Libro de Actas

Universidad de Zaragoza
Escuela de Ingeniería y Arquitectura
6, 7 y 8 de septiembre
Zaragoza

CEA Comité Español de Automática
XLIV Jornadas de AUTOMÁTICA 2023 – ZARAGOZA
Universidad Zaragoza 1542

XLIV JORNADAS DE AUTOMÁTICA : LIBRO DE ACTAS

UNIVERSIDAD DE ZARAGOZA
ESCUELA DE INGENIERÍA Y ARQUITECTURA
6, 7 Y 8 DE SEPTIEMBRE DE 2023
ZARAGOZA

EDITAN:
Servizo de Publicacións. Universidade da Coruña, A Coruña
Comité Español de Automática, Barcelona
Universidad de Zaragoza, Zaragoza
2023

UNIVERSIDADE DA CORUÑA

CEA Comité Español de Automática

Universidad Zaragoza 1542

ORGANIZAN:

Comité Español de Automática
Universidad de Zaragoza

EDITORES:

José Manuel Andújar Márquez

Ramón Costa Castelló

Luis Montano Gella

Alejandro Mosteo Chagoyen

Vanesa Loureiro Vázquez

Pedro Jesús Cabrera Santana

Elisabet Estévez Estévez

Raúl Marín Prades

Eduardo Rocón de Lima

David Muñoz de la Peña Sequedo

Luis Payá Castelló

Manuel Gil Ortega

Óscar Reinoso García

Carlos Vilas Fernández

DOI: https://doi.org/10.17979/spudc.9788497498609

ISBN: 978-84-9749-860-9

THEMA: TJFM, TJF

CDU: 681.05(063)

XLIV Jornadas de Automática 2023

https://jautomatica.es

Estimación de zonas transitables en nubes de puntos 3D con redes convolucionales dispersas

Santo, A.[a,b,*], Gil, A.[a], Valiente, D.[a], Ballesta, M.[a], Reinoso, O.[a,b]

[a] *Instituto de Investigación en Ingeniería de Elche (I3E), Universidad Miguel Hernández de Elche, Avda. de la Universidad s/n, 03202 Elche (Alicante), España.*
[b] *Valencian Graduate School and Research Network of Artificial Intelligence (valgrAI), Camí de Vera S/N, Edificio 3Q, 46022 Valencia, España*

To cite this article: Santo, A., Gil, A., Valiente, D., Ballesta, M., Reinoso, O. 2023. Estimación de zonas transitables en nubes de puntos 3D con redes convolucionales dispersas. XLIV Jornadas de Automática, 732-737 https://doi.org/10.17979/spudc.9788497498609.732

Resumen

La correcta evaluación del entorno en tareas como la navegación de robots terrestres o la conducción autónoma de vehículos se debe considerar un requisito mínimo para dotar de independencia a un sistema robótico. En concreto, la navegación de un robot en entornos desconocidos, naturales y desestructurados precisa contar con técnicas que permitan seleccionar sobre qué zonas puede circular el robot. Con el fin de poder aumentar la soberanía de las decisiones autónomas del sistema, en este artículo se propone un método para la evaluación de las nubes de puntos 3D obtenidas mediante un LiDAR con objeto de obtener las zonas transitables, tanto en entornos de carretera como en entornos naturales. Concretamente, se propone una configuración codificador-decodificador dispersa entrenada con características invariantes a rotación, que tiene como fin replicar los datos de entrada asociando a cada punto las características de transitabilidad aprendidas. Los resultados experimentales muestran la robustez y efectividad del método propuesto en entornos de exteriores, llegando a mejorar los resultados de otros enfoques.

Palabras clave: Robots móviles autónomos, Inteligencia Artificial, Redes Neuronales, Segmentación Semántica, Aprendizaje y adaptación en vehículos autónomos, Percepción y detección

Estimation of traversable zones in 3D point clouds with sparse convolutional networks.

Abstract

The correct assessment of the environment in tasks such as ground robot navigation or autonomous vehicle driving should be considered a minimum requirement to provide independence to a robotic system. In particular, the navigation of a robot in unknown, natural and unstructured environments requires techniques to select over which areas the robot can circulate. In order to increase the autonomy of the system's decisions, this paper proposes a method for the evaluation of 3D point clouds obtained by LiDAR in order to obtain traversable areas, both in road and natural environments. Specifically, a sparse encoder-decoder configuration trained with rotation invariant features is proposed, which aims to replicate the input data by associating to each point the learned traversability features. Experimental results show the robustness and effectiveness of the proposed method in outdoor environments, improving the results of other approaches.

Keywords: Autonomous Mobile Robots, Artificial Intelligence, Neural Networks, Learning and adaptation in autonomous vehicles, Perception and sensing

1. Introducción

La respuesta a la pregunta ¿Por dónde debería caminar?, formulada en (Wellhausen et al., 2019) contiene implícito el entendimiento de todo aquello que rodea al robot con objeto de poder navegar por el entorno. Este concepto que se presupone innato del ser humano, es deseable extrapolarlo a robots móviles autónomos ya que permite la planificación y navegación segura en diversas aplicaciones como, por ejemplo, la explora-

*Autor para correspondencia: a.santo@umh.es

ción de entornos desconocidos, la conducción autónoma, aplicaciones de búsqueda y rescate o agricultura (Sancho-Pradel and Gao, 2010).

Hasta la fecha, los algoritmos de planificación de trayectorias se clasifican atendiendo a dos conceptos fundamentales: a) la forma de definir y representar el espacio; b) la manera en la que se representan las zonas transitables en el mapa. Así, atendiendo al primer concepto, encontramos diferentes representaciones del espacio como pueden ser: mapas de ocupación 2D (Moravec and Elfes, 1985), mapas de ocupación basados en vóxeles 3D (Hornung et al., 2013; Oleynikova et al., 2017) o mapas de elevación (DEM) (Langer et al., 1994), en los cuales se establece con cierta probabilidad que un determinado espacio esté o no ocupado, y se considera si es transitable o no según parámetros físicos del robot, confluyendo así, en el segundo de los conceptos fundamentales mencionados.

Sin embargo, los enfoques clásicos mencionados anteriormente, no son lo suficientemente robustos cuando se generaliza a todo tipo de entornos (Xiao et al., 2022). Este hecho, junto con una sensorización de los equipos más sofisticada, si atendemos a características como el coste, resolución y ligereza, justifica abordar el cálculo de la transitabilidad bajo un nuevo paradigma: el aprendizaje automático supervisado, basado en redes neuronales. Concretamente en los últimos años se ha popularizado el empleo de sensores LiDAR para extraer los datos de partida de algoritmos basados en redes neuronales, debido a los motivos mencionados anteriormente sobre sensorización y por la inmutabilidad que presentan en diferentes condiciones de iluminación frente a otros sensores ópticos como las cámaras. Este artículo constituye una contribución a la estimación de la transitabilidad en terrenos complejos utilizando técnicas de aprendizaje profundo, en particular métodos de segmentación de una escena descrita por medio de nubes de puntos 3D.

Se ha organizado el resto del documento de la siguiente manera: en el apartado 2 se presenta un resumen de los trabajos más significativos en el ámbito del Aprendizaje Profundo para el cálculo de la transitabilidad. Seguidamente, se comentarán los conceptos en los que se basa nuestro método y su posterior explicación. Para finalizar se realizará un estudio de los resultados experimentales obtenidos teniendo en consideración los diferentes tipos de entornos que se han destinado al proceso de entrenamiento y las conclusiones que se pueden extraer de dicha experimentación.

2. Estado del arte

2.1. Métodos convencionales de ML

Se trata de algoritmos que generalmente parten de representaciones alternativas de los datos de entrada, es decir, actúan sobre características extraídas a partir de los datos y que se consideran discriminantes para el problema a resolver. Esta estrategia se emplea en (Bellone et al., 2017), donde se utilizan pares de imágenes estéreo como datos de entrada. Se realiza un estudio de las características geométricas y de apariencia más discriminantes para el problema de la transitabilidad en entornos urbanos a partir del entrenamiento de un clasificador SVM (*Support-Vector Machine*) (Vapnik, 1999), concluyendo que las características que incluyen el cálculo de los vectores normales son las más adecuadas para esta tarea. En (Kragh et al., 2015) se propone el cálculo de características basadas en un vecindario local para cada punto (obtenido mediante un sensor LiDAR 3D), con el fin de clasificarlos en: suelo, vegetación u objeto. En esta propuesta se propone un radio de vecindad adaptativo para paliar la pérdida de densidad de puntos inherente a los sensores LiDAR. De esta manera, se garantiza una alta resolución a corta distancia y evita las características ruidosas a larga distancia.

Uno de los mayores problemas que tienen los trabajos anteriores, es la necesidad de que un experto genere etiquetas de la clase a la que pertenece cada punto. Por ello existen trabajos que automatizan este proceso entrenando los clasificadores con datos simulados como (Martinez et al., 2020) que trata de describir nubes de puntos extraídas del simulador GAZEBO a partir del análisis de las direcciones principales (PCA) en un determinado entorno de vecindad.

2.2. Redes neuronales

Los sensores LiDAR producen, a su salida, la posición de un conjunto de puntos 3D. Estos puntos 3D corresponden con la primera reflexión producida por un objeto cuando se ilumina con un rayo láser colimado. La condición 3D de este tipo de dato, implica un aumento de memoria y computación en un factor cúbico. En relación con este hecho, se han desarrollado alternativas que permiten de forma eficiente trabajar con datos tridimensionales. Por ejemplo, en (Velas et al., 2018) se transforman nubes de puntos en imágenes multicanal que almacenan la profundidad, altura y reflectividad de cada punto. Estas imágenes se procesan a través de capas convoluciones densas para aprender qué zonas son transitables. Otra solución es la presentada en (Razani et al., 2021) donde se realizan proyecciones esféricas de las nubes de puntos para, posteriormente, aplicar capas convolucionales 2D y resolver un problema de segmentación semántica. Una solución diferente se presenta en (Wang et al., 2017), donde se hace uso de árboles octales u *octrees* para reducir la complejidad del espacio que describen las nubes de puntos. En esta propuesta se restringen las operaciones de convolución densa a aquellos octantes que están ocupados. Esta misma idea se extiende en (Frey et al., 2022) quienes calculan la transitabilidad del espacio a través de la generalización de la operación de convolución a n-dimensiones basándose en (Choy et al., 2019) y empleando una configuración de codificador-decodificador disperso.

3. Descripción del método

3.1. Formulación del problema

Se considera el problema de la evaluación de zonas transitables como una tarea de segmentación semántica de una nube de puntos $B = (P, F, L) = \{(\vec{p}_i, \vec{f}_i, l_i),\ i = 1, \ldots, N\}$, donde N es el número total de puntos de la nube, l_i la condición de transitabilidad, $\vec{p}_i \in \mathbb{R}^3$ y $\vec{f}_i \in \mathbb{R}^{d_{\text{in}}}$, siendo d_{in} la dimensionalidad de las características de entrada asociadas a cada punto de la nube. Se considera que cada punto originado por el sensor LiDAR tiene coordenadas $\vec{p}_i = (x, y, z)_i$, expresadas en el sistema de coordenadas del propio sensor LiDAR. El objetivo consiste en inferir una clasificación $l_{\text{i}} \in [0, 1]$ que representa la clase a la que pertece cada punto, en este caso, transitable (1) o no transitable (0). Definiendo así el problema como una clasificación binaria del entorno punto por punto.

3.2. Convolución dispersa

La operación de convolución discreta originariamente nace en el ámbito del procesamiento de la señal. Sin embargo, en los últimos años, se vincula directamente con el procesamiento de imágenes y el aprendizaje automático supervisado (Redes Neuronales), dando lugar a las Redes Neuronales convolucionales.

Si bien, este tipo de redes muestran una gran eficacia en problemas como clasificación y segmentación en imágenes. Su propia naturaleza secuencial e iterativa provocaría una gran ineficiencia computacional de la operación de convolución 2D en datos que no necesariamente sean vectores, matrices o características densas sino datos en los que exista una dispersión. Entendida la dispersión como el distanciamiento del conjunto de valores que conforman el dato, se aprecia en matrices dispersas o nubes de puntos 3D. Es por ello que el concepto de convolución en una imagen, es decir en 2D, se debe generalizar a cualquier número de dimensiones; así nace la convolución dispersa.

Este tipo de convolución discreta (conocida como *sparse convolution* en la literatura inglesa) permite centrar el kernel de convolución en aquellos espacios discretizados donde existe un valor distinto de cero, rompiendo así con el clásico desplazamiento de una máscara 2D en el que se basa la operación de convolución en imágenes. Especialmente en nubes de puntos resulta muy eficiente este enfoque ya que existen muchos lugares de la nube donde no existen puntos y por tanto la operación de convolución en esos puntos solo resultaría en un consumo de tiempo y recursos innecesario.

Por tanto, dada una nube de puntos cualquiera B, se construye un tensor disperso $\mathbf{S}$, formado, a su vez, por dos tensores $\mathbf{S}= (\mathbf{T}_C, \mathbf{T}_F)$:

- $\mathbf{T}_C$ define las coordenadas de los puntos que conforman la nube original, $\vec{p}_i$, a los cuales se les aplica una función de parte entera para discretizar el espacio. Estos puntos pueden ser modificados según un factor de escala, v, que determina cómo se discretiza el espacio. Además, se añade el lote b_k al que pertenece cada nube de puntos, para facilitar el entrenamiento de la red. Así pues, el tensor $\mathbf{T}_C$ se define como:

$$\mathbf{T}_C = \begin{bmatrix} b_1 & \vec{p}_1 \\ \vdots & \vdots \\ b_N & \vec{p}_M \end{bmatrix}, \text{con } \vec{p}_j = floor(\vec{p}_i) = floor\left(\frac{x_i}{v}, \frac{y_i}{v}, \frac{z_i}{v}\right) \tag{1}$$

- $\mathbf{T}_F$ almacena y promedia las características $\vec{f}_i$ asociadas a los puntos, m, que ocupan el mismo espacio, es decir que comparten las mismas coordenadas, $\vec{p}_j$, tras aplicar el factor de escala v y la función de parte entera.

$$\mathbf{T}_F = \begin{bmatrix} \vec{f}_1 \\ \vdots \\ \vec{f}_M \end{bmatrix}, \text{ donde } \vec{f}_j = \frac{1}{m}\sum_{i=1}^{m} \vec{f}_i \text{ para } \vec{f}_i \in \vec{p}_j \tag{2}$$

Este procesamiento de los datos de entrada se lleva a cabo empleando la librería *Minkowski Engine* [1] (Choy et al., 2019).

3.3. Red Neuronal Dispersa

Se propone el empleo de una red neuronal con una configuración codificador-decodificador, cuya implementación es una variedad dispersa de la red neuronal convolucional Resnet20 (He et al., 2016) y la arquitectura de la red U-net (Ronneberger et al., 2015). Por tanto, la red se divide principalmente en dos partes:

- **Codificador**: la parte codificadora de la red es la encargada de la generación de descriptores de los puntos basándose en la convolución dispersa 3D de las características pertenecientes a cada punto. En este caso, durante la investigación, se probaron diferentes combinaciones de características de entrada tales como, las propias coordenadas de cada punto $\vec{p}_i = (x, y, z)_i$, los vectores normales a cada punto $\vec{N}_i = (n_x, n_y, n_z)_i$ y la combinación de ambas. Sin embargo con el fin de conseguir una invarianza de los resultados al aplicar rotación a la nube de puntos, se ha considerado que el vector $\vec{f}_i = (n_z, Z)$ siendo n_z la componente unitaria de la dirección Z del vector normal a cada punto, N_i, y la coordenada $Z \in [0, 1]$.

- **Decodificador**: la parte decodificadora de la red trata de reconstruir y extrapolar la información latente generada por el codificador hacia las coordenadas de la nube de puntos de entrada. A tal efecto, se emplean capas de convolución traspuesta. Además, se utiliza un mapeado de coordenadas donde se registra la traza de cambios en las coordenadas debido a las convoluciones.

La configuración de la red empleada se presenta en la Figura 1. Se trata de una representación esquemática que trata de distinguir los diferentes niveles de la red neuronal de forma descendente, mediante convoluciones dispersas y bloques de convolución, que simbolizan convoluciones dispersas secuenciales. Además, una vez finaliza el codificador el esquema continúa de forma ascendente a través de la concatenación de los descriptores de distintos niveles, indicada mediante el símbolo $\oplus$, y de convoluciones traspuestas para recuperar la forma original de la nube de puntos de entrada.

4. Resultados experimentales

4.1. Datasets

1) SemanticKITTI: se trata de un conjunto de datos basado en *KITTI Vision Benchmark* (Geiger et al., 2012) que reúne posiciones de odometría y la representación del entorno, recogida por el modelo de sensor *Velodyne* HDL-64E, asociada a dichas posiciones de forma independiente. Contiene en total 22 secuencias urbanas, que describen entornos altamente estructurados, de las cuales 10 contienen etiquetas para cada punto orientadas a problemas de segmentación semántica.

2) Rellis-3D (Jiang et al., 2021): consiste en un conjunto de datos formado por 13.556 nubes de puntos divididas en 4 secuencias distintas capturadas por medio de un LiDAR OS1-64. A diferencia de SemanticKITTI se describen entornos y caminos rurales muy desestructurados.

[1] https://github.com/NVIDIA/MinkowskiEngine

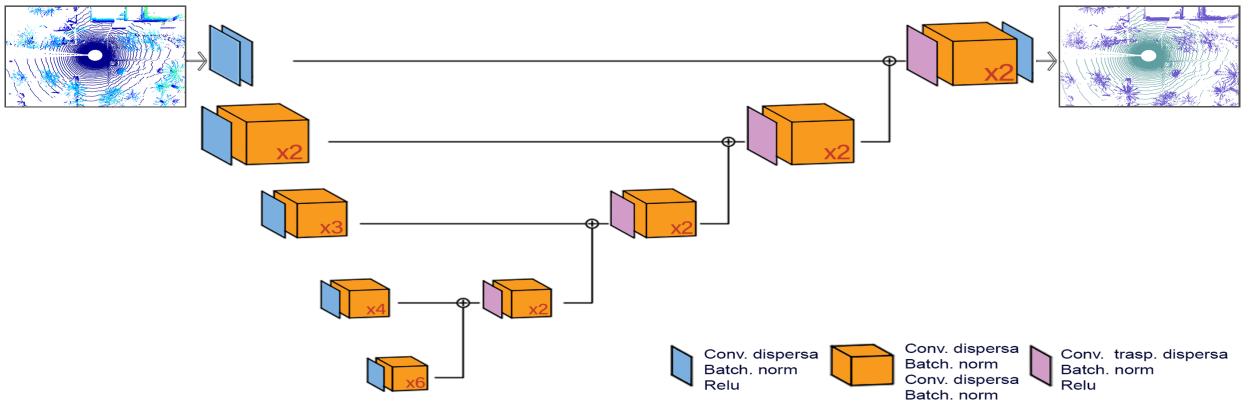


Figura 1: Configuración codificador-decodificador empleada, MinkUnet34 (Choy et al., 2019).

3) SemanticUSL (Jiang and Saripalli, 2021): la recogida de datos se da en una plataforma robótica, Clearpath Warthog, con un LiDAR OS1-64. Contiene 1200 nubes de puntos etiquetadas bajo el mismo formato que SemanticKITTI entre las cuales se incluyen escenas de carretera y ambientes naturales.

Todos los *datasets* mencionados, contienen alrededor de 25 etiquetas distintas a las que un punto puede pertenecer. Estas etiquetas incluyen conceptos semánticos como: arbusto, barro, asfalto, acera... etc. Atendiendo al problema de la transitabilidad tal y como lo hemos formulado anteriormente se han reetiquetado los *datasets* usando únicamente dos clases. Las Figuras 2(a) 2(c), 2(e), representan en diferentes colores el conjunto de clases originales de los *datasets*. Convirtiendo así etiquetas como acera, asfalto, vegetación baja o hierba, tierra, cemento y barro en la clase "transitable", representada en color verde. Las etiquetas que se han convertido a la clase "no transitable", representada en color violeta, son: árbol, persona, coche, camión, edificio, entre otras. Obteniendo así, unos datos de entrenamiento como los representados en las Figuras 2(b), 2(d), 2(f).

4.2. *Entrenamiento de la red*

De los conjuntos de datos mencionados anteriormente, únicamente se han utilizado para la etapa de entrenamiento algunas secuencias de *SemanticKITTI* y la totalidad de las secuencias de *Rellis-3D*. El motivo de esta elección es contar con un número de ejemplos de entrenamiento equilibrado que incluya, por igual, entornos urbanos, entornos desectructurados y naturales. De esta manera, se evita que la red se especialice en un tipo de entorno en concreto. El uso de la base de datos SemanticUSL se limita exclusivamente a procesos de *test* con objeto de demostrar el poder de generalización de la red en entornos nunca vistos durante el entrenamiento.

Además, se han llevado a cabo diferentes entrenamientos empleando diferentes parámetros de escala para discretizar el espacio, como se verá en los siguientes apartados.

4.3. *Efecto distancia*

Al estudiar cómo interactúan los planos del LiDAR con el entorno que lo rodea, se observa que la distancia a la que cortan rayos o planos consecutivos del LiDAR con el suelo aumenta de acuerdo con la inversa de la tangente del ángulo que forman el rayo del sensor con el plano de suelo, multiplicado por la altura a la que está situado el sensor. Así pues, a distancias muy alejadas, los diferentes planos láser se encuentran muy separados entre sí. Este efecto es fácil de apreciar en las vistas cenitales de un *scan* láser en la Figura 2. En consecuencia, la descripción de algunas regiones en el entorno del robot es muy inexacta, ya que la densidad de puntos LiDAR es muy baja. Este argumento se apoya en la Figura 3, que muestra la probabilidad de la existencia de puntos en base a la distancia según las observaciones de 2000 nubes de puntos. Se observa cómo, a partir de 45 metros, la probabilidad de encontrar puntos es casi nula.

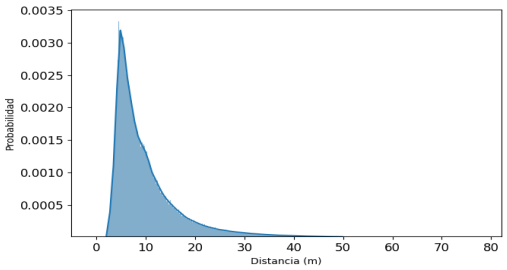


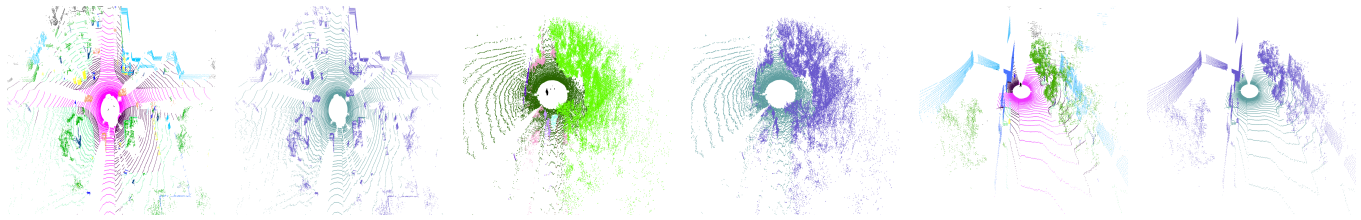
Figura 3: Función de densidad de probabilidad según la distancia.

Por tanto, como solución se propuso, considerar únicamente los puntos que se encuentran dentro de un radio de 45 metros del sensor y, por tanto, todas las evaluaciones y entrenamientos se han realizado bajo esta condición.

4.4. *Evaluación cuantitativa*

En la Figura 4 se presentan los resultados en términos de *precision* y *recall*. Para ello, se han realizado inferencias sobre todas las nubes de puntos que conforman el conjunto de datos de *test*. La clasificación de cada punto inferida por la red se compara con su *ground truth*, dando lugar a verdaderos positivos (1), verdaderos negativos (0), falsos positivos y falsos negativos.

Se puede observar como en las curvas pertenecientes a datos que describen entornos urbanos y más estructurados, Figuras 4(a) y 4(c), la relación entre *precision* y *recall* es cercana al máximo (esquina superior derecha). Por tanto podemos asumir que los modelos entrenados aprenden las zonas que son transitables y no transitables de forma muy consistente, logrando



(a) KITTI original (b) KITTI traducido (c) Rellis-3D original (d) Rellis-3D traducido (e) SemanticUSL original (f) SemanticUSL traducido

Figura 2: Traducción de las etiquetas originales a dos etiquetas de transitabilidad. Color verde: Transitable. Color violeta: No transitable.

valores de *precision* y *recall* superiores al 95 % para determinados puntos de trabajo. Además en estas curvas, se aprecia como el rendimiento es indiferente al parámetro de discretización, tamaño de *voxel*, o parámetro de escala, representado como v en la Ecuación (1), ya que se consiguen prácticamente los mismos valores.

Por otro lado, se observa en la Figura 4(b) que entornos desorganizados por la propia naturaleza, exentos de un control humano, son más difíciles de inferir correctamente qué zonas son o no adecuadas para atravesar. Igualmente, en este caso sí que es determinante la discretización del espacio de las nubes de puntos ya que se observan diferencias en el rendimiento dependiendo de este factor. La mejora de unos vóxeles con respecto de otros no parece seguir un orden lógico, sino que para esos entornos en concreto, los valores intermedios del rango de vóxeles que se han probado son los que mejor relación *precisión-recall* presentan.

Además, se ha realizado una comparación, Tabla 1, con diferentes trabajos enfocados principalmente a segmentación semántica pero que han sido reentrenados para el estudio de la transitabilidad por (Fusaro et al., 2023) de la misma forma que se ha planteado en el apartado 4.1, con el conjunto de datos *SemanticKITTI*.

Método	**Accuracy**	**F1 score**	**mIoU**
(Redmon and Farhadi, 2018)	93.4	93.0	87.4
(Wu et al., 2018)	90.1	89.4	81.4
(Wu et al., 2019)	92.3	91.9	85.5
(Qi et al., 2017)	90.0	93.0	87.4
(Fusaro et al., 2023)	89.2	91.4	84.9
Método propuesto	**96.6**	**95.9**	**92.3**

Tabla 1: Comparación de resultados sobre secuencias de SemanticKITTI 0-10.

4.5. Evaluación cualitativa

En la Figura 5(b), se pueden observar los resultados de una forma visual, en la que se muestran en distintos colores los parámetros que conforman la matriz de confusión: verdaderos positivos (azul), verdaderos negativos (verde), falsos positivos (rojo) y falsos negativos (negro). La Figura 5(a) representa una nube de puntos perfectamente etiquetada y la Figura 5(b) representa la inferencia de la red neuronal con los errores en color negro y rojo y los aciertos en color verde y azul según se ha descrito anteriormente. Las zonas más conflictivas en las que la inferencia no es correcta son aquellas zonas donde las geometrías que forman los puntos son desordenadas (falsos positivos) y donde se da el comienzo de una geometría y el final de otra (bordes), ya que los vóxeles toman puntos de ambos elementos y la media de las características se distorsiona porque se convierte en una fusión de ambas geometrías.

En este punto, es importante considerar que los falsos positivos (clasificados por la red como transitables, pero que son realmente no transitables) son realmente peligrosos en una tarea de navegación del robot.

5. Conclusión

En este trabajo se ha presentado un método para la estimación de la transitabilidad en nubes de puntos que utiliza una red neuronal dispersa y una configuración de codificador-decodificador. Los resultados obtenidos demuestran una gran robustez de la solución tanto en entornos muy estructurados como en entornos naturales o desestructurados, atendiendo al análisis realizado en cuanto a la discretización del entorno. El estudio demuestra que la estimación de la transitabilidad es una tarea más complicada en entornos naturales muy desordenados y con disposiciones que no siguen ninguna norma de los elementos.

6. Reconocimientos

Este trabajo ha sido financiado por la Fundación ValgrAI, *Valencian Graduate School and Research Network of Artificial Intelligence* a través de una beca predoctoral. Además, esta publicación forma parte del proyecto TED2021-130901B-I00, financiado por MCIN/AEI/10.13039/501100011033 y por la Unión Europea "NextGenerationE"/PRTR" y del proyecto PROMETEO/2021/075 financiado por la Generalitat Valenciana.

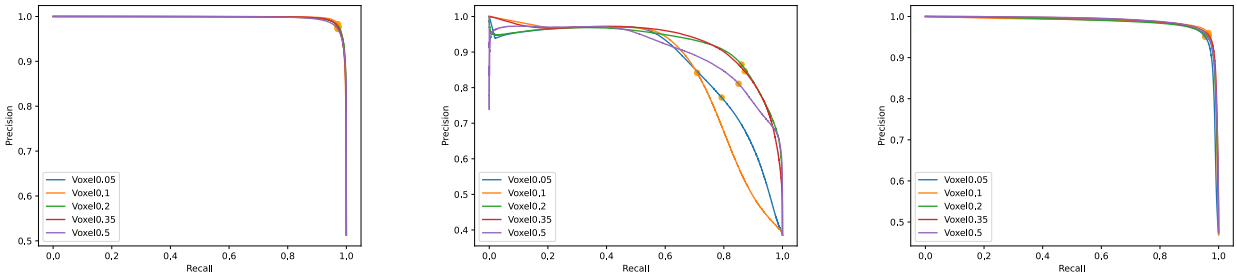
Referencias

Bellone, M., Reina, G., Caltagirone, L., Wahde, M., 2017. Learning traversability from point clouds in challenging scenarios. IEEE Transactions on Intelligent Transportation Systems 19 (1), 296–305.

Choy, C., Gwak, J., Savarese, S., 2019. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3075–3084.

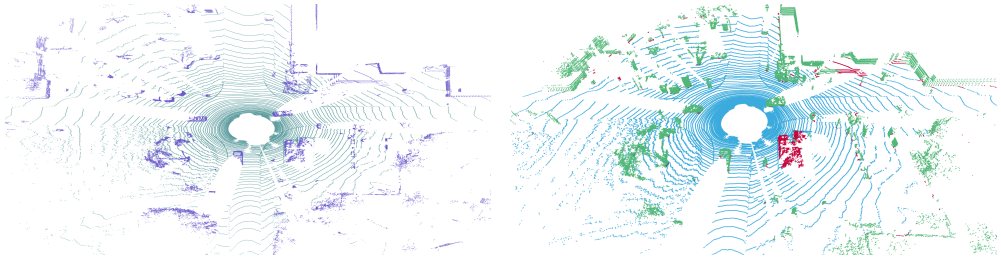
Frey, J., Hoeller, D., Khattak, S., Hutter, M., 2022. Locomotion policy guided traversability learning using volumetric representations of complex environments. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 5722–5729.

Fusaro, D., Olivastri, E., Evangelista, D., Imperoli, M., Menegatti, E., Pretto, A., 2023. Pushing the limits of learning-based traversability analysis for autonomous driving on cpu. In: Intelligent Autonomous Systems 17: Proceedings of the 17th International Conference IAS-17. Springer, pp. 529–545.



(a) Curva P-R de SemanticKITTI (b) Curva P-R de Rellis-3D (c) Curva P-R de USLSemantic

Figura 4: Curvas Precisión-Recall de la inferencia de la red sobre datos de test.



(a) Nube de puntos etiquetada (b) Inferencia de la red

Figura 5: Representación visual de los resultados de inferencia de la red. En color azul, verdaderos positivos (TP), en color verde los verdaderos negativos (TN), en color rojo los falsos positivos (FP) y por último en color negro los falsos negativos (FN).

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp. 3354–3361.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., Burgard, W., 2013. Octomap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous robots 34, 189–206.

Jiang, P., Osteen, P., Wigness, M., Saripalli, S., 2021. Rellis-3D dataset: Data, benchmarks and analysis. In: 2021 IEEE international conference on robotics and automation (ICRA). IEEE, pp. 1110–1116.

Jiang, P., Saripalli, S., 2021. Lidarnet: A boundary-aware domain adaptation model for point cloud semantic segmentation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 2457–2464.

Kragh, M., Jørgensen, R. N., Pedersen, H., 2015. Object detection and terrain classification in agricultural fields using 3D lidar data. In: Computer Vision Systems: 10th International Conference, ICVS 2015, Copenhagen, Denmark, July 6-9, 2015, Proceedings. Springer, pp. 188–197.

Langer, D., Rosenblatt, J., Hebert, M., 1994. A behavior-based system for off-road navigation. IEEE Transactions on Robotics and Automation 10 (6), 776–783.

Martinez, J. L., Moran, M., Morales, J., Robles, A., Sanchez, M., 2020. Supervised learning of natural-terrain traversability with synthetic 3D laser scans. Applied Sciences 10 (3), 1140.

Moravec, H., Elfes, A., 1985. High resolution maps from wide angle sonar. In: Proceedings. 1985 IEEE international conference on robotics and automation. Vol. 2. IEEE, pp. 116–121.

Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., Nieto, J., 2017. Voxblox: Incremental 3D euclidean signed distance fields for on-board mav planning. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 1366–1373.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017. Pointnet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660.

Razani, R., Cheng, R., Taghavi, E., Bingbing, L., 2021. Lite-hdseg: Lidar semantic segmentation using lite harmonic dense convolutions. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 9550–9556.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer, pp. 234–241.

Sancho-Pradel, D. L., Gao, Y., 2010. A survey on terrain assessment techniques for autonomous operation of planetary robots. JBIS-Journal of the British Interplanetary Society 63 (5-6), 206–217.

Vapnik, V., 1999. The nature of statistical learning theory. Springer science & business media.

Velas, M., Spanel, M., Hradis, M., Herout, A., 2018. Cnn for very fast ground segmentation in velodyne lidar data. In: 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE, pp. 97–103.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., Tong, X., 2017. O-cnn: Octree-based convolutional neural networks for 3D shape analysis. ACM Transactions On Graphics (TOG) 36 (4), 1–11.

Wellhausen, L., Dosovitskiy, A., Ranftl, R., Walas, K., Cadena, C., Hutter, M., 2019. Where should I walk? predicting terrain properties from images via self-supervised learning. IEEE Robotics and Automation Letters 4 (2), 1509–1516.

Wu, B., Wan, A., Yue, X., Keutzer, K., 2018. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3D lidar point cloud. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE, pp. 1887–1893.

Wu, B., Zhou, X., Zhao, S., Yue, X., Keutzer, K., 2019. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In: 2019 International Conference on Robotics and Automation (ICRA). IEEE, pp. 4376–4382.

Xiao, X., Liu, B., Warnell, G., Stone, P., 2022. Motion planning and control for mobile robot navigation using machine learning: a survey. Autonomous Robots 46 (5), 569–597.