



**Centro de Investigación en Ciencias de Información Geoespacial,
A.C.**

CentroGeo

Centro Público de Investigación CONAHCYT

**“Evaluación de la eficacia de las técnicas de
proyección para la segmentación semántica de
nubes de puntos capturadas por LIDAR”.**

TESIS

Para obtener el grado de:

Maestro en Ciencias de la información Geoespacial

Presenta:

Miguel Ángel Hernández Valencia

Director de Tesis:

Dr. Hugo Carlos Martínez

Codirector de tesis:

Dr. Ángel Ramón Aranda Campos

Ciudad de México

Abril 2024

**Centro de Investigación en Ciencias de Información Geoespacial,
A.C.**

**“Evaluación de la eficacia de las técnicas de
proyección para la segmentación semántica de
nubes de puntos capturadas por LIDAR”.**

TESIS

Para obtener el grado de:

Maestro en Ciencias de la información Geoespacial

Presenta:

Miguel Ángel Hernández Valencia

Director de Tesis

Dr. Hugo Carlos Martínez

Sinodales

Dr. Hugo Carlos Martínez

Codirector de Tesis

**Dr. Ángel Ramón Aranda
Campos**

**Dra. Daniela
Alejandra Moctezuma**

**Dr. Jesús Antonio
Sosa Herrera**

Ciudad de México

Abril, 2024

Agradecimientos

A mis padres, Sergio y Zulmara, por su amor y apoyo incondicional.
A mis hermanos, Sergio y Adriana, por su compañerismo y respaldo.
A mis asesores, Dr. Hugo Carlos y Dr. Ángel Ramón, por su orientación y paciencia.
A los amigos que hice en el camino, por su amistad y motivación.

Resumen

Esta investigación evalúa la efectividad de técnicas de proyección ortogonal, multivista y esférica en la segmentación semántica de nubes de puntos capturadas por LIDAR. Esta investigación puede resultar de utilidad para aplicaciones como el catastro, donde la precisión en la segmentación de nubes de puntos es fundamental. Utilizando U-Net, una arquitectura de aprendizaje profundo diseñada originalmente para la segmentación de imágenes biomédicas, el estudio procesó proyecciones 2D de nubes de puntos para generar imágenes segmentadas. Después de la segmentación, se empleó el algoritmo de "Label Spreading" para asignar estas etiquetas a la nube de puntos 3D. La proyección ortogonal resultó ser la más competente, logrando una precisión global del 92% y un IoU medio del 49,69%. La proyección multivista, que captura superficies ocluidas, registró métricas ligeramente más bajas, mientras que la proyección esférica quedó rezagada con un 82% de precisión debido a la deformación inherente de los datos. Los hallazgos subrayan el papel fundamental de las técnicas de proyección y U-Net en la segmentación de nubes de puntos, lo que sugiere que la proyección ortogonal es el enfoque óptimo a comparación de las otras proyecciones para la mayoría de las aplicaciones.

Summary

This research evaluates the effectiveness of orthogonal, multi-view, and spherical projection techniques in the semantic segmentation of LIDAR-captured point clouds. Using U-Net, a deep learning architecture originally designed for biomedical image segmentation, the study processed 2D projections of point clouds to generate segmented images. Post-segmentation, label spreading was employed to map these labels back to the 3D point cloud. The orthogonal projection emerged as the most proficient, achieving an overall accuracy of 92% and a mean IoU of 49.69%. Multi-view projection, capturing occluded surfaces, registered slightly lower metrics, while spherical projection lagged with 82% accuracy due to inherent data warping. The findings underscore the pivotal role of projection techniques and U-Net in point cloud segmentation, suggesting orthogonal projection as the optimal approach for the most applications.

Índice general

| | |
|--|-----------|
| 1. Introducción | 1 |
| Introducción | 1 |
| 1.1. Problemática | 2 |
| 1.2. Objetivos de la investigación | 6 |
| 1.3. Limitaciones y Supuestos | 7 |
| 1.3.1. Disponibilidad y Calidad de los Datos | 7 |
| 1.3.2. Requerimientos de Datos Etiquetados | 7 |
| 1.3.3. Factores Clave en la Selección del Modelo | 8 |
| 1.3.4. Compensaciones y Consecuencias | 8 |
| 1.3.5. Evaluación de Algoritmos | 9 |
| 1.3.6. Recursos Computacionales | 9 |
| 1.3.7. Veracidad de los Datos | 10 |
| 1.3.8. Avances en el Campo | 10 |
| 2. Antecedentes | 12 |
| 2.1. Nubes de Puntos | 12 |
| 2.1.1. LiDAR | 14 |
| 2.1.2. Fotogrametría | 16 |
| 2.2. Segmentación Semántica | 19 |

| | | |
|-----------|--|-----------|
| 2.2.1. | Métodos basados en proyecciones | 22 |
| 2.2.2. | Métodos basados en discretización | 25 |
| 2.2.2.1. | Representación discreta densa | 26 |
| 2.2.2.2. | Representación discreta escasa | 27 |
| 2.2.3. | Métodos basados en puntos | 28 |
| 2.2.3.1. | Métodos de perceptrón multicapa (MLP) por puntos | 29 |
| 2.2.3.2. | Métodos de convolución por puntos | 30 |
| 2.2.3.3. | Métodos basados en redes neuronales recurrentes (RNN) | 31 |
| 2.2.3.4. | Métodos basados en grafos | 32 |
| 2.3. | Estado del Arte de Segmentación Semántica de Nube de Puntos Aéreos | 33 |
| 2.3.1. | KPConv (Kernel Point Convolution) | 34 |
| 2.3.1.1. | Ventajas de KPConv | 37 |
| 2.3.1.2. | Limitaciones de KPConv | 38 |
| 2.3.2. | Fusión de Datos Fotogramétricos y Lidar a través de U-Net | 40 |
| 3. | Metodología | 44 |
| 3.1. | Introducción a la Metodología | 44 |
| 3.2. | Preprocesamiento de Nubes de Puntos | 46 |
| 3.2.1. | Normalización de altura | 48 |
| 3.2.1.1. | Fundamentos de la simulación de tela | 48 |
| 3.2.1.2. | Modificación de la simulación de tela para el filtrado de nubes de puntos | 49 |
| 3.2.1.3. | Implementación de CSF | 49 |
| 3.2.1.4. | Proceso de normalización utilizando CSF | 51 |
| 3.2.1.5. | Proceso de Normalización | 52 |

| | | |
|------------|--|----|
| 3.2.2. | Alineación del conjunto de datos | 54 |
| 3.2.2.1. | Cálculo del centroide | 54 |
| 3.2.2.2. | Proceso de alineación de los puntos | 55 |
| 3.2.3. | Proyección de nubes de puntos | 56 |
| 3.2.3.1. | Proyección ortogonal | 57 |
| 3.2.3.1.1. | Cálculo del número de filas y columnas | 58 |
| 3.2.3.1.2. | Inicialización de la matriz | 59 |
| 3.2.3.1.3. | Mapeo de puntos a píxeles | 59 |
| 3.2.3.1.4. | Almacenamiento de datos del punto en píxeles | 59 |
| 3.2.3.2. | Proyección esférica | 60 |
| 3.2.3.2.1. | Proyección esférica y cálculo angular | 61 |
| 3.2.3.2.2. | Transformación y mapeo de datos | 62 |
| 3.2.3.2.3. | Corrección de orientación y perspectiva | 63 |
| 3.2.3.3. | Proyección multi-vista | 64 |
| 3.2.3.3.1. | Matriz de rotación a partir de los ángulos de Euler | 65 |
| 3.2.3.3.2. | Matriz de transformación | 67 |
| 3.2.3.3.3. | Proyección perspectiva | 67 |
| 3.2.3.3.4. | Proyección ortogonal | 67 |
| 3.2.3.3.5. | Generación de imágenes | 68 |
| 3.3. | Entrenamiento y Posprocesamiento | 69 |
| 3.3.1. | Introducción al entrenamiento y posprocesamiento | 69 |
| 3.3.2. | Arquitectura U-Net | 70 |
| 3.3.2.1. | Estructura de la arquitectura | 70 |
| 3.3.2.2. | Convolución y función de activación | 72 |

| | | |
|-----------|---|-----------|
| 3.3.2.3. | Upsampling y skip connections | 73 |
| 3.3.2.4. | Función de pérdida y optimización . . . | 73 |
| 3.3.2.5. | Ventajas y limitaciones | 73 |
| 3.3.3. | Optimizadores | 75 |
| 3.3.3.1. | Importancia de los optimizadores . . . | 75 |
| 3.3.3.2. | Optimizador Adam | 75 |
| 3.3.3.3. | Otros optimizadores | 77 |
| 3.3.4. | Técnicas de posprocesamiento | 78 |
| 3.3.5. | Mejorando la segmentación con Label Spreading | 78 |
| 3.3.5.1. | Funcionamiento de Label Spreading . . | 79 |
| 3.3.5.2. | Ventajas y limitaciones | 80 |
| 3.3.6. | Resumen y próximos pasos | 81 |
| 4. | Experimentos y Resultados | 83 |
| 4.1. | Conjunto de datos | 83 |
| 4.1.1. | Recolección de Datos para el Conjunto de Datos DALES | 84 |
| 4.1.2. | Preprocesamiento y Anotación | 86 |
| 4.1.3. | Métricas | 87 |
| 4.1.4. | Benchmarks del Conjunto de Datos Dales | 89 |
| 4.2. | Resultados | 91 |
| 4.2.0.1. | Proyección Ortogonal | 92 |
| 4.2.0.2. | Proyección Ortogonal Multivista | 93 |
| 4.2.0.3. | Proyección Esférica | 95 |
| 4.2.1. | Validación de los Modelos | 97 |
| 4.2.1.1. | Proyección Ortogonal | 97 |
| 4.2.1.2. | Proyección Ortogonal Multivista | 102 |
| 4.2.1.3. | Proyección Esférica | 106 |

| | |
|---|------------|
| 5. Conclusiones y trabajo futuro | 111 |
| 5.1. Trabajos Futuro | 113 |
| Referencias | 115 |

Índice de cuadros

| | |
|--|-----|
| 4.1. Descripción general de los métodos seleccionados en el conjunto de datos DALES. | 90 |
| 4.2. Resultados de métricas de rendimiento de la proyección ortogonal | 93 |
| 4.3. Resultados de métricas de rendimiento de la proyección ortogonal multivista | 95 |
| 4.4. Resultados de métricas de rendimiento de la proyección esférica | 96 |
| 4.5. Resultados de IoU por clase de la proyección ortogonal en Nube de Puntos | 99 |
| 4.6. Resultados de métricas de rendimiento de la proyección ortogonal en Nube de Puntos | 100 |
| 4.7. Resultados de IoU por clase de la proyección ortogonal en Nube de Puntos Desconocida Completa | 100 |
| 4.8. Resultados de métricas de rendimiento de la proyección ortogonal en Nube de Puntos Completa | 100 |
| 4.9. Resultados de IoU por clase de la proyección ortogonal multivista en Nube de Puntos Desconocida | 103 |
| 4.10. Resultados de métricas de rendimiento de la proyección ortogonal multivista en Nube de Puntos | 103 |

| | |
|---|-----|
| 4.11. Resultados de IoU por clase de la proyección ortogonal multivista en Nube de Puntos Desconocida Completa . . | 104 |
| 4.12. Resultados de métricas de rendimiento de la proyección ortogonal multivista en Nube de Puntos | 104 |
| 4.13. Resultados de IoU por clase de la proyección ortogonal en Nube de Puntos Desconocida | 107 |
| 4.14. Resultados de métricas de rendimiento de la proyección ortogonal en Nube de Puntos | 108 |
| 4.15. Resultados de IoU por clase de la proyección esférica en Nube de Puntos Desconocida | 108 |
| 4.16. Resultados de métricas de rendimiento de la proyección ortogonal en Nube de Puntos | 108 |

Índice de figuras

| | |
|--|----|
| 2.1. Ejemplo de Nube de Puntos con valores X, Y, Z y R, G, B. | 13 |
| 2.2. Proceso simplificado de captura mediante sensor LiDAR. | 15 |
| 2.3. Principios básicos de la geometría epipolar en Structure from Motion. | 18 |
| 2.4. Línea de tiempo de los trabajos relacionados con la segmentación semántica de nubes de puntos 3D. | 20 |
| 2.5. Funcionamiento de KPConv en una arquitectura. Imagen recuperada de (Thomas y cols., 2019). | 35 |
| 2.6. Funcionamiento de KPConv en puntos 2D. Imagen recuperada de (Thomas y cols., 2019). | 36 |
| 2.7. Nube de Puntos de Lidar, Fotogramétrica y Fusionados | 41 |
| 3.1. Diagrama de flujo de la metodología propuesta. Fuente: Elaboración propia. | 46 |
| 3.2. Pasos principales en el algoritmo simulación de tela. Imagen recuperada de (W. Zhang y cols., 2016). | 50 |
| 3.3. Descripción gráfica del algoritmo de simulación de tela. Imagen recuperada de (W. Zhang y cols., 2016). | 51 |
| 3.4. Comparación de una nube de puntos sin normalizar y normalizada sus valores de Z. Elaboración propia. | 52 |

| | |
|---|----|
| 3.5. Comparación de una nube de puntos sin normalizar y normalizada sus valores de Z | 54 |
| 3.6. Proyección ortogonal. Imagen recuperada de http://www.grad.hr/geomteh3d/Monge/06projekcije/projekcije_eng.html | 58 |
| 3.7. Proyección ortogonal de una nube de puntos. Imagen recuperada de (Jovančević y cols., 2017). | 60 |
| 3.8. Proyección esférica de una nube de puntos. Imagen recuperada de (Y. Wang, Shi, Yun, Tai, y Liu, s.f.). | 61 |
| 3.9. Proyección esférica de una nube de puntos con valores de clase. Elaboración propia. | 63 |
| 3.10. Proyección esférica de una nube de puntos con valores de altura normalizados. Elaboración propia. | 64 |
| 3.11. Proyección multi-vista ortogonal de una nube de puntos (altura normalizada). Elaboración propia. | 65 |
| 3.12. Rotación de un objeto según los ángulos de Euler. Imagen recuperada de (Jähne, 2005). | 66 |
| 3.13. Arquitectura de U-Net. Imagen adaptada de (Ronneberger, Fischer, y Brox, 2015a). | 71 |
| 4.1. Imagen representa una parte de Dayton, Ohio | 85 |
| 4.2. Algoritmo por distribución de rendimiento de IoU de clase | 89 |
| 4.3. Gráfica de la función de pérdida contra epochs de datos de entrenamiento y validación del entrenamiento de proyección ortogonal | 93 |
| 4.4. Gráfica de la función de pérdida contra epochs de datos de entrenamiento y validación del entrenamiento de proyección ortogonal multivista | 94 |

| | |
|---|-----|
| 4.5. Gráfica de la función de pérdida contra epochs de datos de entrenamiento y validación del entrenamiento de proyección esférica | 96 |
| 4.6. Proyección ortogonal con las etiquetas reales | 98 |
| 4.7. Proyección ortogonal con la clasificación de U-net | 99 |
| 4.8. Matriz de Confusión de Segmentación con modelo Ortogonal | 101 |
| 4.9. Escena de la nube de Puntos con Valores Predichos | 102 |
| 4.10. Proyección ortogonal multivista con la clasificación correcta | 103 |
| 4.11. Proyección ortogonal multivista con la clasificación de U-net | 103 |
| 4.12. Matriz de Confusión de Segmentación con modelo Multivista | 105 |
| 4.13. Escena de la nube de Puntos con Valores Predichos | 106 |
| 4.14. Proyección esférica de una nube de Puntos (Datos Reales) | 107 |
| 4.15. Proyección esférica de una nube de Puntos (Datos Predichos) | 107 |
| 4.16. Matriz de Confusión de Segmentación con modelo Ortogonal | 109 |
| 4.17. Escena de la nube de Puntos con Valores Predichos | 110 |

Capítulo 1

Introducción

Las nubes de puntos, obtenidas mediante tecnologías avanzadas como LiDAR, fotogrametría y cámaras de profundidad, ofrecen representaciones tridimensionales detalladas de nuestro entorno. Estas contienen una cantidad significativa de datos que revelan aspectos cruciales de los elementos presentes en el entorno. No obstante, para interpretar efectivamente estos datos, es indispensable segmentar la nube de puntos en componentes diferenciados. La segmentación implica clasificar secciones de la nube según atributos distintivos tales como color, textura y forma geométrica, lo que facilita la identificación y aislamiento de elementos específicos dentro de la escena. Este proceso enriquece nuestra comprensión del entorno, proporcionando una perspectiva integral de la escena y sus componentes, y es de gran utilidad para una amplia gama de aplicaciones.

1.1. Problemática

El avance acelerado en tecnologías de captura de datos tridimensionales, como LiDAR y fotogrametría, ha impulsado un aumento en la demanda de métodos robustos y eficaces para procesar, analizar y comprender estos datos. Las nubes de puntos resultantes, colecciones de puntos 3D sin una estructura predefinida, ofrecen una representación espacial detallada de objetos y entornos reales. Sin embargo, para aprovechar plenamente estas representaciones, es crucial asignar una categoría específica a cada punto, un proceso conocido como segmentación semántica. Esta técnica es esencial en aplicaciones como la navegación de vehículos autónomos, planificación urbana, monitoreo ambiental y robótica, ya que facilita la identificación de elementos y mejora la interpretación de la escena.

Específicamente, en el ámbito de los vehículos autónomos, la segmentación semántica ha permitido avances significativos en la detección de obstáculos y la identificación de rutas seguras, elevando la seguridad y fiabilidad de estos sistemas. Además, en los sectores de realidad virtual y aumentada, una segmentación precisa es clave para generar experiencias inmersivas y simulaciones realistas, habilitando una interacción natural con elementos virtuales.

Adicionalmente, las nubes de puntos capturadas por vehículos aéreos, como los UAVs, se han convertido en una fuente valiosa de datos geoespaciales, proporcionando información 3D de alta resolución útil para diversas aplicaciones. Estos datos pueden incluir detalles sobre la topografía, infraestructura, vegetación, y otros elementos tanto naturales como artificiales. Recogidos mediante técnicas de detección remota como LiDAR y fotogrametría desde plataformas como aviones, helicópteros o UAVs, estos conjuntos de datos son

fundamentales para una amplia gama de aplicaciones prácticas en diferentes industrias y campos de estudio.

Entre las aplicaciones más destacadas se encuentran:

1. **Planificación Urbana:** Mediante la identificación precisa de estructuras urbanas, como edificios, carreteras y espacios verdes, la segmentación semántica facilita la toma de decisiones en cuanto a uso de suelo, zonificación y desarrollo de infraestructura, apoyando así a planificadores y administradores urbanos.
2. **Monitoreo Ambiental:** Es esencial para investigadores y agencias ambientales, ya que permite el seguimiento y análisis de ecosistemas, cambios en la cobertura terrestre y gestión de recursos naturales, clasificando elementos como vegetación, cuerpos de agua y características topográficas.
3. **Gestión de Desastres:** La segmentación facilita la evaluación rápida de daños a la infraestructura y al medio ambiente causados por desastres naturales tales como terremotos, inundaciones y huracanes, contribuyendo a optimizar las respuestas de emergencia y esfuerzos de recuperación.
4. **Agricultura y Silvicultura:** Permite el monitoreo de la salud y variedad de los cultivos, así como la estimación de biomasa vegetal, mejorando las prácticas de manejo agrícola y promoviendo la sostenibilidad forestal.
5. **Arqueología y Conservación del Patrimonio Cultural:** La capacidad de segmentar y clasificar datos de nubes de puntos de sitios arqueológicos o monumentos históricos brinda a los

investigadores herramientas precisas para la documentación, preservación e interpretación de estos lugares.

6. **Mapeo y Cartografía en 3D:** La segmentación semántica es fundamental para la creación de mapas tridimensionales detallados, reconociendo y categorizando elementos del paisaje y características geográficas, lo que resulta invaluable para diversas aplicaciones geoespaciales.

Estas aplicaciones demuestran la amplitud de campos que se benefician de la segmentación semántica, destacando su importancia en la comprensión y gestión de nuestro entorno.

Sin embargo, la segmentación semántica de nubes de puntos enfrenta múltiples desafíos debido a la naturaleza inherente de estos datos. Los obstáculos más significativos incluyen:

1. **Irregularidad y Desestructuración de los Datos:** A diferencia de las imágenes o mallas de vóxeles, las nubes de puntos carecen de una estructura ordenada, lo que complica la aplicación directa de técnicas como las redes neuronales convolucionales (CNN) diseñadas para datos estructurados.
2. **Variabilidad en la Densidad:** Factores como el método de captura, limitaciones de los sensores y la distancia al objeto de interés pueden resultar en variaciones de densidad dentro de la misma nube de puntos, afectando negativamente la precisión de la segmentación.
3. **Ruido e Incompletitud:** Las limitaciones de los sensores, obstrucciones y variaciones ambientales pueden introducir ruido y generar datos incompletos, lo que representa un reto

significativo para la identificación precisa de elementos dentro de la nube de puntos.

4. **Variación de Tamaño y Escala:** La amplia gama de tamaños y escalas de los objetos representados en las nubes de puntos puede dificultar que los algoritmos de segmentación identifiquen y clasifiquen correctamente los distintos elementos.
5. **Desequilibrio de Clases:** En muchos conjuntos de datos reales, ciertas clases de objetos pueden estar subrepresentadas, lo que conduce a un desequilibrio de clases. Esto puede sesgar los algoritmos de segmentación hacia las clases más predominantes.
6. **Alto Costo Computacional:** El procesamiento y análisis de nubes de puntos a gran escala demandan recursos computacionales significativos, especialmente para técnicas basadas en aprendizaje profundo, lo que supone un desafío adicional.

Estos retos evidencian la complejidad inherente a la segmentación semántica de nubes de puntos, destacando la imperiosa necesidad de innovar y desarrollar metodologías que superen estas barreras. La segmentación semántica no solo es fundamental, sino un pilar esencial en la cadena de procesamiento y análisis de datos 3D, habilitando una comprensión más profunda y aplicaciones avanzadas en diversos ámbitos.

Aunque esta área ha experimentado avances significativos, aún persiste un amplio margen para el mejoramiento y la innovación de los algoritmos de segmentación semántica. Es crucial explorar nuevos horizontes en este dominio, desarrollando soluciones robustas capaces

de adaptarse a la variabilidad en los conjuntos de datos, así como a las fluctuaciones en ruido y densidad de puntos.

En el marco de esta tesis, se analizarán los desarrollos más recientes en el campo del aprendizaje profundo y algoritmos basados en la geometría aplicados a la segmentación semántica. Se evaluará su eficacia, explorando sus límites y el estado actual de la tecnología, con el fin de introducir una propuesta innovadora. Este esfuerzo busca contribuir significativamente a los procesos de clasificación en entornos urbanos, facilitando la realización de proyectos de urbanismo, catastro, y más allá.

Esta investigación pretende, por tanto, no solo abordar las brechas existentes sino también ampliar el conocimiento y las herramientas disponibles para enfrentar los desafíos de la segmentación semántica de nubes de puntos, marcando un paso adelante en la optimización de la interpretación de datos tridimensionales.

1.2. Objetivos de la investigación

El objetivo principal de esta tesis es:

Desarrollar y evaluar un método efectivo, eficiente y escalable para la segmentación de nube de puntos.

Por otra lado se plantean los objetivos particulares como:

1. Desarrollar y optimizar una metodología basada en aprendizaje automático para la segmentación de nube de puntos.
2. Comparar el rendimiento del modelo propuesto con otros métodos existentes en términos de precisión, velocidad y escalabilidad.

3. Realizar un análisis de sensibilidad para evaluar cómo varía el rendimiento del modelo propuesto ante diferentes parámetros y condiciones de los datos.

1.3. Limitaciones y Supuestos

La realización de esta investigación implica enfrentarse a varias limitaciones y supuestos que son cruciales para interpretar adecuadamente los hallazgos y entender el alcance de los resultados. A continuación se detallan las limitaciones y supuestos.

1.3.1. Disponibilidad y Calidad de los Datos

La calidad y disponibilidad de conjuntos de datos de nubes de puntos aéreos varían considerablemente, lo que puede influir en la aplicabilidad y generalización de los resultados. Para mitigar esta variabilidad, se utilizarán dos conjuntos de datos previamente validados en estudios relevantes, aunque es importante reconocer que esto no garantiza la aplicabilidad universal de los hallazgos a todas las variedades de datos de nubes de puntos, especialmente aquellos con características únicas o recolectados bajo diferentes circunstancias.

1.3.2. Requerimientos de Datos Etiquetados

Los modelos de segmentación semántica dependen significativamente de la disponibilidad de datos etiquetados para su entrenamiento. La precisión de estos modelos aumenta con el volumen de datos etiquetados disponibles, hasta cierto punto, después del cual los beneficios tienden a disminuir. La necesidad de una gran cantidad de datos etiquetados plantea un compromiso entre la precisión del modelo y el coste (tanto

económico como temporal) asociado a la recopilación y etiquetado de estos datos.

1.3.3. Factores Clave en la Selección del Modelo

Para avanzar en esta investigación, es esencial considerar varios factores que influyen en la selección del modelo adecuado, tomando en cuenta tanto la precisión como la viabilidad. Estos factores incluyen:

- El costo asociado con la adquisición y anotación de datos.
- La complejidad inherente al modelo.
- El fenómeno de retornos decrecientes en precisión con el aumento de los datos etiquetados.
- La capacidad del modelo para generalizar a datos nuevos e invisibles.

1.3.4. Compensaciones y Consecuencias

La selección del modelo implica compensaciones que pueden afectar directamente el rendimiento, la escalabilidad del modelo y su impacto en aplicaciones del mundo real. Entre estas se destacan:

- Limitaciones en la capacidad de rendimiento del modelo.
- Desafíos asociados con la escalabilidad de la solución propuesta.
- El impacto potencial de estas limitaciones en la aplicabilidad práctica de los hallazgos en entornos del mundo real.

Estos factores no solo guían el desarrollo y la evaluación de los modelos propuestos, sino que también contextualizan los resultados dentro de un espectro realista de aplicabilidad y eficacia.

Considerando la limitada disponibilidad de datos etiquetados en ciertas regiones de México, se priorizará el uso de un modelo que requiera menos datos etiquetados para su entrenamiento. No obstante, este modelo será evaluado y comparado con otros que han demostrado ser eficaces según el estado actual de la tecnología, garantizando así una perspectiva amplia y fundamentada en la selección de la metodología.

Además, es fundamental abordar ciertas limitaciones y suposiciones inherentes a nuestra investigación para asegurar la transparencia y facilitar la interpretación adecuada de los resultados. En el contexto de nuestra tesis, que se enfoca en la segmentación semántica de nubes de puntos aéreas, destacamos los siguientes puntos críticos:

1.3.5. Evaluación de Algoritmos

El rendimiento de los algoritmos de segmentación semántica está influenciado por diversos factores, incluyendo los criterios de evaluación, procedimientos de entrenamiento y técnicas de preparación de datos adoptadas. Es importante reconocer que los resultados presentados en este trabajo pueden ser específicos a las condiciones de evaluación utilizadas y podrían variar bajo diferentes configuraciones.

1.3.6. Recursos Computacionales

La exploración detallada de modelos de aprendizaje profundo avanzados o el análisis de conjuntos de datos extensos puede ser restringida por limitaciones en los recursos computacionales disponibles. El acceso a recursos más potentes o la aplicación de técnicas de optimización podrían, en teoría, mejorar significativamente el desempeño de los algoritmos de segmentación propuestos.

1.3.7. Veracidad de los Datos

Esta investigación asume la exactitud de las etiquetas proporcionadas para los conjuntos de datos de nubes de puntos utilizados. Cualquier inexactitud o error en el proceso de etiquetado podría influir en la fiabilidad de nuestras conclusiones y hallazgos.

1.3.8. Avances en el Campo

Aunque este estudio se centra en las técnicas más avanzadas de segmentación semántica disponibles hasta la fecha, el rápido progreso en este ámbito sugiere que nuevas y posiblemente más eficaces metodologías podrían no haber sido consideradas. Es crucial reconocer que el campo de la segmentación semántica está en constante evolución, y lo que hoy es avanzado, mañana podría ser superado.

La inclusión de estas consideraciones asegura una comprensión profunda de los desafíos y limitaciones que enfrentamos en nuestra investigación, poniendo en contexto los esfuerzos realizados y los resultados obtenidos.

A pesar de estas consideraciones, es imperativo reconocer el valor sustancial de la investigación en segmentación semántica de nubes de puntos aéreas. Esta contribuye significativamente al avance del análisis de datos geoespaciales, facilitando una toma de decisiones más informada y eficaz en una variedad de campos y aplicaciones. El reconocimiento y entendimiento de las limitaciones y supuestos descritos no solo asegura la integridad científica de este trabajo, sino que también orienta futuras investigaciones hacia el desarrollo de soluciones más robustas y versátiles.

En conclusión, mientras que los desafíos inherentes a la segmentación semántica de nubes de puntos no deben subestimarse, el potencial para

mejorar la comprensión y gestión de nuestro entorno mediante el análisis de datos tridimensionales es inmenso. Este trabajo considera las bases para futuros avances en este campo, alentando la continua innovación y exploración de nuevas metodologías que puedan superar las limitaciones actuales.

Capítulo 2

Antecedentes

2.1. Nubes de Puntos

Una nube de puntos 3D representa una colección de puntos en el espacio tridimensional que captura la forma de un objeto en tres dimensiones. Cada punto en esta colección se define por sus coordenadas en relación con tres ejes perpendiculares, proporcionando una representación precisa de la superficie del objeto. Dependiendo del método de captura utilizado, las nubes de puntos pueden enriquecerse con información adicional, como color (valores RGB) y normales de superficie, lo que aporta detalles adicionales sobre las características del objeto, como se ilustra en la Figura 2.1.

A diferencia de las imágenes 2D, que proyectan la realidad tridimensional en un plano, las nubes de puntos mantienen la dimensionalidad espacial y, por ende, ofrecen una representación más fiel y detallada de los objetos. Esta característica hace que las nubes de puntos sean especialmente valiosas en entornos de iluminación variable y en la oscuridad, donde las proyecciones 2D pueden perder

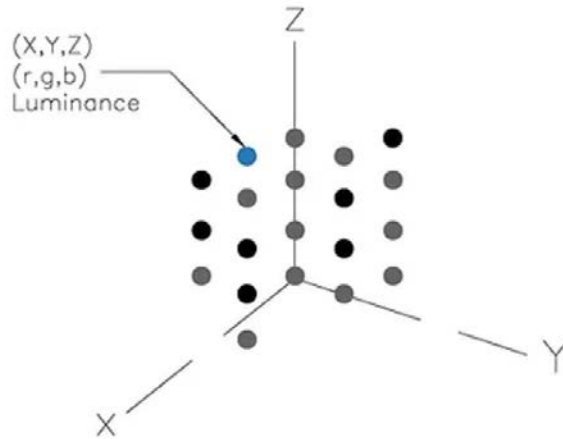


Figura 2.1. Ejemplo de Nube de Puntos con valores X, Y, Z y R, G, B.

Fuente: Imagen recuperada de <https://www.sixtyedge.ca/what-is-a-point-cloud>

información crucial. Sin embargo, el desafío radica en su naturaleza desordenada; a diferencia de las imágenes 2D, que se organizan en una cuadrícula regular, las nubes de puntos carecen de un orden predefinido, lo que exige enfoques de procesamiento innovadores. Estos métodos deben ser invariantes a la permutación de los puntos, garantizando que el procesamiento no se vea afectado por el orden de los datos.

En contraste con las imágenes 2D, donde la estructura organizada permite la aplicación directa de técnicas como la convolución —facilitando así el uso de redes neuronales convolucionales (CNN)—, la estructura desorganizada e irregular de las nubes de puntos

introduce complicaciones significativas para la realización de convoluciones directas. Adicionalmente, las nubes de puntos frecuentemente enfrentan desafíos debido a la dispersión de los datos, así como la presencia de valores atípicos y ruido.

Una alternativa es el uso de cuadrículas de vóxeles, que proporcionan una representación volumétrica estructurada mediante la discretización del espacio 3D en cubos fijos. Aunque esto permite la aplicación de convoluciones 3D, la conversión de datos de nubes de puntos a cuadrículas de vóxeles puede ser computacionalmente intensiva, especialmente para nubes de puntos de gran tamaño.

Las mallas, por otro lado, ofrecen una aproximación geométrica a las superficies 3D. Mientras que un punto aislado proporciona información limitada, un conjunto de puntos cercanos puede revelar detalles significativos como una muestra discreta de una superficie continua. Mediante la generación de mallas, que ajustan polígonos a los puntos 3D, es posible estimar estas superficies continuas (Liu, Zhang, Kadam, y Kuo, 2021).

En las secciones siguientes, exploraremos las tecnologías más comunes utilizadas para obtener nubes de puntos, destacando su importancia y las consideraciones prácticas para su procesamiento y análisis.

2.1.1. LiDAR

La generación de nubes de puntos 3D difiere significativamente del proceso de captura de imágenes 2D. Mientras que las imágenes 2D se producen al capturar la luz reflejada por los objetos a través de una lente, resultando en una representación plana, las nubes de puntos 3D generalmente se obtienen mediante el uso de tecnología LiDAR. A diferencia del RADAR y SONAR, que se basan en ondas de radio y

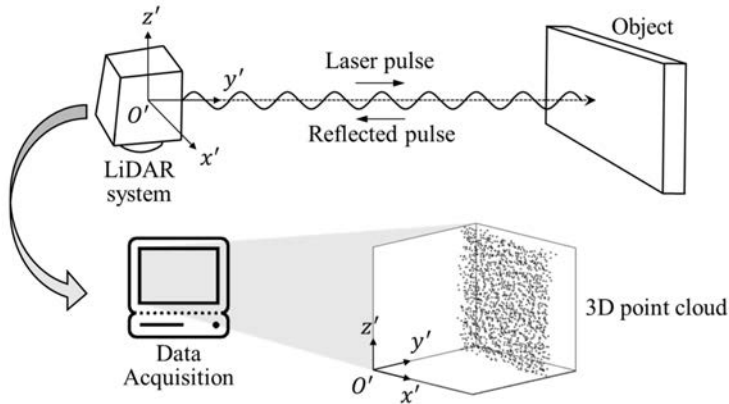


Figura 2.2. Proceso simplificado de captura mediante sensor LiDAR.

Fuente: Imagen adaptada de (Jo y cols., 2020)

sonido respectivamente, el LiDAR emplea pulsos de luz láser para mapear el entorno. Este sistema se compone de tres elementos principales: un emisor láser, un sistema rotatorio y un fotodetector.

El emisor dispara un haz de luz láser de alta intensidad en una frecuencia fuera del espectro visible. Este haz se proyecta en línea recta hasta impactar contra un objeto, momento en el cual se refleja de regreso hacia el sistema LiDAR. Gracias al mecanismo rotativo, el emisor puede dirigir el láser en múltiples direcciones, permitiendo así la captura de datos tridimensionales de los objetos alrededor. El fotodetector, ajustado para detectar la longitud de onda específica del láser utilizado, recoge la luz reflejada y analiza su retorno para determinar las características espaciales del entorno.

La distancia entre el sensor LiDAR y el objeto se calcula con precisión midiendo el tiempo que tarda el pulso láser en viajar desde el emisor hasta el objeto y regresar. La emisión simultánea de múltiples

pulsos láser permite generar una densa nube de puntos, capturando así la estructura tridimensional del entorno, como se ilustra en la Figura 2.2.

La tecnología LiDAR ofrece varias ventajas en comparación con otros métodos de captura 3D. Su capacidad para medir distancias con alta precisión y generar nubes de puntos densas la convierte en una herramienta valiosa para aplicaciones que requieren un mapeo detallado del entorno, como la topografía, la cartografía y la navegación autónoma. Además, el LiDAR es menos sensible a las condiciones de iluminación y puede operar tanto de día como de noche, lo que amplía su rango de aplicabilidad.

Sin embargo, el LiDAR también presenta algunos desafíos. Los sistemas LiDAR pueden ser costosos y requerir un procesamiento computacional significativo para manejar la gran cantidad de datos generados. Además, las superficies reflectantes o transparentes pueden interferir con las mediciones, y las condiciones atmosféricas, como la lluvia o la niebla, pueden afectar la calidad de los datos capturados.

A pesar de estos desafíos, la tecnología LiDAR sigue siendo ampliamente utilizada y se encuentra en constante evolución. Los avances en el hardware y el software han permitido el desarrollo de sistemas LiDAR más compactos, asequibles y eficientes, ampliando así su adopción en una variedad de industrias y aplicaciones.

2.1.2. Fotogrametría

La fotogrametría es una técnica que permite reconstruir la geometría tridimensional de objetos o escenas a partir de una serie de imágenes bidimensionales. Dentro de la fotogrametría, Structure from Motion (SfM) es un método específico que ha ganado popularidad en los últimos años debido a su capacidad para generar nubes de puntos

3D de alta calidad utilizando cámaras digitales convencionales. A diferencia de la fotogrametría tradicional, que se basa en la geometría y posiciones conocidas de las cámaras, SfM utiliza algoritmos de visión por computadora para estimar automáticamente las posiciones y orientaciones de las cámaras a partir de las imágenes, lo que lo hace más flexible y escalable.

Al igual que LiDAR, SfM puede crear modelos 3D basados en nubes de puntos y es capaz de producir modelos digitales de superficie y elevación de alta resolución utilizando cámaras digitales de consumo. Esta técnica se ha vuelto más accesible debido a los avances en computadoras, cámaras digitales y sistemas aéreos no tripulados (UAV) (Barramou, El Brirchi, Mansouri, y Dehbi, 2022).

SfM se basa en los principios de la fotogrametría estereoscópica, que utiliza la triangulación para determinar las posiciones tridimensionales relativas de los objetos a partir de pares estéreo. Sin embargo, la estereofotogrametría tradicional requiere un equipo y un software especializados y costosos. Con SfM, todo lo que se necesita es una gran cantidad de imágenes de un área o un objeto tomadas desde diferentes ángulos con una cámara estándar de consumo comercial (Barramou y cols., 2022). En la Figura 2.3, se observan las bases de la geometría epipolar para lograr el proceso de SfM. La idea general consiste en identificar características coincidentes en múltiples imágenes, rastrearlas de una imagen a otra y producir estimaciones de las posiciones y orientaciones de la cámara, así como las coordenadas de las características. Esto genera una nube de puntos de coordenadas X, Y, Z .

Generalmente se requiere un mínimo de tres imágenes para obtener características visibles, se recomienda tener tantas imágenes como sea posible para un procesamiento más rápido y una mayor calidad del

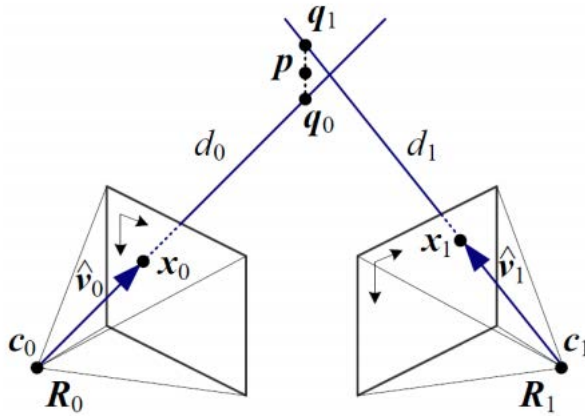


Figura 2.3. Principios básicos de la geometría epipolar en Structure from Motion.

Fuente: Imagen recuperada de
<https://m.blog.naver.com/dnjswns2280/222127017232>

modelo. SfM se ha utilizado para crear modelos tridimensionales de edificios históricos y otros monumentos utilizando imágenes de múltiples fuentes. Las nubes de puntos 3D a menudo se generan en un sistema de coordenadas relativo a las posiciones de la cámara, pero se pueden alinear con un sistema de coordenadas del mundo real utilizando puntos de control terrestre (GCP) e imágenes georreferenciadas (Barramou y cols., 2022).

La fotogrametría y SfM ofrecen varias ventajas en comparación con otras técnicas de captura 3D. Al utilizar cámaras convencionales, el costo de adquisición de datos es relativamente bajo en comparación con los sistemas LiDAR. Además, la capacidad de capturar información de color junto con la geometría proporciona un nivel adicional de detalle y realismo a las nubes de puntos generadas. Sin

embargo, la fotogrametría también presenta algunos desafíos. La calidad de la reconstrucción 3D depende en gran medida de la calidad y superposición de las imágenes capturadas. Además, las condiciones de iluminación variables y las superficies homogéneas o repetitivas pueden dificultar el proceso de correspondencia de características.

A pesar de estos desafíos, la fotogrametría y SfM se han establecido como técnicas valiosas para la generación de nubes de puntos 3D en una variedad de aplicaciones, desde la documentación del patrimonio cultural hasta la cartografía y la inspección de infraestructuras. Con los avances continuos en algoritmos y hardware, se espera que la fotogrametría siga desempeñando un papel importante en la captura y análisis de datos tridimensionales.

2.2. Segmentación Semántica

La segmentación semántica es un proceso fundamental en el análisis y comprensión de las nubes de puntos 3D. Su objetivo es asignar una etiqueta semántica a cada punto de la nube, lo que permite obtener una representación más estructurada y significativa de los elementos presentes en la escena. Al etiquetar cada punto con una categoría específica, como edificios, vegetación, terreno o carreteras, se facilita la extracción de información relevante y se mejora la interpretación de los datos 3D (Liu y cols., 2021).

En los últimos años, se han desarrollado diversos métodos de aprendizaje profundo para abordar el desafío de la segmentación semántica en nubes de puntos 3D. Estos métodos se pueden clasificar en tres categorías principales: métodos basados en proyecciones, métodos basados en discretización y métodos que operan directamente sobre la nube de puntos. La Figura 2.4 presenta una línea de tiempo que ilustra la evolución de estos enfoques desde 2017 hasta 2021.

Los métodos basados en proyecciones buscan simplificar el problema de la segmentación semántica proyectando las nubes de puntos 3D en representaciones 2D, como imágenes o mapas de profundidad. Esto permite aprovechar las arquitecturas de redes neuronales convolucionales (CNN) ampliamente utilizadas en el procesamiento de imágenes 2D. Aunque estos métodos pueden ser computacionalmente eficientes, pueden perder información geométrica importante durante el proceso de proyección (Guo y cols., 2021).

Por otro lado, los métodos basados en discretización convierten las nubes de puntos en representaciones discretas, como octrees o vóxeles, lo que permite aplicar operaciones de convolución 3D. Estos enfoques capturan mejor la geometría 3D de la escena, pero pueden ser computacionalmente costosos y sufrir de una pérdida de resolución debido a la discretización (Guo y cols., 2021).

Finalmente, los métodos que operan directamente sobre la nube de puntos han ganado popularidad en los últimos años. Estos enfoques utilizan arquitecturas de redes neuronales especialmente diseñadas para manejar datos no estructurados, como las redes neuronales basadas en puntos (PointNet) (Qi, Su, Mo, y Guibas, 2017) y sus extensiones. Estos métodos pueden capturar directamente las relaciones espaciales y las características geométricas de los puntos, lo que conduce a resultados de segmentación más precisos.

Además de las categorías mencionadas, también se han propuesto enfoques híbridos que combinan diferentes representaciones y técnicas para mejorar aún más el rendimiento de la segmentación semántica. Por ejemplo, algunos métodos utilizan tanto información 2D (imágenes) como 3D (nubes de puntos) para obtener una comprensión más completa de la escena (P. Zhang y cols., 2022).

La segmentación semántica de nubes de puntos 3D tiene numerosas aplicaciones en diversos campos, como la conducción autónoma, la robótica, la planificación urbana y la realidad virtual. Al proporcionar una comprensión de alto nivel de la escena, facilita tareas como la detección de objetos, la navegación y la interacción con el entorno.

A pesar de los avances significativos en los últimos años, la segmentación semántica de nubes de puntos 3D sigue siendo un área de investigación activa. Desafíos como la variabilidad en la densidad de puntos, la presencia de ruido y outliers, y la necesidad de modelos eficientes y escalables aún requieren atención. Además, la disponibilidad de conjuntos de datos etiquetados de alta calidad es crucial para el entrenamiento y la evaluación de los modelos de segmentación semántica.

En las siguientes secciones, se explorará con más detalle los diferentes enfoques mencionados, con un énfasis particular en los métodos más recientes y prometedores en el campo de la segmentación semántica de nubes de puntos 3D.

2.2.1. Métodos basados en proyecciones

Los métodos basados en proyecciones abordan el problema de la segmentación semántica de nubes de puntos 3D mediante la transformación de los datos 3D en representaciones 2D más familiares, como imágenes o mapas de profundidad. Esta estrategia permite

aprovechar las arquitecturas de redes neuronales convolucionales (CNN) ampliamente utilizadas y optimizadas para el procesamiento de imágenes 2D. Al simplificar la representación de los datos, estos métodos buscan reducir la complejidad computacional y facilitar la aplicación de técnicas de aprendizaje profundo bien establecidas.

Dentro de los métodos basados en proyecciones, existen dos categorías principales: los métodos de vista múltiple y los métodos de representación esférica (Guo y cols., 2021). En los métodos de vista múltiple, las nubes de puntos 3D se proyectan en planos 2D desde diferentes vistas virtuales de cámara. Cada proyección captura una vista parcial del objeto o escena 3D, y estas vistas se combinan para obtener una representación más completa. Luego, los algoritmos procesan todas las imágenes obtenidas desde estas perspectivas para obtener información completa de la escena. Por ejemplo, el enfoque FCN multiseuencia (Zeng y Zheng, 2017) utiliza una red neuronal convolucional completamente conectada para predecir puntuaciones por píxel en las imágenes proyectadas. La etiqueta semántica final de cada punto en la nube se obtiene fusionando las puntuaciones re proyectadas de las diferentes vistas. Sin embargo, los métodos de vista múltiple pueden ser sensibles a la selección de los puntos de vista y sufrir de oclusiones, lo que puede resultar en una pérdida de información durante el proceso de proyección.

Los métodos basados en representaciones esférica transforman la nube de puntos en un sistema de coordenadas esféricas, donde cada punto se representa por su distancia radial y sus ángulos azimutales y polares. Esta representación captura la geometría global de la escena. Esto tiene como objetivo lograr una segmentación semántica rápida y precisa de las nubes de puntos 3D utilizando redes neuronales especializadas. Un ejemplo notable es SqueezeNet (Iandola y cols.,

2016), una arquitectura de red compacta y eficiente. Las versiones mejoradas, como SqueezeSegV2 (B. Wu, Zhou, Zhao, Yue, y Keutzer, 2018), abordan el problema del cambio de dominio mediante técnicas de adaptación de dominio no supervisadas. Otro enfoque interesante es RangeNet++ (Milioto, Vizzo, Behley, y Stachniss, 2019), que ofrece segmentación semántica en tiempo real de nubes de puntos LiDAR. Este método transfiere etiquetas semánticas de imágenes de rango 2D a nubes de puntos 3D y emplea un paso de posprocesamiento basado en KNN acelerado por GPU para mitigar los errores de discretización y las salidas de inferencia borrosas. En comparación con los métodos de proyección de vista única, la representación esférica retiene más información geométrica y es más adecuada para etiquetar nubes de puntos LiDAR.

Los métodos basados en proyecciones han demostrado ser efectivos en diversas tareas de análisis de nubes de puntos 3D, como la clasificación de objetos y la segmentación semántica. Al aprovechar las CNN 2D, estos enfoques pueden beneficiarse de arquitecturas de red bien estudiadas y optimizadas, como ResNet (He, Zhang, Ren, y Sun, 2016), U-Net (Ronneberger, Fischer, y Brox, 2015b) y DeepLab (Chen, Papandreou, Kokkinos, Murphy, y Yuille, 2017). Además, la representación 2D resultante suele ser más compacta y eficiente en términos de almacenamiento y procesamiento en comparación con las nubes de puntos 3D sin procesar. Sin embargo, también tienen algunas limitaciones. Durante el proceso de proyección, se puede perder información geométrica importante, especialmente cuando se trata de estructuras complejas o oclusiones. Además, la selección de los puntos de vista o las proyecciones óptimas puede ser un desafío y puede requerir un ajuste cuidadoso para capturar adecuadamente las características relevantes de la escena.

A pesar de estas limitaciones, los métodos basados en proyecciones siguen siendo una estrategia popular y efectiva para la segmentación semántica de nubes de puntos 3D. Su capacidad para aprovechar las técnicas de procesamiento de imágenes 2D bien establecidas y su eficiencia computacional los convierten en una opción atractiva en muchas aplicaciones. Además, se pueden combinar con otros enfoques, como los métodos basados en puntos o los métodos basados en discretización, para obtener una representación más completa y precisa de la escena 3D.

En resumen, los métodos basados en proyecciones ofrecen una forma simplificada y eficiente de abordar la segmentación semántica de nubes de puntos 3D al transformar los datos en representaciones 2D más manejables. Aunque pueden perder cierta información geométrica, su capacidad para aprovechar las CNN 2D y su eficiencia computacional los convierten en una herramienta valiosa en el análisis y la comprensión de las nubes de puntos 3D.

2.2.2. Métodos basados en discretización

Los métodos basados en discretización abordan la segmentación semántica de nubes de puntos 3D mediante la transformación de los datos continuos en representaciones discretas estructuradas. Estas representaciones discretas, como las cuadrículas de ocupación o los octrees, permiten aplicar operaciones de convolución 3D de manera eficiente y efectiva. Al discretizar el espacio 3D, estos métodos buscan capturar la geometría y las relaciones espaciales de los puntos de una manera más regular y ordenada.

2.2.2.1. Representación discreta densa

La representación discreta densa es un enfoque común en los métodos basados en discretización. En este enfoque, las nubes de puntos 3D se dividen en vóxeles de ocupación, que son elementos volumétricos discretos que representan una región del espacio 3D. Luego, se aplican redes neuronales convolucionales 3D (3D-CNN) a estos vóxeles para realizar la segmentación semántica. Cada vóxel se etiqueta con una clase semántica y todos los puntos dentro de un mismo vóxel reciben la misma etiqueta. Este enfoque permite aprovechar la estructura regular de los vóxeles y aplicar operaciones de convolución estándar en 3D.

Sin embargo, la representación discreta densa tiene algunas limitaciones. La precisión de la segmentación está limitada por la resolución de los vóxeles, y los artefactos pueden aparecer en los límites de los vóxeles debido a la partición discreta de la nube de puntos (Guo y cols., 2021). Para abordar estos problemas, se han propuesto mejoras, como SEGCloud (Tchapmi, Choy, Armeni, Gwak, y Savarese, 2017), que utiliza interpolación trilineal determinista para mapear las predicciones de vóxeles gruesos de vuelta a la nube de puntos original, logrando una segmentación más fina y consistente.

Otro enfoque interesante son las redes volumétricas, que presentan una buena escalabilidad para las 3D-CNN. Estas redes pueden ser entrenadas y probadas en nubes de puntos de diferentes tamaños espaciales. Un ejemplo notable es la red de puntos totalmente convolucional (FCPN) (Rethage, Wald, Sturm, Navab, y Tombari, 2018), que abstrae jerárquicamente diferentes niveles de relaciones geométricas en las nubes de puntos utilizando convoluciones 3D y agrupación de características. FCPN puede procesar eficientemente

nubes de puntos a gran escala y ofrece una buena escalabilidad durante la inferencia.

2.2.2.2. Representación discreta escasa

Aunque la representación discreta densa ha demostrado ser efectiva, puede ser computacionalmente costosa y consumir mucha memoria, especialmente para nubes de puntos grandes y de alta resolución. Para abordar este problema, se han propuesto métodos de representación discreta escasa. Estos métodos aprovechan la naturaleza escasa de las nubes de puntos y utilizan estructuras de datos eficientes, como los octrees, para representar y procesar los datos de manera más eficiente.

Uno de los enfoques más destacados en esta categoría son las redes convolucionales dispersas de subvariedades (Graham, Engelcke, y Van Der Maaten, 2018). Estas redes utilizan estructuras de indexación especializadas para restringir las operaciones de convolución solo a los vóxeles ocupados, lo que reduce significativamente los costos computacionales y de memoria. Además, estas redes controlan la escasez de las características aprendidas, lo que permite un procesamiento eficiente de datos dispersos y de alta dimensión.

Otro método notable es MinkowskiNet (Choy, Gwak, y Savarese, 2019), una red neuronal convolucional espacio-temporal 4D que utiliza convoluciones dispersas generalizadas. MinkowskiNet puede manejar eficientemente nubes de puntos dispersas y ha demostrado un buen rendimiento en tareas de segmentación semántica. Además, Sparse Lattice Networks (SPLATNet) (Su y cols., 2018) utiliza una estructura de datos de lattice disperso permutohedral para interpolar eficientemente las nubes de puntos y aplicar convoluciones bilaterales, lo que permite un procesamiento conjunto flexible de imágenes y nubes

de puntos.

Recientemente, LatticeNet (Rosu, Schütt, Quenzel, y Behnke, 2021) ha propuesto un enfoque innovador para procesar nubes de puntos grandes y dispersas. LatticeNet utiliza un módulo de interpolación dependiente de los datos llamado DeformSlice para proyectar las características aprendidas de vuelta a la nube de puntos original, logrando una segmentación precisa y eficiente.

En resumen, los métodos basados en discretización abordan la segmentación semántica de nubes de puntos 3D mediante la transformación de los datos en representaciones discretas estructuradas. Tanto los enfoques densos como los escasos han demostrado ser efectivos, con avances recientes que permiten un procesamiento más eficiente y escalable de grandes nubes de puntos. Estos métodos aprovechan la estructura regular de las representaciones discretas para aplicar operaciones de convolución 3D y capturar las relaciones espaciales y geométricas de los puntos. Aunque la discretización puede introducir cierta pérdida de información, los métodos basados en discretización siguen siendo una opción popular y efectiva para la segmentación semántica de nubes de puntos 3D.

2.2.3. Métodos basados en puntos

Los métodos basados en puntos son enfoques que operan directamente sobre las nubes de puntos 3D sin requerir una transformación previa a representaciones 2D o discretas. Estos métodos buscan aprovechar la naturaleza irregular y dispersa de las nubes de puntos, y han ganado popularidad en los últimos años debido a su capacidad para capturar de manera eficiente las características geométricas y las relaciones espaciales de los puntos.

Sin embargo, trabajar directamente con nubes de puntos presenta

algunos desafíos. A diferencia de las imágenes 2D o las representaciones discretas, las nubes de puntos carecen de una estructura regular y ordenada, lo que dificulta la aplicación directa de las operaciones de convolución tradicionales. Además, las nubes de puntos pueden tener una densidad variable y pueden contener un gran número de puntos, lo que requiere un procesamiento eficiente y escalable.

Para abordar estos desafíos, se han propuesto diferentes enfoques dentro de los métodos basados en puntos. Estos enfoques se pueden clasificar en cuatro categorías principales: métodos de perceptrón multicapa (MLP) por puntos, métodos de convolución por puntos, métodos basados en redes neuronales recurrentes (RNN) y métodos basados en grafos.

2.2.3.1. Métodos de perceptrón multicapa (MLP) por puntos

Los métodos de MLP por puntos utilizan redes neuronales de perceptrón multicapa para procesar directamente las nubes de puntos. Estos métodos aplican capas MLP de manera compartida a cada punto de la nube, lo que permite extraer características locales y globales de manera eficiente.

Uno de los métodos pioneros en esta categoría es PointNet (Qi, Su, y cols., 2017), que aplica transformaciones MLP individualmente a cada punto y luego agrega las características resultantes para obtener una representación global de la nube de puntos. PointNet ha demostrado ser efectivo en tareas de clasificación y segmentación de nubes de puntos. Sin embargo, PointNet no captura explícitamente las relaciones locales entre los puntos.

Para abordar esta limitación, se han propuesto extensiones de PointNet, como PointNet++ (Qi, Yi, Su, y Guibas, 2017), que

introduce una jerarquía de agrupamiento de puntos y aplica PointNet a diferentes escalas y resoluciones. Esto permite capturar características locales y relaciones espaciales de manera más efectiva.

Otros métodos de MLP por puntos se han centrado en mejorar la capacidad de capturar las relaciones locales entre los puntos. Por ejemplo, DGCNN (Y. Wang y cols., 2019) utiliza un gráfico dinámico para modelar las relaciones entre los puntos y actualiza el gráfico en cada capa de la red. Esto permite un aprendizaje más flexible y adaptativo de las características locales.

2.2.3.2. Métodos de convolución por puntos

Los métodos de convolución por puntos extienden el concepto de convolución a las nubes de puntos irregulares. Estos métodos definen operaciones de convolución especializadas que pueden operar directamente sobre los puntos y sus vecindarios locales.

Uno de los enfoques más destacados en esta categoría es la convolución de puntos (PointConv) (W. Wu, Qi, y Fuxin, 2019). PointConv define una operación de convolución que toma en cuenta la densidad local de los puntos y utiliza funciones de interpolación para agregar la información de los puntos vecinos. Esto permite capturar de manera efectiva las características locales y las relaciones espaciales en las nubes de puntos.

Otro método notable es la convolución de puntos del núcleo (KPConv) (Thomas y cols., 2019), que utiliza un núcleo de convolución definido en el espacio continuo y lo aplica a los puntos vecinos. KPConv permite un aprendizaje flexible de las características locales y ha demostrado un rendimiento sobresaliente en tareas de segmentación semántica de nubes de puntos.

Además de estos enfoques, se han propuesto otras variantes de

convolución por puntos, como la convolución de puntos adaptativa (AdaptConv) (H. Zhou y cols., 2021) y la convolución de puntos atencional (AttentionConv) (W. Wu, Fuxin, y Shan, 2023), que incorporan mecanismos de atención para capturar de manera más efectiva las relaciones entre los puntos.

2.2.3.3. Métodos basados en redes neuronales recurrentes (RNN)

Los métodos basados en RNN abordan el procesamiento secuencial de las nubes de puntos utilizando arquitecturas de redes neuronales recurrentes. Estos métodos tratan las nubes de puntos como secuencias de puntos y aplican operaciones recurrentes para capturar las dependencias y relaciones temporales entre los puntos.

Uno de los enfoques pioneros en esta categoría es PointRNN (Fan y Yang, 2019), que aplica una red neuronal recurrente a las nubes de puntos para capturar las dependencias espaciales y temporales. PointRNN utiliza una estructura jerárquica para procesar los puntos a diferentes escalas y combina las características aprendidas para obtener una representación global de la nube de puntos.

Otros métodos basados en RNN incluyen 3D-RNN (Ye, Li, Huang, Du, y Zhang, 2018), que utiliza una RNN bidireccional para capturar las relaciones espaciales en las nubes de puntos, y Grid-GRU (Xie, Li, Wang, Li, y Qu, 2023), que combina una estructura de cuadrícula con unidades recurrentes GRU para procesar eficientemente las nubes de puntos.

2.2.3.4. Métodos basados en grafos

Los métodos basados en grafos representan las nubes de puntos como grafos, donde los puntos son los nodos y las aristas representan las relaciones entre los puntos. Estos métodos aprovechan la estructura de los grafos para capturar las relaciones espaciales y geométricas de manera explícita.

Uno de los enfoques destacados en esta categoría es DGCNN (Y. Wang y cols., 2019), mencionado anteriormente en los métodos de MLP por puntos. DGCNN construye un gráfico dinámico en cada capa de la red para modelar las relaciones entre los puntos y actualiza el gráfico adaptivamente durante el entrenamiento.

Otro método notable es GraphPointNet (Pistilli, 2020), que combina las fortalezas de PointNet y los grafos para capturar tanto las características locales como las relaciones globales en las nubes de puntos. GraphPointNet construye un grafo de k -vecinos más cercanos para cada punto y aplica operaciones de convolución de grafos para aprender características locales y globales.

Otros enfoques basados en grafos incluyen GAC (L. Wang, Huang, Hou, Zhang, y Shan, 2019), que utiliza una capa de convolución de atención de grafos para capturar las relaciones entre los puntos, y GACNet (L. Wang y cols., 2019), que extiende GAC con un módulo de atención de grafos jerárquico para capturar características a diferentes escalas.

En resumen, los métodos basados en puntos abordan la segmentación semántica de nubes de puntos 3D operando directamente sobre los puntos y sus relaciones espaciales. Estos métodos han demostrado ser efectivos y han logrado un rendimiento destacado en diversas tareas de análisis de nubes de puntos. Aunque trabajar

directamente con nubes de puntos presenta desafíos debido a su naturaleza irregular y dispersa, los avances recientes en métodos de MLP por puntos, convolución por puntos, RNN y grafos han permitido un procesamiento eficiente y preciso de las nubes de puntos. Estos métodos aprovechan las características geométricas y las relaciones espaciales de los puntos para obtener una segmentación semántica detallada y de alta calidad.

2.3. Estado del Arte de Segmentación Semántica de Nube de Puntos Aéreas

En esta sección, se exploran dos artículos destacados que han tenido un impacto significativo en el campo de la segmentación semántica de nubes de puntos aéreas. Estos artículos han sido seleccionados debido a su relevancia, innovación y contribución al avance del estado del arte en esta área de investigación.

El primer artículo se centra en el algoritmo de segmentación semántica conocido como KPConv (Kernel Point Convolution) (Thomas y cols., 2019). Este método ha demostrado un rendimiento excepcional en diversas métricas de evaluación y ha establecido un nuevo punto de referencia en la segmentación semántica de nubes de puntos. KPConv introduce un novedoso operador de convolución que opera directamente sobre los puntos y sus vecindarios locales, lo que permite capturar de manera efectiva las características geométricas y las relaciones espaciales en las nubes de puntos.

El segundo artículo aborda la fusión de datos fotogramétricos y LiDAR a través de una arquitectura de red neuronal convolucional conocida como U-Net (Ronneberger y cols., 2015b). Este enfoque combina las fortalezas de ambas modalidades de datos para mejorar la

segmentación semántica de nubes de puntos aéreas. Aunque los métodos empleados en este artículo difieren de los propuestos en nuestro trabajo, ha sido una referencia valiosa para comprender las técnicas y desafíos involucrados en la fusión de datos de diferentes fuentes.

A continuación, se profundizará en cada uno de estos artículos, destacando sus contribuciones clave, metodologías y resultados obtenidos. Se discutirán las fortalezas y limitaciones de cada enfoque, así como su impacto en el campo de la segmentación semántica de nubes de puntos aéreas.

2.3.1. KPConv (Kernel Point Convolution)

KPConv es un método innovador de segmentación semántica de nubes de puntos propuesto por Thomas et al. (Thomas y cols., 2019). Este método introduce un nuevo operador de convolución llamado KPConv, que está específicamente diseñado para operar en nubes de puntos desordenadas y dispersas. A diferencia de las convoluciones tradicionales que operan en datos estructurados como imágenes o vóxeles, KPConv define un núcleo de convolución en el espacio continuo y lo aplica directamente a los puntos y sus vecindarios locales.

La idea central detrás de KPConv es aprender una función de kernel que mapea la geometría local alrededor de cada punto a un espacio de características. El kernel se define como una combinación lineal de funciones base, cuyos pesos son aprendidos durante el entrenamiento. Estas funciones base pueden ser polinomios, funciones gaussianas o cualquier otra función adecuada para capturar las características geométricas locales.

Una de las principales ventajas de KPConv es su capacidad para adaptarse a la geometría local de la nube de puntos. Al aprender los

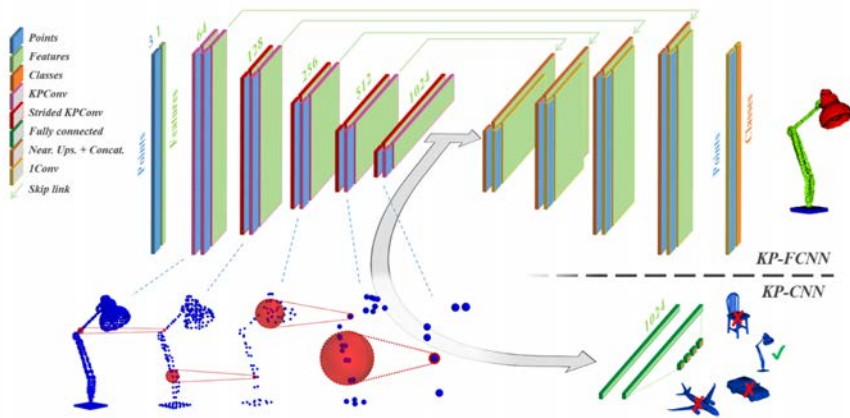


Figura 2.5. Funcionamiento de KPConv en una arquitectura.
Imagen recuperada de (Thomas y cols., 2019).

pesos del kernel en función de la distribución espacial de los puntos, KPConv puede capturar de manera efectiva las relaciones y patrones geométricos locales. Esto es especialmente importante en el contexto de las nubes de puntos aéreas, donde la densidad de puntos puede variar significativamente y las estructuras geométricas pueden ser complejas.

Además de su capacidad para adaptarse a la geometría local, KPConv también introduce el concepto de puntos de kernel deformables. Estos puntos de kernel se aprenden durante el entrenamiento y pueden ajustar su posición y orientación para capturar mejor las características relevantes de la nube de puntos. Esto permite una mayor flexibilidad y capacidad de representación en comparación con los enfoques de convolución tradicionales.

La Figura 2.5 muestra el funcionamiento de KPConv en una arquitectura de red neuronal. Se puede observar cómo el operador de convolución KPConv se aplica a la nube de puntos de entrada en

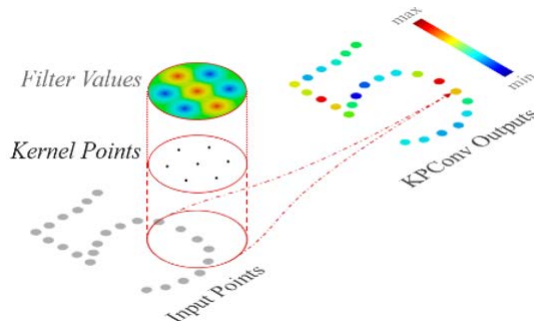


Figura 2.6. Funcionamiento de KPCConv en puntos 2D.
Imagen recuperada de (Thomas y cols., 2019).

diferentes escalas y niveles de la red. Esto permite capturar características locales y globales de la nube de puntos y generar una segmentación semántica precisa.

Otro aspecto importante de KPCConv es su capacidad para manejar la variabilidad en la densidad de puntos. A diferencia de otros enfoques que requieren una densidad de puntos uniforme, KPCConv puede adaptarse a nubes de puntos con densidad variable. Esto se logra mediante el uso de un radio de influencia adaptativo para cada punto de kernel, lo que permite capturar características relevantes incluso en regiones con baja densidad de puntos.

La Figura 2.6 ilustra el funcionamiento de KPCConv en un ejemplo de puntos 2D. Se puede observar cómo los puntos de kernel se adaptan a la geometría local de los puntos y capturan las características relevantes. Esto permite una convolución efectiva incluso en nubes de puntos irregulares y dispersas.

Los experimentos realizados por Thomas et al. (Thomas y cols., 2019) demuestran el potencial de KPCConv para la segmentación

semántica de nubes de puntos aéreas. En conjuntos de datos desafiantes como ScanNet (Dai y cols., 2017) y S3DIS (Armeni y cols., 2016), KPConv logró resultados de vanguardia, superando a otros métodos de segmentación semántica basados en puntos y a enfoques basados en vóxeles.

Sin embargo, KPConv también presenta algunas limitaciones. El cálculo de las convoluciones en el espacio continuo puede ser computacionalmente costoso, especialmente para nubes de puntos grandes. Además, la selección de las funciones base adecuadas y la inicialización de los puntos de kernel pueden requerir ajustes cuidadosos para obtener un rendimiento óptimo.

A pesar de estas limitaciones, KPConv ha demostrado ser un método poderoso y versátil para la segmentación semántica de nubes de puntos aéreas. Su capacidad para adaptarse a la geometría local y aprender características discriminativas lo convierte en una herramienta valiosa para una amplia gama de aplicaciones, como la cartografía urbana, la planificación del territorio y la gestión de infraestructuras.

2.3.1.1. Ventajas de KPConv

KPConv presenta varias ventajas significativas en comparación con otros métodos de segmentación semántica de nubes de puntos (Thomas y cols., 2019). Una de las principales fortalezas de KPConv radica en su capacidad para capturar y adaptarse a las estructuras geométricas locales presentes en las nubes de puntos. Al aprender los pesos del kernel en función de la distribución espacial de los puntos, KPConv puede extraer de manera efectiva las relaciones y patrones geométricos locales, lo que resulta crucial para una segmentación precisa.

Otra ventaja destacada de KPConv es su flexibilidad para manejar nubes de puntos con densidad variable. A diferencia de otros enfoques

que requieren una densidad de puntos uniforme, KPConv utiliza un radio de influencia adaptativo para cada punto de kernel. Esto permite capturar características relevantes incluso en regiones con baja densidad de puntos, lo que es especialmente beneficioso en escenarios donde la densidad de puntos puede variar significativamente, como en el caso de las nubes de puntos aéreas.

Además, KPConv introduce el concepto innovador de puntos de kernel deformables. Estos puntos de kernel se aprenden durante el proceso de entrenamiento y tienen la capacidad de ajustar su posición y orientación para adaptarse de manera óptima a las características específicas de la nube de puntos. Esta capacidad de deformación permite una mayor flexibilidad y poder de representación en comparación con las técnicas de convolución tradicionales, lo que se traduce en una segmentación más precisa y detallada.

La eficiencia computacional es otra ventaja notable de KPConv. Al operar directamente sobre los puntos y sus vecindarios locales, KPConv evita la necesidad de convertir la nube de puntos en una representación volumétrica o de malla, lo que puede ser computacionalmente costoso y consumir una gran cantidad de memoria. Esto hace que KPConv sea más eficiente en términos de recursos computacionales en comparación con otros enfoques que requieren una representación intermedia.

2.3.1.2. Limitaciones de KPConv

A pesar de sus numerosas ventajas, KPConv también presenta algunas limitaciones que deben considerarse. Una de las principales limitaciones es el costo computacional asociado al cálculo de las convoluciones en el espacio continuo. A medida que aumenta el tamaño y la complejidad de las nubes de puntos, el tiempo de procesamiento y los requisitos de memoria pueden volverse significativos. Esto puede ser

un desafío en aplicaciones que requieren un procesamiento en tiempo real o que involucran nubes de puntos extremadamente grandes.

Otra limitación de KPConv está relacionada con la selección de las funciones base utilizadas para definir el kernel. La elección adecuada de estas funciones base es crucial para capturar de manera efectiva las características geométricas locales. Sin embargo, determinar las funciones base óptimas puede requerir experiencia y ajustes cuidadosos. Una selección inapropiada de las funciones base puede limitar la capacidad de KPConv para aprender y representar de manera efectiva las estructuras geométricas presentes en la nube de puntos.

Además, la inicialización de los puntos de kernel también puede ser un desafío. Los puntos de kernel iniciales deben distribuirse de manera adecuada para cubrir de manera efectiva la nube de puntos y capturar las características relevantes. Una inicialización deficiente de los puntos de kernel puede afectar negativamente el rendimiento de KPConv y requerir un ajuste manual o heurísticas adicionales.

Es importante tener en cuenta que KPConv, como otros métodos de aprendizaje profundo, requiere una cantidad suficiente de datos de entrenamiento etiquetados para alcanzar un rendimiento óptimo. La disponibilidad de conjuntos de datos de nubes de puntos aéreas etiquetados puede ser limitada, lo que puede dificultar la generalización del modelo a nuevos escenarios o dominios.

A pesar de estas limitaciones, KPConv sigue siendo un método poderoso y versátil para la segmentación semántica de nubes de puntos. Sus ventajas, como la capacidad para adaptarse a la geometría local y manejar nubes de puntos con densidad variable, superan en gran medida sus limitaciones en muchos casos de uso prácticos. A medida que se desarrollen nuevas técnicas y se aborden estas

limitaciones, es probable que KPConv y sus variantes sigan desempeñando un papel importante en el análisis y la comprensión de las nubes de puntos aéreas.

2.3.2. Fusión de Datos Fotogramétricos y Lidar a través de U-Net

Este método se centra en la extracción de edificios urbanos en 3D mediante la fusión de LiDAR aerotransportado y nubes de puntos fotogramétricas. El procedimiento práctico tiene como objetivo brindar una precisión confiable en la extracción de edificios en 3D, que es crucial para diversas aplicaciones, como la planificación urbana, la gestión de ciudades, la evaluación de desastres, la cartografía digital, la planificación del transporte, el catastro y la gestión de redes de telecomunicaciones(P. Zhang y cols., 2022).

El estudio presenta un método práctico de extracción de edificios urbanos en 3D mediante la fusión de LiDAR y nubes de puntos fotogramétricas basadas en la segmentación del modelo de aprendizaje profundo de U-Net. El método implica usar el modelo U-Net para extraer polígonos de construcción de imágenes y aplicar una estrategia de fusión de nubes de puntos en cada polígono para aumentar la densidad de nubes de puntos para futuras extracciones, como se puede observar en la figura 2.7.

El procedimiento del método propuesto es el siguiente(P. Zhang y cols., 2022):

1. Generación de ortofotos y preparación de conjuntos de datos: Las imágenes capturadas por cinco cámaras aéreas se utilizan para generar una ortofoto, que se divide en conjuntos de entrenamiento, validación y prueba. Las técnicas de mejora de datos, como la

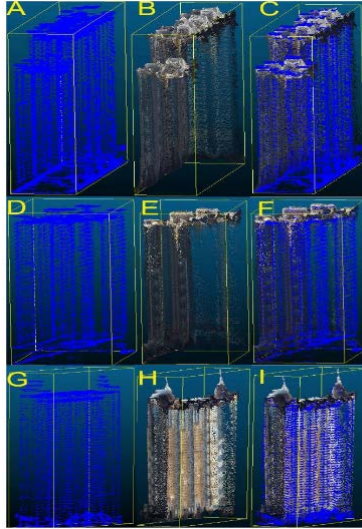


Figura 2.7. Nube de Puntos de Lidar, Fotogramétrica y Fusionados

Fuente: Imagen recuperada de (P. Zhang y cols., 2022)

traducción de imágenes, el volteo, la transformación de colores y la adición de ruido, se aplican a los conjuntos de capacitación y validación para mejorar el rendimiento del modelo de aprendizaje profundo.

2. Modelo de aprendizaje profundo de U-Net: El modelo U-Net es una red neuronal convolucional (el cual abordaremos más a detalle en la sección de metodología de la tesis).
3. Entrenamiento y Validación del Modelo U-Net: El modelo U-Net se entrena y valida mediante la librería de deep learning Pytorch en una plataforma computacional con un sistema operativo Ubuntu 18.04, CPU Intel Xeon, 125 GB de RAM y GPU

NVIDIA TITAN X. El tamaño del lote del modelo se establece en 8, con 400 iteraciones y una tasa de aprendizaje inicial de 0,0001. Adam se utiliza como optimizador de algoritmos y ReLU sirve como función de activación no lineal.

4. Generación de nubes de puntos fusionados: Las nubes de puntos LiDAR y fotogramétricas se fusionan en cada polígono de segmentación. La profundidad de cada píxel se calcula utilizando el método de coincidencia densa parches para densificar las nubes de puntos.
5. Geolocalización inicial y alineación: La densa nube de puntos se alinea con los metadatos de imágenes del GPS para reducir las diferencias significativas en rotación, escala y traducción entre LiDAR y las nubes de puntos fotogramétricas. Los resultados de segmentación de la predicción de U-Net se utilizan para separar las nubes de puntos de fotogrametría y LiDAR en diferentes partes.
6. Registro preciso de nubes de puntos: El algoritmo Iterative Closest Point (ICP) se utiliza para registrar con precisión las nubes de puntos fotogramétricas y LiDAR dentro de cada fragmento. Este algoritmo asigna iterativamente la correspondencia en función del criterio de distancia más cercana y encuentra la transformación rígida utilizando un enfoque de mínimos cuadrados.
7. Extracción de edificios utilizando el enfoque de diferencia de normales (DoN): Las nubes de puntos fusionados se utilizan para extraer edificios según el enfoque DoN. Este método calcula la diferencia entre las superficies normales en diferentes radios y clasifica los puntos en función de sus diferencias de superficies

normales. Los puntos se segmentan en función de sus distancias y conectividad, lo que facilita la extracción de arquitecturas de edificios complejas.

En resumen, el método presenta un procedimiento práctico que aprovecha las fortalezas de LiDAR y las nubes de puntos fotogramétricas para mejorar la extracción de edificios urbanos densos en 3D. Al fusionar estos dos tipos de datos, la nube de puntos resultante puede proporcionar información más precisa y detallada para la extracción de edificios urbanos en 3D.

Capítulo 3

Metodología

3.1. Introducción a la Metodología

La metodología propuesta en este trabajo se presenta como una serie de fases interconectadas, cada una de las cuales se enfoca en un aspecto específico del procesamiento y clasificación de nubes de puntos. Este enfoque modular permite una comprensión profunda de cada etapa, a la vez que garantiza la reproducibilidad y adaptabilidad del proceso a diversas aplicaciones, tales como análisis topográfico, modelado 3D, entre otras.

El flujo de trabajo comienza con la adquisición de la nube de puntos, seguida de una fase crucial de preprocesamiento. Esta etapa inicial es fundamental para garantizar la calidad y consistencia de los datos, sentando las bases para un análisis preciso y confiable en las fases posteriores.

Una vez que los datos han sido adecuadamente preprocesados, se someten a una serie de transformaciones y proyecciones que los preparan para su uso en algoritmos de aprendizaje automático. Estas

proyecciones permiten representar la información tridimensional en un formato bidimensional, facilitando la aplicación de técnicas de procesamiento de imágenes y aprendizaje profundo.

El núcleo de la metodología reside en el entrenamiento de un modelo de segmentación semántica basado en la arquitectura U-Net. Este modelo, ampliamente reconocido por su eficacia en tareas de segmentación de imágenes, es adaptado y optimizado para abordar las particularidades de las nubes de puntos proyectadas.

Tras el entrenamiento, el modelo U-Net es empleado para clasificar nuevas nubes de puntos, asignando a cada punto una etiqueta semántica. Este proceso de inferencia es seguido por una fase de posprocesamiento, donde se aplican técnicas como la propagación de etiquetas para refinar y mejorar la clasificación inicial.

El resultado final es una nube de puntos clasificada semánticamente, donde cada punto está asociado a una categoría específica. Esta información enriquecida abre un abanico de posibilidades para análisis posteriores y aplicaciones en diversos campos, desde la planificación urbana hasta la realidad virtual y aumentada.

En resumen, la metodología propuesta en este trabajo integra técnicas de preprocesamiento de datos, proyección, aprendizaje profundo y posprocesamiento en un flujo de trabajo coherente y efectivo. Cada fase ha sido cuidadosamente diseñada y validada para abordar los desafíos específicos del procesamiento y clasificación de nubes de puntos.

A continuación, se detallarán los aspectos técnicos de cada fase, profundizando en los algoritmos empleados, sus fundamentos matemáticos y su papel en el contexto global de la metodología. Se prestará especial atención a la interconexión entre las diferentes

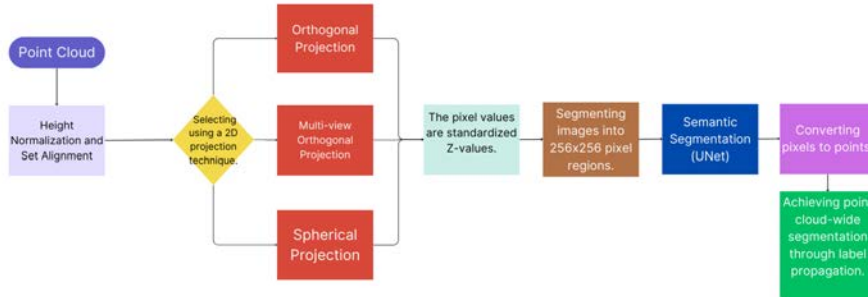


Figura 3.1. Diagrama de flujo de la metodología propuesta.
Fuente: Elaboración propia.

etapas, destacando cómo cada una contribuye a la consecución del objetivo final: una clasificación semántica precisa y útil de las nubes de puntos. La Figura 3.1 presenta un diagrama esquemático del flujo de trabajo de la metodología, desde la adquisición de la nube de puntos hasta la obtención de la nube de puntos clasificada.

3.2. Preprocesamiento de Nubes de Puntos

El preprocesamiento de nubes de puntos es una etapa crucial en la metodología de análisis y clasificación de datos tridimensionales. Su objetivo principal es garantizar la calidad, consistencia y adecuación de los datos para las etapas posteriores del proceso, como la segmentación semántica y la clasificación.

La importancia del preprocesamiento radica en su capacidad para

abordar diversos desafíos inherentes a las nubes de puntos crudas, como la presencia de ruido, outliers, variaciones en la densidad de puntos y diferencias en los sistemas de referencia. Al tratar estos aspectos, el preprocesamiento sienta las bases para un análisis preciso y confiable.

Entre los objetivos específicos del preprocesamiento se encuentran:

- Normalización de la altura: Este proceso busca ajustar las alturas de los puntos en relación a un plano de referencia común, como el nivel del suelo. La normalización de altura es esencial para distinguir entre puntos pertenecientes al terreno y aquellos que representan objetos o estructuras sobre él.
- Alineación del conjunto de datos: Para garantizar una proyección precisa y consistente de la nube de puntos, es necesario alinearla en torno a un punto de referencia común, como su centroide. La alineación facilita la comparación y la integración de diferentes conjuntos de datos.
- Proyección: La transformación de los datos tridimensionales a un espacio bidimensional es un paso clave para aplicar técnicas de procesamiento de imágenes y aprendizaje profundo. Diferentes métodos de proyección, como la proyección ortogonal, esférica o multi-vista, pueden ser empleados según las características de la nube de puntos y los objetivos del análisis.

Cada una de estas etapas del preprocesamiento implica la aplicación de algoritmos y técnicas específicas, que serán detalladas en las siguientes subsecciones. Desde el algoritmo Cloth Simulator Filter (CSF) para la normalización de altura, hasta las transformaciones matriciales para la proyección multi-vista, cada componente del

preprocesamiento juega un papel vital en la adaptación de los datos para su posterior análisis.

3.2.1. Normalización de altura

La normalización de altura es un proceso fundamental en el análisis de nubes de puntos, especialmente en aplicaciones relacionadas con la teledetección y el análisis topográfico. El objetivo principal de este proceso es ajustar las alturas de los puntos en relación a un plano de referencia común, típicamente el nivel del suelo. Esta normalización facilita la identificación y segmentación de objetos y estructuras sobre el terreno, como vegetación o edificaciones.

Para llevar a cabo la normalización de altura, se empleó el algoritmo Cloth Simulator Filter (CSF), propuesto por Zhang et al. (W. Zhang y cols., 2016). Este algoritmo se basa en la simulación del comportamiento de un paño virtual que se ajusta a la superficie del terreno, permitiendo separar los puntos pertenecientes al suelo de aquellos que representan objetos sobre él.

3.2.1.1. Fundamentos de la simulación de tela

La simulación de tela es un concepto ampliamente utilizado en gráficos por computadora para modelar el comportamiento de tejidos y superficies deformables. En el contexto del filtrado de nubes de puntos, la idea central es representar el terreno como una superficie de tela virtual que se adapta a los puntos de la nube.

En este modelo, la tela se representa como una malla compuesta por partículas interconectadas mediante resortes virtuales, siguiendo la ley de Hooke. Cada partícula tiene una masa asociada y su posición en el espacio tridimensional define la forma y geometría de la tela.

La simulación de tela se rige por la segunda ley de Newton, que relaciona las fuerzas aplicadas a una partícula con su aceleración y masa. La ecuación que describe este comportamiento es:

$$m \frac{\partial^2 X(t)}{\partial t^2} = F_{\text{ext}}(X, t) + F_{\text{int}}(X, t) \quad (3.1)$$

donde m es la masa de la partícula, $X(t)$ es su posición en función del tiempo, F_{ext} representa las fuerzas externas (como la gravedad) y F_{int} representa las fuerzas internas (como la tensión de los resortes).

3.2.1.2. Modificación de la simulación de tela para el filtrado de nubes de puntos

Para adaptar la simulación de tela al filtrado de nubes de puntos, Zhang et al. (W. Zhang y cols., 2016) propusieron varias modificaciones clave, como se ilustra en la Figura 3.2. Estas modificaciones incluyen:

- Restricción del movimiento de las partículas a la dirección vertical.
- Definición de partículas inmóviles cuando alcanzan el nivel del suelo.
- División de las fuerzas en dos pasos discretos para simplificar el cálculo.

3.2.1.3. Implementación de CSF

La implementación del algoritmo CSF implica una serie de pasos que se detallan en la Figura 3.3. Estos pasos incluyen la inversión de la nube de puntos, la inicialización de la malla de tela, la proyección de los puntos en un plano horizontal, el cálculo de las fuerzas externas e internas, y la clasificación final de los puntos como terreno o objetos.

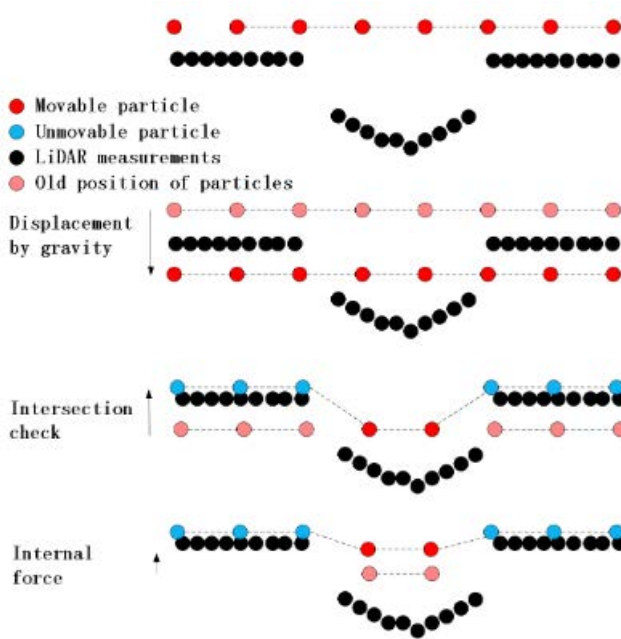


Figura 3.2. Pasos principales en el algoritmo simulación de tela. Imagen recuperada de (W. Zhang y cols., 2016).

Un aspecto clave de la implementación es el cálculo de las fuerzas que actúan sobre las partículas. La fuerza externa, típicamente la gravedad, se calcula utilizando la ecuación:

$$X(t + \Delta t) = 2X(t) - X(t - \Delta t) + \frac{G}{m}\Delta t^2 \quad (3.2)$$

donde $X(t)$ es la posición de la partícula en el tiempo t , Δt es el intervalo de tiempo, G es la magnitud de la fuerza externa y m es la masa de la partícula. Por otro lado, las fuerzas internas se calculan considerando el desplazamiento relativo de las partículas vecinas:

$$\vec{d} = \frac{1}{2}b(\vec{p}_i - \vec{p}_0) \cdot \vec{n} \quad (3.3)$$

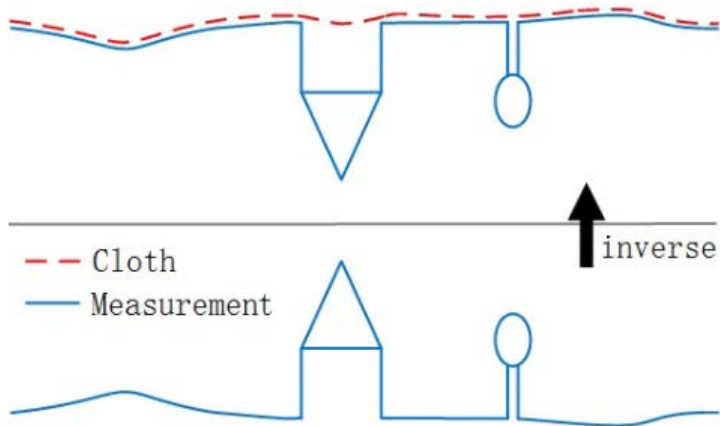


Figura 3.3. Descripción gráfica del algoritmo de simulación de tela. Imagen recuperada de (W. Zhang y cols., 2016).

donde \vec{d} es el vector de desplazamiento, \vec{p}_i y \vec{p}_0 son las posiciones inicial y de referencia de la partícula, respectivamente, b es un factor de escala y \vec{n} es el vector normal a la superficie.

3.2.1.4. Proceso de normalización utilizando CSF

Una vez que se ha aplicado el algoritmo CSF y se han clasificado los puntos como terreno o objetos, el proceso de normalización de altura se lleva a cabo restando la altura del terreno (representada por el DEM generado por CSF) de la altura original de cada punto.

Dado un punto $p = (x, y, z)$ en la nube de puntos y la altura del terreno en esa ubicación $z_{DEM}(x, y)$, la altura normalizada z_{norm} se calcula como:

$$z_{norm} = z - z_{DEM}(x, y) \quad (3.4)$$

La Figura 3.5 muestra una comparación entre una nube de puntos

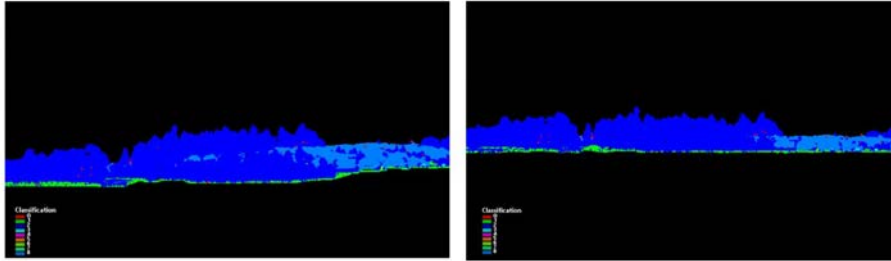


Figura 3.4. Comparación de una nube de puntos sin normalizar y normalizada sus valores de Z. Elaboración propia.

sin normalizar y una normalizada, destacando la importancia de este proceso para el análisis posterior.

En resumen, la normalización de altura utilizando el algoritmo CSF es un paso crucial en el preprocesamiento de nubes de puntos. Al separar los puntos del terreno de los objetos sobre él, este proceso facilita la segmentación y clasificación posteriores. La implementación de CSF implica la simulación de una tela virtual que se adapta a la superficie del terreno, aplicando principios físicos y geométricos para clasificar los puntos. El resultado final es una nube de puntos normalizada en altura, lista para ser analizada y procesada en las etapas siguientes de la metodología.

3.2.1.5. Proceso de Normalización

La estandarización de valores de píxeles a valores Z es un procedimiento de normalización que transforma la intensidad de los píxeles para reflejar características específicas, como la distancia o la altura relativa, permitiendo una interpretación más precisa de la información espacial.

La normalización implica sustraer la altura del DEM creado a partir de los puntos de suelo clasificados por CSF, de la altura original de cada punto. Definimos un punto p en la nube como:

$$p = (x, y, z)$$

x: Representa la coordenada en el eje horizontal que usualmente define la anchura o longitud en el espacio. E

y: Representa la coordenada en el eje vertical que usualmente define la profundidad o latitud en el espacio.

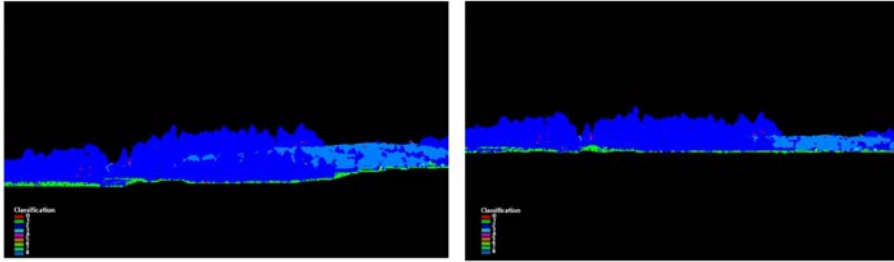
z: Representa la coordenada en el eje que define la altura o elevación del punto en el espacio. Esta es la distancia vertical del punto respecto a un plano de referencia, que podría ser el nivel del suelo o cualquier otro plano horizontal definido como origen para z.

Si $z_{DEM}(x, y)$ representa la altura del DEM en esa ubicación, entonces la altura normalizada z_{norm} es:

$$z_{norm} = z - z_{DEM}(x, y) \quad (3.5)$$

Esta operación de normalización es crucial para interpretar características superficiales como vegetación o estructuras humanas. La figura 3.5 muestra una comparativa de una nube de puntos sin normalizar y normalizada sus valores de Z.

Figura 3.5. Comparación de una nube de puntos sin normalizar y normalizada sus valores de Z



Fuente: Elaboración Propia

3.2.2. Alineación del conjunto de datos

Tras la normalización de altura, el siguiente paso en el preprocesamiento de nubes de puntos es la alineación del conjunto de datos. Este proceso tiene como objetivo establecer un sistema de referencia común para todos los puntos, facilitando así la comparación y la integración de diferentes nubes de puntos.

La importancia de la alineación radica en su capacidad para garantizar una proyección precisa y consistente de los datos. Al alinear la nube de puntos en torno a un punto de referencia común, como su centroide, se eliminan posibles desplazamientos o rotaciones entre distintos conjuntos de datos, lo que resulta esencial para un análisis comparativo y una visualización coherente.

3.2.2.1. Cálculo del centroide

El centroide de una nube de puntos es un concepto clave en la alineación del conjunto de datos. Se define como el punto central o el

promedio ponderado de todas las posiciones de los puntos en la nube. Matemáticamente, el centroide C de una nube de puntos con n puntos se calcula como:

$$C = (\bar{x}, \bar{y}) \quad (3.6)$$

donde \bar{x} y \bar{y} son las coordenadas medias de todos los puntos:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.7)$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (3.8)$$

El cálculo del centroide es un paso esencial para la alineación, ya que proporciona un punto de referencia común para todos los puntos de la nube.

3.2.2.2. Proceso de alineación de los puntos

Una vez que se ha calculado el centroide, el proceso de alineación de los puntos implica la traslación de cada punto de la nube en relación al centroide. Esto se logra restando las coordenadas del centroide a las coordenadas de cada punto:

$$(x'_i, y'_i) = (x_i - \bar{x}, y_i - \bar{y}) \quad (3.9)$$

donde (x'_i, y'_i) son las coordenadas del punto i después de la alineación, y (x_i, y_i) son las coordenadas originales del punto.

Este proceso de alineación garantiza que el centroide de la nube de puntos coincida con el origen del sistema de coordenadas, lo que facilita la comparación y la integración de diferentes conjuntos de datos.

La alineación del conjunto de datos es un paso crucial en el preprocesamiento de nubes de puntos, ya que establece una base común para el análisis y la visualización posteriores. Al alinear los puntos en torno al centroide, se eliminan posibles desplazamientos o rotaciones entre diferentes nubes de puntos, lo que resulta esencial para una proyección precisa y consistente de los datos.

En las secciones anteriores, se abordó la normalización de altura utilizando el algoritmo Cloth Simulator Filter (CSF) para separar los puntos del terreno de los objetos sobre él. La alineación del conjunto de datos complementa este proceso al establecer un sistema de referencia común para todos los puntos, independientemente de su clasificación como terreno u objeto.

Una vez que la nube de puntos ha sido normalizada en altura y alineada en torno al centroide, está lista para ser proyectada a un espacio bidimensional. En la siguiente sección, se explorarán diferentes métodos de proyección, como la proyección ortogonal, esférica y multi-vista, y se discutirá cómo estas técnicas transforman los datos tridimensionales en representaciones bidimensionales adecuadas para el análisis y la visualización posteriores.

3.2.3. Proyección de nubes de puntos

Tras la normalización de altura y la alineación del conjunto de datos, el siguiente paso en el preprocesamiento es la proyección de la nube de puntos a un espacio bidimensional. Este proceso es fundamental para aplicar técnicas de procesamiento de imágenes y aprendizaje automático, ya que muchos algoritmos están diseñados para trabajar con datos 2D.

La proyección de nubes de puntos implica la transformación de los datos tridimensionales en una representación bidimensional,

manteniendo la mayor cantidad de información relevante posible. Existen diferentes métodos de proyección, cada uno con sus propias características y aplicaciones. En esta sección, se explorarán tres métodos principales: proyección ortogonal, proyección esférica y proyección multi-vista.

La elección del método de proyección depende de diversos factores, como la geometría de la escena, la distribución de los puntos y el objetivo del análisis. Cada método tiene sus ventajas y limitaciones, y es importante seleccionar el más adecuado para cada caso específico.

3.2.3.1. Proyección ortogonal

La proyección ortogonal es uno de los métodos más simples y ampliamente utilizados para proyectar nubes de puntos a un espacio bidimensional. En este tipo de proyección, los puntos se proyectan perpendicularmente sobre un plano, generalmente el plano XY .

Una de las principales ventajas de la proyección ortogonal es que preserva las distancias y los ángulos en el plano de proyección. Esto significa que las relaciones espaciales entre los puntos se mantienen, lo que es especialmente útil para análisis métricos y visualizaciones precisas.

La Figura 3.6 ilustra el concepto de proyección ortogonal, donde los rayos de proyección son perpendiculares al plano de proyección.

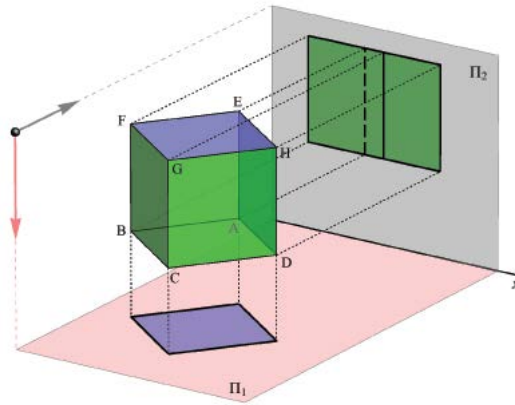


Figura 3.6. Proyección ortogonal. Imagen recuperada de
http://www.grad.hr/geomteh3d/Monge/06projekcije/projekcije_eng.html.

Para aplicar la proyección ortogonal a una nube de puntos, se siguen los siguientes pasos:

3.2.3.1.1. Cálculo del número de filas y columnas El primer paso es determinar las dimensiones de la imagen resultante, es decir, el número de filas y columnas. Estas dimensiones se calculan en función de los límites de la nube de puntos y la resolución deseada:

$$\text{número de columnas} = \left\lceil \frac{x \text{ máxima} - x \text{ mínima}}{\text{resolución}} \right\rceil \quad (3.10)$$

$$\text{número de filas} = \left\lceil \frac{y \text{ máxima} - y \text{ mínima}}{\text{resolución}} \right\rceil \quad (3.11)$$

donde x máxima, x mínima, y máxima y y mínima son los límites de

la nube de puntos en las direcciones X e Y, respectivamente, y resolución es la resolución espacial deseada.

3.2.3.1.2. Inicialización de la matriz A continuación, se inicializa una matriz de ceros con las dimensiones calculadas en el paso anterior. Esta matriz representará la imagen resultante de la proyección ortogonal:

$$\text{imagen} = \text{ceros}(\text{número de filas}, \text{número de columnas}, 4) \quad (3.12)$$

La tercera dimensión de la matriz (4) corresponde a los canales de la imagen: coordenadas X, Y, Z y la etiqueta de clasificación de cada punto.

3.2.3.1.3. Mapeo de puntos a píxeles Para cada punto p de la nube de puntos, se calculan las coordenadas del píxel correspondiente en la imagen resultante:

$$\text{columna} = \left\lfloor \frac{p_x - x \text{ mínima}}{\text{resolución}} \right\rfloor \quad (3.13)$$

$$\text{fila} = \left\lfloor \frac{y \text{ máxima} - p_y}{\text{resolución}} \right\rfloor \quad (3.14)$$

donde p_x y p_y son las coordenadas X e Y del punto p , respectivamente.

3.2.3.1.4. Almacenamiento de datos del punto en píxeles Finalmente, los datos de cada punto (coordenadas X, Y, Z y etiqueta de clasificación) se almacenan en el píxel correspondiente de la matriz de imagen:

$$\text{imagen}_{\text{fila}-1, \text{columna}-1} = [p_x, p_y, p_z, \text{clase}] \quad (3.15)$$

Si múltiples puntos se proyectan en el mismo píxel, se conserva el punto con mayor valor de Z .

La Figura 3.7 muestra un ejemplo de una nube de puntos proyectada ortogonalmente, donde se puede apreciar cómo los puntos se mapean a píxeles en la imagen resultante.

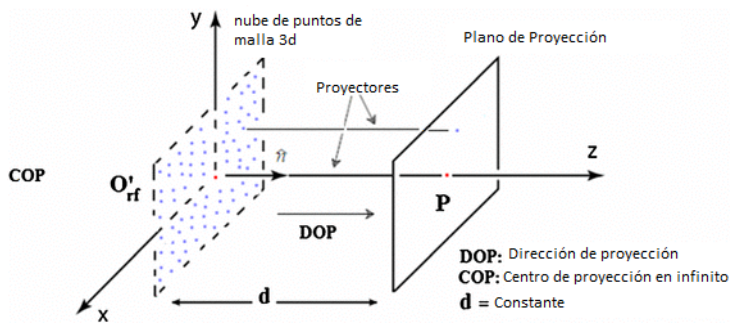


Figura 3.7. Proyección ortogonal de una nube de puntos. Imagen recuperada de (Jovančević y cols., 2017).

La proyección ortogonal es un método sencillo y eficaz para transformar nubes de puntos en imágenes bidimensionales. Sin embargo, en algunos casos, puede haber oclusiones o pérdida de información debido a la naturaleza de la proyección. En las siguientes subsecciones, exploraremos otros métodos de proyección que abordan estas limitaciones y ofrecen diferentes perspectivas de la nube de puntos.

3.2.3.2. Proyección esférica

La proyección esférica es otro método utilizado para transformar nubes de puntos tridimensionales en representaciones bidimensionales.

A diferencia de la proyección ortogonal, que proyecta los puntos perpendicularmente sobre un plano, la proyección esférica mapea los puntos sobre una superficie esférica.

Este tipo de proyección es especialmente útil cuando se trabaja con nubes de puntos capturadas por sensores LiDAR, ya que estos sensores a menudo registran los puntos en coordenadas esféricas, con ángulos de azimut y elevación. La Figura 3.8 ilustra el concepto de proyección esférica aplicado a una nube de puntos.

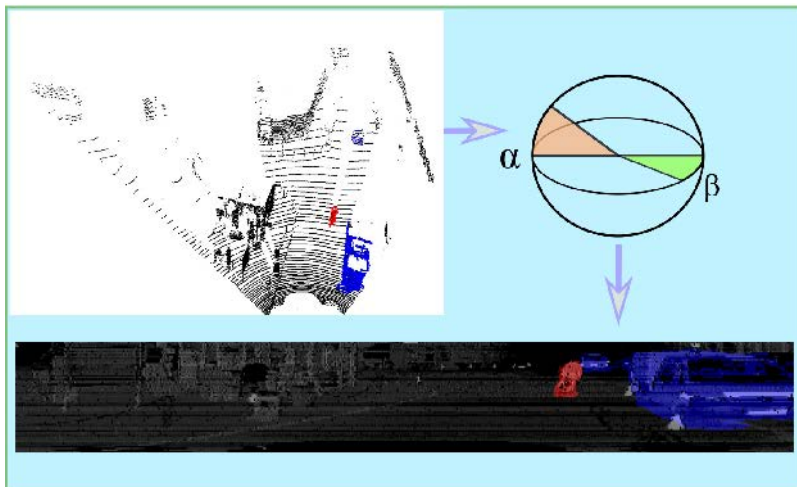


Figura 3.8. Proyección esférica de una nube de puntos.
Imagen recuperada de (Y. Wang y cols., s.f.).

Para aplicar la proyección esférica a una nube de puntos, se siguen los siguientes pasos:

3.2.3.2.1. Proyección esférica y cálculo angular El primer paso es calcular los ángulos de azimut y elevación para cada punto de la nube. Estos ángulos se calculan a partir de las coordenadas cartesianas del

punto:

$$\theta_{\text{horizontal}} = \arctan 2(-y, x) \quad (3.16)$$

$$\theta_{\text{vertical}} = \arctan 2(z, \sqrt{x^2 + y^2}) \quad (3.17)$$

donde x , y y z son las coordenadas del punto en el espacio tridimensional, $\theta_{\text{horizontal}}$ es el ángulo de azimut y θ_{vertical} es el ángulo de elevación.

3.2.3.2.2. Transformación y mapeo de datos Una vez calculados los ángulos de azimut y elevación, estos se utilizan para mapear cada punto a una imagen bidimensional:

$$\text{columna} = \frac{\theta_{\text{horizontal}}}{h_{\text{res_rad}}} \quad (3.18)$$

$$\text{fila} = \frac{\theta_{\text{vertical}}}{v_{\text{res_rad}}} \quad (3.19)$$

donde $h_{\text{res_rad}}$ y $v_{\text{res_rad}}$ son las resoluciones angulares horizontales y verticales, respectivamente, expresadas en radianes.

Los datos de cada punto (coordenadas X, Y, Z, intensidad y distancia) se almacenan en el píxel correspondiente de la matriz de imagen:

$$\text{imagen}[\text{fila} - 1, \text{columna} - 1] = [x, y, z, \text{intensidad}, \text{distancia}] \quad (3.20)$$

Si múltiples puntos se mapean al mismo píxel, se conserva el punto con menor distancia al sensor.

3.2.3.2.3. Corrección de orientación y perspectiva Después de la proyección esférica, puede ser necesario ajustar la orientación y la perspectiva de la imagen resultante para obtener una vista más intuitiva. Esto implica aplicar una transformación de inversión vertical a la matriz de imagen:

$$\text{imagen}'[i, j] = \text{imagen}[m - 1 - i, j] \quad (3.21)$$

donde m es el número de filas de la imagen.

Las Figuras 3.9 y 3.10 muestran ejemplos de una nube de puntos proyectada esféricamente, con los valores de clase y altura normalizados, respectivamente.

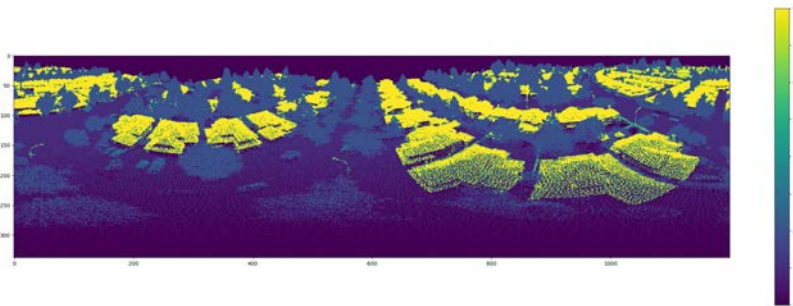


Figura 3.9. Proyección esférica de una nube de puntos con valores de clase. Elaboración propia.

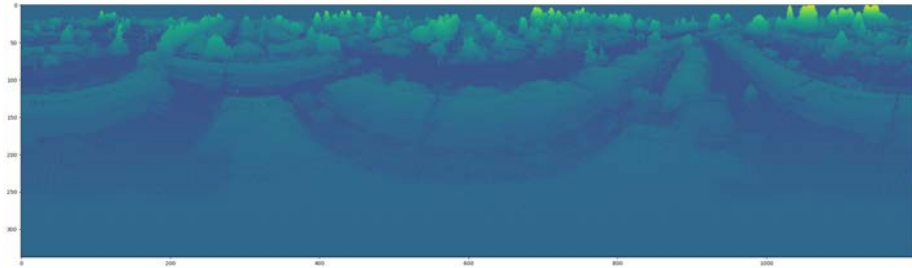


Figura 3.10. Proyección esférica de una nube de puntos con valores de altura normalizados. Elaboración propia.

La proyección esférica ofrece una representación intuitiva de las nubes de puntos capturadas por sensores LiDAR, ya que preserva la geometría angular de los puntos. Sin embargo, esta proyección puede introducir distorsiones en las distancias y formas de los objetos, especialmente en los bordes de la imagen.

En la siguiente subsección, exploraremos la proyección multi-vista, que combina múltiples proyecciones para obtener una representación más completa de la nube de puntos.

3.2.3.3. Proyección multi-vista

La proyección multi-vista es una técnica que combina múltiples proyecciones de una nube de puntos para obtener una representación más completa y detallada de la escena. A diferencia de las proyecciones ortogonal y esférica, que generan una única imagen bidimensional, la proyección multi-vista produce un conjunto de imágenes desde diferentes perspectivas.

La idea principal detrás de la proyección multi-vista es capturar diferentes aspectos de la nube de puntos, como la geometría, la textura y las oclusiones, mediante la generación de múltiples vistas

complementarias. Estas vistas pueden ser ortogonales, como las proyecciones en los planos XY, YZ y ZX, o pueden ser perspectivas, simulando la vista desde diferentes ángulos y posiciones.

La Figura 3.11 muestra un ejemplo de proyección multi-vista ortogonal de una nube de puntos, donde se generan tres imágenes correspondientes a las vistas frontal, lateral y superior.

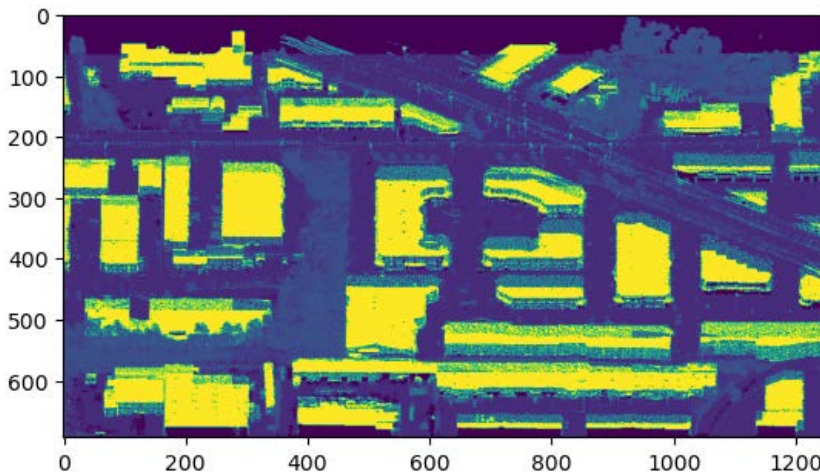


Figura 3.11. Proyección multi-vista ortogonal de una nube de puntos (altura normalizada). Elaboración propia.

Para aplicar la proyección multi-vista a una nube de puntos, se emplean los siguientes conceptos:

3.2.3.3.1. Matriz de rotación a partir de los ángulos de Euler

El primer paso es definir las matrices de rotación que se utilizarán para generar las diferentes vistas. Estas matrices se calculan a partir de los ángulos de Euler *yaw*, *pitch*, and *roll* con ángulos α , β , γ , utilizando las siguientes fórmulas (Jähne, 2005):

$$R_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (3.23)$$

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix} \quad (3.24)$$

La matriz de rotación completa se obtiene multiplicando las matrices individuales:

$$R = R_z(\alpha) \times R_y(\beta) \times R_x(\gamma) \quad (3.25)$$

La Figura 3.12 ilustra el efecto de los ángulos de Euler en la rotación de un objeto.

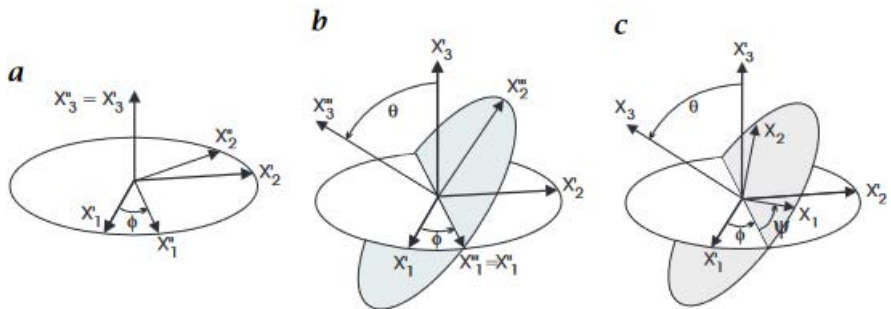


Figura 3.12. Rotación de un objeto según los ángulos de Euler. Imagen recuperada de (Jähne, 2005).

3.2.3.3.2. Matriz de transformación A continuación, se define la matriz de transformación que combina la rotación y la traslación:

$$T = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} \quad (3.26)$$

donde R es la matriz de rotación y \mathbf{t} es el vector de traslación que define el desplazamiento de la nube de puntos.

3.2.3.3.3. Proyección perspectiva Para generar vistas perspectivas, se aplica una proyección perspectiva a los puntos transformados. La proyección perspectiva mapea los puntos tridimensionales (x, y, z) a coordenadas bidimensionales (x', y') utilizando las siguientes fórmulas:

$$x' = \frac{fx \cdot x}{z} + cx \quad (3.27)$$

$$y' = \frac{fy \cdot y}{z} + cy \quad (3.28)$$

donde fx y fy son las distancias focales en los ejes X e Y, respectivamente, y cx y cy son los puntos principales que definen el centro de la proyección.

3.2.3.3.4. Proyección ortogonal Para generar vistas ortogonales, se aplica una proyección ortogonal a los puntos transformados. La proyección ortogonal simplemente descarta la coordenada Z y mapea los puntos tridimensionales (x, y, z) a coordenadas bidimensionales (x', y') :

$$x' = x \tag{3.29}$$

$$y' = y \tag{3.30}$$

3.2.3.3.5. Generación de imágenes Finalmente, se generan las imágenes correspondientes a cada vista proyectada. Cada imagen se inicializa como una matriz de ceros y se mapean los puntos proyectados a los píxeles correspondientes, de manera similar a las proyecciones ortogonal y esférica descritas anteriormente.

La proyección multi-vista ofrece varias ventajas en comparación con las proyecciones únicas. Al generar múltiples vistas complementarias, se captura una mayor cantidad de información sobre la geometría y la apariencia de la nube de puntos. Esto puede mejorar la precisión y la robustez de tareas posteriores, como la segmentación y la clasificación (Yan, Song, Kong, y Huang, 2023).

Sin embargo, la proyección multi-vista también presenta algunos desafíos. El costo computacional aumenta con el número de vistas generadas, y la gestión y el procesamiento de múltiples imágenes pueden ser más complejos. Además, la selección óptima de los ángulos de vista y las posiciones de la cámara puede requerir un ajuste cuidadoso según la geometría de la escena y los objetivos específicos de la tarea (Jovančević y cols., 2017).

En resumen, la proyección multi-vista es una técnica poderosa para generar representaciones bidimensionales enriquecidas de nubes de puntos tridimensionales. Al combinar múltiples vistas complementarias, se captura una mayor cantidad de información sobre la geometría y la apariencia de la escena, lo que puede mejorar el rendimiento de tareas posteriores de análisis y procesamiento.

En la siguiente sección, exploraremos cómo las imágenes generadas a partir de las proyecciones de nubes de puntos se utilizan en el entrenamiento de modelos de aprendizaje profundo, específicamente la arquitectura U-Net, para la segmentación semántica y la clasificación de puntos.

3.3. Entrenamiento y Posprocesamiento

3.3.1. Introducción al entrenamiento y posprocesamiento

El entrenamiento y el posprocesamiento son dos etapas fundamentales en el flujo de trabajo del aprendizaje automático. El entrenamiento es el proceso mediante el cual un modelo aprende a partir de los datos, ajustando sus parámetros internos para optimizar su rendimiento en una tarea específica. Por otro lado, el posprocesamiento se refiere a las técnicas aplicadas después de que el modelo ha generado sus predicciones, con el objetivo de mejorar la calidad y la interpretabilidad de los resultados.

En el contexto de la segmentación semántica de imágenes, el entrenamiento implica alimentar el modelo con imágenes etiquetadas y ajustar sus parámetros para que aprenda a asignar cada píxel a la clase correspondiente. Esto se logra mediante la minimización de una función de pérdida que mide la discrepancia entre las predicciones del modelo y las etiquetas reales.

Después del entrenamiento, las técnicas de posprocesamiento se utilizan para refinar las predicciones del modelo. Estas técnicas pueden incluir la aplicación de umbrales, la eliminación de ruido, la suavización de bordes y la incorporación de conocimiento del dominio para mejorar la coherencia y la precisión de la segmentación.

En las siguientes secciones, exploraremos en detalle la arquitectura U-Net, una potente red neuronal convolucional ampliamente utilizada para la segmentación semántica, y discutiremos las técnicas de entrenamiento y posprocesamiento asociadas.

3.3.2. Arquitectura U-Net

U-Net es una arquitectura de red neuronal convolucional diseñada específicamente para la segmentación de imágenes. Fue introducida por Ronneberger et al. (Ronneberger y cols., 2015a) en 2015 y ha ganado gran popularidad debido a su capacidad para producir resultados precisos con un número relativamente pequeño de imágenes de entrenamiento.

3.3.2.1. Estructura de la arquitectura

La arquitectura U-Net tiene una estructura única en forma de U , que consta de dos partes principales: el codificador (encoder) y el decodificador (decoder), conectados por un puente (bridge) en la parte inferior de la U (ver Figura 3.13).

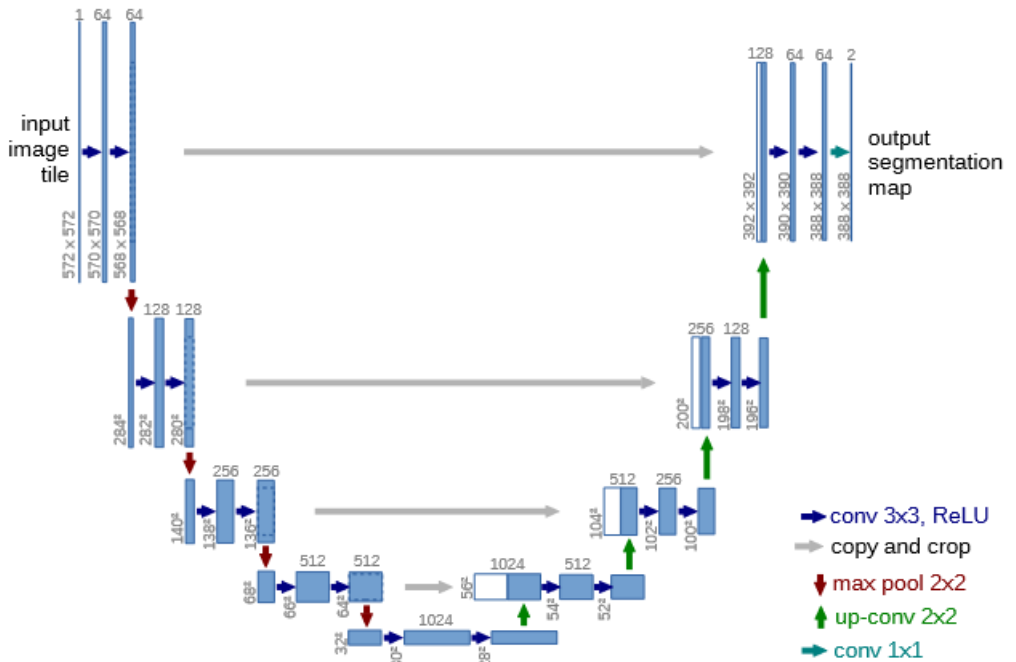


Figura 3.13. Arquitectura de U-Net. Imagen adaptada de (Ronneberger y cols., 2015a).

El codificador es la parte izquierda de la arquitectura y su objetivo es capturar el contexto y las características relevantes de la imagen de entrada. Consiste en una serie de bloques convolucionales seguidos de operaciones de max pooling. Cada bloque convolucional típicamente contiene dos capas convolucionales con activación ReLU, que introducen no linealidad en la red. La operación de max pooling reduce la resolución espacial de las características y ayuda a aumentar el campo receptivo de la red.

El puente es la capa más profunda de U-Net y conecta el codificador con el decodificador. Esta capa captura las características

más abstractas y de alto nivel de la imagen de entrada.

El decodificador es la parte derecha de la arquitectura y su objetivo es recuperar la resolución espacial y generar la máscara de segmentación final. Consiste en una serie de bloques de upsampling seguidos de capas convolucionales. Cada bloque de upsampling aumenta la resolución espacial de las características y luego se concatena con las características correspondientes del codificador a través de conexiones de salto (skip connections). Estas conexiones permiten la propagación de información de alta resolución desde el codificador hasta el decodificador, lo que ayuda a preservar los detalles espaciales en la máscara de segmentación final.

3.3.2.2. Convolución y función de activación

Las capas convolucionales son el componente principal de la arquitectura U-Net. La convolución es una operación matemática que aplica un filtro (kernel) a una imagen para extraer características relevantes. En U-Net, se utilizan convoluciones 2D con kernels de tamaño 3x3 o 5x5.

Después de cada capa convolucional, se aplica una función de activación no lineal para introducir no linealidad en la red. La función de activación más comúnmente utilizada en U-Net es la unidad lineal rectificadora (ReLU), que se define como:

$$ReLU(x) = \max(0, x) \quad (3.31)$$

La ReLU permite que la red aprenda características más complejas y ayuda a evitar el problema del desvanecimiento del gradiente.

3.3.2.3. Upsampling y skip connections

En el decodificador de U-Net, se utilizan operaciones de upsampling para aumentar la resolución espacial de las características. El upsampling se puede realizar mediante convoluciones transpuestas (deconvoluciones) o interpolación bilineal seguida de una convolución.

Una característica única de U-Net son las skip connections, que conectan el codificador con el decodificador. Estas conexiones concatenan las características del codificador con las características correspondientes del decodificador, lo que permite la propagación de información de alta resolución y ayuda a preservar los detalles espaciales en la máscara de segmentación final.

3.3.2.4. Función de pérdida y optimización

Durante el entrenamiento de U-Net, se utiliza una función de pérdida para medir la discrepancia entre las máscaras de segmentación predichas y las máscaras de segmentación reales (ground truth). La función de pérdida más comúnmente utilizada es la entropía cruzada binaria o categórica, dependiendo de si se trata de un problema de segmentación binaria o multiclase (Goodfellow, Bengio, y Courville, 2016).

El objetivo durante el entrenamiento es minimizar la función de pérdida ajustando los pesos de la red mediante un algoritmo de optimización, como el descenso de gradiente estocástico (SGD) o Adam.

3.3.2.5. Ventajas y limitaciones

U-Net tiene varias ventajas que la hacen atractiva para tareas de segmentación de imágenes:

1. Requiere relativamente pocas imágenes de entrenamiento para lograr resultados precisos, gracias a su arquitectura eficiente y al uso de skip connections.
2. Puede manejar imágenes de diferentes tamaños debido a su estructura completamente convolucional.
3. Es capaz de producir máscaras de segmentación precisas con detalles espaciales bien preservados.

Sin embargo, U-Net también tiene algunas limitaciones:

1. Puede ser computacionalmente costosa, especialmente para imágenes de alta resolución, debido a su estructura de codificador-decodificador.
2. Requiere un ajuste cuidadoso de los hiperparámetros, como la tasa de aprendizaje y el tamaño del batch, para obtener los mejores resultados.
3. Puede tener dificultades para segmentar objetos muy pequeños o detalles finos, especialmente si no están bien representados en el conjunto de entrenamiento.

En resumen, U-Net es una poderosa arquitectura de red neuronal convolucional diseñada específicamente para la segmentación de imágenes. Su estructura única en forma de U , que consta de un codificador, un puente y un decodificador, junto con las skip connections, permite la captura de características a diferentes escalas y la generación de máscaras de segmentación precisas. Aunque tiene algunas limitaciones, U-Net ha demostrado ser altamente efectiva en una amplia gama de tareas de segmentación y sigue siendo una de las arquitecturas más populares en este campo.

3.3.3. Optimizadores

Los optimizadores son algoritmos utilizados para ajustar los parámetros de una red neuronal durante el entrenamiento, con el objetivo de minimizar la función de pérdida. Estos algoritmos actualizan los pesos de la red en función del gradiente de la función de pérdida con respecto a cada parámetro.

3.3.3.1. Importancia de los optimizadores

La elección del optimizador adecuado es crucial para el rendimiento y la convergencia de la red neuronal. Un buen optimizador debe ser capaz de:

- Converger rápidamente hacia un mínimo de la función de pérdida.
- Evitar quedar atrapado en mínimos locales subóptimos.
- Ser computacionalmente eficiente y escalar bien a grandes conjuntos de datos y arquitecturas profundas.

Algunos de los optimizadores más populares incluyen el descenso de gradiente estocástico (SGD), el optimizador Adam y RMSprop.

3.3.3.2. Optimizador Adam

Adam (Adaptive Moment Estimation) (Kingma y Ba, 2017) es uno de los optimizadores más ampliamente utilizados en el aprendizaje profundo. Combina las ventajas de otros dos optimizadores populares: AdaGrad y RMSprop.

Adam adapta las tasas de aprendizaje para cada parámetro individual en función de las estimaciones de los momentos de primer y segundo orden de los gradientes. Esto permite que el optimizador converja rápidamente y se adapte a diferentes escalas de características.

La actualización de los parámetros en Adam se realiza de la siguiente manera:

Primero, se calculan las estimaciones de los momentos de primer y segundo orden:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (3.32)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (3.33)$$

donde g_t es el gradiente en la iteración t , y β_1 y β_2 son hiperparámetros que controlan las tasas de decaimiento exponencial para los momentos.

Luego, se calculan las estimaciones corregidas de los momentos para contrarrestar el sesgo inicial:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (3.34)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (3.35)$$

Finalmente, se actualizan los parámetros utilizando las estimaciones corregidas de los momentos:

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (3.36)$$

donde α es la tasa de aprendizaje y ϵ es un pequeño valor para evitar la división por cero.

Adam tiene varias ventajas, como la adaptación automática de las tasas de aprendizaje, la eficiencia computacional y la robustez frente a gradientes ruidosos. Sin embargo, también puede presentar algunos desafíos, como la sensibilidad a la elección de los hiperparámetros y la posibilidad de no converger a un óptimo global en algunos casos.

3.3.3.3. Otros optimizadores

Además de Adam, existen otros optimizadores populares en el aprendizaje profundo (Goodfellow y cols., 2016), como:

1. Descenso de gradiente estocástico (SGD): Actualiza los parámetros en la dirección opuesta al gradiente de la función de pérdida, utilizando un subconjunto aleatorio de los datos en cada iteración.
2. RMSprop: Adapta las tasas de aprendizaje para cada parámetro en función de la media móvil de los gradientes al cuadrado, lo que lo hace adecuado para problemas con gradientes ruidosos.
3. AdaGrad: Adapta las tasas de aprendizaje para cada parámetro en función de la suma histórica de los gradientes al cuadrado, lo que lo hace adecuado para problemas con características escasas.

La elección del optimizador depende del problema específico, la arquitectura de la red neuronal y los recursos computacionales disponibles. En muchos casos, Adam es una buena opción por defecto debido a su robustez y eficiencia. Sin embargo, puede ser beneficioso experimentar con diferentes optimizadores y ajustar los hiperparámetros para obtener el mejor rendimiento en una tarea determinada.

En resumen, los optimizadores desempeñan un papel crucial en el entrenamiento de redes neuronales profundas, ya que determinan cómo se ajustan los parámetros del modelo en función de la señal de error proporcionada por la función de pérdida. Adam es uno de los optimizadores más populares y eficaces, pero existen otros optimizadores que pueden ser más adecuados para ciertos problemas. La elección del optimizador y el ajuste de sus hiperparámetros pueden

tener un impacto significativo en el rendimiento y la convergencia de la red neuronal.

3.3.4. Técnicas de posprocesamiento

Una vez completada la segmentación semántica con la arquitectura U-Net, un desafío crítico es la correcta propagación de estas etiquetas a través de la nube de puntos 3D original. Dado que las proyecciones empleadas para la segmentación pueden llevar a una compresión espacial, numerosos puntos en la nube de puntos 3D pueden terminar sin etiquetas asignadas o con etiquetas incorrectas. Para abordar este problema y mejorar la consistencia de las etiquetas en toda la nube de puntos, aplicamos técnicas de posprocesamiento, siendo *Label Spreading* el procesamiento elegido.

3.3.5. Mejorando la segmentación con Label Spreading

Después de aplicar la arquitectura U-Net para la segmentación semántica de las imágenes proyectadas, es necesario propagar las etiquetas resultantes a la nube de puntos 3D original. Debido a la compresión de información durante la proyección, algunos puntos pueden quedar sin una etiqueta asignada o con una etiqueta incorrecta. Para abordar este problema y mejorar la consistencia de la segmentación, utilizamos la técnica de Label Spreading como un paso de posprocesamiento.

Label Spreading (D. Zhou, Bousquet, Lal, Weston, y Schölkopf, 2003) es un algoritmo semi-supervisado que propaga las etiquetas de los puntos con alta confianza a los puntos vecinos no etiquetados o con baja confianza. Este proceso se basa en la suposición de que los puntos cercanos y similares tienen más probabilidades de compartir la misma

etiqueta.

3.3.5.1. Funcionamiento de Label Spreading

El algoritmo de Label Spreading construye un grafo de similitud entre los puntos de la nube, donde cada punto está conectado a sus vecinos más cercanos. Luego, las etiquetas se propagan iterativamente a través de este grafo hasta alcanzar una condición de convergencia.

Formalmente, dado un conjunto de puntos $\mathcal{X} = \{x_1, \dots, x_n\}$, donde $\mathcal{X}_l = \{x_1, \dots, x_l\}$ son los puntos etiquetados y $\mathcal{X}_u = \{x_{l+1}, \dots, x_n\}$ son los puntos no etiquetados, el objetivo es asignar etiquetas a \mathcal{X}_u utilizando la información proporcionada por \mathcal{X}_l .

El proceso de Label Spreading se puede resumir en los siguientes pasos:

1. Construir la matriz de afinidad W , donde W_{ij} representa la similitud entre los puntos x_i y x_j , utilizando una función de similitud como el kernel gaussiano:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (3.37)$$

2. Normalizar la matriz de afinidad para obtener la matriz de transición T :

$$T = D^{-1}W \quad (3.38)$$

donde D es una matriz diagonal con $D_{ii} = \sum_{j=1}^n W_{ij}$.

3. Inicializar la matriz de etiquetas Y , donde $Y_{ij} = 1$ si el punto x_i tiene la etiqueta j , y 0 en caso contrario. Para los puntos no etiquetados, se asignan valores iniciales de 0.

4. Propagar las etiquetas mediante la siguiente fórmula iterativa:

$$Y^{(t+1)} = \alpha TY^{(t)} + (1 - \alpha)Y^{(0)} \quad (3.39)$$

donde $\alpha \in [0, 1]$ es un parámetro que controla la influencia de las etiquetas iniciales y $Y^{(0)}$ es la matriz de etiquetas inicial.

5. Repetir el paso 4 hasta que Y converja o se alcance un número máximo de iteraciones.
6. Asignar a cada punto no etiquetado $x_i \in \mathcal{X}_u$ la etiqueta correspondiente al elemento máximo en la fila i de la matriz Y final.

3.3.5.2. Ventajas y limitaciones

Label Spreading presenta varias ventajas para mejorar la segmentación de nubes de puntos:

- Aprovecha la información espacial y la similitud entre los puntos para propagar las etiquetas de manera consistente.
- Puede manejar datos no etiquetados y mejorar la segmentación en regiones donde la red neuronal puede tener dificultades.
- Es relativamente eficiente en términos de computación y puede aplicarse a nubes de puntos de gran tamaño.

Sin embargo, Label Spreading también tiene algunas limitaciones:

- La calidad de los resultados depende de la elección adecuada de la función de similitud y los parámetros del algoritmo.
- Puede tener dificultades para manejar datos con distribuciones complejas o superpuestas.

- La propagación de etiquetas puede verse afectada por ruido o errores en las etiquetas iniciales.

A pesar de estas limitaciones, Label Spreading es una técnica valiosa para mejorar la segmentación de nubes de puntos y asegurar que cada punto tenga una etiqueta asignada. Al combinar la segmentación inicial obtenida por la red U-Net con la propagación de etiquetas mediante Label Spreading, podemos obtener una segmentación más consistente y completa de la nube de puntos 3D.

En resumen, Label Spreading es un paso de posprocesamiento crucial en nuestra metodología de segmentación de nubes de puntos. Al propagar las etiquetas de los puntos con alta confianza a los puntos vecinos no etiquetados o con baja confianza, mejoramos la consistencia y la completitud de la segmentación. Aunque tiene algunas limitaciones, Label Spreading aprovecha la información espacial y la similitud entre los puntos para refinar los resultados de la segmentación inicial obtenida por la red U-Net.

3.3.6. Resumen y próximos pasos

A lo largo de este capítulo, hemos presentado una metodología integral para abordar el desafío de segmentar y clasificar nubes de puntos 3D. Nuestra propuesta combina técnicas de preprocesamiento, como la normalización de altura y la alineación de datos, con métodos avanzados de proyección para transformar las nubes de puntos en representaciones 2D adecuadas para el análisis mediante redes neuronales convolucionales.

En el núcleo de nuestra metodología se encuentra la arquitectura U-Net, una poderosa red neuronal diseñada específicamente para la segmentación de imágenes. Hemos explorado en detalle la estructura y

los componentes clave de U-Net, así como las técnicas de entrenamiento y optimización asociadas, como la función de pérdida de entropía cruzada y el optimizador Adam.

Además, hemos destacado la importancia del posprocesamiento para refinar los resultados de la segmentación. En particular, nos hemos centrado en la técnica de Label Spreading, un algoritmo semi-supervisado que propaga las etiquetas de los puntos con alta confianza a los puntos vecinos no etiquetados o con baja confianza. Este enfoque aprovecha la información espacial y la similitud entre los puntos para mejorar la consistencia y la completitud de la segmentación final.

En el próximo capítulo, nos centraremos en la validación experimental de nuestra propuesta. A través de una serie de experimentos rigurosos en conjuntos de datos seleccionados, evaluaremos la eficacia de nuestra metodología y compararemos nuestros resultados con el estado del arte en segmentación de nubes de puntos 3D.

Capítulo 4

Experimentos y Resultados

En este capítulo se presenta una evaluación exhaustiva de tres técnicas de proyección aplicadas a la segmentación semántica de nubes de puntos: proyección ortogonal, proyección ortogonal multivista y proyección esférica. Se describe detalladamente el conjunto de datos utilizado, la metodología experimental empleada y se analizan en profundidad los resultados obtenidos. El objetivo es comparar el rendimiento de estas técnicas y extraer conclusiones sobre su eficacia en la segmentación semántica de nubes de puntos.

4.1. Conjunto de datos

Para llevar a cabo nuestros experimentos, se seleccionó el dataset "DALES" (Varney, Asari, y Graehling, 2020), ampliamente reconocido por su robustez y confiabilidad en el ámbito de las nubes de puntos. DALES destaca como el conjunto de datos LiDAR aéreo públicamente disponible más extenso y denso para tareas de segmentación semántica. Lanzado en 2020 por el laboratorio de visión de la

Universidad de Dayton, este dataset contiene 40 escenas de datos aéreos densos etiquetados manualmente en ocho categorías, abarcando diversos tipos de entornos como áreas urbanas, suburbanas, rurales y comerciales. Con una cantidad y resolución de puntos significativamente superior a otros datasets similares, DALES se posiciona como un recurso valioso para el desarrollo y evaluación de algoritmos de aprendizaje profundo aplicados a la segmentación semántica de nubes de puntos LiDAR.

4.1.1. Recolección de Datos para el Conjunto de Datos DALES

El proceso de adquisición de datos para DALES se llevó a cabo empleando el sistema LiDAR aéreo Riegl Q1560 de doble canal, montado en una aeronave Piper PA31. Los datos fueron proyectados en el sistema de coordenadas UTM 10N, con datum horizontal NAD83 y vertical CGVD28, utilizando el geoide de Vancouver. Se emplearon receptores Trimble R10 GNSS en tierra para recopilar puntos de control con una precisión de 1 a 2 cm. La elección de la ciudad de Dayton, Ohio, como área de escaneo LiDAR fue intencional, dado que su diversidad paisajística, combinando complejidad urbana con variaciones topográficas, permite obtener datos representativos de diferentes contextos de segmentación (Varney y cols., 2020).

Figura 4.1. Imagen representa una parte de Dayton, Ohio



Fuente: Imagen recuperada de <https://www.daytonlocal.com/blog/community/25-must-dos-around-dayton-ohio.asp>

Cada área fue escaneada con una densidad mínima de 20 puntos por metro cuadrado, registrando al menos 5 pulsos láser por metro en las direcciones cardinales para reducir oclusiones. La precisión fue evaluada comparando con los puntos de control terrestres y mediante inspección visual, determinando un error de ± 8.5 cm con un 95% de confianza en la precisión vertical sobre superficies duras. Esta rigurosidad en la recolección de datos sienta las bases para los posteriores procesos de preprocesamiento y anotación.

A pesar de que el conjunto de datos se limita a un solo municipio, ofrece una mezcla rica de paisajes, reduciendo el riesgo de sobreajuste (Varney y cols., 2020). Categorizaron las escenas en cuatro tipos, basándose en la naturaleza y densidad de las estructuras presentes:

- Comercial: almacenes y parques de oficinas
- Urbano: edificios de gran altura, mayores de cuatro pisos

- Rural: objetos naturales con algunos edificios dispersos
- Suburbano: concentración de viviendas unifamiliares

4.1.2. Preprocesamiento y Anotación

Dado que el LiDAR aéreo puede presentar ruido disperso en la nube de puntos debido a efectos atmosféricos, superficies reflectantes o errores de medición, se aplicó un filtro estadístico de eliminación de valores atípicos para descartar puntos aislados. Además, para facilitar su gestión y análisis, la nube fue dividida en mosaicos de 0.5 km², considerando factores como la topografía y urbanización para asegurar que los datos reflejen fielmente el paisaje real. Este preprocesamiento garantiza la calidad de los datos y los prepara para la crucial etapa de anotación.

La anotación de los datos fue un proceso meticuloso que involucró tanto etiquetado automático como manual. Inicialmente, se etiquetaron los puntos cercanos al Modelo Digital de Elevación (DEM) como puntos de suelo. Luego, se calcularon normales superficiales para identificar objetos planos, probablemente edificios, que fueron refinados manualmente. El resto de objetos fueron etiquetados manualmente por anotadores expertos, empezando por los de mayor tamaño. Tras la anotación inicial, al menos dos anotadores diferentes revisaron cada escena para garantizar la calidad y consistencia de las etiquetas.

Se definieron ocho categorías de objetos: suelo, vegetación, automóviles, camiones, líneas eléctricas, postes, vallas y edificios. También se incluyó una categoría "desconocida" para objetos etiquetados manualmente pero sin suficientes puntos para ser clasificados eficazmente por algoritmos de aprendizaje profundo. Esta categorización exhaustiva permite una segmentación semántica

detallada y sienta las bases para la evaluación del rendimiento de los modelos.

4.1.3. Métricas

En la evaluación de modelos para segmentación de nubes de puntos, las métricas estándar como el IoU (Intersección sobre Unión) promedio y la precisión general (Overall Accuracy, OA) son ampliamente utilizadas. Estas métricas proporcionan una base para comparar el rendimiento de diferentes enfoques en tareas de clasificación y segmentación.

Intersección sobre Unión (IoU)

El IoU, una métrica clave para la evaluación de la precisión de segmentación, se define por clase y luego se promedia para obtener una medida global del rendimiento. Por cada clase i , el IoU se calcula como:

$$IoU_i = \frac{c_{ii}}{c_{ii} + \sum_{j \neq i} c_{ij} + \sum_{k \neq i} c_{ki}} \quad (4.1)$$

donde c_{ii} representa el número de verdaderos positivos para la clase i , c_{ij} el número de falsos positivos, y c_{ki} el número de falsos negativos. El IoU promedio, excluyendo categorías desconocidas, se obtiene a través del promedio de IoU para todas las categorías relevantes:

$$IoU = \frac{\sum_{i=1}^N IoU_i}{N} \quad (4.2)$$

Precisión General (OA)

Aunque el IoU proporciona una medida detallada del rendimiento de la segmentación, la precisión general ofrece una visión global del

rendimiento del modelo. La OA se calcula sumando todos los verdaderos positivos y dividiéndolos por el total de muestras:

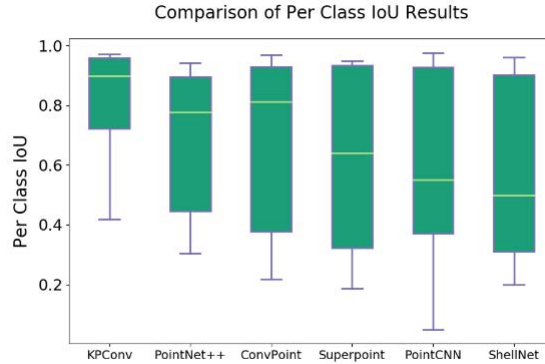
$$OA = \frac{\sum_{i=1}^N c_{ii}}{\sum_{j=1}^N \sum_{k=1}^N c_{jk}} \quad (4.3)$$

Aquí, c_{ii} son los verdaderos positivos para cada clase, y c_{jk} representa el total de predicciones hechas, incluyendo tanto predicciones correctas como incorrectas, para todas las clases.

Estas métricas de evaluación son fundamentales para entender el rendimiento de los modelos en la segmentación de nubes de puntos, permitiendo a los investigadores identificar fortalezas y debilidades en las técnicas de clasificación y segmentación aplicadas.

De la misma manera, los investigadores evalúan los resultados en términos de sus gráficos de caja y bigote y observan la distribución entre el cuartil inferior y superior, como se puede observar en la figura 4.2. Al evaluar el éxito de un algoritmo, buscan asegurarse de que los resultados tengan tanto un alto IoU promedio como una baja desviación estándar en todas las clases. Examinan los cuartiles inferior y superior como una medida de la robustez en el rendimiento de un método.

Figura 4.2. Algoritmo por distribución de rendimiento de IoU de clase



Fuente: Imagen recuperada de (Varney y cols., 2020)

Finalmente, establecen una métrica llamada Índice de Consistencia de Clase (CCI, por sus siglas en inglés), que definen como el complemento de la varianza dentro de la clase sobre el IoU promedio, mostrado a continuación:

$$CCI = 1 - \frac{\sigma^2}{|IoU|} \quad (4.4)$$

También examinan el CCI versus el IoU promedio en forma de un Mapa de Rendimiento del Algoritmo. Un algoritmo robusto tiene tanto un alto IoU promedio como un alto CCI, indicando que no solo tiene un alto rendimiento, sino que el rendimiento es uniforme en cada clase.

4.1.4. Benchmarks del Conjunto de Datos Dales

La verdadera validez y robustez de cualquier conjunto de datos se revela a través de su evaluación y benchmarking. Es esencial someter

los datos a pruebas rigurosas para determinar su aplicabilidad, precisión y relevancia en escenarios del mundo real. DALES, con su enfoque en la calidad y precisión, fue sometido a una evaluación exhaustiva para determinar su valor en el campo del análisis de datos LiDAR.

Para evaluar la validez y robustez del dataset DALES, se seleccionaron seis algoritmos de referencia destacados en benchmarks similares de MLS y TLS, con énfasis en métodos de aprendizaje profundo sobresalientes en Semantic 3D y Paris-Lille-3D. Las redes elegidas fueron PointNet++, KPConv, PointCNN, ConvPoint, ShellNet y Superpoint Graphs (Varney y cols., 2020). La tabla 4.1 presenta las redes y su desempeño en el conjunto de datos DALES:

Cuadro 4.1. Descripción general de los métodos seleccionados en el conjunto de datos DALES.

| Algoritmo | Precision | Promedio IoU | Suelo | Edificios | Coches | Camiones | Postes | Lineas de Luz | Cercas | Vegetacion |
|------------|-----------|--------------|-------|-----------|--------|----------|--------|---------------|--------|------------|
| KPConv | 0.978 | 0.811 | 0.971 | 0.966 | 0.853 | 0.419 | 0.75 | 0.955 | 0.635 | 0.941 |
| PointNet++ | 0.957 | 0.683 | 0.941 | 0.891 | 0.754 | 0.303 | 0.4 | 0.799 | 0.462 | 0.912 |
| ConvPoint | 0.972 | 0.674 | 0.969 | 0.963 | 0.755 | 0.217 | 0.403 | 0.867 | 0.296 | 0.919 |
| SuperPoint | 0.955 | 0.606 | 0.947 | 0.934 | 0.629 | 0.187 | 0.285 | 0.652 | 0.336 | 0.879 |
| PointCNN | 0.972 | 0.584 | 0.975 | 0.957 | 0.406 | 0.048 | 0.576 | 0.267 | 0.526 | 0.917 |
| ShellNet | 0.964 | 0.574 | 0.96 | 0.954 | 0.322 | 0.396 | 0.2 | 0.274 | 0.6 | 0.884 |

Fuente: Imagen recuperada de (Varney y cols., 2020)

Los resultados de esta evaluación fueron esclarecedores. Por ejemplo, el algoritmo KPConv mostró un rendimiento excepcional en varias métricas, destacándose como una herramienta potencialmente valiosa para el análisis de datos LiDAR. La arquitectura KPConv se destacó con un desempeño notablemente fuerte en el conjunto de datos, con un IoU promedio del 81.1 %, más de 10 % superior al de otras redes.

Se observó una tendencia común en las redes. Las categorías de suelo, vegetación y edificios mostraron consistentemente buenos desempeños.

Esto probablemente se deba a la abundancia de puntos y ejemplos de estas categorías en el conjunto de datos. Por otro lado, camiones, vallas y postes tuvieron un IoU mucho más bajo, correlacionando con el menor número de puntos en cada categoría.

Se observaron desempeños consistentemente bajos en la categoría de objeto camión. Esta categoría comprende principalmente camiones y camiones de caja situados en áreas comerciales. Los camiones mostraron una alta correlación de normal superficial tanto con las categorías de suelo como de edificios, que tienen significativamente más ejemplos de entrenamiento. Este problema de distribución explica el desempeño deficiente en esta categoría de objeto en todos los métodos, resaltando un área para mejora potencial.

Estos benchmarks no solo validan la calidad y relevancia del dataset DALES, sino que también proporcionan una base para comparar el rendimiento de diferentes algoritmos de segmentación semántica. Los resultados obtenidos servirán como referencia para evaluar la eficacia de las técnicas de proyección propuestas en nuestros experimentos.

4.2. Resultados

En esta sección se describe el proceso de entrenamiento de la arquitectura U-Net para la segmentación semántica de nubes de puntos utilizando las tres técnicas de proyección mencionadas. Antes del entrenamiento, las imágenes fueron divididas en segmentos de 256x256 píxeles y los valores de los píxeles se normalizaron al rango $[0,1]$. Este preprocesamiento es esencial para que la red neuronal pueda procesar los datos de manera eficiente.

Se utilizó un conjunto de datos de entrenamiento compuesto por

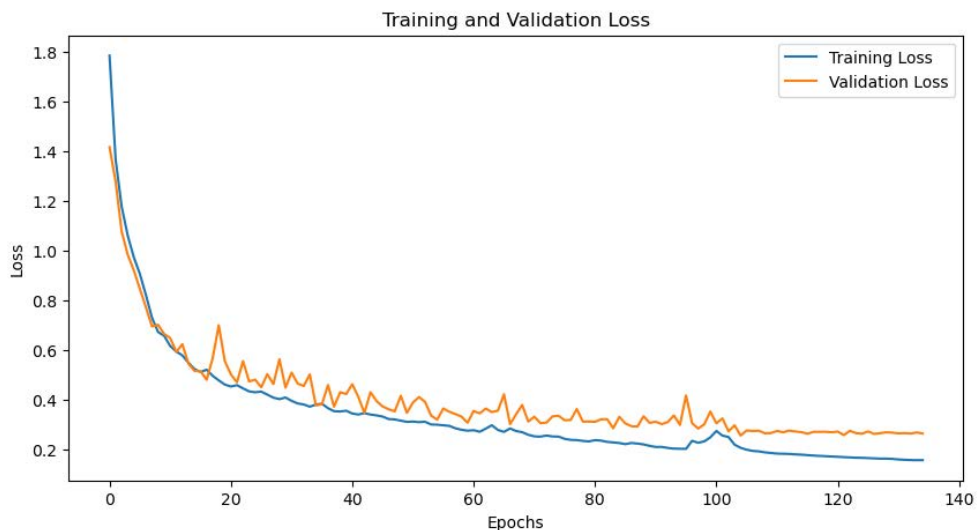
imágenes de nubes de puntos etiquetadas, dividido en subconjuntos de entrenamiento y validación. Se empleó la función de pérdida de entropía cruzada y el optimizador Adam, entrenando durante 1000 épocas con un tamaño de lote de 32 imágenes. La tasa de aprendizaje inicial fue de 0.001, reduciéndose en un 20% a lo largo del entrenamiento. Estos hiperparámetros fueron seleccionados basándose en experiencias previas y mejores prácticas en el campo.

Para evitar el sobreajuste, se implementó early stopping basado en la función de pérdida de validación, deteniendo el entrenamiento si no había mejora durante 15 épocas consecutivas. Durante el proceso, se evaluó el rendimiento en el conjunto de validación utilizando métricas como el índice de Jaccard, la precisión y la IoU media. Estas métricas permiten monitorear el progreso del entrenamiento y realizar ajustes si es necesario.

4.2.0.1. Proyección Ortogonal

El entrenamiento con proyección ortogonal mostró una disminución consistente de la función de pérdida tanto en el conjunto de entrenamiento como en el de validación (Figura 4.3), indicando un aprendizaje efectivo. Los valores del índice de Jaccard, precisión y mIoU en el conjunto de validación fueron altos (Tabla 4.2), sugiriendo una segmentación precisa y completa. Estos resultados preliminares son prometedores y sientan las bases para la posterior validación del modelo.

Figura 4.3. Gráfica de la función de pérdida contra epochs de datos de entrenamiento y validación del entrenamiento de proyección ortogonal



Fuente: Elaboración Propia

Cuadro 4.2. Resultados de métricas de rendimiento de la proyección ortogonal

| | Precisión General | Índice de Jaccard | F1 | Promedio IoU | Función de Pérdida |
|------------------------|-------------------|-------------------|--------|--------------|--------------------|
| Datos de Entrenamiento | 0.9532 | 0.8658 | 0.9272 | 0.4702 | 0.1474 |
| Datos de Validación | 0.9333 | 0.8527 | 0.9178 | 0.4698 | 0.2685 |

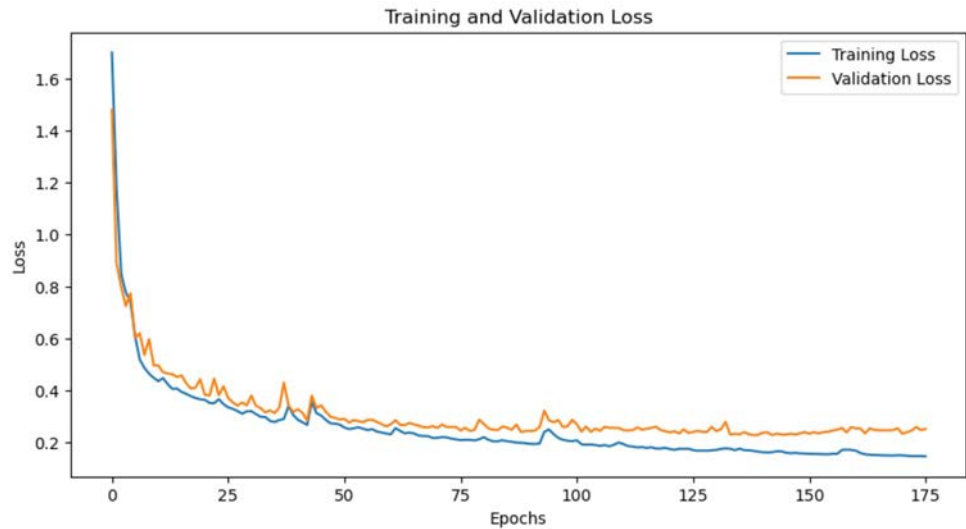
Fuente: Elaboración propia

4.2.0.2. Proyección Ortogonal Multivista

De manera similar, el entrenamiento con proyección ortogonal multivista exhibió una reducción progresiva de la función de pérdida

(Figura 4.4). Las métricas de evaluación en el conjunto de validación (Tabla 4.3) también reflejaron un buen desempeño, aunque ligeramente inferior al de la proyección ortogonal simple. Esto sugiere que la proyección ortogonal multivista puede capturar información adicional, pero también introduce cierta complejidad en el proceso de segmentación.

Figura 4.4. Gráfica de la función de pérdida contra epochs de datos de entranamiento y validación del entranamiento de proyección ortogonal multivista



Fuente: Elaboración Propia

Cuadro 4.3. Resultados de métricas de rendimiento de la proyección ortogonal multivista

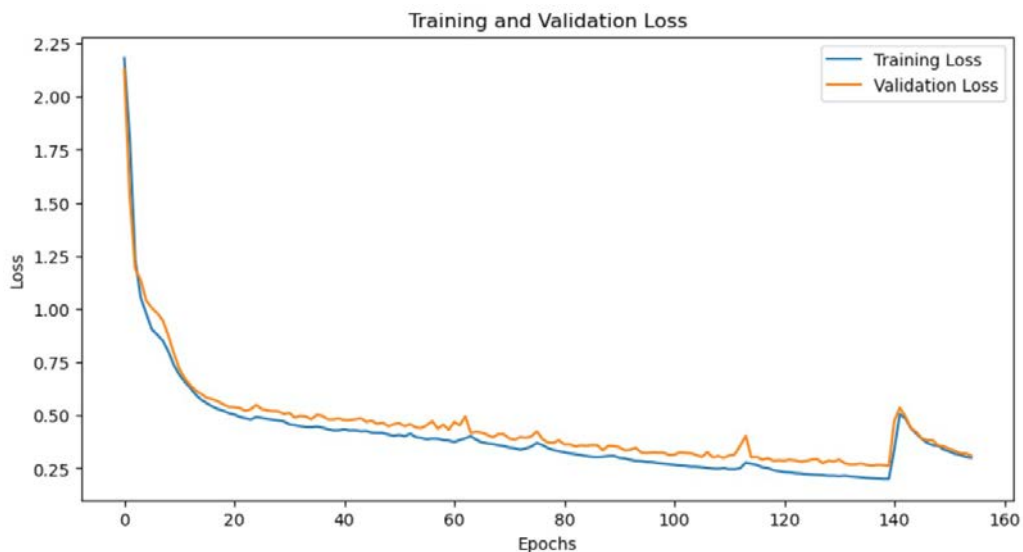
| | Precisión General | Índice de Jaccard | F1 | Promedio IoU | Función de Pérdida |
|------------------------|-------------------|-------------------|--------|--------------|--------------------|
| Datos de Entrenamiento | 0.9496 | 0.8619 | 0.9258 | 0.5675 | 0.2501 |
| Datos de Validación | 0.9340 | 0.8414 | 0.9138 | 0.5653 | 0.1444 |

Fuente: Elaboración propia

4.2.0.3. Proyección Esférica

Por otro lado, el entrenamiento con proyección esférica arrojó resultados mixtos. Si bien la función de pérdida disminuyó (Figura 4.5), los valores de las métricas de evaluación fueron más bajos en comparación con las otras dos técnicas de proyección (Tabla 4.4), sugiriendo ciertos desafíos en la segmentación utilizando esta representación. Estos resultados indican que la proyección esférica puede requerir ajustes adicionales o un enfoque diferente para mejorar su rendimiento.

Figura 4.5. Gráfica de la función de pérdida contra epochs de datos de entranamiento y validación del entranamiento de proyección esférica



Fuente: Elaboración Propia

Cuadro 4.4. Resultados de métricas de rendimiento de la proyección esférica

| | Precisión General | Índice de Jaccard | F1 | Promedio IoU | Función de Pérdida |
|------------------------|-------------------|-------------------|--------|--------------|--------------------|
| Datos de Entrenamiento | 0.8974 | 0.7445 | 0.8501 | 0.6463 | 0.3007 |
| Datos de Validación | 0.8946 | 0.7327 | 0.8437 | 0.6083 | 0.3103 |

Fuente: Elaboración propia

Habiendo analizado el proceso de entrenamiento y los resultados preliminares para cada técnica de proyección, procedemos a la validación de los modelos en un conjunto de datos de prueba independiente. Esta validación nos permitirá evaluar de manera más

completa la eficacia de cada técnica y extraer conclusiones sobre su idoneidad para la segmentación semántica de nubes de puntos.

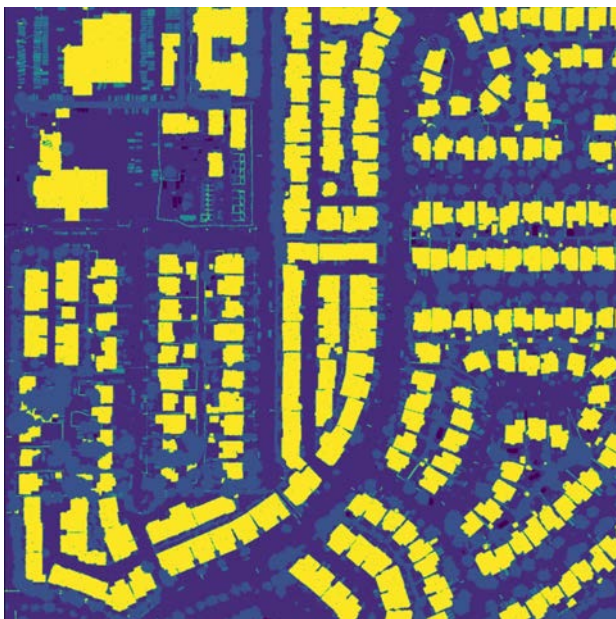
4.2.1. Validación de los Modelos

Tras el entrenamiento, se procedió a validar los modelos en un conjunto de datos de prueba independiente para evaluar su capacidad de generalización y rendimiento en datos no vistos durante el entrenamiento. Los resultados se analizaron tanto cuantitativa como cualitativamente para cada técnica de proyección, permitiendo una comparación exhaustiva de su eficacia en la segmentación semántica de nubes de puntos.

4.2.1.1. Proyección Ortogonal

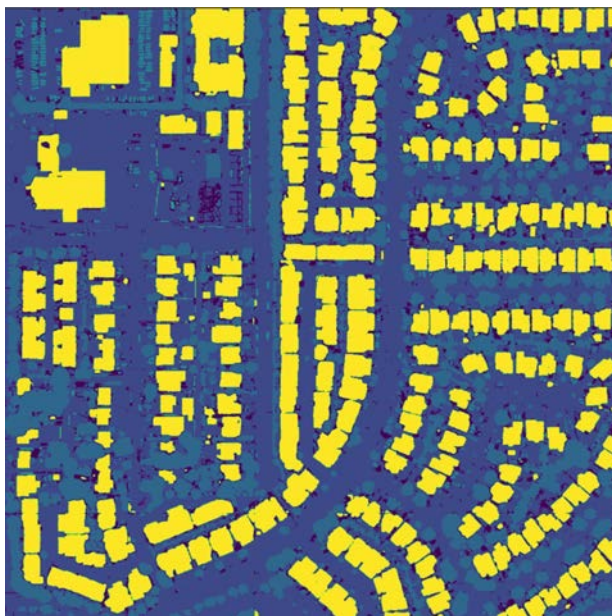
La segmentación obtenida con la proyección ortogonal demostró ser precisa y bien definida (Figuras 4.6 y 4.7). Las clases de suelo y construcciones alcanzaron valores de IoU superiores al 90% (Tabla 4.5), indicando una excelente capacidad para distinguir estas categorías.

Figura 4.6. Proyección ortogonal con las etiquetas reales



Fuente: Elaboración Propia

Figura 4.7. Proyección ortogonal con la clasificación de U-net



Fuente: Elaboración Propia

Cuadro 4.5. Resultados de IoU por clase de la proyección ortogonal en Nube de Puntos

| Clase | Suelo | Vegetación | Coches | Camionetas | Lineas Electricas | Rejas | Postes | Construcciones |
|-------|--------|------------|---------|------------|-------------------|-------|--------|----------------|
| IoU | 0.9512 | .7668 | .461198 | 0.0338 | 0.6788 | .3723 | .1576 | 0.9293 |

Fuente: Elaboración propia

La precisión general y la IoU media también fueron altas (Tabla 4.6), confirmando la efectividad de esta técnica para la segmentación semántica.

Cuadro 4.6. Resultados de métricas de rendimiento de la proyección ortogonal en Nube de Puntos

| Métrica | Precisión | Media de IoU |
|---------|-----------|--------------|
| Valor | 0.9355 | 0.5486 |

Fuente: Elaboración propia

Sin embargo, es importante destacar que los resultados obtenidos representan píxeles segmentados convertidos en puntos individuales, los cuales deben ser propagados adecuadamente en la nube de puntos completa. Para abordar este desafío, se aplicó el algoritmo de Label Spreading, obteniendo una mejora en los valores de IoU por clase (Tabla 4.7) y en las métricas generales (Tabla 4.8). Esto demuestra la importancia de considerar la propagación de etiquetas en la nube de puntos para obtener una segmentación coherente y precisa.

Cuadro 4.7. Resultados de IoU por clase de la proyección ortogonal en Nube de Puntos Desconocida Completa

| Clase | Suelo | Vegetación | Coches | Camionetas | Lineas Electricas | Rejas | Postes | Construcciones |
|-------|--------|------------|--------|------------|-------------------|--------|--------|----------------|
| IoU | 0.9127 | 0.7672 | 0.4381 | 0.0430 | 0.5157 | 0.3354 | 0.0750 | 0.8888 |

Fuente: Elaboración propia

Cuadro 4.8. Resultados de métricas de rendimiento de la proyección ortogonal en Nube de Puntos Completa

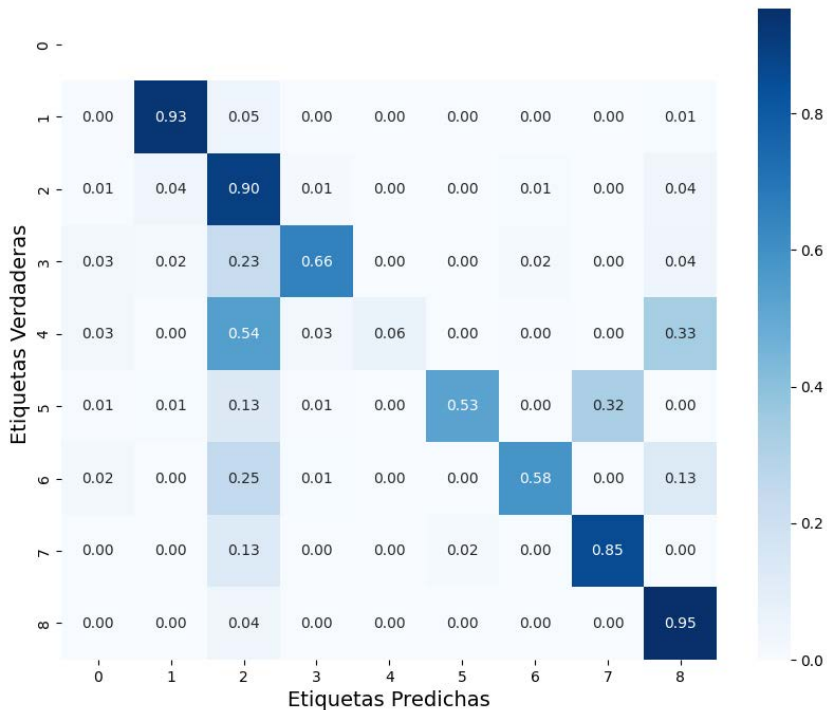
| Métrica | Precisión | Media de IoU |
|---------|-----------|--------------|
| Valor | 0.9186 | 0.4969 |

Fuente: Elaboración propia

La matriz de confusión (Figura 4.8) permitió una evaluación detallada de la precisión del modelo, identificando tanto aciertos como

errores en la clasificación. Esta herramienta proporciona información valiosa sobre el rendimiento del modelo en cada clase y ayuda a identificar áreas de mejora.

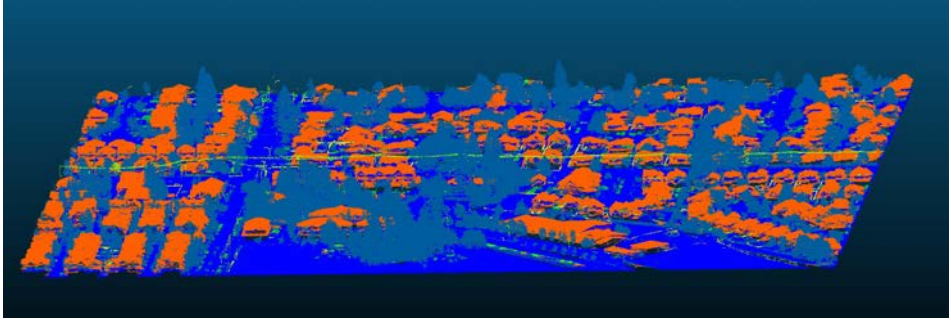
Figura 4.8. Matriz de Confusión de Segmentación con modelo Ortogonal



Fuente: Elaboración Propia

Finalmente, la visualización de la nube de puntos completamente segmentada (Figura 4.9) mostró una clara delimitación entre las diferentes clases, preservando las estructuras principales de la escena original. Esto valida la eficacia de la proyección ortogonal para la segmentación semántica de nubes de puntos.

Figura 4.9. Escena de la nube de Puntos con Valores Predichos

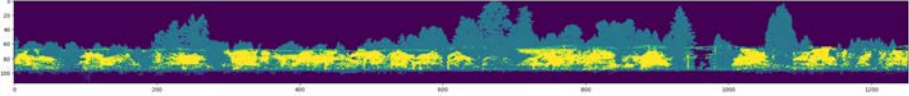


Fuente: Elaboración Propia

4.2.1.2. Proyección Ortogonal Multivista

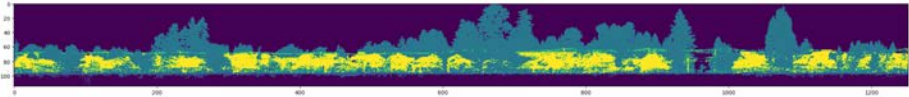
La proyección ortogonal multivista produjo una segmentación más detallada en comparación con la proyección ortogonal simple (Figuras 4.10 y 4.11). Los valores de IoU por clase (Tabla 4.9) y las métricas generales (Tabla 4.10) fueron ligeramente inferiores, pero aún así indicativos de un buen rendimiento. Esto sugiere que la información adicional capturada por las múltiples vistas puede mejorar la segmentación en ciertos aspectos, pero también introduce cierta complejidad en el proceso.

Figura 4.10. Proyección ortogonal multivista con la clasificación correcta



Fuente: Elaboración Propia

Figura 4.11. Proyección ortogonal multivista con la clasificación de U-net



Fuente: Elaboración Propia

Cuadro 4.9. Resultados de IoU por clase de la proyección ortogonal multivista en Nube de Puntos Desconocida

| Clase | Suelo | Vegetación | Coches | Camionetas | Lineas Eléctricas | Rejas | Postes | Construcciones |
|-------|--------|------------|--------|------------|-------------------|--------|--------|----------------|
| IoU | 0.9436 | 0.7465 | 0.5248 | 0.1738 | 0.3788 | 0.0535 | 0.00 | 0.9257 |

Fuente: Elaboración propia

Cuadro 4.10. Resultados de métricas de rendimiento de la proyección ortogonal multivista en Nube de Puntos

| Métrica | Precisión General | Promedio IoU |
|---------|-------------------|--------------|
| Valor | 0.94 | 0.4683 |

Fuente: Elaboración propia

Al igual que con la proyección ortogonal, se aplicó Label Spreading

para propagar los puntos segmentados en la nube completa, obteniendo mejoras en los valores de IoU por clase (Tabla 4.11) y en las métricas generales (Tabla 4.12). Esto confirma la importancia de considerar la propagación de etiquetas para obtener una segmentación coherente y precisa en la nube de puntos completa.

Cuadro 4.11. Resultados de IoU por clase de la proyección ortogonal multivista en Nube de Puntos Desconocida Completa

| Clase | Suelo | Vegetación | Coches | Camionetas | Lineas Eléctricas | Rejas | Postes | Construcciones |
|-------|--------|------------|--------|------------|-------------------|--------|--------|----------------|
| IoU | 0.9316 | 0.8015 | 0.4855 | 0.1725 | 0.3220 | 0.0593 | 0.0000 | 0.9065 |

Fuente: Elaboración propia

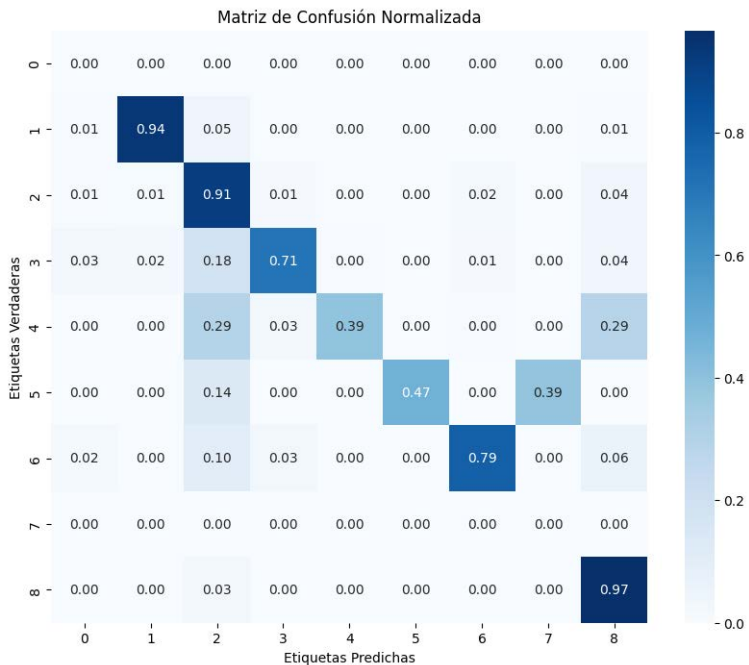
Cuadro 4.12. Resultados de métricas de rendimiento de la proyección ortogonal multivista en Nube de Puntos

| Métrica | Precisión General | Promedio IoU |
|---------|-------------------|--------------|
| Valor | 0.9344 | 0.4598 |

Fuente: Elaboración propia

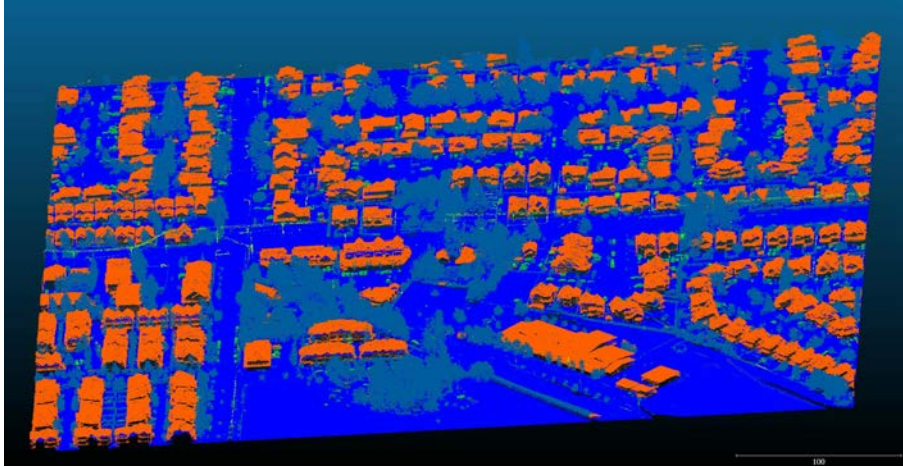
La matriz de confusión (Figura 4.12) permitió evaluar la precisión del modelo, mientras que la visualización de la nube de puntos segmentada (Figura 4.13) mostró una clara delimitación de las diferentes categorías, aunque con algunas zonas menos perfectamente definidas en comparación con la proyección ortogonal. Estos resultados sugieren que la proyección ortogonal multivista es una técnica prometedora para la segmentación semántica de nubes de puntos, pero puede requerir ajustes adicionales para alcanzar el mismo nivel de precisión que la proyección ortogonal simple.

Figura 4.12. Matriz de Confusión de Segmentación con modelo Multivista



Fuente: Elaboración Propia

Figura 4.13. Escena de la nube de Puntos con Valores Predichos

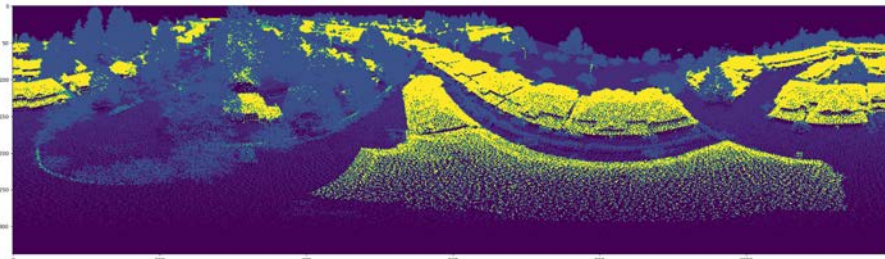


Fuente: Elaboración Propia

4.2.1.3. Proyección Esférica

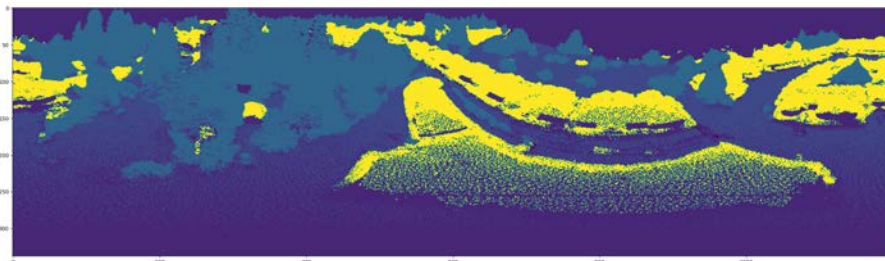
La proyección esférica arrojó resultados mixtos en la validación del modelo. Aunque en algunos casos la segmentación fue precisa, en otros se observaron errores (Figuras 4.14 y 4.15). Los valores de IoU por clase (Tabla 4.13) y las métricas generales (Tabla 4.14) fueron más bajos que los obtenidos con las otras dos técnicas, lo que sugiere que la proyección esférica enfrenta desafíos adicionales en la segmentación semántica de nubes de puntos.

Figura 4.14. Proyección esférica de una nube de Puntos (Datos Reales)



Fuente: Elaboración Propia

Figura 4.15. Proyección esférica de una nube de Puntos (Datos Predichos)



Fuente: Elaboración Propia

Cuadro 4.13. Resultados de IoU por clase de la proyección ortogonal en Nube de Puntos Desconocida

| Clase | Suelo | Vegetación | Coches | Camionetas | Lineas Electricas | Rejas | Postes | Construcciones |
|-------|--------|------------|--------|------------|-------------------|-------|--------|----------------|
| IoU | 0.8405 | 0.6934 | 0.0 | 0.0 | 0.0015 | 0.0 | 0.0 | 0.7143 |

Fuente: Elaboración propia

Cuadro 4.14. Resultados de métricas de rendimiento de la proyección ortogonal en Nube de Puntos

| Métrica | Precisión | Media de IoU |
|---------|-----------|--------------|
| Valor | 0.8267 | 0.28 |

Fuente: Elaboración propia

Tras aplicar Label Spreading, se observaron mejoras en los valores de IoU por clase (Tabla 4.15) y en las métricas generales (Tabla 4.16), aunque seguían siendo inferiores a los de las proyecciones ortogonales. Esto indica que, si bien la propagación de etiquetas es beneficiosa, la proyección esférica puede requerir enfoques adicionales o modificaciones en la arquitectura de la red para mejorar su rendimiento.

Cuadro 4.15. Resultados de IoU por clase de la proyección esférica en Nube de Puntos Desconocida

| Clase | Suelo | Vegetación | Coches | Camionetas | Lineas Eléctricas | Rejas | Postes | Construcciones |
|-------|---------------|------------|--------|------------|-------------------|-------|--------|----------------|
| IoU | 0.8436 0.6126 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.6340 | |

Fuente: Elaboración propia

Cuadro 4.16. Resultados de métricas de rendimiento de la proyección ortogonal en Nube de Puntos

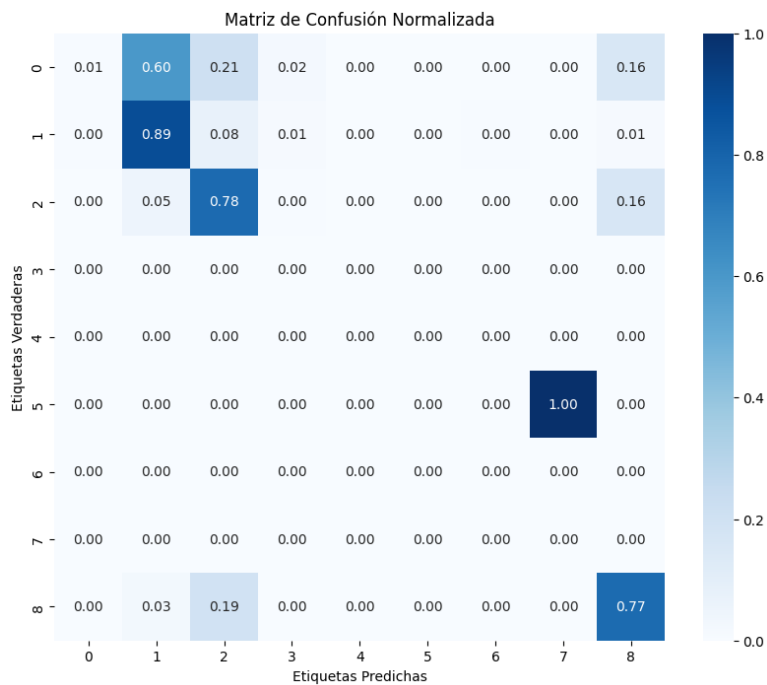
| Métrica | Precisión | Media de IoU |
|---------|-----------|--------------|
| Valor | 0.8234 | 0.20612 |

Fuente: Elaboración propia

La matriz de confusión (Figura 4.16) reveló áreas donde el modelo tuvo un rendimiento destacado y otras que podrían beneficiarse de ajustes. La visualización de la nube de puntos segmentada (Figura 4.17) mostró zonas menos delimitadas en comparación con las otras

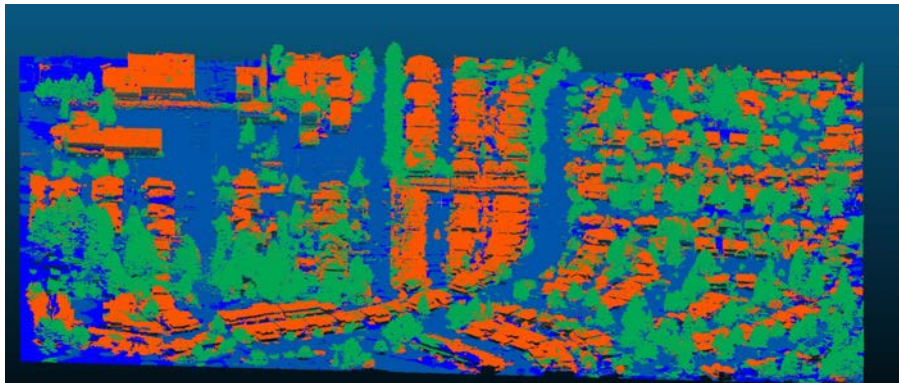
proyecciones, con cierta entremezcla de puntos de diferentes clases. Estos resultados sugieren que, aunque la proyección esférica tiene potencial para la segmentación semántica de nubes de puntos, puede requerir un mayor desarrollo y optimización para alcanzar el mismo nivel de rendimiento que las proyecciones ortogonales.

Figura 4.16. Matriz de Confusión de Segmentación con modelo Ortogonal



Fuente: Elaboración Propia

Figura 4.17. Escena de la nube de Puntos con Valores Predichos



Fuente: Elaboración Propia

Capítulo 5

Conclusiones y trabajo futuro

La segmentación semántica de nubes de puntos sigue siendo un desafío en el campo de la visión por computadora, y la elección de la técnica de proyección tiene un impacto significativo en los resultados. Este estudio ha explorado en profundidad las fortalezas y limitaciones de tres métodos de proyección destacados: ortogonal, ortogonal multivista y esférica. A través de una serie de experimentos rigurosos utilizando el dataset DALES y la arquitectura U-Net, hemos obtenido hallazgos relevantes que contribuyen al avance de este campo en rápida evolución.

Los resultados demuestran que la proyección ortogonal se posiciona como el enfoque más equilibrado y eficaz para la segmentación semántica de nubes de puntos. Con una precisión del 91.86 % y un IoU promedio del 49.69 % en el conjunto de prueba, este método supera a las otras técnicas evaluadas. La capacidad de la proyección ortogonal para representar la nube de puntos en 2D preservando las relaciones espaciales resulta en

segmentaciones más precisas y coherentes.

Por otro lado, la proyección ortogonal multivista también mostró un rendimiento prometedor, alcanzando una precisión del 93.44 % y un IoU promedio del 45.98 %. Aunque ligeramente inferior a la proyección ortogonal simple, este enfoque tiene el potencial de capturar información adicional de la escena al considerar múltiples vistas. Sin embargo, también introduce cierta complejidad en el proceso de segmentación, lo que puede requerir ajustes adicionales para optimizar su rendimiento.

En contraste, la proyección esférica evidenció desafíos significativos en términos de precisión, con una precisión del 82.67 % y un IoU promedio del 20 % en el conjunto de prueba. Los resultados sugieren que la distorsión introducida por esta técnica afecta negativamente la calidad de la segmentación, limitando su utilidad en aplicaciones que requieren una alta precisión.

Un análisis detallado de las matrices de confusión reveló que la proyección ortogonal logra bajas tasas de error en clases críticas como suelo y construcciones, además de mantener un rendimiento superior en otras categorías en comparación con la proyección ortogonal multivista. Esta precisión en categorías clave explica su alto IoU global y refuerza su idoneidad para la segmentación semántica de nubes de puntos.

En resumen, este estudio ha demostrado que la proyección ortogonal es el método más apropiado para la segmentación semántica de nubes de puntos aéreas utilizando la arquitectura U-Net. Su capacidad para preservar el contexto espacial y la visibilidad de superficies permite una segmentación 2D precisa que se propaga eficazmente de vuelta a la nube de puntos 3D. Aunque la proyección ortogonal multivista ofrece mejoras para áreas ocluidas, lo hace a costa de algunos artefactos. Por otro

lado, la proyección esférica, a pesar de sus limitaciones actuales, abre la puerta a futuras investigaciones para mejorar su rendimiento y explorar enfoques alternativos.

5.1. Trabajos Futuro

Los hallazgos de este estudio sientan las bases para futuras líneas de investigación y desarrollo en el campo de la segmentación semántica de nubes de puntos. Entre las direcciones prometedoras se encuentran:

- **Enfoques Híbridos:** Combinar las fortalezas de las proyecciones ortogonales y de múltiples vistas podría dar lugar a un método que capture todas las superficies sin introducir artefactos significativos. Esta combinación podría explorar formas de fusionar las vistas para maximizar la cobertura de superficies y minimizar las distorsiones.
- **Técnicas de Costura Avanzadas:** Para las proyecciones de múltiples vistas, algoritmos de costura avanzados podrían mitigar la introducción de artefactos, llevando a segmentaciones más consistentes. Estas técnicas podrían emplear algoritmos de aprendizaje profundo para mejorar la fusión de las diferentes vistas y garantizar la continuidad en las transiciones.
- **Uso de Modelos Fundacionales:** Implementar modelos fundacionales, como los basados en GPT y otras arquitecturas de aprendizaje profundo, para analizar y procesar nubes de puntos. Estos modelos pueden ser entrenados con grandes conjuntos de datos y adaptarse para tareas específicas de segmentación, ofreciendo mejoras en la precisión y la eficiencia. Además,

pueden facilitar la integración de información contextual y semántica en el proceso de segmentación, lo que es particularmente útil en escenarios complejos como los entornos urbanos.

- **Proyecciones Adaptativas:** Dependiendo de la complejidad de la escena y la distribución de los objetos, se podrían emplear técnicas de proyección adaptativas para elegir dinámicamente el mejor método de proyección. Esto implicaría desarrollar sistemas inteligentes capaces de analizar la escena y determinar qué tipo de proyección maximizaría la precisión y eficiencia de la segmentación.

En conclusión, este trabajo ha arrojado luz sobre la eficacia de las técnicas de proyección ortogonal y ortogonal multivista para la segmentación semántica de nubes de puntos, al tiempo que ha identificado los desafíos y oportunidades asociados con la proyección esférica. Estos hallazgos contribuyen al avance del campo y abren nuevas vías de investigación para mejorar aún más el rendimiento y la aplicabilidad de estas técnicas en diversos dominios, desde la topografía y la planificación urbana hasta la robótica y la percepción automatizada del entorno. A medida que la demanda de segmentación 3D precisa continúa creciendo, la mejora de estas técnicas seguirá siendo una prioridad en el campo de la visión por computadora, impulsando innovaciones que transformarán la forma en que percibimos y analizamos el mundo que nos rodea.

Referencias

- Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., y Savarese, S. (2016). 3d semantic parsing of large-scale indoor spaces. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1534–1543).
- Barramou, F., El Brirchi, E. H., Mansouri, K., y Dehbi, Y. (Eds.). (2022). *Geospatial Intelligence: Applications and Future Trends*. Cham: Springer International Publishing. Descargado 2023-03-17, de <https://link.springer.com/10.1007/978-3-030-80458-9> doi: 10.1007/978-3-030-80458-9
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., y Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 834–848.
- Choy, C., Gwak, J., y Savarese, S. (2019). *4d spatio-temporal convnets: Minkowski convolutional neural networks*.
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., y Nießner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 5828–5839).
- Fan, H., y Yang, Y. (2019). Pointrnn: Point recurrent neural network for moving point cloud processing. *arXiv preprint arXiv:1910.08287*.
- Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep learning*. MIT Press. (<http://www.deeplearningbook.org>)
- Graham, B., Engelcke, M., y Van Der Maaten, L. (2018). 3d semantic segmentation with submanifold sparse convolutional networks. En *Proceedings of the ieee conference on computer vision and pattern*

- recognition* (pp. 9224–9232).
- Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., y Bennamoun, M. (2021, diciembre). Deep Learning for 3D Point Clouds: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12), 4338–4364. Descargado 2023-03-09, de <https://ieeexplore.ieee.org/document/9127813/> doi: 10.1109/TPAMI.2020.3005434
- He, K., Zhang, X., Ren, S., y Sun, J. (2016). Deep residual learning for image recognition. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).
- Iandola, F. N., Moskewicz, M. W., Ashraf, K., Han, S., Dally, W. J., y Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, *abs/1602.07360*. Descargado de <http://arxiv.org/abs/1602.07360>
- Jo, H. C., Sohn, H.-G., y Lim, Y. M. (2020). A lidar point cloud data-based method for evaluating strain on a curved steel plate subjected to lateral pressure. *Sensors*, 20(3). Descargado de <https://www.mdpi.com/1424-8220/20/3/721> doi: 10.3390/s20030721
- Jovančević, I., Pham, H.-H., Orteu, J.-J., Gilblas, R., Harvent, J., Maurice, X., y Brèthes, L. (2017, octubre). 3D Point Cloud Analysis for Detection and Characterization of Defects on Airplane Exterior Surface. *Journal of Nondestructive Evaluation*, 36(4), 74. Descargado 2023-08-16, de <https://doi.org/10.1007/s10921-017-0453-1> doi: 10.1007/s10921-017-0453-1
- Jähne, B. (2005). *Digital image processing 6th edition*. Berlin [u.a.]: Springer. Descargado de http://www.amazon.com/Digital-Image-Processing-Bernd-J%C3%A4hne/dp/3540240357/ref=sr_1_1?s=books&ie=UTF8&qid=1328626244&sr=1-1

- Kingma, D. P., y Ba, J. (2017). *Adam: A method for stochastic optimization*.
- Liu, S., Zhang, M., Kadam, P., y Kuo, C.-C. J. (2021). *3D Point Cloud Analysis: Traditional, Deep Learning, and Explainable Machine Learning Methods*. Cham: Springer International Publishing. Descargado 2023-03-09, de <https://link.springer.com/10.1007/978-3-030-89180-0> doi: 10.1007/978-3-030-89180-0
- Milioto, A., Vizzo, I., Behley, J., y Stachniss, C. (2019, 11). Rangenet ++: Fast and accurate lidar semantic segmentation. En (p. 4213-4220). doi: 10.1109/IROS40897.2019.8967762
- Pistilli, F. (2020). *Graphpointnet: Graph convolutional neural network for point cloud denoising* (Tesis Doctoral no publicada). University of Illinois at Chicago.
- Qi, C. R., Su, H., Mo, K., y Guibas, L. J. (2017). *Pointnet: Deep learning on point sets for 3d classification and segmentation*.
- Qi, C. R., Yi, L., Su, H., y Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.
- Rethage, D., Wald, J., Sturm, J., Navab, N., y Tombari, F. (2018). Fully-convolutional point networks for large-scale point clouds. En *Proceedings of the european conference on computer vision (eccv)* (pp. 596–611).
- Ronneberger, O., Fischer, P., y Brox, T. (2015a). *U-net: Convolutional networks for biomedical image segmentation*.
- Ronneberger, O., Fischer, P., y Brox, T. (2015b). U-net: Convolutional networks for biomedical image segmentation. En *Medical image computing and computer-assisted intervention—miccai 2015: 18th international conference, munich, germany, october 5-9, 2015*,

- proceedings, part iii 18* (pp. 234–241).
- Rosu, R. A., Schütt, P., Quenzel, J., y Behnke, S. (2021). *Latticenet: Fast spatio-temporal point cloud segmentation using permutohedral lattices*.
- Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.-H., y Kautz, J. (2018). *Splatnet: Sparse lattice networks for point cloud processing*.
- Tchapmi, L. P., Choy, C. B., Armeni, I., Gwak, J., y Savarese, S. (2017). *Segcloud: Semantic segmentation of 3d point clouds*.
- Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., y Guibas, L. J. (2019). Kpconv: Flexible and deformable convolution for point clouds. En *Proceedings of the ieee/cvf international conference on computer vision* (pp. 6411–6420).
- Varney, N., Asari, V. K., y Graehling, Q. (2020, abril). *DALES: A Large-scale Aerial LiDAR Data Set for Semantic Segmentation*. arXiv. Descargado 2023-02-28, de <http://arxiv.org/abs/2004.11985> (arXiv:2004.11985 [cs, stat])
- Wang, L., Huang, Y., Hou, Y., Zhang, S., y Shan, J. (2019). Graph attention convolution for point cloud semantic segmentation. En *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 10296–10305).
- Wang, Y., Shi, T., Yun, P., Tai, L., y Liu, M. (s.f.). Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. arxiv 2018. *arXiv preprint arXiv:1807.06288*.
- Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., y Solomon, J. M. (2019). *Dynamic graph cnn for learning on point clouds*.
- Wu, B., Zhou, X., Zhao, S., Yue, X., y Keutzer, K. (2018). *Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud*.

- Wu, W., Fuxin, L., y Shan, Q. (2023). Pointconvformer: Revenge of the point-based convolution. En *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 21802–21813).
- Wu, W., Qi, Z., y Fuxin, L. (2019). Pointconv: Deep convolutional networks on 3d point clouds. En *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 9621–9630).
- Xie, G., Li, Y., Wang, Y., Li, Z., y Qu, H. (2023). 3d point cloud object detection algorithm based on temporal information fusion and uncertainty estimation. *Remote Sensing*, 15(12), 2986.
- Yan, S., Song, C., Kong, Y., y Huang, Q. (2023, junio). *Learning from Multi-View Representation for Point-Cloud Pre-Training*. arXiv. Descargado 2023-08-16, de <http://arxiv.org/abs/2306.02558> (arXiv:2306.02558 [cs])
- Ye, X., Li, J., Huang, H., Du, L., y Zhang, X. (2018). 3d recurrent neural networks with context fusion for point cloud semantic segmentation. En *Proceedings of the european conference on computer vision (eccv)* (pp. 403–417).
- Zeng, G., y Zheng, G. (2017). Multi-stream 3d FCN with multi-scale deep supervision for multi-modality isointense infant brain MR image segmentation. *CoRR*, *abs/1711.10212*. Descargado de <http://arxiv.org/abs/1711.10212>
- Zhang, P., He, H., Wang, Y., Liu, Y., Lin, H., Guo, L., y Yang, W. (2022). 3D Urban Buildings Extraction Based on Airborne LiDAR and Photogrammetric Point Cloud Fusion According to U-Net Deep Learning Model Segmentation. *IEEE Access*, 10, 20889–20897. Descargado 2022-10-19, de <https://ieeexplore.ieee.org/document/9716925/> doi: 10.1109/ACCESS.2022.3152744
- Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., y Yan, G.

- (2016, junio). An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sensing*, 8(6), 501. Descargado 2022-10-19, de <http://www.mdpi.com/2072-4292/8/6/501> doi: 10.3390/rs8060501
- Zhou, D., Bousquet, O., Lal, T., Weston, J., y Schölkopf, B. (2003). Learning with local and global consistency. *Advances in neural information processing systems*, 16.
- Zhou, H., Feng, Y., Fang, M., Wei, M., Qin, J., y Lu, T. (2021). Adaptive graph convolution for point cloud analysis. En *Proceedings of the ieee/cvf international conference on computer vision* (pp. 4965–4974).