

JORNADAS NACIONALES DE ROBÓTICA Y BIOINGENIERÍA 2023



14 al 16 de junio de 2023

E.T.S. Ingenieros Industriales

Universidad Politécnica de Madrid

ISBN: 978-84-09-51892-0

DOI: 10.20868/UPM.book.74896

Editores:

- Óscar Reinoso García,
- Eduardo Rocon de Lima,
- Manuel Ferre Pérez,
- Paloma de la Puente Yusty,
- Daniel Galán Vicente,
- Ramón Antonio Suárez Fernández, y
- Francisco Javier Badesa Clemente.

Índice

Libro de actas	i
Índice.....	iii
Presentación.....	v
Organizadores	vii
Colaboradores	viii
Comités Organizadores	ix
Índice de comunicaciones.....	x
Listado de autores.....	xiv

Presentación

Prólogo de los coordinadores

Las Jornadas de Robótica y Bioingeniería de 2023 tienen lugar en la Escuela Técnica Superior de Ingeniería Industrial de la Universidad Politécnica de Madrid, entre los días 14 y 16 de junio de 2023. En este evento propiciado por el Comité Español de Automática (CEA) tiene lugar la celebración conjunta de las XII Jornadas Nacionales de Robótica y el XIV Simposio CEA de Bioingeniería.

Las Jornadas Nacionales de Robótica es un evento promovido por el Grupo Temático de Robótica (GTRob) de CEA para dar visibilidad y mostrar las actividades desarrolladas en el ámbito de la investigación y transferencia tecnológica en robótica. Asimismo, el propósito de Simposio de Bioingeniería, que cumple ahora su decimocuarta edición, es el de proporcionar un espacio de encuentro entre investigadores, desarrolladores, personal clínico, alumnos, industriales, profesionales en general e incluso usuarios que realicen su actividad en el ámbito de la bioingeniería. Estos eventos se han celebrado de forma conjunta en la anualidad 2023.

Esto ha permitido aunar y congregar un elevado número de participantes tanto de la temática robótica como de bioingeniería (investigadores, profesores, desarrolladores y profesionales en general), que ha posibilitado establecer puntos de encuentro, sinergias y colaboraciones entre ambos.

El programa de las jornadas aúna comunicaciones científicas de los últimos resultados de investigación obtenidos, por los grupos a nivel español más representativos dentro de la temática de robótica y bioingeniería, así como mesas redondas y conferencias en las que se debatirán los temas de mayor interés en la actualidad.

En relación con las comunicaciones científicas presentadas al evento, se ha recibido un total de 46 ponencias, lo que sin duda alguna refleja el alto interés de la comunidad científica en las Jornadas de Robótica y Bioingeniería. Estos trabajos serán expuestos y presentados a lo largo de un total de 10 sesiones, distribuidas durante los diferentes días de las Jornadas. Las temáticas de los trabajos cubren los principales retos científicos relacionados con la robótica y la bioingeniería: robótica aérea, submarina, terrestre, percepción del entorno, manipulación, robótica social, robótica médica, teleoperación, procesamiento de señales biológicas, neurorehabilitación etc.

Confiamos, y estamos seguros de ello, que el desarrollo de las jornadas sea completamente productivo no solo para los participantes en las Jornadas que podrán establecer nuevos lazos y relaciones fructíferas entre los diferentes grupos, sino también aquellos investigadores que no hayan podido asistir. Este documento que integra y recoge todas las comunicaciones científicas permitirá un análisis más detallado de cada una de las mismas.

Óscar Reinoso García

Coordinador de CEA – GTRob

Eduardo Rocon de Lima

Coordinador de CEA – Bioingeniería

Organiza

Comité Español de Automática



Grupo Temático de Robótica



Grupo Temático de Bioingeniería



Centro de Automática y Robótica (CAR) UPM-CSIC



Colaboradores



Comités Organizadores

Organizadores

Manuel Ferre Pérez
Oscar Reinoso García
Eduardo Rocón de Lima

Comité Científico

Pedro Albertos Pérez
Manuel Berenguel Soria
Itziar Cabanes Axpe
Víctor Fernando Muñoz Martínez
Manuel Ferre Pérez
Fernando Gómez Bravo
José María Azorín Poveda
Raúl Marín Prades
Rafael Raya López
Óscar Reinoso García
Eduardo Rocon de Lima
Matilde Santos Peñas
Ángel Valera Fernández
Eduardo Zalama Casanova

Comité organizador local

Francisco Javier Badesa Clemente
Paloma de la Puente Yusty
Manuel Ferre Pérez
Ramón Antonio Suárez Fernández
Daniel Galán Vicente

Índice de las comunicaciones

Robótica

Ensamblaje automático de juguetes de plástico mediante un robot colaborativo. <i>D. Sánchez, C. A. Jara, F. Gomez</i>	1
Arquitectura Software para el Sistema Robótico de Manipulación Dual HortiRobot. <i>R. Fernández, E. Navas, M. Ojeda, M. Armada, P. González-de-Santos</i>	7
Metodología para el desarrollo de estrategias de manipulación remota. <i>V. Redondo, M. Ferre, M. Á. Sánchez-Urán</i>	13
Teleoperación de un robot colaborativo mediante realidad virtual. <i>M. García-Gómez, E. Zalama, J. Gómez-García-Bermejo, J. Duque-Domingo</i>	21
Análisis de estabilidad de un robot caminante en terrenos naturales. <i>P. Mollá-Santamaría, A. Peidró, M. Fabregat-Jaén, L. M. Jiménez, Ó. Reinoso</i>	27
Desarrollo de robots modulares altamente configurables con capacidades de auto-ensamblado. <i>V. Gomez, M. Hernando, E. Aguado, D. Bajo, R. Sanz, C. Rossi</i>	35
Diseñar lo inesperado: experiencias en la realización de experimentos de campo realistas en robótica para emergencias. <i>J. J. Fernández Lozano, A. Mandow, J. Miranda Páez, J. Trillo Legaz, J.M. Ravina Vergara, A.J. García Cerezo</i>	41
Impresión 3D multimaterial, alternativas de diseño de componentes en sistemas robóticos. <i>J. Minguella-Canela, N. Tegas, J. Amat, A. Casals</i>	49
Robótica cuántica. Elementos principales. <i>S. Altares-López, J. J. García-Ripoll, Angela Ribeiro</i>	59
Simulación de microactuador hidráulico para la navegación de catéteres. <i>A. J. Serrano-Balbontín, I. Tejado, B. M. Vinagre</i>	65
Detección de áreas transitables en entornos no estructurados mediante un LIDAR 2D. <i>A. Olivas, F. Torres</i>	73
Evaluación de métodos de planificación de rutas de cobertura en sistemas multi-UAV para aplicaciones de teledetección. <i>M. Luna, P. Campoy, M. Molina</i>	79
Planificador global SE(2) para la navegación de robots móviles manipuladores en ROS. <i>A. De Matías-Martínez, F. J. Naranjo-Campos, J. G. Victores, C. Balaguer</i>	85
Segmentación de planos a partir de nubes de puntos 3D en estructura reticulares. <i>F. J. Soler, A. Peidró, M. Fabregat-Jaén, L. Payá, Ó. Reinoso,</i>	91
Predicción de gestos no verbales para mejorar la interacción con un robot social. <i>E. Fernández Rodicio, Á. Castro-González, J. C. Castillo, M. A. Salichs, T. Onorati</i>	99

Protocolo de evaluación para la interacción entre el robot social JARI y niños en etapa escolar. <i>E. P. Madrid Ruiz, C. E. García Cena</i>	107
Sistema de monitorización no intrusiva para vivienda de personas mayores. <i>S. Merino-Fidalgo, E. Zalama, J. Gómez-García-Bermejo, J. Duque-Domingo, R. Gómez, P. Viñas, D. García, H. Urueña</i>	115
«Ilusión de control» una estrategia para favorecer la creación del vínculo entre los usuarios y el robot social Mini. <i>S. Carrasco-Martínez, M. Maroto-Gómez, F. Alonso-Martín, M. Malfaz, M. A. Salichs</i>	123
Diseño mecánico de las transmisiones del dispositivo robótico supernumerario SecondArms. <i>J. Álvarez-Pastor, D. Martínez-Pascual, L. D. Lledó, C. Fernández-Irles, M. Herraiz-Sala, N. García-Aracil</i>	131
Estudio de viabilidad de un nuevo concepto de actuador híbrido para robótica de rehabilitación acuática. <i>J. V. García-Pérez, E. Belmonte-Cerdán, J. Álvarez-Pastor, A. Blanco-Ivorra, J. M. Catalán, N. García-Aracil</i>	137
Actuadores blandos aplicados a robótica submarina. Una revisión. <i>G. Ferrando del Rincón, C. E. García Cena, E. Cortés Torres</i>	143
Desafíos en el desarrollo de nuevos robots submarinos bio-inspirados con morfología adaptable. <i>E. Cortés Torres, C. E. García Cena, G. Ferrando del Rincón</i>	149
Análisis de viabilidad de un robot velocista para la evaluación de la carrera deportiva. <i>A. Brunete, D. Guffanti, M. Hernando, D. Álvarez, E. Gambao, C. Ferreira</i>	155
Estrategia de caminata para el robot humanoide TEO. <i>G. Lipa, Jorge Muñoz, B. Łukawski, C. A. Monje, C. Balaguer</i>	163
Planificación de trayectorias de un robot bípedo trepador de estructuras reticulares. <i>M. Fabregat-Jaén, A. Peidró, P. Mollá-Santamaría, F. J. Soler, Ó. Reinoso</i>	171
Pruebas con simulación guiada para robots aéreos autónomos. <i>R. Pérez-Seguí, J. Melero-Deza, P. Arias-Perez, D. Perez-Saura, M. Fernandez-Cortizas, P. Campoy</i>	179
Análisis de alternativas de control para un exoesqueleto activo del codo. <i>D. Pont-Esteban, N. Barbosa, D. Cantalejo-Escobar, M. A. Sánchez-Urán, M. Ferre</i>	185
Robótica de Rehabilitación 4.0: Diseño y desarrollo del gemelo digital. <i>D. Sosa-Méndez, C. E. García-Cena, A. Martín-González, R. Saltaren-Pazmiño</i>	193
Sutura automatizada para una plataforma robótica de asistencia a la cirugía laparoscópica. <i>Á. Galán-Cuenca, D. De Luis-Moura, J. M. Herrera-López, M. Rollón, I. García-Morales, V. F. Muñoz</i>	201
Experiencias Educativas de Grado y Máster en Robótica y Automática Marina: El Robot de Superficie y la Competición MIR.	209

S. López, P. Marqués, J. Marín, C. del Olmo, S. Fornas, A. Solis, J. Echagüe, J.V. Martí, R. Marín, P.J. Sanz

Validación experimental del Robot Hexápodo Híbrido R3HC. 217
F. Gómez-Bravo, A. Garrocho, S. Rua, P. Villadóniga, G. Carbone, R. Jiménez-Naharr, R. López de Ahumada

Bioingeniería

Reconocimiento de actividades humanas aplicando modelos de Aprendizaje Profundo. 223
C. Sánchez-Girón, E. Zalama, J. Gómez-García-Bermejo, J. Duque-Domingo

Mejora en la clasificación de actividades mediante redes de convolución y CVV-SV. 231
J. Duque-Domingo, R. Gómez-Ramos, E. Zalama, J. Gómez-García-Bermejo

Biomarcador diagnóstico no invasivo para la enfermedad de Alzheimer y demencia frontotemporal basado en inteligencia artificial y movimiento ocular. 239
A. Calvo Córdoba, C. E. García Cena, C. Lage c, P. Sánchez Juan

Diseño de sensores de fuerza basados en Velostat para medida de la presión plantar. 245
J.C. Martínez-Sánchez, G. Delgado-Oleas, J. Lora-Millan, E. Rocon

Desarrollo de un exotraje basado en SMA para pacientes pediátricos. 251
P. Mansilla Navarro, D. Copaci, J. Arias Guadalupe, D. Blanco

Desarrollo de un prototipo de exo-traje para la rehabilitación de miembro superior. 259
C. A. Jara, S. Arias, B.D. Bolaños, A. Barrientos, A. Úbeda

Análisis de simulación del proceso de interacción física entre cuerpo humano y exoesqueleto basado en el software Opensim. 267
K.X. Zhang, F.J. Badesa, A. San Juan, E. Navarro, M. Ferre

Diseño conceptual de un robot de rehabilitación de la marcha pseudoestacionario. 273
J. Ramos, F. González, J.S. Lora, J.A. Castaño, S. Borromeo, R. Nieto, P. Fernández, J.M. Font, J.C. Moreno, A.J. del Ama

Medición del impacto de las vibraciones en usuarios de silla de ruedas 279
N. Perez, A. Mancisidor, I. Cabanes, P. Vermander

Impacto de las terapias de rehabilitación asistidas por robot en pacientes con alteraciones motoras graves. 285
Y. Vales, J.M. Catalán, J.V. García-Pérez, A. Blanco-Ivorra, R. Martín-Batanero, N. García-Aracil

Protocolo de adquisición de señales electroespino-gráficas en posición decúbito prono 291
D.I. Gracia, P. Soriano-Segura, L. Ferrero, M. Ortiz, E. Iañez, J.M. Azorín

Estrategia de estimulación transcraneal por corriente directa para tratamiento de la obesidad. 297
D.I. Gracia, M. Ortiz, E. Iañez, M. Herranz-Lopez, V. Micol, J.M. Azorín

La arquitectura tolerante a fallos del ventilador mecánico Andalucía Respira. 303
M. Rollon, V.F. Muñoz, I. García-Morales, M. Mengual-Mesa, V. Muñoz, A. Galán-Cuenca

Setup para la comparación de fuerzas en cirugía mínima invasiva robotizada. <i>J. Cases-Hurtado, J. Manrique-Córdoba, J.D. Romero-Ante, A. Busque-Nadal, J.M. Vicente-Samper, J.M. Sabater-Navarro</i>	309
Evaluación neuromecánica de la marcha humana en personas con síndrome de Charcot-Marie-Tooth: un estudio preliminar. <i>I. Beviá, J.M. López-Castellanos, C.A. Jara, A. Úbeda</i>	315



Jornadas de Robótica y Bioingeniería

Sesión: <P>

Segmentación de planos a partir de nubes de puntos 3D en estructuras reticulares

Soler, F. J.^{a,*}, Peidró, A.^a, Fabregat-Jaén, M.^a, Payá, L.^a, Reinoso, Ó.^{a,b}

^a Instituto de Investigación en Ingeniería de Elche (I3E), Universidad Miguel Hernández de Elche, Avda. de la Universidad s/n, Edificio Innova, 03202, Elche, Alicante, España.

^b ValgrAI: Valencian Graduate School and Research Network of Artificial Intelligence, Camí de Vera S/N, Edificio 3Q, 46022 Valencia, España.

Citar artículo: Soler, F. J., Peidró, A., Fabregat-Jaén, M., Payá, L., Reinoso, Ó. 2023. **Segmentación de planos a partir de nubes de puntos 3D en estructuras reticulares** Jornadas Nacionales de Robótica y Bioingeniería 2023, 91-98. <https://doi.org/10.20868/UPM.book.74896>

Resumen

La creación de mapas y modelado del entorno es una tarea primordial en la navegación de robots móviles. Obtener un modelo del entorno ligero y robusto es imperativo cuando los procesos se van a ejecutar en un robot con capacidad de cómputo y memoria limitada como es el caso de la mayoría de los robots trepadores. En este artículo se propone el uso de diferentes arquitecturas de redes neuronales para identificar de los datos capturados por un sensor LiDAR aquellos puntos que estén contenidos en planos pertenecientes a estructuras reticulares. Su propósito es eliminar información irrelevante como árboles o suelo a fin de mitigar los requisitos necesarios a posteriori de cómputo y memoria para tareas de mapeado o localización. Para el entrenamiento de dichas redes neuronales se ha utilizado un proceso de generación y etiquetado automático de datasets mediante entornos de simulación. Los experimentos realizados demuestran el correcto funcionamiento de las redes neuronales para la segmentación de elementos de la estructura contenidos en un plano.

Palabras clave: Redes neuronales, Robótica inteligente, Construcción de mapas, Robots móviles autónomos, Navegación programación y visión de robots, Nubes de Puntos, Detección.

Segmentation of planes from 3D point clouds in truss structures

Abstract

Map building and environment modelling is a main task in mobile robot navigation. Obtaining a lightweight and robust model of the environment is crucial when processes are going to be run in a robot with low computing power and memory, as in the case of most climbing robots. This article proposes the use of different neural network architectures to identify from the data captured with a LiDAR sensor those points contained in planes belonging to reticular structures. Our purpose is to remove irrelevant information such as trees or soil in order to reduce the computation and memory requirements for mapping or localization tasks. For training these neural networks, an automatic dataset generation and labelling process has been developed through simulated environments. The experiments evidence the capacity of neural networks to segment elements of the structure contained in a plane.

Keywords: Neural Network, Intelligent robotics, Map building, Autonomous mobile robots, Robot navigation, programming and visión, Point Cloud, Sensing.

*Autor para correspondencia: f.soler@umh.es

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

1. Introducción

Las estructuras reticulares son aquellas construcciones artificiales formadas por multitud de elementos geométricos simples conectados entre sí siguiendo patrones concretos. Debido a sus excelentes propiedades mecánicas como elementos estructurales son uno de los componentes principales en múltiples ámbitos, como puentes, torres de alta tensión, naves industriales, aeropuertos, estaciones de tren (Figura 1), etc. Suelen ser de carácter metálico y como tal, requieren de cierto mantenimiento e inspección para prevenir posibles daños como corrosión, grietas, problemas de soldadura, etc. Estas tareas se llevan a cabo por operarios, los cuales se exponen a grandes riesgos debido a los entornos en los que se suelen situar estas estructuras como pueden ser caídas desde gran altura o riesgo de electrocución.

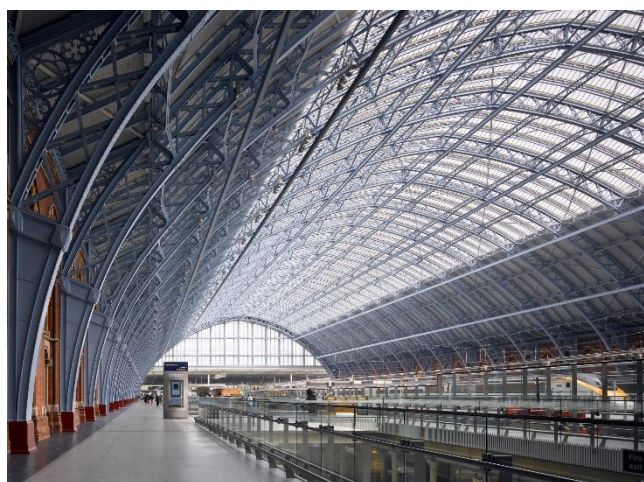


Figura 1: Estructura reticular de la estación de tren *St. Pancras* de Londres.

Con el objetivo de minimizar los daños que puedan sufrir los operarios, el campo de la robótica aparece como solución. Principalmente existen dos soluciones para abordar tareas de inspección y mantenimiento de estructuras reticulares mediante plataformas robóticas, drones o robots trepadores. Habitualmente, los drones tienen dificultades para navegar en entornos angostos, como puede ser el interior de estructuras reticulares. Adicionalmente los robots trepadores tienen una mayor capacidad de cargar peso, sobre todo cuando estos últimos tienen una arquitectura paralela. Por esto, último este trabajo se centra en el uso de robots trepadores para las tareas mencionadas.

Existen multitud de investigaciones sobre robots trepadores cuyo objetivo es llevar a cabo tareas de inspección y mantenimiento en las estructuras mencionadas (Saltaren *et al.*, 1999) (Guan *et al.*, 2011) (Nguyen & La, 2021). Es un campo de estudio abierto desde hace más de una década pero en el cual aún quedan muchos aspectos sin resolver. Con el avance de la tecnología, las posibilidades de este tipo de robots se han visto aumentadas enormemente, ya sea por una mayor autonomía, mejores sensores o mayor capacidad de cómputo y memoria. En este artículo se usará como referencia el robot bípedo HyReCRo (Peidró, 2018), cuyos 10 grados de libertad, arquitectura híbrida y sujeción magnética le confieren grandes

aptitudes para la navegación de estructuras reticulares metálicas.

La navegación autónoma de robots trepadores sobre estructuras reticulares es un campo de estudio en el que existen actualmente numerosos problemas sin resolver. Este tipo de entornos son especialmente demandantes por sus características geométricas repetitivas así como por su simetría y oclusiones encontradas en la información sensorial. Un correcto modelado del entorno es clave para alcanzar una navegación óptima y eficiente.

En este artículo consideraremos que se dispone de un sensor LIDAR que proporciona información 3D del entorno, a partir de la cual es preciso identificar y reconocer los diferentes elementos presentes en este. Existen estudios dentro del ámbito de las estructuras reticulares donde se utiliza toda la información proporcionada por el sensor de rango para construir el mapa y posteriormente identificar y reconocer sus elementos. Trabajos como (Cai *et al.*, 2019), (Chen *et al.*, 2018) o (Pham *et al.*, 2016) emplean este enfoque.

En (Cai, 2019) se emplea *Normal Distributions Transform* (NDT) como método para alinear las nubes de puntos en crudo proporcionadas por un sensor LiDAR. Partiendo de la nube de puntos que modela el entorno completo se identifican aquellos puntos pertenecientes a planos y se construye un nuevo mapa formado únicamente por los vértices de los planos previamente identificados. En (Chen *et al.*, 2018) se propone un algoritmo de segmentación de la estructura partiendo de un punto conocido y un mapa denso del entorno construido con la información proporcionada por una cámara RGB-D mediante el algoritmo de SLAM RTABmap (Labbe & Michaud, 2014). La salida de este algoritmo proporciona una nube de puntos que modela todo el entorno. A partir de dicha nube de puntos y conocido un punto de la estructura (punto de anclaje del robot) se aplica el algoritmo propuesto. Dicho algoritmo consiste en establecer de forma aleatoria una serie de puntos objetivo y determinar si estos son alcanzables o no desde el punto inicial. Para intentar alcanzar los puntos objetivo se basan en un método de Artificial Potential Field (APF) cuyo propósito es simular una fuerza de atracción desde el punto inicial a los puntos finales. Si el algoritmo es capaz de alcanzar un determinado punto, se considera que éste y todos los necesarios para llegar hasta él desde el inicio pertenecen a la estructura. Llegados a este punto la estructura queda segmentada.

El trabajo actual pretende identificar los elementos planares del entorno presentes en cada lectura individual del sensor. Para ello se propone emplear una red neuronal capaz de identificar los elementos presentes en el entorno sin crear un mapa previo. El objetivo es emplear una red capaz de segmentar de forma rápida y eficiente los puntos planares pertenecientes a una estructura reticular. Para ello se han puesto a prueba 3 arquitecturas de redes neuronales aplicando una segmentación binaria de nubes de puntos, donde la clase positiva será asignada a puntos planares contenidos en la estructura.

El artículo está organizado según la siguiente estructura. El segundo apartado se centra en comentar cada una de las arquitecturas empleadas. En una sección posterior se introduce el dataset utilizado y el método empleado para la generación del mismo. A continuación se presentan los experimentos

realizados así como sus resultados. El último apartado recoge las conclusiones apoyándose en los resultados del cuarto apartado.

2. Plataforma Robótica

El presente trabajo tiene como propósito ser implementado en un robot trepador. Más concretamente está pensado para ser puesto en funcionamiento a bordo del robot híbrido trepador HyReCRo (Peidró, 2018).

Se denominan robots híbridos, a aquellas plataformas robóticas formadas por múltiples mecanismos paralelos conectados en serie. Esta configuración les proporciona a los robots trepadores múltiples ventajas, como el amplio espacio de trabajo de los robots serie o la elevada capacidad de carga y rigidez de los robots paralelos.

El robot HyReCRo (Figura 2) es bípedo y posee una configuración híbrida serie-paralela. Dicha configuración se obtiene al conectar ambas patas (formadas por módulos paralelos) a un elemento común (cadera) mediante articulaciones rotativas (conexión serie). Está diseñado para navegar a través de estructuras reticulares adhiriéndose a estas mediante un sistema de imanes permanentes conmutables mecánicamente.

Durante el movimiento del robot una de las patas se mantiene fijada a la estructura mientras la otra alcanza la posición deseada. El robot dispone de un total de 10 grados de libertad (GDL), lo que le convierte en un robot redundante puesto que solamente son necesarios 6 GDL para alcanzar cualquier posición y orientación en el espacio.

A pesar de su configuración híbrida y 10 GDL, puede simplificarse como un robot serie de 8 GDL reduciendo la complejidad de los cálculos de su cinemática directa e inversa.

El robot equipará un sensor Ouster OS1 (LiDAR) o una RealSense d435i (RGB-D). Como punto de partida, sin estar decidido cuál de los dos sensores utilizará el robot finalmente, se ha optado por comenzar con el Ouster OS1 por sus mejores prestaciones. Entre sus características principales destacan su resolución, con un máximo de 2048x128 puntos, un rango de trabajo entre 0,5 y 170 metros, campo de visión vertical de 45° y horizontal de 360° y una máxima resolución en la medida del rango de 0,1cm.

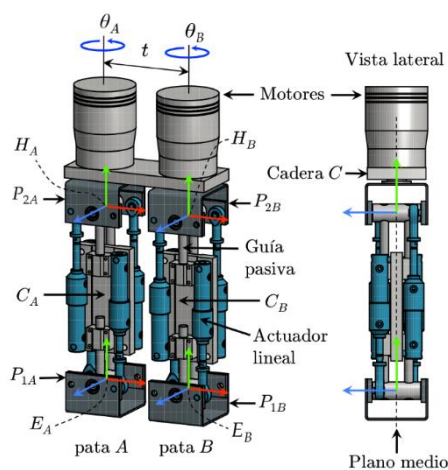


Figura 2: Representación CAD del robot HyReCRo.

3. Arquitecturas Empleadas

Las redes neuronales, parte de la inteligencia artificial que está inspirada en el funcionamiento del cerebro humano, es uno de los campos que más está avanzando en los últimos años. En el marco de la robótica y la visión por computador llevan usándose bastante tiempo para tareas tales como segmentación (Badrinarayanan *et al.*, 2017), detección de objetos (Redmon *et al.*, 2016) y localización a partir de imágenes (Cabrera *et al.*, 2021).

Dados sus buenos resultados y el auge de los sensores de rango 3D como LiDAR o cámaras RGB-D, el uso de las redes neuronales está siendo extendido a este tipo de datos de entrada.

Existen tres enfoques principales a la hora de aplicar redes neuronales a nubes de puntos.

El primero de ellos consiste en proyectar la nube de puntos 3D en múltiples vistas 2D y aplicar arquitecturas ya estudiadas para segmentación 2D (Su *et al.*, 2015). Este enfoque ofrece buenos resultados pero al proyectar la nube en múltiples vistas se pierde la información correspondiente a la distribución espacial de los puntos.

El segundo enfoque se basa en voxelizar la nube de puntos y obtener una estructura de malla en 3D sobre la que aplicar convoluciones 3D (Maturana & Scherer, 2015). Las nubes de puntos son un formato de datos disperso, lo que significa que no hay información en todo el espacio que ocupan, por lo que discretizar todo su espacio en voxeles significa que muchos de estos últimos no tendrán información. Aplicar convoluciones sobre este tipo de estructuras es costoso computacionalmente y poco eficiente ya que muchas de estas últimas se aplican sobre voxeles sin información.

El último planteamiento utiliza directamente la nube de puntos en crudo como el ejemplo de PointNet (Qi *et al.*, 2017a). Este enfoque no aplica ninguna transformación previa a los datos, por lo que estos mantienen su información espacial y los cálculos se realizan de forma dispersa, únicamente donde existan puntos.

La tendencia actual se basa en usar el último planteamiento mencionado como se indica en (Guo *et al.*, 2021). Por sus ventajas respecto al resto de enfoques en nuestro caso utilizaremos arquitecturas que empleen las nubes de puntos en crudo, sin transformaciones previas.

En los siguientes subapartados se comenta el funcionamiento y los aspectos principales de las arquitecturas de red empleadas en este trabajo.

3.1. PointNet

PointNet (Qi *et al.*, 2017a) fue la primera red neuronal en trabajar directamente con nubes de puntos sin transformaciones previas. Se trata de una arquitectura capaz de realizar tareas de segmentación semántica, segmentación de partes y clasificación de objetos. Toma como entrada directamente las nubes todas ellas con un número de puntos fijo. Mediante numerosos *Multilayer Perceptron (MLP)* procesa cada punto de forma independiente. En función de la tarea objetivo, segmentación o clasificación, las últimas capas de la red se modifican para obtener la salida deseada. Dado que se procesa cada punto de forma independiente la información

asociada a la vecindad de los puntos no es tenida en cuenta por esta arquitectura.

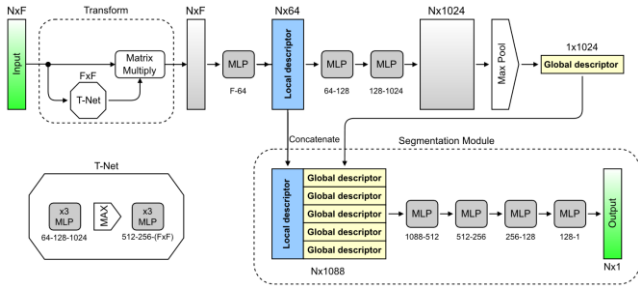


Figura 3: Diagrama de flujo de la arquitectura PointNet empleada en este trabajo.

El diagrama presentado en la Figura 3 muestra el flujo de información a través de la arquitectura PointNet. La entrada a esta red está formada por un tensor de tamaño $N \times F$, donde N es el número de puntos y F el número de características de cada uno de ellos. Dicho tensor pasa en primer lugar por un módulo denominado **T-Net**, el cual aplica una transformación a las características de entrada sin modificar sus dimensiones. El objetivo del módulo anterior es robustecer a la red ante transformaciones geométricas en la nube de entrada. Seguidamente se aplica un *Multilayer Perceptron* (MLP) aumentando el espacio de características y tomando estas como un descriptor a nivel local para cada punto. Se sigue usando MLP para aumentar el espacio de características y se aplica una capa de *max pooling* para obtener un único vector de características que describe la nube globalmente. Concatenado los descriptores locales de cada punto junto con el descriptor global, se aplican de nuevo un conjunto de MLP para reducir el espacio de características hasta el número de clases deseadas de salida. Tras cada MLP se aplica una normalización y una **ReLU** a excepción de la última capa, donde se aplica una Sigmoide (sin normalización) para obtener una salida entre 0 y 1 (no estructura, estructura).

3.2. PointNet++

PointNet++ (Qi et al., 2017b) es una extensión de PointNet. Esta extensión aborda el problema sobre la vecindad de los puntos y un tamaño de entrada variable aplicando un aprendizaje jerárquico a distintas resoluciones.

Su arquitectura se basa en aprender características mediante PointNet a múltiples resoluciones obteniendo así un descriptor más robusto y con información de vecindad (Figura 4).

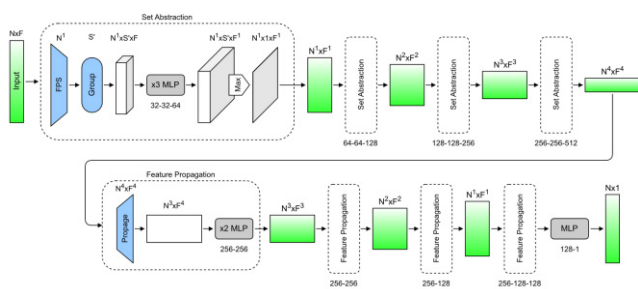


Figura 4: Diagrama de flujo de la arquitectura PointNet++.

Esto último se consigue aplicando de forma jerárquica los módulos llamados “*set abstraction*”.

Cada módulo de este tipo se encarga de muestrear la nube aplicando el algoritmo *Farthest Point Sampling* (FPS), agrupar puntos cercanos en un determinado radio y extraer características de cada agrupación de puntos con una estructura similar a PointNet. Posteriormente se propaga esta información hasta obtener un vector del mismo tamaño que la nube de puntos de entrada.

Sus resultados mejoran con respecto a la versión anterior y son más robustos frente a nubes de entrada con densidad variable. En contraposición su tiempo de entrenamiento e inferencia son considerablemente más elevados que PointNet al necesitar de muestreo previo y realizar un aprendizaje jerárquico a diferentes resoluciones.

3.3. MinkUNet34C

Esta arquitectura está propuesta como ejemplo de segmentación en el repositorio de GitHub de Minkowski Engine (Choy et al., 2019). Está formada por múltiples convoluciones 3D dispersas utilizando Minkowski Engine, que es una librería de código abierto que permite realizar convoluciones 3D dispersas. Este tipo de convoluciones se realizan únicamente en aquellas zonas donde existe información, hecho que reduce el tiempo de cómputo y la memoria necesaria considerablemente con respecto a las capas convolucionales 3D habituales.

Sus resultados muestran mejoras respecto a las arquitecturas anteriores en tareas de segmentación semántica en datasets reconocidos como S3DIS (Armeni et al., 2016).

Su funcionamiento se ejemplifica en la Figura 5. En primer lugar se discretiza la entrada mediante voxelización y se calcula la media de las características dentro de cada voxel. Partiendo de la entrada discretizada se construye un tensor disperso el cual se utilizará para realizar múltiples convoluciones dispersas. Las salidas de dichas convoluciones dispersas se concatenan con salidas anteriores y se aplican convoluciones traspuestas dispersas hasta llegar a la última capa, donde se aplica una última convolución dispersa seguida de una Sigmoide (sin normalización) para obtener la salida de la red entre 0 y 1.



Figura 5: Diagrama de flujo de la arquitectura MinkUNet34C.

4. Generación de datasets

Puesto que el ámbito de estudio de este artículo, estructuras reticulares y robots trepadores no está muy extendido, no se

dispone de datasets públicos con los que entrenar redes neuronales con objeto de determinar y reconocer este tipo de estructuras. Sumando a esto último el considerable coste que implica etiquetar nubes de puntos manualmente, resulta necesario utilizar algún método alternativo para entrenar modelos de red neuronal. Para ello se propone generar y etiquetar un dataset de forma automática en entornos simulados.

Los datasets empleados para este artículo han sido generados automáticamente en Gazebo Simulator. Esta plataforma de simulación integrada en Robot Operating System (ROS) permite simular multitud de sensores.

Para este apartado nos hemos basado en el trabajo de (Sanchez *et al.*, 2019), donde realizan un etiquetado automático de datos de un sensor LiDAR utilizando este simulador. Para etiquetar los datos se hace uso del campo “intensity” proporcionado por el sensor simulado. Este campo no es más que un valor que se le asigna a cada objeto en el entorno, y el sensor devuelve las coordenadas de dicho punto con su valor de “intensity” asociado. Gracias a esto se puede generar un dataset totalmente etiquetado en multitud de clases.

En nuestro estudio emplearemos el repositorio disponible en (Selby, 2020) para integrar el sensor Ouster OS1 en Gazebo y ROS. Para la configuración se ha establecido una resolución de 512x128 puntos, un rango mínimo de 0.05 metros, un rango máximo de 30 metros, unos campos de visión vertical y horizontal de 45° y 360° respectivamente y un ruido gaussiano de media cero y desviación estándar de 0.008 metros.

A diferencia de (Sanchez *et al.*, 2019), donde se selecciona manualmente la ubicación de cada lectura del sensor, en nuestro caso todo se realiza de forma automática gracias a un plugin desarrollado para este simulador, el cual permite generar un dataset de forma totalmente automática.

Dado que nuestro objetivo es entrenar una red neuronal lo más general posible, hemos decidido generar un dataset formado por paralelepípedos de dimensiones aproximadas a las que se estima que tengan las superficies de las estructuras reticulares objetivo. A cada uno de estos paralelepípedos se le asigna la etiqueta positiva, o 1, significando que pertenece a la clase estructura. Al resto de elementos del entorno se le asigna la etiqueta negativa o 0, implicando que no pertenecen a la estructura.

El entorno simulado es una pequeña modificación del mundo “cpr_inspection_gazebo” proporcionado en (Clearpath Robotics, 2023), sobre el cual se han introducido árboles y otros elementos para generar un entorno realista.

El proceso de generación consiste en introducir una cantidad definida de modelos, en este caso paralelepípedos, en posiciones, orientaciones y escalas aleatorias dentro de unos rangos delimitados. Simultáneamente se introducen un número aleatorio de elementos perturbando el entorno (árboles) en posiciones aleatorias y se mueve el modelo correspondiente al suelo para emular el movimiento del sensor alrededor del entorno. Por último, se le asigna una orientación aleatoria al sensor, el cual permanecerá anclado en la misma posición. Una vez hecho todo lo anterior, se captura la nube de puntos y se almacena.

Cada nube de puntos se obtiene de forma independiente y tiene su origen en la posición del sensor, el cual siempre se sitúa en entornos cercanos a los paralelepípedos o a la

estructura para proporcionar una situación lo más realista posible. En la Figura 6 se muestra una nube de puntos generada para el test, en la que se indica la posición y orientación del sensor mediante sus respectivos ejes de coordenadas.

El programa irá generando entornos hasta llegar a un número especificado como parámetro. Al final se obtendrá un dataset donde cada nube se almacena en un archivo independiente que contiene las coordenadas de cada punto junto a su etiqueta correspondiente, la cual se ha generado automáticamente.

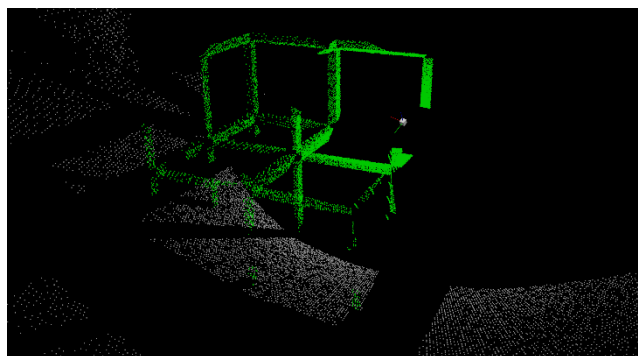


Figura 6: Representación de la posición y orientación del sensor en una nube de test.

4.1. Entrenamiento

Para el proceso de entrenamiento se ha generado un dataset de aproximadamente 10.000 nubes de puntos, con 25.000 puntos por nube y cuyo aspecto se puede ver en la Figura 7.



Figura 7: Renderización del entorno empleado para generar datos de entrenamiento. La circunferencia roja indica la posición del sensor.

4.2. Test

Para la parte de test se han generado 1.000 nubes de puntos totalmente diferentes a las utilizadas para el entrenamiento, en las cuales ya no se introducen paralelepípedos, sino que se utiliza el modelo de una estructura reticular (Figura 8).

El sensor se introduce en posiciones cercanas a la estructura, mayormente en su interior, simulando una situación lo más realista posible. Además, su orientación también se ve

modificada de forma aleatoria, simulando las múltiples orientaciones posibles alcanzables por el robot.



Figura 8: Renderización del entorno empleado para generar los datos de test. La circunferencia roja indica la posición del sensor.

5. Experimentos

Los experimentos se han realizado haciendo uso de la librería Pytorch 1.13.1 con CUDA 11.7 en una RTX 3090. Se han evaluado las arquitecturas anteriores utilizando los datasets de entrenamiento y test mencionados en el apartado anterior. Las métricas se han calculado con respecto a los resultados obtenidos en el dataset de test.

Para todos los experimentos se han utilizado coordenadas normalizadas y se ha empleado como función de pérdida la *Binary Cross Entropy* implementada en Pytorch como “BCELoss”. Como optimizador se ha empleado *Adam Optimizer* y se ha establecido un *Learning Rate* fijo de 0.001.

La decisión de si una predicción pertenece a una clase u otra se realiza mediante un umbral, calculado a partir de la curva de *Precision-Recall* y seleccionando aquel que maximiza la f1-score. Para el preprocesamiento, visualización y cálculo de normales se ha utilizado la librería *Point Cloud Library* más conocida como PCL.

Para cada una de las arquitecturas se han realizado diferentes experimentos variando las características de entrada. Las configuraciones más significativas y sus resultados se muestran en la Tabla 1.

5.1. PointNet

Esta arquitectura ofrece unos resultados óptimos cuando se le proporcionan como características de entrada las coordenadas y las tres componentes de la normal de cada punto. A pesar de ser unos muy buenos resultados, en torno al 95.86% de f1-score, si visualizamos la clasificación realizada por la red, Figura 9(b), se puede observar como aquellos puntos pertenecientes a un plano paralelo al suelo no son identificados como estructura.

5.2. PointNet++

De igual modo que para la arquitectura anterior, PointNet++ ofrece sus mejores resultados tomando como entrada las coordenadas y la normal de los puntos. Sus métricas son muy similares al caso anterior, encontrándose el mejor de los casos con una f1-score en torno al 95.66%, en cambio, cuando se

observan las nubes inferidas por la red se aprecian cambios significativos.

En la Figura 9(e) y Figura 9(h) se observa la inferencia de PointNet respecto a PointNet++ ante una nube de puntos en la que no se presenta ningún tipo de estructura. PointNet clasifica incorrectamente más puntos que PointNet++ en este caso. En contraposición, cuando se le presentan estructuras más complejas, como en la Figura 9(f) y Figura 9(i) PointNet obtiene mejores resultados que PointNet++.

5.3. MinkUNet34C

Con objetivo de mejorar la robustez de los resultados se ha empleado la arquitectura de MinkUNet34C la cual usa convolucionales 3D dispersas.

En este caso el mejor de los resultados se consigue igualmente para las coordenadas y las normales con una f1-score de 96.22%.

Puesto que las métricas pueden ser engañosas como se ha observado en los casos anteriores se ha comprobado el resultado ofrecido por esta arquitectura frente a las mismas nubes de entrada. Como se puede observar en la Figura 9(k) y Figura 9(l), esta arquitectura es capaz de identificar correctamente tanto estructuras más complejas como zonas sin estructuras.

6. Conclusiones

En este trabajo se presenta la aplicación de redes neuronales para segmentación de estructuras reticulares. Los resultados experimentales, muestran que las mejores características de entrada que mejores resultados ofrecen son las coordenadas de los puntos y sus normales.

En cuanto a las arquitecturas, PointNet y PointNet++ no parecen ser capaces de identificar elementos planares como estructura, sino que toman aquellos puntos cercanos dentro de un determinado radio como parte de la estructura. Un ejemplo de esto se observa en la Figura 9(e) donde PointNet identifica todos los árboles cercanos como estructura.

En cambio, cuando se utiliza la arquitectura MinkUNet34C, se consiguen mejores resultados y sí es capaz de identificar elementos planares. Por lo tanto, es más robusta para nuestra aplicación, en cuanto es capaz de identificar elementos planares de forma más robusta que las arquitecturas anteriores. Este hecho induce a pensar que las capas convolucionales 3D son mejores para la segmentación de estructuras.

En un futuro se planea utilizar la salida de estas redes neuronales para la construcción de mapas paramétricos de estructuras reticulares.

Agradecimientos

Este trabajo es parte del proyecto PID2020-116418RB-I00 financiado por MCIN/AEI/10.13039/501100011033.

La presente investigación también ha sido posible gracias al proyecto TED2021-130901B-I00, financiado por MCIN/AEI/10.13039/501100011033 y por la Unión Europea “NextGenerationEU”/PRTR

Tabla 1: Experimentos y resultados obtenidos para cada arquitectura. X Y Z representan las coordenadas, C corresponde a la curvatura y XN YN ZN son las componentes de la normal en cada uno de los ejes.

Arquitectura	Características	Tamaño de voxel	Precisión	Recall	F1-score	TP	FP	TN	FN	Tiempo medio inferencia (ms)
PointNet	X Y Z	-	0,8228	0,9254	0,8622	9455	2439	12188	918	123
	X Y Z C	-	0,8772	0,9146	0,8883	9342	1496	13131	1031	121
	X Y Z XN YN ZN	-	0,9683	0,9435	0,9537	9760	298	14329	613	122
PointNet++	X Y Z	-	0,9136	0,7166	0,7750	7416	745	13884	2955	330
	X Y Z C	-	0,9575	0,8473	0,8833	8791	357	14272	1580	330
	X Y Z XN YN ZN	-	0,9661	0,9378	0,9481	9732	404	14225	639	329
MinkUNet34C	X Y Z XN YN ZN	0,01	0,8288	0,7152	0,7515	7574	1852	12777	2797	63
	X Y Z XN YN ZN	0,05	0,9171	0,9897	0,9508	10269	952	13677	102	49
	X Y Z XN YN ZN	0,1	0,9425	0,9840	0,9622	10213	623	14006	158	45

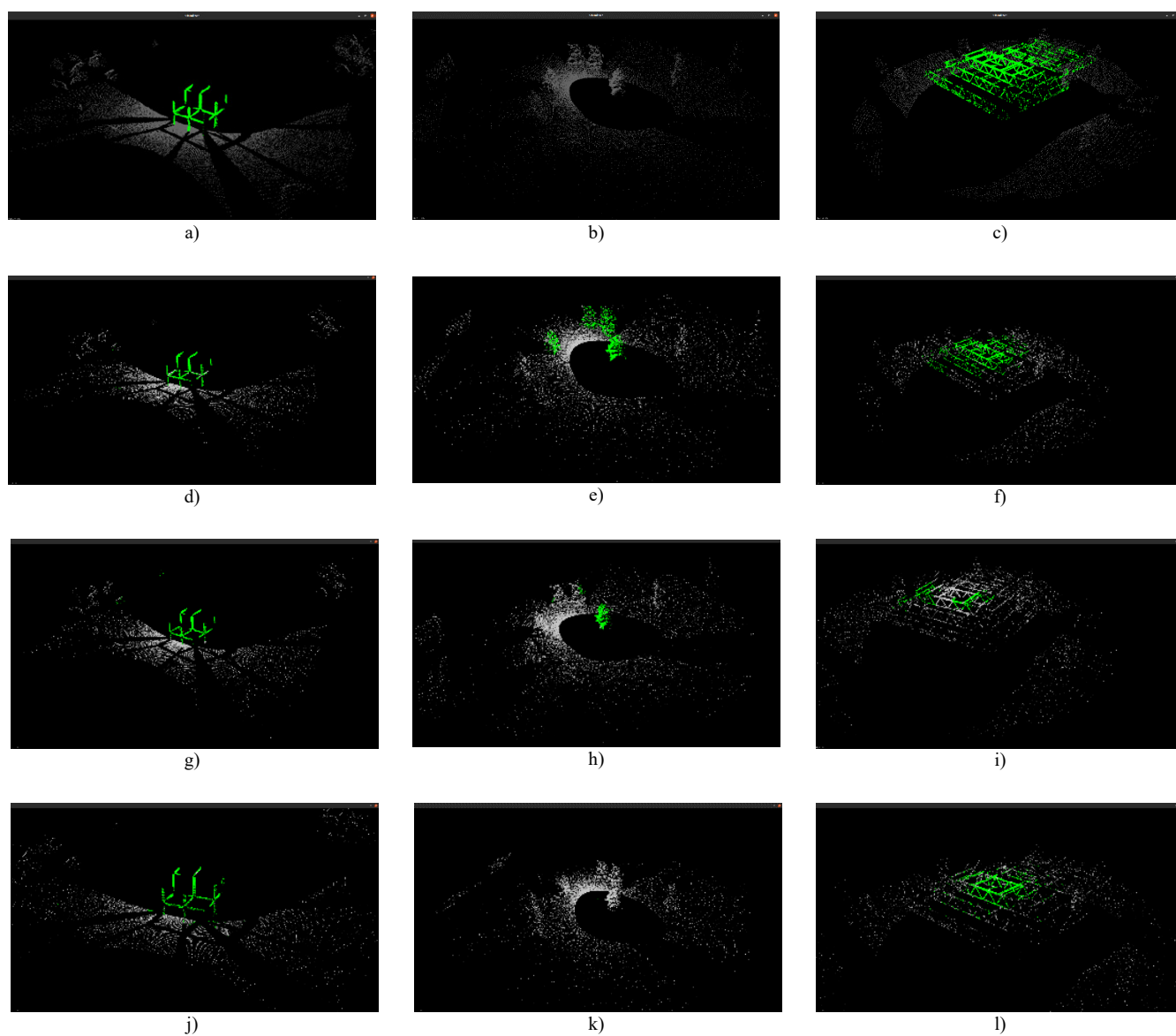


Figura 9: Ejemplos de inferencia en nubes comunes para las 3 arquitecturas. Siguiendo el orden de arriba a abajo, la primera fila muestra el ground truth de 3 nubes seleccionadas y el resto muestran los resultados para PointNet, PointNet++ y MinkUNet34C respectivamente.

Referencias

- Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. 2016. 3D semantic parsing of large-scale indoor spaces. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1534-1543.
DOI: 10.1109/CVPR.2016.170
- Badrinarayanan, V., Kendall, A., & Cipolla, R. 2017. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2481-2495.
DOI: 10.1109/TPAMI.2016.2644615
- Cabrera, J., Cebollada, S., Payá, L., Flores, M., & Reinoso, O. 2021. A robust CNN training approach to address hierarchical localization with omnidirectional images. Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics. Baker, R. C., 1963b.
- Cai, G., Lin, W., Wei, S., Gu, S., Zhu, H., & Guan, Y. 2019. Representation of 3D structure for path planning with biped wall-climbing robots. 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO).
DOI: 10.1109/ROBIO49542.2019.8961590
- Chen, W., Gu, S., Zhu, L., Zhang, H., Zhu, H., & Guan, Y. 2018. Representation of truss-style structures for autonomous climbing of biped pole-climbing robots. Robotics and Autonomous Systems, 101, 126-137.
DOI: 10.1016/j.robot.2018.01.002
- Choy, C., Gwak, J., & Savarese, S. 2019. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
DOI: 10.1109/CVPR.2019.00319
- Clearpath Robotics. 2023. Clearpath Additional Simulation Worlds. Obtenido de Github: https://github.com/clearpathrobotics/cpr_gazebo
Último Acceso: 06/04/2023
- Guan, Y., Jiang, L., Zhu, H., Zhou, X., Cai, C., Wu, W., Li, Z., Zhang, H., & Zhang, X. 2011. Climbot: A modular bio-inspired biped climbing robot. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems.
DOI: 10.1109/IROS.2011.6094406
- Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. 2021. Deep learning for 3D point clouds: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(12), 4338-4364.
DOI: 10.1109/TPAMI.2020.3005434
- Peidró, A. 2018. Kinematic Analysis and Design of the HyReCRo Robot: a Serial- Parallel and Redundant Structure-Climbing Robot.
- Labbe, M., & Michaud, F. 2014. Online global loop closure detection for large-scale multi-session graph-based SLAM. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems.
DOI: 10.1109/IROS.2014.6942926
- Maturana, D., & Scherer, S. 2015). VoxNet: A 3D Convolutional Neural Network for real-time object recognition. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
DOI: 10.1109/IROS.2015.7353481
- Nguyen, S. T., & La, H. M. 2021. A climbing robot for steel bridge inspection. Journal of Intelligent & Robotic Systems, 102(4).
DOI: 10.1007/s10846-020-01266-1
- Pham, N. H., La, H. M., Ha, Q. P., Dang, S. N., Vo, A. H., & Dinh, Q. H. 2016. Visual and 3D mapping for steel bridge inspection using a climbing robot. Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC).
DOI: 10.22260/ISARC2016/0018
- Qi, C. R., Su, H., Mo, K., & Guibas, L. J. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
DOI: 10.1109/CVPR.2017.16
- Qi, C. R., Yi, L., Su, H., & Guibas, L. J. 2017. PointNet++: Deep hierarchical feature learning on point sets in a metric space.
DOI: 10.48550/arXiv.1706.02413
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016. You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
DOI: 10.1109/CVPR.2016.91
- Saltaren, R., Aracil, R., Reinoso, O., Sabater, J. M., & Almonacid, M. 1999. Parallel Climbing Robots for Construction, Inspection and Maintenance. Proceedings of the 16th IAARC/IFAC/IEEE International Symposium on Automation and Robotics in Construction.
DOI: 10.22260/ISARC1999/0055
- Sanchez, M., Martinez, J. L., Morales, J., Robles, A., & Moran, M. 2019. Automatic generation of labeled 3D point clouds of natural environments with gazebo. 2019 IEEE International Conference on Mechatronics (ICM).
DOI: 10.1109/ICMECH.2019.8722866
- Selby, W. 2020. OS-1 Example Code. Obtenido de Github: https://github.com/wilselby/ouster_example
Último acceso: 06/04/2023
- Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. 2015. Multi-view convolutional neural networks for 3D shape recognition. 2015 IEEE International Conference on Computer Vision (ICCV).
DOI: 10.1109/ICCV.2015.114