



Universidad de Oviedo

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.

**GRADO EN INGENIERÍA ELECTRÓNICA
INDUSTRIAL Y AUTOMÁTICA**

ÁREA DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

**EXTRACCIÓN DE OBJETOS 3D A PARTIR DE UNA
NUBE DE PUNTOS**

D. YOEL PALACIO MORENO

TUTOR: D. RAFAEL C. GONZÁLEZ DE LOS REYES

COTUTORA: Dña. SARA ROOS-HOEFGEEST TORIBIO

FECHA: febrero 2024



Índice de contenidos

1. Introducción	5
1.1.- PLANTEAMIENTO DEL PROBLEMA Y OBJETIVO.....	5
2. Marco Teórico.....	7
2.1.- NUBES DE PUNTOS 3D	7
2.2.- ADQUISICION DE NUBES DE PUNTOS	7
2.2.1.- <i>Escaneo Láser y LiDAR</i>	8
2.2.2.- <i>Fotogrametría</i>	9
2.2.3.- <i>Cámaras RGB-D</i>	9
2.3.- CARACTERÍSTICAS DE UNA NUBE DE PUNTOS	10
2.3.1.- <i>Densidad</i>	10
2.3.2.- <i>Vecindad</i>	11
2.3.3.- <i>Normal en un punto</i>	11
2.4.- SEGMENTACIÓN	12
2.4.1.- <i>Métodos basados en bordes</i>	13
2.4.2.- <i>Métodos basados en el principio de fusión</i>	13
2.4.3.- <i>Métodos por ajuste de modelo</i>	15
2.4.4.- <i>Machine Learning</i>	16
3. Estado del arte	22
3.1.- CRECIMIENTO DE REGIONES	22
3.2.- SPLIT AND MERGE	23
3.3.- AJUSTE DE MODELOS: RANSAC	24
3.4.- K-MEANS	24
3.5.- MEAN-SHIFT	25
3.6.- DBSCAN	26
4. Método Propuesto	27
4.1.- TRATAMIENTO INICIAL	28
4.2.- SPLIT	29
4.3.- MERGE ITERATIVO.....	31
4.4.- TRATAMIENTO FINAL	33
5. Experimentación	34
5.1.- NUBES DE PUNTOS ARTIFICIALES.....	35
5.2.- NUBES DE PUNTOS REALES	38



5.3.-	PARÁMETROS DEL ALGORITMO.....	40
5.4.-	PROCEDIMIENTO PARA LA REALIZACIÓN DE LAS PRUEBAS	41
6.	Resultados	43
6.1.-	PRUEBAS INICIALES	43
6.1.1.-	<i>Maqueta 1</i>	44
6.1.2.-	<i>Maqueta 2</i>	45
6.1.3.-	<i>Maqueta 3</i>	46
6.1.4.-	<i>Maqueta 4</i>	47
6.2.-	PRUEBAS FINALES	48
6.2.1.-	<i>Nube Real 1</i>	49
6.2.2.-	<i>Nube Real 2</i>	50
6.2.3.-	<i>Nube Real 3</i>	50
6.2.4.-	<i>Nube Real 4</i>	51
7.	Discusión	52
8.	Conclusiones	58
9.	Bibliografía	60



Índice de Figuras

FIGURA 2.1.- PRINCIPIO DEL ESCÁNER LASER [4]	8
FIGURA 2.2.- CLASIFICACIÓN DE LOS MÉTODOS DE SEGMENTACIÓN	12
FIGURA 2.3.- EJEMPLO DE APLICACIÓN DE K-MEANS	18
FIGURA 2.4.- PROCESO DEL ALGORITMO MEAN-SHIFT	19
FIGURA 4.1.- MÉTODO PROPUESTO	27
FIGURA 4.2.- EJEMPLO DE UNA NUBE DE PUNTOS SOBRESEGMENTADA EN LA QUE CADA CLUSTER ES REPRESENTADO POR UN COLOR DIFERENTE.	30
FIGURA 4.3.- DIAGRAMA DE FLUJO DEL MERGE ITERATIVO	31
FIGURA 5.1.- NUBES DE PUNTOS ARTIFICIALES	35
FIGURA 5.2.- NUBES DE PUNTOS REALES	38
FIGURA 5.3.- EJEMPLO DE REDUCCIÓN DE DENSIDAD DE UNA NUBE DE PUNTOS	42
FIGURA 6.1.- RESULTADO DE LA MAQUETA 1	44
FIGURA 6.2.- RESULTADO DE LA MAQUETA 2	45
FIGURA 6.3.- RESULTADO DE LA MAQUETA 3	46
FIGURA 6.4.- RESULTADO DE LA MAQUETA 4	47
FIGURA 6.5.- RESULTADO DE LA NUBE REAL 1	49
FIGURA 6.6.- RESULTADO DE LA NUBE REAL 2	50
FIGURA 6.7.- RESULTADO DE LA NUBE REAL 3	50
FIGURA 6.8.- RESULTADO DE LA NUBE REAL 4	51
FIGURA 7.1.- IMAGEN DE LA VISTA INFERIOR DE LA MAQUETA 1	52
FIGURA 7.2.- QUILLA Y TIMÓN DEL BARCO DE LA NUBE REAL 1	54
FIGURA 7.3.- RESULTADOS DEL SPLIT Y DEL MERGE DE LA NUBE REAL 1	54
FIGURA 7.4.- RESULTADOS DEL SPLIT Y DEL MERGE DE LA NUBE REAL 2	55



Índice de Tablas

TABLA 6.1.- PARÁMETROS FIJOS PARA LAS PRUEBAS INICIALES	43
TABLA 6.2.- PARÁMETROS PARA LA MAQUETA 1	44
TABLA 6.3.- RESULTADOS DE LA MAQUETA 1	44
TABLA 6.4.- PARÁMETROS PARA LA MAQUETA 2	45
TABLA 6.5.- RESULTADOS DE LA MAQUETA 2	45
TABLA 6.6.- PARÁMETROS PARA LA MAQUETA 3	46
TABLA 6.7.- RESULTADOS DE LA MAQUETA 3	46
TABLA 6.8.- PARÁMETROS PARA LA MAQUETA 4	47
TABLA 6.9.- RESULTADOS DE LA MAQUETA 4	47
TABLA 6.10.- PARÁMETROS FIJOS PARA LAS PRUEBAS FINALES	48
TABLA 6.11.- PARÁMETROS PARA LA NUBE REAL 1	49
TABLA 6.12.- PARÁMETROS PARA LA NUBE REAL 2	50
TABLA 6.13.- PARÁMETROS PARA LA NUBE REAL 3	50
TABLA 6.14.- PARÁMETROS PARA LA NUBE REAL 4	51



1. Introducción

Actualmente, las nubes de puntos tridimensionales son un recurso esencial gracias a la accesibilidad y la precisión de los escáneres 3D. Estas representaciones digitales proporcionan una visión completa y precisa de objetos, estructuras y paisajes, desempeñando un papel fundamental en una amplia gama de campos, desde la arquitectura y la ingeniería hasta la medicina y la realidad virtual. La segmentación de estas nubes de puntos se ha vuelto esencial para extraer información específica, clasificar elementos y facilitar el análisis detallado de datos tridimensionales.

1.1.- PLANTEAMIENTO DEL PROBLEMA Y OBJETIVO

Este proyecto surge de la iniciativa del personal de la Escuela de Marina Civil de Gijón, quienes buscan desarrollar un procedimiento por el cual se puedan identificar las incrustaciones de la obra viva de los barcos. Esto permitiría evaluar de manera rápida y efectiva los daños generados por dichas incrustaciones, posibilitando que se propongan medidas, que puedan ser aplicadas por parte de los responsables de los puertos y los propietarios de las embarcaciones.

La metodología que pretenden aplicar consiste en realizar un mapeado 3D de embarcaciones menores de diferentes puertos de la región asturiana para calcular el porcentaje de incrustaciones que contienen la obra viva de los barcos. Conocer dicho porcentaje resulta crucial para evitar problemas asociados a las invasiones biológicas y la resistencia al avance, lo cual provoca un aumento del consumo de combustible y de las emisiones de gases de efecto invernadero.

El mapeado 3D se realiza a embarcaciones puestas en carena, es decir, fuera del mar y apoyadas sobre una estructura que depende del tipo de barco y de las instalaciones portuarias disponibles. Esto provoca que las escenas capturadas contengan elementos no deseados, los cuales pueden interferir en el proceso de identificación de incrustaciones. Por



ello, es necesario realizar una segmentación que consiga extraer con precisión el casco del barco, eliminando el resto de los objetos presentes en la captura. Hasta el momento, este recorte se estaba realizando a mano utilizando el software Cloud Compare [1], lo cual es un proceso que implica una inversión considerable en tiempo y esfuerzo.

El propósito fundamental de este proyecto consiste en realizar una investigación de las diferentes técnicas disponibles en el campo de la segmentación de nubes de puntos para analizar sus características y aplicaciones. A partir de este análisis, se buscará desarrollar un método viable que pueda abordar la problemática propuesta. Este método aspira a ser una solución práctica y funcional que permita extraer con precisión el casco de los barcos de distintas nubes de puntos.



2. Marco Teórico

2.1.- NUBES DE PUNTOS 3D

Una nube de puntos 3D se refiere a una agrupación de puntos situados en el espacio que representan uno o varios objetos tridimensionales. Cada punto es representado por un vector que indica sus coordenadas XYZ y, de forma opcional, pueden llevar asignados unas características físicas adicionales, como por ejemplo el color o la componente normal. Estas nubes de puntos pueden ser generadas de forma artificial, o provenir de capturas de elementos reales a través de diferentes métodos. [2]

La importancia de las nubes de puntos radica en su capacidad para representar de manera fiel la realidad tridimensional, permitiendo su visualización, análisis y manipulación computacional. Estos conjuntos de datos son fundamentales en numerosos campos, como la ingeniería, la arquitectura, la medicina, la arqueología y la cartografía, entre otros.

La representación de las nubes de puntos es escalable, lo que significa que pueden capturar desde pequeños objetos hasta escenas extensas con millones de puntos. Además, su capacidad para almacenar información detallada de la superficie, así como la información de los atributos de cada punto, proporciona un rico conjunto de datos para realizar diversas tareas.

2.2.- ADQUISICION DE NUBES DE PUNTOS

La captura de datos tridimensionales es una técnica fundamental para comprender con precisión la forma, la estructura y las características visuales de elementos reales. Estos datos se obtienen mediante el muestreo sistemático de la superficie del objeto mediante luz, láser, ultrasonido o fotogrametría, generando conjuntos de puntos que representan la estructura y las características visuales de la superficie capturada.

Existen diversos métodos de adquisición de nubes de puntos 3D que se distinguen principalmente por las tecnologías empleadas y los principios fundamentales que utilizan para capturar la información. A continuación, se citan los métodos más relevantes.

2.2.1.- Escaneo Láser y LiDAR

La tecnología del escaneo laser [3] está basada en la emisión de haces de luz de alta precisión hacia un objeto o entorno. El sensor láser, con sistemas de cronometraje avanzados, puede medir el tiempo que tarda cada pulso de luz en viajar desde el emisor hasta el punto de impacto en una superficie y de regreso al sensor. Este tiempo de vuelo le permite calcular con precisión la distancia entre el sensor y el punto de impacto.

Realizado de manera continua este proceso de emisión, reflexión y medición desde múltiples perspectivas, se adquieren puntos de la superficie escaneada. Estos puntos, en coordenadas tridimensionales, representan la forma y estructura del objeto capturado.

Uno de los dispositivos más utilizados en la adquisición de nubes de puntos que emplean esta tecnología son los LiDAR (Light Detection and Ranging). Estos sensores suelen requerir de un sistema de navegación y posicionamiento para determinar la posición y orientación absoluta del sensor en el caso de que estén montados sobre vehículos automóviles, aeronaves o UAV.[4]

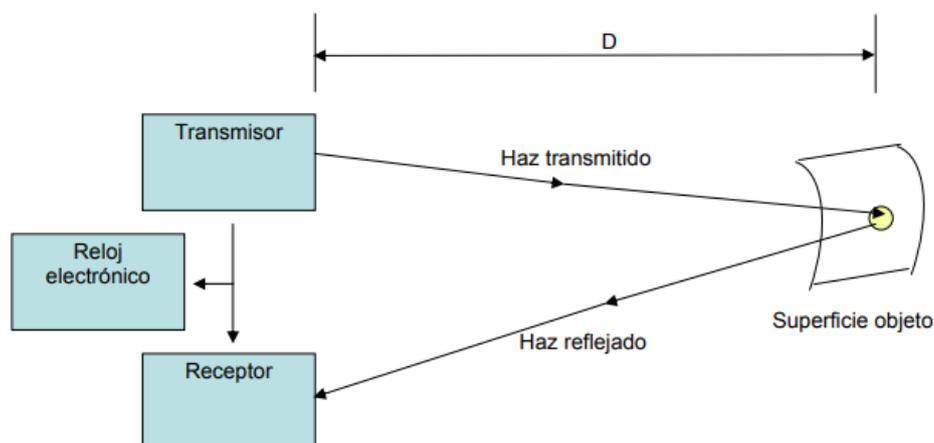


Figura 2.1.- Principio del escáner laser [4]



2.2.2.- Fotogrametría

La fotogrametría es una técnica que se basa en el análisis de imágenes fotográficas para obtener información tridimensional precisa de objetos, terrenos o estructuras. Esta técnica aprovecha la información contenida en las imágenes tomadas desde diferentes ángulos para calcular la posición tridimensional de puntos específicos en el espacio.

El proceso de fotogrametría implica la captura de imágenes desde múltiples ubicaciones o ángulos alrededor del objeto o área de interés. Estas imágenes se utilizan para identificar puntos comunes en diferentes fotografías y calcular su posición tridimensional utilizando principios geométricos y matemáticos.

Al analizar la disposición de estos puntos comunes en las imágenes, se determinan las coordenadas tridimensionales de esos puntos en el espacio. Con esta información, se genera una representación digital precisa de la superficie y la estructura del objeto en cuestión.

Las imágenes pueden provenir de cualquier cámara digital, desde un smartphone hasta una DSLR (Digital Single-Lens Reflex) estándar o incluso un dron. [5]

2.2.3.- Cámaras RGB-D

Las cámaras RGB-D son dispositivos que combinan una cámara convencional RGB (Red, Green, Blue) con un sensor de profundidad. Estos sensores de profundidad capturan información adicional sobre la distancia entre la cámara y los objetos en la escena. La información de profundidad se sincroniza con la obtenida a partir de la cámara RGB, lo que permite obtener la textura y geometría de los objetos escaneados.

A partir de una imagen, la información de profundidad que precisa una nube de puntos en la que cada punto $p = \{p_x, p_y, p_z\}$ se obtiene de la siguiente forma



$$\begin{aligned}p_x &= (x - c_x) * d(x, y) / f_x \\p_y &= (y - c_y) * d(x, y) / f_y \\p_z &= d(x, y)\end{aligned}\tag{2.1}$$

donde c_x y c_y son los centros de la cámara, f_x y f_y son las distancias focales de la cámara y $d(x, y)$ es la profundidad en xy .

Para integrar los datos de color, utiliza la imagen RGB capturada para agregar el color a cada punto de la nube. [6]

2.3.- CARACTERÍSTICAS DE UNA NUBE DE PUNTOS

Para dar contexto a las diferentes técnicas de procesado y segmentación de una nube de puntos, es fundamental explicar algunos conceptos útiles.

2.3.1.- Densidad

La densidad de una nube es la cantidad de puntos que se encuentran en un área específica del espacio tridimensional. Esta característica puede ser evaluada de diferentes maneras en función del radio escogido:

- Densidad Local: Se refiere a la cantidad de puntos en una vecindad cercana a un punto específico. Esto puede utilizarse para identificar áreas de alta o baja concentración de puntos.
- Densidad global: Representa la distribución general de puntos en toda la nube.
- Densidad por área: Es útil para medir la cantidad de puntos dentro de una región o un volumen específico en la nube de puntos.

La densidad alta de una región puede indicar la presencia de objetos, características o superficies significativas en la escena, mientras que una densidad baja puede implicar áreas con menos información o espacios vacíos. Esta propiedad se puede utilizar para



realizar segmentaciones, agrupamientos, caracterización de objetos o ajustar modelos estadísticos.

2.3.2.- Vecindad

La vecindad es una propiedad fundamental que define el conjunto de puntos que están más próximos a un punto en particular. Esta proximidad se puede determinar mediante un criterio de distancia o por alguna otra norma de conectividad, como el color o la normal.

Determinar la vecindad de los puntos dentro de una nube es crucial para la efectividad de los algoritmos de agrupamiento. Esta capacidad nos permite formar conjuntos coherentes que comparten características similares, siempre y cuando logremos definir con precisión una serie de criterios de vecindad adecuados.

2.3.3.- Normal en un punto

La normal de un punto en un espacio tridimensional es un vector que indica la dirección perpendicular a la superficie local alrededor de dicho punto.

La estimación de la normal en un punto se lleva a cabo mediante tres fases: la determinación de su vecindad, la estimación de una superficie local que se ajuste adecuadamente a esos puntos (normalmente un plano) y, finalmente, el cálculo de la orientación de dicha superficie en el punto dado.

Las normales juegan un papel crucial en la segmentación de nubes de puntos ya que nos proporcionan información necesaria para identificar transiciones entre superficies, detectar bordes y agrupar áreas con propiedades superficiales similares. También son útiles en la eliminación de ruido para filtrar valores atípicos que no siguen la estructura de una superficie local.

2.4.- SEGMENTACIÓN

La segmentación en el análisis de datos es un proceso esencial que busca dividir el conjunto total de datos en partes significativas y suficientemente distintas. Este enfoque permite identificar, aislar y comprender diferentes componentes, patrones o características dentro de los datos.

Cuando se aplica a nubes de puntos, la segmentación se convierte en una herramienta esencial para delimitar regiones coherentes en los datos tridimensionales, lo que posibilita el análisis detallado de objetos, la identificación de formas en el espacio y agiliza el procesado de información.

Ya que el objetivo de este trabajo consiste en la investigación de cómo segmentar una nube de puntos, resulta conveniente hacer una clasificación (Figura 2.2) y descripción de los métodos más utilizados en este campo de la visión por computador.

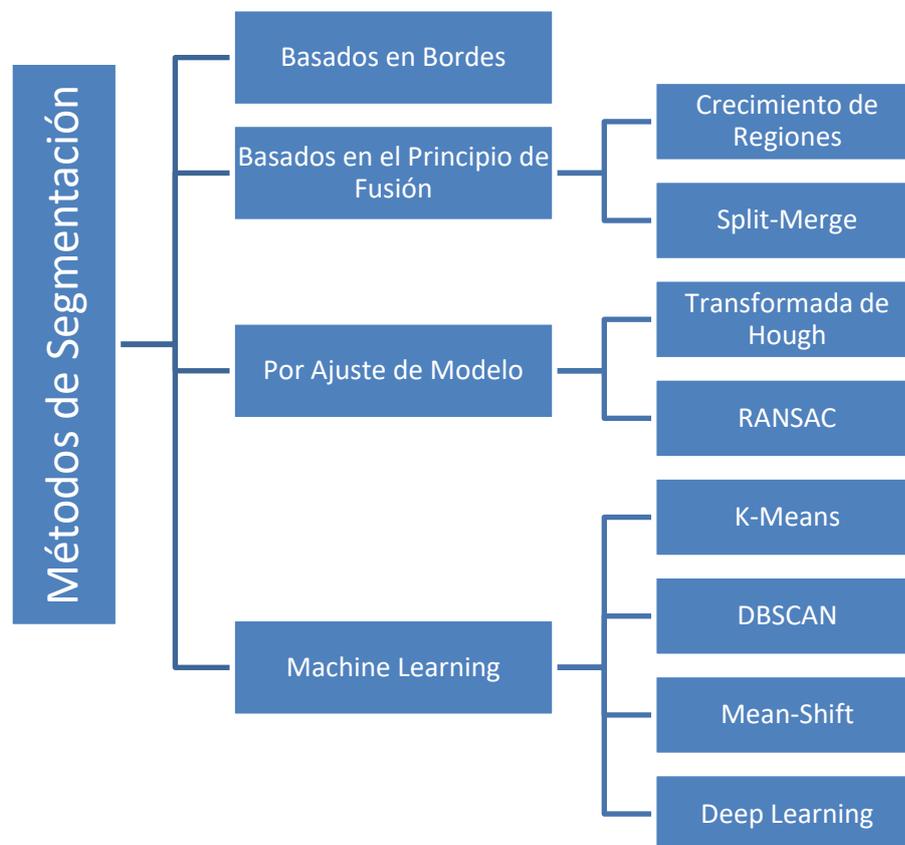


Figura 2.2.- Clasificación de los métodos de segmentación



2.4.1.- Métodos basados en bordes

La segmentación basada en bordes es un método de segmentación que consiste en identificar los contornos de los objetos. Dichos contornos indican la existencia de transiciones significativas en la distancia o geometría tridimensional de la nube de puntos, mostrando así los límites entre objetos o regiones de esta.

Los algoritmos de segmentación basados en bordes consisten en dos etapas:

- 1) La detección de bordes según una umbralización para separar unas regiones de otras. Las propiedades de umbralización más utilizadas son las normales, gradientes, curvaturas principales o derivadas de orden superior.
- 2) La agrupación de puntos dentro de los límites establecidos para obtener los segmentos finales.

2.4.2.- Métodos basados en el principio de fusión

Crecimiento de Regiones

Besl y Jain [7] proponen un método basado en el crecimiento de regiones. Este método comienza con la selección de una serie de puntos iniciales, llamados puntos semilla, que actúan como punto de partida para la formación de regiones.

El algoritmo examina los puntos vecinos de cada semilla evaluando si estos cumplen ciertos criterios de similitud preestablecidos, como el color, la normal o cualquier otra característica relevante. Si un vecino cumple con estos criterios, se incorpora a la región en crecimiento.

Este proceso se repite hasta que se alcanza un punto en el que los nuevos vecinos ya no cumplen con los criterios de similitud, lo que indica que la región ha llegado a su límite. En este momento, se detiene la expansión de la región, y se repite el proceso para las demás semillas.



La efectividad de este método depende principalmente de una correcta selección de los puntos semilla ya que afectará directamente al resultado final pudiendo llegar a provocar segmentaciones incompletas o la inclusión de ruido en las regiones.

Los umbrales utilizados para decidir si un punto debe incorporarse a una región también son parámetros cruciales en la segmentación por crecimiento de regiones. Una selección incorrecta de estos umbrales puede ocasionar sobresegmentación (división de un conjunto de datos en más regiones de las deseadas) o infrasegmentación (agrupación de regiones que deberían ser distintas o separadas), afectando a la calidad y consistencia de la segmentación.

Split-Merge

El método “Split-Merge” (Dividir y Fusionar) es una técnica de segmentación que se utiliza principalmente en la segmentación de imágenes, aunque también se pueden realizar adaptaciones para aplicarlo a nubes de puntos tridimensionales para segmentar los datos en regiones coherentes.

Este algoritmo comienza con una única región que engloba a todo el espacio y se va dividiendo en subregiones cada vez más pequeñas siguiendo un criterio de homogeneidad, es decir, si una región no cumple ese criterio, se divide en k regiones de menor tamaño. Este proceso se repite hasta que todas las regiones cumplan el mismo criterio o hasta que se llegue a un tamaño de región mínimo fijado. A este primer paso del algoritmo se le llama “Split”.

El siguiente paso, “Merge”, parte de un espacio ya dividido en varias subregiones, las cuales se van analizando junto con regiones vecinas para comprobar si son lo suficientemente homogéneas para ser fusionadas en una sola región. Nuevamente, se utilizan ciertos criterios para determinar si las regiones son lo suficientemente similares para fusionarlas. Si se cumplen los criterios de fusión, las regiones se combinan en una sola región más grande. Esta fase se repite hasta que no se puedan juntar más regiones vecinas o hasta que se cumpla un número máximo de iteraciones.



2.4.3.- Métodos por ajuste de modelo

Los métodos de segmentación por ajuste de modelo buscan detectar y agrupar puntos que pertenezcan a la misma estructura geométrica mediante el ajuste de un modelo matemático que describa esa estructura. Algunos de los modelos que se suelen utilizar son planos, líneas, esferas, cilindros, cubos, etc. Los algoritmos más utilizados son la Transformada de Hough y RANSAC.

Transformada de Hough

La transformada de Hough, presentada por Paul Hough en 1962 [8], es una técnica que se utiliza comúnmente en el procesamiento de imágenes para detectar formas geométricas, como líneas, círculos o elipses. Esta técnica se suele adaptar para su uso en la segmentación de nubes de puntos tridimensionales con el objetivo de detectar planos en el espacio.

La Transformada de Hough en 3D utiliza un espacio de parámetros de cuatro dimensiones (A, B, C, D) para caracterizar y representar la ecuación de un plano ($Ax + By + Cz + D = 0$). Por cada punto en la nube, se generan todas las posibles combinaciones de planos que podrían pasar por ese punto, lo que se traduce en una acumulación de votos en el espacio de parámetros.

Los puntos que están cerca de un mismo plano en el espacio votarán por los mismos parámetros (A, B, C, D) correspondientes al plano que probablemente comparten. Finalmente, se buscan regiones con alta densidad de votos, lo que indica la presencia de un plano en la nube de puntos.

RANSAC

Random Sample Consensus (RANSAC) [9] es un método comúnmente utilizado en el procesamiento de datos tridimensionales para estimar modelos a partir de una nube de puntos ruidosa. Su objetivo se basa en encontrar un modelo que mejor se ajuste a una parte de los datos, ignorando los valores atípicos (ruido) que pueden afectar en la estimación.



El método se basa en el supuesto de que los datos consisten en una combinación de “inliers” y “outliers”. Los primeros siguen una distribución que se puede explicar mediante un conjunto específico de parámetros de un modelo. En cambio, los “outliers” son datos que se desvían significativamente de la distribución esperada o del modelo. [10]

El algoritmo RANSAC consiste en realizar dos pasos repetidos de forma iterativa. En el primer paso, se generan hipótesis sobre la forma o la estructura de los datos. Esto se hace tomando un subconjunto mínimo de n puntos de manera aleatoria. El número n es el mínimo de puntos necesarios para estimar un modelo de forma única, por ejemplo, si se seleccionaron 3 puntos, se estimarán los parámetros de un plano que mejor se ajusten a esos puntos.

El segundo paso consiste en verificar cuántos puntos restantes de la nube de puntos se ajustan bien al modelo (“inliers”). Este paso se realiza comparando cada punto con el modelo y se verifica si está dentro de cierto margen de error. Por ejemplo, se puede verificar que un punto pertenece al modelo de un plano calculando la distancia a dicha superficie y la diferencia angular entre la normal del punto y la del plano. Comparando estos dos parámetros con unos umbrales previamente fijados se determina si el punto pertenece o no al modelo.

2.4.4.- Machine Learning

El machine learning, o aprendizaje automático, es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y mejorar automáticamente a partir de la experiencia. En lugar de ser programadas explícitamente para realizar una tarea específica, estos algoritmos utilizan datos para identificar patrones, aprender de ellos y tomar decisiones o realizar predicciones.

Existen 3 tipos de machine learning en función de la forma en que el modelo aprende de los datos:



- Aprendizaje Supervisado: Se alimenta al modelo con un conjunto de datos etiquetados, donde se conocen las entradas y las salidas esperadas. El modelo aprende a hacer predicciones o a clasificar nuevas instancias basándose en ejemplos pasados.
- Aprendizaje No Supervisado: El modelo trabaja con datos no etiquetados, y su objetivo es encontrar patrones intrínsecos, estructuras o características relevantes sin la guía de salidas conocidas. En este caso, el modelo busca agrupar los datos en categorías o identificar relaciones sin tener información previa sobre las clases. Una de las metodologías clave dentro del aprendizaje no supervisado es el clustering, que se enfoca en la segmentación de datos en grupos coherentes o clústeres, donde los elementos dentro de cada grupo son similares entre sí en comparación con aquellos en otros grupos.
- Aprendizaje por Refuerzo: El modelo aprende a través de interacciones con un entorno. Toma decisiones y recibe retroalimentación en forma de recompensas o castigos, buscando maximizar una señal de recompensa a lo largo del tiempo. Este enfoque se asemeja al aprendizaje humano en el sentido de que el modelo aprende a través de la experiencia y la retroalimentación de sus acciones, mejorando su rendimiento a lo largo del tiempo.

Dentro de esta rama se encuentran varios de los métodos más relevantes y sofisticados para la segmentación de nubes de puntos. A continuación, se explican algunas de estas técnicas.

K-Means

El algoritmo K-Means es una técnica de clustering propuesto por primera vez por James MacQueen en 1967 [11]. Este método de segmentación se basa en dividir un

conjunto de datos en un número predeterminado de grupos (“k”) basados en similitudes entre sus características.

Dado un conjunto de datos, el algoritmo comienza seleccionando aleatoriamente “k” centroides iniciales. Posteriormente, calcula las distancias de cada punto a cada uno de los centroides y se asigna al más cercano. Una vez clasificados todos los puntos, se recalcula el centroide de cada grupo y se vuelve a iterar. El algoritmo finaliza cuando los centroides convergen, es decir, cuando su variación de posición respecto a la iteración previa es inferior a un umbral fijado.

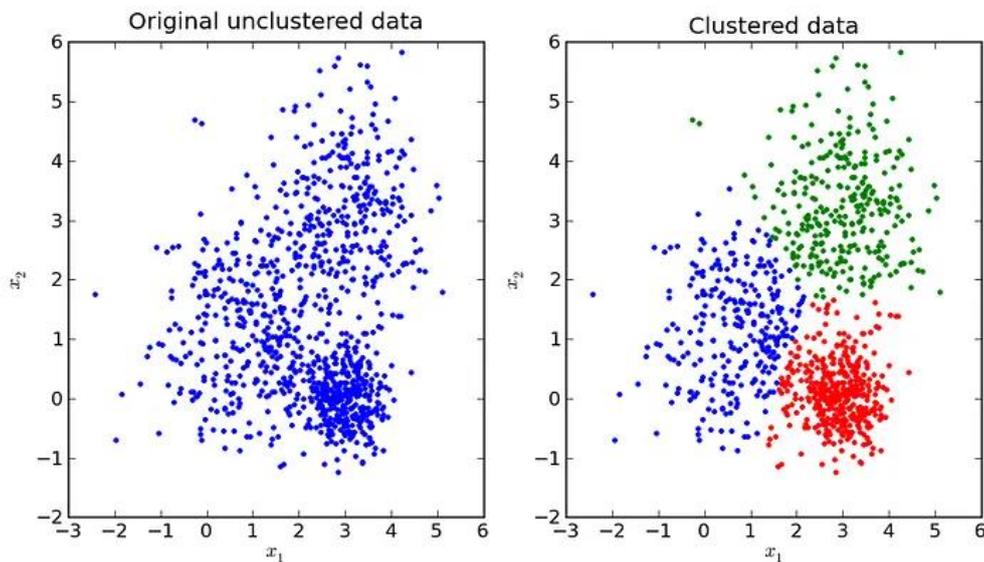


Figura 2.3.- Ejemplo de aplicación de k-means

DBSCAN

Otro algoritmo que pertenece a la rama del clustering es DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [12], el cual se basa en la densidad de puntos y la conectividad espacial para agrupar puntos similares en clústeres.

DBSCAN requiere la definición de dos parámetros iniciales: "épsilon" (ϵ), que determina la distancia máxima entre puntos para considerarlos vecinos, y "MinPts", que establece el número mínimo de puntos dentro de un vecindario para formar un cluster.

El proceso comienza seleccionando un punto aleatorio del conjunto de datos. Luego, identifica todos los puntos que están a una distancia ϵ o menos del punto inicial, formando un vecindario. Si este vecindario contiene al menos MinPts puntos, se considera como parte de un cluster. Se expande el cluster agregando puntos vecinos y sus respectivos vecindarios si cumplen con los criterios de densidad establecidos. Esto continúa hasta que no se pueden agregar más puntos al cluster. Los puntos que no cumplen con los criterios de densidad se clasifican como ruido.

Mean-Shift

El algoritmo mean-shift, presentado por Cheng en 1995 [13], es un método de clustering basado en la densidad y en la búsqueda de modas o máximos locales en la distribución de los datos.

El proceso iterativo de Mean-Shift comienza seleccionando un punto inicial, ya sea de forma arbitraria o basado en criterios específicos, junto con la definición de un "kernel de ponderación". Este kernel es una especie de ventana que se posiciona alrededor del punto inicial, y asigna pesos a los puntos dentro de ella basándose en su distancia al centro: aquellos más cercanos reciben mayores pesos que los más alejados.

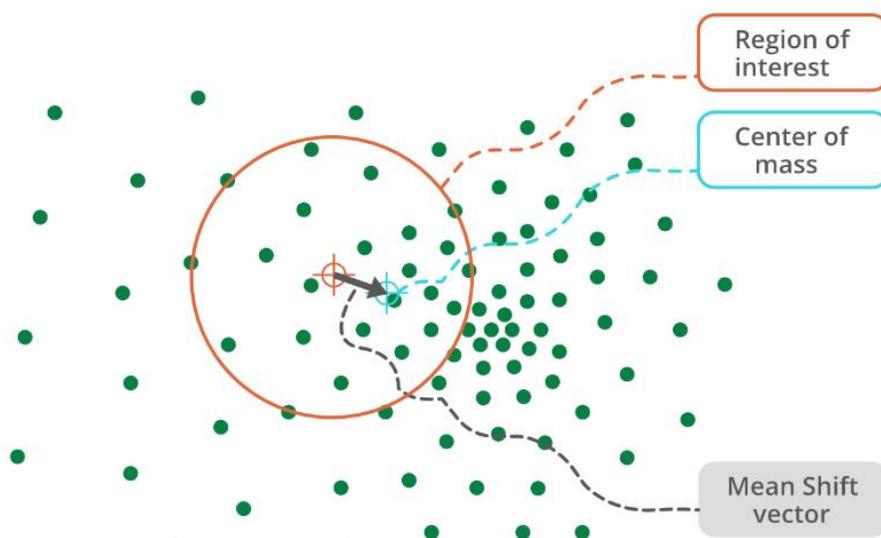


Figura 2.4.- Proceso del algoritmo Mean-Shift



Basándose en estos pesos, se calcula una media ponderada que sirve para ajustar la posición actual hacia la dirección donde se concentra la mayor densidad de puntos dentro del kernel. Este proceso, mostrado en la Figura 2.4, continúa hasta que la posición ya no varía significativamente, lo que indica la convergencia hacia un máximo local de la nube de puntos. Este procedimiento se aplica a cada punto de la nube hasta que convergen hacia centroides particulares.

El resultado final es la agrupación de los puntos que convergen hacia el mismo centroide, formando así una serie de clústeres. Este enfoque permite identificar estructuras complejas y variadas en la nube de puntos 3D, proporcionando una segmentación efectiva al agrupar puntos similares en clústeres con densidades y formas diversas. Además, al no requerir la especificación previa del número de clústeres, el Mean-Shift es útil en escenarios donde la cantidad y forma de los clústeres no son conocidos a priori.

Deep Learning

El Deep Learning, una rama avanzada del Machine Learning, se centra en entrenar redes neuronales profundas compuestas por múltiples capas para abordar tareas complejas y jerarquizadas en los datos [14]. En el contexto de las nubes de puntos, estas redes se emplean para identificar objetos y regiones, comenzando con la recolección de los datos ya etiquetados. Estas etiquetas asignan a cada punto de la nube su correspondencia con objetos o áreas específicas.

Para implementar la técnica del Deep Learning es necesario seleccionar una arquitectura de red adecuada para la tarea de segmentación, entrenándola con los datos etiquetados para que pueda predecir con precisión las etiquetas de otros datos similares. La optimización de la red implica validar su rendimiento utilizando datos separados y ajustar parámetros clave, como la tasa de aprendizaje y la estructura de la red, para mejorar su capacidad predictiva.



Una vez entrenada y validada, la red se somete a pruebas con nuevos datos no etiquetados para evaluar su capacidad de segmentación en un entorno real. Esta etapa de prueba es esencial para comprobar la efectividad y precisión de la red neuronal antes de aplicarla a nubes de puntos tridimensionales en situaciones prácticas o en tiempo real. Este enfoque brinda la capacidad de identificar objetos y regiones dentro de nubes de puntos de manera eficiente y precisa.



3. Estado del arte

La segmentación de nubes de puntos es una de las tareas fundamentales en el ámbito de la visión por computador, con aplicaciones en robótica, vehículos autónomos, escaneo 3D, inspección industrial...Este proceso, como ya se ha mencionado anteriormente, implica dividir una nube de puntos 3D en regiones coherentes con unas características similares, lo que permite identificar objetos y extraer información de ciertos segmentos.

En las últimas décadas, esta especialidad ha sido un área de investigación en ingeniería donde se han desarrollado muchos métodos y técnicas de segmentación. Estos métodos han ido evolucionando junto con el avance de la tecnología desde el uso de la geometría y la topología, hasta el uso de información de color y textura, y más recientemente, la implementación del aprendizaje profundo.

Tras realizar una investigación minuciosa sobre los métodos y enfoques empleados en el ámbito de la segmentación de nubes de puntos, se han seleccionado los siguientes métodos, considerados los más relevantes y útiles para el objetivo de este trabajo: Crecimiento de Regiones, Split and Merge, K-Means, Mean-Shift y DBSCAN.

A continuación, se estudiarán detalladamente una serie de metodologías propuestas por investigadores y expertos en este campo que hayan empleado alguna de las técnicas mencionadas anteriormente.

3.1.- CRECIMIENTO DE REGIONES

Existe una gran cantidad de proyectos que utilizan la técnica de crecimiento de regiones para encontrar puntos que pertenecen a un mismo objeto e ir agrupándolos de forma progresiva.



Michela Bertolotto et al. [15] se enfrentan al desafío de la segmentación de nubes de puntos mediante un enfoque innovador basado en la detección de planos en el espacio tridimensional. Para lograr esto, el estudio emplea una metodología que comienza con la voxelización de la nube utilizando una estructura tipo 'octree', estrategia destinada a simplificar y reducir el tamaño de los datos a procesar. Posteriormente, se procede a calcular el vector normal y el valor residual de cada descomposición espacial con respecto al plano de ajuste del voxel correspondiente. Finalmente, estas características son utilizadas para ejecutar un proceso de crecimiento de regiones, permitiendo así agrupar los voxels vecinos que cumplen con los criterios de similitud establecidos.

3.2.- SPLIT AND MERGE

Otra técnica similar y bastante utilizada es el “Split and Merge”, el cual ha sido considerado una herramienta crucial en el ámbito de la segmentación de nubes de puntos. Reconocido por su eficiencia en la partición y unión de áreas coherentes, este enfoque ha sido objeto de exhaustivos estudios en el campo de la segmentación de datos tridimensionales.

Los investigadores Yun-Ting Su, James Bethel y Shuowen Hu [16], propusieron un método basado en Split and Merge para realizar una segmentación de tuberías, recipientes y paredes a partir de datos LiDAR terrestres tomados en plantas industriales. El algoritmo propuesto comienza con la descomposición de la nube de puntos mediante un “octree” [17]. A continuación, realiza un split dividiendo cada voxel en componentes, siendo un componente un conjunto de puntos lo suficientemente cercanos. Finalmente, realiza el merge de los componentes mediante criterios de proximidad, orientación y curvatura de los objetos. Este último criterio resulta ser muy efectivo para determinar si un objeto es cilíndrico o plano, lo que les permite fusionar únicamente los componentes cercanos que sean del mismo tipo.



3.3.- AJUSTE DE MODELOS: RANSAC

La implementación de algoritmos de ajuste de modelos matemáticos es otro de los métodos más comunes en la segmentación de nubes de puntos. Un ejemplo de este tipo de algoritmos, como ya se ha visto anteriormente, es el “Random Sample Consensus” (RANSAC), que se basa en encontrar un modelo que mejor se ajuste a una parte de los datos, ignorando los valores atípicos (ruido) que pueden afectar en la estimación.

F. Yang et al. [18] proponen un enfoque innovador de RANSAC para la segmentación de nubes de puntos. En este método, la nube de puntos se descompone en celdas, y se emplea el algoritmo RANSAC para detectar planos en cada una de ellas. La novedad de este método radica en el uso de un nuevo parámetro para detectar estos planos: la distribución normal de los puntos dentro de las celdas. Esta mejora adaptativa del algoritmo RANSAC, al considerar la estructura celular de la nube de puntos, permite una identificación más exacta y rápida de los planos presentes en los datos tridimensionales capturados.

3.4.- K-MEANS

El método K-means, es una técnica fundamental en el campo de la agrupación y análisis de datos. A lo largo de los años, K-means ha evolucionado y se ha ido adaptando a las demandas cambiantes de la industria, demostrando una gran versatilidad y eficacia.

Una de las mejoras clave ha sido la consideración de la inicialización de centroides. En su versión básica, K-means puede ser sensible a la selección inicial de centroides, lo que puede implicar resultados cambiantes e imprecisos. Métodos como el K-means++ han surgido para mejorar la inicialización de los centroides, ya que se basa en que los centros de agrupación iniciales deben estar lo más alejados posible unos de otros.

Un ejemplo del potencial de K-Means ++ se refleja en un artículo [19], donde utilizan este algoritmo para segmentar nubes de puntos de equipos eléctricos. Para ello,



proponen un método basado en diferentes técnicas que mejoran la eficiencia del algoritmo principal de K-Means ++. En primer lugar, emplean el método del codo (“Elbow Method” [20]) para hallar un número óptimo de clústeres, lo que facilita la tarea de encontrar un número k adecuado para su implementación. Por otro lado, para calcular las distancias entre puntos, utiliza la distancia geodésica en lugar de la euclidiana, lo que se traduce en una mejor representación de las relaciones espaciales entre los puntos. Esta estrategia hace que la segmentación sea más robusta y no dependa de parámetros tan sensibles como el número de clústeres o la posición de los centroides iniciales.

3.5.- MEAN-SHIFT

El algoritmo Mean Shift ha emergido como una técnica poderosa en el ámbito del análisis y la segmentación de datos, especialmente en la percepción y comprensión de patrones en conjuntos de datos complejos. Esta técnica, perteneciente a la familia de Machine Learning, ha ganado popularidad debido a su capacidad para realizar segmentaciones sin suposiciones predefinidas sobre el número de clústeres, adaptándose dinámicamente a la densidad y estructura intrínseca de los datos.

Mean-Shift se aplica principalmente a proyectos e investigaciones donde se trabaja con mallas tridimensionales en lugar de nubes de puntos. Las mallas o “meshes” son una buena forma de representar objetos con detalles superficiales relevantes, aunque suelen requerir mayor nivel de computación. [21]

Un ejemplo de aplicación es el método propuesto por X. Zhang et al. [22], los cuales realizan un proceso de segmentación que comienza con la definición de un núcleo de búsqueda alrededor de cada vértice de la malla. A continuación, mediante el algoritmo mean-shift se ajusta iterativamente dicho núcleo para encontrar la moda de curvatura más cercana. Finalmente realiza un crecimiento de regiones a partir de los vértices con mayor curvatura para agrupar todas las caras con características de curvatura similares.



3.6.- DBSCAN

DBSCAN ha destacado por su capacidad para identificar clústeres de diferentes formas y tamaños en conjuntos de datos con densidades variables, además de su robustez frente al ruido y su independencia en la especificación previa del número de clústeres.

Un ejemplo de su alcance es el trabajo realizado por C. Wang et al. [23], donde emplean una mejora de DBSCAN para segmentar un escaneado LIDAR aéreo y otro terrestre. La mejora del algoritmo radica en la estimación automática del parámetro ϵ , que es la distancia máxima que determina si dos puntos están conectados espacialmente. Dicha estimación se basa en el cálculo de la media de la distancia máxima de los k vecinos más cercanos (KNN) [24] de cada punto semilla. Esta modificación del algoritmo original hace que mejore la precisión y la adaptabilidad de la segmentación. También demuestra gran versatilidad al ser aplicado en diferentes escaneados laser.

4. Método Propuesto

Tras el análisis de las diversas técnicas de segmentación citadas anteriormente, se procederá a describir el método propuesto para abordar la problemática planteada en este trabajo. El método consistirá principalmente en un split de la nube mediante K-means, seguido de un merge iterativo de los clústeres adyacentes en función de sus características.

La elección de esta estrategia se fundamenta en la dificultad de separar de manera precisa objetos adyacentes en una nube de puntos. Al implementar esta metodología, con especial atención a la elección de sus parámetros, se busca lograr la extracción selectiva de los objetos de interés dentro de la nube tridimensional.

Por otro lado, este procedimiento no solo busca resolver la segmentación de manera efectiva para este contexto, sino que también sirve de base para implementar futuras mejoras y adaptaciones. Gracias a la flexibilidad del método se podrán ajustar los distintos parámetros que contiene el algoritmo para resolver otros problemas de segmentación.

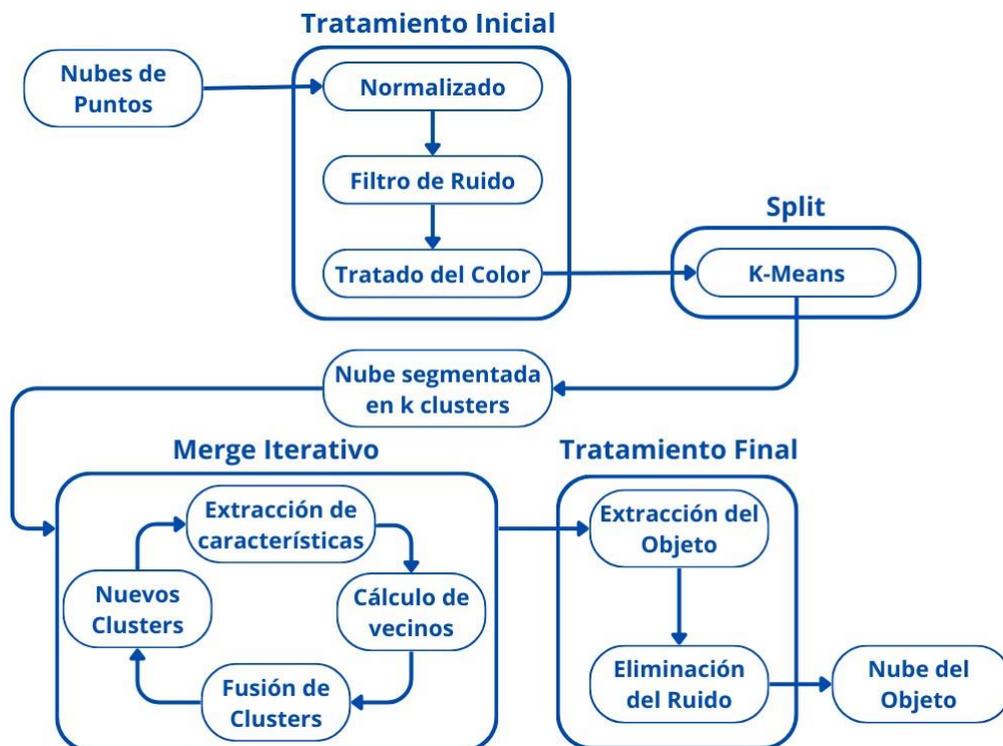


Figura 4.1.- Método Propuesto



El método propuesto, como se muestra en la Figura 4.1, consta de varias etapas de filtrado y procesado de datos. A continuación, se explicarán detalladamente cada una de estas fases y sus motivos de aplicación.

4.1.- TRATAMIENTO INICIAL

El tratamiento inicial de la nube de puntos representa el punto de partida fundamental en cualquier proceso de análisis y segmentación tridimensional. Esta fase inaugural se enfoca en la preparación y el procesamiento de la nube de puntos, con el objetivo de garantizar la calidad de los datos y establecer las bases para un análisis preciso y efectivo.

Este tratamiento comienza con la normalización de los datos, buscando homogeneizar las escalas de las distintas propiedades presentes en la nube de puntos. Este paso busca estandarizar el rango de valores de las características que posteriormente se utilizarán en el proceso de segmentación. En este contexto, se ha empleado una escala uniforme de $[0, 1]$, asegurando que todas las propiedades relevantes (distancias, normales y color) estén contenidas dentro de este intervalo. Esta estandarización garantiza la coherencia en la representación de los atributos, facilitando así una segmentación precisa y consistente basada en estas propiedades.

Una vez normalizados los datos, resulta fundamental abordar la eliminación del ruido inherente a estos conjuntos. Este paso es fundamental ya que la presencia de datos atípicos puede distorsionar significativamente la interpretación de la información. Para ello se ha utilizado un filtro de media aritmética, una técnica comúnmente utilizada en el procesamiento de imágenes y datos. Este filtro consiste en recorrer cada punto de la nube calculando el valor promedio de los vecinos que se encuentren dentro de una región determinada y asignarle este promedio a dicho punto. Con esto conseguimos reducir la dispersión de los datos, suavizando así la nube de puntos.



Por último, para facilitar la segmentación es importante que haya una distinción clara de los colores de cada objeto. Con este propósito, se ha considerado aumentar la saturación de los colores presentes en la nube de puntos, lo que favorecerá su distinción. Esta tarea se ha abordado mediante la transformación de los valores RGB de la nube de entrada a valores HSV. Esta conversión nos proporciona la capacidad de amplificar la saturación al multiplicar el componente de saturación (S) por un factor de intensidad.

4.2.- SPLIT

El siguiente paso del proceso implica la segmentación de la nube de puntos mediante el algoritmo K-Means. Como ya se ha explicado anteriormente, K-means busca particionar un conjunto de datos en “k” grupos, conocidos como clústeres. Su objetivo principal es asignar cada punto de la nube a un clúster de manera que la suma de las distancias al cuadrado entre los puntos y el centroide del clúster sea mínima.

Para aumentar la efectividad y precisión de K-Means, se ha realizado una serie de adaptaciones mediante la incorporación de pesos relativos al color, la normal y la distancia. A diferencia del algoritmo original, que opera únicamente basándose en la distancia euclidiana entre puntos, esta adaptación pondera cada característica para influir en el proceso de agrupamiento. La inclusión de estos pesos ofrece una consideración más precisa de las propiedades visuales, la orientación espacial y la distancia en la formación de los clústeres.

Por otro lado, para abordar la selección de los centroides iniciales, se ha implementado una estrategia alternativa. En lugar de elegir los centroides de manera aleatoria, se ha dividido el conjunto de datos en “k” segmentos y se han seleccionado los puntos que se encuentran entre cada segmento como centroides iniciales. Por ejemplo, en el caso de “k=10” con un conjunto de datos de 1000 puntos, los centroides iniciales serían los puntos situados en los índices 100, 200, y así sucesivamente, hasta llegar al punto 1000. Esta técnica proporciona centroides iniciales más representativos y mejor distribuidos, lo

que mejora la eficacia y evita cambios en la salida del algoritmo cada vez que este se ejecuta.

Es crucial destacar que el procedimiento de split está pensado para sobresegmentar la nube de puntos, es decir, fragmentarla en un conjunto más amplio de clústeres de lo estrictamente necesario. Esta estrategia se implementa utilizando valores elevados de "k", lo que garantiza la formación de clústeres lo más homogéneos posible.

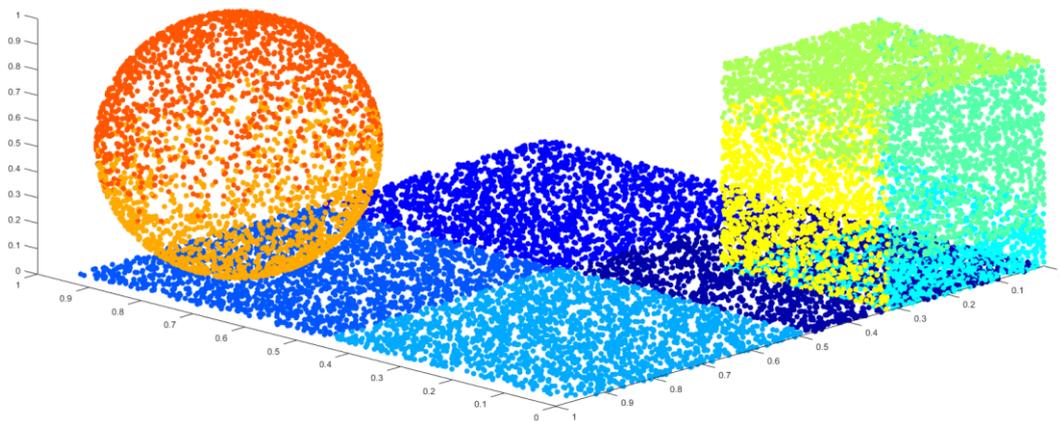


Figura 4.2.- Ejemplo de una nube de puntos sobresegmentada en la que cada cluster es representado por un color diferente.

4.3.- MERGE ITERATIVO

Con el objetivo de realizar una fusión coherente de los clústeres, se ha implementado un algoritmo de Merge Iterativo. Este algoritmo está basado en una serie de procedimientos que permiten la identificación y fusión de los clústeres vecinos de manera progresiva y controlada.

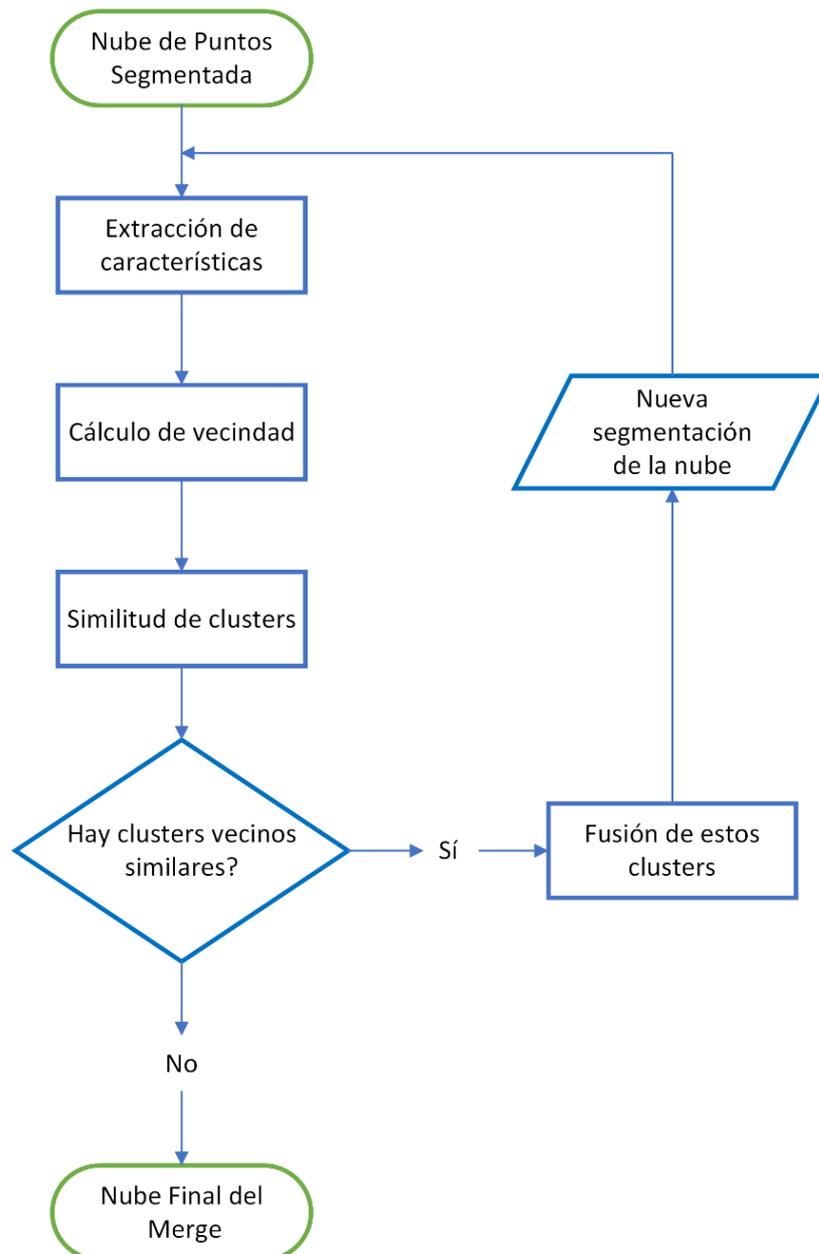


Figura 4.3.- Diagrama de flujo del Merge Iterativo



El proceso iterativo comienza calculando las características de color y normal de cada clúster. Este enfoque se centra en determinar valores representativos que describan las propiedades visuales y geométricas predominantes dentro de cada grupo de puntos. Para obtener dichas características, se realiza una media de los valores de color y de normal de los puntos del cluster. Estos valores medios son esenciales para la evaluación y comparación entre clústeres vecinos, facilitando así el proceso de fusión.

El siguiente paso implica el cálculo de la vecindad de cada cluster con el objetivo de analizar únicamente la similitud entre grupos de puntos adyacentes. La determinación de esta vecindad se fundamenta en dos parámetros: un umbral de distancia y un umbral de puntos próximos. El umbral de distancia establece la separación máxima permitida entre dos puntos de grupos diferentes, definiendo si son cercanos o no. Si la cantidad de puntos cercanos entre dos clústeres supera el umbral establecido, estos se consideran vecinos.

La fusión de los clústeres se decide analizando cada cluster con cada uno de sus vecinos mediante el cálculo de dos valores de similitud ponderados: la similitud de color y la similitud de orientación. Ambas similitudes se calculan utilizando la misma fórmula con sus respectivos pesos:

$$similitud = peso \times |media_{cluster} - media_{vecino}| \quad (4.1)$$

Para determinar si un cluster vecino debe de ser fusionado con el cluster que se está analizando, se aplica un criterio basado en la suma de la similitud de color y la similitud de orientación. Si esta suma es menor que un umbral de similitud predefinido, entonces se procede a la fusión:

$$(similitud_{color} + similitud_{normal}) < umbral_{similitud} \quad (4.2)$$

La flexibilidad de este procedimiento es fundamental, ya que permite ajustar los pesos de ambas características y el umbral de fusión, adaptándose así a diferentes conjuntos de datos, escenarios y objetivos específicos. Estos ajustes proporcionan una capacidad de adaptación que garantiza una mayor precisión en la identificación y fusión de



clústeres, optimizando así el proceso de segmentación tridimensional para diversos análisis de nubes de puntos.

Si en la etapa previa de análisis de similitud se ha decidido fusionar dos o más clústeres, se genera una nueva segmentación de la nube de puntos, lo que implica la modificación de los clústeres y sus características. En este escenario, el proceso de fusión se reinicia para esta nueva configuración, evaluando nuevamente la existencia de clústeres vecinos que cumplan con las condiciones de similitud establecidas. Este ciclo de evaluación y posible fusión se repite hasta que, durante el análisis de similitud, no se encuentren clústeres que satisfagan estas condiciones.

4.4.- TRATAMIENTO FINAL

Después de completar el proceso de segmentación utilizando la técnica de "Split and Merge", se procede al tratamiento final de la nube de puntos. En este contexto, donde la nube de puntos se ha segmentado adecuadamente según nuestro objetivo, se selecciona el índice del clúster que contiene la región de interés, en este caso, el clúster que representa el casco del barco. Este conjunto específico de puntos se utiliza para crear una nueva nube de puntos que se someterá a un último filtro para mejorar su calidad.

El último filtro aplicado a esta nueva nube de puntos se centra en la eliminación del ruido, el cual proviene mayormente de partes de otros objetos incorporadas en el clúster de interés. Para ello, se ha diseñado un filtro de densidad, cuya función principal consiste en eliminar puntos que, dentro de un radio definido, están rodeados por una cantidad de puntos inferior a un umbral establecido. La lógica detrás de este filtro es eliminar los puntos que se encuentran en los márgenes o límites de la nube de puntos y que no presentan una densidad suficientemente alta de puntos vecinos. Estos puntos periféricos suelen representar inclusiones no deseadas de otros objetos o elementos irrelevantes para el área de interés, por lo que eliminarlos contribuye a mejorar la representación del objeto.

5. Experimentación

En esta sección se pretende detallar el proceso por el cual se ha llevado a cabo las pruebas, evaluaciones y análisis del algoritmo propuesto. Para el desarrollo del algoritmo de segmentación y la realización de las pruebas se ha utilizado un ordenador equipado con un procesador Intel Core i7-9750H y una capacidad de memoria RAM de 16GB. El algoritmo se ha desarrollado en el entorno de programación Matlab, utilizando la versión “R2023b”. La utilidad y versatilidad de Matlab se han potenciado con la incorporación específica de la “Computer Vision Toolbox” [25], una librería que contiene herramientas especializadas para el manejo y análisis de nubes de puntos e imágenes. Por otro lado, para la visualización de los resultados obtenidos en las pruebas se ha utilizado la versión “2022.02” de Meshlab [26], un software de código abierto que ofrece una gran variedad de herramientas para limpiar, editar, visualizar y manipular modelos 3D. Este programa también ha sido muy útil para agilizar el cálculo de las normales.

En primer lugar, se ha considerado necesario realizar una serie de nubes de puntos artificiales antes de abordar el problema real. Estas nubes de puntos artificiales han sido creadas para facilitar el desarrollo del código, ya que al realizar pruebas con estas maquetas es más sencillo detectar y solucionar fallos. Además, mediante la aplicación del código a estas nubes de puntos artificiales, se pueden identificar áreas problemáticas y realizar ajustes en el algoritmo de forma más eficiente, antes de aplicarlo a datos reales.

Las maquetas han sido configuradas para abarcar desde estructuras simples hasta representaciones progresivamente más elaboradas, acercándose a maquetas de barcos con mayor detalle y complejidad. A continuación, se detallará el procedimiento empleado para generar dichas nubes de puntos.

5.1.- NUBES DE PUNTOS ARTIFICIALES

La creación de las nubes de puntos artificiales se ha realizado mediante la elaboración de elementos básicos (planos, esferas, cilindros...), los cuales se han combinado para formar estructuras más complejas y elaboradas. Esta estrategia ha sido diseñada específicamente para facilitar la evaluación de la efectividad del algoritmo propuesto. En este contexto, se ha desarrollado un algoritmo para calcular el porcentaje de puntos que han sido asignados correctamente. Al tener un conocimiento detallado de los puntos que componen cada objeto simple dentro del espacio, será posible determinar qué datos han sido clasificados correctamente por el algoritmo y en cuáles ha fallado. Este análisis detallado del rendimiento del algoritmo proporciona una comprensión más profunda de su eficacia y contribuye significativamente a la mejora continua del mismo.

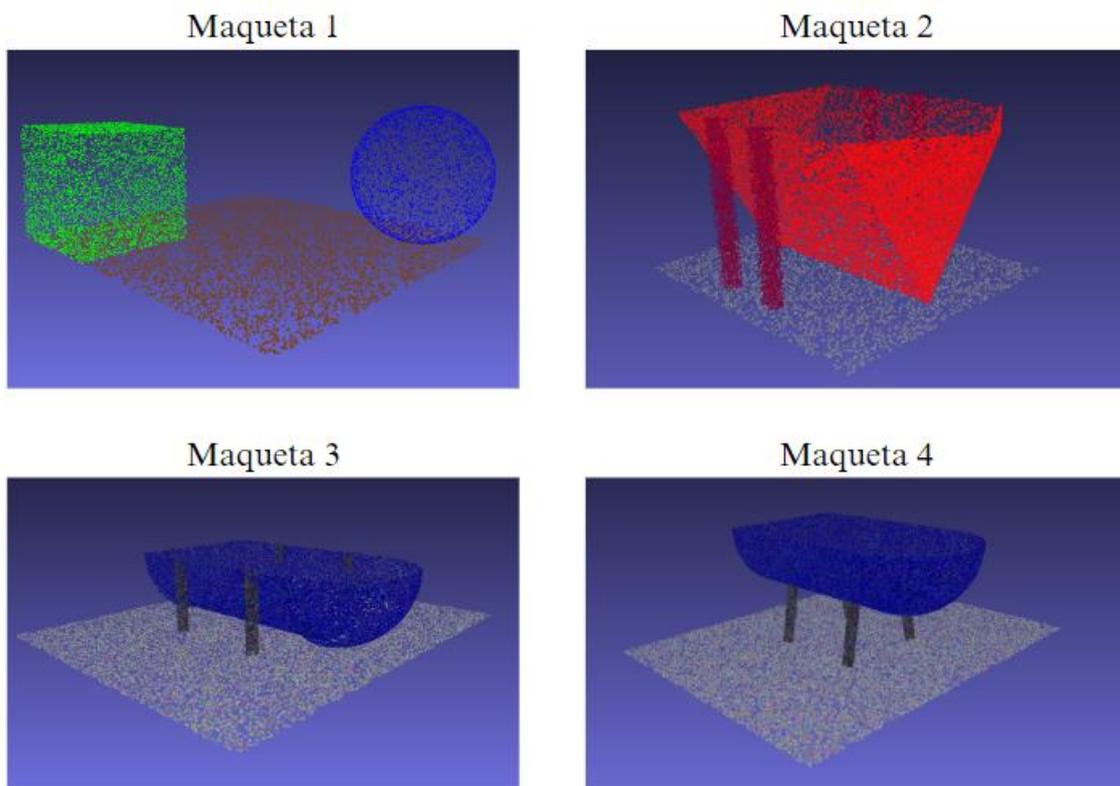


Figura 5.1.- Nubes de puntos artificiales



Por otro lado, con el objetivo de mejorar la semejanza entre las nubes de puntos artificiales y las reales, se han asignado colores distintivos a cada uno de los objetos de la nube. Además, se han introducido variaciones de color en algunos puntos para simular el ruido que puede haber en las nubes de puntos reales.

Como se puede apreciar en la Figura 5.1, las distintas maquetas están ordenadas de menor a mayor complejidad de segmentación, comenzando con estructuras básicas y colores bien diferenciados, hasta llegar a representaciones más reales y desafiantes.

La primera maqueta representa una configuración elemental que consta de un plano representando el suelo, sobre el cual se sitúan una esfera y un cubo. Estos dos objetos simples están claramente definidos por sus formas distintivas, con la esfera caracterizada por su superficie curva y el cubo por sus caras planas. Además, cada objeto está asociado a un color, lo que facilita su diferenciación. La simplicidad de esta nube facilita la identificación de los objetos, sirviendo como punto de partida para comenzar a aplicar el método propuesto.

La maqueta número dos está diseñada para representar el casco de un barco apoyado sobre un suelo plano y sujetado por 4 postes verticales. Con esta estructura se pretende simular las nubes de puntos reales con elementos planos para simplificar su segmentación. Por otro lado, se han asignado colores similares al casco del barco y a los postes, lo que dificulta la diferenciación entre estos elementos. Esta estrategia busca desafiar la capacidad del algoritmo de segmentación para discernir y separar objetos adyacentes con colores prácticamente idénticos. Finalmente, es importante mencionar que, a las maquetas restantes se les ha agregado un efecto de ruido al color de algunos puntos, emulando las variaciones visuales presentes en las nubes reales.

Las dos últimas maquetas presentan una evolución en la complejidad con la introducción de elementos curvos. Estos elementos incluyen medio cilindro replicando el centro del barco, un cuarto de esfera para la proa y 4 cilindros verticales para los postes. Con esto se busca reflejar con mayor fidelidad la forma de los cascos de barcos reales, desafiando el algoritmo de segmentación con formas más elaboradas.



En la última maqueta, los postes están posicionados por debajo del casco del barco, replicando la disposición observada en algunas nubes de puntos que representan barcos fuera del agua sostenidos por puntales. Esta disposición es la que se suele observar con mayor frecuencia en entornos náuticos reales.

5.2.- NUBES DE PUNTOS REALES

Las nubes de puntos reales han sido capturadas por estudiantes de doctorado y grado de la Escuela de Marina de Gijón mediante un Iphone 14. Para ello han utilizado la aplicación móvil Polycam [27] que emplea la cámara del dispositivo para generar representaciones 3D del entorno a partir de múltiples imágenes tomadas desde diferentes ángulos. Esta aplicación permite exportar las capturas en formato “.ply” [28], lo que nos ofrece la posibilidad de trabajar con ellas desde Matlab y Meshlab.

Para realizar las pruebas destinadas a la validación del método propuesto en este trabajo, han sido seleccionadas cuatro nubes de puntos distintas, las cuales se muestran en la Figura 5.2. Estas representaciones sirven como ejemplos de los distintos tipos de embarcaciones y soportes que se pueden encontrar en los puertos.

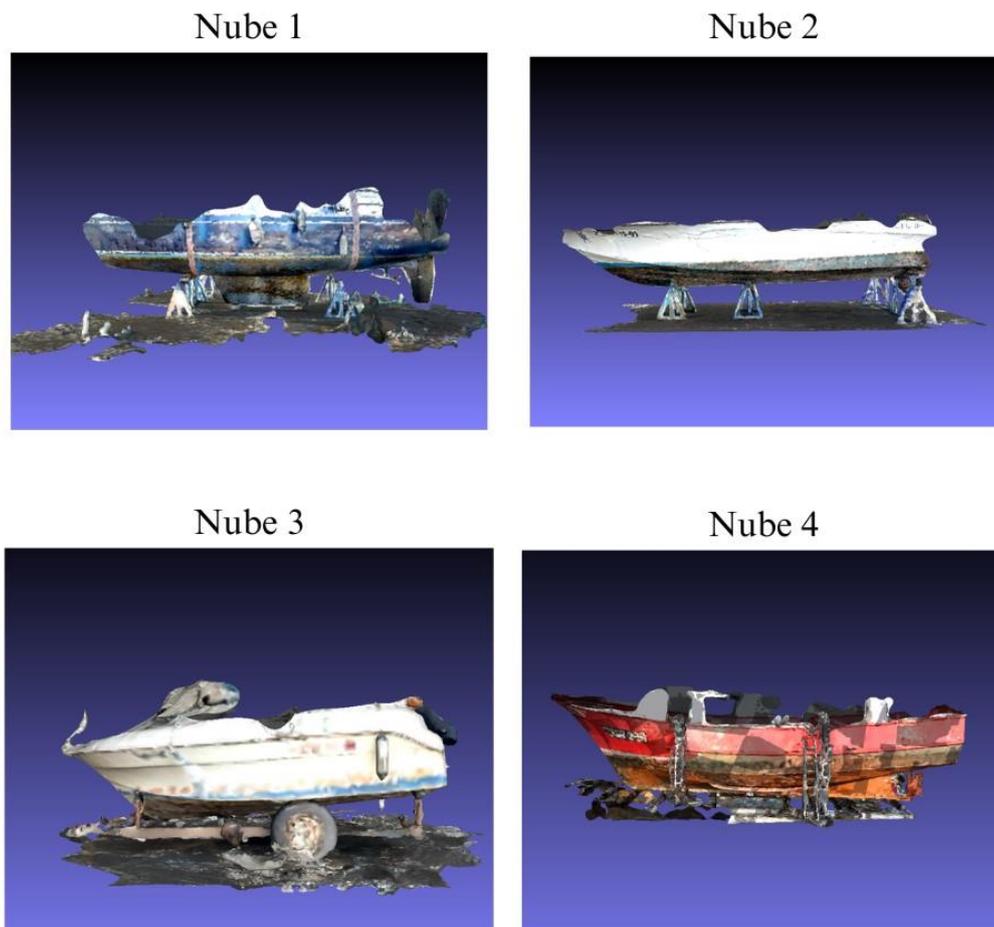


Figura 5.2.- Nubes de Puntos Reales



En primer lugar, la nube real 1 contiene una representación de un velero de 10 metros de eslora sobre un soporte formado por caballetes. Esta embarcación contiene una quilla muy pronunciada y con un color similar al del suelo, lo que puede dificultar su segmentación.

En la segunda nube de puntos, se presenta una lancha motora de 8 metros de longitud, descansando una vez más sobre caballetes. En esta ocasión, la parte superior de la embarcación, conocida como la "obra muerta", exhibe un tono blanco que la distingue notablemente del resto, mientras que la obra viva presenta un tono más oscuro y azulado, similar al de los propios caballetes.

La tercera nube representa otra lancha motora, en este caso de 5 metros de eslora, apoyada sobre un soporte de tipo remolque. Este soporte dificulta la captura completa del casco del barco debido a su proximidad, lo que puede influir de forma negativa en el proceso de segmentación.

Por último, la cuarta nube real ilustra un barco pesquero con una eslora de 16 metros. Esta representación se distingue por su complejidad, resaltando la mínima separación entre la embarcación y los soportes. En este caso, la embarcación descansa sobre rieles, ubicados debajo de la misma y empleados para facilitar su extracción o introducción en el agua. La presencia de estos rieles, en combinación con la sujeción adicional proporcionada por postes laterales, añade complejidad al proceso de extraer el casco del barco.



5.3.- PARÁMETROS DEL ALGORITMO

Los parámetros de un algoritmo representan los ajustes fundamentales que moldean su comportamiento y rendimiento. En el contexto de la segmentación de nubes de puntos 3D, la selección y ajuste preciso de estos parámetros juega un papel crítico en la eficacia y precisión del algoritmo. Esta sección se enfoca en la comprensión detallada de cada uno de los parámetros del método propuesto:

- **Kernel de Filtro de Media (KFM):** es el kernel utilizado para aplicar el filtro de media aritmética sobre las normales y colores. A medida que se aumenta el tamaño del kernel, se logra una mayor suavidad en las variaciones de estas propiedades.
- **Factor de Saturación (FS):** Este parámetro se emplea para incrementar o reducir la saturación del color en la nube de puntos.
- **Número de clústeres (K):** Es el número de clústeres finales que se desean obtener al aplicar K-Means.
- **Peso del Color (ω_C), Peso de Normal (ω_N), Peso de Distancia (ω_D):** Estos tres parámetros determinan la importancia del color, la normal y la distancia al implementar K-Means.
- **Umbral de Distancia (θ_D):** Distancia máxima permitida entre puntos para que sean considerados cercanos.
- **Umbral de Puntos Cercanos (θ_{PC}):** Es la cantidad mínima de puntos entre dos clústeres para que se consideren vecinos.
- **Peso de Color en Similitud (σ_C), Peso de Normal en Similitud (σ_N):** Son los pesos utilizados para calcular la similitud entre clústeres según la Ecuación 4.1.
- **Umbral de Similitud (δ_S):** Es el umbral de similitud para fusionar clústeres utilizado en la Ecuación (3).
- **Radio de Vecindad (r_V):** Es el radio de búsqueda de vecinos alrededor de un punto en particular.
- **Umbral de Densidad (ϵ_D):** Es el número mínimo de vecinos que debe contener un punto para no ser eliminado por el filtro de densidad.



5.4.- PROCEDIMIENTO PARA LA REALIZACIÓN DE LAS PRUEBAS

Al tratarse de un proyecto de investigación, es importante comentar el procedimiento por el cual se han llevado a cabo las pruebas necesarias para determinar y comparar la efectividad del algoritmo al aplicarlo en distintos escenarios. El procedimiento que se ha seguido consta de 2 tipos de pruebas:

- **Pruebas Iniciales:** Se llevaron a cabo aplicando el algoritmo sobre las nubes de puntos artificiales. Estas han servido para evaluar el desempeño inicial del algoritmo en entornos controlados y predefinidos. Se han realizado experimentos múltiples con cada modelo, ajustando progresivamente los parámetros en base a los resultados obtenidos. Este proceso permitió observar cómo el código responde a cada parámetro, lo que ha aportado una mejor comprensión de su comportamiento para abordar las pruebas finales. Por último, es importante mencionar que en estas pruebas iniciales el proceso finaliza con el resultado del merge, es decir, no se ha aplicado la etapa de tratamiento final. Esta decisión se fundamenta en que el objetivo principal de estas pruebas es evaluar exclusivamente la efectividad del proceso de segmentación. Además, al tener la información de qué puntos pertenecen a cada objeto de la nube, se ha logrado calcular el porcentaje de los puntos que se han asignado correctamente. Esta información resulta muy útil para analizar mejor la efectividad del algoritmo en cada una de las maquetas.

- **Pruebas Finales:** En estas pruebas se ha aplicado el método propuesto a las nubes de puntos reales. Estas capturas contienen un mayor número de datos con respecto a las nubes artificiales, lo que aumentaba significativamente los cálculos computacionales y, por ello, el tiempo de ejecución del algoritmo. Para solucionar este problema, se ha realizado una reducción de la densidad de puntos de las nubes empleando la función “downsample” de Matlab para eliminar la mitad del conjunto de datos sin alterar la estructura real de la escena capturada. Un ejemplo de esta reducción de densidad se muestra en la Figura 5.3. Una vez realizado este proceso, ya se pudo aplicar el algoritmo siguiendo la misma metodología que en las pruebas iniciales, es decir, se fueron modificando los parámetros hasta llegar a un resultado aceptable. Los resultados de estas pruebas finales, que en este caso ya se les ha aplicado el tratamiento final, permitieron evaluar la aplicabilidad y efectividad del algoritmo en entornos no controlados.

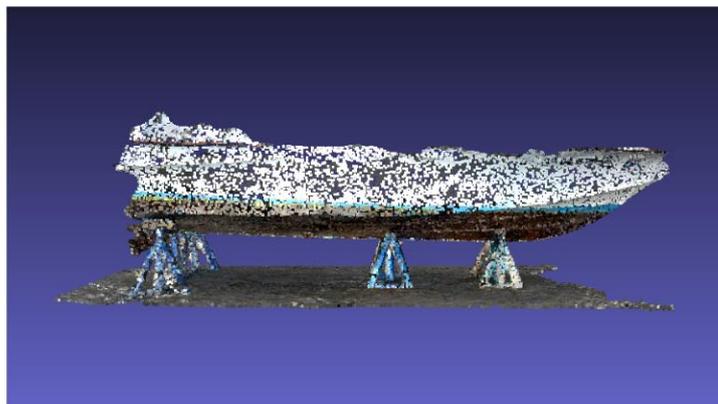
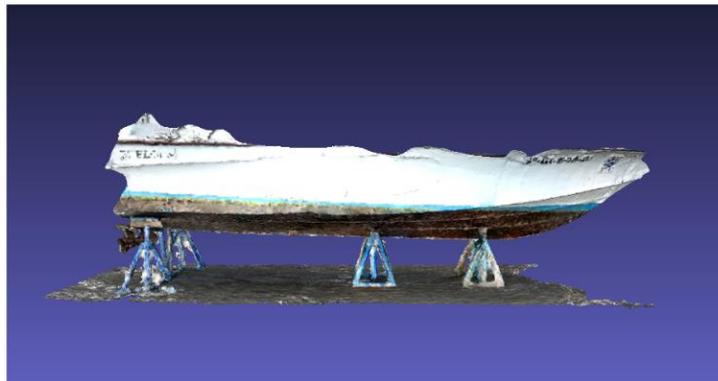


Figura 5.3.- Ejemplo de reducción de densidad de una nube de puntos

6. Resultados

En esta sección se expondrán los resultados obtenidos tras la aplicación del código a las distintas nubes de puntos mencionadas anteriormente.

6.1.- PRUEBAS INICIALES

Para simplificar la realización de las pruebas, algunos parámetros del algoritmo se han mantenido con un valor constante durante todo el proceso de experimentación. Los valores de dichos parámetros se muestran a continuación en la Tabla 6.1 .

Parámetros	Valores
KFM	Matriz 3x3 de Unos
FS	4
θ_D	0.1
θ_{PC}	400
R_V	1
ϵ_D	100

Tabla 6.1.- Parámetros fijos para las pruebas iniciales

A continuación, se presentarán los resultados óptimos obtenidos de cada una de las nubes de puntos artificiales, acompañados de los parámetros utilizados para alcanzar estos resultados. Se mostrarán dos imágenes representativas: la primera ilustrará la salida generada al aplicar k-means, mientras que la segunda exhibirá la segmentación final después de haber aplicado el proceso de merge. Finalmente, se indicará el porcentaje de puntos que han sido asignados correctamente al objeto al que pertenecieron originalmente.

6.1.1.- Maqueta 1

K	ω_C	ω_N	ω_D	σ_C	σ_N	δ_s
5	2	1	0.5	3	2.5	0.5

Tabla 6.2.- Parámetros para la Maqueta 1

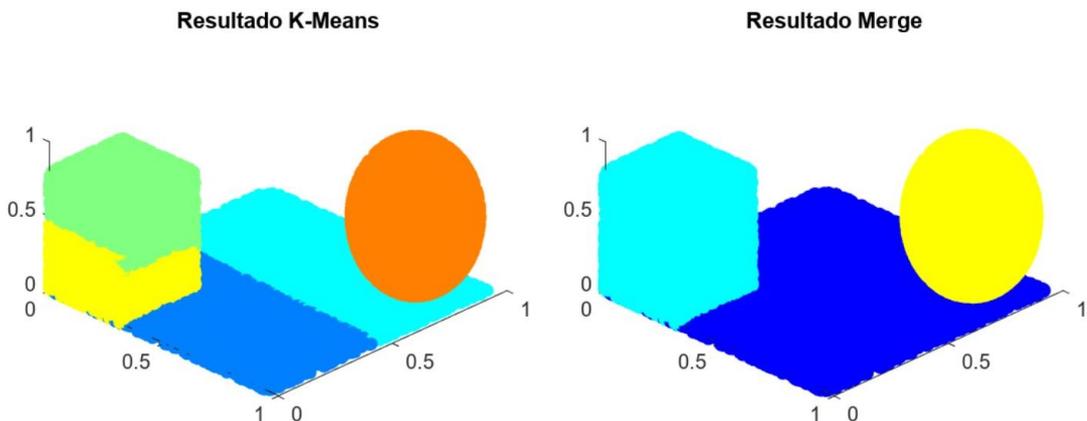


Figura 6.1.- Resultado de la Maqueta 1

Puntos de la nube	Puntos Bien Asignados	Puntos Mal Asignados	Porcentaje de Acierto
17000	14000	3000	82,35%

Tabla 6.3.- Resultados de la Maqueta 1

Para este primer experimento se ha optado por un valor k pequeño debido a la escasez de objetos en la escena. Como se puede apreciar, el algoritmo k-means ya llega a segmentar correctamente mucha parte de la nube de puntos y, tras la aplicación del merge se consigue una segmentación muy satisfactoria. El porcentaje de acierto es relativamente bajo debido a la mala asignación de algunos puntos de la base del cubo, los cuales considera parte del suelo.

6.1.2.- Maqueta 2

K	ω_C	ω_N	ω_D	σ_C	σ_N	δ_s
10	2	1.5	0.75	2	1.5	0.5

Tabla 6.4.- Parámetros para la Maqueta 2

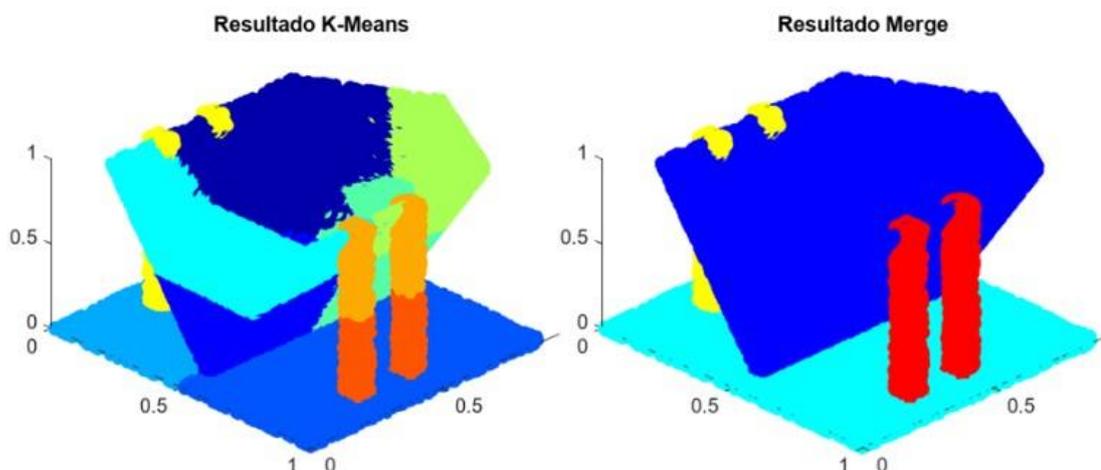


Figura 6.2.- Resultado de la Maqueta 2

Puntos de la nube	Puntos Bien Asignados	Puntos Mal Asignados	Porcentaje de Acierto
36332	36132	200	99,45%

Tabla 6.5.- Resultados de la Maqueta 2

En este caso, se ha decidido aumentar el valor de k ya que la nube consta de elementos más complejos. Además, se ha incrementado el peso del color debido a la diversidad de ángulos presentes en las superficies de la figura del barco. Con estos parámetros se consigue un muy buen resultado, llegando a obtener casi el 100% de porcentaje de acierto.

6.1.3.- Maqueta 3

K	ω_C	ω_N	ω_D	σ_C	σ_N	δ_s
10	3	1	0.2	3.25	1.5	0.3

Tabla 6.6.- Parámetros para la Maqueta 3

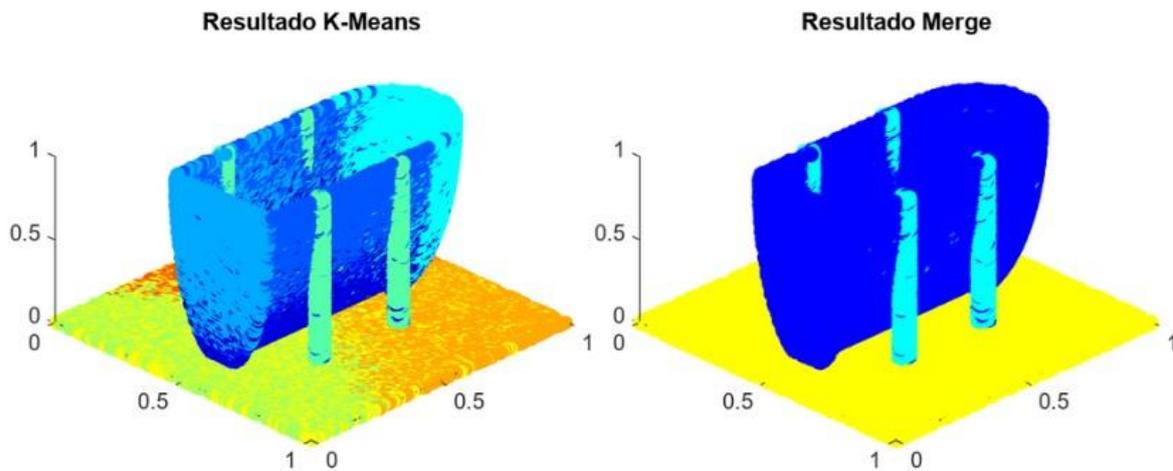


Figura 6.3.- Resultado de la Maqueta 3

Puntos de la nube	Puntos Bien Asignados	Puntos Mal Asignados	Porcentaje de Acierto
52681	48261	4420	91,61%

Tabla 6.7.- Resultados de la Maqueta 3

En esta salida de k-means, se observan clústeres entremezclados, una consecuencia directa del bajo peso asignado a la distancia. Esta situación provoca que algunos puntos pertenecientes al mismo cluster aparezcan a una distancia considerada. A pesar de esta condición, la segmentación resulta coherente ya que no se mezclan clústeres de distintos objetos. Finalmente, tras la aplicación del merge, se obtiene una segmentación bastante buena gracias a la adecuada fusión de los clústeres. Sin embargo, algunos puntos pertenecientes a los postes han sido asignados al casco del barco debido a su proximidad y similitud.

6.1.4.- Maqueta 4

K	ω_C	ω_N	ω_D	σ_C	σ_N	δ_s
10	3	1	0.2	3.25	1.5	0.3

Tabla 6.8.- Parámetros para la Maqueta 4

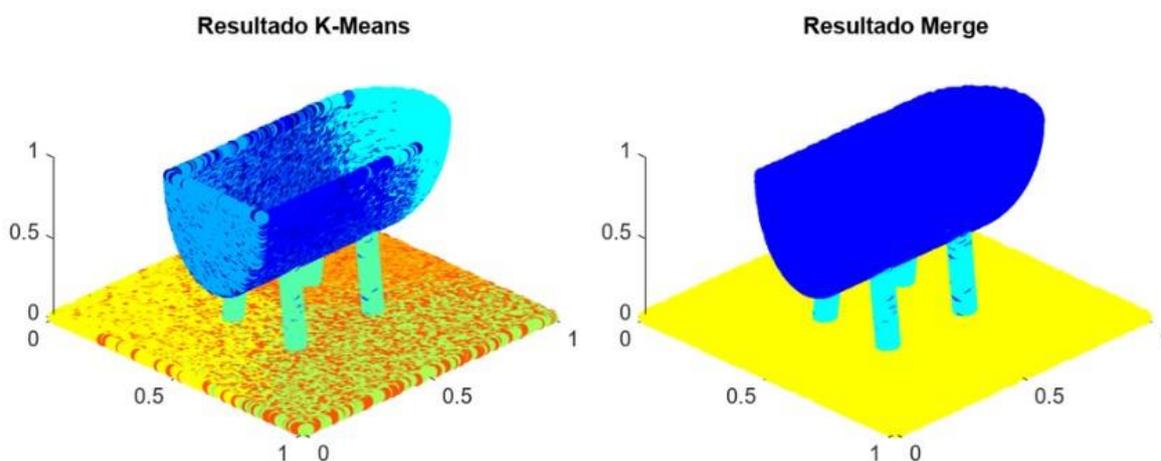


Figura 6.4.- Resultado de la Maqueta 4

Puntos de la nube	Puntos Bien Asignados	Puntos Mal Asignados	Porcentaje de Acierto
53516	53405	111	99,79%

Tabla 6.9.- Resultados de la Maqueta 4

La maqueta 4 ha sido segmentada utilizando los mismos parámetros que con el modelo anterior y ha resultado ser eficaz, lo cual es un aspecto muy positivo. Nuevamente, se obtiene un porcentaje muy alto de puntos bien asignados.



6.2.- PRUEBAS FINALES

En la realización de las pruebas finales también se han mantenido varios parámetros fijos mostrados a continuación en la Tabla 6.10

Parámetros	Valores
KFM	Matriz 3x3 de Unos
FS	4
θ_D	0.01
θ_{PC}	150
R_V	0.5
ϵ_D	100

Tabla 6.10.- Parámetros fijos para las pruebas finales

Algunos de estos parámetros se han modificado respecto a las pruebas iniciales para adaptar el código a la nueva naturaleza de las nubes de puntos. Estas nubes, a pesar de la reducción de densidad, presentan una alta densidad de puntos, lo que provoca una menor distancia entre ellos. Como consecuencia, se ha reducido el umbral de distancia “ θ_D ” para garantizar una segmentación más precisa. Además, se han modificado los parámetros del filtro de densidad “ R_V ” y “ ϵ_D ” con el objetivo de mejorar la limpieza del cluster final. Este ajuste es crucial ya que estas nubes contienen un mayor nivel de ruido y resulta más difícil recortar correctamente la unión de los objetos.

Los resultados de las pruebas finales se mostrarán a través de capturas realizadas en Meshlab debido a que las nubes reales son más complejas y así se podrán apreciar mejor los detalles. Se exhibirán dos imágenes: la primera corresponderá a la nube de puntos capturada mediante el escaneo láser, mientras que la segunda mostrará el resultado final del cluster que incluye el casco del barco. De nuevo se incluirán los parámetros utilizados para obtener cada uno de los resultados.

6.2.1.- Nube Real 1

K	ω_C	ω_N	ω_D	σ_C	σ_N	δ_s
10	1	0.25	2.5	0.5	0.75	0.05

Tabla 6.11.- Parámetros para la Nube Real 1



Figura 6.5.- Resultado de la Nube Real 1

En esta primera prueba con una nube real se han modificado significativamente algunos parámetros adaptando el código a este nuevo tipo de nubes. Como se puede observar, se ha realizado K-Means dándole un mayor peso al color y la distancia respecto a la normal. Por otro lado, se ha elegido un umbral de similitud mucho más restrictivo para fusionar únicamente los clústeres con mayor semejanza.

6.2.2.- Nube Real 2

K	ω_C	ω_N	ω_D	σ_C	σ_N	δ_s
10	1	0.25	2.5	0.5	1	0.05

Tabla 6.12.- Parámetros para la Nube Real 2

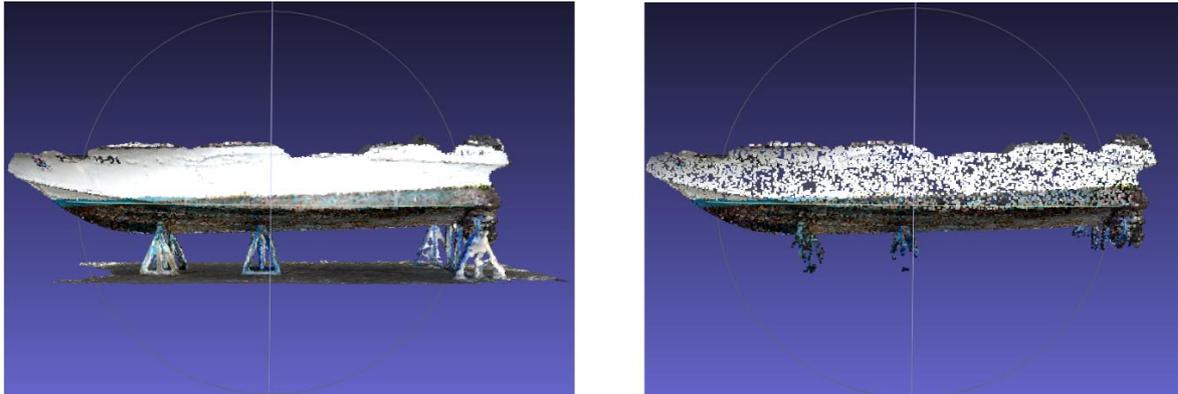


Figura 6.6.- Resultado de la Nube Real 2

6.2.3.- Nube Real 3

K	ω_C	ω_N	ω_D	σ_C	σ_N	δ_s
10	1	0.25	2.5	1.5	0.5	0.05

Tabla 6.13.- Parámetros para la Nube Real 3

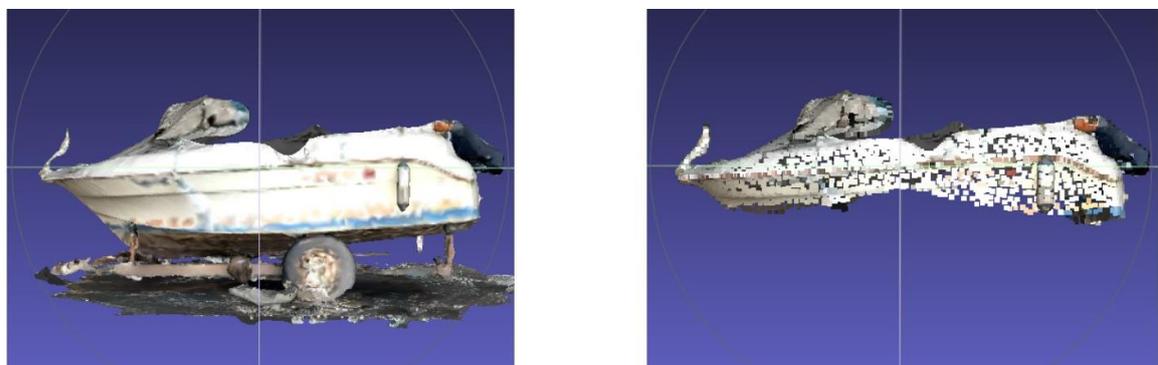


Figura 6.7.- Resultado de la Nube Real 3

6.2.4.- Nube Real 4

K	ω_C	ω_N	ω_D	σ_C	σ_N	δ_s
10	1	0.25	2.5	1.5	0.5	0.05

Tabla 6.14.- Parámetros para la Nube Real 4

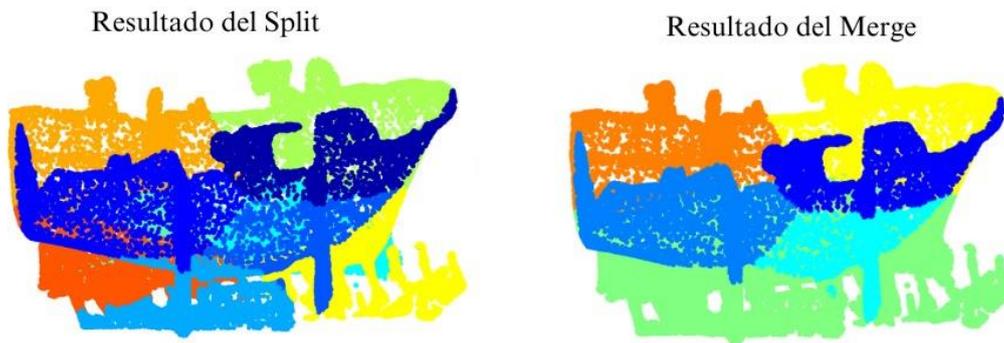


Figura 6.8.- Resultado de la Nube Real 4

En esta última nube de puntos real no se ha logrado obtener un solo cluster que contenga la mayor parte del barco, por lo que, para realizar un mejor análisis del resultado, se muestran únicamente las salidas del Split y del Merge. Como se puede apreciar en la Figura 6.8, los clústeres pertenecientes al barco no se consiguen fusionar, quedando un barco totalmente sobresegmentado.

7. Discusión

En este apartado de discusión se examinarán con mayor detalle los resultados mostrados en la sección anterior para evaluar la efectividad del método propuesto y su contribución al problema planteado.

En primer lugar, con las pruebas iniciales se ha demostrado que el método funciona de manera aceptable en la mayoría de los casos, obteniendo porcentajes de acierto muy elevados. En estos entornos sencillos y controlados, el algoritmo consigue una segmentación bastante precisa a pesar de tener algunas limitaciones.

La maqueta 1 es un ejemplo de estas limitaciones ya que el resultado no es del todo correcto, obteniendo el porcentaje más bajo de puntos bien asignados. Este fallo se produce en la etapa del Split con k-means, donde no se consigue segmentar correctamente la base del cubo. Como se puede observar en la Figura 7.1, algunos puntos de la base se mezclan con los del suelo ya que se encuentran contenidos en el mismo plano. Debido a su gran proximidad, el algoritmo no consigue identificar qué puntos pertenecen al suelo y cuales a la base del cubo.

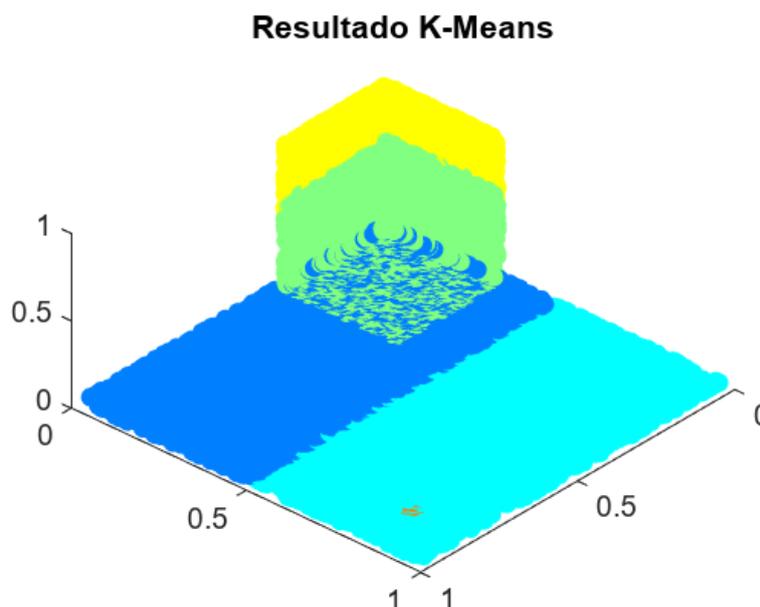


Figura 7.1.- Imagen de la vista inferior de la Maqueta 1



También es importante analizar los puntos que no han sido correctamente asignados en las maquetas 3 y 4. Estos puntos, pertenecientes a los postes, se han asignado al cluster del casco del barco, lo que introduciría ruido en la extracción de dicho objeto. Nuevamente, el fallo se produce en la etapa del split al aplicar K-Means a la nube. La dificultad de segmentar de forma coherente la escena planteada ha provocado la necesidad de disminuir el peso de la distancia al aplicar K-Means. Con esto se consigue una mejor segmentación por color y normal, pero provoca la generación de clústeres menos compactos y más entrelazados unos con otros. Esta mezcla de clústeres en el espacio ha sido la causante de la incorrecta asignación de los puntos mencionados anteriormente.

En general, se consigue una muy buena segmentación en las pruebas iniciales, aunque ha sido necesaria una modificación notable de los parámetros para llegar a dichos resultados. La necesidad de ajustar significativamente los parámetros representa un aspecto negativo, ya que sería más deseable si los mismos parámetros pudiesen generar resultados satisfactorios en cualquier escena, lo que implicaría un proceso mucho más automatizado.

En cuanto a las pruebas finales, estas han sido satisfactorias, aunque también presentan algunos fallos por la naturaleza de los datos y limitaciones del algoritmo. En el resultado de la primera nube real se puede observar cómo el casco del barco no se ha llegado a extraer por completo, quedando la quilla y el timón recortados prácticamente a la mitad. Este fallo reside nuevamente en la aplicación de K-Means y su dificultad para separar objetos colindantes con colores muy similares. Como se puede observar en la Figura 7.2, la quilla del barco en la nube inicial está dividida claramente en dos partes, la superior de color marrón y la inferior con un color más azulado. Por otro lado, el timón tiene un color más homogéneo, aunque existe una variación en la intensidad, siendo la sección inferior más oscura.

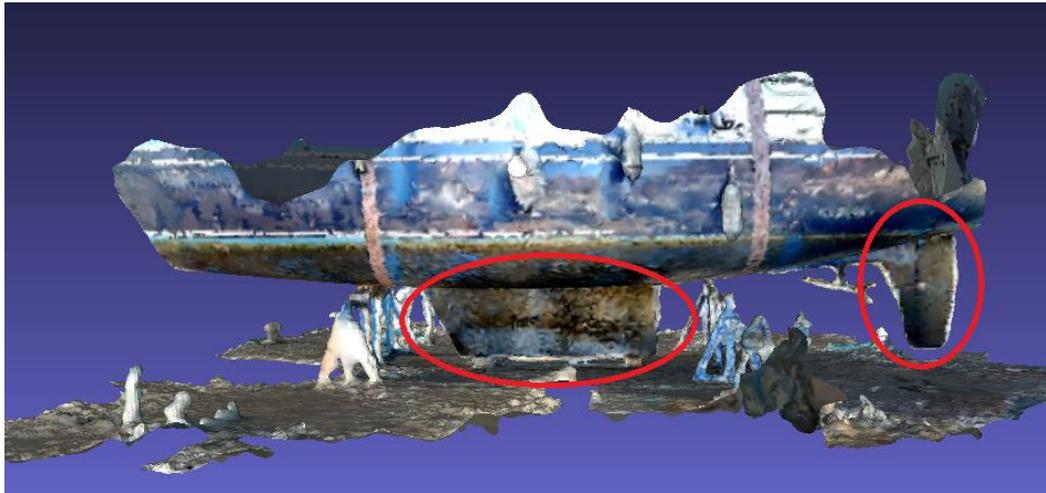


Figura 7.2.- Quilla y Timón del barco de la Nube Real 1

La utilización de un peso de color elevado con el fin de garantizar una segmentación coherente en el resto de la nube conduce a una sobresegmentación en estas partes del barco. Este fenómeno se produce debido a la presencia de valores de color significativamente distintos. Las partes inferiores de la quilla y del timón, como se puede apreciar en la Figura 7.3, se acaban asignando a clústeres del suelo.



Figura 7.3.- Resultados del Split y del Merge de la Nube Real 1

Con la nube real 2 vuelve a suceder el mismo problema que con la anterior, es decir, el proceso K-Means no llega a ser del todo efectivo y esto provoca que el resultado final contenga errores. En este caso, el casco del barco se mantiene completo en un solo cluster, se le agregan las porciones superiores de los soportes. Este problema se puede apreciar claramente en la Figura 7.4, donde se muestra el proceso de segmentación de la nube.

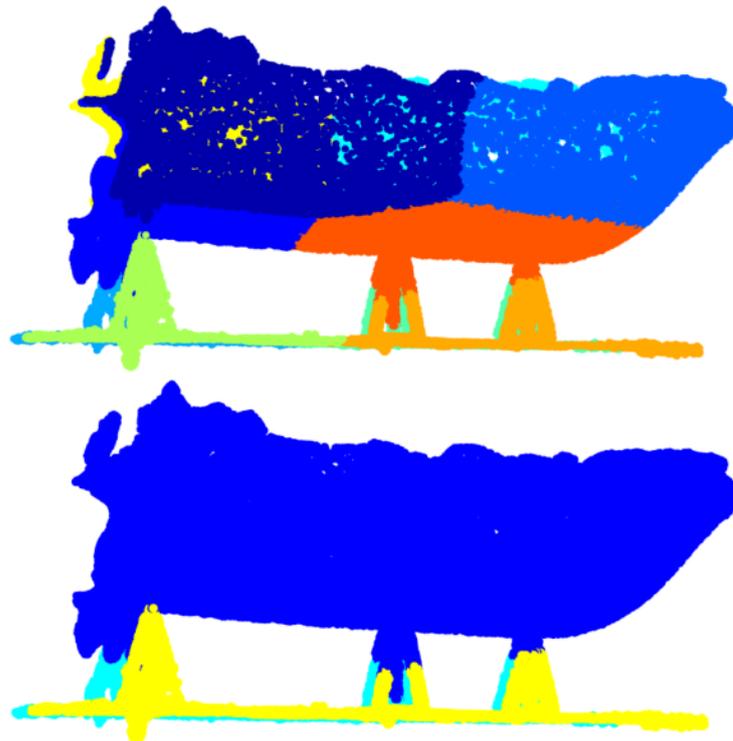


Figura 7.4.- Resultados del Split y del Merge de la Nube Real 2

La incrustación de parte de los soportes al casco del barco se debe principalmente a su proximidad y similitud de color. Aunque en la Figura 6.6 pueda parecer que el casco del barco es más oscuro debido al desgaste causado por la exposición al mar, su color intrínseco es azul, al igual que los soportes. Esto hace que algunas partes superiores de los soportes se acaben asignando al mismo cluster que las partes inferiores del barco.

Respecto a la nube real 3, el proceso no consigue extraer gran parte del casco del barco. En la Figura 6.7 nos encontramos con un cluster que contiene únicamente la parte superior del casco ya que el algoritmo no ha logrado realizar una correcta fusión de las secciones inferiores del barco con las superiores. Nuevamente, el error no se produce en la



fusión de los clústeres, si no en K-Means, el cual no consigue separar el soporte del barco de la parte inferior del casco. En este caso el color de estos dos elementos es muy similar y dificulta mucho su segmentación.

El resultado más desfavorable se obtiene en la segmentación de la nube real 4, donde el algoritmo no es capaz de obtener un único cluster que contenga los puntos pertenecientes al barco. En la Figura 6.8 se puede observar como el K-Means no es capaz de segmentar correctamente los puntos inferiores del barco y acaba asignándolos a clústeres pertenecientes a los raíles que soportan la embarcación. Este error provoca que al realizar el proceso del merge con pesos relativamente bajos, se fusionen algunos clústeres de los raíles con clústeres del barco. Al aumentar el valor de estos pesos ocurre lo contrario, es decir, no se consiguen fusionar clústeres que contienen puntos del mismo objeto, provocando la sobresegmentación del casco del barco mostrada en el resultado.

Por otro lado, se ha encontrado una limitación en la fusión iterativa de los clústeres durante la realización de las pruebas. Esta debilidad se encuentra en las iteraciones más avanzadas del proceso, cuando los clústeres se vuelven más grandes debido a su fusión con otros conjuntos. En estos casos, se produce una alta dispersión de los datos del color y la normal en los clústeres más extensos. Por ello, el cálculo de la media de estos datos resulta poco eficaz como indicador para determinar la similitud entre dos clústeres. Esto produce fusiones indeseadas entre objetos distintos o, por otro lado, la imposibilidad de fusionar clústeres que pertenecen al mismo objeto.

Para finalizar este análisis, es importante mencionar otro aspecto negativo del algoritmo propuesto. Uno de los fallos más significativos del código es la necesidad de realizar muchas pruebas antes de alcanzar un resultado óptimo con las nubes de puntos reales. Aunque el resultado del algoritmo K-Means se mantiene constante a lo largo de las pruebas debido a la fijación de centroides iniciales, el orden de los clústeres varía en cada ejecución. Esta variación en la disposición de los cluster influye directamente en el proceso de fusión (merge), ya que este analiza la similitud entre los clústeres de manera ordenada, es decir, comenzando por el cluster con índice uno. La alteración del orden de los clústeres afecta al resultado final de manera significativa en cada prueba realizada. Por lo tanto, hay



que realizar varias pruebas y ajustar los parámetros del algoritmo hasta alcanzar un resultado aceptable. Este comportamiento resulta una tarea laboriosa y disminuye el grado de automatización del proceso.



8. Conclusiones

A pesar de la dificultad que supone realizar una segmentación precisa de datos tridimensionales, se ha logrado cumplir el objetivo marcado al comienzo del proyecto. Tras el análisis de la efectividad del método propuesto, se puede concluir que resulta una solución factible para la segmentación de las nubes de puntos planteadas. Además, el estudio inicial y el estado del arte proporcionan una visión general de las diferentes técnicas de segmentación de nubes de puntos que existen actualmente.

Las pruebas realizadas han servido para demostrar que el algoritmo funciona de manera aceptable tanto en entornos controlados como en escenarios reales. Sin embargo, se señalaron una serie de limitaciones, las cuales sirven para enfatizar la necesidad de mejoras en los procesos de K-Means y Merge Iterativo para lograr una segmentación más automatizada y precisa.

A continuación, se citan algunas mejoras que se podrán realizar en el futuro para aumentar la automatización y efectividad del algoritmo.

En primer lugar, para solucionar el problema de la reducción de la densidad de la nube, se podría aplicar alguna técnica de reconstrucción a la salida del algoritmo. Corregir este problema facilitaría la posterior identificación de incrustaciones en la obra viva del barco.

Otra mejora podría ser la modificación del algoritmo K-Means para que consiga segmentar de manera más precisa los datos, facilitando así el proceso de fusión. También sería interesante poder lograr un proceso más automatizado realizando segmentaciones coherentes con los mismos pesos para todas las nubes de puntos.

Por último, sería necesaria una modificación del algoritmo de Merge Iterativo para que consiga que la salida del proceso se mantenga constante durante las pruebas con parámetros fijos. Este ajuste garantizaría una estabilidad consistente en los resultados



obtenidos, ofreciendo una base más sólida para la evaluación y comparación de diferentes configuraciones. Además, se propone la implementación de nuevas características, como la curvatura, para mejorar la precisión del algoritmo y solucionar los fallos en la fusión de clústeres en iteraciones avanzadas.

En el ámbito personal, cabe destacar el aprendizaje adquirido durante la realización del proyecto. En primer lugar, conocer mejor el ámbito de las representaciones tridimensionales y sus diferentes técnicas de segmentación. Además, el desarrollo del código no solo cumplió con los objetivos planteados, sino que también amplió mis conocimientos y habilidades en el manejo de la “Computer Vision Toolbox” de Matlab. Este aspecto específico del proyecto ha fortalecido mi competencia en el uso de herramientas especializadas para el procesamiento de datos tridimensionales.



9. Bibliografía

- [1] «CloudCompareWiki». Accedido: 23 de enero de 2024. [En línea]. Disponible en: https://www.cloudcompare.org/doc/wiki/index.php/Main_Page
- [2] «Nube de puntos | Escaner laser 3D | España | Escaneado láser | Ingeniería inversa | Scan to BIM», scanphase. Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: <https://www.scanphase.com/nubes-de-puntos>
- [3] «¿Qué es el escaneo 3D láser? | Soluciones profesionales de escaneo 3D | Artec3D». Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: <https://www.artec3d.com/es/learning-center/laser-3d-scanning>
- [4] «¿Cómo funciona el LiDAR?», YellowScan. Accedido: 21 de noviembre de 2023. [En línea]. Disponible en: <https://www.yellowscan.com/es/knowledge/how-does-lidar-work/>
- [5] Lynn, «LiDAR vs Photogrammetry: Which is better for point cloud creation? -», Mosaic51.com. Accedido: 21 de noviembre de 2023. [En línea]. Disponible en: <https://www.mosaic51.com/technology/lidar-vs-photogrammetry-which-is-better-for-point-cloud-creation/>
- [6] N. M. H. Favieri, «Reconocimiento de Objetos en Imágenes RGB-D», mar. 2015.
- [7] P. J. Besl y R. C. Jain, «Segmentation through variable-order surface fitting», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, n.º 2, pp. 167-192, mar. 1988, doi: 10.1109/34.3881.
- [8] P. V. C. Hough, «Method and means for recognizing complex patterns», US3069654A, 18 de diciembre de 1962 Accedido: 6 de diciembre de 2023. [En línea]. Disponible en: <https://patents.google.com/patent/US3069654A/en>
- [9] M. A. Fischler y R. C. Bolles, «Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography», *Commun. ACM*, vol. 24, n.º 6, pp. 381-395, jun. 1981, doi: 10.1145/358669.358692.
- [10] «Random sample consensus (RANSAC)». Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: https://rstudio-pubs-static.s3.amazonaws.com/752943_8e2a5a8a45484f7a9b50b596be47f217.html
- [11] J. MacQueen, «Some methods for classification and analysis of multivariate observations», en *Proceedings of the Fifth Berkeley Symposium on Mathematical*



- Statistics and Probability, Volume 1: Statistics*, vol. 5.1, University of California Press, 1967, pp. 281-298. Accedido: 6 de diciembre de 2023. [En línea]. Disponible en: <https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992>
- [12] M. Ester, H.-P. Kriegel, y X. Xu, «A Density-Based Algorithm for Discovering Clústeres in Large Spatial Databases with Noise».
- [13] Y. Cheng, «Mean shift, mode seeking, and clustering», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, n.º 8, pp. 790-799, ago. 1995, doi: 10.1109/34.400568.
- [14] «¿Qué son las redes neuronales? | IBM». Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/neural-networks>
- [15] A.-V. Vo, L. Truong-Hong, D. Laefer, y M. Bertolotto, «Octree-based region growing for point cloud segmentation», *ISPRS J. Photogramm. Remote Sens.*, vol. 104, jun. 2015, doi: 10.1016/j.isprsjprs.2015.01.011.
- [16] Y.-T. Su, J. Bethel, y S. Hu, «Octree-based segmentation for terrestrial LiDAR point cloud data in industrial applications», *ISPRS J. Photogramm. Remote Sens.*, vol. 113, pp. 59-74, mar. 2016, doi: 10.1016/j.isprsjprs.2016.01.001.
- [17] «Octree data structure», OpenGenus IQ: Computing Expertise & Legacy. Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: <https://iq.opengenus.org/octree/>
- [18] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, y L. Tang, «An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells», *Remote Sens.*, vol. 9, n.º 5, Art. n.º 5, may 2017, doi: 10.3390/rs9050433.
- [19] C. Hui, W. Tingting, D. Zuoxiao, L. Weibin, y M. I. Menhas, «Power Equipment Segmentation of 3D Point Clouds Based on Geodesic Distance with K-means Clustering», en *2021 6th International Conference on Power and Renewable Energy (ICPRE)*, Shanghai, China: IEEE, sep. 2021, pp. 317-321. doi: 10.1109/ICPRE52634.2021.9635211.
- [20] C. Shi, B. Wei, S. Wei, W. Wang, H. Liu, y J. Liu, «A quantitative discriminant method of elbow point for the optimal number of clústeres in clustering algorithm», *EURASIP J. Wirel. Commun. Netw.*, vol. 2021, n.º 1, p. 31, feb. 2021, doi: 10.1186/s13638-021-01910-w.



- [21] D. R. Pajarola, «Advanced 3D Computer Graphics», oct. 2000.
- [22] X. Zhang, G. Li, Y. Xiong, y F. He, *3D Mesh Segmentation Using Mean-Shifted Curvature*. 2008, p. 474. doi: 10.1007/978-3-540-79246-8_35.
- [23] C. Wang, M. Ji, J. Wang, W. Wen, T. Li, y Y. Sun, «An Improved DBSCAN Method for LiDAR Data Segmentation with Automatic Eps Estimation», *Sensors*, vol. 19, n.º 1, p. 172, ene. 2019, doi: 10.3390/s19010172.
- [24] «¿Qué es el algoritmo de k vecinos más cercanos? | IBM». Accedido: 23 de enero de 2024. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/knn>
- [25] «Computer Vision Toolbox Documentation - MathWorks España». Accedido: 30 de diciembre de 2023. [En línea]. Disponible en: https://es.mathworks.com/help/vision/index.html?s_tid=CRUX_lftnav
- [26] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, y G. Ranzuglia, «MeshLab: an Open-Source Mesh Processing Tool», *Eurographics Ital. Chapter Conf.*, p. 8 pages, 2008, doi: 10.2312/LOCALCHAPTEREVENTS/ITALCHAP/ITALIANCHAPCONF2008/129-136.
- [27] «Polycam - LiDAR & 3D Scanner for iPhone & Android». Accedido: 13 de enero de 2024. [En línea]. Disponible en: <https://poly.cam/>
- [28] K. Iqbal, «PLY - Formato de archivo 3D de polígono». Accedido: 13 de enero de 2024. [En línea]. Disponible en: <https://docs.fileformat.com/es/3d/ply/>