ICINCO 2023

20th International Conference on Informatics in Control, Automation and Robotics

PROCEEDINGS

Volume 1

13 - 15 November, 2023

EDITORS

Giuseppina Gini
Henk Nijmeijer
Dimitar Filev

https://icinco.scitevents.org

SPONSORED BY

INSTICC

PAPERS AVAILABLE AT

SCITEPRESS DIGITAL LIBRARY

ICINCO 2023

Proceedings of the
20th International Conference on
Informatics in Control, Automation and Robotics

Volume 1

Rome - Italy

November 13 - 15, 2023

Sponsored by
INSTICC - Institute for Systems and Technologies of Information, Control and Communication

Technically Co-sponsored by
IFAC - International Federation of Automatic Control

ACM In Cooperation
SIGAI - ACM Special Interest Group on Artificial Intelligence

In Cooperation with
AAAI - Association for the Advancement of Artificial Intelligence
INNS - International Neural Network Society

Edited by Giuseppina Gini, Henk Nijmeijer and Dimitar Filev

Printed in Portugal

ISSN: 2184-2809

ISBN: 978-989-758-670-5

DOI: 10.5220/0000168300003543

Depósito Legal: 519637/23

https://icinco.scitevents.org

icinco.secretariat@insticc.org

Brief Contents

INVITED SPEAKERS

Luís Paulo Reis
University of Porto
Portugal

Wim Michiels
KU Leuven
Belgium

Anuradha Annaswamy
MIT
United States

Sergio M. Savaresi
Politecnico di Milano
Italy

Organizing Committees

Conference Chair

Dimitar Filev, Research & Advanced Engineering, Ford Motor Company, United States

Program Co-chairs

Giuseppina Gini, Politecnico di Milano, Italy
Henk Nijmeijer, Eindhoven University of Technology, Netherlands

Local Chairs

Paolo Di Giamberardino, Sapienza University of Rome, Italy
Daniela Iacoviello, Sapienza University of Rome, Italy

Secretariat

Ana Rita Paciência, INSTICC, Portugal

Graphics Production and Webdesigner

André Fernandes, INSTICC, Portugal
Inês Teles, INSTICC, Portugal

Webmaster

João Francisco, INSTICC, Portugal
Carolina Ribeiro, INSTICC, Portugal

Program Committee

Hussein Abdullah, University of Guelph, Canada

El-Houssaine Aghezzaf, Ghent University, Faculty of Engineering and Architecture, Belgium

Eugenio Aguirre, University of Granada, Spain

Rudy Agustriyanto, University of Surabaya, Indonesia

Mojtaba Ahmadieh Khanesar, University of Nottingham, United Kingdom

Carlos Aldana, University of Guadalajara, Mexico

Manuel Aleixandre, Tokyo Institute of Technology, Japan

Joaquin Alvarez, Center Scientific Research Higher Education Ensenada Cicese, Mexico

Mihail Antchev, Technical University - Sofia, Bulgaria

Rui Araujo, University of Coimbra, Portugal

Mohd Ashraf Ahmad, University Malaysia Pahang, Malaysia

Ramiro Barbosa, ISEP/IPP - School of Engineering, Polytechnic Institute of Porto, Portugal

Michele Basso, University of Florence, Italy

Eric Baumgartner, Milwaukee School of Engineering, United States

Mahmoud Belhocine, CDTA, Algeria

Juri Belikov, Tallinn University of Technology, Estonia

Karsten Berns, University of Kaiserslautern-Landau, Germany

Sylvain Bertrand, ONERA - Univ. Paris-Saclay, France

Zafer Bingul, Kocaeli University, Turkey

Claudia-Adina Bojan-Dragos, Politehnica University of Timisoara, Romania

Magnus Boman, The Royal Institute of Technology, Sweden

Mohammad Bozorg, Yazd University, Iran, Islamic Republic of

Richard Braatz, Massachusetts Institute of Technology, United States

Laurent Burlion, Rutgers, the State University of New Jersey, United States

Simeon C. Calvert, TU Delft, Netherlands

Kenneth Camilleri, University of Malta, Malta

Enrique Carrera, Army Polytechnic School Ecuador, Ecuador

Marco Castellani, University of Birmingham, United Kingdom

Paul Christodoulides, Cyprus University of Technology, Cyprus

Feng Chu, University of Evry Val d'Essonne, France

Marco Costanzo, Università degli Studi della Campania Luigi Vanvitelli, Italy

Michel Dambrine, Polytechnic University of Hauts-de-France, France

Paolo Di Giamberardino, Sapienza University of Rome, Italy

António Dourado, University of Coimbra, Portugal

Marc Ebner, Ernst-Moritz-Arndt-Universität Greifswald, Germany

Mehmet Önder Efe, Hacettepe Univ., Turkey

Ali Eydgahi, Eastern Michigan University, United States

Mohammed Fadali, UNR, United States

Baris Fidan, University of Waterloo, Canada

Paolo Fiorini, University of Verona, Italy

Thierry Fraichard, INRIA, France

Giuseppe Franze', University of Calabria, Italy

Eduardo Oliveira Freire, Federal University of Sergipe, Brazil

Georg Frey, Automation and Energy Systems, Saarland University, Germany

Toyomi Fujita, Tohoku Institute of Technology, Japan

Andrej Gams, Jožef Stefan Institute, Slovenia

Péter Gáspár, SZTAKI, Hungary

Eduardo Godoy, São Paulo State University (UNESP), Brazil

Paulo Miyagi, University of Sao Paulo, Brazil

Héctor Montes, Technological University of Panama, Panama

Rafael Morales, University of Castilla La Mancha, Spain

Vidal Moreno Rodilla, Universidad De Salamanca, Spain

George Moustris, National Technical University of Athens, Greece

Riccardo Muradore, University of Verona, Italy

Naresh N. Nandola, Siemens Technology, United States

Ciro Natale, Università degli Studi della Campania Luigi Vanvitelli, Italy

Juan J. Nieto, University of Santiago de Compostela, Spain

Shimon Nof, Purdue University, United States

Urbano Nunes, University of Coimbra, Portugal

Andrzej Obuchowicz, University of Zielona Góra, Poland

Fernando Osorio, USP - Universidade de Sao Paulo, Brazil

Stamatios Papadakis, Department of Preschool Education, Faculty of Education, University of Crete, Greece

Evangelos Papadopoulos, NTUA, Greece

Ju H. Park, Yeungnam University, Korea, Republic of

Igor Paromtchik, Robotic Technologies, France

Dariusz Pazderski, Poznan University of Technology, Poland

Qingjin Peng, University of Manitoba, Canada

Tadej Petric, Jožef Stefan Institute, Slovenia

Mark Post, The University of York, United Kingdom

Raul Marin Prades, Jaume I University, Spain

Radu-Emil Precup, Politehnica University of Timisoara, Romania

Kanty Rabenorosoa, Femto-ST Institute, France

Navid Razmjooy, Independent Researcher, Iran, Islamic Republic of

Oscar Reinoso, Miguel Hernandez University, Spain

Gerasimos Rigatos, Industrial Systems Institute, Greece

Paolo Rocco, Politecnico di Milano, Italy

Alejandro Rodriguez-Angeles, Cinvestav-IPN, Mexico

Raul-Cristian Roman, Politehnica University of Timisoara, Romania

Juha Röning, University of Oulu, Finland

Christophe Sabourin, Univ. Paris Est Creteil, LISSI, France

Antonio Sala, Universitat Politecnica de Valencia, Spain

Addisson Salazar, Universitat Politècnica de València, Spain

Javier Sanchis, Universitat Politècnica de València, Spain

Jurek Sasiadek, Carleton University, Canada

Dieter Schramm, University of Duisburg-Essen, Germany

Karol Seweryn, Space Research Centre (CBK PAN), Poland

Antonio Sgorbissa, University of Genova, Italy

Madhavan Shanmugavel, SRM Institute of Science and Technology, India

Jinhua She, Tokyo University of Technology, Japan

Milad Siami, Massachusetts Institute of Technology, United States

Vasile Sima, Technical Sciences Academy of Romania, Romania

Azura Che Soh, Universiti Putra Malaysia, Malaysia

Paolo Stegagno, University of Rhode Island, United States

Adrian-Mihail Stoica, Polytechnic University Bucharest, Romania

Olaf Stursberg, University of Kassel, Germany

József Tar, Óbuda University, Hungary

Tomasz Tarczewski, Nicolaus Copernicus University, Poland

Daniel Thalmann, Ecole Polytechnique Federale de Lausanne, Switzerland

Gui Tian, Newcastle University, United Kingdom

Germano Torres, PS Solutions, Brazil

Andrés Úbeda, University of Alicante, Spain

José Valente de Oliveira, Universidade do Algarve, Portugal

Luigi Villani, Università di Napoli Federico II, Italy

Blas Vinagre, University of Extremadura, Spain

Antonio Visioli, University of Brescia, Italy

Damir Vrancic, Jožef Stefan Institute, Slovenia

Qiangda Yang, Northeastern University, China

Sho Yokota, Toyo University, Japan

Hongnian Yu, Bournemouth University, United Kingdom

Jie Zhang, Newcastle University, United Kingdom

Cezary Zielinski, Warsaw University of Technology, Poland

AUXILIARY REVIEWERS

Fabiano Correa, University of Sao Paulo, Brazil

Raul Cruz-Morales, UNAM-FES-Cuautitlan, Mexico

Luís Garrote, ISR-UC, Portugal

Vesna Javor, Independent Researcher, Serbia

Abdelhai Lati, University of Ouargla, Algeria

Balazs Nemeth, Institute for Computer Science and Control, Hungary

Pouya Panahandeh, University of Waterloo, Canada

João Paulo, Instituto de Sistemas e Robótica (ISR-UC), Portugal

Ricardo Pereira, Instituto de Sistemas e Robótica ISR-UC, Portugal

Nenad Petrovic, University of Nis, Faculty of Electronic Engineering, Serbia

Rafal Sobanski, Poznan University of Technology, Poland

Xinyue Wang, University of Paris-Saclay, France

SELECTED PAPERS BOOK

A number of selected papers presented at ICINCO 2023 will be published by Springer in a LNEE Series book. This selection will be done by the Conference Chair and Program Co-chairs, among the papers actually presented at the conference, based on a rigorous review by the ICINCO 2023 Program Committee members.

FOREWORD

This book contains the proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2023), held in Rome, Italy from 13 – 15 November, 2023.

ICINCO is sponsored by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC), technically co-sponsored by the International Federation of Automatic Control (IFAC) and held in cooperation with the ACM Special Interest Group on Artificial Intelligence, Association for the Advancement of Artificial Intelligence (AAAI) and the International Neural Network Society (INNS).

The ICINCO conference series has now become a major forum to debate technical and scientific advances presented by researchers and developers both from academia and industry, working in areas related to Control, Automation and Robotics that benefit from Information Technology.

The high quality of the ICINCO 2023 program is enhanced by four keynote lectures, given by internationally recognized researchers, namely: Luís Paulo Reis (University of Porto, Portugal), Wim Michiels (KU Leuven, Belgium), Anuradha Annaswamy (MIT, United States), and Sergio M. Savaresi (Politecnico di Milano, Italy).

ICINCO 2023 received 180 paper submissions from 47 countries of which 20% were accepted as full papers. To evaluate each submission, a double-blind paper review process was performed and lead by the Program Committee. As in previous editions of the conference, based on the reviewer's evaluations and the presentations, selected authors with the best papers will be invited to submit extended versions for a special issue in the Springer Nature Computer Science Journal, and a book in the Springer Lecture Notes in Electrical Engineering series.

We would like to express our thanks and appreciations to all participants. First, to the authors, whose quality work is the essence of this conference. Second, to all the members of the Program Committee and all reviewers, who helped with their expertise and valuable time they have invested in reviewing submitted papers. We would also like to deeply thank the invited speakers for their excellent contributions and sharing their knowledge and vision. Finally, a word of appreciation for the hard work of the INSTICC team; organizing a conference of this level is a task that can only be achieved by the collaborative effort of a dedicated and highly capable team.

We hope that you will enjoy the conference, the discussions with other participants, and the nice Italian atmosphere.

Finally, we look forward to have additional research results presented at the next edition of ICINCO.

Giuseppina Gini
Politecnico di Milano, Italy

Henk Nijmeijer
Eindhoven University of Technology, Netherlands

Dimitar Filev
Research & Advanced Engineering, Ford Motor Company, United States

CONTENTS

SHORT PAPERS

ROBOTICS AND AUTOMATION

FULL PAPERS

Comparative Analysis of Segmentation Techniques for Reticular Structures

Francisco J. Soler[1][a], Luis M. Jiménez[1][b], David Valiente[1][c], Luis Payá[1][d] and Óscar Reinoso[1,2][e]

[1] *Engineering Research Institute of Elche (I3E), Miguel Hernandez University, Elche, Spain*

[2] *ValgrAI: Valencian Graduate School and Research Network of Artificial Intelligence, Valencia, Spain*

{*f.soler, luis.jimenez, dvaliente, lpaya, o.reinoso*}*@umh.es*

Keywords: Plane Segmentation, Point Clouds, Region Growing, RANSAC, Neural Networks, Climbing Robots.

Abstract: Nowadays neural networks are widely used for segmentation tasks and there is a belief that these approaches are synonymous of advances and improvements. This article aims to compare the performance of a neural network, trained in our previous work, and an algorithm which is specifically designed for the segmentation of reticular structures. As shown in this paper, in certain cases it is feasible to use conventional techniques outside the paradigm of artificial intelligence achieving the same performance. To prove this, in this article a quantitative and qualitative comparative analysis is carried out between an ad hoc algorithm for segmenting reticular structures and the model of neural network that provided the best results in our previous work in this task. Established techniques such as Random Sample Consensus (RANSAC) and region growing have been used to implement the proposed algorithm. For the quantitative analysis, standard metrics such as *precision*, *recall* and *f1-score* are used. These metrics will be calculated with a self-generated dataset, consisting of a thousand point clouds that were generated automatically in the previous work. The studied algorithm is tailor-made for this database. For reproducibility, code and datasets are provided at https://github.com/Urwik/rrss_grnd_filter.git.

1 INTRODUCTION

Most existing large-scale buildings use lattice systems as structural elements due to their outstanding mechanical properties. These properties enable them to withstand high loads, achieve a balanced distribution of forces, exhibit high rigidity, and demonstrate efficiency in terms of material usage.

Due to these excellent properties, lattice structures (Figure 1) are widely used in high-voltage transmission lines, tower cranes, bridges, and other large-scale infrastructures. Typically, these structures are assembled using metallic bodies, in most cases these bodies are composed of flat surfaces, as seen in the case of transmission lines. In other cases these structures can be built with cylindrical parts.

As a general rule, these infrastructures respond adequately to adverse weather conditions and hostile environments, nevertheless require regular maintenance and inspection. A constantly evolving field of research involves the use of climbing robots to perform these types of tasks (Fang and Cheng, 2023). Climbing robots are devices known for their ability to move and operate on various surfaces, both horizontal and vertical, such as walls, ceilings, or metallic structures. Additionally, they can perform multiple inspection or maintenance tasks in complex environments that are difficult to access and pose significant risks to human operators, who may be exposed to various hazards such as falls or electric shocks.

These types of inspections and maintenance tasks have been carried out by aerial robots in recent years ((Akahori et al., 2016), (Jung et al., 2019)). However, quite often this type of robotic platform is unable to complete such tasks due to its limitation in accessing internal areas of the structures.

In order to carry out such tasks effectively, it is necessary to have a proper environmental perception. One of the sensors most widely used today for sensing surroundings are the so-called LiDAR. These sensors, widely used in numerous applications today, such as

[a] https://orcid.org/0009-0006-7396-6596
[b] https://orcid.org/0000-0003-3385-5622
[c] https://orcid.org/0000-0002-2245-0542
[d] https://orcid.org/0000-0002-3045-4316
[e] https://orcid.org/0000-0002-1065-8944

Soler, F., Jiménez, L., Valiente, D., Payá, L. and Reinoso, Ó.
Comparative Analysis of Segmentation Techniques for Reticular Structures.
In *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2023) - Volume 1*, pages 413-423
ISBN: 978-989-758-670-5; ISSN: 2184-2809

Figure 1: Example of reticular structure used in an industrial building.

localization (Liu et al., 2022), map building (Zhou et al., 2021), object detection and segmentation (Zhu et al., 2021), provide excellent range information. Their high accuracy in providing detailed information about the spatial distribution of the environment has made this type of sensor widely used for localisation and navigation in robotics in recent years.

A key aspect for accurate navigation is the clear identification of flat surfaces in the surrounding environment (Xu et al., 2020). The presence of planes in the structure allows us to build a parametric representation of it, enabling a lightweight environment model. Consequently, detecting these types of elements in the environment is a task of interest for navigation in lattice structures.

Providing a climbing robot with a LiDAR sensor allows it to know the spatial distribution of the surrounding environment. Combined with the ability to identify planar surfaces, it may constitute an ideal solution to address navigation tasks in lattice structures, where virtually all components are made up of planes. By combining the information from the LiDAR sensor and the detection of flat surfaces, the robot can effectively navigate and interact with the structure, leveraging its knowledge of the planar elements within the environment.

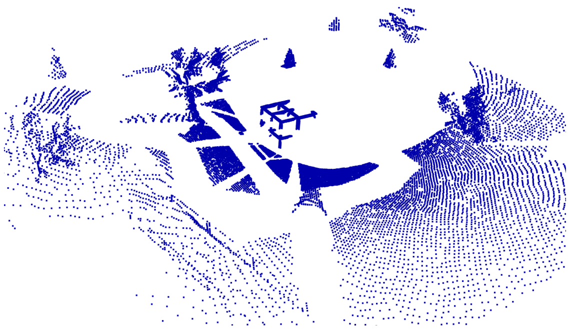


Figure 2: Example of captured cloud by the simulated sensor.

Indeed, LiDAR sensors often capture a large amount of information, sometimes more than necessary (Figure 2). Therefore, it becomes crucial to remove undesired data. At first, the main objective to enable the navigation of the robotic platform throughout this type of structures, is to identify from all the information provided by the available sensory systems, the information relating only to the structure, being necessary to identify which part of the information belongs to it and which does not. This task could be defined as a per point classification or a segmentation of the original information into certain classes.

In recent years, we can find related works that employ artificial intelligence systems and algorithms to address this problem. Thus, we find some works on plane segmentation with neural networks such as (Yang and Kong, 2020) or (Lee and Jung, 2021). In these works, neural networks are used to identify planes in indoor and outdoor environments (urban environments) respectively. However, these environments differ significantly from the target environments of our work, which is why in previous works in the research group we approached this task with a segmentation proposal using specific neural networks (Soler et al., 2023).

Additionally, there are many methods for plane identification with algorithms outside the artificial intelligence paradigm as in (Su et al., 2022). It proposes a two-step segmentation, a first stage where planes are selected by region growing and a second stage where the border points between two planes are classified, where the region growing algorithm is not able to work correctly. In (Gaspers et al., 2011) they similarly use a two-stage segmentation but over multiple resolutions. They extract for each resolution key features based on the normals named *surfels*. These *surfels* are intended to be associated with planes at lower resolutions. Those surfels that are not associated with any known plane are attempted to be grouped according to their coplanarity using the Hough transform. On the other hand, RANSAC is applied to the sets of surfels that have been associated with the same plane at lower resolutions to improve accuracy. Once the maximum resolution is reached, nearby coplanar segments are identified and merged into a single plane.

The algorithm proposed in this study is similar to the ones mentioned above, as it employs a two-stage strategy, but unlike the previous ones (segmenting the point cloud provided by the LiDAR sensor into multiple flat sets) its objective is to split the point cloud only into two sets, structure and non-structure. For this purpose, a coarse classification by RANSAC is used and then a region growing procedure is performed to improve the result.

The aim of this article is to evaluate the performance of neural networks against the application of such specific algorithm for segmenting reticular structures in a specific environment. To do so, we will compare our previous work, (Soler et al., 2023), with a proposed application-specific algorithm using a dataset that contains point clouds obtained from reticular structures.

To provide a clear overview, this articles is divided into the following parts. Second section (2) briefly recounts the previous work in order to put the reader in context for further comparison. Section 3 explores the proposed method for segmenting the lattice structures of our database and discusses all its steps. Section 4 presents the conducted experiments. Then, Section 5 develops a comparative analysis between the segmentation of reticular structures using neural networks and conventional approaches. Finally, Section 6 discusses some brief reflections about the obtained results.

2 PREVIOUS WORK

In this section, we provide a brief overview of the fundamental idea of our previous work. In previous studies (Soler et al., 2023), an specific training of neural networks was made to identify reticular structures based on environment information provided by a LiDAR sensor.

Reticular structures are interconnected systems by rigid joints forming a three-dimensional lattice configuration. This type of system can be found in a multitude of infrastructures, such as bridges, buildings, electricity pylons or cranes, and is normally made by metallic elements composed by multiple flat surfaces.

The aforementioned work was carried out with the purpose of being implemented in the HyReCRo (Peidro et al., 2015) robot. This series-parallel climbing robot has ten degrees of freedom (DOF) and has the ability to navigate through metallic structures by using a magnetic adhesion mechanism based on mechanically switched permanent magnets.

2.1 Dataset Generation

One of the first challenges to be solved in order to meet the objective of previous work was the lack of training dataset.

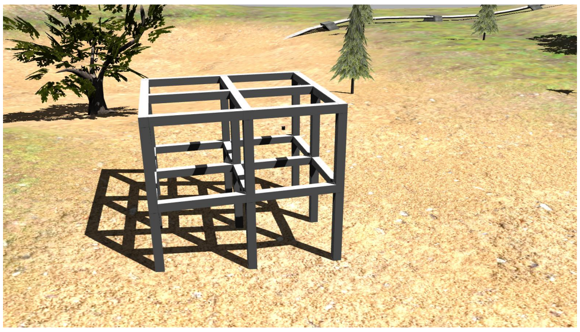
There are recent studies on the use of simulators for the automatic generation of labelled datasets. In (Sanchez et al., 2019) Gazebo Simulator is used to generate labelled 3D scans of natural environments, but it has a limitation when it comes to generating large datasets, as the position of the sensor has to be indicated by the user. In a more autonomous way, in (Wang et al., 2019) a modification of the CARLA simulator (Dosovitskiy et al., 2017) is used to generate driving LiDAR point clouds with per point automatic labels during the movement of a vehicle. Moreover, there are studies that merge real and synthetic information such as (Fang et al., 2018), where similar to the work mentioned above its objective is to generate driving data. To achieve that, they use a static labelled point cloud as a background and introduce synthetic elements, like cars or people, with pre-defined labels in order to obtain a more realistic representation of the environment.

Figure 3: Example of environment used for training. Red circle indicates the position of the sensor.

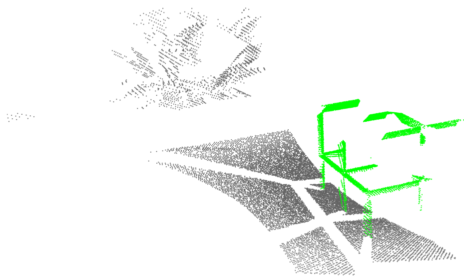
The works mentioned above mainly focus on autonomous driving navigation and segmentation of such environments. To address the segmentation of lattice structures where the environment differs significantly from the previous ones, we have developed a plugin in Gazebo Simulator. Similar to (Sanchez et al., 2019), it uses this simulation software to generate labelled datasets automatically, with the advantage that the position of the sensor and the elements of the environment change automatically, enabling the generation of large databases.

The training database was conformed by ten thousand point clouds simulating the properties of a real LiDAR sensor (Ouster OS1-128 channels). Each measure is taken from the sensor origin and the sensor pose is set randomly around the environment. With the objective of generalising the database for a variety of lattice structures, the training dataset is formed using environments composed of parallelepipeds and elements such as trees and soil modelling a real environment (Figure 3).

In the same way as for the training data, the evaluation dataset (which is the one used to evaluate the metrics in this paper) has been generated automati-



(a) Example of the evaluation environment.



(b) Example of the point cloud generated with automatic labels.

Figure 4: Example of the evaluation dataset.

cally. It is composed of 904 point clouds and with a lattice structure model instead of parallelepipeds. A representation of the evaluation environment and the simulated data are shown in the Figure 4.

Different neural network architectures were trained and analysed for reticular structure segmentation, PointNet (Qi et al., 2016), PointNet++ (Qi et al., 2017) and MinkUNet34C (Choy et al., 2019). The best results were obtained with the MinkUNet34C architecture, which uses the MinkowskiEngine to perform 3D convolutions in a sparse way, only on those points that contain information. The latter architecture shows better results in terms of recall and f1 score than the others as shown in Figure 5, therefore it has been selected as the best model for comparison.

Their results are discussed and compared in further detail with the algorithm presented in the present article in Section 5.

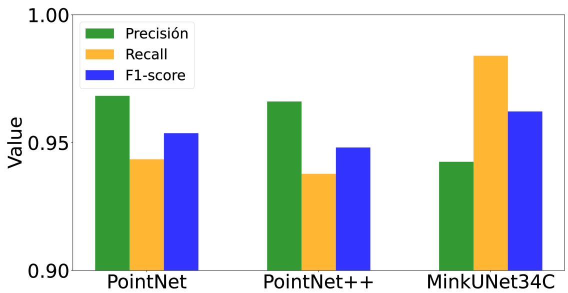


Figure 5: Metrics obtained in previous work.

3 PROPOSED ALGORITHM

The algorithm proposed in this study is structured in two steps. The first step performs a coarse classification, identifying the ground plane and elements close to it. Secondly, a process is run in which the classification is refined. This method adopts well-known algorithms in the literature for plane segmentation and identification such as region growing or Random Sample Consensus (RANSAC) with the aim of classifying points into two classes, structure and non-structure.

It is important to notice that this algorithm has been specifically designed to work in the environments that have been generated for the test dataset in our previous work. While defining its behaviour, it is taken into consideration that within the sensor reading there will be a large number of points belonging to the ground, in addition to the fact that these occupy a large area of the environment. Figure 6 shows a flowchart that describes the process followed to complete the segmentation. The implementation of this work relies on the *Point Cloud Library* (PCL) (Rusu and Cousins, 2011) library to perform the point cloud processing. The following subsections describe each of its stages in more detail.

To reduce the computational cost of the algorithm, a two-stage approach is adopted. Based on the preliminary experiments, the highest computational cost of the algorithm is due to the normal estimation, a process that consumes about 80% of the total execution time of the algorithm. If the fine classification step is used directly, it would be necessary to estimate the normals of the entire cloud, in addition to having to calculate and evaluate the eigenvalues of a larger number of ensembles. Such an approach would require on average 20% more runtime per cloud to obtain the same results.

3.1 Coarse Segmentation

In the initial stage of the algorithm a coarse classification is performed to extract the ground points. This stage involves Voxel filtering followed by the extraction of the largest plane in the environment using RANSAC.

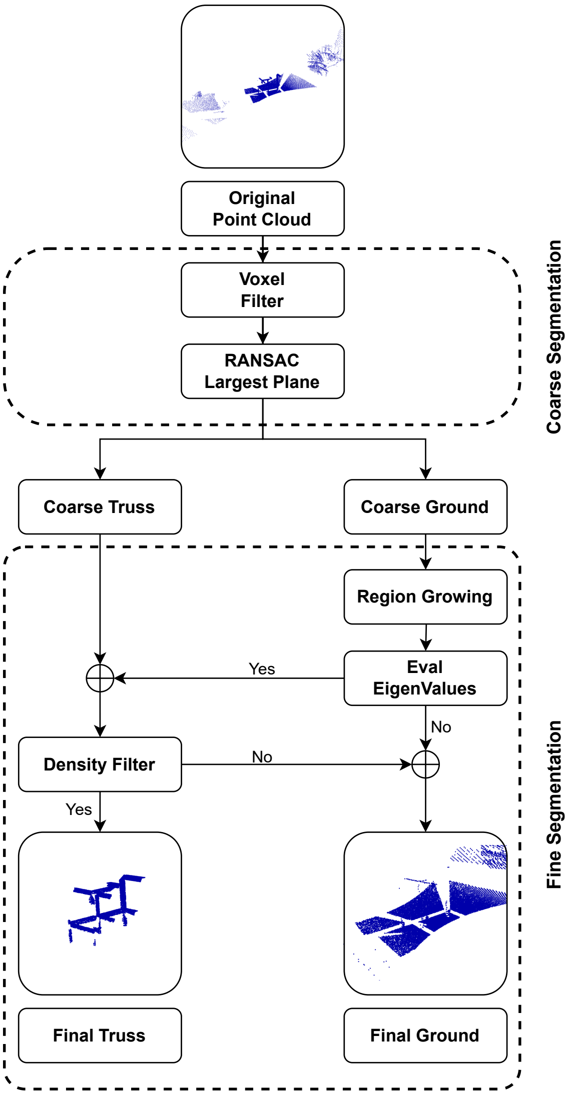


Figure 6: Flowchart of the proposed algorithm.

Since RANSAC selects the best plane candidate based on the number of inliers, if there are areas with high point density, RANSAC tends to select planes in these areas. To avoid this problem, Voxel filtering is first applied, a process by which the point density is homogenised across the entire cloud, thus favouring the extraction of the largest plane (ground plane).

In order to cover as many ground points as possible, a high threshold is established to obtain the largest plane, around 0.5 metres, thus avoiding slopes in the terrain.

3.2 Fine Segmentation

Coarse classification produces a smaller cloud containing ground points as well as points close to it within a certain threshold. On this reduced cloud, a more accurate classification is applied. The idea of this stage is to split the cloud into planar clusters and to classify them according to their size.

In order to segment the different planar clusters, region growing based on the normals is applied. The clustering process is based on the similarity of the normals of nearby points, so the estimation of these features is a very important aspect in the resultant clusters.

In the following subsections, the normal estimation for each point and the decision criteria for cluster filtering are further discussed.

3.2.1 Normal Estimation

As mentioned in the previous section, normal estimation is a key component in establishing correct planar clusters. The normal estimation function implemented in PCL consists of computing the eigenvalues and eigenvectors over the neighbourhood of each point, where the eigenvector associated with the smallest eigenvalue is considered the normal vector of the point. Eigenvalues and eigenvectors are obtained by principal component analysis (PCA) on the covariance matrix for each point and its neighbourhood environment.

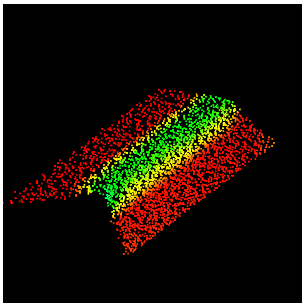
This matrix (C) follows the formulation indicated in Equation 1, where k is the number of neighbors, $\overline{p}$ is the centroid of the neighbor set, λ_j is the eigenvalue, and $\vec{v}_j$ is the eigenvector for j.

$$C = \frac{1}{k}\sum_{i=1}^{k} \cdot (p_i - \overline{p}) \cdot (p_i - \overline{p})^T \qquad (1)$$
$$C \cdot \vec{v}_j = \lambda_j \cdot \vec{v}_j, \; j \in \{0,1,2\}$$

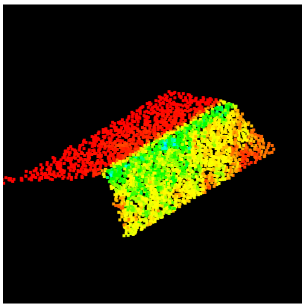
The normal estimation method described in the previous paragraph requires the selection of a set of neighbouring points to fulfil its task. The implementation of the library allows two exclusive options for this purpose: to select all points located within a sphere of defined radius, or to select those closest points with a limit in number.

Depending on the neighbour selection method, the eigenvectors and eigenvalues can differ significantly. An example of this can be seen in Figure 7 where the point cloud is represented as a function of its curvature (2), defined as the coefficient between the smallest eigenvalue and the sum of all eigenvalues. In Figure 7(a) neighbours are selected by radius and in Figure 7(b) by nearest neighbours.

$$Curvature = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \qquad (2)$$



(a) By radius



(b) By neighbours

Figure 7: Point clouds displayed by its curvature.

Based on the conducted experiments, it has been concluded that the estimation of the normal vector of each point is more accurate when a certain number of close neighbours are used. The normal estimation by radius is highly erroneous in areas with a low density of points, i.e. there are not enough points in the indicated radius to estimate the normal vector. In contrast, the calculation of curvature, which is somewhat indicative of point spread, is more accurate when neighbour selection by radius is used, as seen in Figure 7, where in the radius selection method, points with high curvature correspond to the cutting edges of the different planes of the structure. The curvature value is important since, as described below, it is used as a termination criteria for growing regions. As indicated in (Pauly et al., 2002), if we look at equation (2), the maximum curvature value will be $\lambda_{max} = 1/3$ which occurs when $\lambda_0 = 1$, since $\lambda_0 \leq \lambda_1 \leq \lambda_2$ and $\lambda_{0-2} \in \{0, 1\}$.

This means that small values of curvature indicate that the points are poorly sparse along the smallest eigenvector (most of the points fall on a plane) and values close to 1/3 in curvature mean that the points are uniformly sparse throughout the selected neighbourhood space.

3.2.2 Region Growing

The region growing method is a frequently employed approach for detecting and grouping data sets into cohesive regions. It operates on the fundamental premise that neighboring points exhibiting similar attributes should be grouped together.

In this study, the existing implementation in the PCL library is used, which facilitates region growth based on the normal vector associated with each point. Primarily, it is imperative to estimate the normals and their corresponding curvature. Once this estimation is accomplished, the initial seed point is selected based on the lowest curvature value across the entire point cloud. Subsequently, the cluster's size expands by incorporating neighboring points that satisfy specific criteria. In order to include a new point in the set, two conditions must be satisfied. The first condition involves assessing the angular disparity between the point's normal and the initial seed's normal. If the difference falls below a specific threshold, the new point is incorporated into the set. The second condition entails examining the curvature of the newly added points. If the curvature is below a certain threshold, these points become new seeds for further expansion of the set. This growth process continues until no more seeds are available, indicating the completion of set expansion.

3.2.3 Eigenvalues Evaluation

To determine whether a region belongs to the structure or not, the eigenvalues and eigenvectors are employed as indicators. These properties are derived through a principal component analysis conducted on the covariance matrix of the local neighborhood surrounding a specific point, similar to the method described in the preceding section for normal estimation.

The algorithm has been assessed with three different variations, differing in the approach utilized to determine whether a cluster of points belongs to the structure or not. These variations focus on distinct methods of utilizing eigenvalues or eigenvectors for this discrimination process.

By Ratio. This particular variant builds upon existing knowledge of the structure, assuming that structural planes exhibit an elongated and slender geometry. Taking this into account, this variant relies solely on the ratio between the two largest eigenvalues (referred to as the "ratio" in Equation 3) to calculate a value representing the length-to-width ratio of the candidate set. This value is then utilized to determine the classification of the cluster in question.

$$ratio = \frac{\lambda_1}{\lambda_2} \quad (3)$$

By Module. Applying a similar methodology to the previous variant, having prior knowledge of the bar-like geometry of the structure enables filtering based on the projection module of the farthest point along its principal axes. Thresholds are then defined for the two dimensions of the plane formed by the set of points.

Groups of points exceeding the specified thresholds (maximum length and width of the structural elements) are excluded from the classification as part of the structure.

Hybrid. In the final variant, a combination of the two preceding approaches is employed, incorporating both the module filtering and the ratio-based classification. This integrated method yields the most favorable outcomes, as evident from the results presented in Table 1.

3.2.4 Density Filter

Lastly, considering the proximity of the robot and sensor to the structure, it is expected that areas belonging to the structure will exhibit a higher point density, while the ground and surrounding environment will show lower density. Taking advantage of this observation, the final step aims to eliminate remaining spurious points in the environment, retaining only the points corresponding to the structure. This step effectively filters out outlier points, resulting in a representation that solely encompasses the desired structure.

3.3 Limitations

It is important to emphasize that the algorithm is specifically tailored for the available dataset, which encompasses a wealth of information regarding the environment and the ground. This dataset is obtained through a realistic simulation of the commercial Ouster OS1 LiDAR sensor, ensuring that the generated data adheres to the sensor's specifications. The configuration of the simulated sensor includes a resolution of 512x128 points, a maximum range of 30 meters, vertical and horizontal fields of view of 45 degrees and 360 degrees respectively. Additionally, the dataset incorporates Gaussian noise with a mean of zero and a standard deviation of 0.008 meters. The extensive fields of view and range of the sensor enable comprehensive information capture from the ground and surrounding environment, enabling the algorithm's initial stage to successfully identify the ground plane.

4 EXPERIMENTS

The hardware used for the experiments is as follows. An NVIDIA RTX3090 graphics card in the case of the neural network. An Intel i7-10700 processor for all variants of the proposed ad hoc algorithm. The results obtained during the experiments are shown in Table 1, where the metrics evaluated (Section 5.1) are listed together with the execution time in milliseconds.

4.1 Ratio

Experiments with this algorithm variant have been performed with a threshold for the ratio given by the prior knowledge of the structure, taken as the quotient between the width of the beams and the height mentioned in Section 3.1 by which the points close to the estimated ground plane are selected. Assessing clusters with this variant yields poor results because it is not able to classify correctly large clusters belonging to the ground, whose elongated proportions are similar to those of the beams.

4.2 Module

For this approach, the threshold has been taken as the height above which points close to the ground are selected to obtain a coarse classification. This distance is taken as the threshold since it is the maximum bar length visible after coarse classification. The results of this variant are considerably better than the previous one, since it is able to identify the clusters obtained according to their size, thus overcoming the problem of the previous method. Despite this, it is possible that the clusters meet the size requirement, but not the ratio requirement (Figure 8).

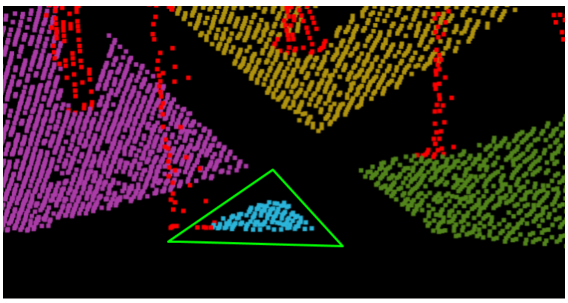


Figure 8: Example of a cluster that can not be correctly classified by its module.

Table 1: Evaluated results in the comparison. W/O = without.

	Precision	Recall	F1-Score	TP	FP	TN	FN	Execution Time (ms)
MinkUNet34C	0,9425	**0,9840**	0,9622	**10213**	623	14006	**158**	45
RATIO	0,6972	0,9658	0,7961	10024	5133	9492	349	62
MODULE	0,9920	0,9645	**0,9775**	10007	70	14555	366	72
HYBRID	0,9922	0,9643	**0,9775**	10004	67	14558	368	72
W/O Coarse Seg	**0,9952**	0,4422	0,6123	4587	**22**	**14603**	5786	89
W/O Fine Seg	0,9935	0,9575	0,9744	9940	52	14574	433	**31**

4.3 Hybrid

The same thresholds are used for this variant as for the previous ones. The hybrid filter allows us to take into account those clusters that meet the modulus requirement but do not meet the ratio requirement, as in the example in Figure 8. This result is not significant in the evaluated metrics as it hardly appears in the available data (only when there are certain occlusions). In Figure 8, an example of this type of situation is shown, where a cluster that meets the module requirements, does not meet the ratio requirements and therefore has to be discarded.

4.4 Without Coarse Segmentation

This experiment has been carried out with the hybrid method as it is the most complete for the given task. Applying directly a fine classification, i.e. region growing to segment the input cloud does not provide the best results. This fact is supported by the results in Table 1.

In the later one, it can be seen that in this case a good precision is achieved, which implies that those points identified as structure are indeed structure. On the other hand, its recall is only close to 50%, which means that only this percentage of the structure can be identified.

This method is mainly based on the accurate estimation of the clusters by region growing, which depends on a multitude of parameters and requires an exhaustive adjustment of these for an ideal performance. By evaluating each of the clusters formed using the decision criteria (hybrid criteria in this case), the sets are classified as structure and non-structure. In order to obtain better results with this method, it would be necessary to carry out an improvement process to adjust the parameters of normal estimation and region growing.

Besides, in this case it is necessary to adjust the thresholds used in the previous experiments, setting as module the maximum length of the bars of the structure (since they are complete and not trimmed) and consequently the ratio with mentioned length.

4.5 Without Fine Segmentation

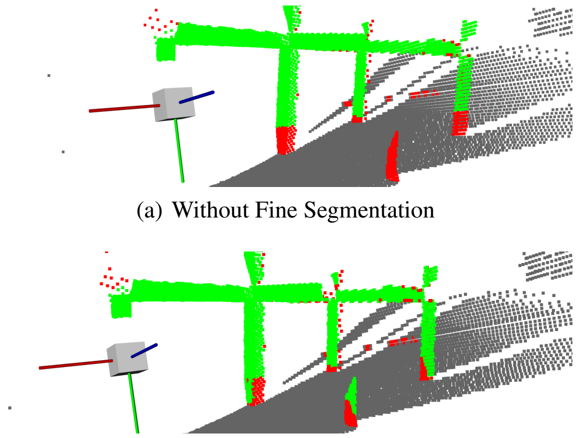
A further studied scenario is the use of the algorithm without the fine classification section. Its accuracy is very high, because with ground voxelization and RANSAC we are able to accurately identify the ground plane, as these are structures that rise from the ground, everything above a certain height is easily classified as structure. By using this method, the execution time can be reduced by almost half.

Despite these facts, this method is not able to identify the points where the structure meets the ground and discards all of them. Since the point density in these areas of the structure is not very high and their number is very small compared to the rest, the metrics evaluated are not affected to any large extent.

However, in order to obtain the best results, the fine classification stage is proposed to meet the needs of identifying areas where the structure meets the ground. Although its behaviour is far from ideal due to the reduced density of points in these areas, its use implies an improvement in certain cases. An example of this type of situation is shown in Figure 9, where the fine segmentation stage is able to identify the points corresponding to the elements of the structure in contact with the ground, improving the final classification. In this case, the use of this type of classification with respect to the hybrid method means an increase in precision, as expected, but a reduction in recall of around 4%.

4.6 Density Filter

Applying the density filter to the initial cloud or after the proposed methods has also been evaluated. It has been observed that the best results of the algorithm are obtained when the density filter is used last. This may be due to the larger amount of information available to the fine and coarse stages to operate.



(a) Without Fine Segmentation

(b) With Hybrid Method

Figure 9: Example of improvement using fine segmentation with the hybrid method versus just coarse segmentation.

5 COMPARATIVE ANALYSIS

To assess and compare the effectiveness of the aforementioned custom algorithm with the most successful neural network model derived from prior research, this section conducts a comparative analysis. The objective is to highlight the strengths and weaknesses of each method. First, the metrics used are presented and then the most important aspects of each method are discussed separately. Finally, the numerical results of both methods are discussed.

5.1 Evaluated Metrics

For the assessment of performance, we employ identical evaluation metrics as in our previous research, which are widely utilized to evaluate neural networks. These metrics include Precision, Recall, and F1-Score, which effectively measure the accuracy and effectiveness of the segmentation. Furthermore, we also evaluate the inference time, representing the average computational time needed to obtain the segmentation of an input point cloud. In the following equations TP, FP, TN, FN are the well-known parameters representing true positives, false positives, true negatives and false negatives respectively.

Precision (*Precision*) eq. (4) reflects the algorithm's or neural network's certainty or confidence level. In other words, it indicates the percentage of correct predictions.

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

Recall (*Recall*) eq. (5) measures the volume of data that we are able to predict correctly.

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

Finally, the F1-score (*F1-score*) eq. (6) is a metric that combines the previous ones, providing a single indicator of the overall performance of the process.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6}$$

5.2 Neural Network

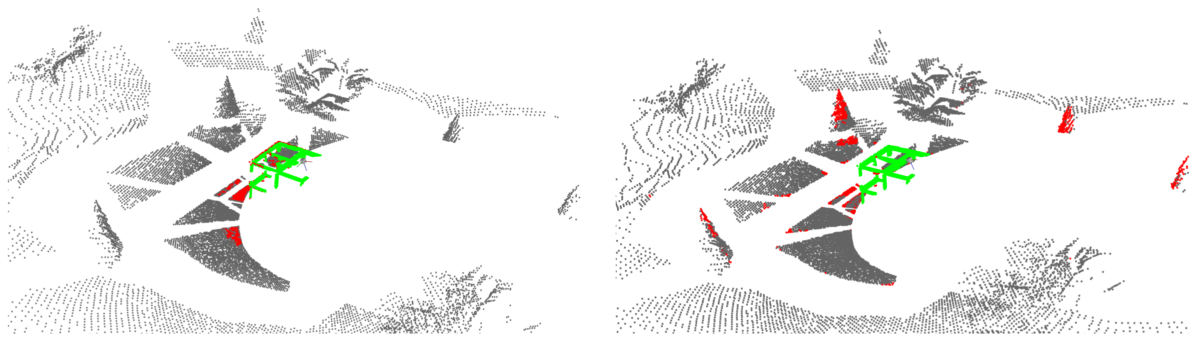
In this comparison, the neural network employed is MinkUNet34C, which is a 3D convolutional network that uses sparse convolutions. This architectural choice significantly reduces computational time compared to conventional convolutions. The network demonstrates promising outcomes in segmenting reticular structures and presents remarkable generalization capabilities. It successfully identifies various types of reticular structures across diverse environments, highlighting its versatility and adaptability.

Some of the drawbacks of the neural network include the need for a sufficiently large training and evaluation dataset to achieve good performance. In addition to this, the network requires a long training time, leading to extended waiting times whenever any configuration changes are applied. Moreover, the network requires specific hardware for fast execution. The recorded training times are approximately 12 hours on an NVIDIA RTX 3090 with 24GB of memory for a dataset consisting of 10.000 point clouds with 25.000 points per cloud.

5.3 Ad Hoc Algorithm

The method presented in this article attains remarkably good results, compared to those achieved by the neural network. Notably, it offers several advantages over the neural network, including independence from specific databases, hardware requirements, and training time. In terms of training time, the algorithm possesses a significant advantage as parameter adjustments can be made, and immediate results can be obtained without the need for a complete learning process. Furthermore, the algorithm has the capability to run on multiple CPU cores, thereby further reducing the execution time shown in this study.

Nevertheless, the primary limitation of this algorithm resides in its lack of generalizability. Customization and adaptation of the algorithm to each specific environment and structural geometry are imperative for its effective application.



(a) Results of the proposed algorithm. (b) Results of MinkUNet34C

Figure 10: Examples of the segmentation performed by both methods. Red color shows classification errors.

5.4 Results

Examining the results presented in Table 1, it is evident that the modular and the hybrid versions of our algorithm outperform the neural network in terms of precision, resulting in an improvement of around 5% in this metric. Conversely, the neural network exhibits higher recall (98%), denoting its capability to identify a greater percentage of structure points. The F1-score, encompassing both precision and recall, shows a 1% improvement in the proposed algorithm over the neural network. Figure 10 visually illustrates the striking similarity in results obtained by both approaches, corroborating the findings outlined in Table 1.

6 CONCLUSIONS

The findings from this study underscore that neural networks are not always the optimal choice for every task. Remarkably similar outcomes to those of neural networks can be achieved without the need of a training process, which requires a labelled dataset for training and subsequent evaluation. Upon careful examination of the comparative results, it becomes evident that an algorithm specifically tailored for the desired task, with a shorter development time compared to neural networks, can be more advantageous in certain scenarios. Furthermore, the proposed algorithm can be executed in parallel, significantly reducing the current execution time and enabling its utilization on small mobile devices with limited computational capabilities.

ACKNOWLEDGEMENTS

This work is part of the project PID2020-116418RB-I00 funded by MCIN/AEI/10.13039/501100011033.

The present research has also been possible thanks to the project TED2021-130901B-I00, funded by MCIN/AEI/10.13039501100011033 and the European Union ”NextGenerationEU”/PRTR.

REFERENCES

Akahori, S., Higashi, Y., and Masuda, A. (2016). Development of an aerial inspection robot with epm and camera arm for steel structures. In *2016 IEEE Region 10 Conference (TENCON)*, pages 3542–3545.

Choy, C., Gwak, J., and Savarese, S. (2019). 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084.

Dosovitskiy, A., Ros, G., Codevilla, F., López, A. M., and Koltun, V. (2017). CARLA: an open urban driving simulator. *CoRR*, abs/1711.03938.

Fang, G. and Cheng, J. (2023). Advances in climbing robots for vertical structures in the past decade: A review. *Biomimetics*, 8(1).

Fang, J., Yan, F., Zhao, T., Zhang, F., Zhou, D., Yang, R., Ma, Y., and Wang, L. (2018). Simulating lidar point cloud for autonomous driving using real-world scenes and traffic flows. *ArXiv*, abs/1811.07112.

Gaspers, B., Stückler, J., Welle, J., Schulz, D., and Behnke, S. (2011). Efficient multi-resolution plane segmentation of 3d point clouds. In *4th International Conference on Intelligent Robotics and Applications (ICIRA)*, pages 145–156.

Jung, S., Song, S., Kim, S., Park, J., Her, J., Roh, K., and Myung, H. (2019). Toward autonomous bridge inspection: A framework and experimental results. In *2019 16th International Conference on Ubiquitous Robots (UR)*, pages 208–211.

Lee, H. and Jung, J. (2021). Clustering-based plane segmentation neural network for urban scene modeling. *Sensors*, 21:8382.

Liu, Y., Wang, C., Wu, H., Wei, Y., Ren, M., and Zhao, C. (2022). Improved lidar localization method for mobile robots based on multi-sensing. *Remote Sensing*, 14(23).

Pauly, M., Gross, M., and Kobbelt, L. (2002). Efficient simplification of point-sampled surfaces. In *IEEE Visualization, 2002. VIS 2002.*, pages 163–170.

Peidro, A., Gil, A., Marin, J., and Reinoso, O. (2015). Inverse kinematic analysis of a redundant hybrid climbing robot. *International Journal of Advanced Robotic Systems*, 12:1.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2016). Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413.

Rusu, R. B. and Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4.

Sanchez, M., Martínez, J., Morales, J., Robles, A., and Moran, M. (2019). Automatic generation of labeled 3d point clouds of natural environments with gazebo. pages 161–166.

Soler, F. J., Peidró, A., Fabregat, M., Payá, L., and Reinoso, O. (2023). Segmentación de planos a partir de nubes de puntos 3d en estructuras reticulares. In *XIII Jornadas Nacionales de Robótica y Bioingeniería*, pages 91–98, Madrid, Spain.

Su, Z., Gao, Z., Zhou, G., Li, S., Song, L., Lu, X., and Kang, N. (2022). Building plane segmentation based on point clouds. *Remote Sensing*, 14(1).

Wang, F., Zhuang, Y., Gu, H., and Hu, H. (2019). Automatic generation of synthetic lidar point clouds for 3-d data analysis. *IEEE Transactions on Instrumentation and Measurement*, 68(7):2671–2673.

Xu, Y., Ye, Z., Huang, R., Hoegner, L., and Stilla, U. (2020). Robust segmentation and localization of structural planes from photogrammetric point clouds in construction sites. *Automation in Construction*, 117:103206.

Yang, H. and Kong, H. (2020). 3dpmnet: Plane segmentation and matching for point cloud registration. In *2020 3rd International Conference on Unmanned Systems (ICUS)*, pages 439–444.

Zhou, L., Wang, S., and Kaess, M. (2021). Pi-lsam: Lidar smoothing and mapping with planes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5751–5757.

Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., and Lin, D. (2021). Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9939–9948.