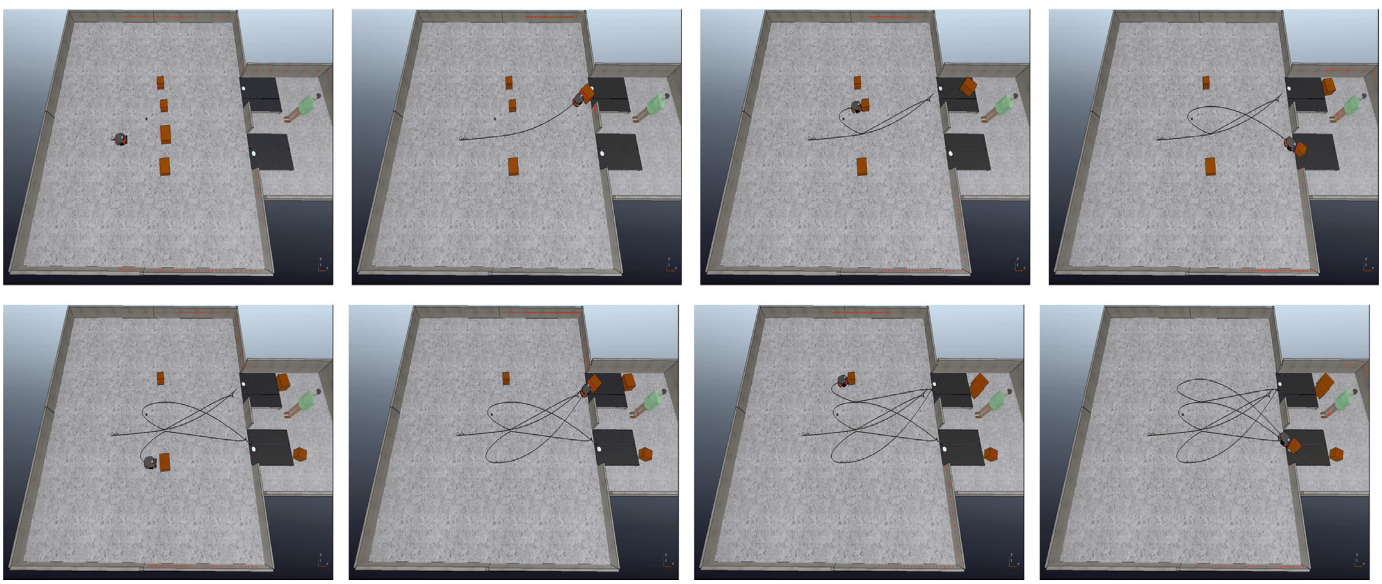


Fig. 10. Snapshots of the steps in Scenario 2.

Table 1
Scenario comparison: Proposed Optimized Strategy (POS) vs Non-Optimized Strategy (NOS).

	Distance travelled (m)		Time spent (s)	
N of boxes	POS	NOS	POS	NOS
4	36.39	36.94	90.94	92.31
5	37.11	38.21	92.73	96.09
6	38.23	41.55	95.53	103.82
7	40.19	46.87	100.43	117.12

Declaration of Competing Interest

We wish to draw the attention of the Editor to the following facts which may be considered as potential conflicts of interest and to significant financial contributions to this work.

Acknowledgments

This work was supported by CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico (Grant 409905/2016-0), a Brazilian agency that supports scientific and technological development and FAPEMIG - Fundação de Amparo á Pesquisa de Minas Gerais (Grant TEC - APQ-03469-16), an agency of the State of Minas Gerais, Brazil, for scientific development. Mr. Quemelli thanks CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Grant 88882.437225/2019-01) for providing his scholarship.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.rcim.2019.101913

References

[1] F. Blochliger, M. Fehr, M. Dymczyk, T. Schneider, R. Siegwart, Topomap: topological mapping and navigation based on visual slam maps, 2018 IEEE International

Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 1–9, https://doi.org/10.1109/ICRA.2018.8460641.
[2] S. Cailhol, P. Fillatreau, Y. Zhao, J.-Y. Fourquet, Multi-layer path planning control for the simulation of manipulation tasks: involving semantics and topology, Robot. Comput.-Integr. Manuf. 57 (2019) 17–28, https://doi.org/10.1016/j.rcim.2018.10.010.
[3] D. Herrero-Perez, H. Martinez-Barbera, Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems, IEEE Trans. Ind. Inform. 6 (2) (2010) 166–180, https://doi.org/10.1109/TII.2009.2038691.
[4] B. Hichri, J. Fauroux, L. Adouane, I. Doroftei, Y. Mezouar, Design of cooperative mobile robots for co-manipulation and transportation tasks, Robot. Comput.-Integr. Manuf. 57 (2019) 412–421, https://doi.org/10.1016/j.rcim.2019.01.002.
[5] K.J.W.M. Hirayama M. Guivant J., Path planning for autonomous bulldozers, Mechatronics 58 (4) (2019) 20–38, https://doi.org/10.1016/j.mechatronics.2019.01.001.
[6] H. Jo, H.M. Cho, S. Jo, E. Kim, Efficient grid-based rao–blackwellized particle filter slam with interparticle map sharing, IEEE/ASME Trans. Mechatron. 23 (2) (2018) 714–724, https://doi.org/10.1109/TMECH.2018.2795252.
[7] A. Khozaee, A.H. Aminaiee, A. Ghaffari, A swarm robotic approach to distributed object pushing using fuzzy controllers, 2008 IEEE International Conference on Robotics and Biomimetics, IEEE, 2009, pp. 1117–1122, https://doi.org/10.1109/ROBIO.2009.4913157.
[8] Y. Kwon, D. Kim, I. An, S. Yoon, Super rays and culling region for real-time updates on grid-based occupancy maps, IEEE Trans. Robot. (2019), https://doi.org/10.1109/TRO.2018.2889262.
[9] B. Lau, C. Sprunk, W. Burgard, Efficient grid-based spatial representations for robot navigation in dynamic environments, Robot. Auton. Syst. 61 (10) (2013) 1116–1130, https://doi.org/10.1016/j.robot.2012.08.010.
[10] X. Li, A. Zell, Path following control for a mobile robot pushing a ball, IFAC Proc. Vol. 39 (15) (2006) 49–54, https://doi.org/10.3182/20060906-3-IT-2910.00010.
[11] L.B. Marinho, P.P. Reboucas Filho, J.S. Almeida, J.W.M. Souza, A.H.S. Junior, V.H.C. de Albuquerque, A novel mobile robot localization approach based on classification with rejection option using computer vision, Comput. Electr. Eng. 68 (2018) 26–43, https://doi.org/10.1016/j.compeleceng.2018.03.047.
[12] H. Martínez-Barberá, D. Herrero-Pérez, Autonomous navigation of an automated guided vehicle in industrial environments, Robot. Comput.-Integr. Manuf. 26 (4) (2010) 296–311, https://doi.org/10.1016/j.rcim.2009.10.003.
[13] F.N. Martins, M. Sarcinelli-Filho, R. Carelli, A velocity-based dynamic model and its properties for differential drive mobile robots, J. Intell. Robot. Syst. 85 (2) (2017) 277–292, https://doi.org/10.1007/s10846-016-0381-9.
[14] T. Meriçli, M. Veloso, H.L. Akın, Push-manipulation of complex passive mobile objects using experimentally acquired motion models, Auton. Robots 38 (3) (2015) 317–329, https://doi.org/10.1007/s10514-014-9414-z.
[15] K. de Meyer, O. Lemon, U. Nehmzow, Multiple resolution mapping for e cient mobile robot navigation (1997).
[16] G.A. Pereira, B.S. Pimentel, L. Chaimowicz, M.F. Campos, Coordination of multiple mobile robots in an object carrying task using implicit communication, Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), 1 IEEE, 2002, pp. 281–286, https://doi.org/10.1109/ROBOT.2002.1013374.
[17] A.M. Pinto, L.F. Rocha, A.P. Moreira, Object recognition using laser range finder and machine learning techniques, Robot. Comput.-Integr. Manuf. 29 (1) (2013) 12–22, https://doi.org/10.1016/j.rcim.2012.06.002.
[18] T. Sakuyama, J.D. Figueroa Heredia, T. Ogata, T. Hara, J. Ota, Object transportation by two mobile robots with hand carts, Int. Sch. Res. Not. 2014 (2014), https://doi.org/10.1155/2014/684235.
[19] K.R. Simba, N. Uchiyama, S. Sano, Real-time smooth trajectory generation for nonholonomic mobile robots using Bézier curves, Robot. Comput.-Integr. Manuf. 41 (2016) 31–42, https://doi.org/10.1016/j.rcim.2016.02.002.
[20] A. Singha, A.K. Ray, A.B. Samaddar, Grid-based UGV navigation in a dynamic environment using neural network, 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, 2018, pp. 509–514, https://doi.org/10.1109/ICIRCA.2018.8597389.
[21] A. Sudsang, F. Rothganger, J. Ponce, Motion planning for disc-shaped robots pushing a polygonal object in the plane, IEEE Trans. Robot. Autom. 18 (4) (2002) 550–562, https://doi.org/10.1109/TRA.2002.801049.
[22] G. Tan, H. He, S. Aaron, Global optimal path planning for mobile robot based on improved Dijkstra algorithm and ant system algorithm, J. Central South Univ. Technol. 13 (1) (2006) 80–86, https://doi.org/10.1007/s11771-006-0111-8.
[23] L. Tang, Y. Wang, X. Ding, H. Yin, R. Xiong, S. Huang, Topological local-metric framework for mobile robots navigation: a long term perspective, Auton. Robots 43 (1) (2019) 197–211, https://doi.org/10.1007/s10514-018-9724-7.
[24] E. Tuci, M.H. Alkilabi, O. Akanyeti, Cooperative object transport in multi-robot systems: a review of the state-of-the-art, Front. Robot. AI 5 (2018) 59, https://doi.org/10.3389/frobt.2018.00059.
[25] D.S. Yershov, S.M. LaValle, Simplicial Dijkstra and a* algorithms for optimal feedback planning, 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 3862–3867, https://doi.org/10.1109/IROS.2011.6095032.

Matheus Berger Quemelli holds a degree in Electrical Engineering from the Federal University of Viçosa (UFV, 2017). One semester studying Biomedical Engineering abroad at Hanze University of Applied Sciences (Groningen NL 2017). Master degree student in Computer Science working with applied robotics in cargo transportation tasks at UFV. Researcher member of the Núcleo de Especialização em Robótica (NERO).

Alexandre Santos Brandão holds a degree in Electrical Engineering from the Federal University of Viçosa (UFV, 2006), a Master's and Ph.D. in Electrical Engineering from the Federal University of Espírito Santo (UFES, 2008 and 2013), with emphasis on Automation, working in aerial and terrestrial mobile robotics, in the following themes: non-linear control and cooperative robotics. In 2014, he received a Doctorate in Control Systems Engineering from the Instituto de Automática (INAUT) of the National University of San Juan, Argentina. Since 2010, he is with the Department of Electrical Engineering, Federal University of Viçosa, Brazil, and coordinator of the Núcleo de Especialização em Robótica (NERO). Since 2016, he is a professor in the Post-graduate program in Electrical Engineering at UFES, and in Computer Science at UFV. His research interest is linear and nonlinear control, social robotics, human-machine interaction, navigation and formation control of unmanned aerial and terrestrial vehicles.