



DECLARACIÓN DE
AUTORÍA Y ORIGINALIDAD

(Este documento debe remitirse a seceina@unizar.es dentro del plazo de depósito)

D./D^a. Daniel Capuz Garcia

en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de Estudios de la titulación de Grado en Ingeniería Electrónica y Automática (Título del Trabajo)

Modelo digital de galerías subterráneas inundadas mediante un barco robotizado usando SLAM/ICP

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 5 de Junio de 2023

Fdo: Daniel Capuz Garcia



Universidad
Zaragoza

Trabajo Fin de Grado

Modelo digital de galerías subterráneas inundadas
mediante un barco robotizado usando SLAM/ICP

Digital model of flooded underground galleries by
robotic boat using SLAM/ICP

Autor

Daniel Capuz Garcia

Director

José Luis Villarroel Salcedo

Grado en ingeniería Electrónica y Automática

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2023

Índice

1.- Introducción	- 1 -
2.- Algoritmo ICP.....	- 2 -
2.1.- Cálculo punto a punto.....	- 3 -
2.2.- Cálculo punto a plano	- 4 -
3.- RoboBoat	- 7 -
4.- SLAM	- 10 -
5.- Cálculo Predicciones.....	- 11 -
5.1.- Predicción sin IMU.....	- 11 -
5.2.- Predicción con IMU.....	- 12 -
6.- Métricas.....	- 14 -
7.- Reconstrucciones parciales	- 19 -
8.- Reconstrucciones completas	- 25 -
9.- Conclusiones	- 31 -
10.- Bibliografía	- 34 -
Anexo 1	- 35 -

1.- Introducción

Este Trabajo Fin de Grado ha sido realizado en el seno del Grupo de Robótica de la Universidad de Zaragoza y dentro del marco de los proyectos Autonomous or semi-autonomous Robot Deployment for Underground Applications (ARDUA, PID2019-105390RB-I00, Ministerio de Ciencia e Innovación, Proyectos de I+D+i Retos Investigación) y “Estudio e Intervención en Cueva Román (CLUNIA, Burgos)” financiado por la Diputación Provincial de Burgos.

El objetivo de este trabajo es desarrollar un algoritmo SLAM basado en ICP adaptado a entornos no estructurados confinados y a una aplicación de obtención de modelos 3D de este tipo de entornos. Proporciona autolocalización y cartografía 3D a partir de nubes de puntos capturadas por sensores LiDAR, con precisión suficiente para la reconstrucción de cuevas.

El trabajo y el algoritmo están implementados en Matlab con la utilización de los datos obtenidos por los sensores que van equipados en el RoboBoat. Para la realización del trabajo se emplearon las toolboxes de Robotic System, ROS y Computer Vision, para tratar los datos del robot, así como la implementación del algoritmo.

El trabajo está estructurado de la siguiente forma: en primer lugar, se explicará el funcionamiento del algoritmo ICP. Posteriormente se comentarán los distintos sensores que tiene el RoboBoat. A continuación, se comentará brevemente en que consiste el SLAM. Después se explicarán como se calculan las predicciones de las posiciones del RoboBoat. Posteriormente ya entraremos en la explicación de las métricas empleadas para obtener la mejor parametrización del algoritmo. A continuación, se expondrán los resultados obtenidos en cuanto a la parametrización del algoritmo. Y por último, se agruparán todas las conclusiones obtenidas durante la realización del trabajo.

2.- Algoritmo ICP

El algoritmo ICP (*Iterative Closest Point*) es un algoritmo utilizado en la reconstrucción 3D para alinear nubes de puntos en computación gráfica y visión por computador [1][2]. Su objetivo es encontrar la mejor transformación (rotación y translación) que puede ser aplicada a una nube de puntos, para que se ajuste lo mejor posible a otra nube de puntos en tres dimensiones. Es un algoritmo iterativo que converge de forma monótona al mínimo local de la distancia métrica entre nubes de puntos en los seis grados de libertad.

Desde la década de los 90 el algoritmo ICP ha sido utilizado como el mejor algoritmo para la obtención de modelos geométricos de escáneres 3D. Este algoritmo nos permite relacionar dos nubes de puntos de un mismo objeto o de un escenario. Busca minimizar una métrica de error entre las parejas de puntos que se crean en dichas nubes.

El algoritmo se ejecuta por iteraciones y en cada iteración el algoritmo intenta ajustar dos nubes de puntos que denominaremos la nube “móvil” es a la que se le aplica la transformación para ajustarla a la otra nube. La otra nube de puntos que se mantiene fija la denominaremos nube “fija”. Estos adjetivos también se aplicarán a los puntos pertenecientes a dichas nubes, hablando así de puntos “móviles” y puntos “fijos”.

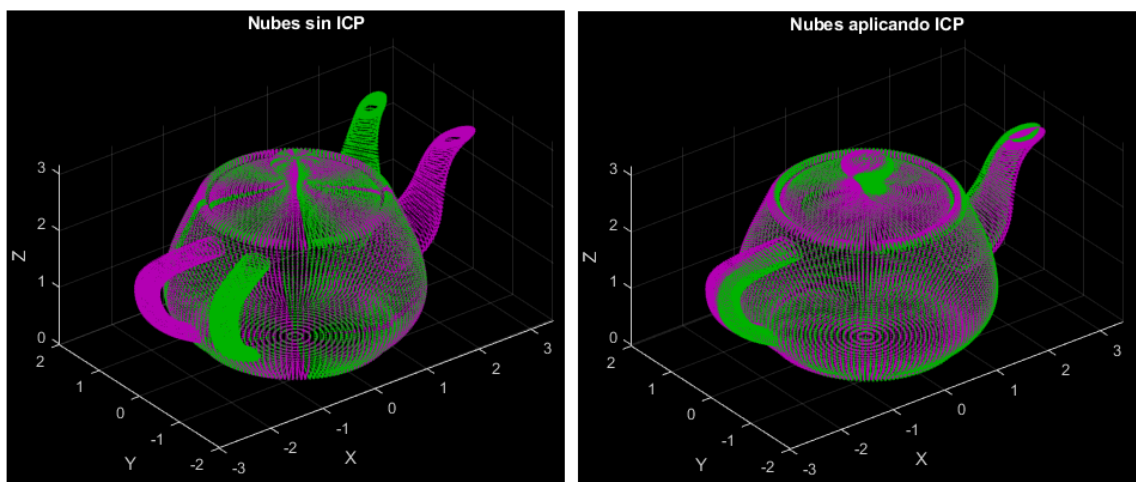


Figura 1. En la parte de la izquierda se observan las dos nubes de puntos que se van a emparejar mediante el ICP y a la derecha el resultado tras aplicar el algoritmo.

En la figura 1 podemos observar un ejemplo de funcionamiento del ICP. La nube de puntos verdes sería la nube de puntos móvil a la cual tras el uso del algoritmo se le aplica una rotación para que coincida con la nube de puntos lila que sería la nube de puntos fija.

El algoritmo busca emparejar todos los puntos de la nube móvil con su punto correspondiente en la nube fija. Si el emparejamiento es correcto y por lo tanto se está aplicando de forma adecuada el algoritmo, los dos puntos son correspondientes al mismo punto en el objeto o escenario real escaneado. Una vez hemos obtenido los puntos que son correspondientes se busca emparejarlos de la forma más precisa, de forma que todos los puntos finalmente sean coincidentes con su pareja aplicando una transformación a la nube móvil.

La transformación que aplicamos a la nube móvil se representa como una matriz 4x4 llamada matriz de transformación donde se incluye la rotación y traslación que hay que aplicar a la nube móvil para ajustarla a la nube fija. Esta matriz de transformación se recalcula en cada iteración del algoritmo y busca minimizar una función de costo.

Esta función de costo puede ser de dos formas principalmente. Puede ser el sumatorio de distancias de todos los puntos de la nube móvil a su punto correspondiente en la nube fija, esto da lugar al método conocido como cálculo de punto a punto. Otra función de costo puede ser el sumatorio de las distancias mínimas de todos los puntos de la nube móvil a los planos tangentes a la superficie más cercana a ese punto en la nube fija, método conocido como cálculo punto a plano.[3]

2.1.- Cálculo punto a punto

Se selecciona un punto en la nube móvil y se busca el punto más cercano en la nube fija, mediante el cálculo de la distancia euclídea entre ese punto y todos los puntos de la otra nube. Una vez tenemos el punto más cercano se obtiene el vector de error entre el punto móvil y el punto fijo (el vector que los une) y se guarda. Al calcular todas las distancias mínimas de todos los puntos de la nube móvil a todos los puntos de la nube fija obtenemos todos los vectores de error. Calculando la media de los vectores de error obtenemos la translación que le realizamos a la nube de puntos móviles para ajustarla a la nube de puntos fija. Cuando ya hemos realizado la translación de la nube móvil repetimos el proceso hasta alcanzar la convergencia de las dos nubes de puntos. En la figura 2 podemos observar el esquema de funcionamiento explicado.

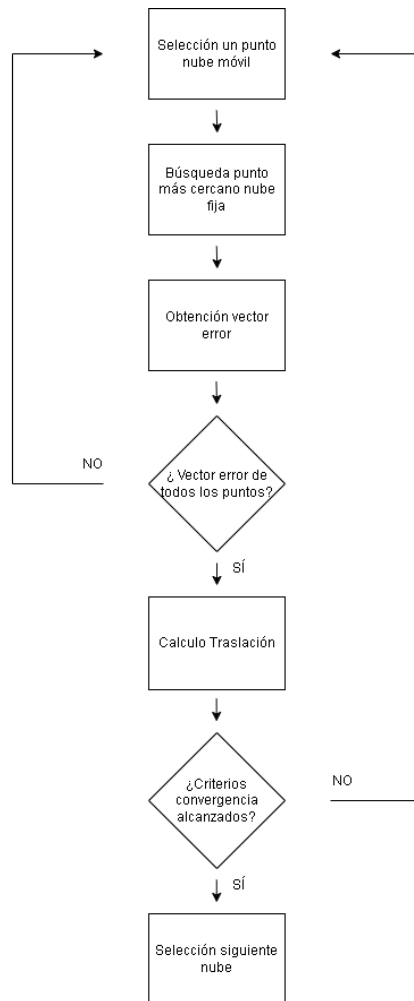


Figura 2. Esquema funcionamiento método punto a punto

2.2.- Cálculo punto a plano

La función de costo es la distancia de cada punto de la nube móvil y el plano tangente a la superficie de la nube fija más cercana a cada punto móvil. La distancia entre el punto móvil y el plano tangente se calcula como la distancia perpendicular al plano. Con el plano tangente a cada punto móvil se puede minimizar la suma de todas las distancias para obtener la transformación rígida óptima. Para calcular el plano tangente a la superficie de la nube fija más cercana a cada punto, primero tenemos que calcular el punto más próximo de la nube fija a nuestro punto móvil como se realiza en el método de punto a punto mediante la distancia euclídea. Una vez obtenemos este punto fijo en lugar de calcular el vector de error como en el método de punto a punto, seleccionamos unos cuantos puntos vecinos a este y calculamos el plano que estos forman. Con el plano obtenido calculamos la distancia del punto móvil al plano formado y obtenemos la matriz de transformación intentando minimizar esas distancias. En la figura 3 podemos observar el esquema de funcionamiento explicado.

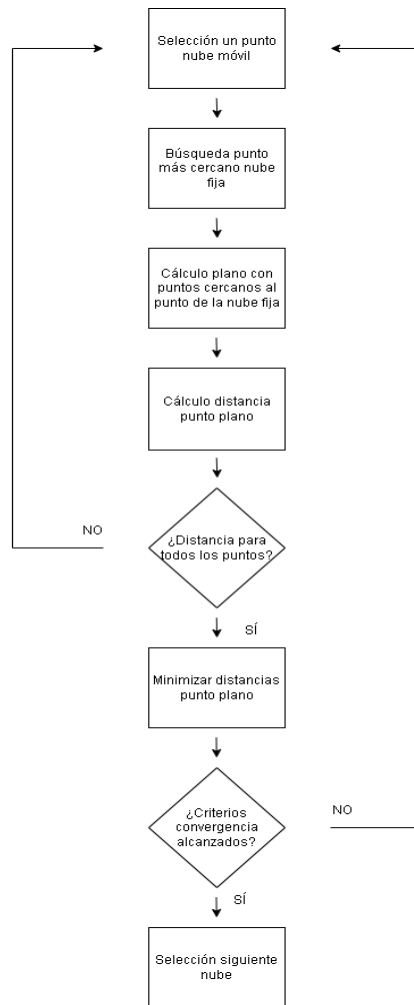


Figura 3. Esquema funcionamiento método punto a plano

El método de punto a punto es más rápido y sencillo, pero puede ser sensible a ruido en los datos y *outliers* (datos anómalos o que no tienen correspondencia en la otra nube). Mientras que el método de punto a plano es computacionalmente más costoso pero debido a que utiliza más información de la nube de puntos es más robusto frente a ruido y *outliers*. [3]

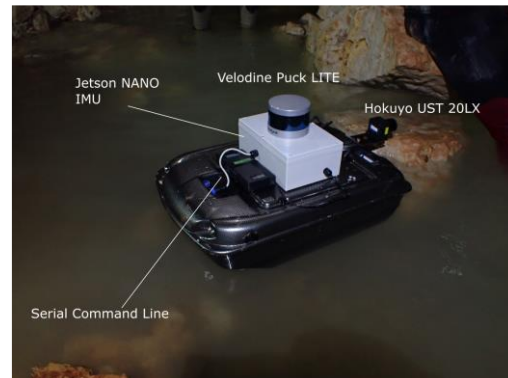
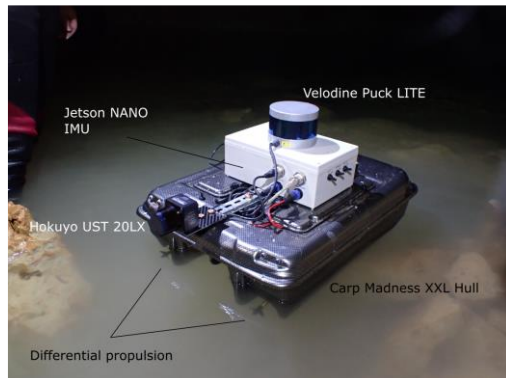
Por lo tanto, en general el método de punto a plano suele arrojar unos resultados más precisos y robustos, pero en situaciones donde se dispone de datos limpios y es necesaria una alineación rápida el método punto a punto puede ser de mayor utilidad.

A parte del método que se emplea para calcular las distancias, existen otros parámetros relacionados con el ICP:

- El porcentaje de puntos que se consideran *outliers*: es decir que es ruido de los sensores o puntos que no tienen una correspondencia. Si los datos son limpios y todos los puntos tienen su correspondencia este porcentaje puede ser menor.
- La tolerancia de convergencia: es el error que estamos dispuestos a asumir para considerar que el emparejamiento ha sido adecuado. Una tolerancia mayor puede provocar que las dos nubes no se unan de una forma adecuada, pero una tolerancia muy pequeña puede resultar imposible de alcanzar en algunos casos y en todos los casos aumentará el tiempo requerido para alcanzarla si es posible.
- El número máximo de iteraciones: es el límite de iteraciones que le permitimos realizar al algoritmo en caso de que no consiga alcanzar la tolerancia que hemos considerado. Al igual que la tolerancia de convergencia un número máximo de interacciones muy reducido puede provocar que la unión de las dos nubes no sea la adecuada, pero un número muy elevado provocará en el caso de que no se alcance la tolerancia que el coste computacional aumente considerablemente.

3.- RoboBoat

El RoboBoat con el que se han recogido los datos es un barco que se basa en el barco comercial Carp Madness XXL, con propulsión diferencial proporcionada por dos hélices impulsadas por dos motores con reguladores de velocidad, pero sin realimentación de velocidad. El RoboBoat cuenta con 3 sensores instalados: IMU, Velodyne y Hokuyo.



Figuras 4 y 5. Imagen del RoboBoat y sus sensores.

El Velodyne LiDAR (*Ligh Detection and Ranging*) es un sensor comúnmente utilizado en robótica y vehículos autónomos para la detección del entorno en un espacio tridimensional. En este caso el sensor cuenta con 16 planos con un rango de apertura de $\pm 15^\circ$, que nos permiten obtener un campo de visión de 360 grados del escaneo y tiene un rango máximo de 100 metros. Este sensor nos permitirá obtener las nubes de puntos que utilizaremos para obtener el mapeo de la cueva.

El IMU Phidgets Spatial 3/3/3 es un sensor de medición inercial que nos permite obtener la aceleración lineal, velocidad angular y la orientación del RoboBoat mediante un acelerómetro, un giróscopo y magnetómetro. Los datos obtenidos por este sensor los utilizaremos para realizar una predicción más precisa de la ubicación del RoboBoat y que la posición inicial de las dos nubes de puntos obtenidas del Velodyne estén mejor ajustadas y que de esta forma el algoritmo de ICP funcione mejor.

Los datos obtenidos de ambos sensores serán tratados antes de utilizarlos en el algoritmo del ICP para intentar obtener un mejor resultado.

Por un lado, a los datos obtenidos de aceleración lineal tendremos que restarle la gravedad para ello deberemos tener en cuenta la orientación del barco y descomponer la gravedad para restarle a cada eje la contribución que tiene.

En cuanto a los datos del sensor Velodyne, podremos realizar varias modificaciones para trabajar con una nube de puntos más pequeña y que cuente únicamente con los puntos que más nos puedan interesar, para ello ajustaremos los siguientes parámetros:

- Distancia máxima: Esta es la distancia máxima que se puede encontrar un punto del RoboBoat para que lo utilicemos. El sensor cuenta con un rango de 100m, pero no es necesario tener en cuenta puntos muy alejados al RoboBoat. Esto es debido a que pueden llevar a errores y teniendo en cuenta que el RoboBoat va a recorrer toda la orilla de la cueva esos puntos lejanos en algún momento estarán próximos al robot. Por otro lado, una distancia muy elevada provoca que todas las nubes tengan muchos puntos aumentando de esta manera el tiempo de cómputo de cada iteración.
- Distancia mínima: Debido a que el RoboBoat siempre navegará a una distancia segura de la orilla los puntos que se encuentren muy cerca del RoboBoat probablemente sean ruido que ha tenido el sensor. Por lo tanto, tratar de eliminarlos antes de realizar el emparejamiento evitará que estos puntos molesten a la hora de emparejar las dos nubes de puntos, debido a que probablemente el ruido de una medición a la siguiente no tenga correspondencia.
- Altura máxima: Esta es la altura máxima a la que se puede encontrar un punto para que lo utilicemos. Puntos muy elevados nos pueden ayudar a realizar el ajuste con mayor precisión, pero aumentan significativamente la cantidad de puntos en la nube. Lo que repercute en un tiempo de cómputo mayor a la hora de realizar el emparejamiento de las nubes.
- Altura mínima: Esta es la altura mínima a la que se puede encontrar un punto para que lo utilicemos. Debido a que el RoboBoat está navegando por agua calmada los puntos que se encuentren por debajo del RoboBoat son puntos sumergidos en el agua o son puntos erróneos debido a que el agua ha provocado un rebote del haz láser del sensor. Por lo tanto, es conveniente eliminar estos puntos.
- Método de Reducción del Muestreo (*DownSamplingMethod*): Este parámetro nos permite reducir la cantidad de puntos con la que trabajamos reduciendo el tiempo de cómputo, este método se puede aplicar de 3 formas:
 1. Promedio de cuadrícula (*gridAverage*) este método divide la nube de puntos en una rejilla y agrupa todos los puntos que se encuentran dentro de una misma cuadrícula, calculando la media de sus posiciones y normales. Para ello hay que calibrar otro parámetro el tamaño de la rejilla. A mayor tamaño el número de puntos se verá reducido de manera más significativa, pero los datos obtenidos después de la aplicación del filtro pueden no resultar de utilidad debido a que se han visto muy reducidos. Por otro lado, una rejilla muy pequeña provoca que los puntos no se vean reducidos.

2. Promedio de cuadrícula no uniforme (*nonuniformGridSample*) este método al igual que el *gridAverage* divide la nube de puntos en cuadrículas y agrupa los puntos de dentro de cada cuadrícula. Pero a diferencia del método *gridaverage* este método da una mayor importancia a la normal de los puntos dentro de cada cuadrícula a la hora de agruparlos.
 3. Aleatorio (*random*) este método reduce el número de puntos de forma aleatoria.
- Intensidad del rebote (*Min_Intensity*) este parámetro nos permite filtrar los puntos por la intensidad que llega de vuelta el haz laser, lo que nos permite eliminar puntos que provengan de un rebote del agua. El rango de este parámetro es de 0 a 100.

4.- SLAM

En el trabajo se propone el uso de un método ICP SLAM (*Simultaneous Location And Mapping*) para mapear el entorno de una cueva. Como ya se ha mencionado el ICP es un método comúnmente utilizado en robótica para alinear dos conjuntos de puntos 3D y determinar la posición y orientación relativa entre dos momentos consecutivos. Por otro lado, el SLAM es una técnica utilizada en robótica y sistemas de navegación para construir un mapa del entorno y realizar una estimación de la ubicación del robot en el entorno. Para ello se utilizarán los sensores Velodyne para obtener las nubes de puntos y el sensor IMU para obtener las aceleraciones lineales y las velocidades y posiciones angulares del barco, para poder calcular una predicción de la posición del barco como se explica en el apartado siguiente.

Para la reconstrucción del mapa en primer lugar se seleccionarán las nubes de puntos obtenidas del sensor Velodyne. Estas se filtrarán y tratarán para intentar mejorar los resultados. Posteriormente se calculará una predicción de la posición del barco como se explicará en el apartado siguiente, con la intención de mejorar las posiciones relativas de las dos nubes consecutivas. Después se filtrarán y tratarán las nubes de puntos. Posteriormente se aplicará el algoritmo ICP para emparejar las nubes de puntos. Tras el emparejamiento se realizarán los cálculos relativos a las métricas que utilizamos para analizar los resultados obtenidos. Este proceso se realizará para todas las nubes de puntos y poder reconstruir el mapa y la ubicación del RoboBoat en todo momento. En la figura 6 podemos observar el funcionamiento del SLAM.

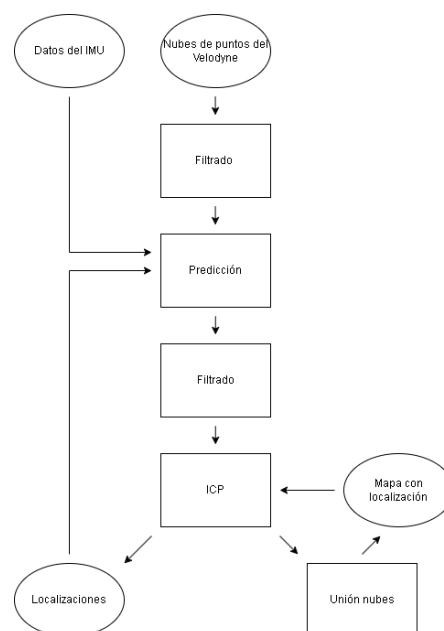


Figura 6. Esquema funcionamiento del algoritmo de SLAM.

5.- Cálculo Predicciones

En esta sección se van a describir como se calculan las dos predicciones implementadas para la mejora de la posición inicial de las dos nubes, la que no utiliza el sensor IMU y la que utiliza el sensor.

5.1.- Predicción sin IMU

Esta predicción consistirá en suponer que el barco sigue con la misma velocidad angular y lineal de una nube de puntos a la siguiente. Por lo tanto, sabiendo la frecuencia de muestreo del sensor Velodyne (10 Hz) podemos predecir la posición y orientación que tendrá el barco en el siguiente muestreo. Con esta predicción se espera obtener una mejora con respecto a los resultados obtenidos sin utilizar ninguna predicción, debido a que ahora no suponemos que las dos nubes de puntos se encuentran en la misma posición. Si la predicción es buena las primeras iteraciones ya no tendrán valores de error tan elevados debido a que las nubes ya estarán en una posición y orientación óptima para su emparejamiento. Por otro lado, esta predicción puede provocar que en algunos momentos obtengamos picos de error debido a que la velocidad lineal o angular del barco varía mucho con respecto a la que tenía previamente (un ejemplo podría ser que el barco está girando a la derecha, pero repentinamente empieza a girar a la izquierda, esto provocaría que la predicción en lugar de mejorar la posición inicial de las dos nubes la empeore).

Como ya se ha mencionado el algoritmo nos permite obtener la posición del robot. Cuando ya se ha muestreado 2 veces con el sensor Velodyne podemos calcular una velocidad, calculando primero las diferencias de las coordenadas 3D, para saber cuanto ha avanzado, así como han variado los ángulos de Euler, para saber la variación de su orientación. Una vez hemos obtenido estas diferencias la dividimos por el periodo de muestreo de esas dos medidas y obtenemos la velocidad lineal y angular y al multiplicarlo por el periodo de muestreo obtenemos el desplazamiento del barco y al sumarlo a la última posición del barco obtenemos la predicción de la posición. En la figura 7 podemos ver el esquema de funcionamiento de la predicción.

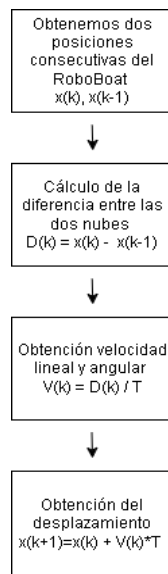


Figura 7. Esquema cálculo predicción sin IMU.

5.2.- Predicción con IMU

Ahora no tendremos que suponer que la velocidad lineal y angular del barco se mantiene constante durante el periodo de muestreo, debido a que el IMU nos dará esta variación en caso de existir. El periodo de muestreo del IMU es 3 veces menor que el del Velodyne. Por lo tanto, para cada predicción de posición haremos 3 correcciones de la velocidad del barco con respecto a la calculada.

Para calcular esta predicción calcularemos la velocidad actual del barco como en la predicción anterior con las 2 posiciones anteriores del barco. Una vez tenemos esta velocidad lineal, aplicamos las correcciones con los valores que nos devuelve el IMU. El IMU nos devuelve la orientación del barco con lo cual para la predicción solo hace falta sustituirla. Por otro lado, nos da las aceleraciones lineales del barco. Con estas aceleraciones y descomponiendo la aceleración provocada por la gravedad obtenemos con su periodo de muestreo la variación de velocidad en este periodo. Ello nos permite calcular la posición de una forma más precisa. Repitiendo este proceso 3 veces, es decir hasta que coincidan el muestreo del IMU y del Velodyne, podemos obtener de una forma mucho más precisa la posición del barco. Mediante esta predicción cabría esperar una mejora respecto a la predicción anterior en todos los casos y en los casos que podían provocar errores grandes en la predicción anterior notaremos una gran mejora. Ahora al conocer la orientación, el error producido con respecto la orientación debería ser muy pequeño y por otro lado al conocer la aceleración durante los muestreos del Velodyne

si hay un cambio brusco de velocidad en alguna dirección, provocará unas aceleraciones elevadas que haría que corrigiéramos la velocidad calculada. En la figura 8 se puede observar el esquema de funcionamiento de la predicción.

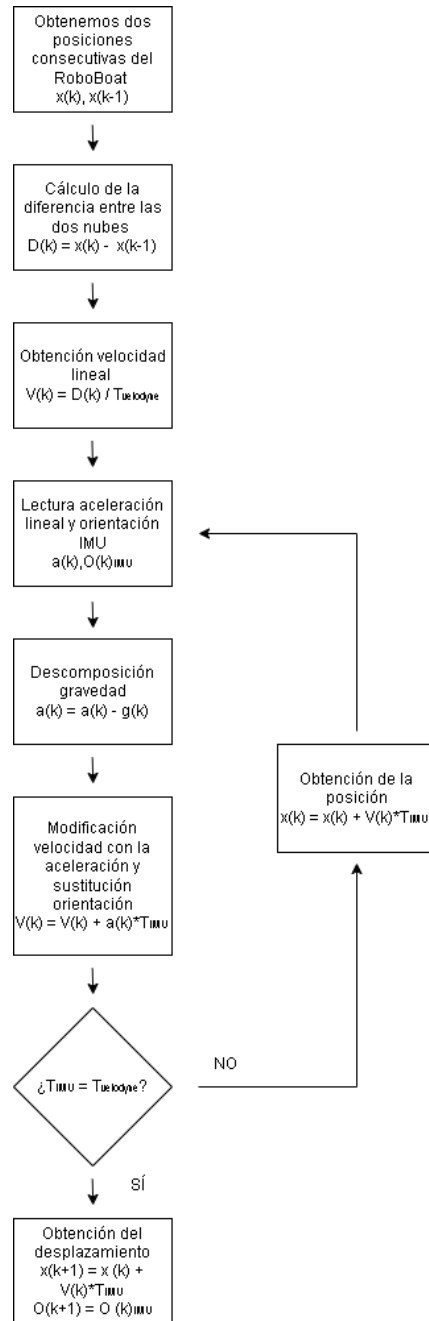


Figura 8. Esquema cálculo predicción con IMU.

6.- Métricas

Para valorar si la predicción suponía una mejora como cabría esperar respecto del cálculo sin predicción, y para valorar cuando una reconstrucción es mejor se utilizan distintas gráficas y valores numéricos, así como la visualización del mapa obtenido. Un ejemplo de una reconstrucción de la cueva sería la figura 9.

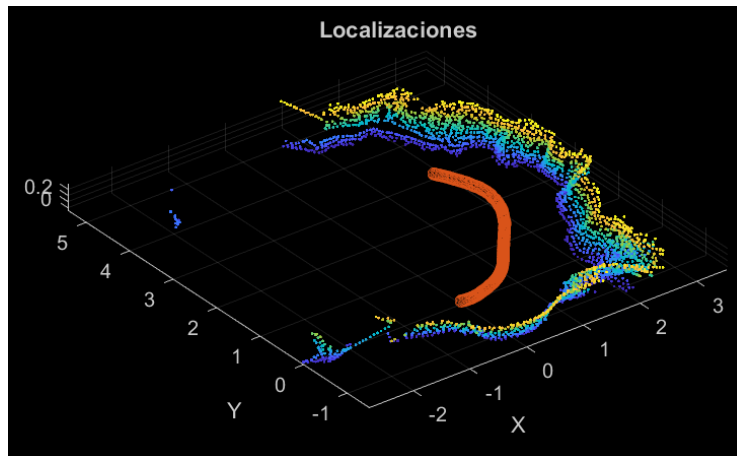


Figura 9. Mapa obtenido tras la reconstrucción de una zona de la cueva.

En primer lugar, la inspección visual de la reconstrucción es lo más importante debido a que las métricas pueden arrojar unos resultados mejores, pero si el mapa obtenido no es el de la cueva no sirve de nada.

Métrica de error del ICP: es una gráfica que nos devuelve el propio algoritmo con el error cometido tras la implementación del algoritmo. Se puede observar en la Figura 10.

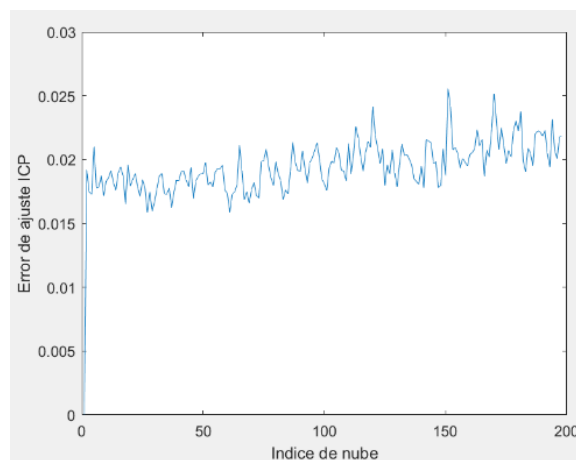
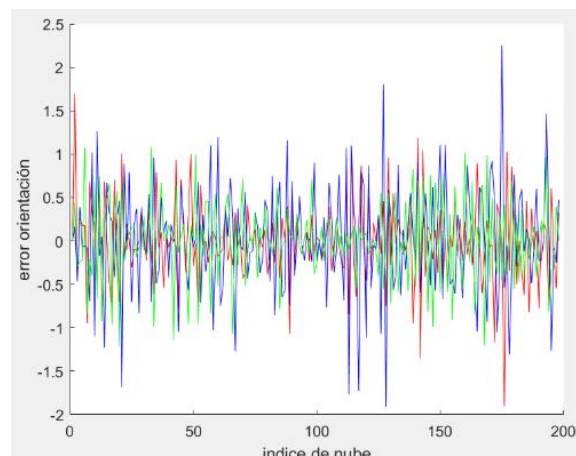
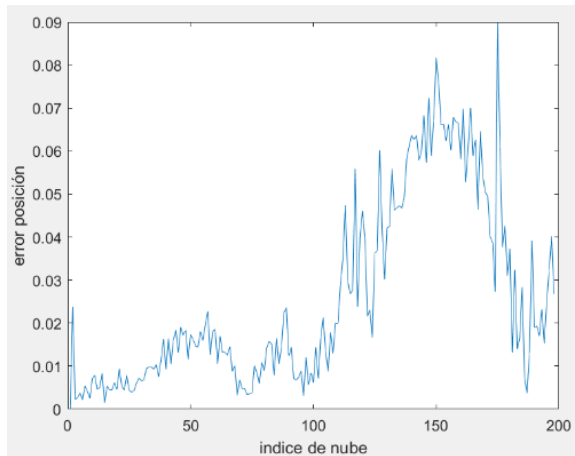


Figura 10. Métrica error ICP.

Esta gráfica nos da el error al emparejar cada nube con todas las anteriores. Por lo tanto, nos permite encontrar zonas con errores muy elevados y el error medio cometido.

Métrica de error de posición y orientación: Es el error en la posición inicial de las dos nubes con la posición de las dos nubes después de realizar el emparejamiento. Esas métricas son las correspondientes a las Figuras 11 y 12



Figuras 11 y 12. Muestran las gráficas del error de posición y el error de orientación.

Estas gráficas nos permiten visualizar picos en ciertas zonas donde se ha cometido un mayor error en la posición inicial de las nubes de puntos. A la hora de compararlas en la variación de parámetros pueden no resultar muy útiles. Por lo tanto, la media de los errores de posición y de orientación nos permitirá comparar de una mejor forma la calidad de la reconstrucción.

Métrica de error en Z: Como el barco está navegando en aguas calmadas que no tienen oleaje, podemos asumir que cualquier variación en el eje z corresponde con un error cometido en el emparejamiento de las nubes. Esta métrica se puede visualizar en la Figura 13.

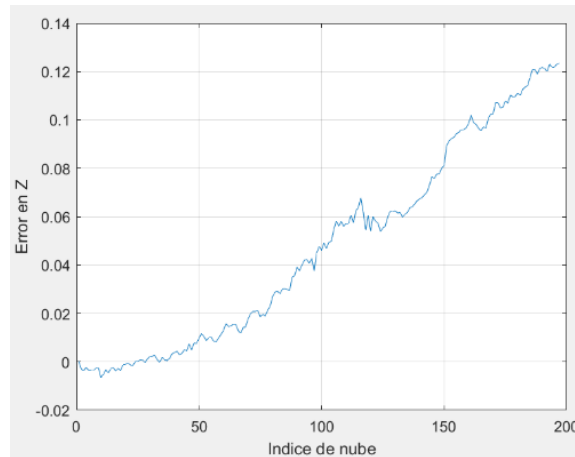


Figura 13. Métrica error en Z.

Estas primeras métricas son las que primero se utilizaron para una toma de contacto con el funcionamiento del algoritmo. Posteriormente se implementaron nuevas métricas para intentar medir el grosor de la pared. Si la reconstrucción es perfecta la orilla de la cueva y por lo tanto la pared tendría un grosor de 1 punto. Por lo tanto, si este grosor es mayor la reconstrucción ha sido peor.

Número de puntos vecinos: Partiendo de la base de que si el grosor de la pared es menor la pared tendrá menos puntos, contando los puntos que hay dentro de una esfera con centro un punto de la pared, podemos obtener un valor numérico relacionado con el grosor de la pared. A mayor grosor de la pared, mayor cantidad de puntos y peor calidad en la reconstrucción.

En primer lugar se planteó buscar zonas con un grosor mayor utilizando esta métrica. Para ello se implementó un método que generaba una esfera alrededor de todos los puntos y que contaba los puntos que había en el interior de dicha esfera. Posteriormente se calculaba la media de los puntos que tenía cada punto alrededor. Y teniendo en cuenta ese valor se realizaba el mapeado pintando los puntos en función de si estaban por encima o por debajo de la media.

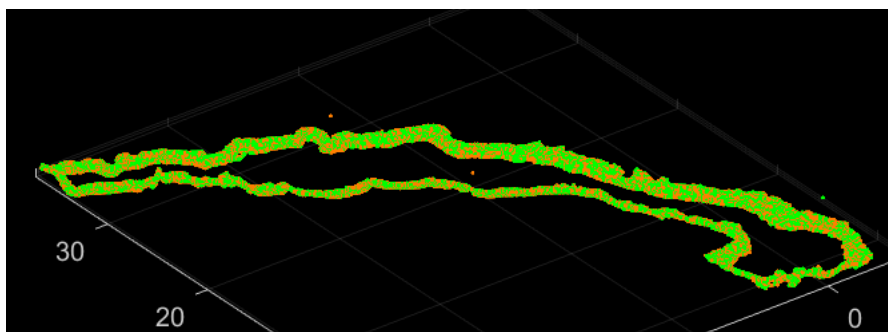


Figura 14. Mapa coloreado con número de puntos alrededor.

Como se observa en la Figura 14 para esta comprobación ya se utilizaba una reconstrucción completa es decir con los datos que recogía el RoboBoat alrededor de toda la cueva y no solo de una franja. En la figura estan pintados de verde los puntos por debajo de la media y de rojo los que estan por encima de la media

En las primeras reconstrucciones variando el tamaño de la esfera para coger más o menos puntos próximos no se observaban zonas donde se apreciara que había una concentración de puntos mayor. Por lo tanto, se decidió implementar cierta tolerancia y agregar un color neutro. Los puntos que estuvieran por debajo de la media y la tolerancia serían de color verde, los puntos alrededor de la media es decir dentro de la tolerancia de color rojo y los puntos que estuvieran por encima de la media y la tolerancia de color naranja.

Después de realizar esto y de intentar ajustar la tolerancia y el tamaño de la esfera para encontrar zonas de mayor concentración de puntos, como se observa en la figura 15 no se pudo encontrar tales zonas, lo que si se observó es que en los extremos inferiores y superiores de la nube de puntos aparecían por debajo de la media y los puntos de la zona central por encima como se observa en la figura 16. Resultado esperable debido a que en la zona inferior y superior si la esfera es lo suficientemente grande habrá una zona sin puntos debido a que se acaba la pared muestreada, mientras que en la zona central eso no ocurre.

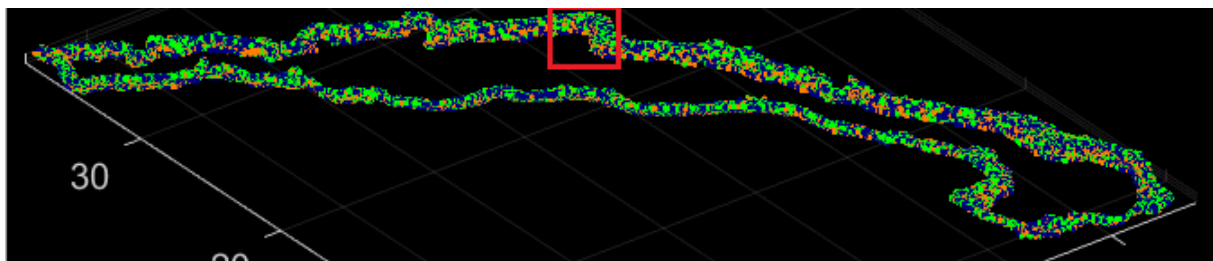


Figura 15. Mapa coloreado con número de puntos alrededor con tolerancia ajustada.

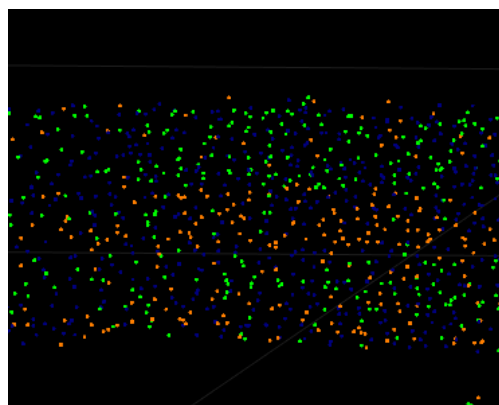


Figura 16. Detalle del rectángulo rojo de la Figura 16.

Distancia al plano: A continuación se implementó otro método para intentar medir el grosor de la pared de una forma distinta. Para ello se calculaba la normal a los puntos con los puntos de alrededor y se generaba un plano perpendicular a esta. Posteriormente se calculaba la distancia de los puntos de alrededor al plano que pasaba por el punto. Al igual que antes se intentó buscar zonas donde el grosor de la pared fuera mayor, pero ocurría algo similar que en el caso anterior, en el extremo inferior de la nube de puntos el grosor era menor que en la zona superior, como se observa en la figura 17.

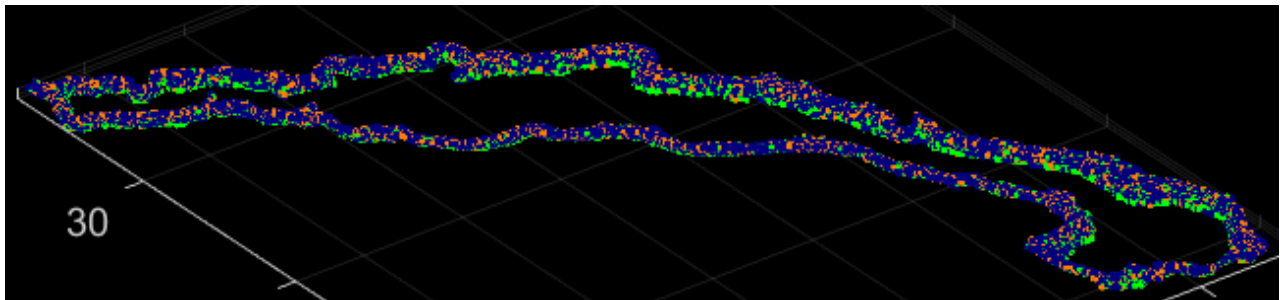


Figura 17. Mapa coloreado con distancia al plano.

Por lo tanto estas dos nuevas métricas no arrojaban resultados muy útiles para comparar la calidad de las reconstrucciones de la cueva o si había zonas que presentan un error mayor. Pero lo que si se podía apreciar es que la media de puntos alrededor de un punto era menor cuando el mapa era mejor o que la distancia media al plano era mejor cuando el mapa era mejor. Ambos resultados tienen sentido debido a que si la reconstrucción, y por lo tanto el mapeo de la cueva es perfecto, el grosor de la pared es menor y cada punto tendrá el mínimo de puntos alrededor y el plano que pasa por cada punto será coincidente con la pared. De estos dos valores numéricos es muy sencillo sacar un tercero que implicaría una mejor reconstrucción que es que el número de puntos sea menor, si hay menos puntos quiere decir que el emparejamiento ha sido mejor y por lo tanto todos o casi todos los puntos coinciden.

Por lo tanto, obtenemos 3 nuevas métricas: **número de puntos total, media de puntos alrededor de un punto y media de distancia al plano.**

7.- Reconstrucciones parciales

Para realizar una primera toma de contacto con los parámetros del algoritmo y como estos influían en la reconstrucción se realizaron reconstrucciones parciales del mapa de la cueva. Para reducir significativamente el tiempo de cómputo y evitar que la reconstrucción fuera errónea debido a configuraciones de parámetros extrema que no permitía la reconstrucción de la cueva.

Los parámetros para las primeras reconstrucciones fueron los que se observan en la tabla 1. Con la variación del parámetro a analizar.

DownSamplingMethod = 'gridAverage'	Este parámetro corresponde al método de reducción de muestreo mencionado en el apartado 3.
BaseGrid = 0.05	Este parámetro corresponde al tamaño de la rejilla mencionado en el apartado 3.
ICP_Metric = 'pointToPlane'	Este parámetro corresponde al método utilizado para calcular la distancia entre puntos explicado en el apartado 2.
ICP_Inlier = 0.9	Este parámetro corresponde al porcentaje de puntos seleccionado, es el opuesto al porcentaje de outliers mencionado en el apartado 2.
ICP_MaxIter = 40	Este parámetro corresponde al número máximo de iteraciones mencionado en el apartado 2.
ICP_Tolerance = [0.001 0.001]	Este parámetro hace referencia a la tolerancia de convergencia mencionado en el apartado 2.
MaxDistance = 10.0 MinDistance = 0.5 Max_Height = 0.5 Min_Height = -0.05	Estos parámetros corresponden a la distancia máxima y mínima de los puntos de la nube, así como a la altura máxima y mínima de los mismos mencionados en el apartado 3.
Min_Intensity = 15	Este parámetro corresponde al filtrado por la intensidad del rebote mencionado en el apartado 3.
ns = 20	Tiempo total de la reconstrucción en segundos.
offset = 10	Tiempo de offset de la reconstrucción debido a que los primeros valores obtenidos de los sensores pueden ser erróneos.

Tabla 1. Parámetros del algoritmo y de las nubes de puntos.

En las tablas posteriores los valores sombreados en verde corresponden al mejor valor obtenido en el análisis de ese parámetro. Por otro lado, los valores de la propiedad sombreados en amarillo corresponden a los valores que nos permiten obtener la mejor reconstrucción. Los parámetros que se van a analizar corresponden a Min_Intensity (Intensidad mínima del rebote), ICP_Metric (método de cálculo de distancia), MaxDistance (Máxima distancia), Max_Height (Altura máxima), todos ellos explicados en la tabla 1 y en los apartados anteriores, también se analizarán las predicciones.

En estas reconstrucciones se utilizaron las siguientes métricas: Gráfica de error del ICP, gráfica de error en Z, gráfica de errores de posición y orientación, media de errores de posición en metros (Media_P en las tablas) y media de errores de orientación mediante los 3 ángulos de Euler en grados (Media 1º ángulo, Media 2º ángulo, Media 3º ángulo en las tablas). Como se ha comentado en el apartado 3, la intensidad de rebote (Min_Intensity) tiene un rango de 0 a 100.

Min_Intensity	15	20	25	30	35	40	45
Media_P [m]	0.0268	0.0263	0.0258	0.0262	0.0261	0.0249	0.0263
Media 1º ángulo [°]	0.4061	0.3618	0.3512	0.3770	0.3323	0.3297	0.3610
Media 2º ángulo [°]	0.6427	0.6206	0.5571	0.5432	0.5677	0.5080	0.5396
Media 3º ángulo [°]	0.4088	0.3657	0.3012	0.3293	0.2973	0.3209	0.3747

Tabla 2. Análisis de la intensidad de rebote.

El filtrado de puntos por la intensidad de rebote analizado en la tabla 2 nos demuestra que un valor muy bajo no elimina posibles puntos que son reflejos. Mientras que un valor muy elevado para un mayor filtrado puede provocar que eliminemos puntos que no son reflejos y empeorar el emparejamiento debido a que en un escaneo el mismo punto puede que se considere reflejo y en el siguiente no.

ICP_Metric	Punto			Plano		
Min_Intensity	25	35	40	25	35	40
Media_P [m]	0.0303	0.0292	0.0276	0.0258	0.0261	0.0249
Media 1° ángulo [°]	0.3070	0.2554	0.2938	0.3512	0.3323	0.3297
Media 2° ángulo [°]	0.5498	0.5240	0.5117	0.5571	0.5677	0.5080
Media 3° ángulo [°]	0.3436	0.2966	0.3111	0.3012	0.2973	0.3209

Tabla 3. Análisis del método de cálculo de la distancia.

En cuanto al método de cálculo de distancia que se observa en la Tabla 3. Aunque en la orientación es mejor en general el método de punto a punto, las diferencias son de apenas unas centésimas de grado mientras que la diferencia de la media de error de posición es de algunos centímetros. Por lo tanto, podemos deducir que el método de punto a plano es mejor que el método de punto a punto.

MaxDistance [m]	4	6	8	10	12
Media_P [m]	0.0251	0.0252	0.0247	0.0248	0.0246
Media 1° ángulo [°]	0.3329	0.3026	0.3253	0.3227	0.2948
Media 2° ángulo [°]	0.4638	0.5194	0.4715	0.5068	0.4658
Media 3° ángulo [°]	0.3133	0.3133	0.3078	0.2965	0.3142

Tabla 4. Análisis de la distancia máxima.

Max_Heigh [m]t	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Media_P [m]	0.0261	0.0259	0.0258	0.0255	0.0257	0.0255	0.0258
Media 1° ángulo [°]	0.3748	0.3260	0.3498	0.3362	0.3371	0.3450	0.3405
Media 2° ángulo [°]	0.5215	0.4562	0.5058	0.4966	0.4845	0.4980	0.4829
Media 3° ángulo [°]	0.3243	0.3220	0.3614	0.2965	0.2805	0.2910	0.3399

Tabla 5. Análisis altura máxima.

De los parámetros de distancia y altura máximas analizados en las tablas 4 y 5 respectivamente. Se observa que minimizar el número de puntos de las nubes disminuye el tiempo de cómputo como era de esperar. Por lo tanto, reducir los parámetros de máxima distancia y de altura máxima nos permitirá reducir de forma notoria el tiempo de cómputo. También se observa que cuantos más puntos hay en las nubes el emparejamiento es algo mejor. Reducir mucho estos parámetros puede provocar que los emparejamientos no sean demasiado buenos.

Predicción sin IMU	Todo	Solo posición	Solo orientación	Ignorando Z	Ignorando Euler Z	Ignorando Z y Euler Z	Sin predicción
Media_P [m]	0.0259	0.0257	0.0298	0.0259	0.0254	0.0246	0.0294
Media 1° ángulo [°]	0.3260	0.6785	0.3285	0.3174	0.0974	0.0917	0.6808
Media 2° ángulo [°]	0.4562	0.3208	0.5177	0.5068	0.3302	0.2819	0.3224
Media 3° ángulo [°]	0.3220	0.2672	0.3569	0.3273	0.3336	0.3465	0.3211

Tabla 6. Análisis Predicciones sin IMU.

En la tabla 6 observamos el análisis de la predicción sin IMU. Se observa como la predicción mejora los resultados obtenidos respecto a las reconstrucciones sin predicción. Por otro lado, se comprobó si predecir únicamente la posición u orientación mejora de alguna forma el resultado y se observa que no mejora el resultado obtenido. Por último, dado que el barco navega en agua calmada y que en teoría la coordenada Z del barco debería ser constante y no variar respecto de la inicial. Se comprobó que sucedía si ignorábamos esta coordenada tanto en posición como en orientación al calcular la predicción. Y se observa como el error de posición y de orientación se disminuye notablemente si ignoramos la

predicción en la coordenada Z, y suministramos la misma como posición inicial de la segunda nube y que si ignoramos el segundo y tercer ángulo de Euler, que son los que afectan a la coordenada Z, también mejora la predicción. Por lo tanto, concluimos que la mejor predicción sin IMU es la que se obtiene al predecir las coordenadas X e Y del barco, así como el primer ángulo de Euler.

Predicción con IMU	Todo	Solo posición	Solo orientación	Ignorando Z	Ignorando Euler Z	Ignorando Z y Euler Z	Sin predicción
Media_P [m]	0.0234	0.0772	0.0739	0.0233	0.0732	0.0747	0.0294
Media 1° ángulo [°]	0.5363	1.3635	1.3692	0.5388	1.4526	1.4372	0.6808
Media 2° ángulo [°]	0.5061	0.1373	0.1293	0.5127	1.8263	1.7201	0.3224
Media 3° ángulo [°]	0.3824	0.0912	0.0926	0.3843	3.2028	3.1242	0.3211

Tabla 7. Análisis Predicciones con IMU.

Por último, la predicción con IMU que se analiza en la tabla 7 también mejora notablemente el error cometido en las reconstrucciones, pero a diferencia de la predicción sin el IMU ignorar el segundo y tercer ángulo de Euler no obtenemos una reconstrucción mejor e ignorando la coordenada Z mejora ligeramente el error de posición, pero es ligeramente peor en orientación. Por lo tanto, es mejor utilizar la predicción completa

Después de estas primeras reconstrucciones la comparación de las gráficas de error del ICP no aportan muchas diferencias y observando los mapas tampoco se aprecian muchas diferencias. Por otro lado, en las gráficas de error en Z se aprecian mayores diferencias, pero es complicado valorarlas debido a que la variación de la gráfica analizando un mismo parámetro es pequeña y solo es apreciable en los casos extremos.

Durante estas primeras reconstrucciones se observó una zona que podría provocar que la reconstrucción y por lo tanto el mapeo no fuera el adecuado, esta zona se observa en las figuras 18 y 19.

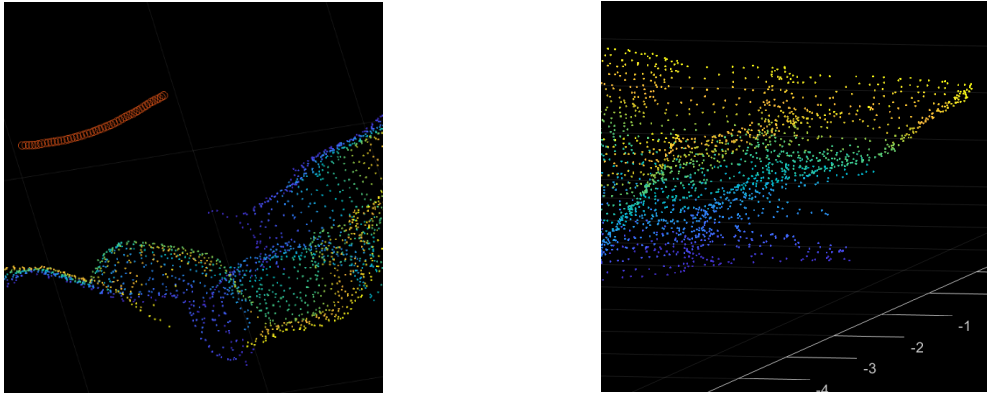


Figura 18 y 19. Zona con puntos erróneos. Estos puntos se encuentran tras la pared y están por debajo del barco.

Esta zona se encuentra por debajo del barco y lo que parece es que la pared se mete hacia dentro, lo que podría ser una grieta en la pared, pero debido a que se encuentra por debajo del barco lo más factible es que se trate de ruido en el sensor producido por el agua. Por lo tanto, es de esperar que si elevamos la altura mínima de los puntos de la nube, al ignorar esos datos las reconstrucciones tendrán una mejor calidad.

8.- Reconstrucciones completas

Con una idea de como influyen algunos de los parámetros y las predicciones funcionando de forma adecuada se busca la mejor parametrización del algoritmo para los datos de esta cueva. Para estas reconstrucciones ya se utilizaron todas las métricas explicadas en el apartado 6.

En una primera reconstrucción completa sin predicción, como se observa en la figura 20, el error cometido y la reconstrucción no es muy precisa. Esto podría deberse a que al tratarse de una reconstrucción completa estamos tratando muchas nubes y muchos puntos y la calidad de la reconstrucción va a ser peor que en las reconstrucciones parciales.

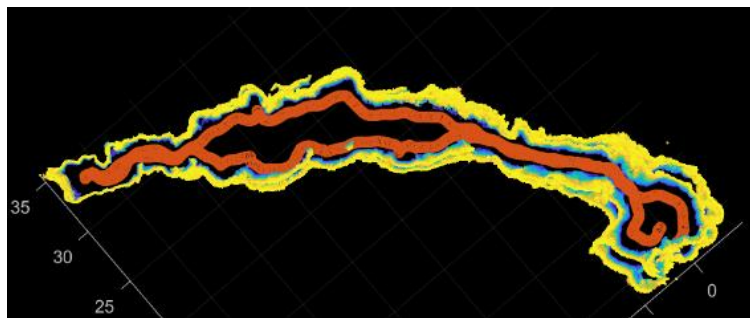


Figura 20. Mapeado completo sin predicción.

Aplicando la predicción no obtenemos un mejor resultado como refleja la figura 21. Aunque la reconstrucción es algo mejor debido a que las posiciones iniciales de las nubes de puntos es mejor, el resultado sigue sin ser el deseado.

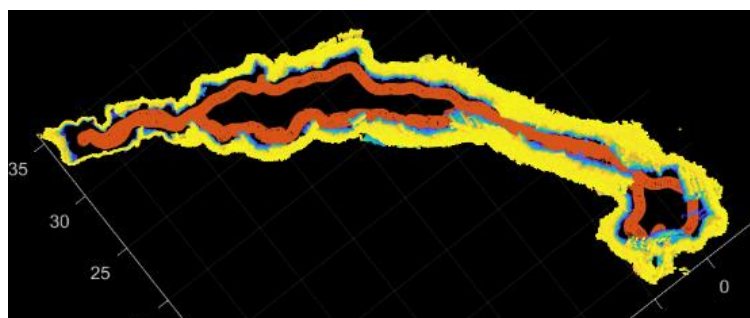


Figura 21. Mapeado completo con predicción.

Tras probar a variar los parámetros que ya se habían analizado en las reconstrucciones parciales se observa que la reconstrucción es algo mejor si se reduce mucho la distancia máxima de los puntos, algo que choca por completo con las reconstrucciones parciales donde a mayor distancia máxima mejor era la reconstrucción, pero mayor el tiempo de cómputo total.

Por otro lado, se observa una zona que en muchas ocasiones detenía la reconstrucción al no poder emparejar las nubes de puntos. La zona se observa en las figuras 22 y 23:

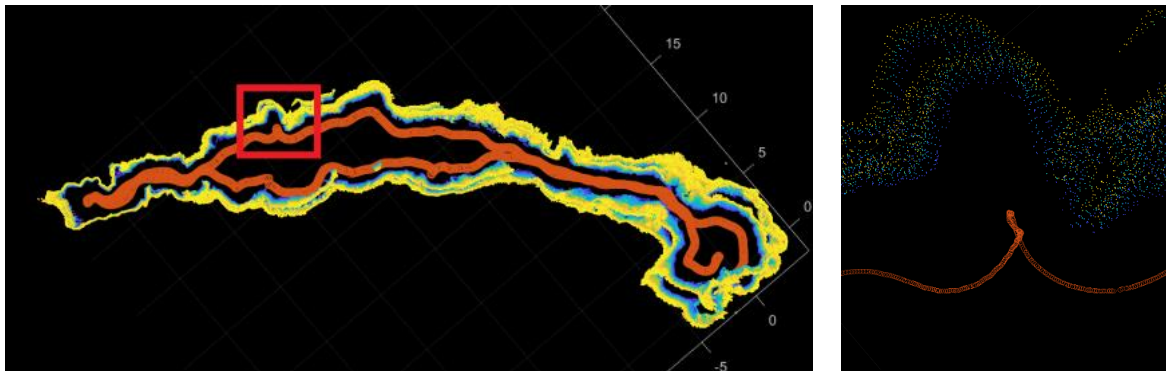


Figura 22, se observa el mapa completo de la cueva y la zona que provoca la detención del algoritmo recuadrada. Figura 23, ampliación de la zona recuadrada.

Esta zona provoca errores y detiene la reconstrucción debido a que el robot hace varios giros sobre sí mismo, lo que implica que las predicciones no sean muy buenas. Por otro lado, provoca que las nubes de puntos que se intentan emparejar sean muy diferentes, debido a que los puntos que están a la derecha, tras pocos muestreos pueden estar a la izquierda, y esto complica mucho el emparejamiento.

Por otro lado, se observa que sobre todo en la segunda mitad del recorrido es decir la vuelta hay ciertas zonas donde el barco va dando saltos en su posición y no es un movimiento continuo. Como se observa en la figura 24. Esto implica que el emparejamiento de al menos una nube de puntos con el resto ha sido erróneo, esto provoca que se emparejen puntos que en la realidad no son coincidentes, provocando un mal mapeado.

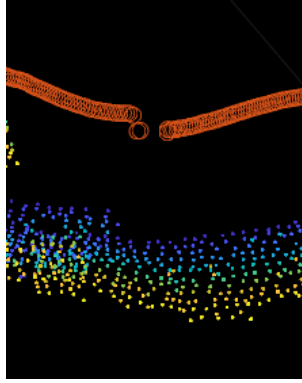


Figura 24. Zona perdida continuidad movimiento RoboBoat durante la vuelta de la reconstrucción.

En algunas reconstrucciones también se observa que el mapa no está bien cerrado y que en la vuelta el barco no está volviendo. Un ejemplo de estas reconstrucciones sería la figura 25.

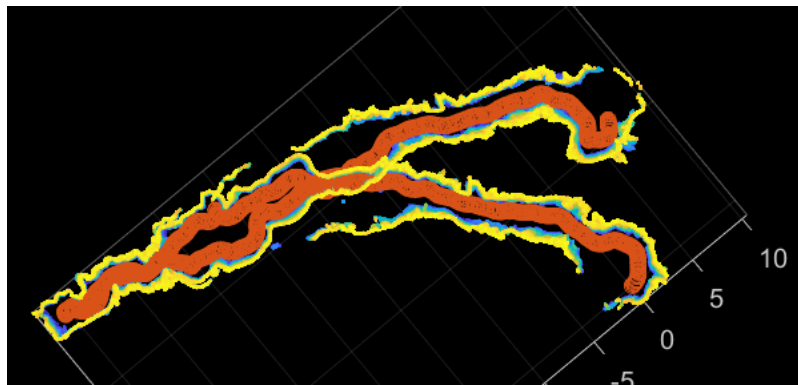


Figura 25. Mapeado completo con error en la vuelta del barco, no se cierra el mapa, la reconstrucción no acaba en el punto que empieza.

Analizando estos errores y sobre todo teniendo en cuenta que a veces no se cierra bien el mapa, se concluye que el motivo por el cual la reconstrucción era mejor para una distancia máxima menor es que cuando da la vuelta el barco y va por la otra orilla le cuesta emparejar los puntos que antes tenía a su derecha con los puntos que ahora tiene a la izquierda. Por lo tanto, lo que se va a hacer a partir de ahora es en vez de realizar la reconstrucción completa de los datos, emplear la mitad de los datos. Es decir, hasta cuando el barco da la vuelta y vuelve al origen. Por lo tanto, se deberá aumentar la distancia máxima para que el barco vea las dos paredes en todo momento.

Al igual que en las reconstrucciones parciales partimos de una configuración inicial y según vamos estudiando cada parámetro mantenemos el mejor resultado para estudiar los siguientes parámetros.

Tras analizar los parámetros que se encuentran en el anexo 1 podemos sacar conclusiones más precisas debido a que, al tratarse de un mapeado mayor, las diferencias en los resultados obtenidos son mayores, debido a que hay más nubes, más emparejamientos y más puntos. Unido a la utilización de todas las métricas podemos concluir de forma más certera que resultados son mejores y cual es la configuración óptima.

Como se observa en el anexo 1, se empezó estudiando la intensidad del rebote y su efecto en la obtención del mapeado. En la tabla 8 del anexo 1, al igual que en las reconstrucciones parciales, se observa que hay que encontrar un equilibrio entre un valor muy elevado y uno muy reducido para intentar eliminar únicamente los puntos que son causados por un rebote en el agua.

Posteriormente, se estudió como afectaba la altura y la distancia máximas de los puntos que tenemos en cuenta para la obtención del mapa mediante el ICP. Se encuentran los resultados en las tablas 9 y 10 del anexo 1 respectivamente. La altura máxima cuanto menor sea mejor, debido a que el número de puntos es menor, y esta disminución en el número de puntos tampoco provoca una peor reconstrucción, pero si disminuye el tiempo de cómputo. Por otro lado, como ya se ha comentado con anterioridad se ha observado que una distancia máxima menor mejora los resultados obtenidos a diferencia de las reconstrucciones parciales, esto es debido al error que provoca que se vean las dos paredes en todo momento. Como ya hemos comentado en este apartado, para las reconstrucciones completas, solo vamos a utilizar la mitad de los datos, por lo tanto, aunque una distancia máxima menor arroja unos mejores resultados, tenemos que ser capaces de realizar el mapeado de las dos orillas. El aumento de estos dos parámetros provocara un aumento en el tiempo de cómputo y en el número de puntos a emparejar, que no provoca una mejora en el caso de la altura máxima y provoca una ligera mejora en el caso de distancia máxima si se utilizan la mitad de los datos.

A continuación, se analizó el efecto que provocaba aumentar el número de iteraciones en la obtención del mapeado, para comprobar si se alcanzaba la tolerancia deseada en el número de iteraciones máximo. Se observa en la tabla 11 del anexo 1 que efectivamente el número de iteraciones máximo para la tolerancia aplicada era el adecuado, debido a que un número menor provocaba un peor resultado y que no se alcanzara la tolerancia, y un número mayor no mejoraba los resultados obtenidos debido a que ya se había alcanzado la tolerancia y se dejaba de iterar. En ciertas ocasiones no se conseguía alcanzar la tolerancia, pero aumentando las iteraciones lo único que hacían estas es aumentar el tiempo de cómputo sin lograr alcanzar la tolerancia, debido a que el algoritmo era incapaz de reducir el error. Posteriormente se aumentó el número de iteraciones y se redujo el valor de la tolerancia para ver si el algoritmo era capaz de ser más preciso, con las mismas iteraciones o aumentando estas, los resultados se pueden observar en las tablas 12 y 13 del anexo 1 respectivamente. Aumentando el número de iteraciones, se puede reducir la tolerancia del algoritmo y los resultados obtenidos serán mejores. Pero esto implica que

el tiempo de cómputo aumenta significativamente debido a que cada proceso de emparejamiento requiere un número mayor de iteraciones. Por lo tanto, estos dos parámetros deben ser estudiados de forma conjunta y hay que buscar un equilibrio y plantear cuan precisos queremos ser y que coste computacional podemos asumir. De todas formas, aunque se pueda reducir la tolerancia, la ganancia en la calidad de la reconstrucción no es muy significativa, por lo tanto, aumentar el tiempo de cómputo de forma tan notoria no es aconsejable.

Posteriormente se analizó como afecta el método de reducción de muestreo. Los resultados se reflejan en la tabla 14 del anexo 1. A priori cabría esperar que el método de promedio de cuadrícula no uniforme obtuviera unos resultados mejores debido a que este método elimina puntos teniendo más en cuenta las normales de los puntos respecto al método de promedio de cuadrícula uniforme como se comenta en el apartado 2. Pero al analizarlo se observa que el método de cuadrícula uniforme arroja mejores resultados en todas las métricas que estamos analizando, menos el número de puntos que es superior, lo que implica un ligero aumento del tiempo de cómputo, pero, a su vez, nos permite obtener unos resultados mejores, con lo que este ligero aumento en el número de puntos es asumible con la mejora que este aumento conlleva. Por otro lado, el método aleatorio es incapaz de realizar el mapeado de la cueva.

Posteriormente y ya sabiendo que el método de cuadrícula uniforme es mejor, se analizó el método de cálculo de la distancia de puntos. Resultados reflejados en la tabla 15 del anexo 1. Como ya se observó en las reconstrucciones parciales el método de cálculo de punto a plano es bastante mejor que el método de punto a punto. Aunque en este caso se observa que el método de punto a punto obtiene unos resultados mejores en la media de errores de posición y en la media de distancia al plano. Los errores cometidos en los ángulos de Euler, el número de puntos vecinos y el número total de puntos en el método de punto a plano es significativamente mejor respecto al método de punto a punto. Con lo que podemos concluir que el método de punto a plano es mejor que el método de punto a punto.

Por último, se intentó analizar el efecto que provocaba reducir la cuadrícula o aumentarla, debido a que aumentándola reducimos notablemente el número de puntos y reduciendo la cuadrícula aumentamos el número de puntos, pero el mapa debería ser más preciso. Como se observa en la tabla 16 del anexo 1, si aumentamos mucho el tamaño de la rejilla el algoritmo es incapaz de finalizar la reconstrucción, y en otros casos, aunque la reconstrucción es completa aparecen zonas con mucha densidad de puntos y zonas con muy pocos puntos, mientras que si reducimos mucho la rejilla con el número de iteraciones y tolerancia actuales la reconstrucción es peor. Si disminuyéramos la tolerancia y aumentáramos el número máximo de iteraciones cabría esperar que el resultado fuera mejor, pero esto provocaría un aumento en el tiempo de cómputo muy elevado respecto a la situación actual. Como se observa en la tabla el aumento en el número de puntos es bastante notable para la rejilla más pequeña aumentando

muy notablemente el tiempo de cómputo, si aumentáramos las iteraciones y redujéramos las iteraciones este se vería aumentado aun más por lo tanto no parece razonable hacerlo.

9.- Conclusiones

A lo largo del trabajo se han ido sacando una serie de conclusiones respecto a la parametrización del ICP, las métricas a utilizar y sobre la mejora que supone la predicción de una posición inicial.

Respecto a la mejora que supone la predicción:

- Los resultados obtenidos que utilizan la predicción basada en las dos últimas posiciones sin la utilización del IMU ignorando la coordenada Z y el segundo y tercer ángulo de Euler o con la utilización del IMU es mejor que no realizar una predicción.

Respecto a las métricas a utilizar:

- Las gráficas no permiten un análisis claro de los parámetros, pero nos permiten localizar zonas donde la reconstrucción ha sido peor.
- La media de errores de posiciones y orientaciones es un valor numérico que ayuda bastante en la comparación de los parámetros a estudiar. Un error menor en posición y orientación es indicativo de un mejor mapeado.
- El mapeado diferenciando los puntos por el número de puntos que tienen alrededor suyo, como el de distancia al plano, no arrojan ningún resultado más que la zona central tiene más puntos alrededor que los extremos superior e inferior del mapa.
- El valor medio de puntos alrededor de un punto y el valor medio de la distancia al plano nos arroja una forma de medir el grosor de la pared y por lo tanto la calidad del mapeado, provocando que un valor mejor implique un mapeado mejor.
- El número de puntos puede implicar un mapeado mejor si es menor dependiendo el parámetro que se esté analizando, por ejemplo, si se está estudiando la altura máxima de los puntos, un aumento del número de puntos no tiene porqué implicar que el mapeado sea peor.

Respecto a los parámetros estudiados:

- Aumentar la altura máxima de los puntos que utilizamos no mejora el mapeado y provoca que aumente el tiempo de cómputo. Valor óptimo 0.3 metros.
- Reducir la distancia máxima de los puntos que utilizamos mejora el mapeado y reduce el tiempo de cómputo. Debido a las características de la cueva estudiada tenemos que mantener una distancia máxima mínima que nos permita ver ambas paredes en todo momento. Valor óptimo 10 metros.

- La intensidad de rebote tiene que ser ajustada buscando un equilibrio entre un valor muy elevado y uno muy reducido. Valor óptimo 40.
- Reducir la tolerancia provoca que aumentemos el tiempo de cómputo notablemente y si no se aumenta también el número de iteraciones no mejorará los resultados, por lo tanto, ambos parámetros deben ser estudiados de forma conjunta. Valores óptimos 40 iteraciones y tolerancia 0.0005.
- El tipo de submuestreo a utilizar tendrá un gran impacto en el mapeado obtenido, mejor método rejilla uniforme.
- El método de cálculo de la distancia entre puntos tiene un gran impacto en el mapeado obtenido, siendo el método de punto a plano mejor respecto al método de punto a punto
- El tamaño de la rejilla tiene un gran efecto en el tiempo de cómputo y en la cantidad de puntos. Valor óptimo 0.05.

En conclusión, la predicción de las posiciones ayuda en gran medida en el correcto funcionamiento del ICP y siempre que sea posible mejorar las dos posiciones iniciales de las dos nubes de puntos mejorará el resultado obtenido. Por otro lado, las gráficas nos permiten observar de forma precisa zonas en la que se haya cometido un error en el mapeado o zonas donde el mapeado es peor pero difícilmente nos van a permitir comparar dos mapeados con una parametrización similar.

Los valores numéricos nos permiten comparar fácilmente dos mapeados con parametrizaciones similares, pero no nos permiten observar zonas donde el mapeado es peor. Por último, el número de puntos que se utilizan en el mapeado influyen de forma muy notoria, por lo tanto, la utilización de los menores puntos, pero sin utilizar menos de los necesarios, nos arrojará un mejor mapeado y nos permitirá reducir el tiempo de cómputo. También la utilización del método de punto a plano mejora notablemente el mapeado obtenido, así como ajustar de forma adecuada la tolerancia y el número de iteraciones.

Tras analizar todos los parámetros y los valores óptimos de estos y las predicciones la reconstrucción obtenida es la de la Figura 26 y Figura 27. Destacar que estas reconstrucciones son para obtener la posición real del RoboBoat durante el recorrido para una vez obtenida mezclar las nubes de puntos obtenidas por el Velodyne y el Hokuyo para obtener el verdadero mapeado de la cueva.

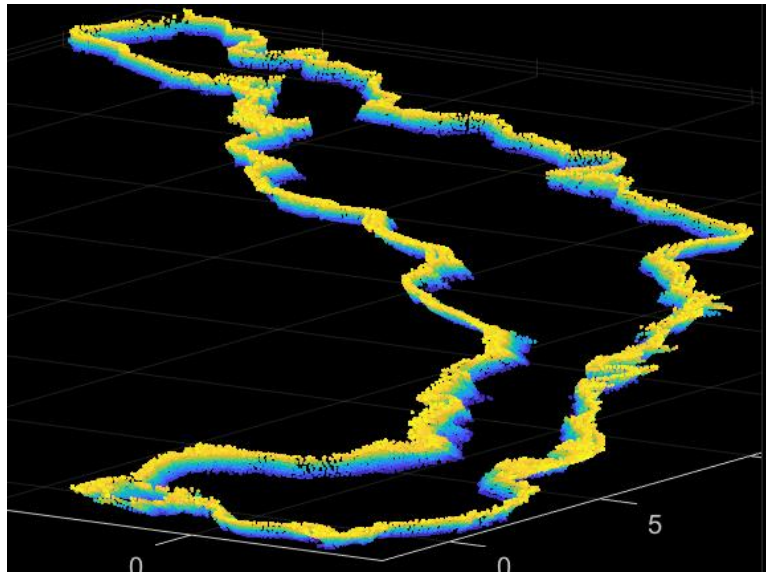


Figura 26. Reconstrucción completa con parámetros óptimos.

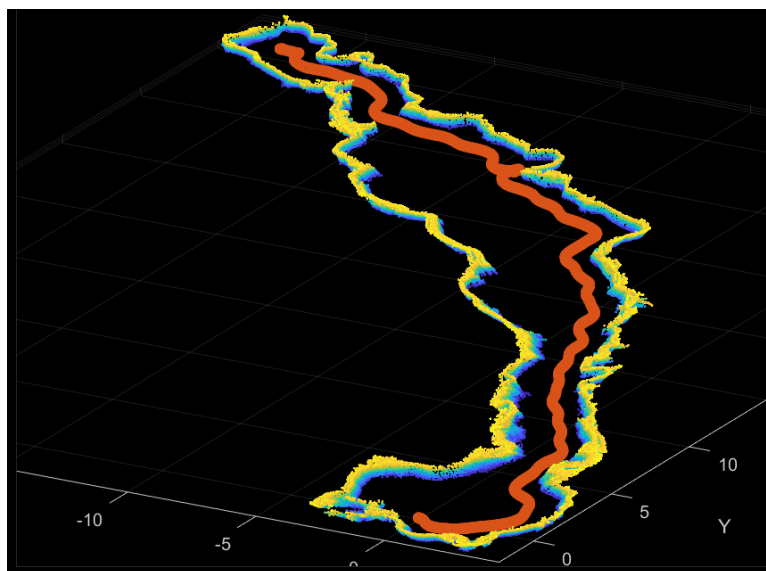


Figura 27 Reconstrucción completa con parámetros óptimos y localizaciones del RoboBoat

10.- Bibliografía

- [1] Paul J. Besl, *A method for registration of 3-D Shapes*, 1992.
- [2] M. L. Szymon Rusinkiewicz, *Efficient Variants of the ICP Algorithm*, 2001.
- [3] Aleksandr V.S., Dirk H y Sebastian T, *Generalized ICP*, 2009.

Anexo 1

Aquí se recogen las tablas obtenidas de las reconstrucciones completas obtenidas analizando los distintos parámetros al igual que en las reconstrucciones parciales. A parte de los valores numéricos aplicados a las reconstrucciones parciales también se utilizaron los valores:

- N puntos, correspondiente al número total de puntos de la reconstrucción.
- Media puntos, correspondiente a la media de puntos vecinos que tiene un punto.
- Media error plano, correspondiente a la media de distancia al plano en metros.

Todas son métricas explicadas en el apartado 6. Y los parámetros han sido explicados en el apartado 7.

Parámetros:

DownSamplingMethod = 'gridAverage'

BaseGrid = 0.05

ICP_Metric = 'pointToPlane'

ICP_Inlier = 0.9

ICP_MaxIter = 40

ICP_Tolerance = [0.001 0.001]

MaxDistance = 5.0

MinDistance = 0.5

Max_Height = 0.7

Min_Height = -0.05

Min_Intensity	20	25	30	35	40
N puntos	66354	64810	66720	67273	67365
Media puntos	317.62	316.91	324.7054	333.5440	332.8812
Media error plano [m]	0.0534	0.0533	0.0534	0.0536	0.0538
Media_P [m]	0.4627	0.4171	0.4163	0.4053	0.4111
Media 1º ángulo [°]	0.1548	0.1564	0.1764	0.1796	0.1938
Media 2º ángulo[°]	0.5618	0.4553	0.4224	0.3840	0.3718
Media 3º ángulo[°]	0.5406	0.4049	0.3854	0.3534	0.3562

Tabla 8. Análisis intensidad del rebote..

Como se ha comentado en el apartado 3 el rango de la intensidad del rebote es de 0 a 100.

DownSamplingMethod = 'gridAverage'

BaseGrid = 0.05

ICP_Metric = 'pointToPlane'

ICP_Inlier = 0.9

ICP_MaxIter = 40

ICP_Tolerance = [0.001 0.001]

MaxDistance = 10.0

MinDistance = 0.5

Min_Height = -0.05

Min_Intensity = 25

Max_Height [m]	0.3	0.5	0.7	0.9
N puntos	26453	43583	64810	107667
Media puntos	201.2101	263.9417	316.91	424.9735
Media error plano [m]	0.0534	0.0531	0.0533	0.0524
Media_P [m]	0.4170	0.4239	0.4171	0.4261
Media 1º ángulo [°]	0.1346	0.1459	0.1564	0.1579
Media 2º ángulo[°]	0.4575	0.4733	0.4553	0.4728
Media 3º ángulo[°]	0.4286	0.4216	0.4049	0.4276

Tabla 9. Análisis altura máxima.

DownSamplingMethod = 'gridAverage'

BaseGrid = 0.05

ICP_Metric = 'pointToPlane'

ICP_Inlier = 0.9

ICP_MaxIter = 40

ICP_Tolerance = [0.001 0.001]

MinDistance = 0.5

Max_Height = 0.3

Min_Height = -0.05

Min_Intensity = 25

MaxDistance [m]	4.0	6	8	10	12
N puntos	mal	21541	23688	26453	28426
Media puntos		167.2081	185.1273	201.2101	214.9667
Media error plano [m]		0.0515	0.0530	0.0534	0.0538
Media_P [m]		0.4244	0.4226	0.4170	0.4242
Media 1° ángulo [°]		0.1742	0.1551	0.1346	0.1373
Media 2° ángulo[°]		0.4653	0.4578	0.4575	0.4606
Media 3° ángulo[°]		0.4394	0.4250	0.4286	0.4321

Tabla 10. Análisis distancia máxima.

DownSamplingMethod = 'gridAverage'

BaseGrid = 0.05

ICP_Metric = 'pointToPlane'

ICP_Inlier = 0.9

ICP_Tolerance = [0.001 0.001]

MaxDistance = 6.0

MinDistance = 0.5

Max_Height = 0.9

Min_Height = -0.05

Min_Intensity = 25

ICP_MaxIter	20	30	40	50	60
N puntos	73474	69497	68639	69114	69038
Media puntos	307.4976	303.9436	296.4578	298.4764	298.4809
Media error plano [m]	0.0533	0.0534	0.0534	0.0536	0.0530
Media_P [m]	0.4301	0.4263	0.4264	0.4298	0.4239
Media 1° ángulo [°]	0.2161	0.1901	0.1713	0.1957	0.1737
Media 2° ángulo[°]	0.4758	0.4592	0.4748	0.4739	0.4821
Media 3° ángulo[°]	0.4333	0.4168	0.4348	0.4319	0.4228

Tabla 11. Análisis iteraciones máxima.

DownSamplingMethod = 'gridAverage'

BaseGrid = 0.05

ICP_Metric = 'pointToPlane'

ICP_Inlier = 0.9

ICP_MaxIter = 60

MaxDistance = 6.0

MinDistance = 0.5

Max_Height = 0.3

Min_Height = -0.05

Min_Intensity = 25

ICP_Tolerance [m]	[0.0001 0.0001]	[0.0003 0.0003]	[0.0005 0.0005]	[0.0007 0.0007]	[0.001 0.001]
N puntos	68656	69020	69450	70617	67469
Media puntos	297.8406	301.4535	299.5163	305.8211	293.0643
Media error plano [m]	0.0534	0.0535	0.0533	0.0534	0.0534
Media_P [m]	0.4282	0.4296	0.4327	0.4289	0.4236
Media 1° ángulo [°]	0.1911	0.1676	0.1864	0.2030	0.1759
Media 2° ángulo[°]	0.4685	0.4543	0.4733	0.4599	0.4615
Media 3° ángulo[°]	0.4310	0.4271	0.4472	0.4323	0.4268

Tabla 12. Análisis tolerancia con más iteraciones.

DownSamplingMethod = 'gridAverage'

BaseGrid = 0.05

ICP_Metric = 'pointToPlane'

ICP_Inlier = 0.9

ICP_MaxIter = 40

MaxDistance = 6.0

MinDistance = 0.5

Max_Height = 0.9

Min_Height = -0.05

Min_Intensity = 25

ICP_Tolerance [m]	[0.0001 0.0001]	[0.0005 0.0005]	[0.001 0.001]
N puntos	67416	67383	68639
Media puntos	294.0225	292.0099	296.4578
Media error plano [m]	0.0536	0.0536	0.0534
Media_P [m]	0.4256	0.4334	0.4264
Media 1° ángulo [°]	0.1956	0.1891	0.1713
Media 2° ángulo[°]	0.4644	0.4510	0.4748
Media 3° ángulo[°]	0.4602	0.4467	0.4348

Tabla 13. Análisis tolerancia con menos iteraciones

BaseGrid = 0.05

ICP_Metric = 'pointToPlane'

ICP_Inlier = 0.9

ICP_MaxIter = 40

ICP_Tolerance = [0.0005 0.0005]

MaxDistance = 6.0

MinDistance = 0.5

Max_Height = 0.9

Min_Height = -0.05

Min_Intensity = 25

DownSamplingMethod	nonuniformGridSample	gridAverage	random
N puntos	61311	66946	Mal
Media puntos	304.5832	291.5205	
Media error plano [m]	0.0545	0.0537	
Media_P [m]	0.4295	0.4279	
Media 1° ángulo [°]	0.2490	0.1811	
Media 2° ángulo[°]	0.4704	0.4552	
Media 3° ángulo[°]	0.4378	0.4264	

Tabla 14. Análisis método de reducción de muestreo.

DownSamplingMethod = 'gridAverage'

BaseGrid = 0.05

ICP_Inlier = 0.9

ICP_MaxIter = 40

ICP_Tolerance = [0.0005 0.0005]

MaxDistance = 6.0

MinDistance = 0.5

Max_Height = 0.9

Min_Height = -0.05

Min_Intensity = 25

ICP_Metric	pointToPlane	pointToPoint
N puntos	69625	113971
Media puntos	300.1727	465.1332
Media error plano [m]	0.0532	0.0517
Media_P [m]	0.4321	0.4251
Media 1° ángulo [°]	0.1987	0.2313
Media 2° ángulo[°]	0.4758	0.5533
Media 3° ángulo[°]	0.4360	0.5145

Tabla 15. Análisis del método de cálculo de la distancia.

DownSamplingMethod = 'gridAverage'

ICP_Metric = 'pointToPoint'

ICP_Inlier = 0.9

ICP_MaxIter = 40

ICP_Tolerance = [0.0005 0.0005]

MaxDistance = 6.0

MinDistance = 0.5

Max_Height = 0.9

Min_Height = -0.05

Min_Intensity = 25

BaseGrid [m]	0.025	0.05	0.075	0.1	0.1
N puntos	799548	110024	32740	Mal	Mal
Media puntos	3.3543e+03	440.8	128.3089		
Media error plano [m]	0.0251	0.05	0.0797		
Media_P [m]	0.4179	0.4322	0.4314		
Media 1° ángulo [°]	0.1714	0.2004	0.2243		
Media 2° ángulo[°]	0.5212	0.5602	0.5622		
Media 3° ángulo[°]	0.4583	0.5133	0.5112		

Tabla 16. Análisis tamaño de rejilla.