

DETECCIÓN DE INSTANCIAS DE OBJETOS EN UN SISTEMA DE LOCALIZACIÓN Y
MAPEO SIMULTÁNEOS BASADO EN VISIÓN DE MÁQUINA PARA ENTORNOS
DINÁMICOS



Ing. Brayan Andru Montenegro Embus

Trabajo de grado. Maestría en Automática

Director:

M.Sc. Juan Fernando Flórez Marulanda

Codirector:

M.Sc. Elena Muñoz España

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Maestría en Automática

Popayán, noviembre 2022

DETECCIÓN DE INSTANCIAS DE OBJETOS EN UN SISTEMA DE LOCALIZACIÓN Y
MAPEO SIMULTÁNEOS BASADO EN VISIÓN DE MÁQUINA PARA ENTORNOS
DINÁMICOS

Trabajo de grado presentado a la Facultad de Ingeniería

Electrónica y Telecomunicaciones de la

Universidad del Cauca para la obtención del

Título de

Magíster en Automática

Director:

M.Sc. Juan Fernando Flórez Marulanda

Codirector:

M.Sc. Elena Muñoz España

Popayán

2022

Doy gracia a mi madre Blanca Embus y mi abuelita María Elvia Embus, pilares y motivación de mi vida, ellas siempre como un regalo de Dios y bendición para mí. Agradezco a mis directores Juan Fernando Flórez Marulanda y Elena Muñoz España, por su incondicional apoyo en este proceso de crecimiento profesional, de los cuales no solo me llevo conocimientos, si no su calidez y la visión de que siempre se debe ser un docente con humanidad. Agradezco al programa de maestría en automática de la Universidad del Cauca y al Sena, entidades que desde diferentes frentes apoyaron este proceso de investigación, y extendiendo mi agradecimiento todos los colegas y personas, que han sido parte de este proceso y me han acompañado hasta este punto.

Resumen

El desarrollo de vehículos autónomos es un tema de gran interés en este siglo, lo que ha impulsado el desarrollo de investigaciones en esta área, como los sistemas de localización y mapeo simultáneo (SLAM), esta es una tecnología emergente y de gran potencial, que a su vez ha originado diferentes focos, como el Visual SLAM., En Visual SLAM se utilizan técnicas de visión artificial para captar el entorno, en este proceso se genera la reconstrucción de escenas y se obtiene el posicionamiento permitiendo así desarrollar mapas 3D de los entornos, estos desarrollos se encuentran en un proceso de constante evolución a la par con los nuevos dispositivos y nuevos métodos de tratamiento de datos; sin embargo aún se tiene diferentes brechas por superar, entre ellas el diseño de sistemas de bajo peso computacional para lograr la ejecución en línea en los sistemas autónomos.

Esta investigación tiene como objetivo “Proponer un método para la detección de instancias de objetos dinámicos por eliminación de la oclusión generada que permita la reconstrucción de la escena en 3D para la aplicación de mapeo y localización visual simultáneos robusto V-SLAM robusto”. Para lograr este objetivo se estructuró una metodología donde primero se realiza un preprocesamiento de los datos de un LiDAR y una cámara RGB-D, posteriormente se realiza la fusión sensorial, lo que permitió realizar la detección del objeto dinámico; con ello se segmenta y eliminan los puntos que pertenecen al objeto dinámico, dando como resultado una nube de puntos que no está completa por la oclusión generada por el objeto dinámico, y a partir de las nubes de puntos previas se realiza la concatenación sucesivas de nubes de puntos lo que permite reconstruir la escena. Se propone un método de reconstrucción por puntos de alta variabilidad de bajo peso computacional con ejecución en línea Rec-HV (reconstruction by points of high variability), y los resultados de pruebas de reconstrucción 3D de diferentes ambientes, con los sensores en posición fija y mientras realizan un recorrido y con presencia de un objeto dinámico que cruza la escena; se presenta también un método para la obtención de la correlación entre dos escenas como propuesta de una métrica para este tipo de desarrollos.

Contenido

| | |
|---|----|
| Capítulo 1. Introducción..... | 1 |
| 1.1. Objetivos..... | 3 |
| 1.1.1. Objetivo general | 3 |
| 1.1.2. Objetivos específicos..... | 4 |
| Capítulo 2. Revisión de la Literatura | 5 |
| 2.1. Introducción..... | 5 |
| 2.2. Estado del arte | 6 |
| 2.3. Marco conceptual | 9 |
| 2.3.1. SLAM..... | 9 |
| 2.3.2. Reconstrucción de ambientes estáticos en SLAM | 10 |
| 2.3.3. Odometría en SLAM..... | 12 |
| 2.3.4. SLAM en ambientes dinámicos | 13 |
| 2.3.5. Métricas..... | 15 |
| 2.4. Conclusiones..... | 17 |
| Capítulo 3. Metodología de la investigación..... | 18 |
| 3.1. Diseño del experimento | 18 |
| 3.1.1. Experimento 1 | 21 |
| 3.1.2. Experimento 2 | 22 |
| 3.1.3. Experimento 3 | 23 |
| 3.1.4. Experimento 4 | 23 |
| 3.1.5. Experimento 5 | 24 |
| 3.1.6. Experimento 6 | 25 |
| 3.1.7. Experimento 7 | 26 |

| | | |
|---------|--|----|
| 3.1.8. | Experimento 8 | 27 |
| 3.1.9. | Experimento 9 | 28 |
| 3.1.10. | Experimento 10..... | 29 |
| 3.1.11. | Experimento 11..... | 29 |
| 3.2. | Modelo Matemático..... | 30 |
| 3.2.1. | Preprocesamiento | 31 |
| 3.2.2. | Ángulo de detección..... | 33 |
| 3.2.3. | Reducción de la PCL..... | 34 |
| 3.2.4. | Marcos de referencia para el posicionamiento | 34 |
| 3.2.5. | Segmentación por ángulo | 35 |
| 3.2.6. | Segmentación por agrupación por profundidad | 36 |
| 3.2.7. | Traslación | 37 |
| 3.2.8. | Reconstrucción | 39 |
| 3.2.9. | Correlación por vistas localizadas | 40 |
| 3.3. | Implementación | 41 |
| 3.3.1. | Ejecución sistema REC-HV | 41 |
| 3.3.2. | Nodo detección..... | 44 |
| 3.3.3. | Nodo segmentación | 46 |
| 3.3.4. | Nodo Reconstrucción | 48 |
| 3.3.5. | Correlación | 48 |
| 3.4. | Materiales | 49 |
| 3.4.1. | Sensores..... | 49 |
| 3.4.2. | Montajes para los sensores..... | 53 |
| 3.4.1. | Objetos dinámicos. | 55 |
| 3.4.2. | Escenas | 56 |

| | |
|---|-----|
| Capítulo 4. Resultados..... | 59 |
| 4.1. Experimento 1..... | 60 |
| 4.2. Experimento 2..... | 62 |
| 4.3. Experimento 3..... | 67 |
| 4.4. Experimento 4..... | 69 |
| 4.5. Experimento 5..... | 70 |
| 4.6. Experimento 6..... | 72 |
| 4.7. Experimento 7..... | 77 |
| 4.8. Experimento 8..... | 79 |
| 4.9. Experimento 9..... | 82 |
| 4.10. Experimento 10..... | 85 |
| 4.11. Experimento 11..... | 87 |
| Capítulo 5. Discusión de los resultados | 96 |
| 5.1. Experimento 1..... | 96 |
| 5.2. Experimento 2..... | 96 |
| 5.3. Experimento 3..... | 97 |
| 5.4. Experimento 4..... | 97 |
| 5.1. Experimento 5..... | 98 |
| 5.2. Experimento 6..... | 99 |
| 5.3. Experimento 7..... | 99 |
| 5.4. Experimento 8..... | 100 |
| 5.5. Experimento 9..... | 101 |
| 5.6. Experimento 10..... | 102 |
| 5.7. Experimento 11..... | 103 |
| Capítulo 6. Conclusiones y recomendaciones..... | 105 |

Anexos.....107

- 6.1. Glosario y abreviaturas.....107
- 6.2. Glosario de ecuaciones.....107
- 6.3. Anexo 1. Error de reproyección110
- 6.4. Anexo 2. Matrices de calibración.....112
- 6.5. Anexo 3. Repositorios.....113

Referencias117

Tabla de Figuras

| | |
|--|----|
| Figura 1. Secuencia de desarrollo experimental en relación con los objetivos propuestos en esta investigación..... | 18 |
| Figura 2. Recorridos seleccionados para los experimentos..... | 21 |
| Figura 3. Factores controlados y no controlados experimento 1..... | 21 |
| Figura 4. Factores controlados y no controlados experimento 2..... | 22 |
| Figura 5. Factores controlados y no controlados experimento 3..... | 23 |
| Figura 6. Factores controlados y no controlados experimento 5..... | 24 |
| Figura 7. Factores controlados y no controlados experimento 4..... | 25 |
| Figura 8. Factores controlados y no controlados experimento 6..... | 26 |
| Figura 9. Factores controlados y no controlados experimento 7..... | 26 |
| Figura 10. Factores controlados y no controlados experimento 8..... | 27 |
| Figura 11. Factores controlados y no controlados experimento 9..... | 28 |
| Figura 12. Factores controlados y no controlados experimento 10..... | 29 |
| Figura 13. Factores controlados y no controlados experimento 11..... | 30 |
| Figura 14. Diagrama del procedimiento matemático del método Rec-HV..... | 31 |
| Figura 15 Marcos de referencia de los elementos que intervienen en los experimentos. | 35 |
| Figura 16. Vector traslación para cada marco de referencia con respecto al punto de origen. | 37 |
| Figura 17. Concatenación y reducción de PCL para la reconstrucción de escenas estáticas..... | 40 |
| Figura 18. Arquitectura del sistema Rec-HV. | 42 |
| Figura 19. SLAM con el método Rec-HV integrado. | 43 |
| Figura 20. Ángulos de segmentación, apertura del RGB-D y plano de detección del LIDAR..... | 47 |
| Figura 21. Segmentación por agrupación por profundidad..... | 47 |
| Figura 22. Correlación..... | 49 |
| Figura 23. (A) instrumentos de medición LIDAR y RGB-D en base soporte. (B) Campo visual de los instrumentos..... | 50 |
| Figura 24. (A) Ubicación de los sensores en el RGB-D. (B) Proyección de campo visual del LIDAR y del RGB-D..... | 51 |
| Figura 25. Imágenes usadas en el proceso de calibración. (A) imágenes de referencia. (B) imágenes donde se localizó la cuadrícula para calibración. | 53 |

| | |
|--|----|
| Figura 26. (A) Sensores sobre plataforma fija. (B) Sensores sobre plataforma de desplazamiento lineal. (C) Sensores sobre plataforma móvil automática..... | 55 |
| Figura 27. (A) objeto dinámico robótico con pantalla, (B) objeto dinámico. | 56 |
| Figura 28. Escenarios de pruebas..... | 58 |
| Figura 29. Distribución de los puntos captados por el LIDAR. (A) Escena estática sin objetos estáticos escenario A. (B) Escena estática con objetos estáticos escenario B. (C) Escena estática con objetos estáticos escenario B con inconsistencias. | 61 |
| Figura 30. Puntos no determinados por ciclo de lectura del LIDAR. | 62 |
| Figura 31. Lectura de puntos del LiDAR sin valores no reales. | 64 |
| Figura 32. Prueba valor de n , Columna n valores asignados a n . Columna A objeto dinámico ingresando a la escena. Columna B objeto dinámico en punto intermedio. Columna C objeto dinámico ya está saliendo de la escena. | 66 |
| Figura 33. Aplicación del promedio móvil para $n = 3$ | 67 |
| Figura 34. Valor de la desviación estándar (dr). (A) Escena estática A. (B) Escena estática A con objeto dinámico. (C) Escena estática B. (D) Escena estática B con objeto dinámico. | 69 |
| Figura 35. Tiempo promedio de ejecución, tiempo promedio de segmentación, tamaño promedio de la PCL para cada valor de <i>voxel grid</i> | 70 |
| Figura 36. Seguimiento del ángulo segmentado en cada ciclo donde fue detectado el objeto dinámico, (A) recorrido de izquierda a derecha del objeto dinámico, (B) recorrido de derecha a izquierda del objeto dinámico. | 71 |
| Figura 37. Aplicación del ángulo de detección sobre la PCL | 74 |
| Figura 38. Variación de la nube de puntos recorrido de derecha a izquierda. | 74 |
| Figura 39. Seguimiento de la segmentación de objetos dinámico aplicando el método de segmentación en dos etapas..... | 77 |
| Figura 40. Variación del número de puntos por la segmentación de dos etapas..... | 77 |
| Figura 41. (A) PCL escena estática, (B) PCL escena reconstruida. | 78 |
| Figura 42. Variación de la correlación entre la escena reconstruida y la escena de referencia. | 79 |
| Figura 43. En las filas escenarios son las pruebas y en las columnas se presenta el resultado de diferentes partes del proceso de reconstrucción..... | 80 |

| | |
|--|-----|
| Figura 44. Seguimiento al recorrido del objeto dinámico, (A) seguimiento de 5 recorridos paralelos eje y de izquierda a derecha. (B) seguimiento de 5 recorridos paralelos eje y de derecha a izquierda. (C) seguimiento a 5 recorridos transversales del objeto dinámico..... | 84 |
| Figura 45. Desplazamiento registrado por el sistema de odometría mecánica y odometría visual. | 85 |
| Figura 46. Error de desplazamiento por diferencia entre el sistema de odometría mecánica y el sistema de odometría visual (A) eje x, (B) eje y, (C) eje z. | 87 |
| Figura 47. Variación de la posición al realizar un recorrido, (A) variación eje x, (B) variación eje y, (C) variación eje z. | 89 |
| Figura 48. Variación de la velocidad durante el desplazamiento..... | 90 |
| Figura 49. Variación de los ángulos de Euler. | 90 |
| Figura 50. Escena reconstruida y recorrido de la plataforma con los sensores. (A) Plano XY. (B) Plano XZ. | 92 |
| Figura 51. Comparación de reconstrucción sin aplicar el método de eliminación de objetos dinámicos y con aplicación del método de eliminación. (A) vista frontal, (B) vista lateral, (C) vista superior de la reconstrucción..... | 94 |
| Figura 52. Variación del número de puntos de la PCL de reconstrucción..... | 95 |
| Figura 53. Variación de la correlación en la reconstrucción durante un recorrido. | 95 |
| Figura 54. Error de reproyección. | 110 |

Capítulo 1. Introducción

El desarrollo de sistemas autónomos ha sido un tema de gran interés en las últimas décadas, dado que muchas industrias ven la posibilidad de implementarlos como parte de sus procesos (Bilberg & Malik, 2018). Dentro de esta línea están los sistemas de navegación autónoma, estos deben tener la capacidad de desplazarse en un entorno, manteniendo el seguimiento de su posición y mapeando el entorno de forma constante, adaptándose a todas las condiciones del ambiente (Cadena, et al., 2016; Rehman, et al., 2018). La localización y mapeo simultaneo (SLAM) (Chong, et al., 2015; Cheng, et al., 2015; Astrup, et al., 2018; El-Sheimy, et al., 2019; Elliott, et al., 2019), se ha convertido en el foco de múltiples desarrollos que buscan generar aportes (Garcia-Fidalgo & Ortiz, 2015; Sahin, et al., 2020).

SLAM ha ido de la mano con la evolución del hardware, donde se han implementado nuevas tecnologías para el censado del ambiente (Chong, et al., 2015; Hsu, Ya-Wen, et al., 2018), a permitiendo tener mejores y más precisas fuentes de información. Desarrollos como cámaras RGB (Huang, et al. 2016; Meng, et al., 2017; Meng, et al., 2018; Fan, et al., 2018; Akanshu, et al., 2018; Oishi, et al., 2019), sensores LIDAR (Arévalo, et al., 2016; Cheng, et al., 2018; Javanmardi, et al., 2019; Miura, et al., 2019) para realizar el mapeo en 2D y 3D y sistemas de posicionamiento GPS (Astrup, et al., 2018), IMU (Astrup, et al., 2018) e INS/GNSS (Chang, et al., 2019), etc. Estos han permitido el desarrollo de sistemas autónomos, siendo la base para nuevas investigaciones, consolidando los aportes de la metodología SLAM.

Dentro de las líneas que han tomado gran relevancia está, el Visual SLAM (Markham et al., 2018), en la que se busca desarrollar sistemas, que basados en la aplicación de métodos de procesamiento de imágenes y nubes de puntos, que permiten obtener el posicionamiento (Abramenko, et al., 2021) y captar el ambiente mientras se realiza un recorrido (Ao, et al., s.f). Este tipo de desarrollos ha tenido un gran impacto, permitiendo generar mapas de reconstrucción de entornos estáticos (Cao, et al., 2019; Hernández & Ortiz, 2020; Hongmin, et al., 2020). Visual SLAM presenta tres líneas de investigación (Markham et al., 2018): SLAM visual robusto, segmentación dinámica de objetos y seguimiento 3D y reconstrucción y segmentación de objetos en movimiento. Cada una de las

líneas presentan diferentes retos sobre los cuales ya se están haciendo aportes (Yan & Zha, 2019); entre ellos se destaca la adquisición y filtrado de datos (Batista et al., 2020; Pan, et al., 2020; Kiani & Escalera, 2021), mapeo y reconstrucción de la escena (Feng, et al., 2020) y determinación del posicionamiento (Hernández & Ortiz, 2020). En cada uno se aplican diferentes técnicas de procesamiento (Czerniawski & Leite, 2020; Ding & Tang, 2020), y se utilizan métodos de segmentación y reconocimiento (Wang, 2020), con un alto flujo de datos sincrónicos (Geng, et al., 2020). Por ello se han propuesto algoritmos de procesamiento que usan métodos matemáticos y estadísticos de gran complejidad (Mazurek, et al., 2020; Huynh-The, et al. 2020), que tienen un alto costo computacional, el cual es medido por el consumo del recurso computacional y el tiempo de ejecución del algoritmo (Cordeiro & Pedrino, 2019). Este ha sido el precursor del desarrollo de métodos que buscan la reducción del costo computacional de procesamiento (Chen et al., 2021; Liu & Wang, 2021), mejorando de forma eficaz los diferentes procesos de SLAM, los cuales son medidos a partir del error de posicionamiento y la comparación cualitativa de la reconstrucción (Chen & Cho, 2018; Li, 2020; Huixia, et al., s.f), siendo esta última una métrica subjetiva y sobre la cual aún se debe generar investigación.

En la reconstrucción de escenas se puede obtener un plano en 2D, el cual se caracteriza por ser el contorno de la escena y tener un bajo costo computacional; sin embargo, esta presenta información limitada del espaciado y distribución volumétrica (Haoran, et al., 2022). Por otro lado, se obtiene una reconstrucción en 3D, la cual se caracteriza por tener una gran cantidad de puntos, lo que permite tener un gran detalle de la distribución de la escena (Guangyun, et al., 2020); sin embargo, estas presentan un alto costo computacional debido al procesamiento de la nube de puntos.

La reconstrucción de escenas en entornos dinámicos-SLAM dinámico (Xiao et al., 2019), ha sido el foco de los procesos de investigación en los últimos años, este trae consigo una serie de retos; entre ellos, generar la detección del objeto dinámico (Ai, et al., 2021). Se han presentado métodos de detección en 2D, que utilizan imágenes RGB sobre las cuales se aplican métodos de comparación (Chacon, et al., 2019) o aplicación de redes neuronales para la detección del objeto dinámico. En otros métodos se realiza la detección sobre nubes de puntos (Wang, et al., 2020), estas se caracterizan por ser complejas y de un alto costo computacional de procesamiento. En estos casos se ha logrado obtener sistemas que detectan el objeto dinámico; sin embargo, debido al

costo computacional, la detección y eliminación de objetos dinámicos y reconstrucción de escenarios presenta una gran dificultad para ser ejecutadas al mismo tiempo en línea (Sun, Liu, & Meng, 2018; Chen, & Cho, 2018; Cordeiro & Pedrino, 2019; Liu, et al. 2019; Wang, et al., 2020; Huixia, et al., 2020; Huixia, et al., s.f). Este tópico está abierto aún a la investigación y presenta gran interés, dada a la integración de vehículos autónomos en ambientes reales con humanos.

La reconstrucción de escenas con objetos dinámicos de bajo peso computacional, son una problemática en SLAM, por ende, se plantea la siguiente pregunta de investigación ¿Qué correspondencia se puede obtener entre el mapa reconstruido por procesamiento y el ambiente real estático, al aplicar detección y eliminación de objetos en entornos dinámicos?

En esta investigación se presenta un nuevo método de bajo peso computacional, para la eliminación de objetos dinámicos y reconstrucción 3D en SLAM, en este se aplicó la técnica de fusión sensorial de un LIDAR 2D de bajo costo y una cámara RGB-D, se incorporó técnicas de Visual-SLAM robusto como ORB-SLAM; para generar la reconstrucción de ambientes con objetos dinámicos, hacia sistemas autónomos de navegación en ambientes interiores. Se presentarán los resultados del procesamiento y tratamiento estadístico de los datos, que permite la detección de las zonas de alta variabilidad donde se aloja el objeto dinámico, se presenta la segmentación y eliminación en línea del objeto dinámico, mientras se realiza un recorrido, y la reconstrucción resultante del ambiente por segmentación y concatenaciones de los *point cloud*. También se propone un método para establecer una métrica de comparación entre dos escenas 3D, en este se establece la correlación entre dos escenas, las cuales tienen la misma perspectiva desde una idéntica posición relativa a la escena. Adicional se presenta el artículo *Dynamic reconstruction in simultaneous localization and mapping based on the segmentation of high variability point zones* (Montenegro et al., 2022), el cual presenta resultados parciales de esta investigación.

1.1.Objetivos

1.1.1. Objetivo general

Proponer un método para la detección de instancias de objetos dinámicos por eliminación de la oclusión generada, que permita la reconstrucción de la escena en 3D para la aplicación de mapeo y localización visual simultáneos robusto V-SLAM robusto.

1.1.2. Objetivos específicos

- Determinar la posición y trayectoria de instancias de objetos dinámicos en un entorno 3D, a partir de la aplicación de algoritmos para el tratamiento de imágenes.
- Aplicar algoritmos para la eliminación de los objetos dinámicos de la escena, para la reconstrucción del mapa.
- Evaluar la correlación del mapa virtual tridimensional reconstruido con relación al entorno real en presencia de instancias de objetos dinámicos.

Capítulo 2. Revisión de la Literatura

2.1.Introducción

La localización y mapeo simultaneo (SLAM) es uno de los temas que ha captado la atención en los últimos años, convirtiéndose en un tópico de gran relevancia para el desarrollo de la robótica (TaHERi & Xia, 2021), resaltando el desarrollo de vehículos autónomos como uno de los retos del siglo XXI.

El desarrollo de sistemas robóticos autónomos es uno de los grandes ejes que capta la atención de los investigadores de diferentes disciplinas, siendo esta una línea de investigación muy diversa, presentando diferentes frentes desde donde brindar aportes y generar nuevas tecnologías. Entre este eje se encuentra el desarrollo de vehículos autónomos. Esta línea presenta grandes retos desde el diseño de hardware y software y su integración. Para tener vehículos autónomos se debe desarrollar diferentes técnicas de procesamiento de grandes volúmenes de datos, ya que estos dispositivos deben trabajar con diferentes sensores de forma simultánea; entre estos, sensores que permitan captar el entorno como, cámaras de profundidad y RGB y sensores para el posicionamiento y así tener sistemas de navegación robustos, los cuales permiten incursionar en cualquier ambiente. Sin embargo, el desafío aumenta de forma significativa cuando se desea tener vehículos con la capacidad de realizar recorridos con presencia de objetos dinámicos en el entorno, siendo uno de los tópicos que causa gran interés por el reto que representa, desde el desarrollo de sistemas de visión de maquina y técnicas de segmentación y reconocimiento. Esto ha permitido el desarrollo de métodos y tecnologías que van generando aportes y van dilucidando la posibilidad de generar sistemas integrados para la navegación.

Todo este énfasis en cual se viene trabajando, ha sido el precursor para que en los últimos 30 años se genere de manera continua diversas herramientas tecnológicas, resultante de los procesos de investigación enfocados a lograr el desarrollo de vehículos autónomos con la capacidad de enfrentarse a entornos reales.

2.2.Estado del arte

En la navegación autónoma, la visión artificial 3D es el punto de partida en el Mapeo y Localización Simultáneo (SLAM) (TAN, et al., 2019). Él tiene como objetivo la reconstrucción en línea del entorno (An, et al. 2017; Akanshu, et al., 2018), generando una nube de puntos tridimensionales (Sahin, et al., 2020), que junto con sensores de posicionamiento (Gallant & Marshall, 2016) y fusión datos (Narváez, et al., 2016), desarrollan el concepto de SLAM en entornos estáticos (Tiwari & Chong, 2020). En SLAM se tienen varios desafíos, entre ellos la investigación de sistemas que tengan la capacidad de definir y detectar los objetos estáticos inmersos en el ambiente, este desafío es aún mayor cuando en el ambiente están inmersos objetos dinámicos (LI, et al., 2019). La detección dinámica de objetos es todavía un reto en crecimiento, ya que la oclusión generada en la escena por el movimiento de los objetos dinámicos imposibilita la reconstrucción total (Chacon, et al., 2019).

La reconstrucción de entornos ha sido el precursor para el surgimiento de líneas como Visual SLAM, en este se utilizan sensores para recolectar imágenes y nubes de puntos, que permiten la extracción del posicionamiento y generan mapas 2D y 3D (LI, et al., 2019; Tan, et al., 2019). Para originar mapas en 2D se usan métodos de bajo peso computacional (Hol'y, 2016). Para el preprocesamiento de los datos, en (Androulakis, et al., 2021) se propone aplicar tratamiento por RANSAC a los puntos del LIDAR 2D con el fin de identificar mejor las esquinas del ambiente estático y recuperar zonas; en (Haoran, et al., 2022) se propone realizar un preprocesamiento por vecinos cercanos en los puntos del LIDAR 2D, permitiendo así, eliminar el ruido y los puntos atípicos en la detección de obstáculos en una escena estática. En (Fukao,et al., 2021), se propone limitar el rango máximo de profundidad y así filtrar los puntos lejanos y los puntos atípicos, para captar entornos en carreteras agrícolas. En todos los casos, se tiene un bajo costo computacional en la reconstrucción de escenas estáticas sobre un plano 2D; sin embargo, este brinda poca información del entorno. Al originar mapas 3D del entorno, se ven inmersos muchos más procesos por la complejidad del tratamiento de las nubes de puntos, entre ellos filtrado de datos (Batista et al., 2020; Pan, et al., 2020; Escalera, et al., 2021) y técnicas de procesamiento (Ma, et al., 2020; Ding, et al., 2020), mapeo y reconstrucción de la escena (Feng, et al., 2020) y determinación del posicionamiento (Hernández & Ortiz, 2020), en donde se usan métodos matemáticos y estadísticos de gran complejidad (Morawski, et al., 2020; Huynh, et al., 2020). Todos estos procesos han

logrado obtener reconstrucción de escenas 3D con gran fidelidad al entorno real; sin embargo, se caracterizan por ser de un alto costo computacional.

El costo computacional es un factor crítico (Dalcin, et al., 2015), este ha sido un precursor para el desarrollo de métodos de reducción del costo computacional, aplicando segmentación por múltiples técnicas y simplificación de *point cloud*. En (Siddiqua & Fan, 2019) se aplica un supervisor para la extracción de características para segmentación y así procesar la escena de forma distribuida, en (Chen, et al., 2021) aplican la extracción de profundidad monocular para la generación de escenas 3D, siendo este de menor costo computacional que el procesamiento de PCL, en (Liu & Wang, 2021) se procesa las PCL por matriz de ocupación segmentándola en áreas de superficie constante, estos métodos si bien reducen el costo computacional solo son funcionales sobre escenas estáticas.

La reconstrucción de escenas dinámicas es aún más compleja, la oclusión es un factor que imposibilita la reconstrucción con un solo *frame* o por la concatenación de *frame* sucesivos por la presencia del objeto dinámico (Guangyun, et al., 2020). Esto conduce a la investigación en métodos de detección de objetos dinámicos en sistemas 2D donde se han presentado grandes avances, en (Chacon, et al., 2019) se presenta un panorama de las técnicas de detección de objetos dinámicos en secuencias de imágenes, resaltando diferentes aportes, pero también las grandes dificultades.

El desafío es aún más significativo en la detección dinámica de objetos en 3D; en (Wang, et al., 2020) se realiza la detección de objetos en movimiento, cuyas velocidades están limitadas por instrumentos como el LIDAR 3D que solo alcanzan 10Hz de muestreo. A pesar de las limitaciones de hardware de procesamiento y de sensado, ya se han expuestos diferentes métodos diseñados para ese fin, en (Cordeiro & Pedrino, 2019) se establece la trayectoria de un objeto dinámico a una máxima velocidad de 3,125m/s. En (Wu, et al., 2019) se presenta el método *On-board Diagnostics* (OBD) para estimar la trayectoria de objetos dinámicos con un LIDAR 3D, con ejecución a 2 Hz. En (Liu, et al., 2018) se realiza la segmentación de un objeto dinámico en una escena, aplicando estimación de flujo óptico denso con ejecución a 8 s por *frame*. En (Fan, et al., 2018) se propone un método para la eliminación de objetos dinámicos por fusión de imágenes, aplicado fuera de línea sobre *datasets*. En estos trabajos se enmarca la necesidad de nuevos desarrollos para la detección de objetos dinámicos a bajas velocidades o de grandes dimensiones, como un siguiente paso para el desarrollo de esta línea de investigación. Todos estos aportes pretenden generar nuevos

métodos para la reconstrucción de nubes de puntos con la menor cantidad de puntos pertenecientes al objeto dinámico y así generar una reconstrucción total de la escena, siendo este un objetivo que no ha sido alcanzado con ejecución en línea y bajo peso computacional.

El desarrollo de nuevos métodos en SLAM trae consigo la necesidad de la incorporación de métricas para la evaluación de los resultados obtenidos, esta es un área aún en desarrollo. Se han diseñado diversos métodos que facilitan la implementación de métricas como el error absoluto, el error cuadrático medio y la desviación estándar. En sistemas que generan reconstrucciones 2D se evalúa el contorno o ruta establecida; en (Haoran, et al., 2022) se evalúa el error de los puntos comparándolos con la superposición del entorno real; en (Androulakis, et al., 2021) se evalúa el promedio del error sobre la ruta total y se evalúa la tasa de éxito al aplicar el algoritmo donde un vehículo de forma automática genera una ruta para superar una serie de obstáculos, en (Hol'y, 2016) se presentan varios sistemas donde se evalúa la diferencia entre la ruta original y los puntos de posicionamiento y orientación registrados.

Cuando se trata de sistemas 3D, las métricas son limitadas. En (Cordeiro & Pedrino, 2019) se muestra una comparación del consumo de los recursos computacionales y la detección se la evalúa de forma subjetiva, en otros sistemas se realiza seguimiento de un recorrido mientras se capta una escena estática en 3D. En (Chen, et al., 2018) se presenta una evaluación cualitativa de los requerimientos y lo obtenido por diferentes sistema; entre ellos si requiere conocer la escala absoluta de posicionamiento, si genera mapa de alta resolución, si genera un mapa 3D y si se tiene mapeo RGB de puntos; posteriormente, se realizan pruebas en diferentes ambientes interiores y exteriores, determinando el resultado de precisión de registro entre cada escaneo estático con el método propuesto y resultado de precisión de registro entre cada escaneo estático, en función la coincidencia del plano. En (Hongmin, et al., 2020) se da a conocer un sistema de reconstrucción en el cual se evalúa el crecimiento de la *point cloud* (PCL), después se compara cualitativamente la reconstrucción y por último, se analiza la tasa de recuperación presente en otros artículos sobre un mismo *dataset*. En (Guangyun, et al., 2020) se realiza la reconstrucción de un ambiente, evaluando el seguimiento del recorrido con el error cuadrático medio y la reconstrucción se evalúa cualitativamente a partir de las imágenes, en (Lombard & Daalen, 2020) se aplica una rejilla para la reconstrucción, en él se evalúa el error medio de profundidad con respecto a la rejilla de reconstrucciones y sobre la reconstrucción final se produce una comparación cualitativa. En

(Abramenko, et al., 2021) se realiza un recorrido donde se capta un ambiente, se evalúa la trayectoria por el error absoluto de la posición y se corrobora de forma cualitativa la reconstrucción realizada, en (Ao, et al., s.f) se muestra un sistema de fusión sensorial, en este se realiza una comparación visual del resultado de la reconstrucción y se evalúa el error de posicionamiento.

Cuando el objetivo del sistema es la detección del objeto dinámico se evalúa la trayectoria de forma estadística, y la detección y segmentación del sistema se aborda de forma cualitativa. En (Xiao et al., 2019) se realiza el seguimiento de un objeto dinámico en una escena fuera de línea, en este trabajo se propone para la comparación diferentes métricas: el error cuadrático medio, el error medio la desviación estándar, el error relativo de dirección y el error relativo de rotación; sobre el recorrido del objeto dinámico, también se realiza una comparación completa en el marco de visual SLAM monocular, utilizando un conjunto de datos , donde la segmentación es evaluada de forma cualitativa apreciando áreas y bordes generados. En (Liu, et al., 2018) se realiza la detección fuera de línea de dos personas sobre una imagen RGB y se aplica sobre la profundidad, a partir de la detección se establece la trayectoria, esta se evalúa por el error cuadrático medio, el error medio la desviación estándar, error absoluto de trayectoria y error relativo de posición, la segmentación sobre la RGB es evaluada de forma cualitativa. En estos se muestra como las métricas para la evaluación del seguimiento son recurrentes, sin embargo el desarrollo de las métricas de evaluación de la similitud entre la reconstrucción y la escena real captada, aún se encuentra abierta a la investigación.

2.3.Marco conceptual

2.3.1. SLAM

SLAM (localización y mapeo simultaneo) es un tema de gran interés para las nuevas investigaciones. En (Nonyelu, et al., 2022) se hace una revisión de los diferentes sistemas que hacen parte de SLAM, entre ellos V-SLAM, en el cual se utilizan cámaras RGB, cámaras estéreo, cámara omnidireccional (campo visual 360°) y cámara RGB-D, para lograr la localización, mapeo y orientación de sistemas autónomos; en V-SLAM se usan sistemas monoculares, estereoscópicos y mixtos. A partir de estos, se han estructurado una serie de arquitecturas referentes para el diseño de algoritmos. Se presenta un panorama de las diferentes métricas en SLAM y sus líneas de

investigación, donde el error relativo de posicionamiento es usado para medir la precisión de la trayectoria y se plantea la necesidad del diseño de nuevas métricas para evaluar con mayor precisión el resultado de los algoritmos de reconstrucción. En (Li, et al., 2022) tras una revisión, concluye que la determinación del posicionamiento y la reconstrucción de ambientes es fundamental dentro del desarrollo de SLAM en vehículos autónomos robóticos.

2.3.2. Reconstrucción de ambientes estáticos en SLAM

El desarrollo de sensores de percepción del entorno ha abierto la posibilidad de generar la reconstrucción de escenas a través de los sistemas de visión artificial y V-SLAM (Markham et al., 2018; LI, et al., 2019). Esta es una línea que ha llegado a diferentes fronteras al utilizar las nuevas tecnologías emergentes para la reconstrucción de escenas, esto ha impulsado diferentes desarrollos.

V-SLAM y la aplicación de redes neuronales para la reconstrucción de escenas. En (Chen, et al., 2021) se muestra la extracción de profundidad monocular para la reconstrucción de escenas, donde se procesan imágenes RGB de una sola cámara y por la aplicación de *Detail-preserving network (DPNet)* se extraen los bordes y con él, las superficies que conforman los objetos dentro de una escena interior estática, esta analiza y asigna una profundidad a partir de los vecinos más cercanos de cada píxel. En este se evalúa el resultado de la reconstrucción de *point cloud (PCL)* de forma visual y se observa que aún se generan inconsistencias en la nube de puntos. En (Bousseau, et al., 2019) se aplica una combinación de predicción por redes neuronales convolucionales aplicadas sobre el resultado de la reducción de nubes de puntos con *voxel grid* y la generación de las normales para bocetos 3D de múltiples vistas, en esta se generan reconstrucción de objetos 3D, al realizar una comparación cualitativa se observa cómo las superficies de los objetos dinámicos están evidentemente más suaves que la original. Estos desarrollos donde se ven inmersos sistemas de inteligencia artificial (Gallant & Marshall, 2016; Quan & MA, 2020) se caracterizan por tener un alto costo computacional.

V-SLAM y la utilización de sistemas de extracción de características. En (Cao, et al., 2021) se presenta el sistema de red estéreo de vista múltiple, basada en la extracción de volumen piramidal para reconstrucción 3D, en él se realiza una extracción de profundidad por medio de la convolución de múltiples vistas de una escena, en este proceso se utiliza la extracción de características y su posición dentro de una serie de imágenes para determinar la profundidad de cada punto, esta es

aplicada sobre escenarios exteriores de gran complejidad. Debido al uso de sistemas matemáticos complejos, esta presenta una ejecución fuera de línea sobre un set de datos.

V-SLAM y la aplicación de algoritmos matemáticos y estadísticos para la reconstrucción de escenas. En (Akanshu, et al., 2018) se propone un método de reconstrucción de ambientes interiores usando las nubes captadas por una cámara RGB-D, en este se captan *frame* los cuales son unidos sucesivamente y procesados por medio del algoritmo *Ransac* e *Iterative Closest Point (ICP)*, estos realizan una corrección del posicionamiento de cada PCL obtenida para así concatenarlas y generar un buen cierre de bucle, reconstruyendo así las escenas interiores. Este se caracteriza por ser de un alto costo computacional.

V-SLAM y la reconstrucción de escenas en línea mientras se recorre la escena. En (Hernández & Ortiz, 2020) se utiliza un método de reconstrucción de escenas estáticas con multi robot, en él, se tiene varios robots que van aportando segmentos del mapa captado, estos son fusionados en tiempo real, generando un mapa del entorno en 2D, sobre el cual se realiza una proyección 3D. Este aplica el algoritmo *Ransac* para el cierre de bucle y es de bajo costo computacional. En (Cýrus, et al., 2020) se presenta un sistema híbrido de captura y reconstrucción de escenas, en este se utilizan los *HoloLens* para captar el entorno mientras se realiza un recorrido, posteriormente, fuera de línea, se aplica un sistema de concatenación de puntos y unido a un filtro de Kalman, logran tener un buen cierre de bucle, este se realiza sobre escenas interiores estáticas.

V-SLAM para la reconstrucción de escenas estáticas por fusión sensorial mientras se realiza un recorrido en la escena. En (Griepentrog, et al., 2018) se propone un método de reconstrucción donde se fusionan los datos, un LIDAR 2D y RGB-D, para lograr mejorar la detección de características del entorno; sin embargo, solo se presenta en tiempo real la reconstrucción generada por el LIDAR 2D, dado su bajo costo computacional de procesamiento, en este se capta una escena exterior (cultivo de maíz) pero con medición a distancia limitada, en él se realiza una comparación visual de la calidad de la escena y se establecen los porcentajes de error en el posicionamiento de los elementos dentro de la escena.

En V-SLAM existe una gran serie de posibilidades de interacción con otras tecnologías, permitiendo que este sea una línea rica en desarrollos, sin embargo, aún se tiene diferentes retos por superar, como el desarrollo de métodos de odometría de alta precisión.

2.3.3. Odometría en SLAM

En SLAM para obtener el posicionamiento, se usan instrumentos como GPS, IMU o INS/GNSS (Pierzchała, et al., 2018; Chiang, et al., 2019), u odometría visual (Bonnin, et al., 2022). En (Gaspar, et al., 2018) se presenta un comparativo de las técnicas para la determinación del posicionamiento y se observa, dependiendo de las condiciones de los recorridos, se logra una mejor precisión. Al realizar una comparación visual, el sistema ORB-SLAM es el que mejor logra obtener la ruta en entornos urbanos. Dentro de la odometría visual se han presentado múltiples algoritmos para disminuir el error.

Odometría visual en SLAM por visión estereoscópica. En (Behnke, et al., 2018) se presenta un sistema de odometría visual, basada en visión estéreo semidenso en tiempo real, en esta se extrae el posicionamiento y es de alto costo computacional; sin embargo, es ejecutado en línea. Las pruebas son desarrolladas en una escena exterior, en un recorrido a bordo de un vehículo, en él se realiza una comparación del recorrido con el real, en el cual se observa cómo el sistema presenta un error cuadrático medio de promedio 0.80m.

Odometría visual en SLAM por sistemas de profundidad. En (Bonnin, et al., 2022) se presenta un nuevo enfoque de odometría y mapeo con LIDAR para la estimación de poses y la construcción de mapas, basado en la minimización de una función de pérdida derivada de un conjunto de correspondencias ponderadas de punto a línea con un mapa local, en esta se presenta una técnica en donde se disminuye el error de posicionamiento hasta el 1% de la medición, y se realizan pruebas fuera de línea en un conjunto de datos en ambientes exteriores.

SLAM y fusión sensorial para odometría. En (Chang, et al., 2020) se plantea que en los sistemas robóticos se tiene dificultad en la precisión del posicionamiento, se propone un método por fusión de datos donde se aplican redes neuronales para disminuir el ruido, en esta se realizan pruebas en un entorno real y se muestra como la fusión ORB-SLAM e IMU es el que proporciona una trayectoria más fiel a la trayectoria propuesta en el recorrido. En (Chen, et al., 2018) se tiene un vehículo que recorre un ambiente, este porta tres LIDAR 2D, con los cuales va capturando el entorno y crea un mapa 3D, este utiliza un sistema de localización por medio de un cuarto LIDAR en 2D y aplica la fusión sensorial, en este se procesan nubes de bajo peso computacional, permitiendo determinar el posicionamiento en cada momento.

Odometría visual en SLAM en entornos dinámicos. En (Song, et al., 2018) se presenta un método de odometría visual para entornos dinámicos, en este se utiliza una cámara RGB-D con la cual se detectan los bordes de los objetos en la escena, y se aplica un sistema de detección de objetos dinámicos para evitar la incorporación de puntos característicos de la escena sobre el objeto dinámico, este disminuye el error de posicionamiento en un 50% aproximadamente en comparación con ORB-SLAM, en entornos de alta dinámica.

2.3.4. SLAM en ambientes dinámicos

Cuando los sistemas autónomos están inmersos en ambientes dinámicos (SLAM dinámico), el grado de dificultad aumenta, dado que los objetos dinámicos al realizar un recorrido están en movimiento constante, situándolos en diferentes posiciones cada vez que se genere un registro por parte de los sensores. En entornos reales se tendrán variables como la velocidad del objeto, dirección y distancia a los sensores, los cuales son factores de gran importancia para la detección del objeto dinámico.

SLAM dinámico es una línea en la cual ha sido difícil generar aportes por la complejidad que implica. En (Chacon, et al., 2019) se realiza una revisión de 119 algoritmos de detección de objetos dinámicos sobre una secuencia de imágenes 2D, en él, se evalúan los sistemas por la capacidad de detección, adaptabilidad, novedad y rapidez, también se identifica que esta área de estudio aún tiene muchas limitaciones que deben ser exploradas para así dar solución. En (Sahin, et al., 2020) se plantean dos categorías de tipo de detección: detección por cajas delimitadoras en 2D o 3D y estimadores de postura de 6D, en estos se aplican clasificadores a partir de sistemas de inteligencia artificial.

SLAM dinámico y detección por redes neuronales. En (Hua, et al., 2020) se realiza la detección de una persona por aplicación de redes neuronales convolucionales, se caracteriza por ser de bajo costo computacional con una ejecución aproximadamente de 90ms sobre imágenes RGB y es probado fuera de línea sobre un set de datos. En (Abad, et al., 2018) se aplica un método de detección de objetos por bolsa de palabras sobre nubes detectadas por un LIDAR 3D, en este se realizan pruebas donde se varía la densidad de puntos en las nubes, para verificar las detecciones, entre más denso más fácil lo detecta, pero más alto es el costo computacional, este es aplicado sobre unos data set, los cuales contienen diferentes figuras que son clasificadas fuera de línea.

SLAM dinámico y detección de movimiento. En (Wang, et al., 2020) se realiza la detección fuera de línea de objetos en movimiento por obtención de cubos de ocupación sobre la nube de puntos, en este se usa una nube de puntos reducida, la velocidad del objeto dinámico es un factor crítico y se plantea que aun si se lograra una ejecución en línea, se tendría limitación en la velocidad de detección del objeto dinámico, dado que el LIDAR 3D utilizado en este experimento, tiene una tasa de muestreo de 10Hz. En este se utilizan ambientes exteriores y se realiza la evaluación a partir de la precisión del sistema de clasificación y de forma cualitativa sobre la segmentación del objeto dinámico. En (Cordeiro & Pedrino, 2019) se realiza la detección y seguimiento sobre un set de datos en el cual una persona realiza un recorrido a una velocidad aproximada de 3,125m/s, en un ambiente interior, en este se utilizan técnicas de visión aplicadas sobre imágenes de profundidad y RGB. En (Wu, et al., 2019) se presenta el método *onboard diagnostics* (OBD) para detectar y estimar la trayectoria de objetos dinámicos con un LIDAR 3D con ejecución a 2 Hz. En este se utilizan nubes de puntos reducidas de entornos dinámicos exteriores con múltiples objetos dinámicos. La evaluación de la detección se realiza por el porcentaje de aciertos en la detección del objeto dinámico alcanzando valores promedio del 95.9%.

Dado la gran complejidad de este proceso, son pocos los métodos propuestos que han logrado detectar y eliminar objetos dinámicos y reconstruir la escena. En (Fan, et al., 2018) se propone un método para la eliminación de objetos dinámicos por fusión de imágenes, aplicado fuera de línea sobre set de datos, en estos, se enmarca la necesidad de nuevos desarrollos para la detección de objetos dinámicos, comenzando por la detección a bajas velocidades y con objetos de grandes dimensiones, como un siguiente paso para el desarrollo de esta línea de investigación, en este se realiza la detección del objeto dinámico y se elimina del *frame*, posteriormente se concatena la PCL para recuperar el fragmento de la escena que se encontraba ocluida y así, obtener la reconstrucción.

SLAM-dinámico es un área donde aún se tiene muchos aspectos por explorar he investigar, dado a su necesidad de ejecución en línea y bajo peso computacional.

2.3.5. Métricas

SLAM está compuesto por diversos sistemas y métodos, los cuales han aportado desde diferentes áreas al desarrollo de este. En este proceso se han establecido métricas como el error cuadrático medio, el error absoluto y la desviación estándar (Androulakis, et al., 2021) y así evaluar el comportamiento de los sistemas de seguimiento de la ruta de los instrumentos y del recorrido de los objetos dinámicos en la escena (Nonyelu, et al., 2022), estos han sido aplicados ampliamente en SLAM.

Métricas en SLAM para sistemas en 2D. En (Haoran, et al., 2022) se presenta un método para el mapeo de escenas de bajo costo computacional en 2D, el error se evalúa a través de la superposición del mapa de objetos en la escena real, esto permite realizar una comparativa visual y se establecen los porcentajes de error en la medida de los fragmentos de la escena, este es aplicado en una escena interior estática. En (Hol´y, 2016) se realiza una comparativa entre un sistema propuesto y otros algoritmos, en él, se evalúa el promedio del error absoluto de posicionamiento en un escenario interior, el cual es un 70% menor que el promedio de los algoritmos evaluados.

Métricas en SLAM para sistemas 3D de reconstrucción. En (Chen, et al., 2018) se presenta un sistema de extracción del posicionamiento, en él, se evalúa de forma cualitativa las características del software a probar. Posteriormente se realizan pruebas en diferentes ambientes, interiores y exteriores, y se determina el resultado de precisión de registro entre cada escaneo estático con el método propuesto y el resultado de precisión de registro entre cada escaneo estático, con función y coincidencia plana, estos son medidos en metros y grado según la proyección. A partir de los resultados de los recorridos, se presentan imágenes de diferentes perspectivas de la reconstrucción, lo cual permite establecer una comparativa cualitativa. En (Hongmin, et al., 2020) se presenta un sistema de reconstrucción donde se observa el crecimiento de la nube de puntos como métrica de crecimiento de la reconstrucción, posteriormente se hace una comparación cualitativa de la reconstrucción obtenida y, por último, se realiza un análisis de la tasa de recuperación que se ha presentado en otros artículos sobre un mismo set de datos. En (Guangyun, et al., 2020) se realiza la reconstrucción de un ambiente interior estático, para ello se realiza el seguimiento del recorrido en el cual es evaluado a partir del error cuadrático medio sobre la ruta establecida, luego se evalúa la reconstrucción de forma cualitativa a partir de imágenes renderizadas de la PCL. En (Lombard & Daalen, 2020) se presenta una técnica para la reconstrucción, aplicando rejilla, se evalúa el error

absoluto medio de profundidad con respecto a la rejilla, la reconstrucción final es evaluada de forma cualitativa.

Métricas en SLAM para sistemas 3D de reconstrucción con objetos dinámicos. En (Cordeiro & Pedrino, 2019) se realiza la detección de una persona mientras realiza un recorrido, en esta se evalúa el consumo de recurso computacional a través del porcentaje de uso y la detección del objeto dinámico se realiza de forma cualitativa. En (Liu, et al., 2018) se realiza la detección de dos personas sobre una imagen RGB, la cual es aplicada sobre el mapa de profundidad, a partir de la detección se establece la trayectoria del objeto dinámico, esta se evalúa por el error absoluto, el error cuadrático medio y la desviación estándar, para evaluar los puntos suministrados del posicionamiento se usa el error absoluto de trayectoria y error relativo de posicionamiento. La segmentación se evalúa sobre la RGB y se compara de forma cualitativa. En (Xiao et al., 2019) se realiza el seguimiento de un objeto dinámico en una escena, en este se propone evaluar el error de posicionamiento absoluto y el error de posicionamiento relativo de las trayectorias establecidas en fotogramas clave, para evaluar la precisión de localización de cada marco y de esta forma, evaluar la trayectoria. Se utiliza el error cuadrático medio, el error medio, el error mediano y la desviación estándar, y para evaluar la detección el número de puntos característicos dinámicos y estáticos a lo largo del proceso en una escena con objetos dinámicos, se realiza una comparación completa del marco SLAM visual monocular.

2.4.Conclusiones

Se observa que en SLAM tiene como factor común, el costo computacional asociado a la ejecución de los algoritmos de procesamiento de los datos, lo cual debe ser limitado para permitir la ejecución en línea.

Se encontró que los sistemas de reconstrucción han tenido gran éxito en la reconstrucción de ambientes estáticos; sin embargo, estos no son aplicables de forma directa sobre escenas con presencia de objetos dinámicos, dado a la oclusión que este genera.

Se encontró que las métricas existentes para evaluar el comportamiento de los algoritmos son insuficientes dado que no se tiene un valor que relacione la similitud entre la escena reconstruida y la escena de referencia.

Se observa que no se encuentra un método que permita generar una reconstrucción 3D de escenas con presencia de objetos dinámicos, de bajo costo computacional.

Capítulo 3. Metodología de la investigación

3.1. Diseño del experimento

Para llevar a cabo esta investigación se diseñaron una serie de experimentos, que parten desde la pregunta de investigación y van acorde al desarrollo de una metodología de investigación que busca alcanzar cada uno de los objetivos propuestos, (ver Figura 1).

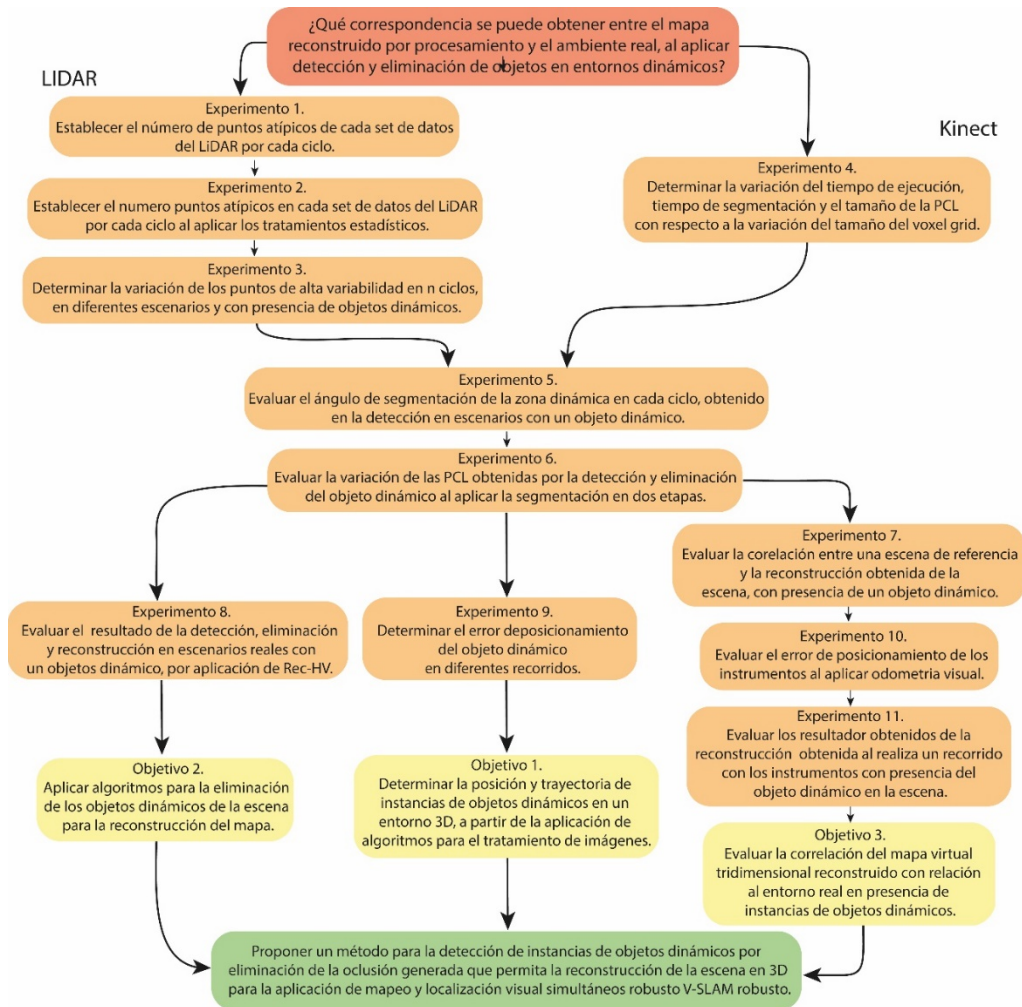


Figura 1. Secuencia de desarrollo experimental en relación con los objetivos propuestos en esta investigación.

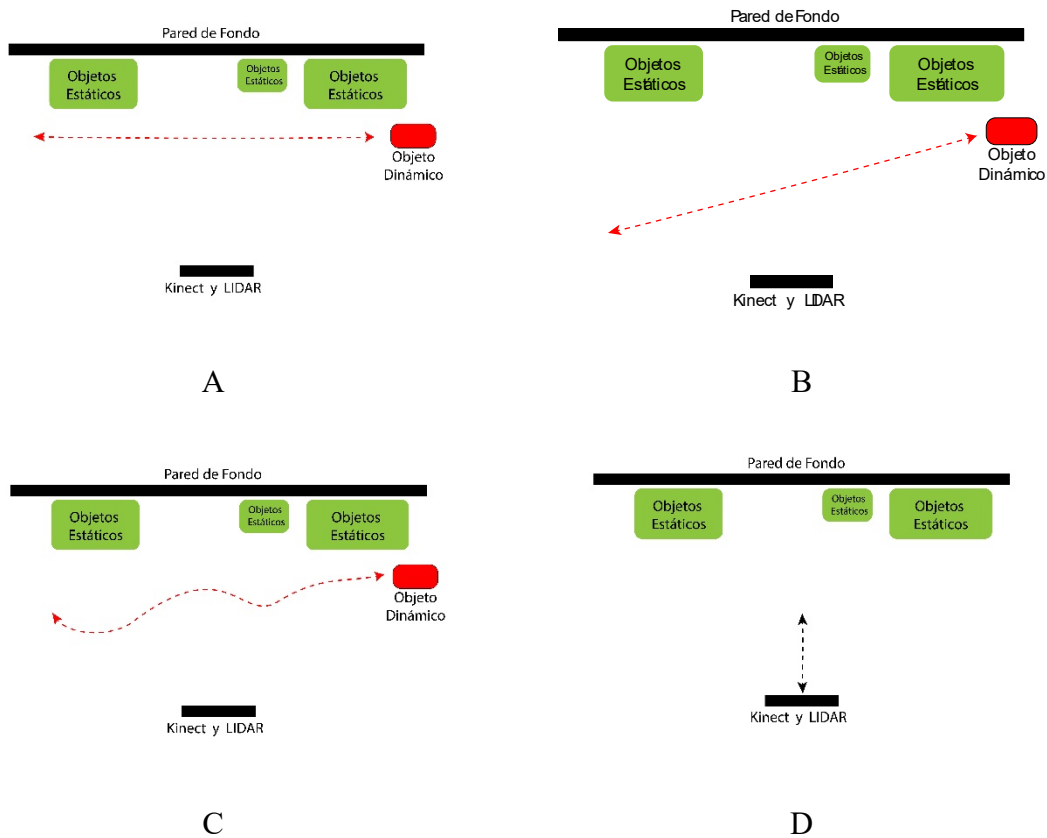
Para verificar el comportamiento de cada uno de los elementos desarrollados en este proceso de investigación, se han diseñado una serie de experimentos, los cuales tendrán una variedad de objetivos, ver *Tabla 1*.

Tabla 1 Objetivos de experimentos.

| Experimento | Objetivo |
|-------------|--|
| 1 | Establecer el numero puntos atípicos en cada set de datos del LiDAR por cada ciclo. |
| 2 | Establecer el numero puntos atípicos en cada set de datos del LiDAR por cada ciclo al aplicar los tratamientos estadísticos |
| 3 | Determinar la variación de los puntos de alta variabilidad en n ciclos, en diferentes escenarios y con presencia de objetos dinámicos. |
| 4 | Determinar la variación del tiempo de ejecución, tiempo de segmentación y el tamaño de la PCL con respecto a la variación del tamaño del <i>Voxel Grid</i> . |
| 5 | Evaluar el ángulo de segmentación de la zona dinámica en cada ciclo, obtenido en la detección, en escenarios con un objeto dinámico. |
| 6 | Evaluar la variación de las PCL obtenidas por la detección y eliminación del objeto dinámico al aplicar la segmentación en dos etapas |
| 7 | Evaluar la correlación entre una escena de referencia y la reconstrucción obtenida de la escena, con presencia de un objeto dinámico. |
| 8 | Evaluar el resultado de la detección, eliminación y reconstrucción en escenarios reales con un objeto dinámico, por aplicación de Rec-HV. |
| 9 | Determinar el error de posicionamiento del objeto dinámico en diferentes recorridos. |

| | |
|----|--|
| 10 | Evaluar el error de posicionamiento de los instrumentos al aplicar odometría visual. |
| 11 | Evaluar los resultados de la reconstrucción obtenida al realizar un recorrido con los instrumentos con presencia del objeto dinámico en la escena. |

Hay factores y requerimientos que se presentan en el respectivo diseño experimental. Dada la recurrencia, se diseñaron una serie de recorridos para los experimentos, (ver Figura 2).



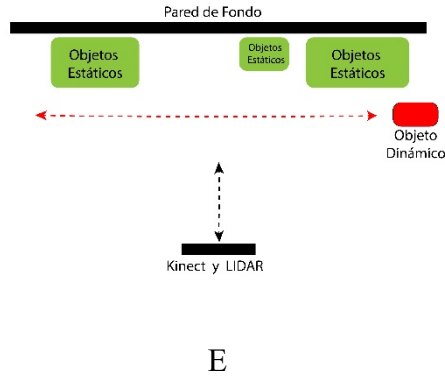


Figura 2. Recorridos seleccionados para los experimentos.

3.1.1. Experimento 1

En este experimento se tendrá como entrada una escena estática interior y se obtendrá de salida el vector de radios del LiDAR (r) el cual son un corte transversal de la escena, adicional se tendrá una serie de factores que influyen en él, (ver Figura 3).

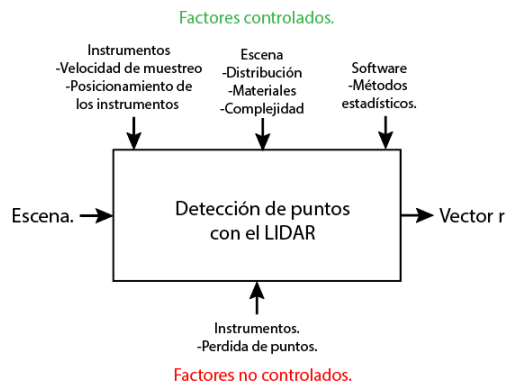


Figura 3. Factores controlados y no controlados experimento 1.

En este experimento se realizarán 2 pruebas en 2 escenarios que tendrán un complejidad diferente, en cada prueba se registrarán 10 ciclos, se realizará una tercera prueba constituida de 500 ciclos sobre uno de los escenarios.

3.1.2. Experimento 2

Para este experimento, se tendrá como entrada el vector r obtenido del experimento 1, sobre el cual se aplicaran una serie de procesos estadísticos y se tendrá como salida el vector r procesado. Este posee una serie de factores que influyen sobre el experimento (ver Figura 4).

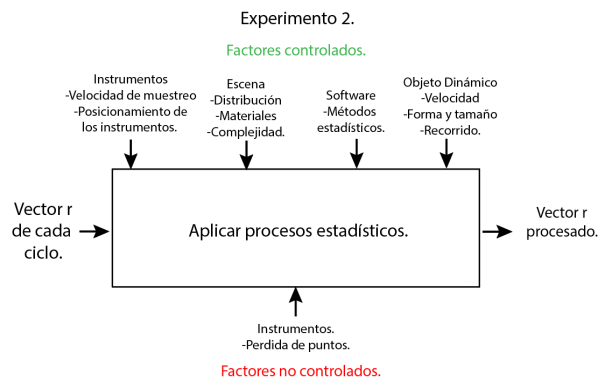


Figura 4. Factores controlados y no controlados experimento 2

Este experimento está constituido por 8 pruebas, se tendrá una primera prueba constituida de 500 ciclos en donde se aplicará el proceso de interpolación y extrapolación en cada ciclo. Se tendrán 6 pruebas constituidas de 25 ciclos cada una, estas cuentan con diferentes tamaños para la ventana del promedio móvil (n), en estas el objeto dinámico estará en la escena para cada prueba y realizará un recorrido prediseñado, ver Figura 2[A]. La última prueba se realizará sobre un escenario de mayor complejidad, esta estará constituida por 25 ciclos y en ella se aplicarán todos los procesos estadísticos.

3.1.3. Experimento 3

En este experimento se tendrá como entrada el vector r procesado en cada ciclo, sobre él, se aplicarán el promedio móvil con n igual a tres y se tendrá como salida la variación de la desviación estándar para cada punto en cada ciclo. Se tendrán una serie de factores que influyen sobre el experimento, (ver Figura 5)

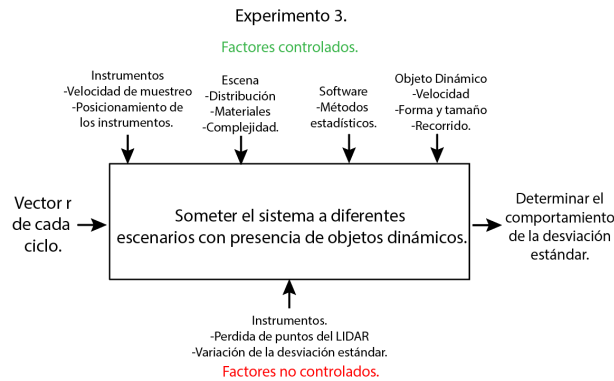


Figura 5. Factores controlados y no controlados experimento 3.

Este experimento estará compuesto de 4 pruebas de 25 ciclos en 2 escenarios, en cada escenario se tendrá una prueba sin objeto dinámico y otra con objeto dinámico, el objeto dinámico realizará un recorrido establecido, ver Figura 2[A].

3.1.4. Experimento 4

Para este experimento se tendrá como entrada los datos del RGB-D y como salida los tiempos de ejecución y segmentación del sistema. Contará una serie de factores que influyen en este experimento, (ver Figura 6).

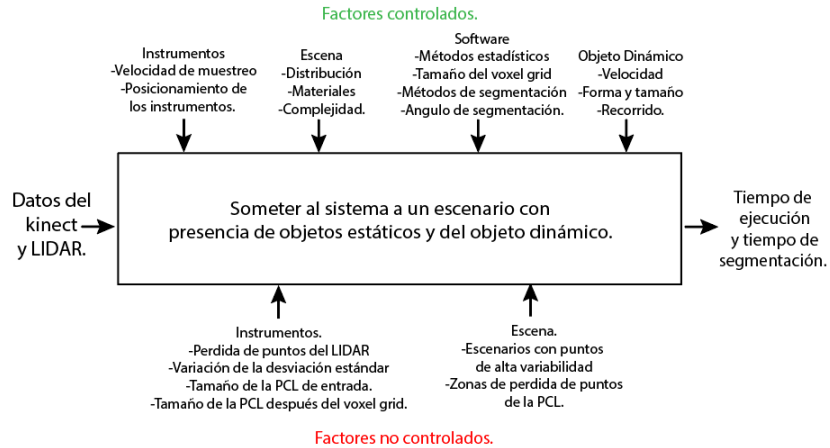


Figura 6. Factores controlados y no controlados experimento 5.

En este experimento se realizarán 13 pruebas, las cuales estarán constituidas por 100 ciclos, en la primera prueba no se aplicará el *voxel grid* con el fin de tener una referencia en tiempo de ejecución, en las siguientes 12 pruebas se tendrá un valor diferente para el *voxel grid* para cada prueba. En las pruebas no se tendrá presencia del objeto dinámico, se conservará la posición de los sensores, se utilizará la misma escena y se definirá constante los ángulos de segmentación, para así mantener los ciclos bajo las mismas características.

3.1.5. Experimento 5

Se tendrá como entrada el vector r procesado de cada ciclo y como salida estará el ángulo de detección. Se tendrán diferentes factores externos que influirán sobre el experimento, (ver Figura 7).

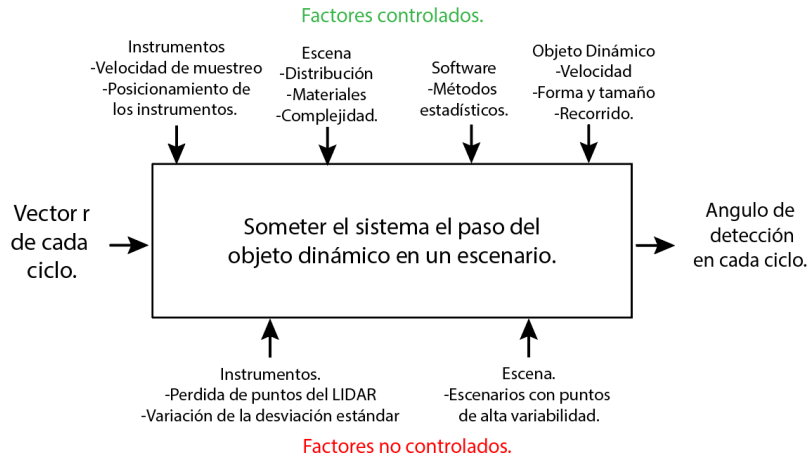


Figura 7. Factores controlados y no controlados experimento 4.

En este experimento se realizarán 2 pruebas de 25 ciclos, estas se realizarán sobre la misma escena y se tendrá presencia del objeto dinámico cruzando la escena con un recorrido establecido, ver Figura 2[A].

3.1.6. Experimento 6

Para este experimento se tendrá como entrada los datos del LIDAR y de la RGB-D y como salida se tendrán las PCL de la zona estática y la PCL de la zona dinámica, en este experimento el sistema podrá realizar la selección del ángulo de segmentación de forma automática, basado en las zonas de alta variabilidad que el sistema encuentre, esto incrementará los factores que estarán no controlados. Se tendrán una serie de factores que influyen en él, (ver Figura 8).

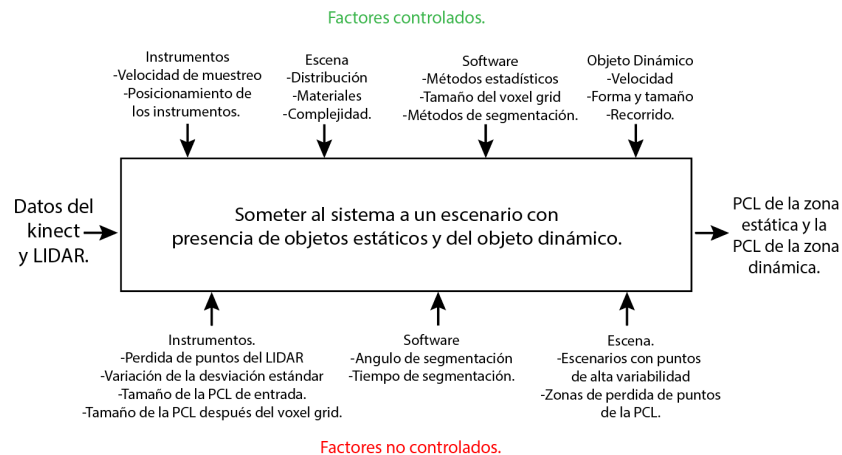


Figura 8. Factores controlados y no controlados experimento 6.

Este experimento estará constituido por pruebas de 24 ciclos, donde se tendrá un escenario estático con objetos estáticos y la presencia el objeto dinámico el cual realiza un recorrido establecido, ver Figura 2[A].

3.1.7. Experimento 7

En este experimento se tendrá como entrada los datos de la RGB-D y el LIDAR y como salida el porcentaje de correlación entre la PCL reconstruida y la PCL de referencia. Se tendrán una serie de factores que influirán sobre el experimento, (ver Figura 9).



Figura 9. Factores controlados y no controlados experimento 7.

Para este experimento se realizarán 10 pruebas, las cuales tendrán 25 ciclos cada una, se captará una escena de referencia, la cual será comparada con la escena obtenida en cada uno de los ciclos de cada prueba, para cada una, el objeto dinámico realizará un recorrido establecido, ver Figura 2[A], se aplicarán dos etapas de software, la primera permitirá obtener en línea las PCL

reconstruidas y en la segunda etapa fuera de línea se aplicará un método para determinar la correlación de la escena estática reconstruida al compararla con la referencia.

3.1.8. Experimento 8

En este experimento se tendrán como entrada los datos de la RGB-D y del LIDAR de diversos escenarios reales, los cuales presentarán diferentes características y se tendrán como salida la reconstrucción de la escena estática y tablas de parámetros resultantes de la ejecución. Al estar en entornos reales con objeto dinámicos reales, se tendrán más variables no controladas, (ver Figura 10).

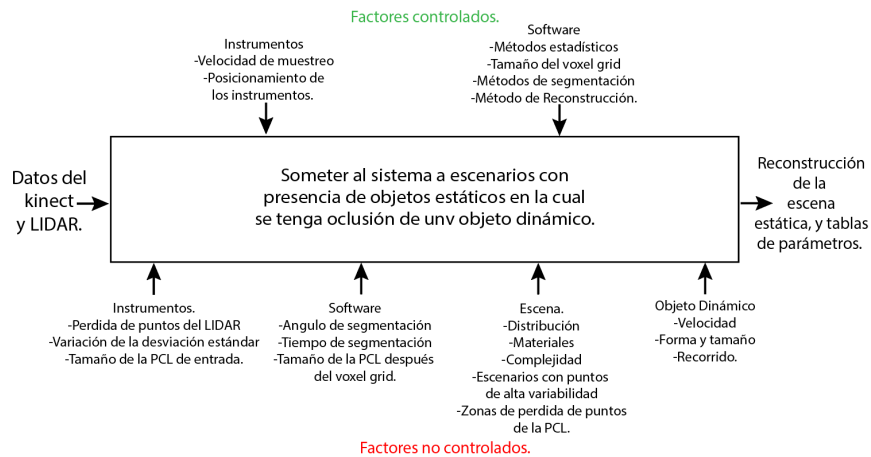


Figura 10. Factores controlados y no controlados experimento 8.

Este experimento estará compuesto por 5 pruebas de 25 ciclos cada una y se tendrá un escenario diferente para cada prueba, los escenarios serán escogidos entre entornos cotidianos interiores, en estos entornos se tendrá objetos estáticos con diversos materiales y formas y un objeto dinámico sobre el cual no se tendrá un control total de la velocidad y ni de su forma, el recorrido no será lineal y tendrá cambios en la profundidad; sin embargo, no incluirá paradas ni cambio en la dirección, ver Figura 2[C] de cada prueba se obtendrá la PCL reconstruida y parámetros como tiempo promedio de ejecución, número de puntos por ciclo, reducción promedio de la PCL, ciclos

con puntos de alta variabilidad, porcentaje de la PCL obtenida en la segmentación por ángulo, porcentaje de la PCL obtenida en la segmentación por agrupación por profundidad, PCL inicial (PCL obtenida en el primer ciclo), PCL_{kT} (PCL k-esima obtenida por la concatenación) y porcentaje de incremento de puntos en la PCL de reconstrucción.

3.1.9. Experimento 9

Para este experimento se tendrá como entrada los datos de la RGB-D y del LIDAR, y como salida, el posicionamiento del objeto dinámico, en este se tendrán una serie de factores que influirán en el experimento (ver Figura 11).

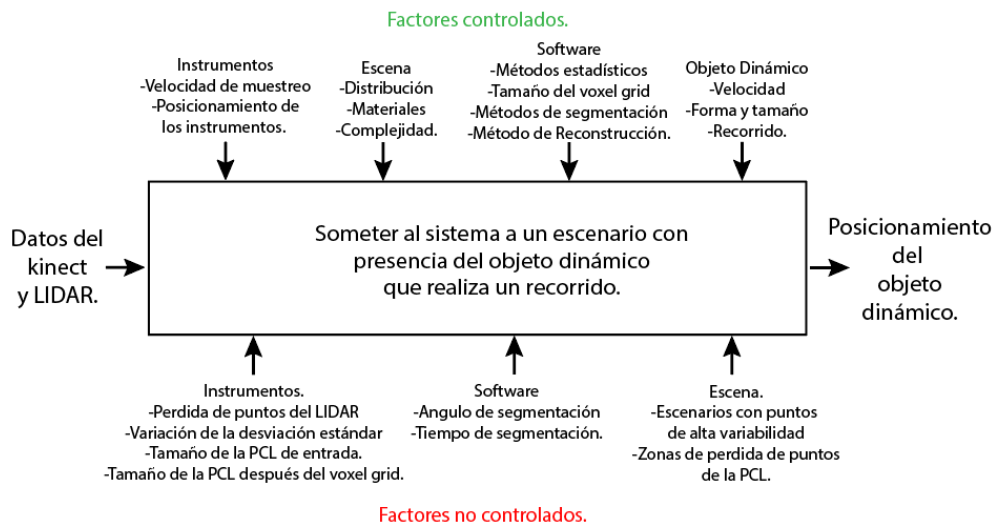


Figura 11. Factores controlados y no controlados experimento 9.

En este experimento se realizarán 15 pruebas de 25 ciclos cada una, en un mismo escenario con objetos estático y con la presencia del objeto dinámico, el cual realizará 3 recorridos, se tendrá un primer recorrido en sentido izquierda derecha, en la cual se realizará 5 pruebas, un segundo recorrido con sentido de derecha a izquierda donde se realizará 5 pruebas, (ver Figura 2[A]). Se tendrá un tercer recorrido, el cual se realizará de izquierda a derecha con variación de la profundidad, (ver Figura 2[B]).

3.1.10. Experimento 10

Para este experimento se tuvo como entrada los datos de la RGB-D y los datos del sensor de desplazamiento lineal. Se tendrán una serie de factores que influirán en el proceso, (ver Figura 12).

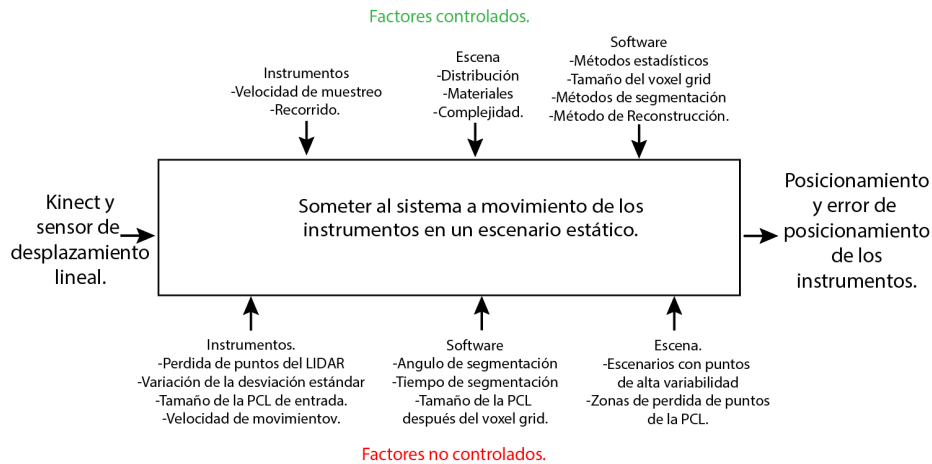


Figura 12. Factores controlados y no controlados experimento 10.

En este experimento se realizará una prueba donde los instrumentos ejecutarán un recorrido lineal sobre un sistema guía que solo permitirá variar la posición sobre el eje x, en él, se moverán primero hacia adelante y posteriormente será regresado a su posición de origen, no se tendrá presencia del objeto dinámico, (ver Figura 2[D]). En este se realizará una comparación entre la variación de la posición reportada por el sistema de odometría mecánica y el sistema de odometría visual, los ciclos se determinarán a partir del tiempo que dure el recorrido.

3.1.11. Experimento 11

Para este experimento se tendrá como entrada los datos de la RGB-D y del LIDAR y se tendrá como salida la reconstrucción de la escena estática, la correlación y la variación del número de

puntos, en este experimento se utilizará una escena interior real, con un objeto dinámico real, lo que incrementará los factores no controlados que afectaran el sistema (ver Figura 13).

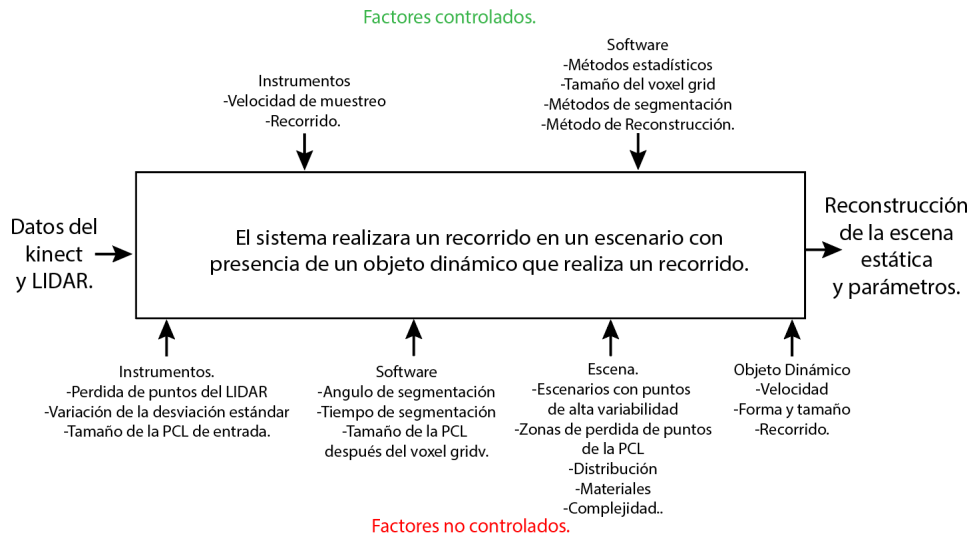


Figura 13. Factores controlados y no controlados experimento 11.

Para este experimento se realizará una prueba donde el sistema realizará un recorrido en un escenario real interior con materiales diversos y con distribución aleatoria, en donde se presentará un objeto dinámico real que cruzará la escena sin que se tenga control total de su velocidad, ni de su forma, ni un control total de su recorrido, en este caso los recorridos de los instrumentos y del objeto dinámico serán de forma perpendicular, ver (Figura 2[E]), el ciclo total y tiempo total será definido por el recorrido realizado.

3.2. Modelo Matemático

El método propuesto de Reconstrucción dinámica basada en segmentación de zonas de puntos de alta variabilidad (Rec-HV), utiliza la desviación estándar de los datos del LIDAR 2D de cada ciclo de ejecución, para determinar el ángulo de detección de la zona dinámica y aplicarlo en la segmentación del *PCL* del RGB-D. Este permite eliminar los puntos del objeto dinámico en cada ciclo, para así reconstruir la escena estática por concatenación de *PCL* sucesivas. Para llevar a cabo este proceso, se realizaron diferentes procesos estadísticos y matemáticos, sobre los datos del LiDAR y del RGB-D.

El procesamiento matemático propuesto para el método Rec-HV implica realizar los siguientes pasos, (ver Figura 14).

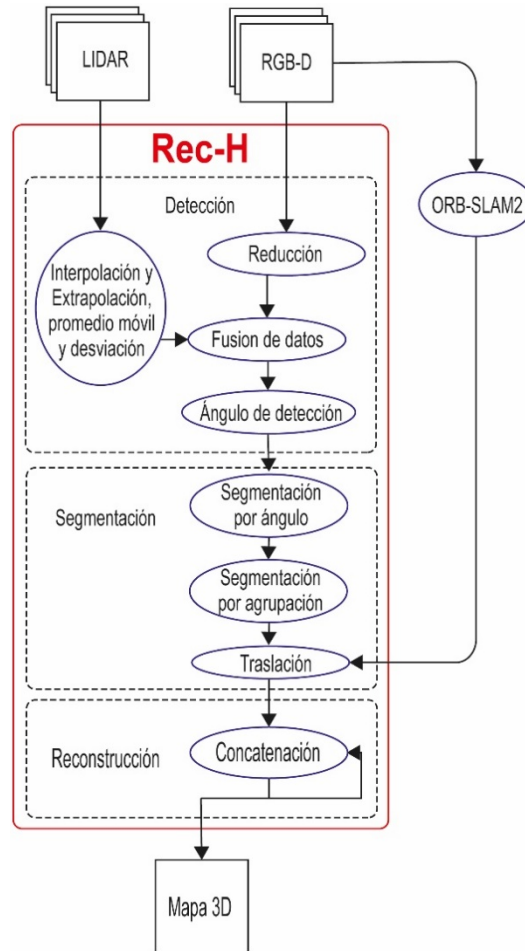


Figura 14. Diagrama del procedimiento matemático del método Rec-HV.

A continuación, se explica en mayor detalle cada uno de los pasos.

3.2.1. Preprocesamiento

Los datos del LIDAR se reciben en cada ciclo de medición. Estos están constituidos por el vector de la profundidad (r) de número de puntos (l), el ángulo de inicio de muestreo (θ_{min}) y el ángulo

de incremento ($\Delta\theta$). A partir de estos datos se construye un vector de ángulos (θ) de longitud l , ver ecuación (1).

$$\begin{aligned} \mathbf{r} &= [r_1, r_2, r_3 \dots r_l] \\ \theta_i &= \theta_{\min} + \Delta\theta * i \\ \boldsymbol{\theta} &= [\theta_1, \theta_2, \theta_3 \dots \theta_l] \end{aligned} \tag{1}$$

Los datos del LIDAR están constituidos por una medición de 360° ; sin embargo, en este estudio solo se analizará una apertura de 90° , lo que establece a l igual a 90, a partir de este se construye el vector de ángulos $\boldsymbol{\theta}$ y se recorta el de vector profundidad \mathbf{r} .

3.2.1.1. Interpolación y Extrapolación

El vector \mathbf{r} presenta algunos valores de profundidad infinito y estos deben ser sustituidos por valores reales, esto se realiza aplicando la interpolación y extrapolación, dependiendo de la posición en el vector \mathbf{r} . La interpolación se aplica en los valores que se encuentran en medio de dos valores reales, no necesariamente consecutivos, y la extrapolación sustituye los datos que presentan valor infinito, que se encuentran en los puntos extremos (inicio y final) del vector \mathbf{r} . Estos realizan la sustitución de los valores del vector al encontrar una línea de tendencia con sus vecinos más cercanos, esta es proyectada sobre los puntos con valores atípicos. Este proceso arroja un set completo de datos del LiDAR, constituido por parejas ordenadas dadas por la posición dentro del vector $\boldsymbol{\theta}$ y \mathbf{r} .

3.2.1.2. Promedio móvil y desviación

El promedio móvil es aplicado con una ventana de n ciclos sobre cada punto del vector de profundidad \mathbf{r} del ciclo actual con los $n - 1$ ciclos anteriores inmediatos al actual, ecuación (2). Sobre estos mismos datos se determina un vector de desviación estándar ($d\mathbf{r}$), constituido por la $d\mathbf{r}$ de cada punto en los n ciclos, ecuación (3).

$$\bar{r}_k = \frac{1}{n} \sum_{k-n+1}^k r_{k-n+1} \quad (2)$$

$$dr_k = \sqrt{\frac{1}{n} \sum_{k-n+1}^k (r_{k-n+1} - \bar{r}_k)^2} \quad (3)$$

Donde r_k es el vector de profundidad para el k -ésimo ciclo de lectura del LiDAR y \bar{r}_k es su promedio, así mismo dr es la desviación estándar para cada r_k del vector de profundidad del k -ésimo ciclo de lectura.

En cada k -ésimo ciclo de datos donde se captaron datos del LiDAR, el tratamiento estadístico proporciona tres vectores de datos: vector promedio de distancias \bar{r}_k , vector de ángulos θ_k y vector de desviación estándar dr_k ; del k -ésimo ciclo, ver ecuación. (4).

$$\begin{aligned} \bar{r}_k &= [\bar{r}_1, \bar{r}_2, \bar{r}_3 \dots r_l] \\ \theta_k &= [\theta_1, \theta_2, \theta_3 \dots \theta_l] \\ dr_k &= [dr_1, dr_2, dr_3 \dots dr_l] \end{aligned} \quad (4)$$

3.2.2. Ángulo de detección

A partir de \bar{r}_k , θ_k y dr_k en cada ciclo se determinan los máximos de variabilidad. Estos están dados por \bar{r}_k cuyo valor de dr sean máximos (dr_{max1} , dr_{max2}) y no consecutivos, estos corresponden a los extremos de la zona de altavariabilidad y tienen una posición (i_1 y i_2) en el vector dr , con la cual se obtienen los ángulos de detección (α_1 y α_2), ver ecuación (5).

$$\alpha_1 = \theta_{\min} + i_1 * \Delta\theta \quad (5)$$

$$\alpha_2 = \theta_{\min} + i_2 * \Delta\theta$$

3.2.3. Reducción de la PCL

La nube de puntos de la RGB-D (PCL_{RGB-D}) tienen un máximo de puntos de 307.200 (para el Kinect v1.0), este no es un tamaño constante dado que este varía dependiendo del ambiente en donde se encuentre; sin embargo, aun cuando no se presenten todos los puntos siguen siendo un gran volumen de datos, los cuales para poder ser procesados, primero fue seleccionada sola la profundidad de cada punto en la PCL_{RGB-D} y posteriormente, se aplicó un *Voxel Grid*, el cual proporciona un promedio de los puntos para cada segmento de una rejilla con un tamaño definido (t_{vg}), esta constituye la $PCL_{Reducida}$, esta nube de puntos se caracteriza por tener un menor número de puntos.

3.2.4. Marcos de referencia.

Para el desarrollo de los experimentos se tuvieron diferentes marcos de referencia con respecto a cada uno de los elementos que interactuaron en la ejecución ver Figura 15.

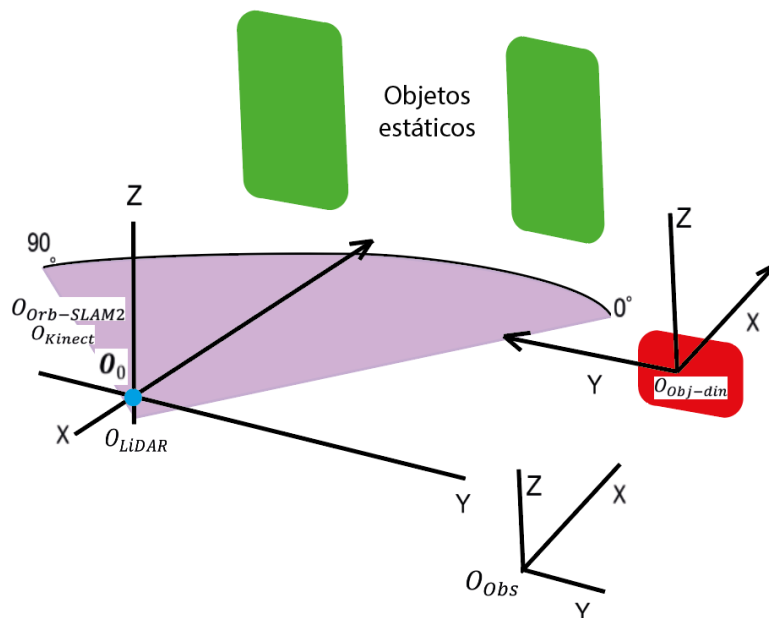


Figura 15 Marcos de referencia de los elementos que intervienen en los experimentos.

O_0 es el punto de origen del marco de referencia (MF_0), sobre el cual se determinará las PCL de reconstrucción, sobre este mismo coincide el marco de referencia O_{Kinect} , el cual corresponde al punto de origen del marco de referencia sobre el cual se determinará las PCL de cada ciclo, a su vez $O_{Orb-SLAM2}$ es el punto de origen del marco de referencia para el posicionamiento de los instrumentos que surge de las imágenes obtenidas del Kinect, estos tres marcos de referencia tienen el mismo punto de origen y se miden en coordenadas cartesianas. O_{LiDAR} es el punto de origen del marco de referencias del LiDAR el cual se mide en coordenadas polares y que se encuentra -5cm en el eje z con respecto a MF_0 y el ángulo del LiDAR se mide con respecto al eje y del MF_0 , estos no varían en la ejecución de las pruebas dado que se encuentran sobre la misma plataforma.

$O_{Obj-din}$ es el punto de origen del marco de referencia del objeto dinámico, este realizará movimiento sobre los ejes y y z, y todos los ejes son paralelos a los ejes del marco MF_0 . O_{Obs} es el punto de origen del marco de referencia del observador y es paralelo al marco de referencia MF_0 .

3.2.5. Segmentación por ángulo

En la segmentación por ángulo detectado, se segmentan la $PCL_{Reducida}$, extrayendo la zona delimitada por los ángulos de detección (α_1 y α_2). Para realizar esta, es necesario obtener el ángulo donde se encuentra cada punto de la PCL con respecto al eje y en el plano XY, ecuación (6).

$$r_{pt} = \sqrt{x_{pt}^2 + y_{pt}^2 + z_{pt}^2} \quad (6)$$

$$\alpha_{pt} = \cos\left(y_{pt}/r_{pt}\right)$$

A partir del r_{pt} radio de los puntos de la $PCL_{Reducida}$, se obtiene el ángulo α_{pt} de cada punto con respecto a eje y, se segmenta los puntos en cada ciclo, donde todo aquel que se encuentre entre α_1 y α_2 es considerado un punto de zona dinámica del ciclo k actual (PCL_{ZD}); asimismo los puntos

que se encuentran por fuera de esta zona angular son considerados puntos de zonas estáticas del ciclo k actual (PCL_{zek}).

3.2.6. Segmentación por agrupación por profundidad

En la segmentación por agrupación por profundidad se segmenta la PCL_{zD} donde se encuentran los puntos del objeto dinámico y los alrededores de la zona estática ocluida. Esta se realiza a partir de seleccionar la agrupación de puntos que presenten un x menor que el punto más lejano (x_{max}), del grupo, cuyo centroide se encuentra más proximo a los sensores estos se seleccionan como los puntos del objeto dinámico (PCL_{Din}). Así mismo, toda agrupación de puntos cuyo centroide se encuentre a mayor profundidad de x_{max} se considera puntos de la zona estática que rodea a la zona ocluida por la PCL_{Din} , y estos constituirá la nube de puntos PCL_{Oek} .

Con la PCL_{zek} y la PCL_{Oek} se realiza una concatenación y así se obtiene la PCL de la zona estática (PCL_{est-k}) del ciclo k-esimo, ecuación(7).

$$PCL_{zek} = \begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_s & y_s & z_s \end{bmatrix}$$

$$PCL_{Oek} = \begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_w & y_w & z_w \end{bmatrix}$$

(7)

$$PCL_{est-k} = PCL_{zek} \parallel PCL_k$$

$$p = s + w$$

$$PCL_{est-k} = \begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_p & y_p & z_p \end{bmatrix}$$

La PCL_{est-k} es la escena estática a la cual se le ha extraído los puntos correspondientes al objeto dinámico.

3.2.7. Traslación

La PCL_{est-k} tiene como marco de referencia de posicionamiento el centro óptico de los instrumentos; sin embargo, para llevar a cabo la reconstrucción, todas las nubes de puntos deben estar sobre el mismo marco de referencia. Para llevar a cabo el cambio de marco de referencia se debe tener la posición de los instrumentos en cada ciclo en referencia a una posición inicial, a partir de este se realiza la traslación de la PCL a la base de coordenadas del punto de origen del movimiento, para determinar la posición absoluta de cada punto de la PCL_{est-k} . El cambio de base de coordenadas se aplica al multiplicar cada punto por la matriz de transformación homogénea correspondiente para cada k-esimo ciclo. Ver Figura 16.

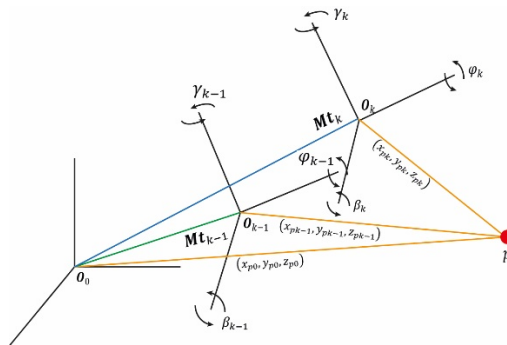


Figura 16. Vector traslación para cada marco de referencia con respecto al punto de origen.

p es un punto de la escena estática, M_t_k es la matriz de transformación homogénea para el k-ésimo ciclo, M_t_{k-1} para el k-ésimo -1 ciclo, x_{pk}, y_{pk} y z_{pk} son las coordenadas de un punto p que pertenece a la PCL_{est-k} del k-esimo ciclo, x_{pk-1}, y_{pk-1} y z_{pk-1} son las coordenadas de un punto que pertenece a la PCL del k-esimo-1 ciclo, O_0 es el punto de origen del marco de referencia

cuando k es igual a cero, \mathbf{O}_k es el punto de referencia para del k-esimo ciclo, \mathbf{O}_{k-1} es el punto de referencia para los puntos registrados en el k-esimo -1 ciclo.

Cada k-esimo ciclo tiene un correspondiente \mathbf{Mt}_k que está determinada por los parámetros en el vector de traslación (\mathbf{Vt}_k) el cual contiene las coordenadas de traslación P_x, P_y y P_z y los ángulos de rotación φ, β y γ para cada eje de coordenadas x, y y z las cuales se obtienen por odometría visual. La traslación parte de los puntos que se encuentran en la PCL_{est-k} , los cuales se encuentran en referencia a la base de coordenadas del k-esimo ciclo \mathbf{O}_k , estos se deben trasladan a el marco de referencia \mathbf{O}_0 por medio la matriz \mathbf{Mt}_k , esto se aplica para cada punto de la PCL_{est-k} , ecuación (8)

$$PCL_{est-k} = \begin{bmatrix} x_{r1} & y_{r1} & z_{r1} \\ \vdots & \vdots & \vdots \\ x_{rp} & y_{rp} & z_{rp} \end{bmatrix}$$

$$v_{rp} = \begin{bmatrix} x_{rp} \\ y_{rp} \\ z_{rp} \\ 1 \end{bmatrix}$$

(8)

\mathbf{Mt}_k

$$= \begin{bmatrix} \cos \varphi \cos \beta & \cos \varphi \sin \beta \sin \gamma + \sin \varphi \cos \gamma & \cos \varphi \sin \beta \cos \gamma + \sin \varphi \sin \gamma & P_x \\ \sin \varphi \cos \beta & \sin \varphi \sin \beta \sin \gamma + \cos \varphi \cos \gamma & \sin \varphi \sin \beta \cos \gamma - \cos \varphi \sin \gamma & P_y \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$v_{rtp} = \mathbf{Mt}_k \cdot v_{rp}$$

$$v_{rtp} = \begin{bmatrix} x_{rtp} \\ y_{rtp} \\ z_{rtp} \\ 1 \end{bmatrix}$$

$$PCL_k = \begin{bmatrix} x_{rt1} & y_{rt1} & z_{rt1} \\ \vdots & \vdots & \vdots \\ x_{rtp} & y_{rtp} & z_{rtp} \end{bmatrix}$$

En este proceso se tiene que cada vector v_{rp} que está constituido por x_{rp} , y_{rp} y z_{rp} , es multiplicado por la \mathbf{Mt}_k y se obtiene v_{rtp} , el cual contiene las coordenadas x_{rtp} , y_{rtp} y z_{rtp} , que conforman la PCL_k que es la PCL reducida del k -ésimo ciclo con respecto al marco de referencia \mathbf{O}_0 .

3.2.8. Reconstrucción

En la reconstrucción se realiza la concatenación sucesiva de las escenas de cada ciclo, para obtener una escena estática reconstruida donde se ha eliminado la oclusión. Este proceso implica completar la PCL_k del ciclo actual a la que le falta un segmento debido a que este se encontraba ocluido por el objeto dinámico, pero que es posible que se encuentre en PCL anteriores, por ello se concatena la PCL_k con la PCL final del ciclo previo $k-1$ (PCL_{kT-1}), ecuación(9).

$$PCL_{kT-1} = \begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_q & y_q & z_q \end{bmatrix}$$

(9)

$$PCL_{Trans.} = PCL_{kT-1} \parallel PCL_k$$

$$PCL_{Trans.} = \begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_{p+q} & y_{p+q} & z_{p+q} \end{bmatrix}$$

Del anterior proceso se obtiene una PCL transitoria ($PCL_{Trans.}$) total del ciclo actual, esta se caracteriza por tener zonas traslapadas, donde PCL_{kT-1} y PCL_k coinciden y tienen puntos que corresponden a la misma zona de la escena estática, ver Figura 17.

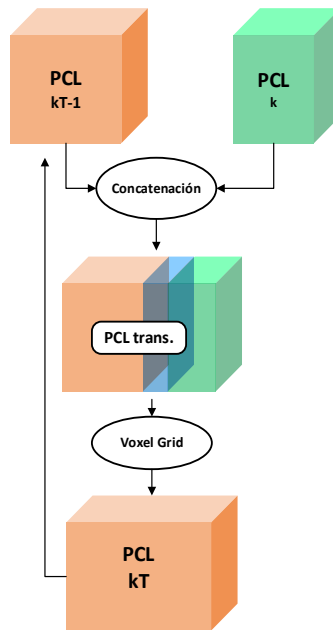


Figura 17. Concatenación y reducción de PCL para la reconstrucción de escenas estáticas

Este proceso de concatenación sucesiva permite obtener la PCL_{kT} que es la PCL reconstruida con los ciclos previos.

3.2.9. Correlación por vistas localizadas.

En esta investigación se propone una medida de la correlación entre una PCL de referencia (PCL_{ref}), que se captura en un fotograma de la escena estática sin objetos dinámicos, y la PCL_{kT}

la cuál se obtuvo de la reconstrucción con el método Rec-HV, para obtener la correlación las *PCL* deben tener el mismo marco de referencia. Las *PCL* contiene 3 columnas *X*, *Y* y *Z* en estas se encuentran las coordenadas de cada punto, primero se organiza la *PCL* de mayor a menor con respecto a la columna *X*, posteriormente se toma el máximo ($x_{max-cor}$) y mínimo ($x_{min-cor}$) de la columna *X* de profundidad de la escena para realizar una transformación a escala de grises, posterior se realiza un escalamiento sobre las columnas *Y* y *Z* y así transformarlas y obtener valores que tengan una correspondencia en pixel, ver ecuación (10).

$$\begin{aligned}
 Color &= 255 * (X - x_{min-cor}) / x_{max-cor} \\
 Y_{px} &= 100 * Y \\
 Z_{px} &= -100 * Z
 \end{aligned} \tag{10}$$

$$Pt = \begin{bmatrix} y_{px(1)} & z_{px(1)} & color_{(1)} \\ \vdots & \vdots & \vdots \\ y_{px(m)} & z_{px(m)} & color_{(m)} \end{bmatrix}$$

Este proceso de transformación se aplica sobre PCL_{ref} y la PCL_{kT} con esto se obtiene la Pt_{kT} y Pt_{ref} los cuales son una transformación de la matriz de tres columnas de *PCL* a una matriz de dos dimensiones y una escala de grises. Con las matrices en dos dimensiones se realiza la comparación, en esta se toma píxel a píxel y se establece la similitud entre las dos matrices de 2D obtenidas, esta comparación arroja el valor de la correlación lineal de las dos escenas.

3.3. Implementación

3.3.1. Ejecución sistema REC-HV

El sistema REC-HV se ejecuta en el sistema operativo Linux, en el lenguaje Python 3.8 con diferentes librerías de procesamiento matemático, estadístico y de ciencia de datos y en el *framework ROS v1.0 (Robotic operation system)* el cual establece una estructura por nodos. Este

está constituido por líneas de publicación y suscripción de mensajes entre los diferentes nodos que se están ejecutando. La arquitectura de software de Rec-HV, esta fue diseñada para tener una conexión de forma paralela con los instrumentos y una ejecución en cascada de los nodos correspondientes al sistema Rec-HV, (ver Figura 18).

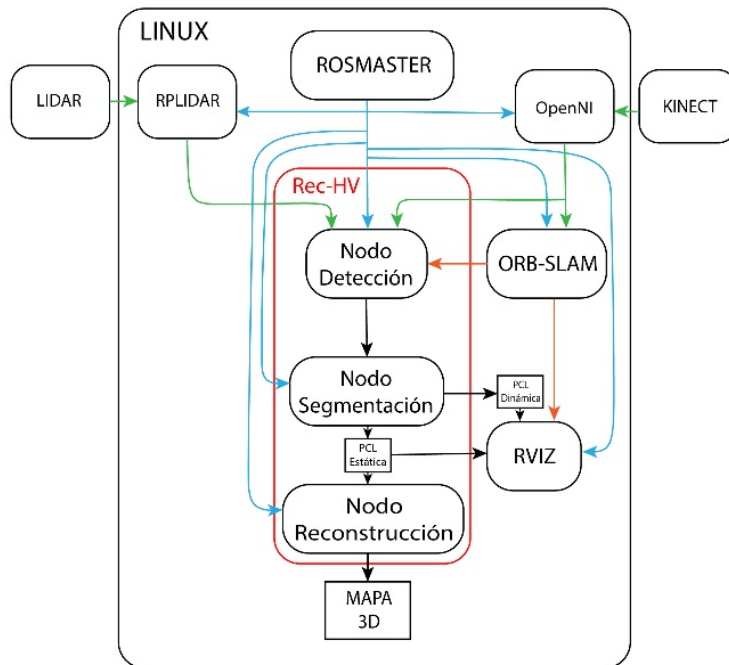


Figura 18. Arquitectura del sistema Rec-HV.

En esta se observa el nodo principal ROSMASTER el cual es el encargado de habilitar los canales para la ejecución del sistema, posterior al llamado de éste se ejecuta el nodo OpenNi, para la conexión con el RGB-D, habilitando los diferentes tipos de publicaciones; entre ellas, la PointCloud2 e imagen RGB, enseguida se activa el nodo de ORB-SLAM2, el cual aplica un set de datos de calibración propios para el RGB-D, este publica un mensaje tipo vector de posición. Posterior se inicia el nodo RPLIDAR, el cual identifica el LIDAR y habilita el canal por donde publica los mensajes del tipo LIDAR. Estos tres mensajes conforman el set de datos que recibe el sistema REC-HV y con los cuales inicia su procesamiento, de último, se inicia los nodos detección, segmentación y reconstrucción, colocando en ejecución el sistema completo.

En la Figura 18, también se muestra la conexión interna del sistema REC-HV, el cual está compuesto por tres nodos y tiene como salida tres mensajes tipo Point Cloud 2, uno con la zona dinámica, el segundo con la zona estática y el último con la reconstrucción de la escena en cada ciclo. Dado que los tres nodos de Rec-HV corresponden al desarrollo realizado en este proceso de investigación, el código correspondiente está como anexo digital, ver Anexo 3. Repositorios.

Rec-HV está compuesto por tres etapas distribuidas en tres nodos. Nodo detección: en él se realiza el análisis estadístico a los datos del LIDAR, se detectan las zonas dinámicas y se reduce la densidad de puntos de la *PCL*. Nodo segmentación: en él se aplica la fusión sensorial de los datos para segmentar la zona dinámica en la *PCL* y así obtener el primer segmento de zonas estáticas de la *PCL*, en seguida y a partir de la zona dinámica, se realiza una segmentación por agrupación y por profundidad, en donde se establecen los puntos del objeto dinámico, los cuales son eliminados, dejando solo los puntos pertenecientes a zona estática; por último, se realiza la traslación de la *PCL*. Nodo reconstrucción: en él se concatena la zona estática del ciclo actual con la del ciclo anterior y así se obtiene la escena estática completa sin la zona ocluida por el objeto dinámico, a esta *PCL* resultante se le aplica una reducción de densidad de puntos, (ver Figura 19).

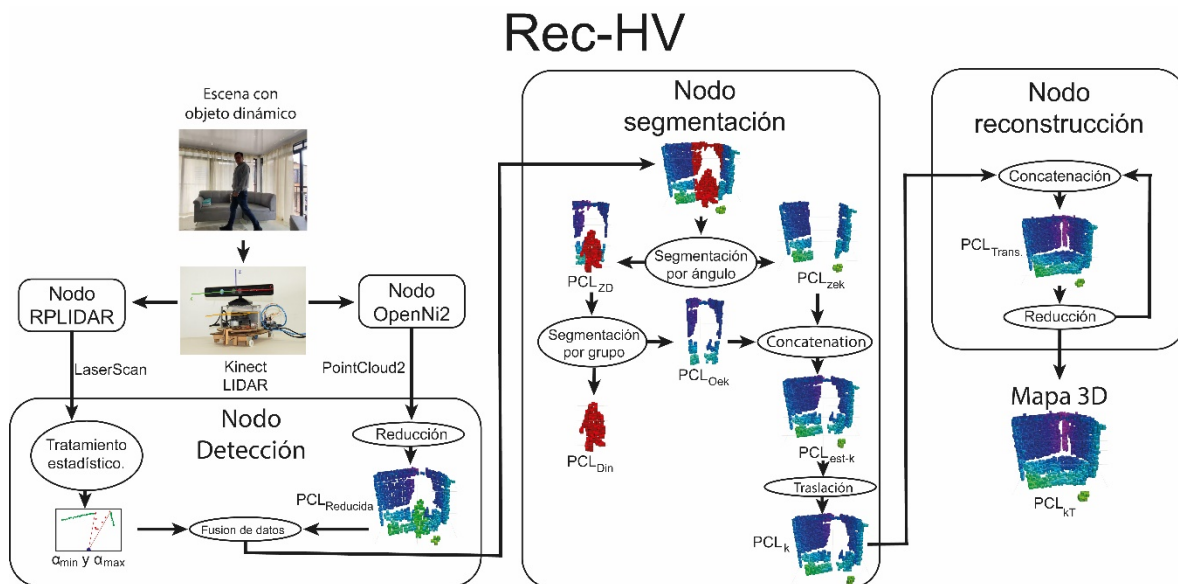


Figura 19. SLAM con el método Rec-HV integrado.

3.3.2. Nodo detección

El nodo detección se encuentra conectado a tres líneas de entrada de mensajes, las cuales proporcionan el set de datos de tipo LaserScan del LIDAR, el set de datos de tipo PointCloud2 del RGB-D que constituye la PCL_{RGB-D} y un mensaje tipo vector con 6 parámetros correspondientes al posicionamiento, el cual lo suministra por el nodo de ORB-SLAM2, el vector de traslación (Vt_k). Se genera como salida un mensaje de tipo PointCloud2, el cual contiene la $PCL_{Reducida}$, el Vt_k y los ángulos de segmentación (α_1, α_2), a través de la línea de mensaje de ROS. El nodo detección realiza el preprocesamiento de los datos del LIDAR, la detección de la zona de alta variabilidad y la reducción de la PCL de la RGB-D.

3.3.2.1. Interpolación y Extrapolación

Los datos del vector r , al provenir de un entorno real, presentan valores no definidos, los cuales deben ser sustituidos. Este proceso se realiza dependiendo de la posición dentro del vector donde se encuentren. Cuando el sistema identifica un valor inconsistente (infinito) que se encuentra en medio de dos valores reales, se guarda el anterior dato real y se van acumulando en un vector de sustitución, todos los ángulos con valor de profundidad inconsistente que sean consecutivos, hasta que dentro del vector r encuentre un nuevo dato real, estos dos valores reales extremos del vector sustitución, constituirán los puntos para realizar la interpolación del segmento detectado con valores irregulares, donde todos los valores inconsistentes se reemplazarán por el valor correspondiente dado por la línea de tendencia de los dos extremos.

Cuando el vector comienza con valores de profundidad infinito, se registran los ángulos para los cuales se detectó infinito, esto se realiza hasta encontrar un primer valor real, este es registrado y se continua con la búsqueda de un segundo valor real consecutivo, con estos dos valores se realiza la extrapolación. De forma similar se aplica cuando los valores infinito se encuentran en el extremo final del vector, en el, al recorrer el vector si se encuentra con un valor infinito se registra el dato anterior real, si al seguir registrando los ángulos para los cuales la profundidad tiene valor infinito, se encuentra que ya no existe un valor real por que se recorrió el vector hasta el extremo, se comienza la búsqueda hacia atrás, comenzando en la posición anterior del valor real ya registrado, esta búsqueda se realiza hasta encontrar un valor real, con estos dos valores se realiza la

extrapolación a partir de la línea de tendencia que se genera por los valores reales. Este proceso genera un vector \mathbf{r} sin valores inconsistentes.

3.3.2.1. Promedio móvil y desviación

Para la aplicación del promedio móvil se utilizaron las herramientas de tratamiento estadístico de la librería Numpy de Python. En este se tiene una matriz donde se guardan los valores del vector \mathbf{r} del ciclo actual y de los $n - 1$ ciclos anteriores, con estos en cada ciclo y para cada set de n datos se crean los vectores $\overline{\mathbf{r}}_k$ y \mathbf{dr}_k , que junto con el vector $\boldsymbol{\theta}_k$, constituyen el set de datos del LiDAR.

3.3.2.2. Ángulo de detección

A partir del conocimiento que brindan los vectores $\overline{\mathbf{r}}_k$, $\boldsymbol{\theta}_k$ y \mathbf{dr}_k en cada ciclo se realiza el análisis de los máximos de variabilidad. En él, se buscan los valores de profundidad de los $\overline{\mathbf{r}}_k$ cuyo valor de dr , sean más altos, con ello se determina los ángulos que delimitan las fronteras de la posición del objeto dinámico para el ciclo k -ésimo. Para ello se proporciona un umbral, dado por la mitad del máximo valor de dr presente en el \mathbf{dr}_k de cada ciclo, todo valor que supere este umbral será clasificado como punto de alta variabilidad, de este conjunto de valores se seleccionan los dos valores máximos (dr_{max1} , dr_{max2}) que no estén en posiciones consecutivas dentro del vector. De esta forma los l datos de \mathbf{dr}_k serán divididos en zonas delimitadas por puntos de alta variabilidad y se determina el máximo número de puntos de alta variabilidad consecutivos, dado que estos corresponden a cambios bruscos de la escena por presencia de objetos dinámicos.

Las zonas delimitadas se clasifican como estáticas o dinámicas, para ello se usa dr_{max1} y dr_{max2} . La zona que se encuentre rodeada por los valores de máxima variabilidad será seleccionada como zona dinámica, si existe más de una zona clasificada como dinámica se tendrá como criterio la zona que se encuentre más cerca de la zona anterior seleccionada como dinámica y, si no existe zona anterior dinámica registrada, se selecciona por el máximo número de puntos de alta variabilidad consecutivos. De este proceso se obtiene dos posiciones de los puntos frontera $i1$ y $i2$ de la zona dinámica. Dado que para cada valor de i existe un θ correspondiente, por ende, se obtienen los ángulos $\alpha1$ y $\alpha2$ del vector $\boldsymbol{\theta}_k$, ver ecuación (5).

3.3.2.3. Reducción de PCL RGB-D

La PCL_{RGB-D} presentan una estructura donde cada punto de un set de datos está compuesto R, G y B como registro del color y x, y, z como coordenadas de posición. Esta PCL_{RGB-D} de entrada se caracteriza por estar constituida por un gran número de datos, lo que dificulta el procesarla en tiempo real ya que esto implica un gran peso computacional. A la PCL_{RGB-D} se le redujo la densidad de puntos con el Voxel Grid, en este se establece una rejilla tridimensional con arista de cada celda (t_{vg}) fija, adicionalmente solo se conservaron los datos de posicionamiento, este proceso genera la $PCL_{Reducida}$ para cada ciclo. En el voxel grid establece una grilla y promedia los puntos dentro de cada celda de la grilla, obteniendo un solo punto representativo, este proceso conserva las características del entorno.

3.3.3. Nodo segmentación

El nodo segmentación se encuentra conectado a la línea de mensaje que proporciona el nodo detección, este suministra un mensaje tipo PointCloud2 el cual contiene la $PCL_{Reducida}$, el Vt_k y los ángulos de segmentación (α_1, α_2). Se genera como salida dos mensajes de datos tipo PointCloud2, los cuales contienen la PCL_{est-k} y PCL_{Din} . En este nodo se realiza la fusión de datos y la segmentación de la $PCL_{Reducida}$ en la PCL_{est-k} y PCL_{Din} .

3.3.3.1. Fusión de datos

El proceso de Fusión de datos requiere alinear los centros geométricos del LIDAR y cámara RGB-D. El LIDAR presenta una apertura angular de 90 grados en un plano XY (zona morada), mientras que el cámara RGB-D presenta una apertura tridimensional de 57 grados (zona verde). La apertura del LIDAR y cámara RGB-D presenta una diferencia, esta corresponde a un ángulo de detección previo del LIDAR, con ello se establece las fronteras de la PCL de la zona dinámica (PCL_{ZD}), antes del ingreso del objeto dinámico en la PCL_{RGB-D} . Estas fronteras corresponden a los ángulos de detección (α_1 y α_2) que determinan la zona dinámica a segmentar sobre la $PCL_{Reducida}$, (ver Figura 20).

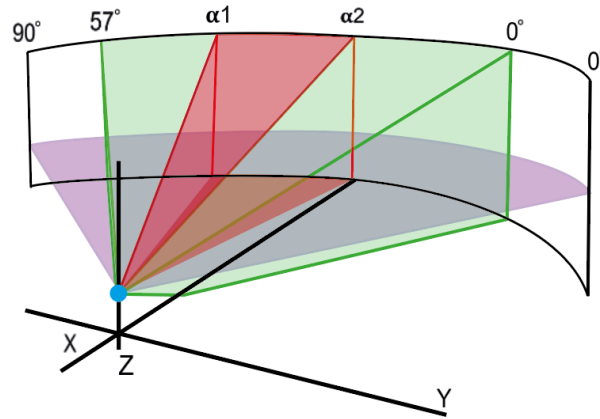


Figura 20. Ángulos de segmentación, apertura del RGB-D y plano de detección del LIDAR.

En la segmentación por agrupación, por profundidad, se segmenta la PCL_{ZD} donde se encuentran los puntos del objeto dinámico y los alrededores de la zona estática ocluida. En este segundo proceso de segmentación se realiza un análisis de la agrupación de los puntos por profundidad en la PCL_{ZD} . Para ello se grafican estos puntos en un vector con respecto al eje x, este estará constituido por cuadros con arista de tamaño t_{vg} , posteriormente se aplica una detección de zonas, y se selecciona la zona más cercana al origen como zona donde se encuentra el objeto dinámico ver Figura 21.

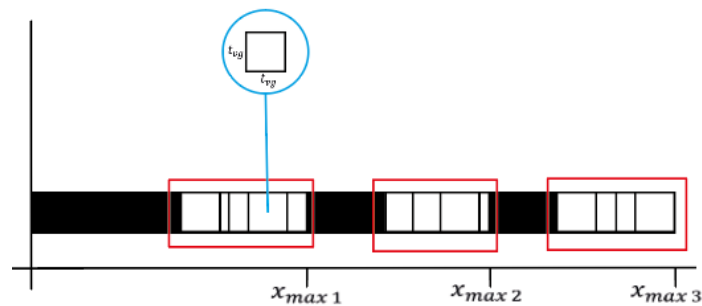


Figura 21. Segmentación por agrupación por profundidad.

De esta se obtiene el parámetro máximo de profundidad (x_{max}) para los puntos con respecto al eje x , donde todo aquel punto que su coordenada x sea menor a este se considera que pertenece a la PCL del objeto dinámico (PCL_{Din}) y, todo punto que sea mayor, se considera que pertenece a la PCL de la zona estática que rodea a la zona ocluida (PCL_{Oek}). Con la PCL_{zek} y con la PCL_{Oek} se realiza una concatenación que da como resultado la PCL_{est-k} , la cual es la escena estática a la cual se le ha extraído los puntos correspondientes al objeto dinámico.

Dada la PCL_{est-k} , se realiza el proceso de traslación en la cual se toma cada v_{rp} de la PCL_{est-k} y es multiplicada por la Mt_k de este proceso se obtiene como resultados los v_{rtp} , que constituyen la PCL_k .

3.3.4. Nodo Reconstrucción

El nodo reconstrucción se encuentra conectado a la línea de mensajes que proporciona el nodo segmentación, este suministra la PCL_k . Se genera como salida una línea de mensajes de datos de tipo PointCloud2 la cual contiene la PCL_{kT} . En este nodo se realiza la concatenación sucesiva de PCL para obtener la reconstrucción de la escena.

Al aplicar la función concatenación de la librería *Numpy Python* entre la PCL_k y PCL_{kT-1} se genera la $PCL_{Trans.}$, en esta se encuentran zonas donde se tiene una mayor densidad de puntos, esta es dada por que la PCL_k y PCL_{kT-1} presentan zonas en común, esto genera un aumento del tamaño de la $PCL_{Trans.}$. Para reducir la densidad de puntos se aplica el *Voxel Grid* con un tamaño de celda de t_{vg} este da como resultado la PCL_{kT} , la cual es la PCL de reconstrucción de la escena estática.

3.3.5. Correlación

La correlación es el único proceso que se ejecuta fuera de línea, este no está incluido dentro del sistema Rec-HV, dado que este es utilizado para evaluar la correlación lineal entre las dos imágenes obtenidas al aplicar el proceso de transformación de la PCL_{kT} y PCL_{ref} las cuales son nubes de puntos 3D, que deben estar sobre el mismo marco de referencia.

En el proceso de transformación se aplica las ecuaciones (10), en estas se obtienen representaciones en dos dimensiones la Pt_{kT} y Pt_{ref} , de las PCL , (ver Figura 22).

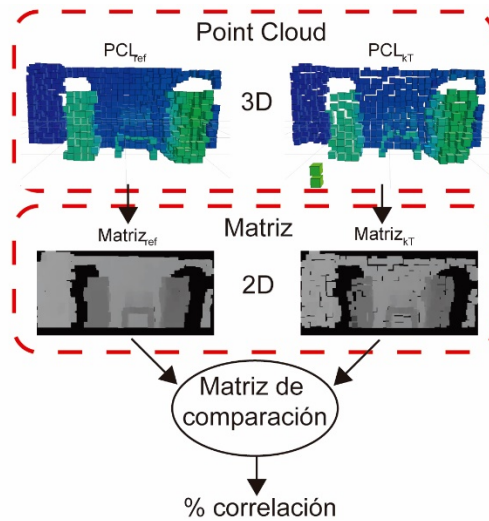


Figura 22. Correlación

Una vez se tiene las imágenes 2D, se procede a utilizar la función `matchTemplate` de la librería `opencv` de Python, esta realiza la convolución de las imágenes hasta observar la coincidencia entre las dos imágenes, para más información el software se encuentra en el Anexo 3. Repositorios.

3.4. Materiales

3.4.1. Sensores.

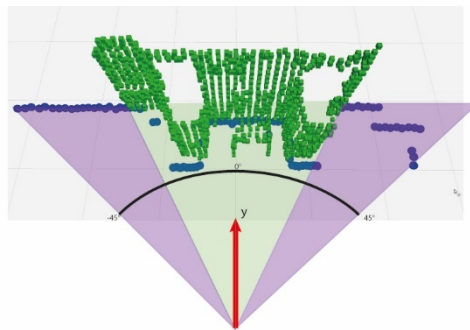
Para desarrollar esta investigación se usó el sistema operativo robótico (ROS V1.0), en Ubuntu 18.06, con una computadora DELL Inspiron N5110, un procesador core i7 de 3.2 GHz con cuatro núcleos reales y cuatro núcleos virtuales, una memoria RAM de 16 GB y una GPU de 512 MB y un computador DELL G15 con procesador Razer 7 5800H de 8 núcleos y 16 hilos de 4.4GHz, RAM de 32GB y una GPU de 4GB. Los sensores utilizados fueron un Kinect V1 tipo RGB-D (con

57° horizontal, 43° vertical, imagen de profundidad de 320 x 240 y una imagen RGB de 640 x 480, con tasa de muestro a 30fps) con un adaptador USB a tipo A; y un LIDAR RPLIDAR A1 (con un máximo profundidad de 8 m, 360° y una tasa de muestreo de 7Hz), ver Figura 23(A). Los sensores se instalaron haciendo coincidir los centros geométricos de los instrumentos; la dirección del eje Y del RGB-D coincidió con el ángulo 0° del LIDAR Figura 23(B).

Los instrumentos presentan diferente amplitud en el campo visual; el RGB-D con 57° (zona verde Figura 23[B]) y el LiDAR cuenta con un campo visual de 90° (zona morada Figura 23[B]), con ello se tiene un ángulo de detección previa de 16,5° hacia los costados sobre el plano XY. (ver Figura 23[B]).



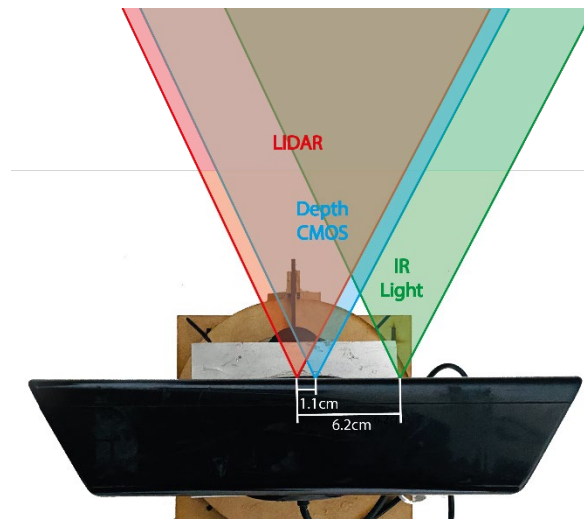
A



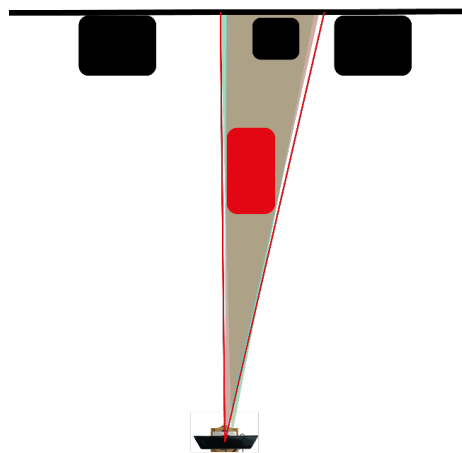
B

Figura 23. (A) instrumentos de medición LIDAR y RGB-D en base soporte. (B) Campo visual de los instrumentos.

Los instrumentos tienen una correspondencia entre su centro geométrico, sin embargo, por el diseño del RGB-D no significa que la posición de los sensores coincida, por ende, no concuerda con el centro visual de los instrumentos, (ver Figura 24(A)). La diferencia de las posiciones de los instrumentos genera que exista un error de proyección en el ángulo detectado por el LIDAR, el cual es proyectado sobre la PCL del RGB-D, ver Figura 24(B).



A



B

Figura 24. (A) Ubicación de los sensores en el RGB-D. (B) Proyección de campo visual del LIDAR y del RGB-D.

Por la diferencia de posición de los sensores, se agregará un margen de tolerancia en el ángulo de segmentación (líneas rojas Figura 24). A este ángulo detectado se adiciona el espacio correspondiente a 1° equivalente a entre 2cm a 6cm, dependiendo de la profundidad. Por otro lado, se tiene que el voxel grid proporciona un promedio de los puntos que se encuentran en una grid tridimensional de 8cm, dado a esto, todos los puntos que se encuentren dentro de esta zona son representados por un punto central, de este modo se agrega un espacio entre el punto y la frontera. Estos dos procesos dan un margen de entre 2cm a 10cm hacia cada lado de la zona a segmentar disminuyendo la posibilidad de que los puntos del objeto dinámico queden por fuera.

3.4.1.1. Calibración del RGB-D

Antes de realizar el proceso de captura de datos se realizó la calibración del RGB-D, el cual por sus características de fabricación y de materiales, se le debe aplicar algunos parámetros de calibración que permitirán eliminar problemas de aberración e inconsistencia entre la imagen de la cámara RGB y de la cámara de profundidad. Para este proceso se utilizó una herramienta que se ejecuta a través de ROS, la cual toma imágenes en donde debe estar una cuadrícula de guía ubicada a diferente profundidad, ver Figura 25 (A). Esta herramienta ubica los puntos sobre la cuadrícula y los enmarca, (ver Figura 25 [B]).



A



B

Figura 25. Imágenes usadas en el proceso de calibración. (A) imágenes de referencia. (B) imágenes donde se localizó la cuadrícula para calibración.

Este proceso proporciona el error de reproyección, el cual tiene un valor en particular para cada instrumento y se mide con respecto a un píxel, ver Anexo 1. Error de reproyección ecuación (11).

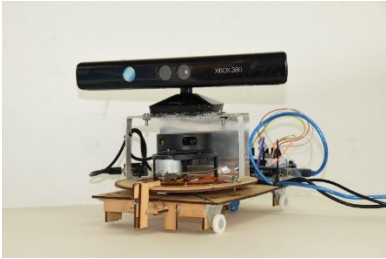
$$e_{rep} = 0.114397 \quad (11)$$

Adicional, el software proporciona una serie de matrices, estas son guardadas dentro de un archivo que el software del RGB-D lee al iniciar su ejecución, con ello logra reducir el error del instrumento.

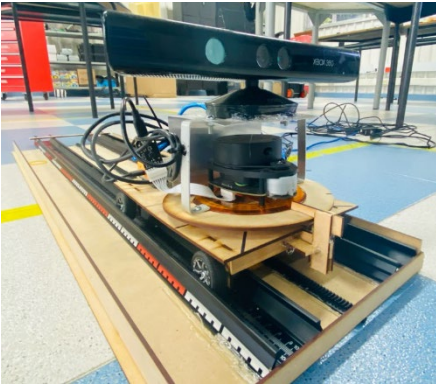
3.4.2. Montajes para los sensores.

Para el desarrollo de las pruebas fue necesario diseñar tres diferentes montajes para los sensores: en el primer montaje, el centro visual del RGB-D se encontraba a 18cm y el LIDAR a 13cm de altura, (ver Figura 26 [A]). En el segundo montaje, el centro visual del RGB-D se encontraba a 21cm y el LIDAR a 16cm de altura y estos se encontraban sobre una plataforma de desplazamiento lineal, la cual permitía medir el desplazamiento en el eje x, mientras es desplazada por propulsión manual, (ver Figura 26 [B]). En el tercer montaje, el centro visual del RGB-D se encontraba

a 120cm y el LIDAR a 115cm de altura, estos se hallaban sobre una plataforma móvil automática, la cual se desplaza a una velocidad promedio de 0.036m/s, (ver Figura 26[C]).



A



B

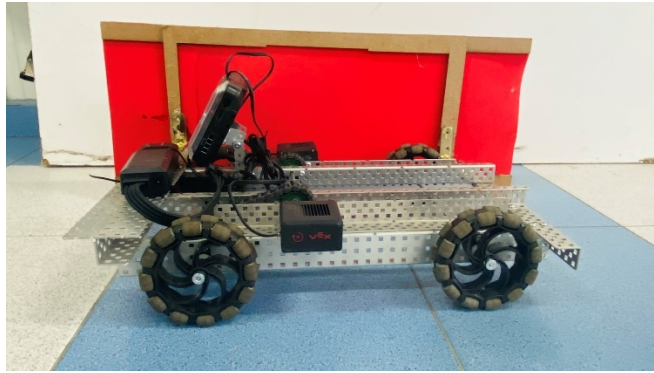


C

Figura 26. (A) Sensores sobre plataforma fija. (B) Sensores sobre plataforma de desplazamiento lineal. (C) Sensores sobre plataforma móvil automática.

3.4.1. Objetos dinámicos.

Se usaron dos tipos de objetos dinámicos, un robot móvil constituido por un controlador V5 Robot Brain y dos motores V5 Smart con caja de reducción 36:1 y configurados al 80% de la potencia, con lo cual el robot alcanzó una velocidad promedio de 0,98 m/s, este fue equipado con una pantalla lateral de 64 cm x 27 cm para dar uniformidad a su superficie, (ver Figura 27[A]). También y dada la necesidad de realizar pruebas en escenarios reales se tuvo y una persona como objeto dinámico, la cual realizó recorridos a una velocidad promedio de 1.4m/s, (ver Figura 27[B]).



A



B

Figura 27. (A) objeto dinámico robótico con pantalla, (B) objeto dinámico.

3.4.2. Escenas

En las pruebas realizadas con el sistema Rec-HV, fueron utilizados diferentes escenarios con diferentes características y complejidad. En la Tabla 2 se muestra una relación de los ocho escenarios utilizados, los cuales se presentan en la Figura 28.

Tabla 2. Características de los escenarios de prueba.

| Escenas. | Características. |
|----------|---|
| A | Escena estática sin objetos estáticos y profundidad promedio de 4m. |
| B | Escena estática con objetos estáticos de materiales de alta reflectancia y profundidad aproximada 4m. |
| C | Escena con superficies de poca reflectividad y profundidad promedio de 4m. |
| D | Escena estática con espacios abiertos materiales diversos, superficies de poca uniformidad y profundidad máxima de 6m. |
| E | Escena estática con espacios abiertos, materiales con alta reflectividad y profundidad máxima de 5m. |
| F | Escena estática con una distribución de objetos estáticos a diversas profundidades, zonas de baja reflectividad y profundidad máxima de 8m. |
| G | Escena estática con objetos de baja reflectividad y profundidad máxima de 6m. |
| H | Escena con poca luz, superficies con poca reflectividad y profundidad promedio de 4m. |



A



B



C



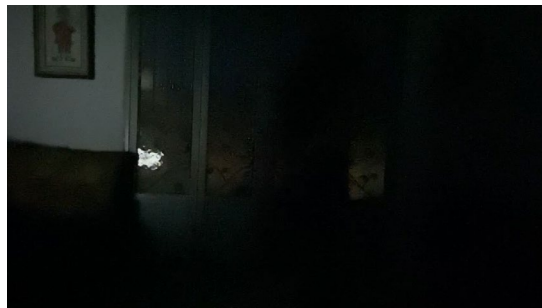
D



E



F



G

H

Figura 28. Escenarios de pruebas.

Capítulo 4. Resultados

En este capítulo se presentan once experimentos, en los cuales se realizaron varias pruebas con el sistema Rec-HV. Los experimentos fueron diseñados con distintos niveles de complejidad, las pruebas realizadas en cada experimento están compuestas por ciclos, en cada ciclo se capta el comportamiento de diferentes instancias del sistema Rec-HV, con ambientes estáticos de diversas características y con la presencia de un objeto dinámico, el cual realizó diferentes recorridos en las escenas. Para cada experimento se usaron una serie de métricas, (ver Tabla 3).

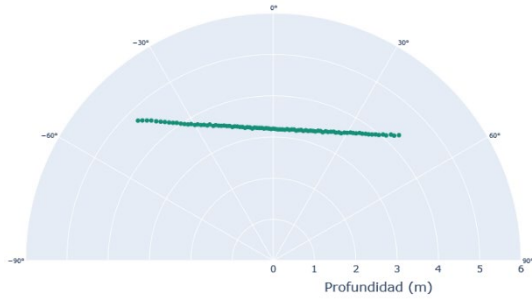
Tabla 3. Métricas de los experimentos.

| Experimento | Métricas |
|-------------|--|
| 1 | Número de puntos con valores atípicos en cada ciclo. |
| 2 | Número de puntos con valores atípicos en cada ciclo posterior al procesamiento estadístico. Evaluación cualitativa de la variación de los puntos del LIDAR ante la variación de la ventana de ciclos para el promedio móvil. |
| 3 | Variación de la desviación estándar (metros) de cada punto del LiDAR en diferentes escenarios. |
| 4 | Variación del tiempo de ejecución (segundos), variación del tiempo de segmentación (s), variación del número de puntos en la PCL con respecto al tamaño del Voxel Grid(metros). |
| 5 | Número de escenas donde se detectó de forma correcta el objeto dinámico |
| 6 | Variación del número de puntos de la PCL de la escena estática y de la zona dinámica. |

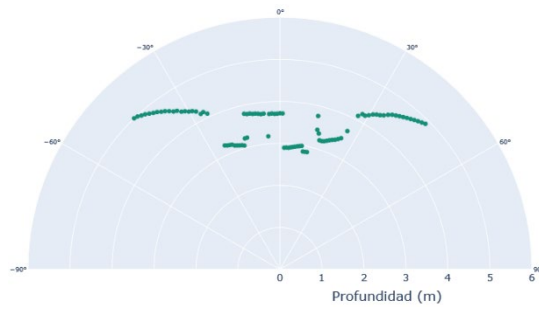
| | |
|----|--|
| 7 | Variación del porcentaje de correlación. |
| 8 | Tiempo promedio de ejecución (segundos), tiempo promedio de segmentación (s), número de puntos por ciclo, porcentaje de reducción promedio de la PCL, porcentaje de ciclos con puntos de alta variabilidad, porcentaje de la PCL obtenida en la segmentación por ángulo, porcentaje de la PCL obtenida en la segmentación por agrupación por profundidad, número de puntos de la PCL inicial (PCL obtenida en el primer ciclo), número de puntos de la PCL final (PCL final obtenida por la concatenación) y porcentaje de incremento de puntos en la PCL de reconstrucción. |
| 9 | Error máximo de posicionamiento (metros), error absoluto de posicionamiento (metros), error cuadrático medio (metros). |
| 10 | Error máximo de posicionamiento (metros), error absoluto de posicionamiento (metros), error cuadrático medio (metros), entre el sistema de odometría de referencia y el sistema de odometría visual. |
| 11 | Variación del porcentaje de correlación de la reconstrucción con la PCL de referencia, variación del número de puntos de la PCL de reconstrucción y comparación cualitativa de los resultados de la reconstrucción. |

4.1. Experimento 1

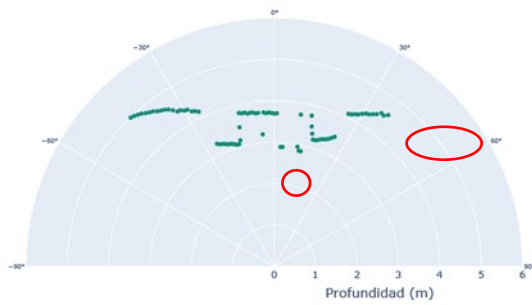
Para la realización de este experimento, se tuvo los escenarios A y B (ver Figura 28). Este experimento está constituido por dos pruebas sin objeto dinámico. En el primero se realizó la captura de los puntos del LIDAR sobre el escenario A, en este los puntos del LIDAR presentan una distribución consistente con la configuración del escenario en cada ciclo, (ver Figura 29[A]). Así mismo, se realizaron captura de datos sobre el escenario B, en este se observó que en algunos de los ciclos se obtiene una lectura consistente con el escenario, (ver Figura 29[B]). Y en algunos ciclos, se tiene una lectura con zonas donde no se logra determinar el valor de la profundidad (zonas enmarcadas con círculo rojo), (ver Figura 29[C]).



A



B



C

Figura 29. Distribución de los puntos captados por el LIDAR. (A) Escena estática sin objetos estáticos escenario A. (B) Escena estática con objetos estáticos escenario B. (C) Escena estática con objetos estáticos escenario B con inconsistencias.

Al encontrar estas zonas faltantes se realizó una prueba de evaluación sobre 500 ciclos, en esta se encontró que en el 94.4% de los ciclos se presentan punto no determinados y el número de estos puntos no determinados varía en cada ciclo (ver Figura 30).

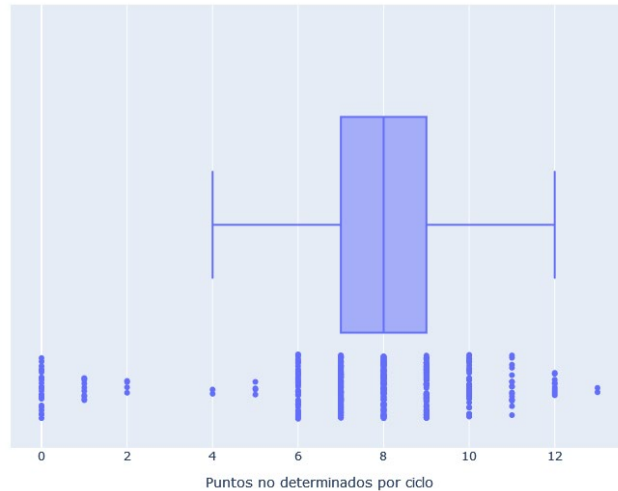


Figura 30. Puntos no determinados por ciclo de lectura del LIDAR.

4.2. Experimento 2

Para la realización de este experimento se tuvo los escenarios A y B (ver Figura 28) y el objeto dinámico robótico (ver Figura 27[A]). En esta primera prueba, se utilizó el escenario B y no se tuvo presencia del objeto dinámico. Al analizar más a fondo el experimento 1 se encontró que en el vector \mathbf{r} , de algunos ciclos, presentaban zonas donde la lectura tenía valores inconsistentes, estos se presentan en medio de dos valores reales (ver columna lectura original Tabla 4[A]). O en los extremos del vector \mathbf{r} (ver columna lectura original Tabla 4[B]). Sobre los valores que están en medio de valores reales en el vector \mathbf{r} , se aplicó interpolación (ver columna interpolación Tabla 4[A]). Sobre los que se encuentran al extremo del vector se aplicó la extrapolación, (ver columna extrapolación Tabla 4[B]).

Tabla 4. Segmentos de datos de los vectores ángulo y profundidad donde se presentaron valores no reales.

| Ángulo rad | Lectura Original | Interpolación |
|---------------|---------------------|---------------|
| 1,588 | 3,697 | 3,697 |
| 1,606 | Inf | 3,293 |
| 1,623 | 2,889 | 2,889 |
| 1,641 | 2,891 | 2,891 |
| 1,658 | Inf | 3,030 |
| 1,676 | Inf | 3,168 |
| 1,693 | Inf | 3,306 |
| 1,710 | Inf | 3,445 |
| 1,728 | Inf | 3,583 |
| 1,745 | 3,722 | 3,722 |

A

| Ángulo rad | Lectura Original | Extrapolación |
|---------------|---------------------|---------------|
| 2,199 | 4,502 | 4,502 |
| 2,217 | 4,576 | 4,576 |
| 2,234 | Inf | 4,650 |

| | | |
|-------|-----|-------|
| 2,251 | Inf | 4,724 |
| 2,269 | Inf | 4,798 |
| 2,286 | Inf | 4,872 |
| 2,304 | Inf | 4,946 |
| 2,321 | Inf | 5,020 |
| 2,339 | Inf | 5,094 |
| 2,356 | Inf | 5,168 |

B

A partir de los puntos con valor real, puntos verdes, (ver Figura 31) se la aplicó la interpolación, puntos de azules, (ver Figura 31). Y extrapolación, puntos rosados, (ver Figura 31). Esto permitió el tener 0% de ciclos con puntos faltantes sobre una prueba de 500 ciclos.

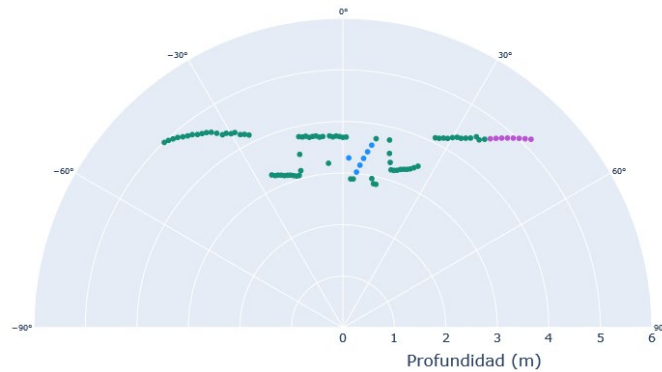
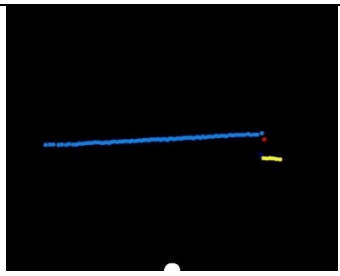
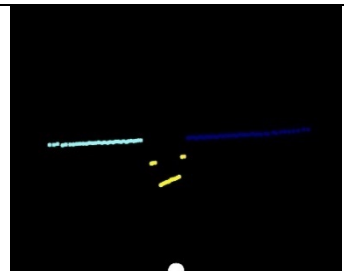
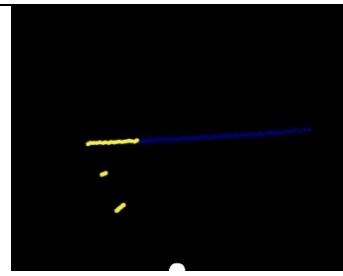

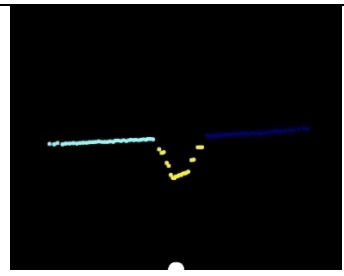
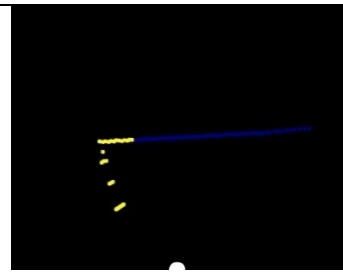
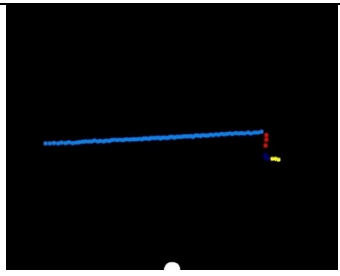
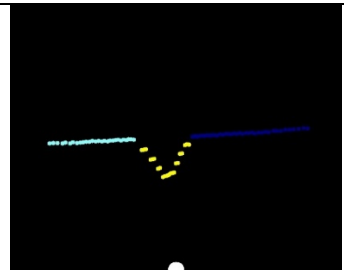
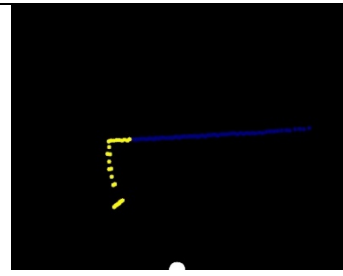


Figura 31. Lectura de puntos del LiDAR sin valores no reales.

Una vez se obtuvo un vector r completo con valores reales, se procedió a aplicar el promedio móvil, sin embargo, no se tenía el valor de n para este promedio.

Para estas pruebas, se utilizó la escena A y el objeto dinámico robótico, este cruzaba la escena de izquierda a derecha en cada prueba y estuvieron constituida por 25 ciclos en total. Con el fin de definir el valor de n , para el número de ciclos sobre el cual se aplica el promedio móvil, se realizaron pruebas donde se le dieron valores a n entre 2 y 7, (ver columna n Figura 32). Para observar el resultado de la prueba, se seleccionaron 3 ciclos representativos de la prueba: cuando ingresa el objeto dinámico, (ver columna A Figura 32), cuando se encuentra en un punto intermedio, (ver columna B Figura 32), y cuando se encuentra próximo a terminar el recorrido, (ver columna C Figura 32).

| n | A | B | C |
|---|---|--|---|
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |

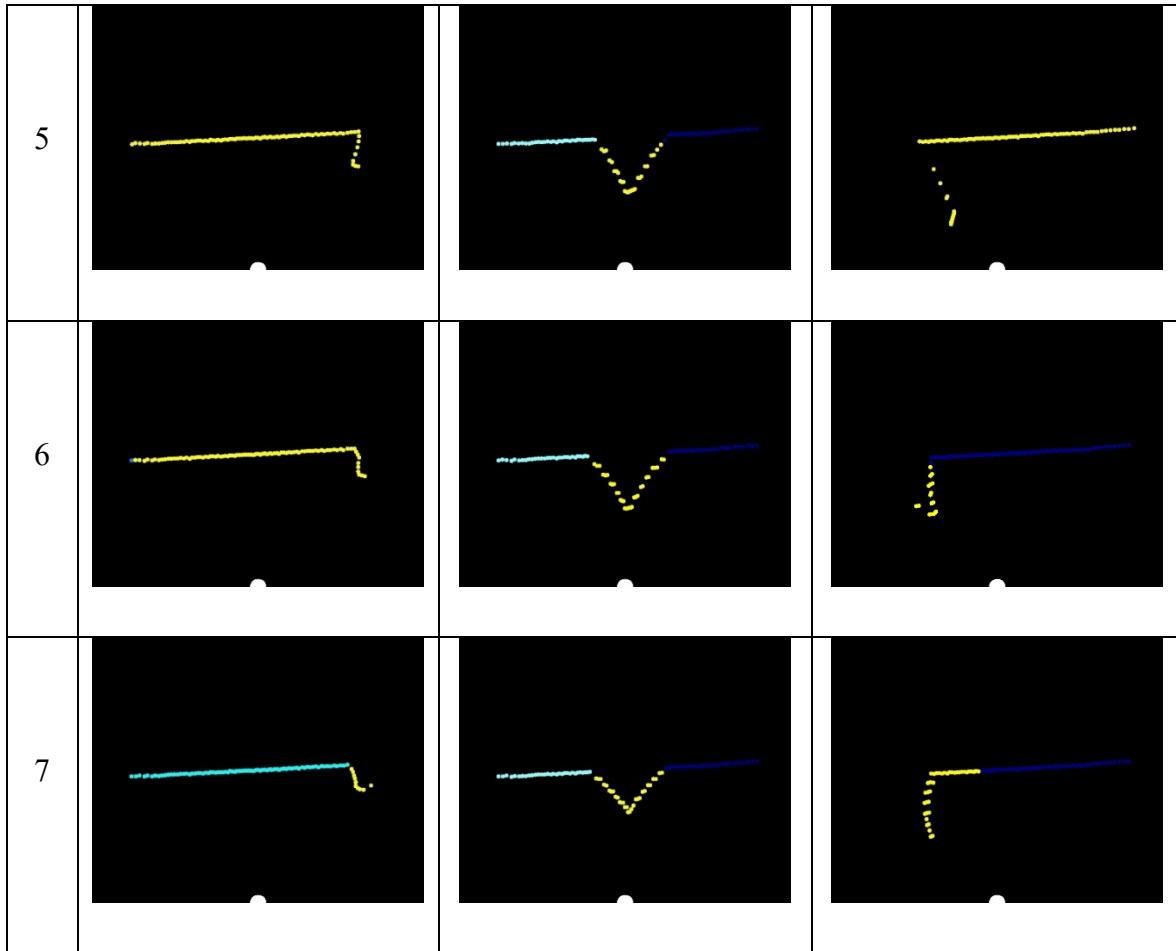


Figura 32. Prueba valor de n , Columna n valores asignados a n . Columna A objeto dinámico ingresando a la escena. Columna B objeto dinámico en punto intermedio. Columna C objeto dinámico ya está saliendo de la escena.

En este experimento se observó cómo el número de ciclos sobre los cuales se aplica el promedio móvil afecta la forma del objeto dinámico, a partir de esto se seleccionó como valor de n igual a 3.

Con el vector r donde todos sus valores son reales y definido el valor de n igual a 3, se procedió a aplicar el proceso estadístico sobre la escena B. En esta prueba se toman los dos ciclos anteriores, puntos color azul y verde, (ver Figura 33), al ciclo actual, puntos color naranja, (ver Figura 33), se promedia cada punto, lo cual da como resultado una distribución de puntos representativos que son consistentes con la escena B, puntos color morado, (ver Figura 33).

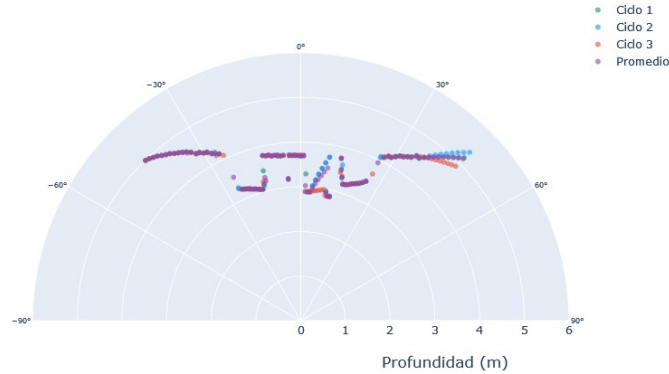
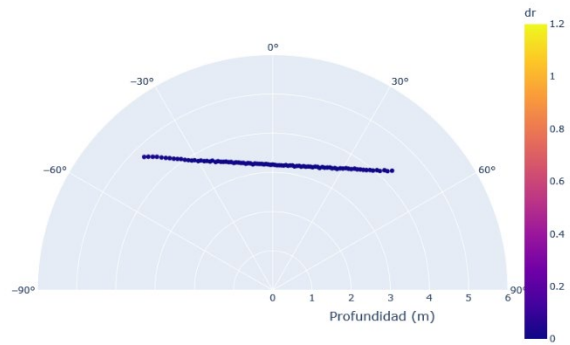


Figura 33. Aplicación del promedio móvil para $n = 3$.

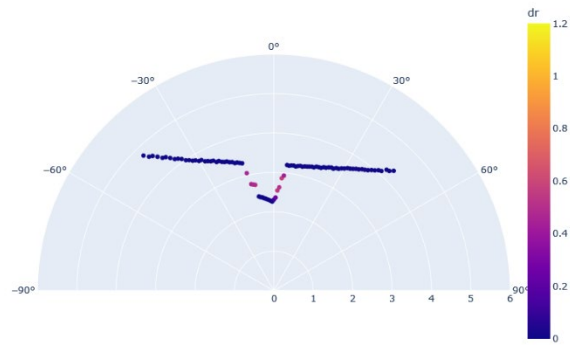
4.3. Experimento 3

Para este experimento se tuvo el escenario A (ver Figura 28) y el objeto dinámico robótico, (ver Figura 27[A]). En este experimento se realizan pruebas para conocer el comportamiento de la desviación estándar (dr) de los puntos ante los cambios generados por la inmersión del objeto dinámico en la escena. Para esta prueba se mantiene el valor de n igual a 3 y sobre esta misma cantidad de ciclos se obtiene la desviación estándar.

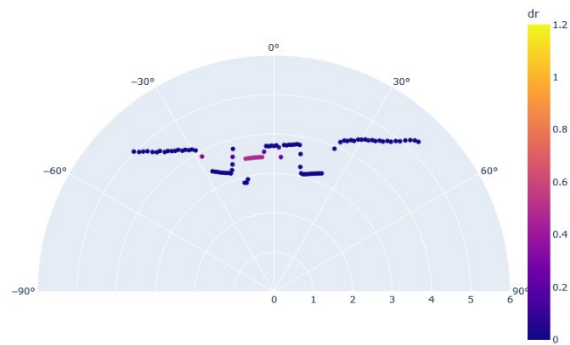
En la primera prueba se contó con la escena estática A, la cual por sus características presenta valores entre 0.000798m y 0.042309m de dr (ver Figura 34 [A]). En la segunda prueba se contó con la escena estática y se tuvo la presencia del objeto dinámico robótico, este al realizar el paso por la escena, genera alteración de los puntos del LIDAR, esto se ve reflejado en la dr , la cual aumenta a un máximo de 0.572288m en los puntos frontera del objeto dinámico (ver Figura 34[B]). En la tercera prueba se contó con la escena estática B, la cual, por sus características presenta algunos valores entre 0.001595m y 0.490999m de dr (ver Figura 34 [C]). En la cuarta prueba se contó con la escena estática B y se tuvo la presencia del objeto dinámico robótico, este al realizar el paso por la escena genera alteración de los puntos del LIDAR, esto se ve reflejado en la dr , la cual aumenta en los puntos frontera a un máximo de 1.053558m del objeto dinámico (ver Figura 34 [D]).



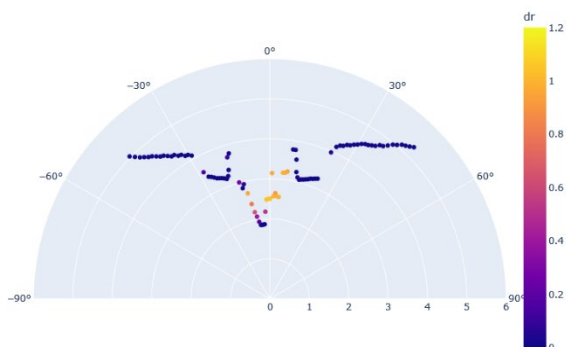
A



B



C



D

Figura 34. Valor de la desviación estándar (dr). (A) Escena estática A. (B) Escena estática A con objeto dinámico. (C) Escena estática B. (D) Escena estática B con objeto dinámico.

4.4. Experimento 4

Al ser un sistema de ejecución en línea para así lograr la detección y eliminación del objeto dinámico, fue necesario realizar una reducción de la densidad de puntos aplicando el *voxel grid*, con el fin de ajustar los tiempos de ejecución del sistema Rec-HV. Para estandarizar las pruebas se mantuvo los sensores en la misma posición, con visión hacia la escena estática B (ver Figura 28), como parámetro de ángulo de segmentación se le proporciono de 30° a 60° y no se tuvo tránsito de objeto dinámico.

Para definir la capacidad de reducción del tiempo de segmentación, se realizó un una primera prueba de referencia donde se procesó la PCL sin aplicar el *voxel grid* esta mostró un tiempo promedio de ejecución de 4.21s, dado este valor de referencia se realizaron pruebas con 12 valores de VX, desde 0.05m hasta 0.16m de arista, para los cuales cada valor del *voxel grid* se presenta el promedio de tiempo de ejecución de 100 ciclos, el promedio de tiempo de segmentación de 100

ciclos y el promedio de la PCL obtenida por la aplicación de cada valor del *voxel grid* de 100 ciclos (ver Figura 35).

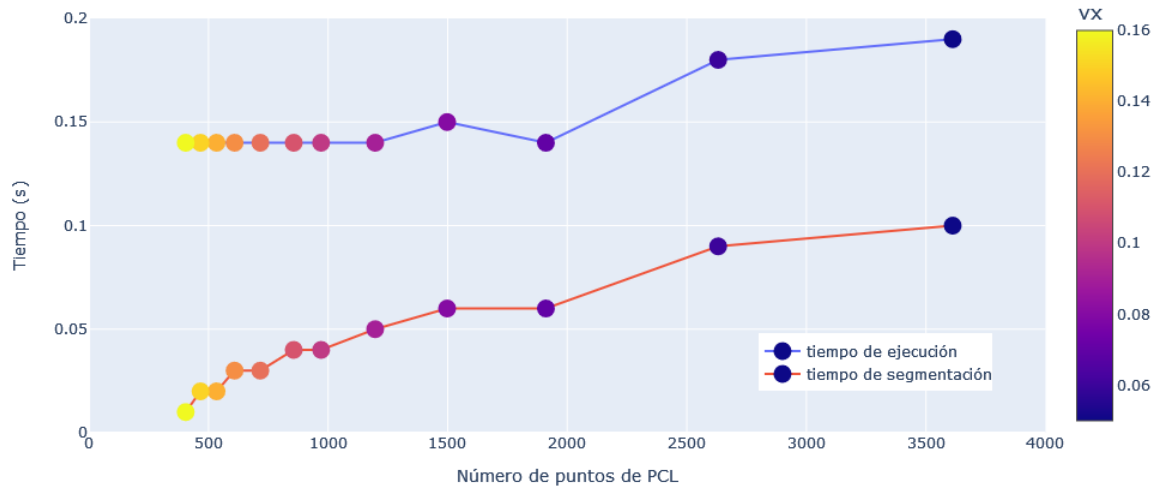
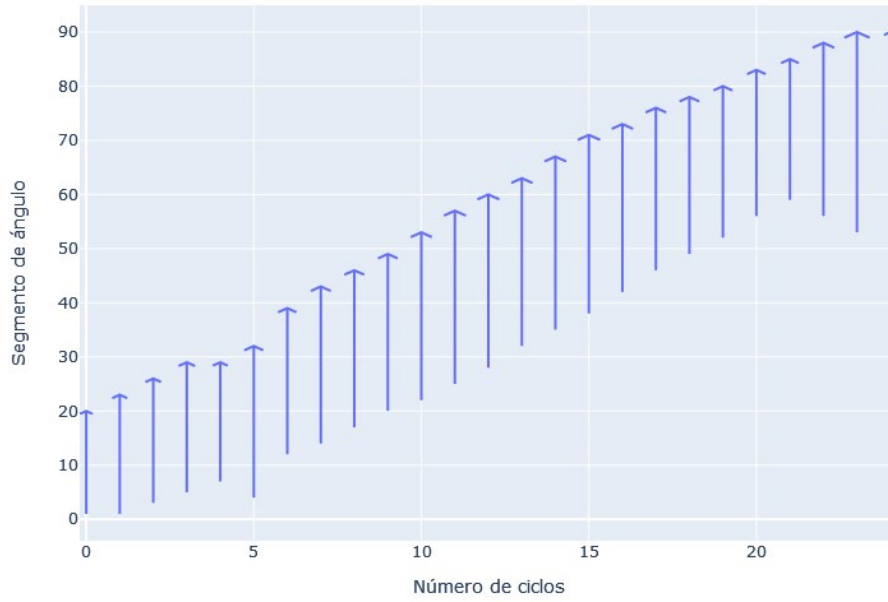


Figura 35. Tiempo promedio de ejecución, tiempo promedio de segmentación, tamaño promedio de la PCL para cada valor de *voxel grid*.

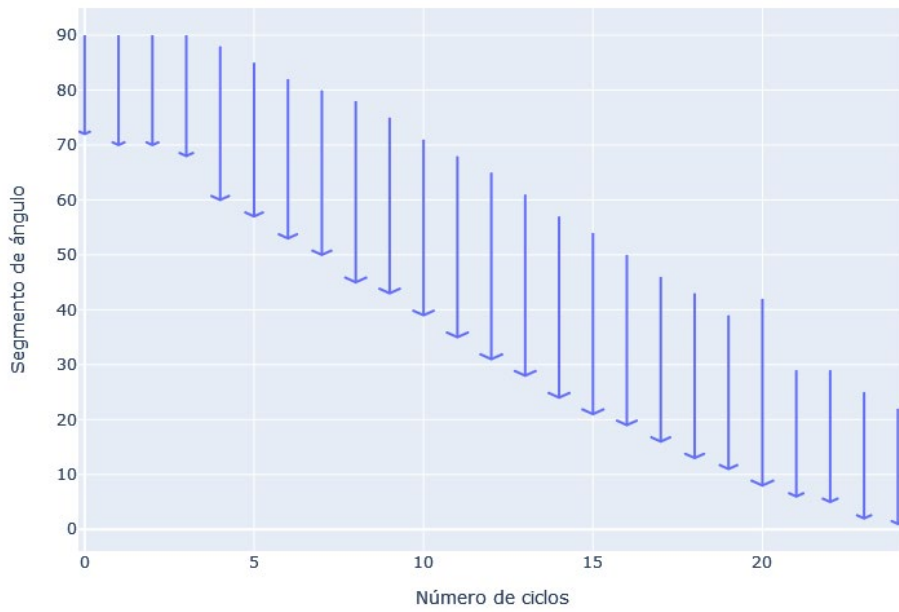
4.5. Experimento 5

Para este experimento se contó con el escenario B (ver Figura 28) y el objeto dinámico robótico (ver Figura 27 [A]). En este el objeto dinámico realizó un recorrido, lo que produce alteración de la *dr* que da como resultado las zonas de puntos de alta variabilidad, con lo cual se obtiene los ángulos de segmentación para cada ciclo de ejecución, permitiendo así, realizar el seguimiento del objeto dinámico.

Esta zona fue obtenida para cada ciclo de las pruebas, en la primera prueba se presentó la detección progresivamente del objeto dinámico, mientras este va cruzando la escena en el 100% de los ciclos donde estuvo presente, con un recorrido de izquierda a derecha compuesto por 23 ciclos (ver Figura 36[A]). También se realizó la prueba con el recorrido de derecha a izquierda compuesto por 25 ciclos, en la que también se presenta la detección progresiva del objeto dinámico en el 100% de los ciclos (ver Figura 36 [B]).



A



B

Figura 36. Seguimiento del ángulo segmentado en cada ciclo donde fue detectado el objeto dinámico, (A) recorrido de izquierda a derecha del objeto dinámico, (B) recorrido de derecha a izquierda del objeto dinámico.

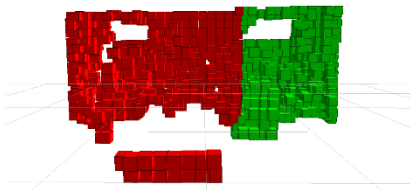
4.6. Experimento 6

En el experimento se aplicaron las dos etapas de segmentación y se realizó un seguimiento frame a frame del objeto dinámico, en este se utilizó el escenario B (ver Figura 28) y el objeto dinámico robótico (ver Figura 27 [A]). Este realizó recorridos de derecha a izquierda, cruzando el escenario de forma paralela al eje y (ver Figura 2 [A]).

En la prueba se obtuvo la variación de la PCL estática y dinámica, aplicando una sola etapa de segmentación, dada por el ángulo de detección (ver Figura 36), a partir de este se segmentó los puntos de la zona dinámica donde se encontraba el objeto dinámico PCL_{ZD} , zona color rojo, y zona estática PCL_{ZE} , zona color verde, (ver Figura 37).



7



8



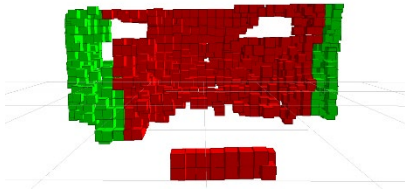
9



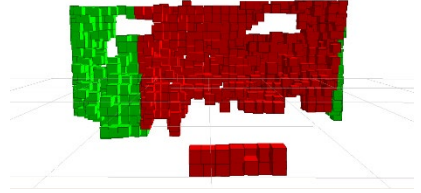
10



11



12



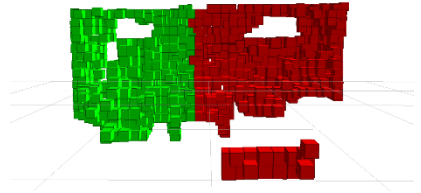
13



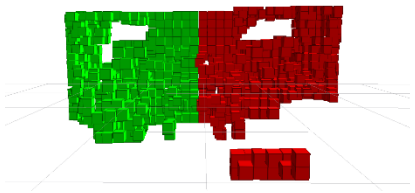
14



15



16



17



18



19



20



21



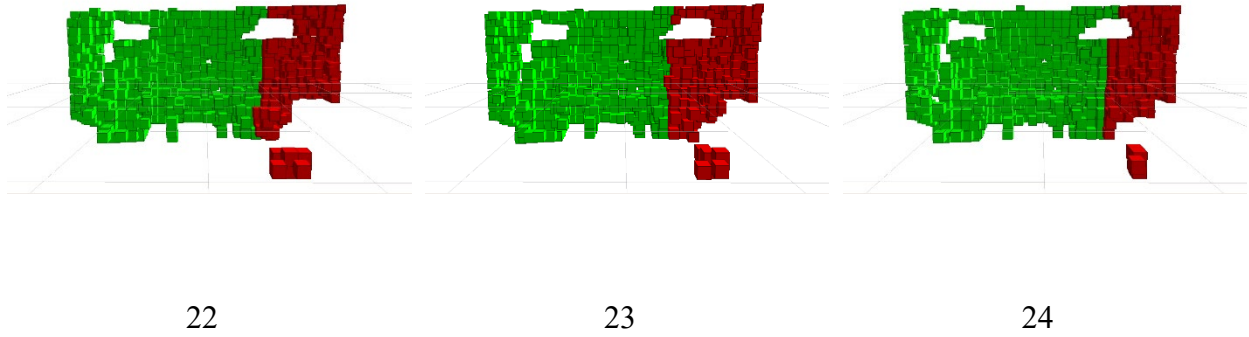


Figura 37. Aplicación del ángulo de detección sobre la PCL

La segmentación parte de la nube de puntos total, esta es de tamaño variable para cada ciclo de ejecución. A partir de esta nube y aplicando el ángulo de segmentación, se obtiene la variación de la nube puntos de la zona dinámica y zona estática (ver Figura 38).

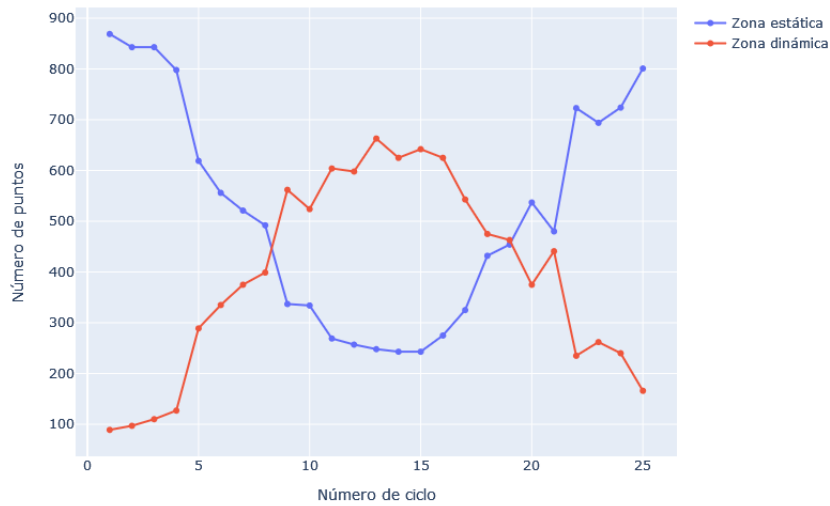


Figura 38. Variación de la nube de puntos recorrido de derecha a izquierda.

En la misma prueba se obtuvo el resultado de la segmentación por ángulo de detección, se aplicó una segunda segmentación por agrupación por profundidad, esta permite la segmentación del objeto dinámico con mayor exactitud (ver Figura 39).

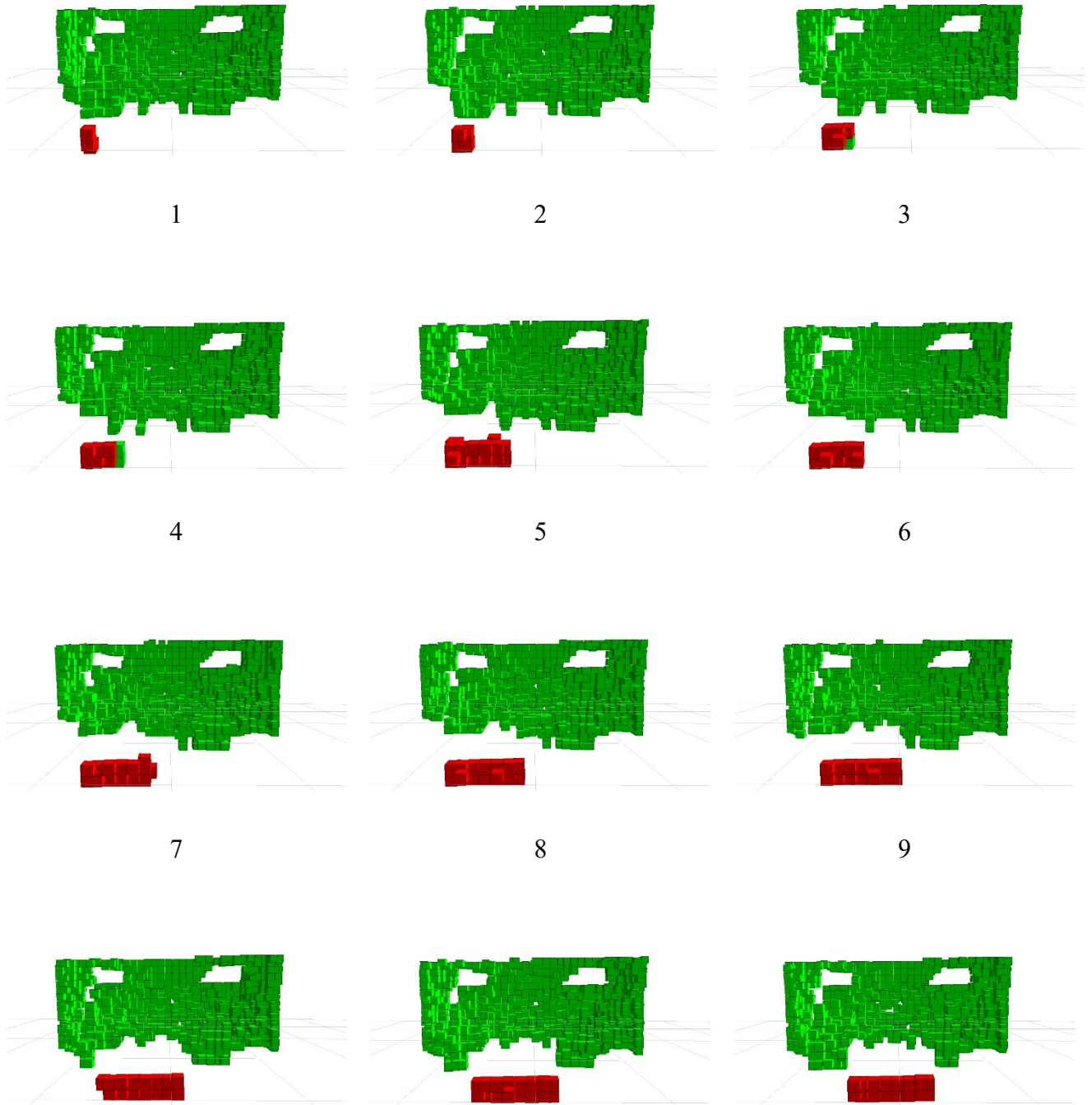


Figura 39. Seguimiento de la segmentación de objetos dinámico aplicando el método de segmentación en dos etapas.

Esta segmentación permite conocer más a fondo el número de puntos por el cual está compuesto el objeto dinámico y disminuye la pérdida de puntos que corresponden a la escena estática (ver Figura 40).

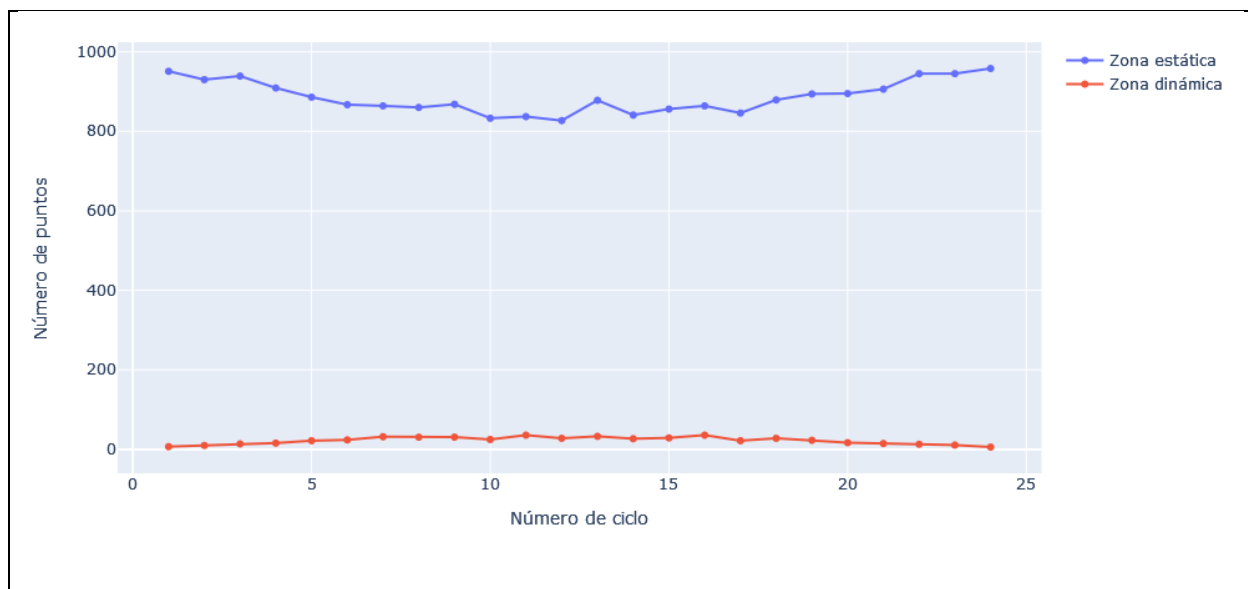
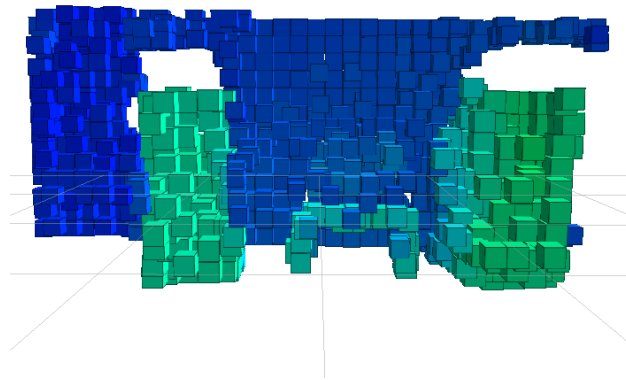


Figura 40. Variación del número de puntos por la segmentación de dos etapas.

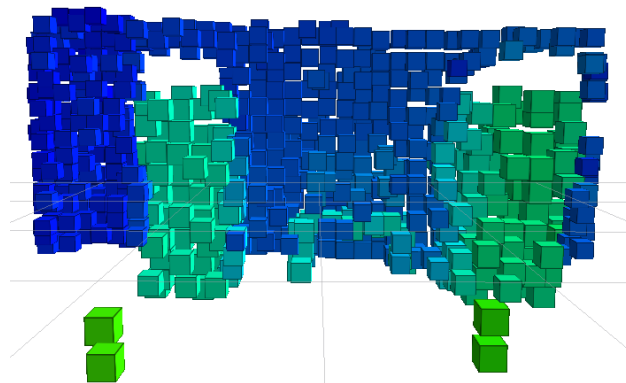
4.7. Experimento 7

El proceso de reconstrucción que se obtiene por la concatenación sucesiva de las escenas permite obtener una escena constituida por los puntos captados en los ciclos anteriores. Para este experimento se se utilizó el escenario B (ver Figura 28) y el objeto dinámico robótico (ver Figura 27 [A]). Este realizó recorridos de derecha a izquierda, cruzando el escenario de forma paralela al eje y (ver Figura 2[A]). En este experimento se realizaron 10 pruebas, en cada una se realizó la concatenación y reducción de las *PCL* consecutivas de 25 ciclos, en ellas se captó primero la escena estática sin el objeto dinámico y posteriormente se captó cada una de las escenas reconstruidas.

En estas pruebas se obtuvo la escena estática inicial como referencia (ver Figura 41[A]) y se obtuvieron las escenas reconstruidas sucesivamente para cada ciclo, dado que son un gran número de escenas se presenta una escena estática reconstruida del ciclo final de una de las pruebas (ver Figura 41[B]).



A



B

Figura 41. (A) PCL escena estática, (B) PCL escena reconstruida.

Para analizar la reconstrucción se obtuvo la correlación entre la escena estática inicial y cada una de las escenas reconstruidas que se obtuvo en cada ciclo y para cada prueba (ver Figura 42), estas en total presentaron un promedio de correlación del 86.43%.

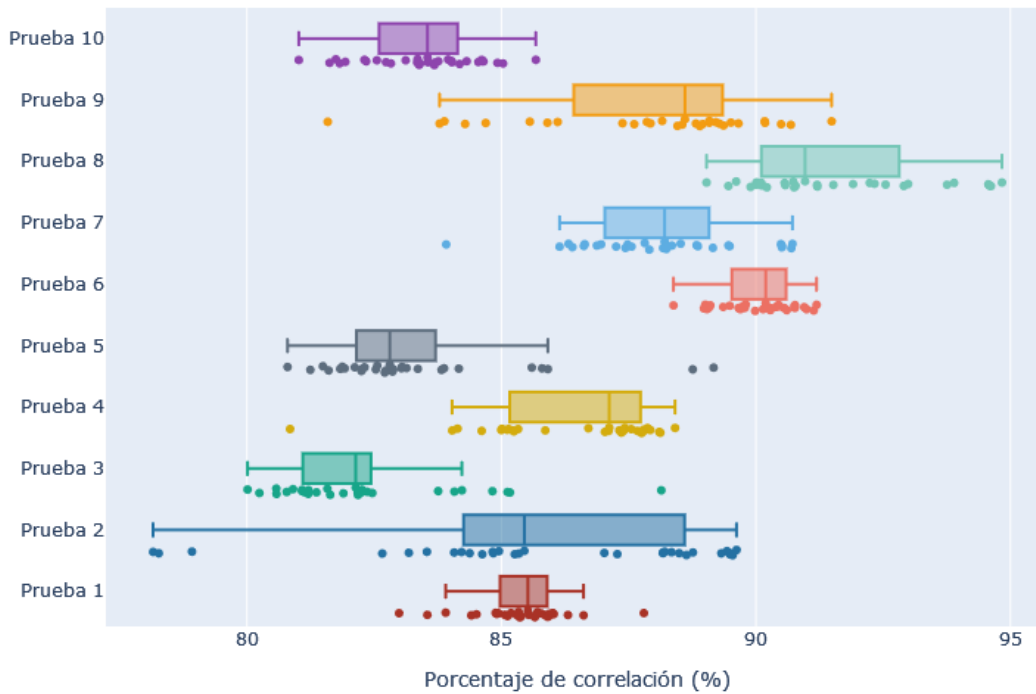


Figura 42. Variación de la correlación entre la escena reconstruida y la escena de referencia.

4.8. Experimento 8

Para este experimento se se utilizó el escenario D a la H (ver Figura 28) y como objeto dinámico un ser humano (ver Figura 27 [B]). En cada escenario se realizaron 10 pruebas, en ellas el objeto dinámico realizó recorridos de derecha a izquierda y de izquierda a derecha, cruzando el escenario sin preservar una trayectoria totalmente lineal.

En los diferentes escenarios se presentó el seguimiento constante del objeto dinámico, se presenta la segmentación por ángulo (ver Figura 43) columna segmentación por ángulo y, la segmentación de agrupación por profundidad (ver Figura 43) columna segmentación por profundidad. Estas dan como resultado una reconstrucción donde no se observa mayor perturbación generada por el paso del objeto dinámico en la escena (ver Figura 43) reconstrucción con aplicación del método. Para tener una perspectiva de comparación, se realizó la concatenación consecutiva de la nube de puntos sin aplicar el método de eliminación del objeto dinámico (ver Figura 43) columna reconstrucción sin el método.


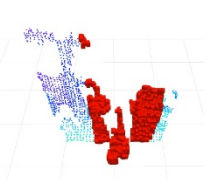
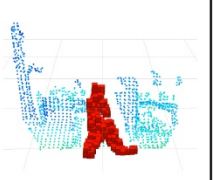
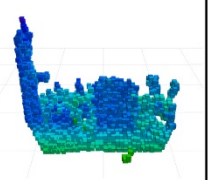
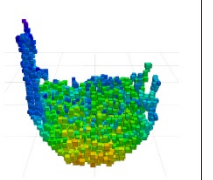

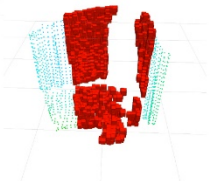
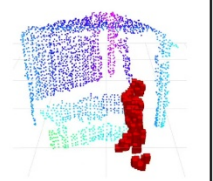
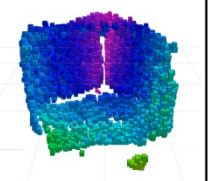
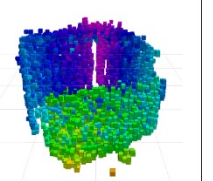

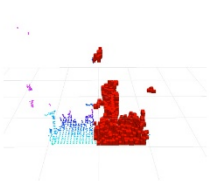
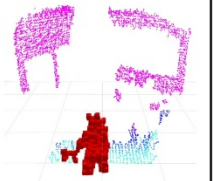
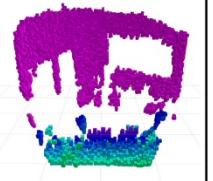
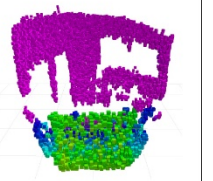

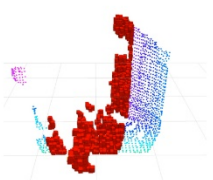
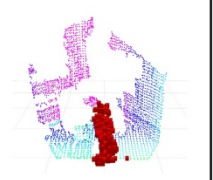
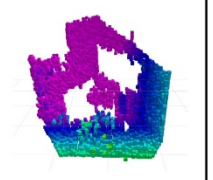
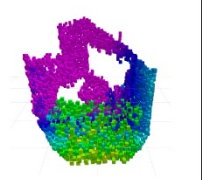

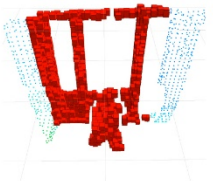
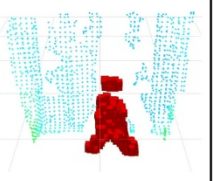

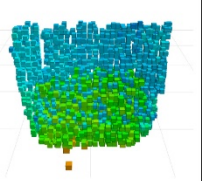
| | Escena | Segmentación por ángulo | Segmentación por agrupación por profundidad | Con Rec-HV | Sin Rec-HV |
|---|---|---|---|--|---|
| A |  |  |  |  |  |
| B |  |  |  |  |  |
| C |  |  |  |  |  |
| D |  |  |  |  |  |
| E |  |  |  |  |  |

Figura 43. En las filas escenarios son las pruebas y en las columnas se presenta el resultado de diferentes partes del proceso de reconstrucción.

Para tener un punto de comparación y conocer la reducción del tiempo de ejecución, se aplicó el proceso de segmentación y reconstrucción sin reducir la densidad de puntos, este dio como resultado un tiempo promedio de ejecución de 4.36s por ciclo, ahora nuestro software con las especificaciones seleccionadas presentó un tiempo promedio de ejecución de 0,148s. En este se procesaron entre 1658 a 5623 puntos por ciclo, lo que representó una reducción promedio de la

PCL del 98.4%. El software detectó ciclos con puntos de alta variabilidad en el 32% hasta el 64% de ciclos de la prueba (ver Tabla 5).

Al aplica la segmentación por ángulo, se obtuvo que entre el 19% al 40% de los puntos de la PCL fueron segmentados y constituyeron la PCL_{ZD} . Posteriormente al aplicar la segmentación por agrupación sobre la PCL_{ZD} , se redujo el porcentaje de puntos segmentados a entre el 4% al 8% (ver Tabla 5)

Tabla 5. Variación de los tiempos de ejecución y ciclos de segmentación y porcentaje de segmentación en cada prueba.

| Esc. | tiempo promedio de ejecución | puntos por ciclo | reducción promedio de la PCL | ciclos con puntos de alta variabilidad | segmentación por ángulo | segmentación por agrupación |
|------|------------------------------|------------------|------------------------------|--|-------------------------|-----------------------------|
| D | 0,144 | 1658 | 98,9% | 43% | 19% | 8% |
| E | 0,147 | 2531 | 98,5% | 32% | 14% | 5% |
| F | 0,151 | 5623 | 97,6% | 55% | 31% | 7% |
| G | 0,150 | 3429 | 98,0% | 64% | 40% | 4% |
| H | 0,149 | 1547 | 99,3% | 47% | 19% | 8% |

La aplicación del proceso de reconstrucción en cada ciclo genera una PCL_{kT} que aumenta su número de puntos, progresivamente, partiendo de una PCL inicial hasta obtener la PCL_{kT} , este proceso generó un aumento entre el 58% hasta el 115% de la PCL inicial, ver Tabla 6.

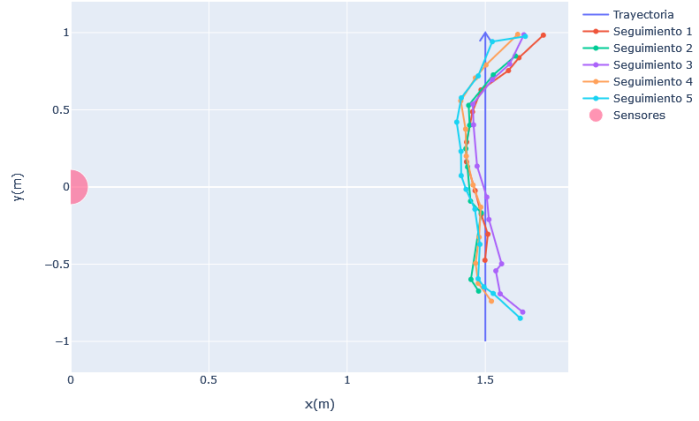
Tabla 6. Variación de la PCL al aplicar la reconstrucción.

| Esc. | PCL inicial. | PCL_{kT} | Porcentaje de incremento |
|------|-------------------|------------|--------------------------------|
| D | 1782 | 2813 | 58% |
| E | 2492 | 5349 | 115% |
| F | 6811 | 13237 | 94% |
| G | 3863 | 7411 | 92% |
| H | 1532 | 2716 | 77% |

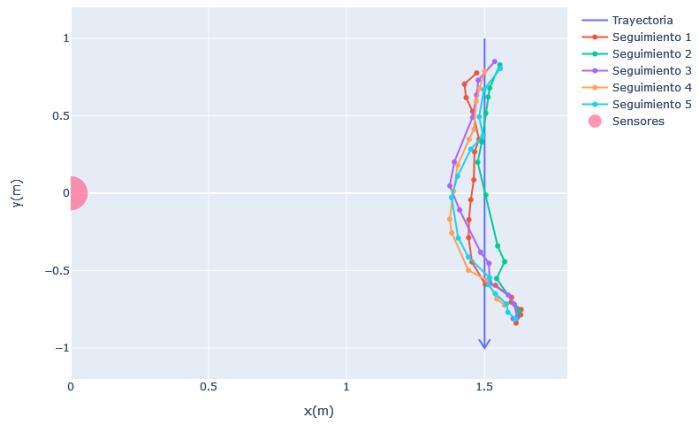
4.9.Experimento 9

Para este experimento se se utilizó el escenario B (ver Figura 28) y como objeto dinámico robótico, (ver Figura 27 [B]). Se realizaron 15 pruebas en las cuales se tuvo un objeto dinámico que cruzó la escena con diferentes recorridos. Al determinar los puntos que pertenecen a la PCL del objeto dinámico por medio de la aplicación de la segmentación en dos etapas, permitió obtener un centroide representativo del promedio de las coordenadas de los puntos pertenecientes a la PCL del objeto dinámico, este centroide fue utilizado para determinar la posición del objeto dinámico.

Las pruebas se distribuyeron en 10 con un recorrido lineal perpendicular al eje y (ver Figura 2[A]) de las cuales 5 fueron en sentido izquierda a derecha (ver Figura 44[A]) y 5 de derecha a izquierda (ver Figura 44[B]). Por último, se realizaron 5 pruebas con un recorrido transversal a la escena (ver Figura 2[B] con sentido izquierda derecha (ver Figura 44[C])).



A



B

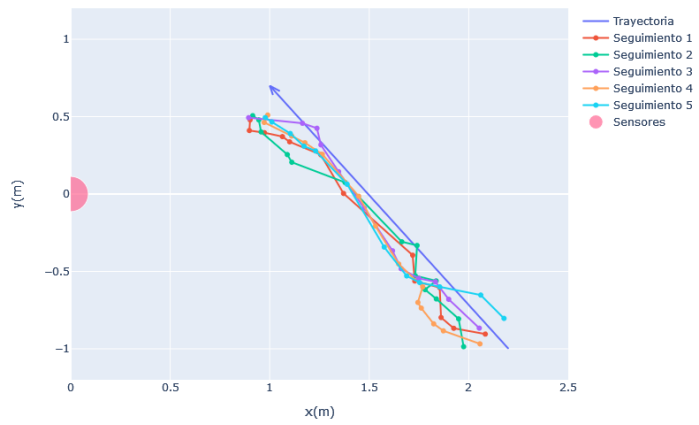


Figura 44. Seguimiento al recorrido del objeto dinámico, (A) seguimiento de 5 recorridos paralelos eje y de izquierda a derecha. (B) seguimiento de 5 recorridos paralelos eje y de derecha a izquierda. (C) seguimiento a 5 recorridos transversales del objeto dinámico.

A partir de los recorridos de la Figura 44 (A y B), se presenta el error máximo, el error absoluto y el error cuadrático medio de la medida de la profundidad, ver Tabla 7.

Tabla 7. Máximo de error, error absoluto y error cuadrático medio(A) recorridos de izquierda a derecha, (B) recorrido de derecha a izquierda

| Recorrido | Máximo error(m) | Error Absoluto(m) | Error Cuadrático Medio(m) |
|-----------|-----------------|-------------------|---------------------------|
| 1 | 0,2099 | 0,0614 | 0,0848 |
| 2 | 0,1088 | 0,0535 | 0,0592 |
| 3 | 0,1396 | 0,0562 | 0,0701 |
| 4 | 0,1171 | 0,0459 | 0,0562 |
| 5 | 0,1445 | 0,0624 | 0,0756 |
| Promedio | 0,1440 | 0,0559 | 0,0692 |

A

| Recorrido | Máximo error(m) | Error Absoluto(m) | Error Cuadrático Medio(m) |
|-----------|-----------------|-------------------|---------------------------|
| 1 | 0,1323 | 0,0628 | 0,0726 |
| 2 | 0,1222 | 0,0447 | 0,0599 |
| 3 | 0,1261 | 0,0658 | 0,0778 |
| 4 | 0,1264 | 0,0603 | 0,0728 |
| 5 | 0,1199 | 0,0568 | 0,0686 |
| Promedio | 0,1254 | 0,0581 | 0,0704 |

B

4.10. Experimento 10

Para esta prueba se utilizó el escenario B (ver Figura 28), como objeto dinámico robótico (ver Figura 27 [B]) y el montaje sobre rieles (ver Figura 26[B]). Para observar la respuesta de ORB-SLAM2, realizaron recorridos lineales de los instrumentos, los cuales se encontraban sobre unos rieles para garantizar un recorrido lineal. La medición del posicionamiento se realizó mediante dos sistemas, uno de referencia, donde se aplicó un sistema de odometría mecánica con un sensor de desplazamiento lineal, el cual permitió medir el desplazamiento en el eje x y se aplicó odometría visual a partir de la incorporación de ORB-SLAM2 (ver Figura 45).

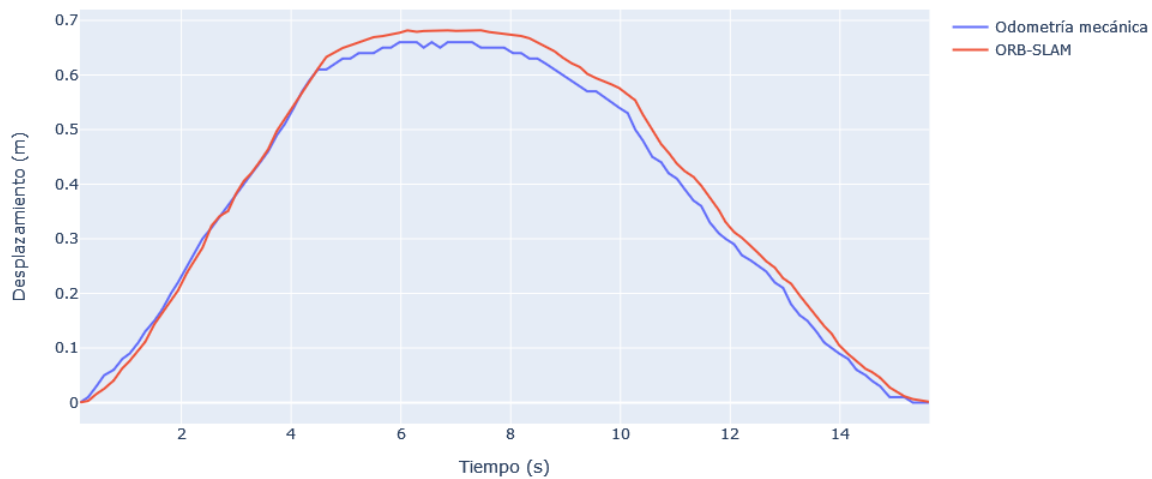
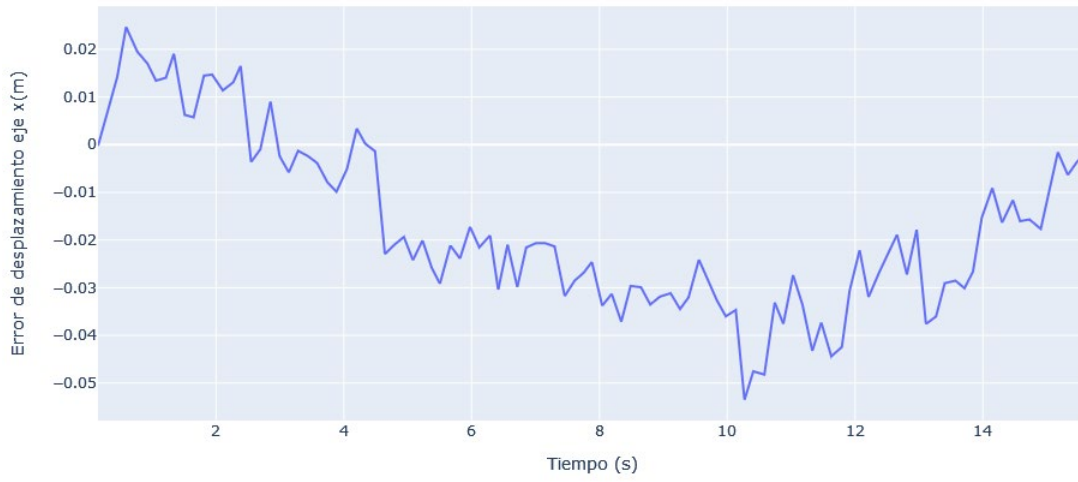
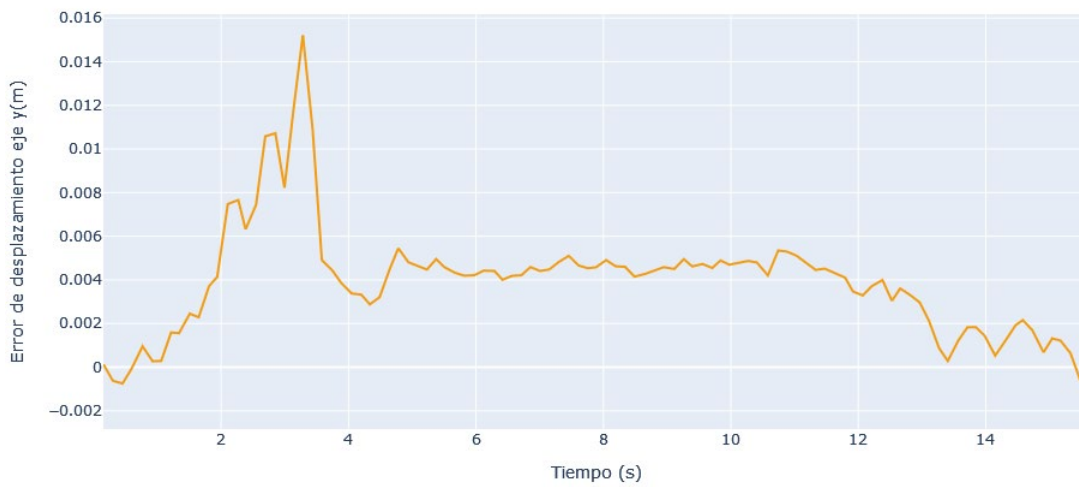


Figura 45. Desplazamiento registrado por el sistema de odometría mecánica y odometría visual.

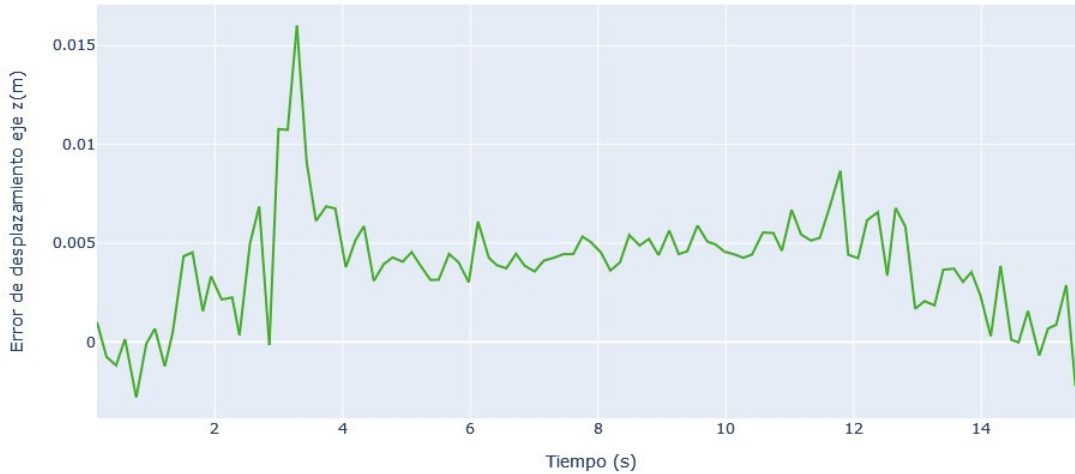
Este movimiento generó respuesta de ORB-SLAM en los tres ejes y dado que solo se realizó movimiento en el eje x , sin embargo, los tres ejes reportaron variación, se presenta el error de posicionamiento en el eje x (ver Figura 46[A]). El error de posicionamiento en el eje y (ver Figura 46[B]). El error de posicionamiento (ver Figura 46[C]).



A



B



C

Figura 46. Error de desplazamiento por diferencia entre el sistema de odometría mecánica y el sistema de odometría visual (A) eje x, (B) eje y, (C) eje z.

En el total del recorrido, se recolectaron 183 datos con un tiempo de ejecución aproximado de 15s y un número de 100 ciclos. Se muestra el máximo error presentado, el error absoluto y el error cuadrático medio del posicionamiento de los instrumentos para cada eje, (ver Tabla 8).

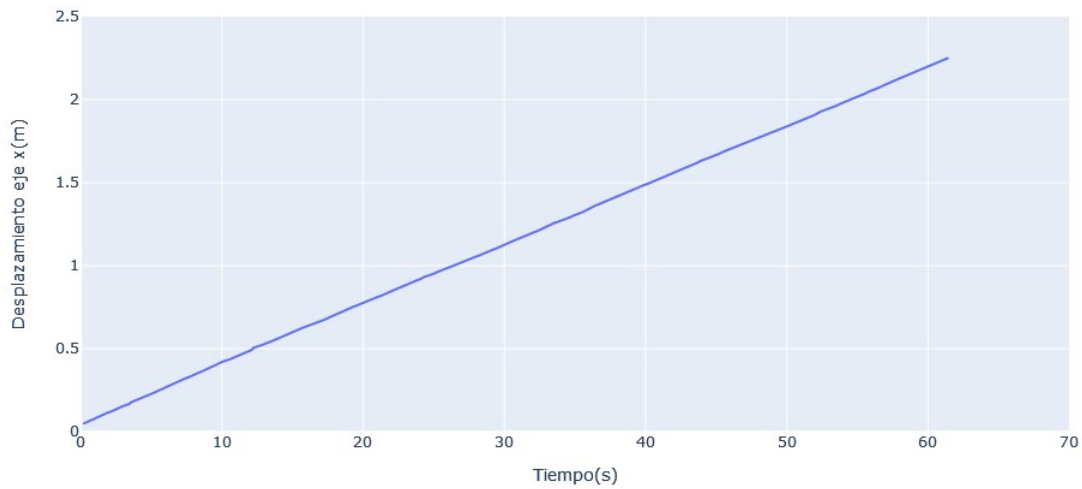
Tabla 8. Máximo error, error absoluto y error cuadrático medio del posicionamiento de los instrumentos.

| Eje | Máximo error(m) | Error Absoluto(m) | Error Cuadrático Medio(m) | Desviación Estándar (m) |
|-----|-----------------|-------------------|---------------------------|-------------------------|
| x | 0.0535 | 0.0124 | 0.0188 | 0.0176 |
| y | 0.0152 | 0.0026 | 0.0036 | 0.0026 |
| z | 0.0160 | 0.0026 | 0.0037 | 0.0027 |

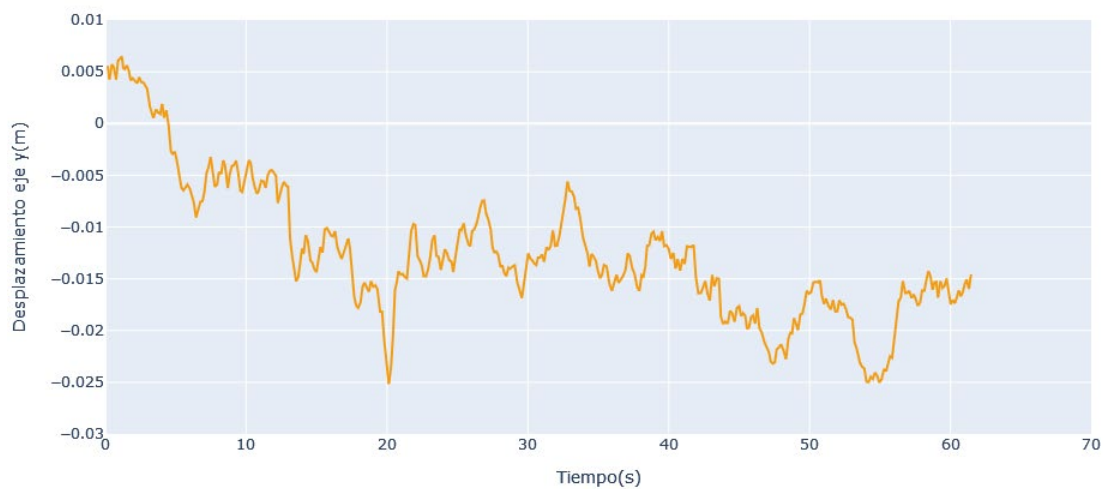
4.11. Experimento 11

Para esta prueba se utilizó el escenario C (ver Figura 28) y como objeto dinámico al humano (ver Figura 27[B]), y el montaje sobre plataforma móvil (ver Figura 26[C]). Se realizó el seguimiento

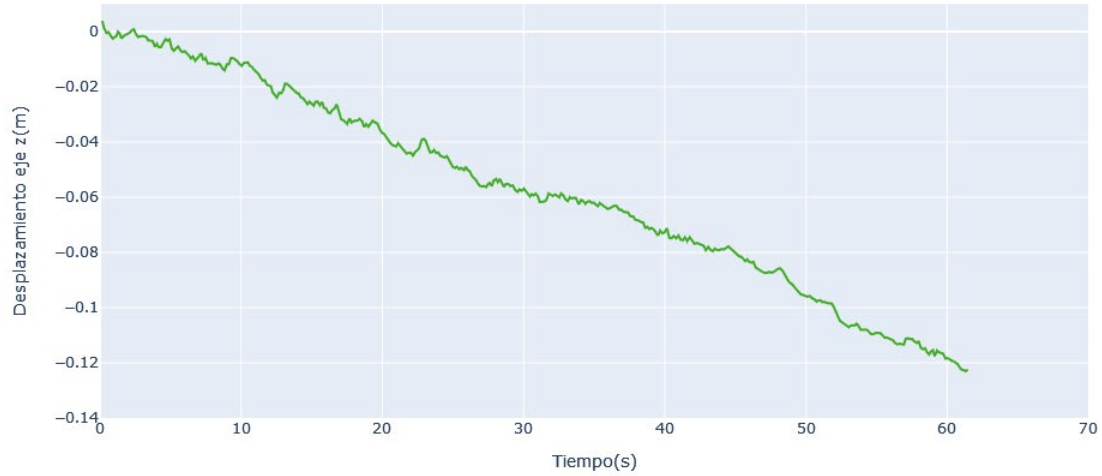
del posicionamiento de la plataforma, a través de las coordenadas de traslación y rotación suministradas por ORB-SLAM2. La plataforma móvil realizó un recorrido lineal sobre el eje x (ver Figura 47[A]). Para conocer más a fondo el comportamiento del sistema de odometría visual durante el recorrido realizado, se muestra el resultado del seguimiento de la variación del eje y (ver Figura 47[B]) y del eje z (ver Figura 47[C]).



A



B



C

Figura 47. Variación de la posición al realizar un recorrido, (A) variación eje x, (B) variación eje y, (C) variación eje z.

La prueba se ejecutó en 407 ciclos y presentó un tiempo promedio de ejecución de 151ms, lo que constituyó un tiempo total aproximado de 109 segundos. Se tuvo como resultado que la plataforma registró una velocidad promedio de 0.036m/s, esta velocidad se registró por medio de la variación del posicionamiento suministrado por ORB-SLAM (ver Figura 48).

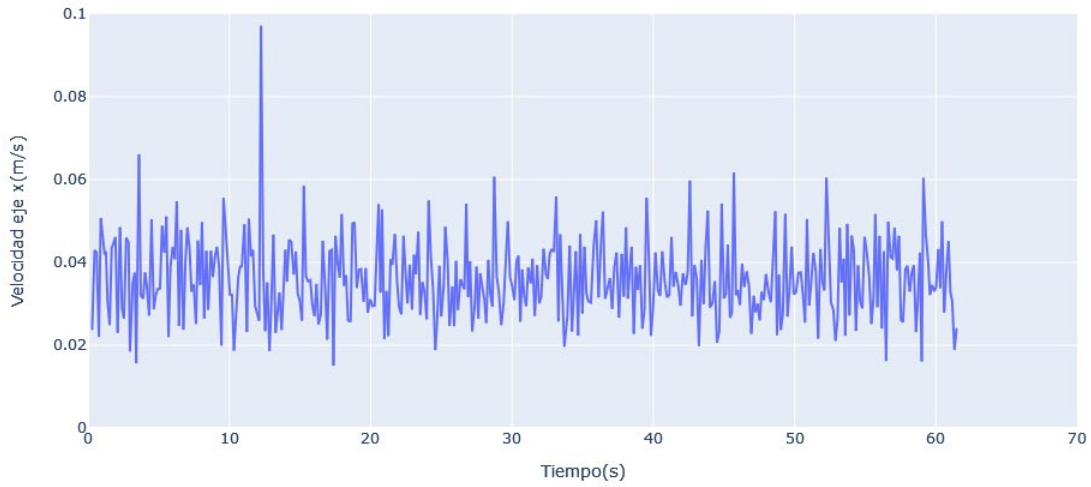


Figura 48. Variación de la velocidad durante el desplazamiento.

Dado que el desplazamiento se está realizando bajo condiciones reales, esto implicó tener en cuenta la variación de las rotaciones generadas por el movimiento de la plataforma, se tiene como resultado la variación de los tres ángulos de Euler que comprenden parte de la matriz de transformación homogénea, α (línea color azul), β (línea de color naranja) y γ (línea de color verde). (Ver Figura 49).

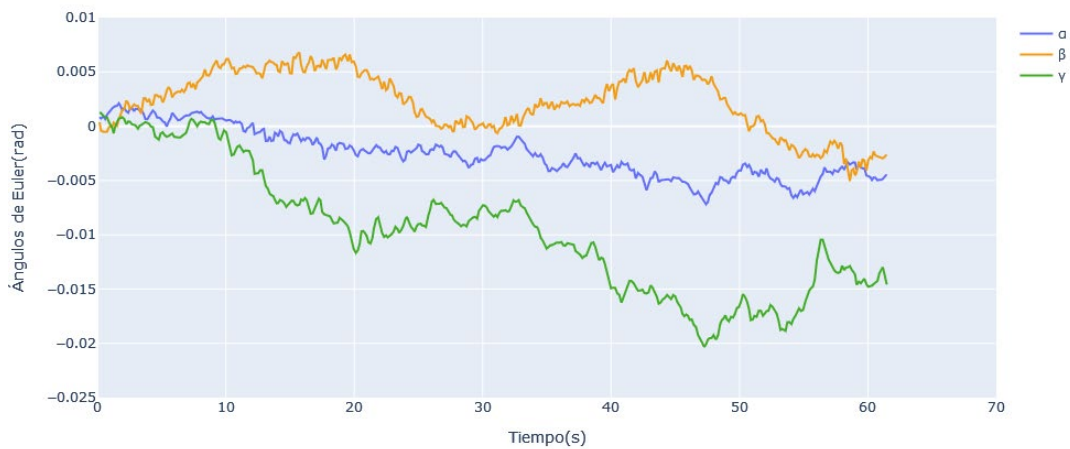
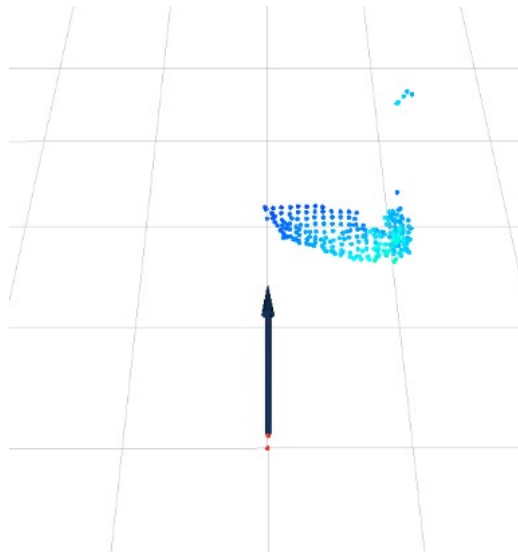


Figura 49. Variación de los ángulos de Euler.

Dado que se pudo obtener el desplazamiento en todo momento del recorrido, fue posible reflejar en la simulación a través de RViz, el resultado de este se muestra como puntos de colores que inician de color rojo (ver Figura 50[A]) y terminan de color verde (ver Figura 50[B]), lo cual simboliza el aumento de la profundidad, además se presenta con una flecha la posición y orientación de los sensores, en la posición inicial (ver Figura 50[A]) y final del recorrido (ver Figura 50[B]), dado que el posicionamiento durante el recorrido se da en 3D se presenta la vista con respecto al plano XZ (ver Figura 50[C]).



A

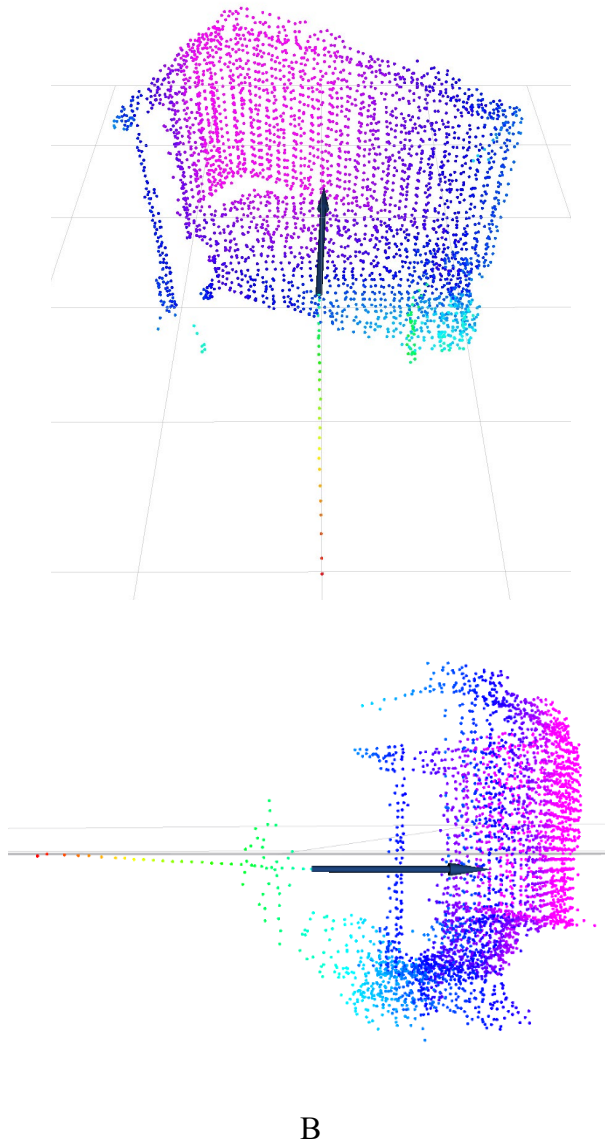
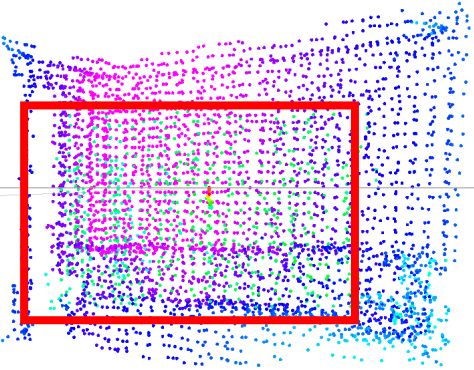
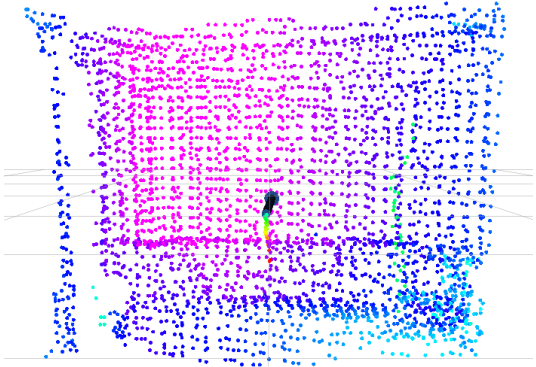
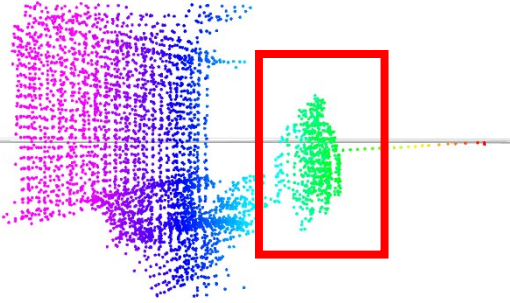
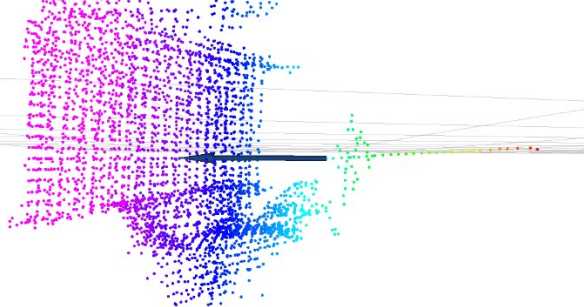


Figura 50. Escena reconstruida y recorrido de la plataforma con los sensores. (A) Plano XY. (B) Plano XZ.

Ya que el movimiento de la plataforma genera una alteración en los puntos detectados por el LIDAR, se obtuvieron puntos de alta variabilidad en el 83% de los ciclos, esta afecta directamente el número de puntos en la zona estática de cada ciclo, sin embargo, se logró obtener una reconstrucción del ambiente estático. Para observar el resultado del proceso de detección y eliminación del objeto dinámico, se proporcionan vistas de la

reconstrucción sin la aplicación del método, en él se enmarca de color rojo las zonas donde se encuentran los puntos correspondientes al objeto dinámico (ver columna sin la aplicación del método de eliminación de objetos dinámicos [Figura 51]). También se proporcionan las vistas de comparación del resultado de la reconstrucción con eliminación de objeto dinámicos, (ver columna Con la aplicación del método de eliminación de objetos dinámicos [Figura 51]).

| | Sin la aplicación del método de eliminación de objetos dinámicos | Con la aplicación del método de eliminación de objetos dinámicos |
|---|---|--|
| A |  |  |
| B |  |  |

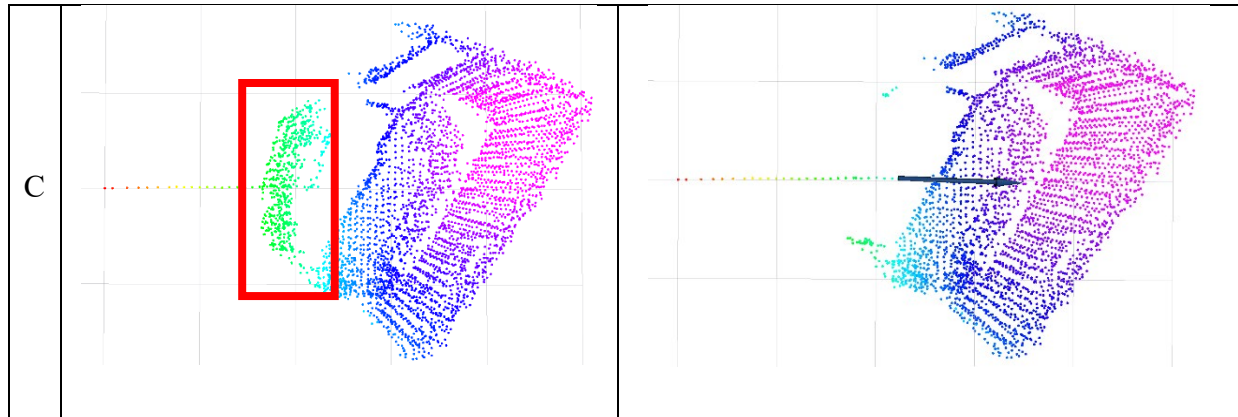


Figura 51. Comparación de reconstrucción sin aplicar el método de eliminación de objetos dinámicos y con aplicación del método de eliminación. (A) vista frontal, (B) vista lateral, (C) vista superior de la reconstrucción.

Dado que la reconstrucción se basa en la captura sucesiva de nuevos segmentos de la escena, se tuvo como resultado el aumento progresivo del número de puntos de la PCL de reconstrucción, en este se presentó que la PCL inició con 188 puntos y finalizó con 3500 puntos, este aumento presenta dos zonas: una primera desde el tiempo 0 hasta el tiempo 42 segundos, la cual presenta un aumento progresivo y, una segunda etapa del tiempo 42 en adelante, donde no presenta una variación significativa (ver Figura 52).

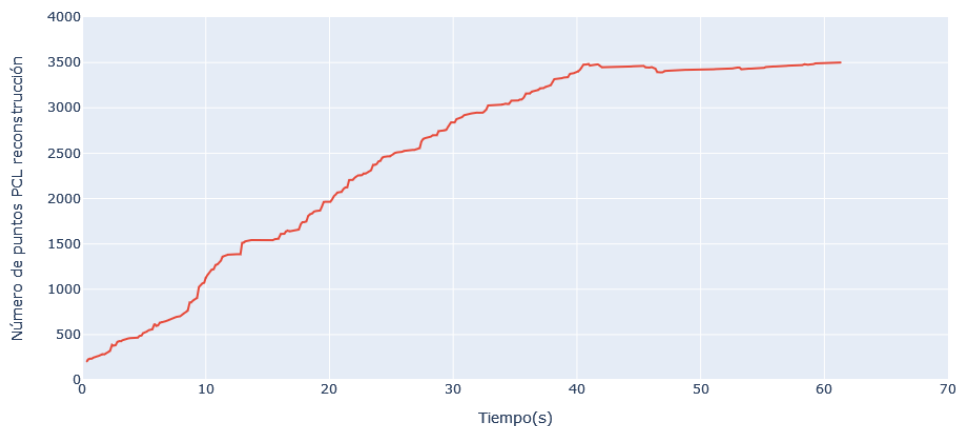


Figura 52. Variación del número de puntos de la PCL de reconstrucción.

Para obtener la correlación, se obtuvo una PCL patrón a partir de la selección de las PCL, donde se tuvo solo la escena estática, a partir de estas se realizó la concatenación sucesiva para reconstruir la escena estática, sin embargo, para generar variaciones en la reconstrucción de la escena se realizó de forma selectiva la reducción de la nube de puntos, esto dio como resultado una PCL de 4526 puntos. Se obtuvo la correlación para la PCL de cada ciclo con respecto a la PCL patrón, puesto que el número de puntos va aumentando en la PCL reconstruida, la similitud entre la escena reconstruida con la presencia de objeto dinámico y la reconstrucción de la escena estática va aumentando también. Esta reconstrucción de la nube de puntos de la escena estática que se genera cada ciclo va aumentando el porcentaje de correlación entre las dos reconstrucciones, la evolución se muestra en la Figura 53.

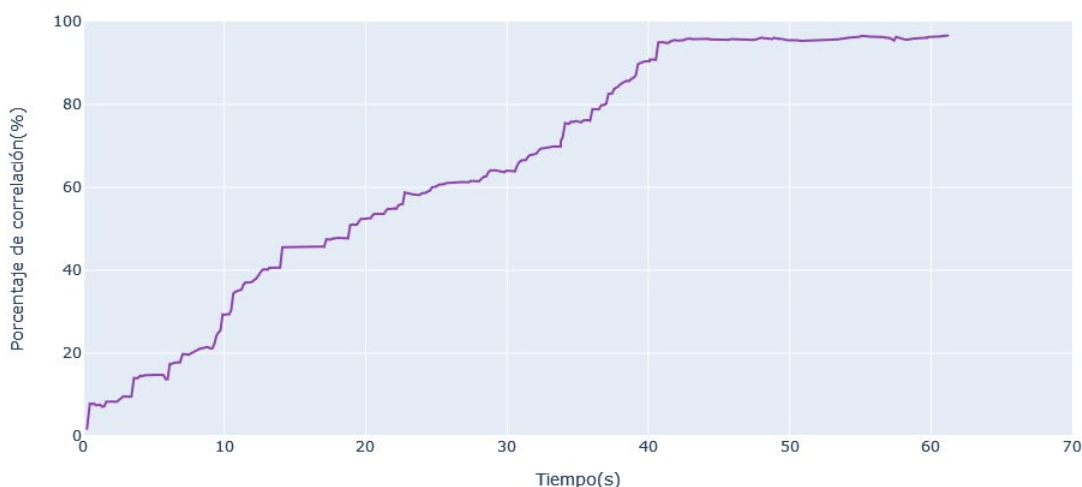


Figura 53. Variación de la correlación en la reconstrucción durante un recorrido.

Para abordar con más profundidad la comprensión de los resultados se deja un repositorio, donde podrá encontrar videos, *dataset* y software relacionado con este desarrollo, ver Anexo 3. Repositorios.

Capítulo 5. Discusión de los resultados

5.1. Experimento 1

Este primer experimento permitió realizar un reconocimiento del comportamiento de los puntos del LIDAR, en este se observa cómo los puntos ante superficies sin mayor complejidad como la del escenario A, tiene la capacidad de presentar de manera constante el set de valores completos, sin embargo, cuando es colocado en ambientes donde se tengan superficies con diferentes materiales y variación de la profundidad, se encuentra que existen zonas donde no es constante la captura de los puntos, si no que se presentan lugares donde no se logra determinar una profundidad como en el escenario B, el cual presenta valores inconsistentes en el 94.4% de los ciclos y con un promedio de 8 puntos con valores inconsistentes por ciclo, aunque no se tiene un referente bibliográfico donde se presente el porcentaje de ciclos con puntos consistentes y se tiene la observación de la necesidad de aplicar métodos de tratamientos sobre los puntos del LiDAR (Androulakis, et al., 2021; Haoran, et al., 2022).

5.2. Experimento 2

Al encontrar zonas con valores inconsistentes se aplicó la interpolación y extrapolación como forma de suplir los valores no presentes. Aunque estos métodos permiten obtener un vector r completo, este vector no es representativo de lo que realmente está ocurriendo en el entorno; por ende, se debe contar con la captura de ciclos previos, en los cuales existe información real del entorno que ayude a mitigar la pérdida de puntos.

En el promedio móvil la información previa atenúa la variabilidad de los datos y disminuye el error por la pérdida de puntos; sin embargo, al explorar los diferentes valores de n del promedio móvil (ver Figura 32) se obtuvo que cuando el objeto dinámico cruza la escena en un ambiente estático, el n afecta el registro del objeto dinámico, cuando el n tiene un valor muy alto, se afecta la posición

y forma del objeto drásticamente, esto ocurre con valores entre 5 y 7. Ahora con n igual a 2 se presenta una poca de distinción de los puntos de alta variabilidad, que se presentan en la frontera del objeto dinámico. Con n igual a 4 o 3 se tiene una buena visión del objeto dinámico y se observa, de forma más clara, los puntos que de alta variabilidad en zonas donde se presentó cambio de profundidad dado por el recorrido del objeto dinámico, estas zonas de alta variabilidad permiten distinguir la zona dinámica, considerando que entre más alto el promedio menor debe ser la velocidad del objeto, siendo este un factor de gran importancia para esta investigación al igual que como ya se ha presentado previamente (Wu, et al., 2019; Cordeiro & Pedrino, 2019; Wang, et al., 2020). En dinámico se seleccionó un promedio móvil de 3 ciclos, en la Figura 33 se presenta como la aplicación de estos métodos estadísticos permite recuperar los segmentos de puntos que se encontraban en la lectura logrando así obtener los puntos de color rojo los cuales son consistentes con el corte transversal del entorno.

5.3.Experimento 3

Dado que el tratamiento estadístico permitió dilucidar el comportamiento de los datos del LiDAR, se profundizó en el estudio sobre los n ciclos implicados en él. La dr de los valores albergados en el vector r es indicador de la variabilidad de la lectura de los puntos del LiDAR. En la Figura 34 se mostró el comportamiento de la dr en una escena con diferentes características, en esta se resalta como aun a pesar de las diferentes escenas la presencia del objeto dinámico generó que se aumente el valor de la desviación estándar, específicamente en los puntos frontera del objeto dinámico dado que estos puntos son los que se encuentren en la transición del cambio de profundidad, este comportamiento permite localizar la zona dinámica, aun cuando no se conozca la forma de velocidad u otras características del objeto dinámico.

5.4.Experimento 4

Dado la posibilidad de la ejecución continua en línea del sistema, se realizó las pruebas de detección. En estas se define un ángulo de segmentación en cada ciclo a partir de las zonas de alta

variabilidad detectadas sobre los puntos del LiDAR, en la Figura 36 se muestra el seguimiento del objeto dinámico en cada cuadro de la prueba, en él se observó cómo se logra realizar una detección de forma consistente desde el momento que ingresa el objeto dinámico a la escena hasta cuando termina el recorrido, solo en algunos ciclos de los dos recorridos se aumenta la apertura del ángulo, lo cual es resultado de las alteraciones, generadas por los objetos estáticos en la escena, con ello se muestra que el sistema tiene la capacidad de realizar una detección continua en el 100% de los *frame* para así generar la reconstrucción de la escena, en contraste se reporta un porcentaje del 95.9%. Rec-HV (Wu, et al., 2019), presenta un promedio de ejecución de 7Hz que es el valor de muestreo del LiDAR, esto contrastado con la literatura en donde se presentan sistemas a 2Hz en línea (Wu, et al., 2019), y 10Hz fuera de línea (Wang, et al., 2020).

5.1. Experimento 5

En cada ciclo se estima el ángulo sobre el cual se encuentra el objeto dinámico, esto obliga a que el LiDAR sea el instrumento que rige los tiempos de ejecución y coloca como restricción que todo proceso que se vaya a ejecutar se encuentre limitado al intermedio entre cada lectura del LIDAR. Por otro lado, se tiene que el RGB-D presenta una nube de puntos que por su tamaño implicaría un tiempo de ejecución de 4.21s, según lo obtenido en la primera prueba del experimento 4, al aplicar el voxel grid se observó que este valor disminuyó considerablemente (ver Figura 35), en esta grafica se observa que el tiempo de ejecución no logra disminuir más allá de los 142ms aproximadamente, dado que este es el tiempo de ejecución del LIDAR, sin embargo, el tiempo de segmentación si disminuye al igual que el tamaño de la PCL.

Al observar el comportamiento del sistema se seleccionó un Voxel Grid de 0.08m, el cual permitirá la ejecución en línea del sistema Rec-HV y, aunque, no es un rango límite para el aumento de los tiempos de ejecución bajo estas condiciones, si brinda un margen de tolerancia en la ejecución dado que se realizarán pruebas en otros escenarios con mayor profundidad y complejidad.

5.2.Experimento 6

Una vez definido el ángulo de segmentación fue posible obtener la PCL_{ZE} y la PCL_{ZD} estas son las PCL obtenidas en el primer proceso de segmentación, como se muestra en la Figura 37, al observar esta secuencia de imágenes se observa cómo el objeto dinámico siempre se encuentra casi en su totalidad dentro de esta zona roja, a excepción de la imagen 3 y 4 donde se observa que algunos puntos del objeto dinámico se quedan por fuera de la zona roja.

Ahora al observar la Figura 39 donde se aplicó la segmentación en dos etapas sobre la misma prueba, se observa cómo esta segunda etapa logra segmentar con mayor precisión el objeto dinámico, obteniendo así la PCL_{Din} y la PCL_{Est-k} .

En esta prueba además se enmarca como el aplicar la segmentación en dos etapas permite obtener una PCL de la zona estática, donde se han recuperado una mayor cantidad de puntos (ver Figura 38 y Figura 40), en estas se muestra como con la segmentación a una etapa se llega a tener una PCL dinámica de aproximadamente 670 puntos que constituye más del 60% de la PCL total, mientras que con la segmentación a dos etapas se logra obtener PCL dinámica de hasta 40 puntos que corresponde aproximadamente al 5% de la PCL total.

5.3.Experimento 7

En este se tiene como referencia la escena estática (ver Figura 41[A]), de primera medida se tiene una escena reconstruida (ver Figura 41[B]), la cual al realizar una comparación visual se observa cómo la distribución es consistente entre las dos escenas, aun en detalles pequeños como los que correspondería a la silla (ver Figura 28[B]), sin embargo, sí se observa una diferencia en la textura que indica errores que se insertar en el proceso de reconstrucción sucesiva, donde siempre existirá un factor error de medida de los instrumentos que está implícito.

La observación realizada es corroborada con los valores de la correlación obtenida (ver Figura 42), en donde se observa la distribución de los porcentajes de correlación para cada prueba realizada, en esta se obtuvo un porcentaje promedio de reconstrucción de 86.43%, que es un valor significativo para el sistema Rec-HV. En esta también se presenta como el sistema de

reconstrucción muestra un valor coherente, siendo esta una métrica que surge de esta investigación y sobre la cual no tenemos referencia bibliográfica registrada.

5.4. Experimento 8

En las pruebas realizadas sobre diferentes escenarios, se encontró que para cada escenario se presentaron puntos de alta variabilidad, en estos puntos se tienden a presentar falsas detecciones, estos son dados por el tipo de material, el ángulo de la superficie o por bordes a espacios abiertos.

En el escenario D se tiene materiales de baja reflexión que tienden a generar problemas de detección, sin embargo, dado que en este escenario el corte sobre el plano XY de medición del LiDAR, no se encuentra sobre el espacio abierto, se observa que se logra realizar una detección consistente y para el RGB-D, el espacio abierto se convierte en una zona donde no logra generar una detección constituyendo un espacio no reconstruido, esto se ve reflejado en la reconstrucción final (ver Figura 43 fila A) y en el valor final del número de puntos de la escena donde se tiene un total de 2813, esta escena tiene como característica que fue la que reportó el menor promedio de tiempo de ejecución con 144ms por ciclo.

En escenario E se tiene un espacio de superficie constante que no presenta ninguna alteración cuando está ausente el objeto dinámico, sin embargo, el espacio abierto en el costado derecho que se encuentra dentro del plano XY del LiDAR, si constituye una zona que altera la detección, sin embargo, al observar los resultados de la prueba se tuvo que solo en el 32% de los ciclos se registró zona de alta variabilidad. Aun con esta zona se logra generar una segmentación del objeto dinámico, logrando así reconstruir la escena (ver Figura 43 fila B), esta escena se caracteriza por ser la escena que más aumentó el número de puntos con respecto a la primera PCL captada logrando aumentar el número de puntos en un 115%.

En el escenario F se presentan varios puntos de alta variabilidad, dado a que en el plano XY se genera la medición del LiDAR, se encuentran sobre los objetos estáticos que por su distribución generan puntos de alta variabilidad, en particular en esta escena se observa que en el 55% de ciclos se encontraron zonas de alta variabilidad, y que por su profundidad se establece un número de puntos final de la escena de 13237, siendo esta la escena más grande (Figura 43 fila C) a pesar de

tener una diferencia tan alta en puntos con respecto a las demás escenas el tiempo promedio de ejecución, solo estuvo 2 ms por encima del promedio general de tiempos de ejecución.

En el escenario G se presenta una zona de baja reflexión (pantalla del televisor y vidrios) que afecta tanto la medición del LIDAR con la del RGB-D, esta escena se caracteriza por presentar el mayor porcentaje de ciclos con zonas de alta variabilidad alcanzando un 64%, pero, al ingreso del objeto dinámico, se rectifica la posición y se realiza el seguimiento del objeto dinámico de forma consistente, permitiendo así, obtener la reconstrucción de la escena (ver Figura 43 fila D).

En el escenario H tiene la caracteriza que se encuentra a condición de baja luz, y con materiales de baja reflexión, aunque, se observa cómo el sistema es inherente a las condiciones, generando la detección de forma correcta y con ello la segmentación y reconstrucción de la escena (ver Figura 43 fila E). También se observa que solo en el 47% de los ciclos, se tuvo zonas de alta variabilidad que está dentro del promedio general, sin embargo, si es la zona en la cual se tiene el menor número de puntos finales en la reconstrucción, pero esta es una característica de los entornos de menor tamaño más no del sistema Rec-HV.

Esta prueba permitió conocer muy a fondo el comportamiento del sistema Rec-HV donde se observa que es más probable generar zonas de alta variabilidad por el tipo de material que, por las zonas de espacios abiertos, además se observa una gran capacidad de adaptabilidad de Rec-HV, independientemente del tamaño forma y distribución de la escena.

5.5.Experimento 9

Dado que el obtener la segmentación en dos etapas permite obtener la *PCL* del objeto dinámico, se utilizó esto como principio para obtener el seguimiento del objeto dinámico en la escena, al observar el resultado, se ve cómo se mantiene un seguimiento coherente del objeto dinámico al realizar los recorridos, sin embargo, hacia los extremos se observa que siempre se desvía del recorrido real, esto se debe a que el sistema busca obtener nubes de puntos y al llegar al final del campo visual del sensor, se toman algunos puntos de la zona estática que se encuentran en el fondo, lo que produce que se vea alterado los promedios con los cuales se obtiene el posicionamiento del objeto dinámico, en la prueba (ver Figura 44[C]) donde se realiza el recorrido transversal, se

observa una mejor consistencia hacia el lado más apartado del fondo, mostrando así, que en este sistemas es más fácil detectar al objeto dinámico cuando este se encuentra más separado de la escena estática.

A partir de los recorridos se mostraron los valores del error máximo de posicionamiento, el error absoluto y error cuadrático medio del posicionamiento en el eje x, solo se realizó en este eje, ya que, por las condiciones del experimento, en donde se trazó una ruta, no se le realizó un seguimiento punto a punto del objeto dinámico, por ende, solo conocemos la profundidad donde se estaba desplazando, más no donde se encontraba en cada instante. El error cuadrático medio reportó valores entre 5.92cm hasta 8.48 cm en contraste con la literatura donde se reporta que métodos han logran un error cuadrático de posicionamiento de entre 0.60cm hasta 3.45cm para Dynamic-SLAM y entre 0.84cm y 10.53cm para ORB-SLAM2 (Xiao et al., 2019).

5.6.Experimento 10

En la Figura 45 se mostró el resultado de la respuesta de ORB-SLAM2 con referencia al sistema de odometría mecánica. En este se muestra como ORB-SLAM2 logra obtener el posicionamiento de forma consistente al reportado por la referencia, como se muestra en la Figura 46(A), donde se tiene el error en cada instante medido, mostrando una desviación estándar de 0.0176 m.

Al observar la respuesta sobre los ejes Y y Z, se observa una serie de perturbaciones, mientras se realiza el recorrido, sin embargo, estas no presentan mayor variación en el eje Y con una desviación estándar de 0.0026m y el eje Z con una desviación estándar de 0.0027m, así mismo, al observar el error absoluto y el error cuadrático medio, se observa cómo estos tienen un valor bajo, mostrando que el sistema aun en entornos dinámicos logra mantener la consistencia en el posicionamiento.

Se mostró el resultado de la integración de la odometría visual, la cual a partir de ORB-SLAM2, permitió realizar el seguimiento del posicionamiento de los instrumentos, al realizar las pruebas comparativas entre el sistema de odometría visual y el de odometría mecánico, se observó que el error de posicionamiento es muy bajo cuando se tiene las condiciones indicadas, ORB-SLAM2 al basarse en un sistema de extracción de características puede llegar a sufrir pérdida de la posición

cuando se tiene un cambio brusco de la escena o cuando la escena en donde se encuentra tiene muy pocos elementos contrastantes y que no permite obtener puntos de referencia.

El ingresar ORB-SLAM2 a el sistema implicó que se activaran los nodos propios de él, lo que generó que se aumentara el costo computacional de procesamiento, sin embargo, dado a la estructura de nodos de ROS, este se distribuyó en los diferentes núcleos de la CPU, facilitando que no se aumentará el costo computacional de forma significativa y se logrará mantener los tiempos de ejecución del sistema.

5.7.Experimento 11

En la Figura 47 se muestra el resultado de la prueba donde se realiza el seguimiento, mientras se hacía un recorrido sobre un escenario complejo, en este se configuró un máximo de profundidad de 3m para la medición de los instrumentos, dado que en observaciones previas se notó, que el RGB-D en profundidades superiores puede llegar a aumentar el error que deteriora la reconstrucción. Por la limitación en la profundidad se tienen aportes de puntos en cada ciclo en el que se está realizando recorrido, lo que genera que se logre ir reconstruyendo la escena en cada ciclo, hasta llegar a la escena final como se muestra en la Figura 50.

Por otro lado, se observa consistencia en el seguimiento que se muestra en la Figura 50, en forma de puntos antes de la flecha de dirección, teniendo en cuenta que en el recorrido se cruzó un objeto dinámico, el cual no generó alteración en la odometría visual. En contraste se muestra la variación de la velocidad obtenida por medio de la odometría visual y el tiempo de cada ciclo mientras se realiza el recorrido, ver Figura 48, de esta se obtiene un promedio de velocidad de 0.036m/s, el cual, es el mismo valor que fue medido al colocar la plataforma y realizar un recorrido de 2m para tomar su tiempo.

De esta prueba se obtuvo una escena donde solo se observan dos zonas menores, en el que dicho sistema no tuvo la capacidad de generar una correcta segmentación (ver Figura 51), la cual puede ser dada a que el sistema trata de compensar el movimiento, sin embargo y por la variación de los puntos de LIDAR, no se logró obtener el ángulo exacto de la segmentación. A pesar de estos dos ciclos donde no se eliminó una porción de los puntos del objeto dinámico, estos resultan ser poco

significativos al comparar la perturbación completa que hubiera generado no realizar el proceso de segmentación como se observan en las imágenes de comparación.

Cuando observamos la correlación se tiene que en los dos ciclos donde no se logró una completa eliminación, fue negativo generar mayor perturbación, de la correlación general que logra obtener un valor final de 94.3%, siendo este un valor obtenido después de comparar la *PCL* obtenida en cada ciclo con una *PCL* generada a partir de ciclos de solo escena estática. Este valor de correlación final es bastante significativo ya que al observar su evolución se obtuvo una consistencia entre el aumento de los puntos con el aumento de la correlación, mostrando que estos dos valores están directamente correlacionados y que el sistema para medir la métrica realmente realiza una comparación donde se verifican las características del entorno (ver Figura 52 y Figura 53).

Capítulo 6. Conclusiones y recomendaciones

-Al observar el comportamiento de los puntos del LiDAR ante diferentes escenarios, se pueden afectar la variación de los puntos LIDAR, siendo la complejidad del entorno una de las más relevantes, se obtuvo que el tratamiento estadístico permite recuperar zonas y atenuar la variabilidad, permitiendo identificar las zonas que son alteradas por la presencia de objetos dinámicos que ingresan en la escena.

-La desviación estándar mostró que dada a su variación se puede identificar los segmentos de los datos del LIDAR que presentan alteración en su comportamiento, permitiendo identificar las zonas dinámicas donde se tiene presencia del objeto dinámico, dado que esta presenta valores de desviaciones estándar superiores 0.6m, mientras que en zonas de objetos estáticos por lo general se obtiene valores menores o iguales de 0.4m.

-Se obtuvo que, con base en la configuración de $n=3$ y el valor mínimo de dr de 0.4m, se logra obtener de forma consistente el seguimiento del objeto dinámico, siendo estos dos los únicos parámetros que son configurados para la realización de la detección a partir de los valores de la dr .

-Después de analizar los tiempos de ejecución y la distribución de la carga computacional en los nodos, se obtuvo que dependiendo de la variación del tamaño del *voxel*, se puede mantener la ejecución en línea, siendo este un parámetro que es configurado, dependiendo de la capacidad de procesamiento del equipo donde se está ejecutando el software.

-Al verificar la aplicación del ángulo de segmentación sobre la nube de puntos, se mostró que es viable aplicar los resultados de la detección sobre el LIDAR siempre y cuando los campos visuales tengan coincidencia, este mostró ser un método efectivo para realizar la reducción de costo computacional del procesamiento de detección.

-De acuerdo con los resultados expuestos de la correlación de los diferentes escenarios, donde se obtuvo un valor máximo de 94.3%, se puede concluir que el método de detección y eliminación del objeto dinámico desarrollado en esta investigación es un método efectivo, puesto que permite

eliminar la oclusión y reconstruir la escena estática, recuperando la forma, constitución y distribución de esta.

-Tras el análisis de las escenas estáticas reconstruidas, las cuales han sido procesadas por un método de reducción de puntos, se puede observar que, aunque la reducción disminuyó de forma drástica el número de puntos, no se pierden las principales características de la escena, permitiendo conocer la distribución, forma y constitución, dando soporte según lo planteado en esta investigación, en cuanto a procesar nubes de puntos reducidas para así reducir el costo computacional.

-Al analizar el comportamiento del costo computacional mientras se ejecuta el sistema, se obtuvo que no era posible ejecutar el sistema en un solo hilo, dando así la necesidad de la distribución de los procesos en diferentes nodos que permite la ejecución en diferentes hilos, los cuales son distribuidos en los núcleos del procesador, aprovechando al máximo los recursos de la arquitectura de hardware que se tiene a disposición.

-Después de analizar los diferentes resultados obtenidos en este proceso de investigación, se concluye que se está aportando a la comprensión del proceso de reconstrucción de escenas estáticas con presencia de objetos dinámicos, resaltando los aportes obtenidos por el análisis de la variabilidad en los datos de los sensores por presencia de objetos dinámicos. Esto permitió generar un nuevo método de detección, segmentación y eliminación de objetos dinámicos, el cual está plasmado en el software desarrollado, resaltando que este es de bajo costo computacional y con ejecución en línea 6.6Hz, lo que representa un aporte de los sistemas autónomos de ejecución en línea y con ello en la inmersión de sistemas robóticos en ambientes con presencia de humanos.

Anexos

6.1. Glosario y abreviaturas.

Esc.: Escenario

GPS: Sistema de Posicionamiento Global.

Hz: Hertz.

IMU: Unidad de Medición Inercial

Inf: infinito.

LIDAR: detección por imagen de laser.

ORB-SLAM2: Oriented fast and Rotated BRIEF SLAM V2.0.

Plano XY: Plano sobre los ejes X y Y .

PCL: Point cloud nube de puntos.

Ransac: Random sample consensus.

Rec-HV: Reconstruction for point high variability

RGB: red, green, blue.

RGB-D: red, green, blue y depht.

ROS: sistema operativo robótico.

RVIZ: visualización de ROS.

SLAM: Localización y Mapeo Simultaneo

V-SLAM: visual SLAM.

SLAM-Dinámico: SLAM en ambientes dinámicos.

2D: dos dimensiones.

3D: tres dimensiones.

6D: seis dimensiones.

6.2. Glosario de ecuaciones

α_1 y α_2 : Ángulos correspondientes a las dr_{max1} , dr_{max2} .

β : Ángulo de rotación sobre el eje y del k -ésimo ciclo.

dr : Desviación estándar.

dr_{max1} , dr_{max2} : Desviaciones estándar máximas.

dr_k : Vector de desviación estándar para el k -ésimo ciclo.

i : posición en el vector r

$i1$ y $i2$: posiciones de los dr_{max1} , dr_{max2} en el vector dr_k .

k : número de lectura del ciclo del LIDAR.

l : longitud del vector.

Mt_k : Matrices de transformación homogénea del k -ésimo ciclo.

Mt_{k-1} : Matrices de transformación homogénea para el k -ésimo -1 ciclo

n : número de ciclos. O_0 : punto de origen de marco de referencia 0.

O_k : punto de origen de marco de referencia del k -ésimo ciclo.

O_{k-1} : punto de origen de marco de referencia del k -ésimo-1 ciclo.

p : número de puntos de la PCL del k -ésimo ciclo.

PCL_{Est-k} : nube de puntos de la zona estática.

PCL_{Din} : nube de puntos del objeto dinámico.

PCL_k : nube de puntos reducida después de aplicar la traslación y rotación.

PCL_{kT} : PCL de reconstrucción del k -ésimo ciclo.

PCL_{kT-1} : PCL de reconstrucción del k -ésimo-1 ciclo.

PCL_{Oek} : nube de puntos de la zona estática extraída de la zona determinada como dinámica en la segmentación por ángulo.

$PCL_{Reducida}$: Nube de puntos reducida.

PCL_{RGB-D} : nube de puntos de la cámara RGB-D

PCL_{ref} : PCL de escena de referencia.

$PCL_{Trans.}$: PCL transitoria

PCL_{ZD} : nube de puntos de la zona dinámica.

PCL_{zek} : nube de puntos de la zona estática obtenida de la primera segmentación.

Pt_{kT} : Imagen de 2 dimensiones de la proyección de la PCL_{kT}

Pt_{ref} : Imagen de 2 dimensiones de la proyección de la PCL_{kT}

q : número de puntos de la PCL del k -ésimo-1 ciclo.

r : profundidad.

r : vector de profundidad.

r_k : vector profundidad del k -ésimo ciclo

\bar{r}_k : vector de promedios de profundidad para el k -ésimo ciclo.

r_{pk} : radio de un punto de la PCL_r del k -ésimo ciclo.

t_{vg} : tamaño dl voxel grid.

\mathbf{vt}_{kx} : traslación de la coordenada x del k -ésimo ciclo.

\mathbf{v}_{rp} : vector con las coordenadas de cada punto de la PCL_r

\mathbf{vt}_{ky} : traslación de la coordenada y del k -ésimo ciclo.

\mathbf{vt}_{kz} : traslación de la coordenada z del k -ésimo ciclo.

Vt_k : Vector traslación de parámetros de traslación del k -ésimo ciclo.

x_{max} : Máximo de profundidad para la segmentación por agrupación

$\mathbf{x}_{pk-1}, \mathbf{y}_{pk-1}, \mathbf{z}_{pk-1}$: coordenadas x, y, z de un punto de PCL_{Est} del k -ésimo-1 ciclo.

$\mathbf{x}_{pk}, \mathbf{y}_{pk}, \mathbf{z}_{pk}$: coordenadas x, y, z de un punto de PCL_{Est} del k -ésimo ciclo.

\mathbf{x} : eje x .

\mathbf{y} : eje y .

\mathbf{z} : eje z .

α_{pt} : Ángulo de un punto de la $PCL_{Reducida}$ con respecto al plano XY

θ : ángulo de una medida de profundidad del LIDAR

θ_{min} : ángulo de inicio de muestreo del LIDAR.

θ_k : vector de ángulos del k -ésimo ciclo.

$\Delta\theta$: ángulo de incremento.

φ : ángulo de rotación sobre el eje x del k -ésimo ciclo.

γ : ángulo de rotación sobre el eje z del k -ésimo ciclo

6.3. Anexo 1. Error de reproyección

Al realizar el proceso de calibración a partir de las imágenes patrón el sistema arroja que RGB-D presentaba un error de reproyección (e_{rep}) el cual se mide sobre un pixel, ecuación (12).

$$e_{rep} = 0.114397 \quad (12)$$

El RGB-D tiene una apertura de 57° y un total de 640 px en el eje y. A partir de este se encuentra una relación de 0.089° por pixel, Con los anteriores valores se obtiene el error de reprojection para un grado de apertura del RGB-D (e_α), ecuación (13).

$$e_\alpha = e_{rep} \frac{0.089^\circ}{1px} \quad (13)$$
$$e_\alpha = 0.01018135^\circ$$

Con este valor podemos calcular el error de reprojection bajo una de las condiciones de las pruebas realizadas, ver Figura 54

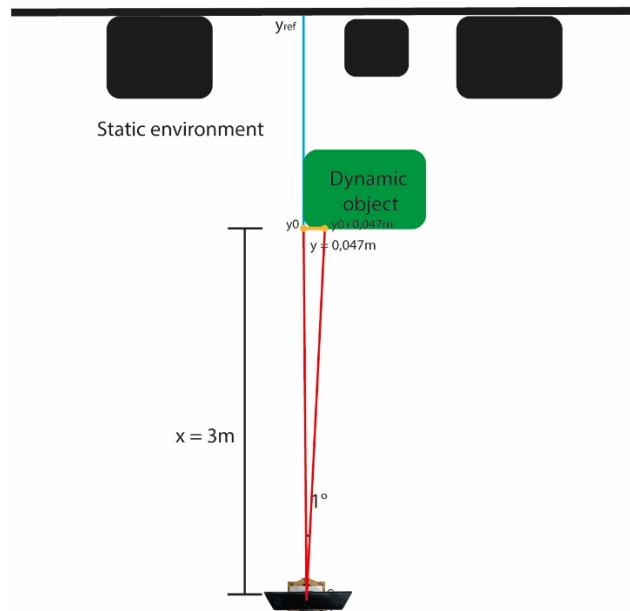


Figura 54. Error de reproyección.

Cuando el objeto dinámico realizaba un recorrido a una distancia del sensor de 3m en el eje x, se obtiene que por cada grado se cubre un fragmento del objeto dinámico en el eje y, ecuación(14).

$$y \approx \tan 1^\circ \cdot 3m = 0.047m \quad (14)$$

Dado que e_α es un corrimiento en el segmento captado se obtiene un error de reprojecion en el eje y (e_y), ecua(15).

$$e_y = \tan 0.01018135^\circ \cdot 3m = 0.004797m$$
$$y_0 \approx y_{ref} \pm 0.004797m \quad (15)$$

Donde y_0 es la medida captada por el sensor que sufre una alteración en referencia a y_{ref} que es la medida del entorno real.

6.4. Anexo 2. Matrices de calibración.

Mono pinhole calibration.

$$D = [0.13826258934353136 \quad -0.2799756919968165 \quad 0.0020722559856376937 \quad 0.008884031820898908 \quad 0.0]$$

$$K = [522.0276786799784 \quad 0.0 \quad 305.8050534416858 \quad 0.0 \quad 518.1331643401821 \quad 248.56968926367864 \quad 0.0 \quad 0.0 \quad 1.0]$$

$$R = [1.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 1.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 1.0]$$

$$P = [528.4050903320312, \quad 0.0, \quad 309.9429018153569, \quad 0.0, \quad 0.0, \quad 528.0941772460938, \\ 248.96161180673516, \quad 0.0, \quad 0.0, \quad 0.0, \quad 1.0, \quad 0.0]$$

Image

width = 640

height = 480

Narrow_stereo

$$\text{camera matrix} = \begin{bmatrix} 522.027679 & 0.000000 & 305.805053 \\ 0.000000 & 518.133164 & 248.569689 \\ 0.000000 & 0.000000 & 1.000000 \end{bmatrix}$$

$$\text{distortion} = [0.138263 \quad -0.279976 \quad 0.002072 \quad 0.008884 \quad 0.000000]$$

$$\text{rectification} = \begin{bmatrix} 1.000000 & 0.000000 & 0.000000 \\ 0.000000 & 1.000000 & 0.000000 \\ 0.000000 & 0.000000 & 1.000000 \end{bmatrix}$$

$$\text{projection} = \begin{bmatrix} 528.405090 & 0.000000 & 309.942902 & 0.000000 \\ 0.000000 & 528.094177 & 248.961612 & 0.000000 \\ 0.000000 & 0.000000 & 1.000000 & 0.000000 \end{bmatrix}$$

6.5. Anexo 3. Repositorios.

<https://github.com/MAB1144-Python/Research-in-SLAM>



https://github.com/MAB1144-Python/Installation_guide



<https://github.com/MAB1144-Python/rotation-and-traslate-Point-Cloud-3D>



<https://github.com/MAB1144-Python/SLAM-REC-HV-version-1.0-test>



<https://github.com/MAB1144-Python/SLAM-REC-HV-version-2.0-test>



Repositorio Drive de Google con software Rec-HV: encontrara el software que se desarrolló en este proyecto de investigación.

https://drive.google.com/drive/folders/11918ufpZPDdhMI1Crm6opmt_3vyi6L8e?usp=share_link



Repositorio Videos youtube: encontrara diferentes videos de los resultados de este proyecto de investigación.

Prueba Rec-HV reconstrucción mientras se hace un recorrido con presencia de objeto dinámico:

<https://youtu.be/spJmmkw1mPo>



Prueba Rec-HV Reconstrucción con varios escenarios con presencia del objeto dinámico:

<https://youtu.be/fpQ7Tt8ZwzU>



<https://www.youtube.com/watch?v=xUmLrVYb-9k>



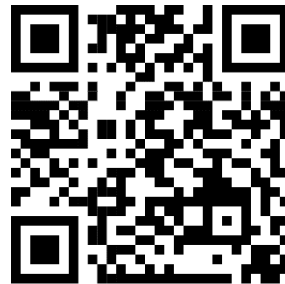
<https://www.youtube.com/watch?v=ycLmwnvg0qk>



<https://www.youtube.com/watch?v=j-mr0cfVcVA>



<https://www.youtube.com/watch?v=3L--IJSgva4>



<https://www.youtube.com/watch?v=Pnj8QZT3dj0&t=46s>

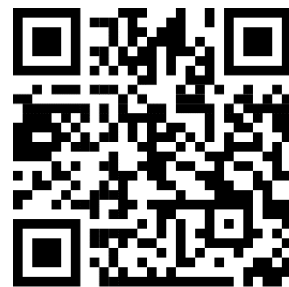


<https://www.youtube.com/watch?v=gn1R-jDKqOM&t=87s>



Prueba detección de objeto dinámico en línea:

<https://www.youtube.com/watch?v=mtzcu8xpaXk>



Referencias

- Arévalo, et al. (2016). *Scene structure registration for localization and mapping*. En: Robotics and Autonomous Systems, 75. (pp. 649-660)
- Astrup, A., et al. (2018). *Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM*. En: Computers and Electronics in Agriculture, 145. (pp. 217-225).
- Abramenko, et al. (2021). *Real-Time Lidar-Based Localization of mobile Ground robot*. En: Procedia computer science, 186. (pp. 440-448). doi:<https://doi.org/10.1016/j.procs.2021.04.164>
- Ai, et al. (2021). *Visual SLAM in dynamic environments based on object detection*. En: Defence Technology, 17(5). (pp. 1712-1721). doi:<https://doi.org/10.1016/j.dt.2020.09.012>
- Akanshu, et al. (2018). *Construction of a 3D Map of Indoor Environment*. En: rocedia Computer Science, 125. (pp. 24-131).
- Agioutantis, Z., et al. (2021). *Navigation system for a semi-autonomous shuttle car in room and pillar coal mines based on 2D LiDAR scanners*. En: Tunnelling and Underground Space Technology Incorporating Trenchless Technology Research., 117.
- Bao, H., et al. (2019). *Survey and evaluation of monocular visual-inertial SLAM algorithms for augmented reality* . En: Virtual Reality & Intelligent Hardware, [1]. (pp. 386-410).
- Batista et al. (2020). *Earth-fixed trajectory and map online estimation: Building on GES sensor-based SLAM filters*. En: Robotics and Autonomous Systems, 130. 103552.
- Bendahan, R., et al. (2018). *Spatially sensitive statistical shape analysis for pedestrian recognition from LIDAR data*. En: Computer Vision and Image Understanding, [171]. (pp. 1-9).
- Behnke, S. et al. (2018). *Feature-based visual odometry prior for real-time semi-dense stereo SLAM*. En: Robotics and Autonomous Systems, [109]. (pp. 38-58). doi:<https://doi.org/10.1016/j.robot.2018.08.002>
- Bhat, A., et al. (2018). *An Efficient ORB based Face Recognition framework for Human-Robot Interaction*. En: Procedia Computer Science, 133. (pp. 913-923).

Bilberg, A. & Malik, A. (2018). digital twins of human robot collaboration in a production setting. *Procedia Manufacturing*, [17]. (pp.278-285).

Bueno, M., et al. (2015). Metrological evaluation of KinectFusion and its comparison with Microsoft Kinect sensor. En: *Measurement* [73]. (pp. 137-145).

Burak Tatar, A., et al. (2020). *Four-legged hunter (FLH) robot: design and shooting control to moving targets with SMC*. En: *Simulation Modelling Practice and Theory*, [104]. (pp. 102-117). doi:<https://doi.org/10.1016/j.simpat.2020.102117>

Burce, M. et al. (2018). *3-D reconstruction of maize plants using a time-of-flight camera*. En: *Computers and Electronics in Agriculture* [145]. (pp. 235- 247). doi:<https://doi.org/10.1016/j.compag.2018.01.002>.

Cadena, et al. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. En: *IEEE Transactions On Robotics*, 32(6).

Cao, X., et al. (2019). *Fast and robust feature tracking for 3D reconstruction*. En: *Optics and Laser Technology*, 110. (pp. 120-128). doi:<https://doi.org/10.1016/j.optlastec.2018.05.036>

Cao, X., et al. (2021). *Attention aware cost volume pyramid based multi-view stereo network for 3D reconstruction*. *ISPRS Journal of Photogrammetry and Remote Sensing*(175), 448-460. doi:<https://doi.org/10.1016/j.isprsjprs.2021.03.010>

Cýrus, J., et al. (2020). Hybrid SLAM modelling of autonomous robot with augmented reality device. *Materials Today: proceedings*, 32(2). (pp. 103-107).

Czerniawski, T., et al. (2020). Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point. En: *Automation in Construction* , 113. (pp. 103-144).

Chacon, et al. (2019). *A novel instrument to compare dynamic object detection algorithms*. En: *Image and Vision Computing*, [88]. (pp. 19-28). doi:<https://doi.org/10.1016/j.imavis.2019.04.006>

Chan, J., et al. (2019). Seamless Navigation And Mapping Using An Ins/Gnss/Grid-Based Slam Semi-Tightly Coupled Integration Scheme. En: *Information Fusion*, [50]. (pp. 181-196).

Chang, L., et al. (2020). *An improved method for odometry estimation based on EKF and Temporal Convolutional Network*. Physical Communication.

doi:<https://doi.org/10.1016/j.phycom.2020.101178>

Chen, J., et al. (2018). *SLAM-driven robotic mapping and registration of 3D point clouds*. En: Automation in Construction, [89]. (pp. 38-48).

Cheein, F., et al. (2016). LiDAR and thermal images fusion for ground-based 3D characterisation of fruit trees. Biosystems Engineering, 151, 479-494.

Chen, Y., et al. (2020). *Land-cover classification of multispectral LiDAR data using CNN with optimized hyper-parameters*. En: ISPRS Journal of Photogrammetry and Remote Sensing, [166]. (pp. 241-254).

Chen, et al. (2021). DPNet: Detail-preserving network for high quality monocular depth estimation. Pattern Recognition, [109].

Chen, L., et al. (2018). SLAM-based dense surface reconstruction in monocular Minimally Invasive Surgery and its application to Augmented Reality. En: Computer Methods and Programs in Biomedicine, [158]. (pp. 135-146).

Cheng, et al. (2015). Robust Linear Pose Graph-based SLAM. Robotics and Autonomous Systems, [72]. (pp. 71-82).

Cheng, et al. (2018). *SLAM-driven robotic mapping and registration of 3D point clouds*. En: Automation in Construction, [89]. (pp. 38-48).

Chen, S., et al. (2021). *DPNet: Detail-preserving network for high quality monocular depth estimation*. En: Pattern Recognition, [109]. doi:<https://doi.org/10.1016/j.patcog.2020.107578>

Chiodini, S., et al. (2019). *An Evaluation of ROS-compatible Stereo Visual SLAM Methods on a nVidia Jetson TX2*. En: Measurement, [140]. (pp. 161-170). doi:<https://doi.org/10.1016/j.measurement.2019.03.038>

Chong, A., et al. (2015). *Sensor Technologies and Simultaneous Localization and Mapping (SLAM)*. Procedia Computer Science [76]. (pp. 174 – 179).

Chong, N. & Tiwari, K. (2020). *Simultaneous Localization and Mapping (SLAM): Where is the robot?* Multi-robot Exploration for Environmental Monitoring, (pp. 31-38).

Cordeiro, N., & Pedrino, E. (2019). *A new methodology applied to dynamic object detection and tracking systems for visually impaired people*. En: Computers and Electrical Engineering. [77]. (pp. 61-71). doi:<https://doi.org/10.1016/j.compeleceng.2019.05.003>

Daalen., C., et al. (2020). *Stochastic Triangular Mesh Mapping: A Terrain Mapping Technique for Autonomous Mobile Robots*. En: Robotics and Autonomous Systems, 127. doi:<https://doi.org/10.1016/j.robot.2020.103449>

Dalcin, L., et al. (2015). *Computational cost of isogeometric multi-frontal solvers on parallel distributed memory machines*. En: Computer Methods in Applied Mechanics and Engineering, [284]. (pp. 971-987).

Delanoy, J., Coeurjolly, D., Lachaud, J.-O., & Bousseau, A. (2019). Combining voxel and normal predictions for multi-view 3D sketching. *Computers & Graphics*, 82, 65-72. doi:<https://doi.org/10.1016/j.cag.2019.05.024>

Ding, H., et al. (2020). *HoPPF: A novel local surface descriptor for 3D object recognition*. En: Pattern Recognition , [103]. (pp. 107-272).

Dynamic-SLAM: *Semantic monocular visual localization and mapping based on deep learning in dynamic environment*. (2019). En: Robotics and Autonomous Systems, [117]. (pp. 1-16). doi:<https://doi.org/10.1016/j.robot.2019.03.012>

Dyo, V., et al. (2022). Visual SLAM algorithms and their application for AR, mapping, localization and wayfinding. *Array*. doi:<https://doi.org/10.1016/j.array.2022.100222>

Elliott, D., et al. (2019). Recent advances in connected and automated vehicles. *Journal of Traffic and Transportation Engineering*, [6]. (pp. 109-131).

Englot, B. (2018). *LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain*. En: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (pág. 18401074). Madrid, Spain. doi:10.1109/IROS.2018.8594299

Escalera, S., et al. (2021). *Sign Language Recognition: A Deep Survey*. En: Expert Systems With Applications, [164].

Fan, et al. (2018). Dynamic objects elimination in SLAM based on image fusion. Pattern Recognition Letters, [22].

Fan, Y., et al. (2018). *Dynamic objects elimination in SLAM based on image fusion*. En: Pattern Recognition Letters, [22]. (pp. 191-201). doi:<https://doi.org/10.1016/j.patrec.2018.10.024>

Fan, G. (2019). Semantics-enhanced supervised deep autoencoder for depth image-based 3D model retrieval. Pattern Recognition Letters, [125]. (pp. 806-812).

Feng, C.(2019). An Occupancy Grid Mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments. En: Automation in Construction, [104]. (pp. 230-245).

Fukao, T., et al. (2021). *LIDAR based localization using simplified 2D map on agricultural roads. Measurement*. En: Sensors, [18]. doi:<https://doi.org/10.1016/j.measen.2021.100298>

Gallant, M. & Marshall, J. (2016). *Automated rapid mapping of joint orientations with mobile LiDAR*. En: International Journal of Rock Mechanics Mining Sciences, [90]. (pp. 1–14). doi:<https://doi.org/10.1016/j.ijrmms.2016.09.014>

Gamallo, C., et al. (2015). *Omnidirectional visual SLAM under severe occlusions*. En: Robotics and Autonomous Systems, [65]. (pp. 76-87).

Garcia-Fidalgo; Ortiz. (2015). *Vision-based topological mapping and localization methods: A survey*. En: Robotics and Autonomous Systems, [64] (pp. 1-20).

Garcia-Fidalgo, E., et al. (2022). *LiODOM: Adaptive local mapping for robust LiDAR-only odometry*. En: Robotics and Autonomous Systems. doi:<https://doi.org/10.1016/j.robot.2022.104226>

Gaspar, A., et al. (2018). *Urban@CRAS dataset: Benchmarking of visual odometry and SLAM techniques*. En: Robotics and Autonomous Systems. (pp. 59-67). doi:<https://doi.org/10.1016/j.robot.2018.08.004>

Geng, et al. (2020). Simultaneous 3D hand detection and pose estimation using single depth images. En: Pattern Recognition Letters, [140]. (pp. 43–48).

Griepentrog, H., et al. (2018). *Iterative individual plant clustering in maize with assembled 2D LiDAR data*. Computers in Industry, [99]. (pp. 42–52).

Guangyun, et al. (2020). *SLAM integrated mobile mapping system in complex urban environments*. En: ISPRS Journal of Photogrammetry and Remote Sensing, [166]. (pp. 316-332). doi:<https://doi.org/10.1016/j.isprsjprs.2020.05.012>

Haoran, et al. (2022). *Mapping method of single LiDAR for indoor degraded environment*. En: Computers and Electrical Engineering, [103].

Henry, P., et al. (2012). *RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments*. En: The International Journal of Robotics Research, 31(5), 647–663.

Hernández, C., & Ortiz, F. (2020). A real-time map merging strategy for robust collaborative reconstruction of unknown environments. En: Expert Systems with Applications, [145].

Hol’y, B. (2016). *Registration of Lines in 2D LIDAR Scans via Functions of Angles*. En: IFAC papers online, [45]. (pp. 109-114). Ostrava-Poruba.

Hongmin, et al. (2020). *Depth-map completion for large indoor scene reconstruction*. En: Pattern Recognition [99]. (pp. 107-112).

Hsu, Ya-Wen, et al. (2018). Application of multisensor fusion to develop a personal location and 3D mapping system. Optik, 172, 328-339.

Hua, C.-H., et al. (2020). Learning 3D spatiotemporal gait feature by convolutional network for person identification. En: Neurocomputing, [397]. (pp. 192-202).

Huang, et al. (2016). *A framework for multi-session RGBD SLAM in low dynamic workspace environment*. En: CAAI Transactions on Intelligence Technology, 1 [1]. (pp. 90-103).

Huixia, et al. (s.f). Indoor multi-sensor fusion positioning based on federated filtering. *Measurement*. doi:<https://doi.org/10.1016/j.measurement.2020.107506>

- Javanmardi, et al. (2019). Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area. En: IATSS Research, 43(1). (pp. 1-13).
- Li, S.-p., et al. (2019). *Semi-direct monocular visual and visual-inertial SLAM with loop closure detection*. En: Robotics and Autonomous Systems, [112]. (pp. 201-210).
- Li, B., et al. (2022). *Overview of deep learning application on visual SLAM*. En: Displays. doi:<https://doi.org/10.1016/j.displa.2022.102298>
- Liu, H., et al. (2019). *Detection and tracking of pedestrians and vehicles using roadside LIDAR sensors*. En: Transportation Research Part C, [100]. (pp. 68-87).
- Liu, H., & Wang, L. (2021). *Collision-free human-robot collaboration based on context awareness*. En: Robotics and Computer Integrated Manufacturing, [67]. doi:<https://doi.org/10.1016/j.rcim.2020.101997>
- Liu, L., et al. (2019). ObjectFusion: An object detection and segmentation framework with RGB-D SLAM and convolutional neural networks. Neurocomputing, [345]. (pp. 3-14).
- MA, J. & Quan, S. (2020). *On shortened 3D local binary descriptors*. En: Information Sciences, [510]. (pp. 33-49).
- Markham et al. (2018). Visual SLAM and Structure from Motion in Dynamic Environments: A Survey. En: ACM Computing Surveys, 51(37).
- Mazurek, P., et al. (2020). Comparison of sixteen methods for fusion of data from impulse-radar sensors and depth sensors applied for monitoring of elderly persons. En: Measurement(154), 107.
- Menassa, et al. (2020). A scene-adaptive descriptor for visual SLAM-based locating applications in built environments. En: Automation in Construction, 112.
- Meng, M., et al. (2017). *Improving RGB-D SLAM in dynamic environments: A motion removal*. En: Robotics and Autonomous Systems, [89]. (pp. 110-122).
- Meng, M., et al. (2018). *Motion removal for reliable RGB-D SLAM in dynamic environments*. En: Robotics and Autonomous Systems, [108]. (pp. 115-128). doi:<https://doi.org/10.1016/j.robot.2018.07.002>

Miura, et al. (2019). *SeqSLAM++: View-based robot localization and navigation*. En: Robotics and Autonomous Systems. (pp. 112, 13-21).

Rehman, et al. (2018). *Iterative mobile robot in a dynamic environment*. IFAC Paper online, 51(30), 354-359.

Rong, Z., et al. (2019). *Dynamic-SLAM: Semantic Monocular Visual Localization and Mapping Based on Deep Learning in Dynamic Environment*. En: Robotics and Autonomous Systems, [117]. (pp. 1-16).

Sahin, et al. . (2020). *A review on object pose recovery: From 3D bounding box detectors to full 6D pose estimators*. Image and Vision Computing, [96], 103898.

Seitz, S., et al. (2006). *Photo Tourism: Exploring Photo Collections in 3D*. ACM Transactions on graphics, 25, 835-846. doi:<https://doi.org/10.1145/1141911.1141964>

Song, H., et al. (2018). *Robust RGB-D Visual Odometry Based on Edges and Points*. Robotics and Autonomous Systems, 107, 2019-220.

Taheri, H., & Xia, Z. C. (2021). *SLAM; definition and evolution*. En: Engineering Applications of Artificial Intelligence, [94].

Tan, P., et al. (2019). *Collaborative visual SLAM for multiple agents: A brief survey*. Virtual Reality & Intelligent Hardware , 1(5), 461-482.

Wang, Y., et al. (2017). *A Survey of Simultaneous Localization and Mapping on unstructured Lunar Complex Environment*.

Wang, Y., et al. (2020). *Drosophila-inspired 3D moving object detection based on point clouds*. En: Information Sciences, [534]. (pp. 154-171). doi:<https://doi.org/10.1016/j.ins.2020.05.006>

Wang, Z. (2020). *Review of real-time three-dimensional shape measurement techniques*. En: Measurement, [156], 107624.

Yan, Z., & Zha, H. (2019). *Flow-based SLAM: From geometry computation to learning*. En: Virtual Reality & Intelligent Hardware, [1]. (pp. 435-460).