MATHEUS BERGER QUEMELLI

DETECTING AND TRANSPORTING OBJECTS BY PUSHING-ONLY APPROACH

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae.*

Orientador: Alexandre Santos Brandão

VIÇOSA - MINAS GERAIS
2020

Ficha catalográfica elaborada pela Biblioteca Central da Universidade Federal de Viçosa - Campus Viçosa

T

Quemelli, Matheus Berger, 1992-

Q3d Detecting and transporting objects by pushing-only
2020 approach / Matheus Berger Quemelli. – Viçosa, MG, 2020.

71 f. : il. (algumas color.) ; 29 cm.

Orientador: Alexandre Santos Brandão.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f.64-71.

1. Detectores ópticos. 2. Robótica. 3. Transporte de cargas. 4. Dijkstra, Edsger W. I. Universidade Federal de Viçosa. Departamento de Engenharia Elétrica. Programa de Pós-Graduação em Ciência da Computação. II. Título.

CDD 22 ed. 621.367

MATHEUS BERGER QUEMELLI

DETECTING AND TRANSPORTING OBJECTS BY PUSHING-ONLY APPROACH

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 24 de março de 2020.

Assentimento:

Matheus Berger Quemelli
Autor

Alexandre Santos Brandão
Orientador

I dedicate this work to those who walked with me until here.

Acknowledgment

First of all, I want to thank my advisor, Professor Alexandre Santos Brandão, for his availability and for the accompaniment he carried out during this work. He was of great importance during the execution of this project and my training in the master's degree program being an example to be followed.

To Universidade Federal de Viçosa for the welcoming and beautiful environment, for providing such great intellectual knowledge and for providing me with unforgettable moments.

To all the professors of the UFV's Department of Informatics and Electrical Engineering that somehow contributed to my intellectual and professional growth. I could not have done if it wasn't for their help and teaching.

To my life friends, my family, and my girlfriend. I am not going to name them, but true ones will know who they are. You are the best and at all times have been by my side, supporting me, making me laugh, and getting up when I fell.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

"Science is more than a body of knowledge. It is a way of thinking; a way of skeptically interrogating the universe with a fine understanding of human fallibility."

Carl Sagan

Abstract

QUEMELLI, Matheus Berger, M.Sc., Universidade Federal de Viçosa, March, 2020. **Detecting and transporting objects by pushing-only approach**. Advisor: Alexandre Santos Brandão.

Due to laser scanners' quality of information, there is a wide spectrum of applications for these sensors in indoor and outdoor, both structured and unstructured environments. Most works present their own case-specific modelling strategies, which are often similar, although presenting no unity or consensus among them. In light of this lack of formalism this dissertation presents an analytical approach for identification and localization of objects using laser sensors. Firstly, the contribution lies in formally defining a laser sensor measurements and their representation, the identification of objects, their main properties and their location in a scene. Secondly, this work presents handling box-shape objects combining mapping, searching, and path planning techniques. Laser scanner data are used to build up a 2D map, which aids the objects' identification in the scene. Thirdly and most important, the dissertation aims a robust algorithm for pushing objects, from random positions to a final destination. Our main contribution is the route recovery strategy, which reacts whenever the box transportation starts going out of the planned one. It provides robustness to objects rotation and slipping during their displacements, thus guarantees all of them are correctly delivered. Besides, a topological map is created by Voronoi Graph in order to avoid collisions and Dijkstra's algorithm finds the optimal route. Then, Bézier curves provides suitable paths taking into account the position of the robot, objects and final destination. 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.

Keywords: Reactive Object's Control. Voronoi Graph. Bezier curves. Dijkstra's algorithm.

Resumo

QUEMELLI, Matheus Berger, M.Sc., Universidade Federal de Viçosa, março de 2020. **Detectando e transportando objetos através de uma abordagem de empurre**. Orientador: Alexandre Santos Brandão.

Devido à qualidade das informações dos lasers escâneres, existe um amplo espectro de aplicações para esses sensores em ambientes internos e externos, tanto em ambientes estruturados quanto não estruturados. Com o intuito de formalizar, esta dissertação apresenta uma abordagem analítica para identificação e localização de objetos utilizando sensores a laser. Em primeiro lugar, a contribuição reside na definição formal de medições de sensores a laser e sua representação, a identificação de objetos, suas principais propriedades e sua localização em uma cena. Em segundo lugar, este trabalho apresenta o manuseio de objetos em formato de caixa, combinando técnicas de mapeamento, pesquisa e planejamento de caminhos. Os dados do escâner a laser são usados para criar um mapa 2D, o que ajuda na identificação dos objetos na cena. Terceiro e mais importante, a dissertação visa um algoritmo robusto para empurrar objetos, de posições aleatórias até um destino final. A principal contribuição é a estratégia de recuperação de rotas, que é acionada sempre que o transporte das caixas começa a sair do planejado. Ele fornece robustez à rotação e ao deslize dos objetos durante seus deslocamentos, garantindo assim que todos os itens sejam entregues corretamente. Além disso, um mapa topológico é criado pelo do diagrama de Voronoi para evitar colisões e o algoritmo de Dijkstra encontra a rota ideal. Em seguida, as curvas de Bézier fornecem caminhos adequados, levando em consideração a posição do robô, objetos e destino final. Por fim, simulações são executadas no V-REP + Matlab, e experimentos reais validam a proposta, que se demonstra bastante eficiente para ambientes sem oclusão dos objetos a serem transportados.

Palavras-chave: Controle Reativo de Objetos. Grafo de Voronoi. Curvas de Bézier. Algoritmo de Dijkstra.

List of Figures

List of Tables

List of abbreviations and acronyms

ABNT	Associação Brasileira de Normas Técnicas
CNPq	Conselho Nacional de Desenvolvimento
CAPES	Conselho Nacional de Desenvolvimento Científico e Tecnológico
FAPEMIG	Fundação de Amparo à Pesquisa do Estado de Minas Gerais
UFV	Universidade Federal de Viçosa
V-REP	Robot Simulator Development Environment
Pioneer 3DX	Mobile Robot
LiDAR	Light Detection and Ranging
SLAM	Simultaneous Localization and Mapping
CAD	Computer-Aided Design

Contents

1 Introduction

Robotics provides innovative solutions in the most diverse areas of activity, especially when talking about transporting objects, luggage, people, information or resources from one point to another. In this regard, the entire range of transport vehicles, including air and land, can be classified as subject to automation. Generally, robots are deployed in controlled industrial zones, but it is expected that they will soon be deployed in public areas with varying degrees of autonomy.

Designing a fully autonomous transportation system is a complex problem. The robot must be prepared to operate alone in dynamic or structured environments and must do so in a reliable and efficient manner. According to (TUCI; ALKILABI; AKANYETI, 2018), this field of research in the automated industry can be classified into at least three main lines of approach defined as pushing, grabbing and carrying strategies.

The push-only strategy is a method of transporting objects in which a force is applied to any surface of the item in order to move it. This type of strategy is mainly used by robots that are unable to pull objects, since they have no means of grabbing them. This approach may seem like a relatively simple method, however, it is on top of the challenges common to all transport techniques (for example, the alignment of forces required to initiate transport, etc.), and requires significant coordination of actions to sustain transportation.

In order to build a solid theoretical basis for the development of transport strategies, it is first necessary to deepen the knowledge in some concepts that facilitate the implementation and execution of path planning in structured environments. In this sense, Voronoi diagram demonstrates a type of spatial decomposition in which a geometric space can be divided into several subsets according to the distance of objects.

Figure 1 illustrates how space can be divided according to the distribution of objects on a map. The application of this geometric computation technique is useful when it is needed to calculate route paths without collision. The diagram

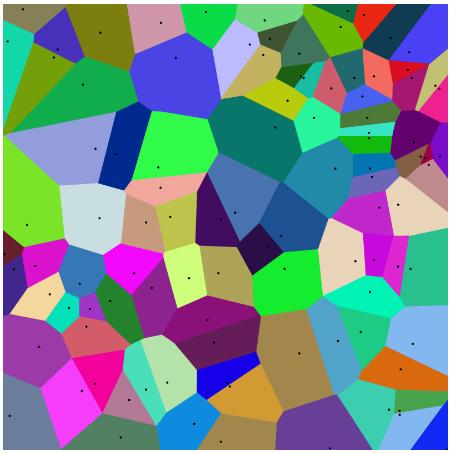


Figure 1 – Voronoi diagram of a random set of points on the plane.

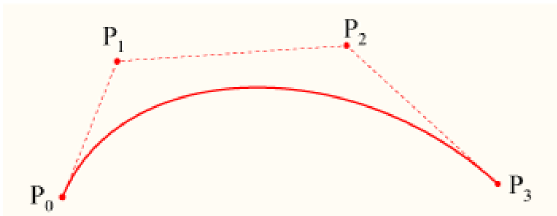


Figure 2 – Bézier curve applied for 4 points.

can be converted into a graph representation, each closed cell demonstrates edges and vertices that can be rewritten in the form of an Adjacency Matrix, making possible the application of search algorithms capable to find the path from point A to B.

Additionally, it is known that in robotics there are problems with respect to the perform of very abrupt paths, it is good to mention that energy expenditure also increases. Due to the need to calculate smooth trajectories, polynomial curves are usually applied to design the most suitable paths to be taken. Thus, the mathematical concept of Bézier Curves treats a polynomial path that a computer can draw easily. Bézier Curves use at least 3 points for its definition, reaching up to “n” control points. However, its most commonly form is the third order, i.e. the cubic curve of Bézier, which is defined by four control points such as Figure 2.

Now, linking the two geometric techniques, it is possible to perform navigation tasks and obstacle avoidance in a robust way.

1.1 The problem and its importance

Systems capable of transporting objects can be extremely effective in a variety of economic and social fields with great potential for impact. The proposal presented here is motivated by the countless applications in which a cargo transport robot would be useful, for example:

1. Agricultural environments, considering a structured scenario, in which an autonomous agent is responsible for finding objects in the farm field and then driving it to a warehouse.
2. In warehouses of companies that require logistics activities, such as post offices, supermarkets and general department stores that receive their products in boxes or pallets.
3. In processes that require product separation, it is possible to apply object detection techniques to distinguish object sizes and shapes.

The improvement of control techniques in robotics provided great advances in industrial processes and in other areas where high repeatability and precision was required. Thanks to the advancement of research in the field of robotics, today it is possible to produce on a large scale various products where previously human labor was needed. In the work proposed here the following advantages can be applied:

1. Replacement of human resources by autonomous robots, thus ensuring safety and integrity;
2. Ability to operate an autonomous system without pause, something impossible to do with human beings;

1.2 Hypothesis

Using techniques of mobile robotics control, mapping and the resources of graphs, it will be possible to define a strategy for the execution of the task of transporting cargo. More specifically Voronoi graph will create paths, and Dijkstra's algorithm will search for the best routes, thus the robot will be able to push objects to their destinations.

1.3 Objective

The main objective of this research project is the development of an algorithm based on graph theory to guide a robot so that it can perform a transport task. Specifically, the sub-objectives of this research are:

1. Use simulation software to test the tasks to be performed by the robots.
2. Analytical approach to identify and locate objects.
3. Define techniques and strategies for mapping and path-planning in order to achieve the objective of transporting objects.
4. Perform practical experiments in a controlled environment.

1.4 Dissertation Structure

This dissertation was elaborated and written according to the formats recommended by the Commission of the Computer Science Graduate Program of the Federal University Viçosa. The body of the text is organized as a collection of articles resulting from the development of the research. This first chapter presented a general introduction of the problem to be treated in this dissertation, as well as its hypothesis and objectives. It is important to mention that the introduction presents only some concepts that will be better explained in the following chapters.

The rest of the dissertation is structured as follows:

Chapter 2 describes the article "Object Detection using Laser Scanner: An Analytical Approach". The reported work refers to the use of the laser scanner dataset and formally define the measurements and their representation to identification of objects. As a classification tool, the paper deals with the problem of formalizing distance measurement and object detection with laser sweep sensors, defining strictly an object and some of its properties, applying such an approach and discussing its results and applications, regarding framework setting for object detection, localization and matching.

Chapter 3 entitled "Handling and Pushing Objects using Unmanned Guided Vehicles" presents the article published on the magazine Robotics and Computer-Integrated Manufacturing. The article proposes a method of identify objects based on the use of the mathematical study correspondent to the paper described on the Chapter 2. Also there is a proposal based on Bézier Curves and Dijkstra's algorithm to design a path to transport objects. We performed tests based on a software simulator called V-Rep, and in order to validate the method, we executed real experiments. The results obtained and presented in the article describe the execution of the experiments in a structured environment. The full reference to the article can be visualized and quoted as follows:

QUEMELLI, Matheus Quemelli; BRANDÃO, Alexandre Santos. Handling and Pushing Objects using Unmanned Guided Vehicles. In: **Robotics and Computer-Integrated Manufacturing**, volume 63, June 2020, pg 101913.

Chapter 4 is regarding to the third article already submitted to Robotics and Autonomous Systems. It presents the most important element of the research, named as "Detecting and transporting objects by pushing-only approach". The article presents an improvement on the work done in the article in chapter 3, adding techniques for approaching objects, Voronoi graphs, and a reactive control algorithm to push objects into the scene. The algorithm previously used in the second article presents some limitations that have been better addressed at this point. Another factor of great relevance in this work is the reactive control that enables the robot to return and retake the objects in case of loss of contact. In this sense, from the second article, the robot was not able to check if the object was performing the right route as calculated. In general, the third article describes the effective execution of the proposal reported in the second article, including some changes that created a robustness system of transportation.

Chapter 5 discusses the results obtained in the research in general, narrating some conclusions about the new line of research suggested and introduced by this work. The conclusion still reflects on how the use of group of robots for autonomous transport can impact the society in many ways. Finally some open research opportunities are left as future works on the field.

For organizational purposes, the references of the articles and other parts of the dissertation were assembled in a single Bibliography section.

2 Object Detection using Laser Scanner: An Analytical Approach

Abstract

Due to laser scanners' quality of information, there is a wide spectrum of applications for these sensors in indoor and outdoor, both structured and unstructured environments. Most works present their own case-specific modelling strategies, which are often similar, although presenting no unity or consensus among them. In light of this lack of formalism this manuscript presents an analytical approach for identification and localization of objects using laser sensors. Our main contribution lies in formally defining a laser sensor measurements and their representation, the identification of objects, their main properties and their location in a scene.

2.1 Introduction

Lasers Scanners are essential tools used for instrumentation and one of its great advantages lies in calculating depth, format and size of objects. In the current context, these devices have been used to enable robots to move independently, autonomously and intelligently through indoor settings such as factories corridors and warehouses (COSTA et al., 2016).

Among the various types of sensors that can be used in robotics, laser sensors are usually the most applied for time series (Wang et al., 2013), point clouds (Huang et al., 2015),(Schlarp; Csencsics; Schitter, 2019) and regular angular distance data (Li; Ruichek; Cappelle, 2013). This is due to the fact that it has a high degree of accuracy and a relatively low acquisition cost, as discussed in the literature (KONOLIGE et al., 2008).

A notorious example of a widely applied technique with the help of lasers scanners is Simultaneous Localization and Mapping (SLAM), the procedure of autonomously building a map while a robot is localizing itself in the environment (KRINKIN et al., 2018). Researches related to this topic within the field of mobile robotics have remained popular for a long time, and recently more effort has been made to contribute in the manufacture of intelligent and autonomous vehicles (BRESSON et al., 2017),(Gargoum; El-Basyouny, 2017).

Still with regard to positioning and navigation, there are applications in places such as electrical substations. The approach is made essential in this case due to the fact that optical sensors do not suffer interference from the large electromagnetic field exerted by machines in electrical plants (LU; ZHANG; SU, 2017).

Besides the approaches mentioned above, there are many others that motivate and drive this work's purpose. In particular the uses of laser scanners for object detection and tracking (including cases when both agent and objects are mobile) (MERTZ et al., 2013; AZIM; AYCARD, 2012; LINDSTROM; EKLUNDH, 2001), object identification and segmentation from local environment (LEHTOMÄKI et al., 2010), (YANG et al., 2015a) and object feature extraction (NUNEZ et al., 2006). These implementations have deep impact on autonomous robotics and decision making using little or no previous knowledge about the environment and objects, nevertheless accurately inferring information and executing tasks based on such data. Thus, it seems valuable and significant to propose and evaluate a formal mathematical definition for object detection and identification in tasks based on segmentation, tracking and feature extraction.

In yet another similar sense, SLAM implementations frequently require map and CAD model building and correction based on laser scanner data. Generally, many such techniques apply triangulation, environment landmarks (Schlarp; Csencsics; Schitter, 2019) and object features detection (Giri; Kharkovsky, 2016), for systematic odometry error compensation in both indoor (XIONG et al., 2013), (SHEN; MICHAEL; KUMAR, 2011), (BISWAS; VELOSO, 2012) and outdoor (WAKITA; NAKAMURA; HACHIYA, 2018), (ORIA-AGUILERA; ALVAREZ-PEREZ; GARCIA-GARCIA, 2018) data. Thus, detecting, locating and matching objects using a laser range scanner as well as point cloud self localization, map matching and correction present strong similarities in spite of no strict structural model definition connection, although presenting less computing cost than image processing and recognition (LINDSTROM; EKLUNDH, 2001).

Nonetheless, other fields also benefit from the use of laser scanners in both education and industry. In the agricultural automation industry for example, there is a variety of researches in the assessment of canopy volume (COLAçO et al., 2017), poplar biomass (ANDÚJAR et al., 2016) and trunks (BARGOTI et al., 2015), crop and weed distinction (ANDúJAR et al., 2013), among other uses. On a different view, the educational competition RoboCup Junior Rescue B (AKIN et al., 2013) has also gained from using laser sweep data as a means of developing and optimizing robot object rescuing competition strategies.

Given the wide array of applications based on and benefiting from LiDAR (Light Detection and Ranging). data, there is yet no rigid definition or analytical approach for the problem of generally detecting objects in semi-structured environments. Therefore, we see fit and necessary to propose a novel strict mathematical formulation and framework for detecting, identifying, matching and tracking objects based on laser scanner distance information, which is our main contribution.

In summary, the paper deals with the problem of formalizing distance measurement and object detection with laser sweep sensors, defining strictly an object and some of its properties, applying such an approach and discussing its results and applications, regarding framework setting for object detection, localization and matching. To address these topics, the paper is laid out on three main sections. First we define the scope and how to represent LiDAR scan measurements mathematically, in sequence, these definitions are used to define and infer properties from objects in a scene. Finally, a guideline for object detection and localization is set with an application, providing insight by applying these techniques, thus validating our proposal.

2.2 Object Identification and Localization

In robotics applications, a navigation environment is labelled according to object quantity and layout in the navigation scene, as well as the agents' freedom of movement. In this context, an environment known as structured is defined when the task-executing agent is previously familiar with the posture of any object and these do not suffer any changes during task completion. In counterpart, when objects move unpredictably as the agent is executing tasks, the environment is labelled unstructured. Finally, those environments in which a certain degree of object mobility is admissible, such as offices, laboratories, residences, storage houses and workshops are known as semi-structured environments.

In the specific case of semi-structured environments, entities in the navigation scene may be mapped by an agent using a distance sensor, which this work will consider to be a laser scanner sensor as a LiDAR. These entities may be fixed objects (as walls, shelves, wardrobes, etc.) or mobile objects (e.g. boxes or even other agents). In definitions to follow, the subscript k denotes a discrete set of elements and n an element belonging to such a set, thus both being discrete.

2.2.1 LiDAR sweep representation

Definition 1 *Let r be an application representing a LiDAR sensor, denoted:*

$$\begin{aligned} r\colon \Theta_k &\to \boldsymbol{D}_k \\ \theta_n &\mapsto D_n = r(\theta_n) \end{aligned}$$

where the domain Θ_k indicates a set containing each discrete angle within the angular scan range and the codomain $\boldsymbol{D}_k$ the set of distance measurements assigned to each angle θ_k. Such an application is shown in Figure 3.

Definition 2 *Let s be a difference function given by:*

$$\begin{aligned} s\colon \Theta_k &\to \boldsymbol{d}_k \\ \theta_n &\mapsto d_n = r(\theta_n) - r(\theta_{n-1}) = s(\theta_n) \end{aligned}$$

where analogously to Definition 1, θ_n is an element in the set Θ_k of all angles within the instrument's angular range and $\boldsymbol{d}_k$ a set of differences between two neighbouring measurements. This is depicted in Figure 4.

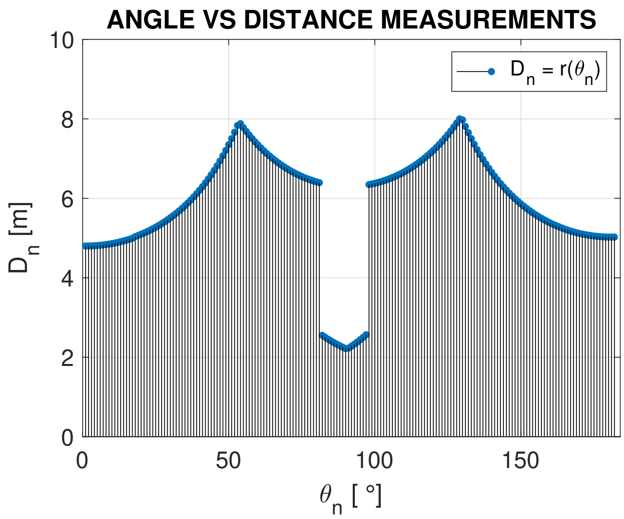


Figure 3 – Polar stem plot representing a laser's discrete distance scan.

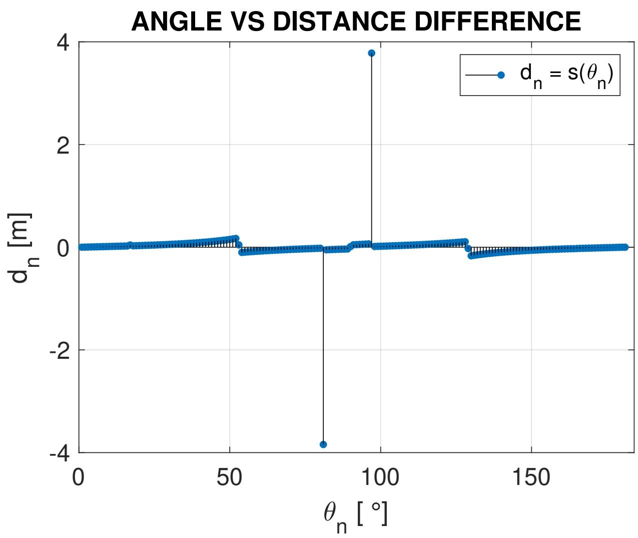


Figure 4 – Example of a $s(\theta)$ as in Def. 2, following from Figure 3.

Definition 3 *Let f be a function coinciding with $r(\theta_n)$ $\forall \theta_n \in \Theta_k$, that is:*

$$\begin{aligned} f\colon \mathbb{R} &\to \mathbb{R} \\ \theta &\mapsto D = r(\theta_n) \end{aligned}$$

such that f is also continuous and monotonic in intervals (θ_{n-1}, θ_n) for every

$n = 1, ..., N$, *whose one-sided limits are:*

$$\lim_{\theta \to \theta_n^-} f(\theta) = f(\theta_{n-1})$$
$$\lim_{\theta \to \theta_n^+} f(\theta) = f(\theta_{n+1})$$

whenever $|s(\theta_n)| > d_{th}$, *where* N *is the sensor's resolution and* d_{th} *is a case-specific threshold value (free paremeter) representing the minimal distance measurement difference for object detection.*

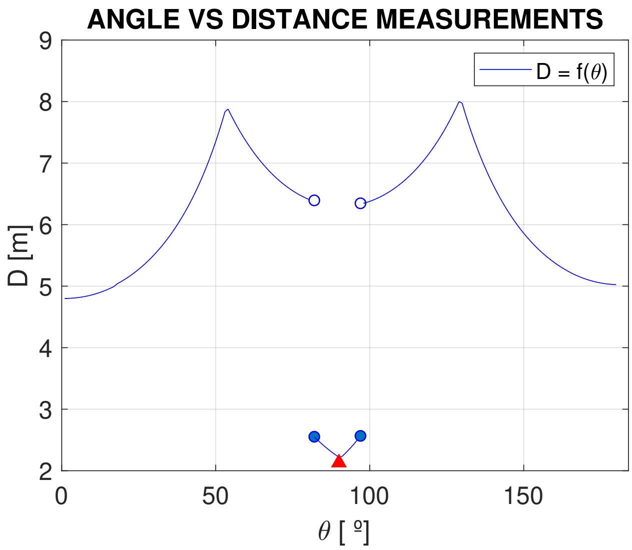


Figure 5 – Continuous representation of a laser's scan as in Def. 3.

Proposition 1 *Given a well-functioning LiDAR sensor,* $\forall \theta_n \in \Theta_k$, $\exists r(\theta_n) \,/\, D_n = r(\theta_n)$.

Proof 1 *The LiDAR sensor attributes a distance measurement reading for each angle within range, unless the sensor malfunctions or has manufacturing errors, which must then be assessed and corrected.*

Corollary 1 *Given that Proposition 1 is satisfied,* r: $\theta_n \to D_k$ *is surjective, by definition.*

Corollary 2 *Following from Proposition 1 and Corollary 1, f is surjective, by definition, since r coincides with f.*

Above, Definition 1 states how the agent visualizes its navigation surroundings. Notice that it follows from Corollary 2 and Definition 3 that f is differentiable over most of its domain, such that points where f is not differentiable have important properties, to be discussed when defining objects in a laser's scan data. Note that $\forall\theta \in [\theta_{\min}, \theta_{\max}]$ and $\forall D \in [0, D_{\max}]$, whose extreme values are specific to model and manufacturer of the sensor device.

2.2.2 Defining objects

If f is a continuous function, it may or may not be differentiable. However if f is differentiable in a, then f is continuous in a and it is laterally continuous $f'_-(a) = f'_+(a)$. In other words, the left-hand and right-hand derivatives in a must exist and have equal value, such that by applying the concept of differentiability, objects, walls and free space can be distinguished in a LiDAR scanner reading.

In particular, it follows that if there exists a point where $f(\theta)$ is not differentiable and that point does not belong to the interval of an object, then that must be an edge of a wall (corner), otherwise that point belongs to the edge of an object.

Definition 4 *Let $\boldsymbol{O}$ be any prism-shaped object in a semi-structured environment. Then $\boldsymbol{O}$ may be defined as a set of points in polar coordinates:*

$$\boldsymbol{O} = \Big\{\Big(\theta, r(\theta)\Big) \in \mathbb{R}^2 \;/\; \theta_i \leq \theta \leq \theta_f\Big\},$$
$$\forall\theta_{i,f} \;/\; f'_-(\theta_{i,f}) = f'_+(\theta_{i,f})$$

where $\theta_i < \theta_f$, such that $P_i = \Big(\theta_i, r(\theta_i)\Big)$ is a point of discontinuity and $P_f = \Big(\theta_f, r(\theta_f)\Big)$ is the first next point of discontinuity to the right-hand side of P_i, thus both encompassing start and final measurements of an objects' body. Thus, $f(\theta)$ is continuous in the open interval (θ_i, θ_f).

Consider a generic prismatic object and its respective polar coordinates comprised in $\mathbf{O}$. Notice that, in any such set $\mathbf{O}$, a discontinuity in the derivative of

$f(\theta)$ must represent an edge, as marked in Figure 3 with a red triangle. Therefore, we can define rigidly both faces and vertices that belong to **O**.

Definition 5 *Let **V** be a set of points representing any edge of any prismatic object, such that:*

$$\boldsymbol{V}_k = \Big\{\Big(\theta, r(\theta)\Big) \in \mathbb{R}^2 \;/\; f'(\theta) = 0, \theta \in \boldsymbol{O}\Big\}, k{=}0,1,2,\ldots,n$$

In other words, according to Definition 6, any of the prism's edges are found in a local minimum or maximum between two faces according to the laser's readings and all faces are found within (θ_i, θ_f).

Definition 6 *Let **O** be any prismatic object, then let $\boldsymbol{F}_k$ be a set of points representing the k-th face of such an object. Therefore, we define in polar coordinates:*

$$\boldsymbol{F}_k = \Big\{\Big(\theta, r(\theta)\Big) \in \mathbb{R}^2 \;/\; \theta_k \leq \theta \leq \theta_{k+1}\Big\}, k{=}0,1,2,\ldots n$$

where $\theta_0 = \theta_i$, $\theta_n = \theta_f$ and all $\boldsymbol{V}_k$ are in $(\theta_k, f(\theta_k))$.

Therefore, as in Figure 5, the function f(θ) is discontinuous in θ_1 and θ_2. From that, it is possible to define that every element $\theta \in [\theta_1, \theta_2]$ represents a measurement from the surface of an object (hereby defining all necessary conditions for proposing the existence of an object).

Note that **O** was defined as prism-shaped for the sake of defining faces and vertices, although the same discontinuity-based definition may be used to identify other - more unusually-shaped - objects. This could, for instance, improve formalism, notation and analysis in (Giri; Kharkovsky, 2016) without great computational effort. Similarly, (Wang et al., 2013) could benefit from notation formalism in point cloud temporal series as a means of data representation as a function of time and reference frame.

In yet another case-oriented illustration, (Li; Ruichek; Cappelle, 2013) presents a scenario where laser data are presented on the Cartesian plane for later use in extrinsic camera parameter calibration. Its worth reinforcing that our work could have been employed in all such cited situations as a guideline for laser sweep representation, region of interest highlighting in data and notation. For better illustration, a generic representative case will be presented in the next subsection.

2.2.3 Detection and localization

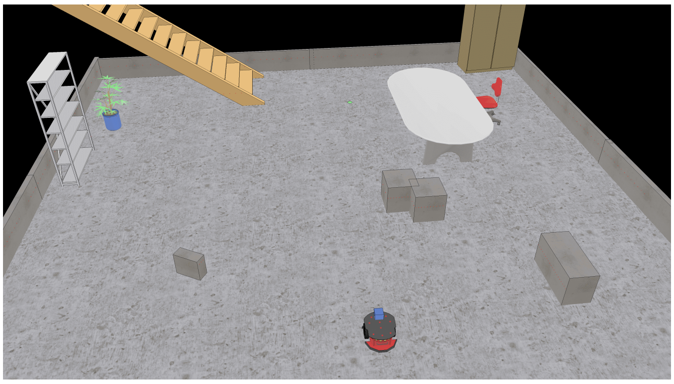


Figure 6 – Generic semi-structured environment with different objects and LiDAR-equipped robot.

The robot simulation environment V-REP was used to develop a means of illustration and validation shown in Figure 6: a 5m x 5m semi-structured environment containing several items, including three specific prismatic brown objects to be handled by the robot. Figure 7a exhibits a LiDAR sweep (r(θ_n), as previously defined), from which it is already possible to intuitively distinguish the highest values as walls and lower readings as objects, depending on how close each one is. All Figures have been generated algorithmically following definitions as presented in the previous subsection.

Upon further analysis, by comparing both Figures 7b, 7c as discussed and defined in Section B, various objects can be identified by setting a threshold difference value (as presented in Definition 3) in s(θ_n) and observing discontinuities in f(θ_n), e.g. a threshold equal to 10% of the greatest value of s, such that any discontinuities occurring for the same angle measurement where the threshold has been surpassed must represent the starting point of an object. Furthermore, local minima in each set representing an object must also represent the edge closest to the scanner, all of which are marked in Figure 5 with red triangles, whereas the

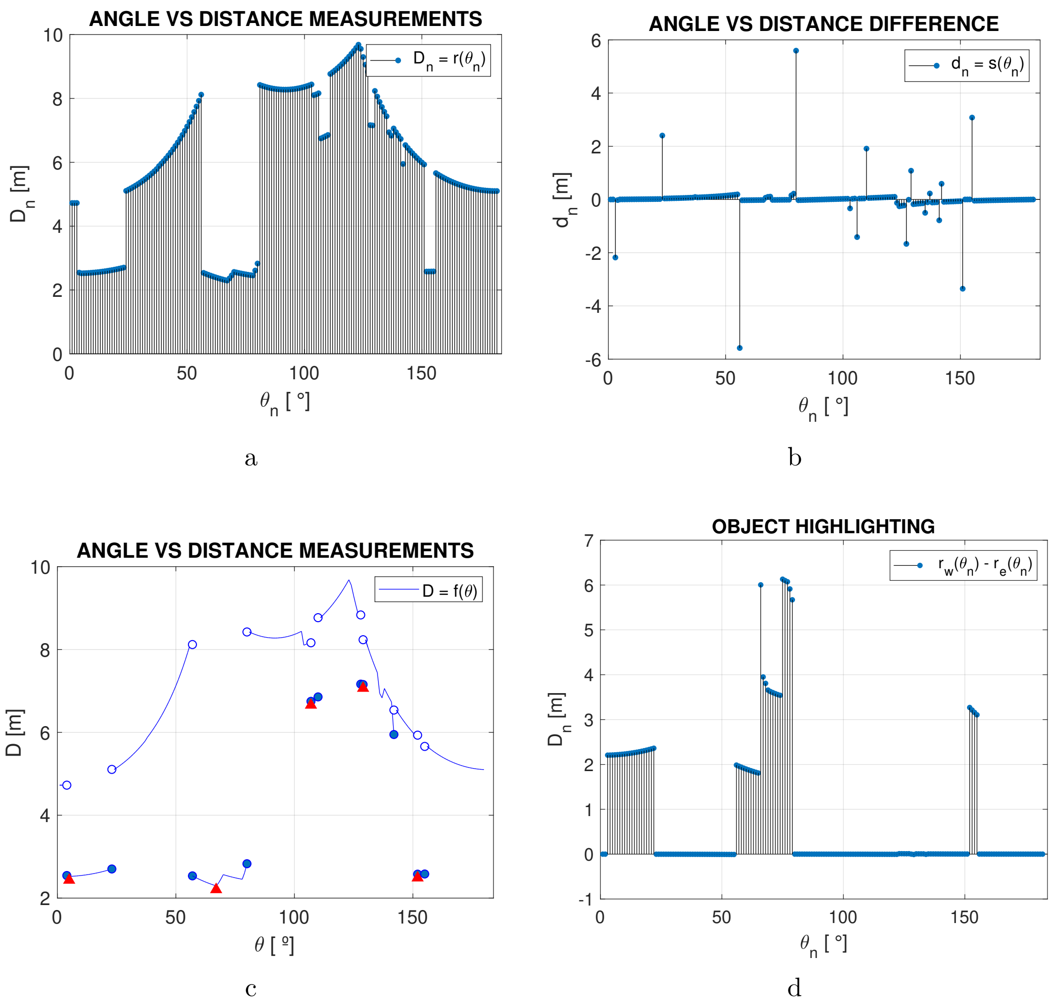


Figure 7 – Function of the LiDAR scan as defined, representing measurements from the semi-structured environment in Figure 6

objects' readings are shown between two dark-blue filled circles, thus exhibiting six objects.

Now, by comparing Figures 3 and 6, one can identify the objects marked in Figure 3 as (in order of appearance) the first and second brown prismatic boxes, the ladder, the potted plant, the metal shelf and the smaller brown box, as seen in Figure 6 in anti-clockwise order. From observing the environment using the laser's scan, the robot is therefore able to identify as well as know the distance from which object in the room.

Supposing the agent has a known start-point (e.g. recharging dock) or a map linking each laser sweep to a certain position, it is also possible to locate objects by storing measurements of the semi-structured environment without any objects of interest to the robot, i.e. no objects that should be handled by the agent, uninteresting objects. Then, one can highlight any new objects by taking the algebraic difference between readings before and after objects where placed, as show in Figure 7d, therefore locating all objects of interest in the environment. Given these features, it is possible to match and track specific objects throughout a scene, for instance, the three brown boxes where highlighted as an example of objects of interest.

2.3 Conclusions

Considering the current scientific effort for fast, accurate and autonomous robot decision-making as highlighted previously, the fundamental task of identifying and locating objects as well as distinguishing goal-essential objects in a scene is key for the further development of robotics. Despite the extensive array of applications, the lack of strict framework has established a possible bottleneck for algorithm comparison or optimization, strategy development and solution of various practical issues.

In view of these circumstances, our manuscript provides an infrastructure for the development and optimization of algorithms requiring autonomous object detection, localization and matching, presenting some essential properties of laser scanner data and guidelines for feature extraction from such measurements.

Despite the need of setting threshold for object detection, which is not generally automatic or dynamic in principle, we understand the definition of such a parameter must attend to the nature of each application, thus being a free variable. In counterpoint, the simple mathematical essence of this analytical approach guarantees low computational effort and efficiency. Overall, we hope the reader will be able to creatively develop solution proposals to a variety of cases, given our analytical structure yields coherent and effective mathematical formalism in view of detection, identification and localization.

3 Handling and Pushing Objects using Unmanned Guided Vehicles

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.

3.1 Introduction

Autonomous system are capable of transporting objects and it can be extremely efficient in multiple applications 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) (MERIÇLI; VELOSO; AKIN, 2015). 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 (TUCI; ALKILABI; AKANYETI, 2018).

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 (KHOZAEE; AMINAIEE; GHAFFARI, 2009), (ii) a single robot performs the handling of a complex shaped material, to drive a rolling ball along with a given path (LI; ZELL, 2006), (iii) three disc-shaped robots execute a manipulation of a polygonal object (SUDSANG; ROTHGANGER; PONCE, 2002), (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 (SAKUYAMA et al., 2014), (v) multiple mobile robots using implicit communication to coordinate a box-carrying task (PEREIRA et al., 2002), (vi) a group of mobile robots called m-bots use strategy based on tightening a payload (HICHRI et al., 2019).

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 (HERRERO-PEREZ; MARTINEZ-BARBERA, 2010) 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 (MARTÍNEZ-BARBERÁ; HERRERO-PÉREZ, 2010). Further, in (SIMBA; UCHIYAMA; SANO, 2016) a piecewise Bézier curve provides a collision-free path. As well, in (TAN; HE; AARON, 2006) Ant system and Dijkstra's algorithms result in an optimal path for deliberative navigation. Following, in (YERSHOV; LAVALLE, 2011) 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.

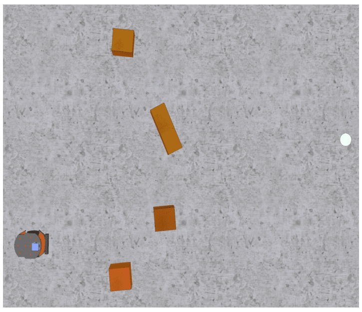
All the techniques presented in this research are already consolidated in the current literature. Meanwhile, from the best of our knowledge it is the first paper to deal with scene representation using laser scanner data, Dijkstra's algorithm, and Bézier curves in order to achieve the goal of handling 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 (MARTINS; SARCINELLI-FILHO; CARELLI, 2017). 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.

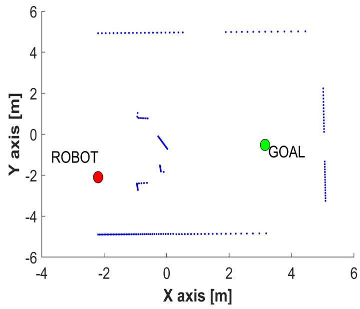
Our strategy runs whenever there is no object occlusion, i.e., our constraint considers the robot might "see" all distinct objects in the structured scene at once. In addition, it is assumed that there is sufficient space for approaching and pushing maneuvers. All the irregularities on the floor are 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,

a Laser scanner mounted on the robot.



b Top view of V-REP environment.



c Geometric map in the Cartesian plane.

Figure 8 – V-REP environment and the geometric map in Cartesian coordinates created in Matlab platform.

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.

3.2 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 (KWON et al., 2019; JO et al., 2018; LAU; SPRUNK; BURGARD, 2013; SINGHA; RAY; SAMADDAR, 2018) and topological approach

(MARINHO et al., 2018; TANG et al., 2019; MEYER; LEMON; NEHMZOW, 1997; BLOCHLIGER et al., 2018). In the first case, the environment is represented by a matrix, where the cells are region spots in the real scenario, and the 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 (CAILHOL et al., 2019).

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 by computer vision methods, whether using CCD or depth cameras. In (PINTO; ROCHA; MOREIRA, 2013), 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 Figure 8a).

In general, a geometric map is the Cartesian representation of the polar laser data measurements, taking into account the robot pose. Figure 8b illustrates the robot, the sensor and four box-shaped objects in V-REP simulator environment, while Figure 8c 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 Figure 8c, 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, other points are associated with the walls that limit the workspace, and the goal.

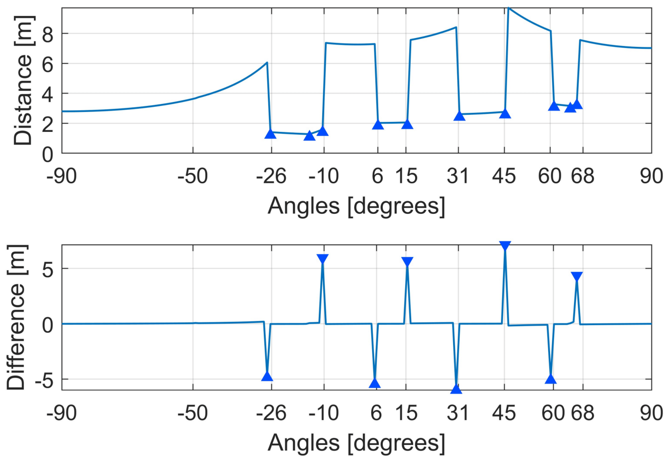


Figure 9 – Searching for objects through the laser measurement data.

Once having the geometric map, a procedure searches patterns to identify the objects. First, we get the sensor data in polar coordinates (Figure 9 top), and then calculate the difference between two consecutive measurements (Figure 9 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 Figure 9, the angular intervals $O_1 \in [-26°, -10°]$, $O_2 \in [6°, 15°]$, $O_3 \in [31°, 45°]$ and $O_4 \in [60°, 68°]$ 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 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.

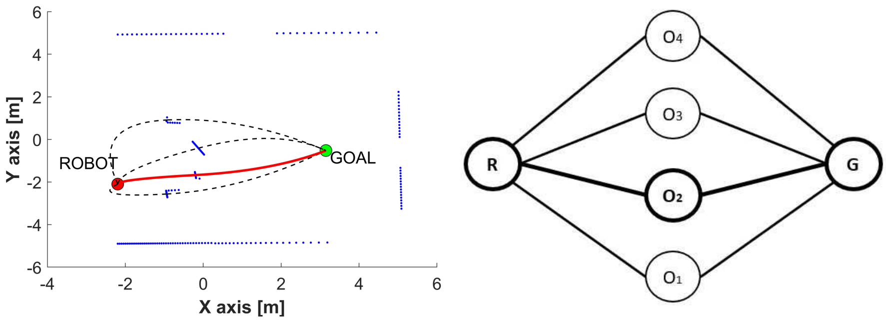
As a result, O_1 presents the faces $F_{1a} \in [-26°, -13°]$ and $F_{1b} \in [-13°, -10°]$, while O_4 presents the faces $F_{4a} \in [60°, 66°]$ and $F_{4b} \in [66°, 68°]$. Thereby, in case of two faces, it is necessary to adopt that one closest to orthogonality in relation to the straight segment 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.

3.3 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. Figure 10a 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 (M. GUIVANT J., 2019).



a Bézier curves connecting the robot, objects, and goal.

b Graph representation for four objects.

Figure 10 – Geometric map and its graph representation.

In this work we adopt a quadratic Bézier curves given by

$$B(t) = (1-t)^2\mathbf{p}_0 + 2t(1-t)\mathbf{p}_1 + t^2\mathbf{p}_2, \text{ with } 0 \leq t \leq 1, \tag{3.1}$$

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

$$D = \int_0^1 \sqrt{\left(\frac{d}{dx}B(t)\right)^2 + \left(\frac{d}{dy}B(t)\right)^2} \quad dt, \tag{3.2}$$

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.

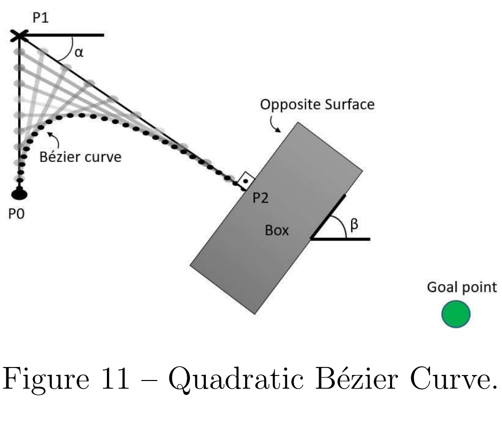
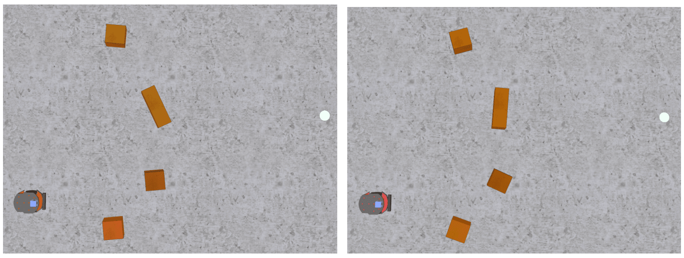
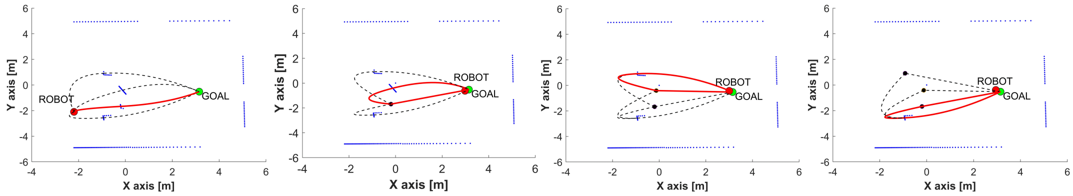


Figure 11 – Quadratic Bézier Curve.



a b

Figure 12 – Simulation scenarios.



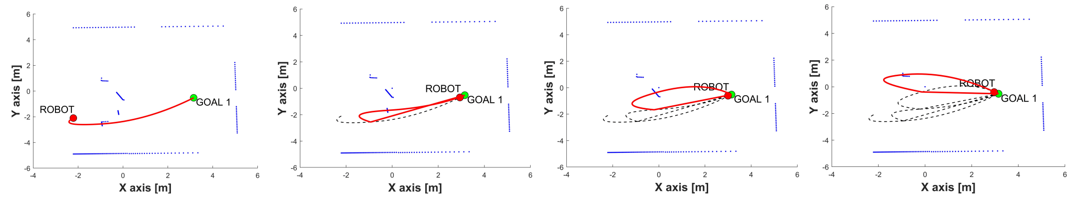
a b c d

Figure 13 – Dijkstra's algorithm working in the Scenario 1.

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 Figure 11), 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.

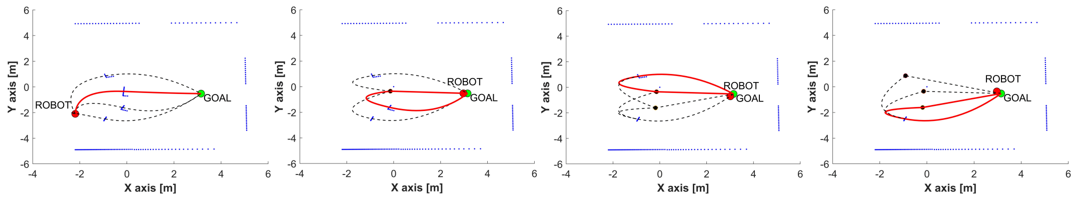
Once completing the graph representation, we should find the shortest path to handle and push each object to its destination. Thus, Dijkstra's algorithm calculates the best route taking into account the robot's home position, the goal localization, and the position of all objects as they are "seen" by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

Figures 13 and 15 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.



a b c d

Figure 14 – Non-Optimized Strategy working in the Scenario 1.



a b c d

Figure 15 – Dijkstra's algorithm working in the Scenario 2.

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 Figure 13 or 15, 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 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 Figure 14.

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.

Algorithm 1 Material Detection and Transport

$P \leftarrow Pioneer3DX$ ▷ Pioneer Robot class constructor
Connect($Matlab, VRep$) ▷ Connect Matlab to V-Rep

$Map \leftarrow$ **P.GetLaserData** ▷ Get Laser Data to build the Map
$ObjectList \leftarrow$ **ObjectSearching**(Map) ▷ Find the objects vertices through the Map

while $ObjectList$ **do** ▷ While Object List is not empty
  $Paths \leftarrow$ **BézierPath**($Robot\ position, Object\ position, Goal\ position$)
  $OptimalPath \leftarrow$ **Dijkstra**($Paths$)
  $Queue \leftarrow$ **PriorityQueue.Insert**($OptimalPath$)
end while

while $Queue$ **do** ▷ While Queue is not empty
  $Path \leftarrow$ **PriorityQueue.Pop**($Queue$)
  Navigation($Path$) ▷ Navigation Task
end while

3.4 Results

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 (MARTINS; SARCINELLI-FILHO; CARELLI, 2017). Knowing the robot limitation, we adopt $0.4\ \mathrm{m\,s^{-1}}$ as a maximum suitable linear velocity to follow the Bézier curves.

Two scenarios shown in Figures 16 and 17 are discussed here. In both of

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

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. Figures 16 and 17 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 Figure 14. 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 Figures 13 and 15, 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

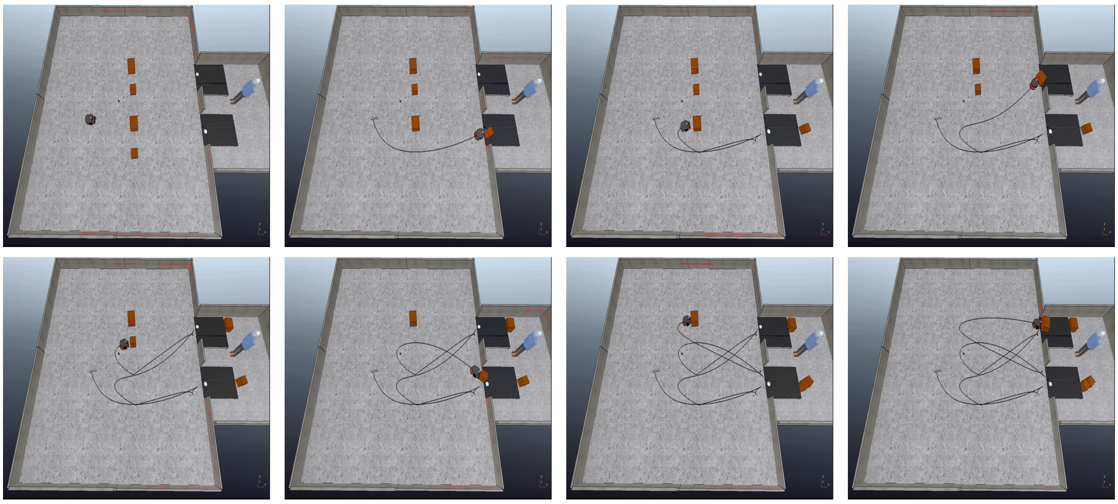


Figure 16 – Snapshots of the steps in Scenario 1.

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.

3.5 Conclusions

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 a 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

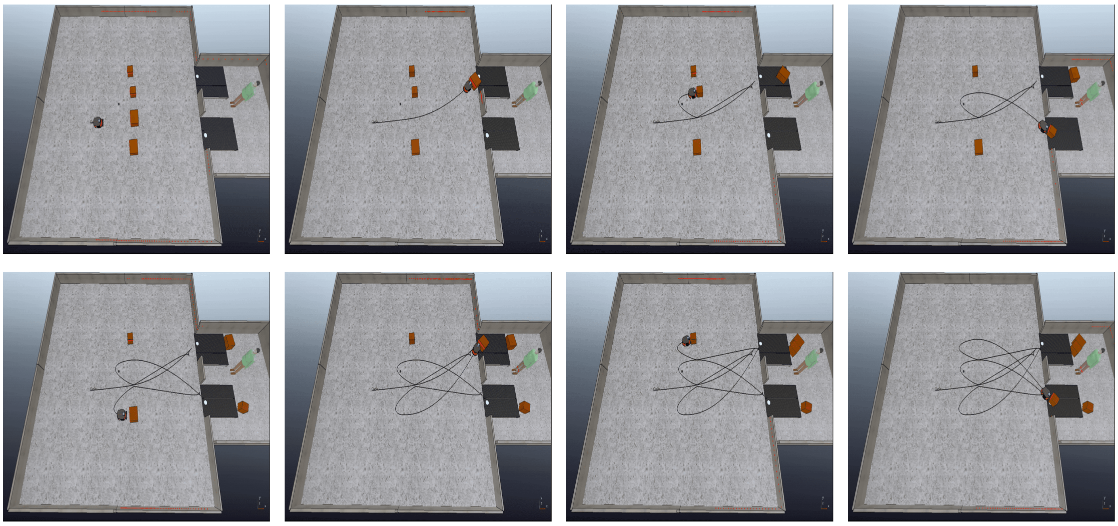


Figure 17 – Snapshots of the steps in Scenario 2.

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.

Finally, the future works 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.

4 Detecting and transporting objects by pushing-only approach

Abstract

This paper presents a robust algorithm for pushing objects, from random positions to a final destination. Our main contribution is the route recovery strategy, which reacts whenever the box transportation starts going out of the planned one. It provides robustness to objects rotation and slipping during their displacements, thus guarantees all of them are correctly delivered. To accomplish it, laser scanner data are used to build up a 2D map, also to identify if the objects are following the desired route. Besides, a topological map is created by Voronoi Graph in order to avoid collisions and Dijkstra's algorithm finds the optimal route. Then, Bézier curves provides suitable paths taking into account the position of the robot, objects and final destination. Finally, numerical simulations run in V-REP + Matlab, which demonstrates quite efficient for environments without occlusion of the objects to be transported.

4.1 Introduction

Autonomous transport systems provide transfer of objects, baggage, people, information or resources from point-to-point with minimal human intervention. It can include the full range of transport vehicles, comprising air and ground. They are most commonly deployed in structured industries zones but they are also expected to soon be deployed in public areas with varying degrees of autonomy.

Designing a fully autonomous transport robot system, whether it flies or not, is a complex problem. The robot has to be prepared to operate all alone in (un)structured environments, and it has to do so both reliably and efficiently. According to (TUCI; ALKILABI; AKANYETI, 2018), this field of research in automated industry can be classified at least in four main lines defined as pushing-only, grasping, carrying and caging strategies.

Pushing-only strategies are methods of transporting items by exerting pushing forces on them. These type of strategies are primarily employed by robots that cannot pull objects, since they have no means to grap them. Pushing-only strategies may appear to be relatively simple methods, however, it is on top of the challenges common to all transport task. Pushing-only strategies require a significant amount of coordination of actions to sustain the transport, such as the alignment of forces required to initiate the transportation and the guidance of the object without losing its contact.

Systems capable of object transport can be extremely effective in a variety of economic and social fields with great potential impact. The proposal here presented is motivated by the numerous applications where a cargo-pushing robot would be handy, such as.

- 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, after all it is possible to apply object detection techniques to distinguish object sizes and shapes.

4.1.1 Contribution

This paper is an enhanced version of our previous work (QUEMELLI; BRANDÃO, 2020), whose contributions were a geometric strategy to determine the posture of objects in the environment and the application of Bezier Curves to create a smooth path from them to the robot. The proposed method was similar to the Sokoban Game, however, in our research the agent could freely move to all directions instead of only Manhattan displacement. Now, our contributions are twofold: (i) an path planning strategy based on Voronoi Graphs, Bézier Curves, and Dijkstra's algorithm, to minimize objects' collision. (ii) a Reactive Control by pushing-only approach to handle the objects and avoid the contact loss. The second is considered more important because a recovery route is generated whenever the object rotates or slips during its displacement, providing robustness and intelligent execution of the pushing-only task.

It is worthy 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 (MARTINS; SARCINELLI-FILHO; CARELLI, 2017). The robot does not know previously the objects position and neither the scenario map. Our strategy runs whenever there is no object occlusion, i.e., our constraint considers the robot should "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 in the floor are minimal in such way it does not trap the objects movement.

4.2 Related work

Visual recognition in multi-robot systems is afflicted with a peculiar problem that observations made from different viewpoints present different perspectives, (TOMITA; SEKIYAMA; FUKUDA, 2014) proposed the Hierarchical Invariants Perception Model (HIPM) in which multiple representations of the target are dynamically evaluated and selected by the robot. Enabling a robot to recognize and classify objects in an environment is the key for developing a good strategy of transporting, (SGORBISSA; VERDA, 2013) implemented an algorithm that allows a mobile robot to identify furniture-like objects composed of assembled parts using a Microsoft Kinect. In a similar way, (TUNGADI; KLEEMAN, 2011) uses an autonomous robot with laser rangefinders to explore and learn about an environment, detecting the changes if some object was moved from the previous position. The paper exploits Simultaneous Localisation and Mapping (SLAM), together with autonomous exploration techniques to achieve the task of placing the items in the same location as it was before.

When it comes to learning how to manipulate objects from experience with minimal prior knowledge, robots encounter significant challenges. (RIDGE; UDE, 2013) employed a self-supervised multi-view online learning algorithm to bootstrap both the discovery of affordance classes in the post-push view, as well as a discriminative model for predicting them in the pre-push view. Another challenge is to construct a robust skill for pushing highly-diverse objects. (KRIVIC; UGUR; PIATER, 2016) presents a strategy for pushing unknown objects that differ widely in their properties. Subsequently, (KRIVIC; PIATER, 2019) developed an obstacle avoidance to complement the transport task. From a different approach, (WEBER et al., 2015) presented an innovative strategy to solve the box pushing problem with a spherical robot.

Besides, predicting the motion of a pushed object is not trivial. In practice, the sensitivity of the task to small changes in contact geometry, along with the variability of friction, hinders accurate predictions. (YU et al., 2016; ZENG et al., 2018; BAUZA; RODRIGUEZ, 2017) show studies based in execution of controlled pushing experiments, and captured a large high-fidelity dataset of pushing interactions. Thereby, those researches present a more complete view of the complexity of developing transport systems to push objects.

Some works already make use of object movement prediction data to create learning transferable forward models for robotic push manipulation, as it presents (STÜBER; KOPICKI; ZITO, 2018). However, some negative points should be raised regarding the need for training with previously chosen objects. One network may well predict motion models for a particular object, and for others it may need adjustments and retraining. Alternatively, (KRIVIC; PIATER, 2018) proposed a probabilistic model, where the robot update and adapt the control using maximum a posterior (MAP) estimation.

There are also object manipulation methodologies for robot groups, since not all tasks can be done with a single agent, depending on conditions such as object size, weight, and shape. (LÓPEZ-NICOLÁS; ÖZGÜR; MEZOUAR, 2015) showed how to push an unknown object in the plane from an initial pose to a target pose with two cooperating mobile robots. (MOON; KWAK; KIM, 2012) designed a control scheme decentralized where each robot has identical control algorithms and it consists in a sequence of three behaviors, approach, align, and push motions. (MARINO; PIERRI, 2018) implemented a distributed algorithm for cooperatively manipulating an object rigidly grasped by a team of mobile manipulators. Besides, (BACA et al., 2015) developed a approach based on the combination of two communication types, the intent was the execution of cooperative tasks such as moving objects or manipulating objects with multiple modular robot configurations. Alternatively, (OHASHI et al., 2016) solved the problem of avoiding overturning of the object by the robots and sliding of the handcart while tilting the object, an outrigger device is used to prevent the first problem of tilting, and a handcart locking device is used to prevent the second problem of sliding.

Planning navigation in environments can be a problem depending on the overall purpose of the task. In cases like transport, it is important to know how to avoid obstacles, especially when performing the pushing movements because it is desired not to crash the objects. One technique that show certain advantages is the searching of free spots from geometric maps by means of the Voronoi diagram. Moreover, in order to smoothly execute the trajectory, it is practical to apply parameterization in polynomial curves such as (KAVURAN, 2017). Likewise our work, (YANG et al., 2015b) proposes a method combining Dijkstra shortest path algorithm, Bézier curve and Voronoi diagram to create geometric paths. Similarly, (XIONG et al., 2019) designed a hybrid Voronoi-based ant colony optimization (V-ACO) technique for multiple autonomous marine vehicles (AMVs) to solve adaptive ocean sampling problem.

4.3 Mapping and Detecting Objects

There are several ways to get information about the environment where the robot is placed. Among the best known techniques, optical sensors can be mentioned such as cameras, lasers and sonars. The mapping approach applied in this work considered the employment of a laser scanner with a reading range of -90 to 90 degrees, with 1 sample per degree.

The goal in this stage is to map the environment and identify box-shaped objects for later transport. After distinguishing each item, a topological map is

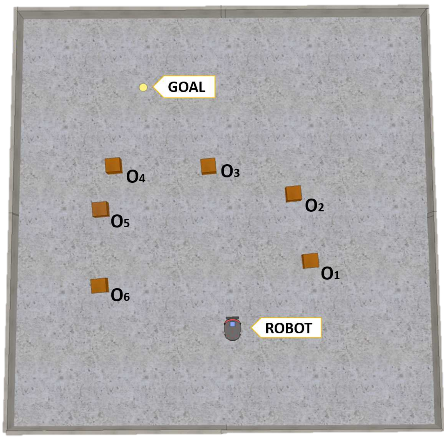
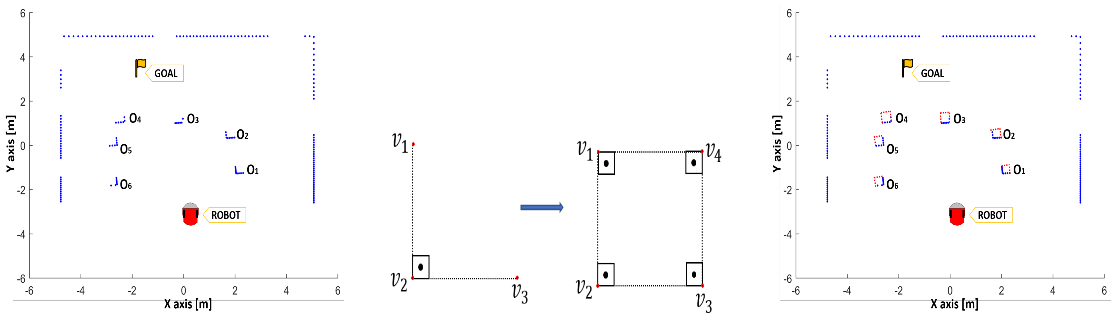


Figure 18 – Simulation environment, V-REP.

built and it is saved the crucial information for the robot navigation. Crucial points can be defined as free navigation spots for the robot. All the strategy applied for mapping and detecting objects are detailed in (QUEMELLI; BRANDÃO, 2020).

Figure 18 presents a simulated scenario where most of the preliminary tasks were developed. The purpose at this moment is to make the robot identifies all 6 boxes randomly arranged in the scenario, distinguishing the vertices of each object. Note that the number of objects may vary as long as all objects are in the robot's field of view.

After the mapping process, we can collect positioning information for each



a Map built by the laser scanner.

b Projection of hidden faces.

c Map with the projection of hidden faces.

Figure 19 – Geometric map in Cartesian coordinates created in Matlab platform.

vertex and size of each object. Since we know that each object has a cubic shape, geometrically we can project other hidden faces. The projection of hidden faces is deeply important for the development of the transport strategy. By knowing the exact positioning of all vertices and edges, it is possible to calculate the shortest path for intelligent maneuvering. Figure 19a illustrates the map built by the robot positioned lower and center. At the top and slightly left it can be found a yellow flag, defined as the final destination for the boxes.

Once the objects' vertices are located, we compute the projection of the hidden faces, those that are not seen by the robot. As can be seen from Figure 19b, v_1, v_2 and v_3 represent the vertices that delimit the view of an object on the map. To define v_4, we must first calculate a vector that links one vertex to another, for instance the vector $\vec{e}_{21}$ joints the vertices v_1 and v_2 is

$$\vec{e}_{21} = v_2 - v_1, \tag{4.1}$$

where the size of $\vec{e}_{21}$ determines the (x, y) coordinate projection of the v_4. Thus, Equation (4.2) calculates the rotation of the vector $\vec{e}_{21}$ by 90 degrees, and summing the position of the vertex v_1 it is possible to find the vertex v_4.

$$v_4 = \begin{bmatrix} cos(\pi/2) & sin(\pi/2) \\ -sin(\pi/2) & cos(\pi/2) \end{bmatrix} \vec{e}_{21} + v_1 \tag{4.2}$$

Afterwards the prediction of all vertex projections of each box, we generated an array with the coordinates (x, y) of each objects' vertex in the scenario. Figure 19c illustrates the result of the objects with its projections of hidden vertices respectively.

Forthwith, it is necessary to calculate the most appropriate face to perform the object maneuver approach. In this stage, two vectors are defined to compute the selection of the appropriate face. As can be seen in Figure 20, $\vec{u}$ is called the unit vector perpendicular to the box's face, $\vec{v}$ represents the vector connecting from the midpoint of the box's face pointing to the final destination place (yellow flag), thus

$$\vec{w} = \vec{u} + \vec{v} \tag{4.3}$$

is the resulting vector.

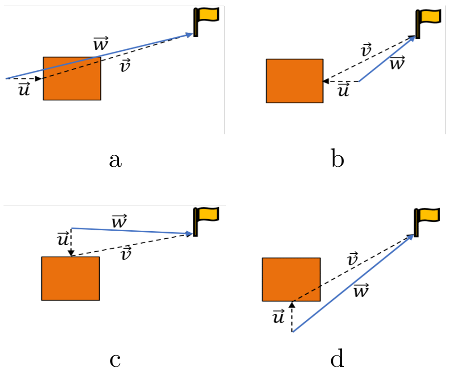


Figure 20 – Calculating the box faces which give the shortest maneuver approach.

The smoothest and most suitable maneuver is the one which the robot needs to perform as few turns as possible. Note that the most appropriate face to perform a transport maneuver is also the one that generates the largest resulting vector $\vec{w}$, see in Figure 20a. This strategy can be applied to any type of convex polygon object.

Finally, when calculating the resulting vector for the four faces of an object, the face that provides the best approximation maneuver is defined by the largest vector $\vec{w}$. Then this object face coordinates are stored.

4.4 Path Planning

For the development of the navigation strategy, three important aspects were considered. First, collision-free path planning is required between the robot position and the desired objects' faces location. Second, the robot must perform a smooth and perpendicular approach maneuver to the desired face in order to have the largest possible contact area, resulting in greater control when handling the object and reducing the chances of loss of control. And third, a collision-free path from the robot-box stage to the final destination place.

Taking into account the items presented above, we designed a hybrid approach combining Voronoi graphs, Bezier curves and Dijkstra algorithm.

To obtain all the free points through which the robot can move, we imple-

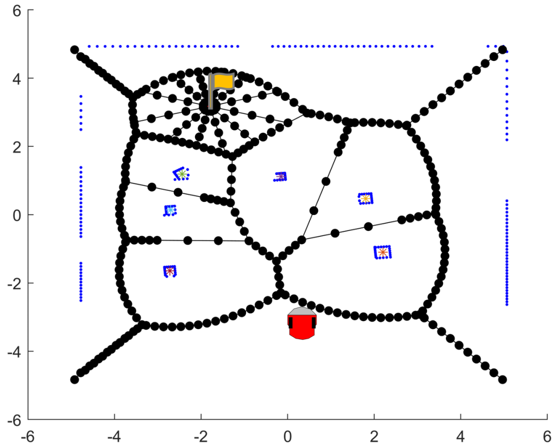


Figure 21 – Voronoi Graph.

mented the Voronoi graph to the scenario map, Figure 21 illustrates the vertices and edges produced. The black dots indicate the vertices through which the robot can navigate without collision.

Besides, each maneuver requires a handling and a pushing stage. 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 $\mathbf{P}_1$ is the robot position, the auxiliary point $\mathbf{P_2}$ is in some place of an orthogonal line that intersects the box surface (see Figure 22), and finally, $\mathbf{P_3}$ is the midpoint at the box surface. Further, the second curve starts at the last point of the first curve ($\mathbf{P_3}$), the auxiliary point is $\mathbf{P_4}$, and it ends at the goal point $\mathbf{P_5}$. The Equation 4.4 describes the simple path with the highlighted points from Figure 22, where there is no need to avoid other objects along the way, however this approach only works when there are few items to be transported. Thus, for a more complete coverage, it is necessary to insert the auxiliary points of the Bezier curve to the Voronoi graph.

$$B(t) = (1-t)^2\mathbf{P}_1 + 2t(1-t)\mathbf{P}_2 + t^2\mathbf{P}_3, \quad 0 \le t \le 1, \tag{4.4}$$

All auxiliary points are inserted into the Voronoi graph, then it enables the

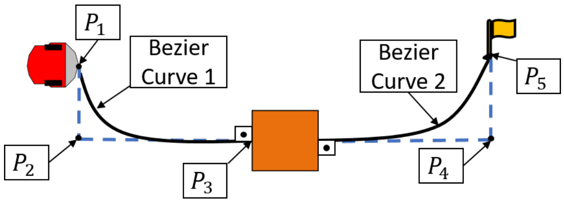


Figure 22 – Bezier Curves.

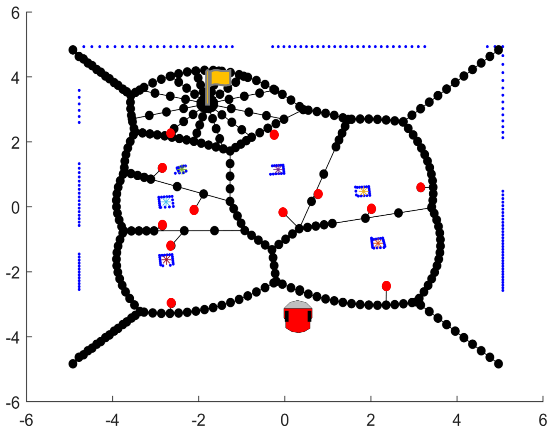


Figure 23 – Insertion of auxiliary points for transport maneuver.

robot to navigate the free paths and draw near to the boxes when needed. Figure 23 illustrates the insertion of control points highlighted in red in the vicinity of each object.

Consequently, to reach any box within Voronoi's graph, we must calculate the shortest path from the robot to the objects. In this regard, we apply the Dijkstra search algorithm to achieve the path through the graph vertices. Thereby, the set of points is saved in an array of vertices.

At this stage, all the strategic points are available and it is feasible to create a smooth path with the Bezier curve expressed in binomial polynomial. The equation 4.5 represents the Bezier function for n auxiliary points, where $\mathbf{P_i}$ represents the set of points found by Dijkstra's Algorithm.

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \mathbf{P_i}, \quad 0 \leq t \leq 1, \tag{4.5}$$

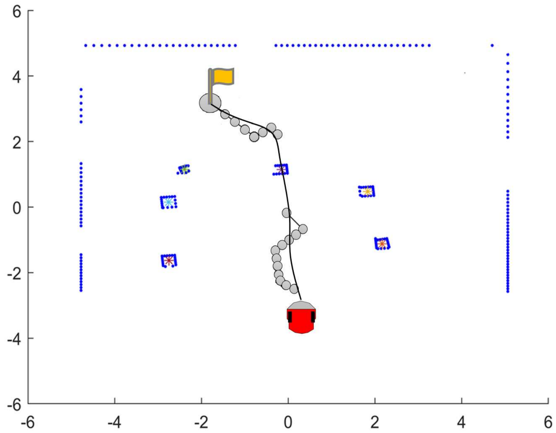


Figure 24 – Bezier curves applied to the vertices defined by Dijkstra's algorithm.

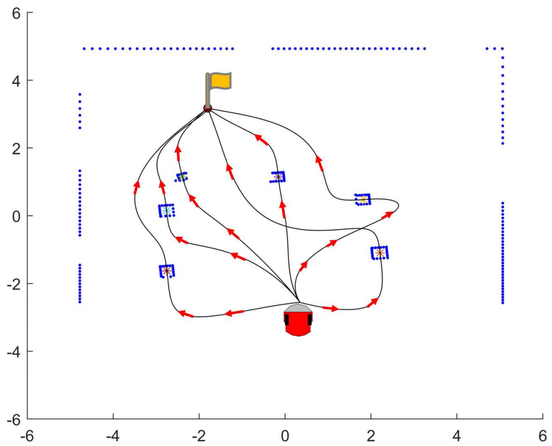


Figure 25 – Bezier curves applied to the Voronoi graph.

To conclude the path planning process, one of the requirements for navigation and transportation is the smoothness of the route. Once we add all points found in $\mathbf{P_i}$, the curve formed is too abrupt and full of corners. For this reason, a points removal treatment is performed in order to decrease the excess of vertices. Therefore, all vertices that are within a distance of less than 80cm are excluded from the set. Figure 24 represents the Bezier curve highlight in black, and on the background the vertices found by Dijkstra's algorithm to move one box.

Figure 25 exemplifies the entire path planning by generating trajectories with the Equation 4.5. Red arrows were added to indicate the direction of the paths.

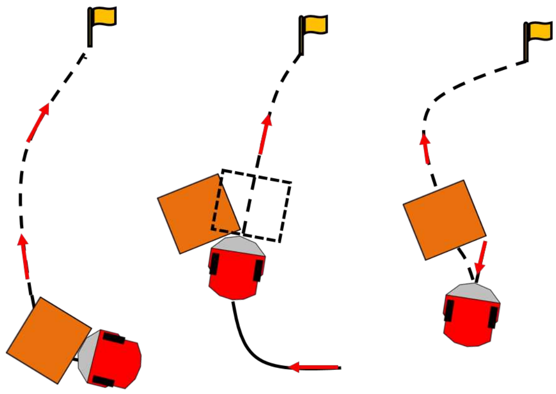


Figure 26 – Loss of object control and path recalculation.

Note that some routes are quite curved, but the strategy prioritizes to take the shortest route hence the scenery becomes less complicated, and the transportation of the following objects becomes easier.

Summarizing the entire strategy developed, Algorithm 2 presents a pseudo-code with all the steps detailed in this section.

Algorithm 2 Object Detection and Path Planning

$H \leftarrow$ **HandlePushing** ▷ Handling and Pushing class constructor
$Map \leftarrow$ **H.GetLaserData** ▷ Get Laser Data to build the Map
$ObjectList \leftarrow$ **H.ObjectSearching**(Map) ▷ Find the objects vertices through the Map
$BoxFaces \leftarrow$ **H.Projection**($ObjectList$) ▷ Projection of hidden faces and identifying the best surface

$Graph \leftarrow$ **H.Voronoi**(*Robot position,BoxFaces,Map,Goal position*)

while $ObjectList(i)$ **do** ▷ While Object List is not empty
    $Vertices \leftarrow$ **H.Dijkstra**(*Graph,Robot position,ObjectList(i), Goal position*)
    $Paths \leftarrow$ **H.BezierPath**(*Vertices*)
    $Queue \leftarrow$ **PriorityQueue.Insert**(*Paths*)
end while

4.5 Navigation and Reactive Transport

The path planning presented in the previous section generates a priority queue of paths to be performed by the robot in the navigation strategy. Some observations must be taken to the success of this task. It is important to say that sometimes the robot does not process the correct size of all objects, requiring a map update as the robot approaches the box, by this way the handling points can be redefined correctly. Another issue worth mentioning, depending on the curvature of the path, the robot may lose control of the object, since it does not have a claw to hold the boxes. Therefore, we developed a reactive control strategy, where the robot constantly updates the information obtained through the laser, and then recalculates the routes to adjust the control of objects when necessary.

Initially, the robot collects from the priority queue the path it must follow, and then, as it approaches in a short distance from the box, a scanning procedure is performed to check dimensions with better accuracy. If the found values match the previously saved, the handling task proceeds naturally. Otherwise, the robot takes the new values into account and update the path handling maneuver.

Following, from the robot-box stage to the final destination, the pushing control is performed. The strategy is based on collecting real-time laser scanner data and comparing whether the center of the box is following the desired trajectory. If at any time the difference between the object's center position and the desired path is greater than 15 cm, instantly the robot stops, and performs a reverse maneuver. At this moment, it recalculates the path to take back the object. Figure 26 shows in detail the step-by-step approach.

Algorithm 3 describes and solves the proposed handling and a pushing tasks. 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. Note that there are two loops, the outermost one executes while there are paths to be travelled. And the innermost loop is executed while a transport sub-task is performed. During navigation, the robot constantly checks the map data. Then there are two possible situations, the first is when the robot is still approaching the box, denoted by $BezierCurve1$; and the second is when the robot is already under object control,

Algorithm 3 Navigation and Reactive Control

```
while Queue do                                  ▷ While Queue is not empty
    Paths ← PriorityQueue.Pop( )
    while t < tmax do                           ▷ While time is not over
        Navigation(Paths)
        if BezierCurve1 then
            Flag ← CheckBoxSize(Map)
            if Flag is 1 then
                Paths ← RemakePath(Map)
            end if
        else if BezierCurve2 then
            Flag ← CheckBoxPath(Map)
            if Flag is 1 then
                Paths ← RemakePath(Map)
            end if
        end if
    end while
end while
```

named as $BezierCurve2$.

4.6 Results and Discussion

This section presents the numerical experiments performed with the V-Rep simulator and Matlab software. The mobile robot used in the research simulations is the Pioneer 3DX, a differential traction robot.

Firstly, all the results discussed here are available on the NERO UFV Channel on YouTube. For comparison, we present the results obtained by the previous article, where it is possible to verify that there is no Recovery strategy. And yet, the robot only moved the objects to some final destination ahead, as well as there is no obstacle avoidance. See the video at the link: <https://youtu.be/frLkUTW0oJk>.

Now, considering the results obtained in the current paper, the following link illustrates how the algorithm works for a single box: <https://youtu.be/hFZ9Qmjdbvg>. The intention in this simulation is to demonstrate the mechanism

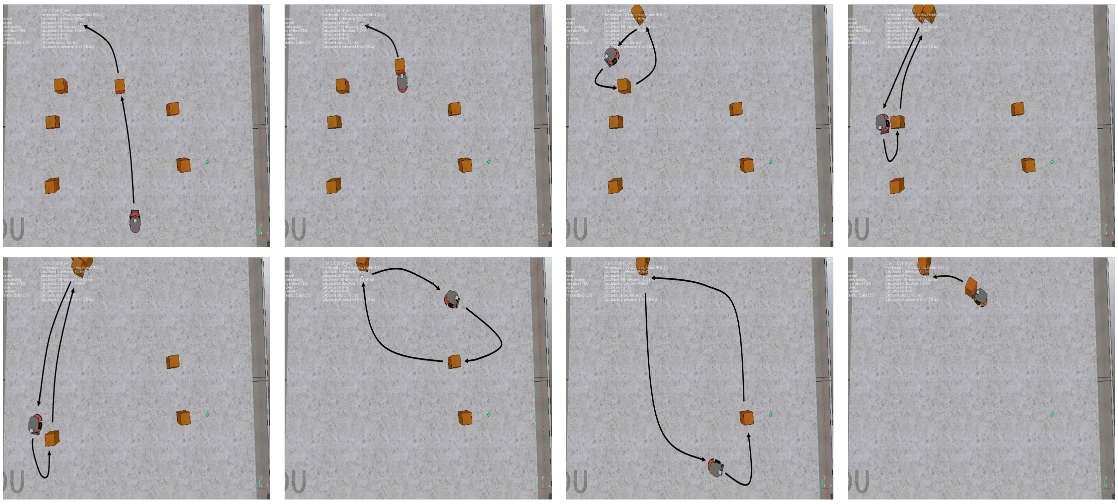


Figure 27 – Snapshots of the steps in Scenario.

of recognition and picking the most appropriate side of the object for transport. Moreover, the Reactive Control acts recovering the box in the minute 1:08.

The main results achieved can be seen in the video available in the following link: <https://youtu.be/ZbmeRQBQjO8>. Also, the snapshots of the steps from the numerical experiment are available on Figure 27 and it will be used to discuss the results. The scenario discussed here refers to Figure 18. The task aims to identify all objects in the environment, calculate the best approximation face and thus guide the boxes to the specified destination, avoiding the collision with others objects in the scene. At the moment the simulation starts, the robot maps and already calculates the approach route to the objects, as well as the route from the robot-box to the destination (yellow circle on the top). Besides, the routine is repeated until all objects are transported. It is possible to notice that the map on the upper left side in the video is always updating the new path planning.

Note that the routes calculated on the map always avoid the collision of objects, which means, the transport path of each box does not pass through the spot of others. This result is due to the use of the Voronoi graph as a way of obtaining empty spaces, and consequently, free-collision routes in the environment.

The Reactive Control algorithm can be seen in the video at the instants

0:29, 1:24 to 1:32, and 1:56. When the robot starts to move the object, the position information of the box is updated with the laser sensor, thus it is possible to check if the route is being carried out correctly. In case the box leaves the planned path, the robot performs a return maneuver and recalculates all the transport.

It is important to say that the moment the robot pushes the penultimate object, apparently it is avoiding an object that is no longer there. However, the fact is that the Voronoi graph is always updating at the end of each box delivery, thus, the less obstacles in the scenario, the less free vertices the algorithm generates. For this reason, the vertices end up being very distant from each other, resulting in a more rounded route.

4.7 Conclusions

This manuscript presents graph, computational geometry, and recovering control techniques for mapping, searching and path planning, which enabled the mobile robot to accomplish a box-shape objects transportation by pushing-only approach. First, an UGV explores the environment where it is and builds a geometric 2D map, then a topological map is created using Voronoi diagram, consecutively, auxiliary points are inserted to facilitate the box control, and finally Dijkstra's algorithm searches for the best robot-object-destination path. Mostly important, our strategy provides a reactive control algorithm, where the boxes' route are constantly checked in case of deviation from the desired path, thus the robot stops and recalculate the route.

Ultimately, despite considering only one robot to perform the task of pushing, as next steps, we intend to add more agents in such 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.

5 Concluding Remarks and Future Works

The work presented in this dissertation, discussed through three articles, contributes significantly to the area of autonomous systems for transport of objects. The work presents itself as an initiative to develop studies on the use of sensors, mapping, path planning, and reactive control for mobile robotics. The general objective of the proposal was reached as well as those specific that subsidized the phases for the conclusion of all work. The results obtained were satisfactory and validated the hypothesis raised at the beginning of the project. Finally, the execution of the work resulted in some contributions that will be discussed in this session and opens the way for new research opportunities in the area.

Initially the main purpose of the research was to promote and contribute to the transport of objects by pushing, using laser sensor to identify the items and path planning to guide the robot in the process. To perform autonomously the task, the work also proposed the use of reactive control method. The lack of examples to follow on the literature resulted in an algorithm that unites techniques of mapping, graphs, geometric computational in such way that no work has done before, and proved to be efficient.

The first contribution is related to the classification tool, formalizing distance measurement and object detection with laser sweep sensors, defining strictly an object and some of its properties. Thereafter, Voronoi graph, Bezier Curve and Dijkstra's algorithm all together are the second concrete contributions of this work for path planning strategy.

The main contribution of this research was the implementation of reactive control. Although it is simple approach, its application can contribute substantially to the task success. The use of the laser sensor in order to check if the boxes are following the path planned is essential for the operation of the algorithm.

Finally, future works to be explored are listed below:

1. Transport a larger load with more than one robot. Due to the complexity of the objects size, sometimes it is important to group two or more robots to complete the task.

2. Transport of a heavier load, analysis of transport feasibility. In cases of difficulties due to the load weight, it is interesting to check if the robot can run alone or needs help.

3. Competitive and collaborative strategies for transporting objects with multiple agents.

Bibliography

AKIN, H. L. et al. Robocup rescue robot and simulation leagues. *AI magazine*, v. 34, n. 1, p. 78–78, 2013. Citado na página 22.

ANDÚJAR, D. et al. A lidar-based system to assess poplar biomass. *Gesunde Pflanzen*, Springer, v. 68, n. 3, p. 155–162, 2016. Citado na página 22.

ANDúJAR, D. et al. Discriminating crop, weeds and soil surface with a terrestrial lidar sensor. *Sensors (Basel, Switzerland)*, v. 13, p. 14662–75, 11 2013. Citado na página 22.

AZIM, A.; AYCARD, O. Detection, classification and tracking of moving objects in a 3d environment. In: IEEE. *2012 IEEE Intelligent Vehicles Symposium*. [S.l.], 2012. p. 802–807. Citado na página 21.

BACA, J. et al. Modular robot systems towards the execution of cooperative tasks in large facilities. *Robotics and Autonomous Systems*, Elsevier, v. 66, p. 159–174, 2015. Citado na página 50.

BARGOTI, S. et al. A pipeline for trunk detection in trellis structured apple orchards. *Journal of field robotics*, Wiley Online Library, v. 32, n. 8, p. 1075–1094, 2015. Citado na página 22.

BAUZA, M.; RODRIGUEZ, A. A probabilistic data-driven model for planar pushing. In: IEEE. *2017 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2017. p. 3008–3015. Citado na página 49.

BISWAS, J.; VELOSO, M. Depth camera based indoor mobile robot localization and navigation. In: IEEE. *2012 IEEE International Conference on Robotics and Automation*. [S.l.], 2012. p. 1697–1702. Citado na página 21.

BLOCHLIGER, F. et al. Topomap: Topological mapping and navigation based on visual slam maps. In: IEEE. *2018 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2018. p. 1–9. Citado na página 36.

BRESSON, G. et al. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, IEEE, v. 2, n. 3, p. 194–220, 2017. Citado na página 21.

CAILHOL, S. et al. Multi-layer path planning control for the simulation of manipulation tasks: Involving semantics and topology. *Robotics and*

Computer-Integrated Manufacturing, Elsevier, v. 57, p. 17–28, 2019. Citado na página 36.

COLAçO, A. F. et al. Orange tree canopy volume estimation by manual and lidar-based methods. *Advances in Animal Biosciences*, Cambridge University Press, v. 8, n. 2, p. 477–480, 2017. Citado na página 22.

COSTA, P. J. et al. Localization and navigation of an omnidirectional mobile robot: the robot@ factory case study. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, IEEE, v. 11, n. 1, p. 1–9, 2016. Citado na página 20.

Gargoum, S.; El-Basyouny, K. Automated extraction of road features using lidar data: A review of lidar applications in transportation. In: *2017 4th International Conference on Transportation Information and Safety (ICTIS)*. [S.l.: s.n.], 2017. p. 563–574. Citado na página 21.

Giri, P.; Kharkovsky, S. Detection of surface crack in concrete using measurement technique with laser displacement sensor. *IEEE Transactions on Instrumentation and Measurement*, v. 65, n. 8, p. 1951–1953, Aug 2016. Citado 2 vezes nas páginas 21 and 27.

HERRERO-PEREZ, D.; MARTINEZ-BARBERA, H. Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems. *IEEE Transactions on Industrial Informatics*, IEEE, v. 6, n. 2, p. 166–180, 2010. Citado na página 33.

HICHRI, B. et al. Design of cooperative mobile robots for co-manipulation and transportation tasks. *Robotics and Computer-Integrated Manufacturing*, Elsevier, v. 57, p. 412–421, 2019. Citado na página 33.

Huang, Z. et al. Accurate 3-d position and orientation method for indoor mobile robot navigation based on photoelectric scanning. *IEEE Transactions on Instrumentation and Measurement*, v. 64, n. 9, p. 2518–2529, Sep. 2015. Citado na página 20.

JO, H. et al. Efficient grid-based rao–blackwellized particle filter slam with interparticle map sharing. *IEEE/ASME Transactions on Mechatronics*, IEEE, v. 23, n. 2, p. 714–724, 2018. Citado na página 35.

KAVURAN, G. Path planning for nonholonomic mobile robot based on bézier curve. In: IEEE. *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. [S.l.], 2017. p. 1–5. Citado na página 50.

KHOZAEE, A.; AMINAIEE, A. H.; GHAFFARI, A. A swarm robotic approach to distributed object pushing using fuzzy controllers. In: IEEE. *2008 IEEE International Conference on Robotics and Biomimetics*. [S.l.], 2009. p. 1117–1122. Citado na página 33.

KONOLIGE, K. et al. A low-cost laser distance sensor. In: IEEE. *2008 IEEE International Conference on Robotics and Automation*. [S.l.], 2008. p. 3002–3008. Citado na página 20.

KRINKIN, K. et al. Evaluation of modern laser based indoor slam algorithms. In: IEEE. *2018 22nd Conference of Open Innovations Association (FRUCT)*. [S.l.], 2018. p. 101–106. Citado na página 21.

KRIVIC, S.; PIATER, J. Online adaptation of robot pushing control to object properties. In: IEEE. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2018. p. 4614–4621. Citado na página 49.

KRIVIC, S.; PIATER, J. Pushing corridors for delivering unknown objects with a mobile robot. *Autonomous Robots*, Springer, v. 43, n. 6, p. 1435–1452, 2019. Citado na página 49.

KRIVIC, S.; UGUR, E.; PIATER, J. A robust pushing skill for object delivery between obstacles. In: IEEE. *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. [S.l.], 2016. p. 1184–1189. Citado na página 49.

KWON, Y. et al. Super rays and culling region for real-time updates on grid-based occupancy maps. *IEEE Transactions on Robotics*, IEEE, 2019. Citado na página 35.

LAU, B.; SPRUNK, C.; BURGARD, W. Efficient grid-based spatial representations for robot navigation in dynamic environments. *Robotics and Autonomous Systems*, Elsevier, v. 61, n. 10, p. 1116–1130, 2013. Citado na página 35.

LEHTOMÄKI, M. et al. Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. *Remote Sensing*, Molecular Diversity Preservation International, v. 2, n. 3, p. 641–664, 2010. Citado na página 21.

LI, X.; ZELL, A. Path following control for a mobile robot pushing a ball. *IFAC Proceedings Volumes*, Elsevier, v. 39, n. 15, p. 49–54, 2006. Citado na página 33.

Li, Y.; Ruichek, Y.; Cappelle, C. Optimal extrinsic calibration between a stereoscopic system and a lidar. *IEEE Transactions on Instrumentation and Measurement*, v. 62, n. 8, p. 2258–2269, Aug 2013. Citado 2 vezes nas páginas 20 and 27.

LINDSTROM, M.; EKLUNDH, J.-O. Detecting and tracking moving objects from a mobile platform using a laser range scanner. In: IEEE. *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. [S.l.], 2001. v. 3, p. 1364–1369. Citado 2 vezes nas páginas 21 and 22.

LÓPEZ-NICOLÁS, G.; ÖZGÜR, E.; MEZOUAR, Y. Image-based control of two mobile robots for object pushing. In: IEEE. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2015. p. 5472–5478. Citado na página 49.

LU, S.; ZHANG, Y.; SU, J. Mobile robot for power substation inspection: a survey. *IEEE/CAA Journal of Automatica Sinica*, IEEE, 2017. Citado na página 21.

M. GUIVANT J., K. J. W. M. H. Path planning for autonomous bulldozers. *Mechatronics*, Elsevier, v. 58, n. 4, p. 20–38, 2019. Citado na página 38.

MARINHO, L. B. et al. A novel mobile robot localization approach based on classification with rejection option using computer vision. *Computers & Electrical Engineering*, Elsevier, v. 68, p. 26–43, 2018. Citado na página 36.

MARINO, A.; PIERRI, F. A two stage approach for distributed cooperative manipulation of an unknown object without explicit communication and unknown number of robots. *Robotics and Autonomous Systems*, Elsevier, v. 103, p. 122–133, 2018. Citado na página 50.

MARTÍNEZ-BARBERÁ, H.; HERRERO-PÉREZ, D. Autonomous navigation of an automated guided vehicle in industrial environments. *Robotics and Computer-Integrated Manufacturing*, Elsevier, v. 26, n. 4, p. 296–311, 2010. Citado na página 33.

MARTINS, F. N.; SARCINELLI-FILHO, M.; CARELLI, R. A velocity-based dynamic model and its properties for differential drive mobile robots. *Journal of Intelligent & Robotic Systems*, Springer, v. 85, n. 2, p. 277–292, 2017. Citado 3 vezes nas páginas 34, 42, and 48.

MERIÇLI, T.; VELOSO, M.; AKIN, H. L. Push-manipulation of complex passive mobile objects using experimentally acquired motion models. *Autonomous Robots*, Springer, v. 38, n. 3, p. 317–329, 2015. Citado na página 32.

MERTZ, C. et al. Moving object detection with laser scanners. *Journal of Field Robotics*, Wiley Online Library, v. 30, n. 1, p. 17–43, 2013. Citado na página 21.

MEYER, K. de; LEMON, O.; NEHMZOW, U. Multiple resolution mapping for e cient mobile robot navigation. 1997. Citado na página 36.

MOON, S.; KWAK, D.; KIM, H. J. Cooperative control of differential wheeled mobile robots for box pushing problem. In: IEEE. *2012 12th International Conference on Control, Automation and Systems*. [S.l.], 2012. p. 140–144. Citado na página 49.

NUNEZ, P. et al. Feature extraction from laser scan data based on curvature estimation for mobile robotics. In: IEEE. *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. [S.l.], 2006. p. 1167–1172. Citado na página 21.

OHASHI, F. et al. Realization of heavy object transportation by mobile robots using handcarts and outrigger. *Robomech journal*, Springer, v. 3, n. 1, p. 27, 2016. Citado na página 50.

ORIA-AGUILERA, H.; ALVAREZ-PEREZ, H.; GARCIA-GARCIA, D. Mobile lidar scanner for the generation of 3d georeferenced point clouds. In: IEEE. *2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*. [S.l.], 2018. p. 1–6. Citado na página 21.

PEREIRA, G. A. et al. Coordination of multiple mobile robots in an object carrying task using implicit communication. In: IEEE. *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*. [S.l.], 2002. v. 1, p. 281–286. Citado na página 33.

PINTO, A. M.; ROCHA, L. F.; MOREIRA, A. P. Object recognition using laser range finder and machine learning techniques. *Robotics and Computer-Integrated Manufacturing*, Elsevier, v. 29, n. 1, p. 12–22, 2013. Citado na página 36.

QUEMELLI, M.; BRANDÃO, A. Handling and pushing objects using unmanned guided vehicles. *Robotics and Computer-Integrated Manufacturing*, Elsevier, v. 63, p. 101913, 2020. Citado 2 vezes nas páginas 47 and 51.

RIDGE, B.; UDE, A. Action-grounded push affordance bootstrapping of unknown objects. In: IEEE. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2013. p. 2791–2798. Citado na página 49.

SAKUYAMA, T. et al. Object transportation by two mobile robots with hand carts. *International scholarly research notices*, Hindawi, v. 2014, 2014. Citado na página 33.

Schlarp, J.; Csencsics, E.; Schitter, G. Optical scanning of a laser triangulation sensor for 3d imaging. *IEEE Transactions on Instrumentation and Measurement*, p. 1–1, 2019. Citado 2 vezes nas páginas 20 and 21.

SGORBISSA, A.; VERDA, D. Structure-based object representation and classification in mobile robotics through a microsoft kinect. *Robotics and Autonomous Systems*, Elsevier, v. 61, n. 12, p. 1665–1679, 2013. Citado na página 48.

SHEN, S.; MICHAEL, N.; KUMAR, V. Autonomous multi-floor indoor navigation with a computationally constrained mav. In: IEEE. *2011 IEEE International Conference on Robotics and Automation*. [S.l.], 2011. p. 20–25. Citado na página 21.

SIMBA, K. R.; UCHIYAMA, N.; SANO, S. Real-time smooth trajectory generation for nonholonomic mobile robots using bézier curves. *Robotics and Computer-Integrated Manufacturing*, Elsevier, v. 41, p. 31–42, 2016. Citado na página 33.

SINGHA, A.; RAY, A. K.; SAMADDAR, A. B. Grid-based ugv navigation in a dynamic environment using neural network. In: IEEE. *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. [S.l.], 2018. p. 509–514. Citado na página 35.

STÜBER, J.; KOPICKI, M.; ZITO, C. Feature-based transfer learning for robotic push manipulation. In: IEEE. *2018 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2018. p. 1–5. Citado na página 49.

SUDSANG, A.; ROTHGANGER, F.; PONCE, J. Motion planning for disc-shaped robots pushing a polygonal object in the plane. *IEEE Transactions on Robotics and Automation*, IEEE, v. 18, n. 4, p. 550–562, 2002. Citado na página 33.

TAN, G.-z.; HE, H.; AARON, S. Global optimal path planning for mobile robot based on improved dijkstra algorithm and ant system algorithm. *Journal of Central South University of Technology*, Springer, v. 13, n. 1, p. 80–86, 2006. Citado na página 33.

TANG, L. et al. Topological local-metric framework for mobile robots navigation: a long term perspective. *Autonomous Robots*, Springer, v. 43, n. 1, p. 197–211, 2019. Citado na página 36.

TOMITA, S.; SEKIYAMA, K.; FUKUDA, T. Consensus-making algorithms for cognitive sharing of object in multi-robot systems. *ROBOMECH Journal*, SpringerOpen, v. 1, n. 1, p. 1–13, 2014. Citado na página 48.

TUCI, E.; ALKILABI, M. H.; AKANYETI, O. Cooperative object transport in multi-robot systems: A review of the state-of-the-art. *Frontiers in Robotics and AI*, Frontiers Media, v. 5, p. 59, 2018. Citado 3 vezes nas páginas 14, 33, and 46.

TUNGADI, F.; KLEEMAN, L. Discovering and restoring changes in object positions using an autonomous robot with laser rangefinders. *Robotics and Autonomous Systems*, Elsevier, v. 59, n. 6, p. 428–443, 2011. Citado na página 48.

WAKITA, S.; NAKAMURA, T.; HACHIYA, H. Laser variational autoencoder for map construction and self-localization. In: IEEE. *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.], 2018. p. 3993–3998. Citado na página 21.

Wang, J. et al. A proposal to compensate platform attitude deviation's impact on laser point cloud from airborne lidar. *IEEE Transactions on Instrumentation and Measurement*, v. 62, n. 9, p. 2549–2558, Sep. 2013. Citado 2 vezes nas páginas 20 and 27.

WEBER, J. et al. Solving the box-pushing problem using a spherical robot. In: IEEE. *2015 16th International Conference on Research and Education in Mechatronics (REM)*. [S.l.], 2015. p. 7–11. Citado na página 49.

XIONG, C. et al. Path planning of multiple autonomous marine vehicles for adaptive sampling using voronoi-based ant colony optimization. *Robotics and Autonomous Systems*, Elsevier, v. 115, p. 90–103, 2019. Citado na página 50.

XIONG, X. et al. Automatic creation of semantically rich 3d building models from laser scanner data. *Automation in construction*, Elsevier, v. 31, p. 325–337, 2013. Citado na página 21.

YANG, B. et al. Hierarchical extraction of urban objects from mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, Elsevier, v. 99, p. 45–57, 2015. Citado na página 21.

YANG, X. et al. Trajectory planning for robocup msl mobile robots based on bézier curve and voronoi diagram. In: IEEE. *2015 IEEE International Conference on Information and Automation*. [S.l.], 2015. p. 2552–2557. Citado na página 50.

YERSHOV, D. S.; LAVALLE, S. M. Simplicial dijkstra and a* algorithms for optimal feedback planning. In: IEEE. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2011. p. 3862–3867. Citado na página 33.

YU, K.-T. et al. More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing. In: IEEE. *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. [S.l.], 2016. p. 30–37. Citado na página 49.

ZENG, A. et al. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In: IEEE. *2018 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS). [S.l.], 2018. p. 4238–4245. Citado na página 49.