FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

RECONSTRUCCIÓN DE UN ENTORNO EN TRES DIMENSIONES MEDIANTE EL USO DE KINECT V.2.

AUTOR

Kevin Vinicio Jaramillo Páez

AÑO

2019

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

RECONSTRUCCIÓN DE UN ENTORNO EN TRES DIMENSIONES MEDIANTE EL USO DE KINECT V.2

Trabajo de titulación presentado en conformidad con los requisitos establecidos para optar por el título de Ingeniero en Electrónica y Redes de información

Profesor Guía
MSc. David Fernando Pozo Espín

Autor
Kevin Vinicio Jaramillo Páez

Año
2019

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo, Reconstrucción de un entorno en tres dimensiones mediante el uso de Kinect v.2, a través de reuniones periódicas con el estudiante Kevin Vinicio Jaramillo Páez, en el semestre 201910, orientando sus conocimientos y competencias para un eficiente desarrollo del tema elegido y cumpliendo con todas las disposiciones vigentes que regulan los Trabajos de titulación”

_____

David Fernando Pozo Espín
Máster en Ciencias en Automatización y Robótica
C.I.: 1717340143

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, Reconstrucción de un entorno en tres dimensiones mediante el uso de Kinect v.2, del estudiante Kevin Vinicio Jaramillo Páez, en el semestre 201910, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Jorge Rosero Beltrán
Máster en Ciencias con Especialidad en Automatización
C.I.: 1803610185

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”

Kevin Vinicio Jaramillo Páez

C.I.: 1716561392

AGRADECIMIENTO

Agradezco a Dios por su guía y fortaleza. A mi familia por todo el apoyo brindado. Al MSc. David Fernando Pozo Espín por su paciencia y dirección en este proyecto de titulación. A la Unidad de Innovación Tecnológica de la Universidad de las Américas, a Ricardo López, Santiago Solorzano y David Ponce ya que sin su ayuda no hubiese sido posible la conclusión exitosa de este proyecto.

Kevin Jaramillo P.

DEDICATORIA

Dedico este trabajo de titulación a mi familia, que siempre me han apoyado. Especialmente, a mis padres y hermanas por su esfuerzo y colaboración incondicional.

Kevin Jaramillo P.

RESUMEN

Con el paso del tiempo se ha buscado que los diferentes avances tecnológicos permitan un alcance más cercano a la manera en que las personas pueden ver, interactuar y percibir su entorno, mediante imágenes en tres dimensiones. Esta búsqueda ha sido denominada visión artificial la cual, puede ser llevada a cabo gracias a la reconstrucción de entornos. Entre las diferentes tecnologías que existen actualmente para reconstrucción en tres dimensiones se encuentra la tecnología del sensor Kinect v.2. Este sensor consiste en tres sistemas diferentes trabajando como uno solo; una cámara RGB y un sensor de profundidad el cual consiste en dos partes; un láser infrarrojo y un sensor monocromático. Obteniendo así una nube de puntos de gran calidad, independientemente de las condiciones de luz que exista en el lugar.

En este trabajo de titulación se busca la reconstrucción de un entorno en tres dimensiones gracias al uso del sensor Kinect v.2. Este será implementado sobre una plataforma, la cual rotará en 360 grados sobre su eje, permitiendo al sensor tomar capturas en los diferentes ángulos para así obtener una reconstrucción completa del entorno.

Para procesar los datos obtenidos se utilizará el software MATLAB, y gracias al uso de diferentes procesos matemáticos y manejo de matrices se obtendrá una la reconstrucción del entorno en 360 grados. De igual manera se desarrollará una herramienta la cual permitirá controlar el ángulo de rotación de la reconstrucción; la calidad de esta; permitirá la visualización de la nube de puntos obtenida y en caso de ser necesario la manipulación de esta mediante el uso de distintos filtros. Finalmente se llevarán a cabo distintas pruebas para comprobar el correcto funcionamiento de la herramienta, tanto en ambientes interiores como exteriores y bajo distintas condiciones de luz.

ABSTRACT

Over time, it has been sought that different technological advances allow a closer reach to the way in which people can see, interact and perceive their environment, through three-dimensional images. This search has been called artificial vision which can be carried out thanks to the reconstruction of environments. Among the different technologies that currently exist for reconstruction in three dimensions is the Kinect v.2 sensor technology. This sensor consists of three different systems working as one; an RGB camera and a depth sensor which consists of an infrared laser combined with a monochromatic sensor, which manages to obtain a point cloud of high quality regardless of the lighting conditions that exist in the place.

This degree work will create a reconstruction of a three-dimensional environment thanks to the use of the Kinect v.2 sensor. This will be implemented on a platform, which will rotate in 360 degrees on its axis, allowing the sensor to take captures at different angles to obtain a complete reconstruction of the environment.

In order to carry out the processing of the data obtained, the MATLAB software will be used, and thanks to the use of different mathematical processes and matrix management, the reconstruction of the environment in 360 degrees will be obtained. In the same way a tool will be developed which will allow to control the angle of rotation of the reconstruction, the quality of this and will allow the visualization of the point cloud obtained and in case of being necessary the manipulation of it by using different filters. Finally, different tests will be carried out to verify the correct functioning of the tool, both in interior and exterior environments and under different lighting conditions.

ÍNDICE

1. Introducción 1
1.1 Alcance 3
1.2 Justificación 4
1.3 Objetivo General 4
1.4 Objetivos Específicos 5
2. Marco Teórico 5
2.1 Técnicas de reconstrucción 3D 5
2.1.1 Reconstrucción mediante tecnología de ultrasonido 5
2.1.2 Reconstrucción mediante tecnología de triangulación infrarroja 6
2.1.3 Reconstrucción mediante tecnología de tiempo de vuelo 7
2.1.4 Reconstrucción mediante tecnología de luz estructurada 8
2.2 Dispositivos comerciales para reconstrucción en tres dimensiones 8
2.2.1 Hokuyo YVT-35LX Scanning Rangefinder 8
2.2.2 SlamTec RPLIDAR A2 M8 10
2.2.3 Waymo System 11
2.2.4 ZED Stereo Camera 12
2.2.5 Sensor Kinect v1 13
2.2.6 Sensor Kinect v.2 15
2.3 Fundamentos matemáticos 19
2.3.1 Matrices de rotación 19
2.3.2 Ángulos de Euler 20
2.3.3 Coordenadas y matrices homogéneas 21
2.4 Procesamiento de datos 23
2.4.1 Nube de puntos 23
2.4.2 Técnicas de filtrado para nubes de puntos 24
3. Diseño e implementación 28

3.1 Metodología 28
3.2 Diseño de plataforma giratoria..... 30
3.2.1 Diseño e implementación 30
3.2.2 Manejo de la plataforma 32
3.3 Obtención y manejo de datos 33
3.3.1 Matlab, librerías y software 33
3.3.2 Tool para reconstrucción en tres dimensiones 34
4. Pruebas y resultados 49
5. Conclusiones y recomendaciones 74
5.1 Conclusiones..... 74
5.2 Recomendaciones..... 75
Referencias 76

1. Introducción

Con el paso del tiempo se ha buscado que los diferentes avances tecnológicos permitan un alcance más cercano a la manera en que las personas pueden ver, interactuar y percibir su entorno, mediante imágenes en tres dimensiones. Esta búsqueda ha sido denominada visión artificial la cual, puede ser llevada a cabo gracias a la reconstrucción de entornos. Permitiendo de cierta manera alcanzar la autonomía de los diferentes dispositivos usados en las diferentes industrias (Silva & Zapata, 2015).

La tecnología de reconstrucción en tres dimensiones ha dado grandes pasos en los últimos años, permitiendo que actualmente se encuentre inmersa en nuestra sociedad en casos como: la reconstrucción de objetos para su posterior modelado (Ilbay, 2014); la reconstrucción de estructuras arqueológicas sin la necesidad de manipulación de estas. (Hernández, 2015); control de una computadora basado en el movimiento del usuario (Demitseva, 2015); la evasión de obstáculos en dispositivos humanoides (Ibarra, Cisneros, García, Lavín, & Yañez, 2008); detección de objetos cercanos y su distancia hacia el objeto (Navas, Vargas, & Morales, 2017); controlar el flujo de los caudales en los ríos (Flanagin, Grenotton, Ratcliff, Abdelguerfi, & Shaw, 2005); en la creación de un copiloto autónomo, el cual ayuda a tomar decisiones en la vía (MIT, 2012); en la medicina gracias a que permite observar y reconocer el movimiento humano y su respiración (Chuya, Cuenca, & Delgado, 2013). De esta manera se logra que las personas no realicen tareas delicadas, repetitivas o de alta complejidad (Navarro, 2017).

Es así como, la tecnología de visión artificial y sus aplicaciones como reconstrucción de entornos es sumamente usada para la toma de decisiones en campos como: la medicina, la navegación robótica, en la milicia o en el sector automotriz y en el desarrollo de vehículos autónomos (MIT, 2012).

Las técnicas para reconstrucción de entornos pueden ser clasificadas como: activas y pasivas. Las técnicas activas son basadas en iluminar la escena ya sea mediante el uso de láser o luz estructurada, mientras que las pasivas analizan una gran cantidad de imágenes de la escena y debido a ello son conocidas como *multi-view stereo vision* o métodos fotogramétricos. Estas últimas poseen ventajas en comparación con las técnicas activas, especialmente en el proceso de captura de imágenes, el cual es simple y económico; a pesar de estos beneficios, las técnicas activas para adquisición 3D han sido usadas predominantemente en el ámbito profesional, gracias a que proveen de alta precisión y menor cantidad de ruido (Wolff et al., 2016).

La reconstrucción 3D se puede dar a partir de un objeto, una escena, varias imágenes o a partir de un video. Con el uso de nuevas tecnologías como sensores ópticos, los cuales logran una representación precisa de la escena gracias al uso de millones de puntos. Actualmente, nuevas aplicaciones requieren de una interacción en tiempo real, lo que conlleva al desarrollo de modelos y herramientas que permitan simplificar el procesamiento de grandes cantidades de datos, con un costo relativamente bajo. Esto se lo realiza a partir de información digital o analógica, obtenida gracias al uso de cámaras o sensores de rango como: LIDAR, infrarrojos, láser, etc. El tipo de tecnología usada influenciará directamente en el desarrollo de la reconstrucción, ya que al usar tecnologías como estéreo visión, se podría generar discontinuidades, afectando así la calidad de la misma (MoBiVAP Group Research, 2013).

La reconstrucción de un entorno en tres dimensiones se puede desarrollar gracias al uso de diferentes tecnologías como sensores LIDAR, cámaras de visión estéreo o sensores Kinect. Los sensores LIDAR o Light Detection and Ranging, es un sistema láser que permite obtener la distancia entre un objeto o superficie y el láser, gracias a la diferencia entre el tiempo de salida y el tiempo de regreso del láser. Obteniendo como resultado un mapa 3D de alta resolución (Flanagin et al., 2005). La tecnología ZED, basada en visión estereoscópica, contiene un conjunto de dos cámaras o también denominadas “ojos” las cuales

gracias a la triangulación proveen una imagen en tres dimensiones de la escena observada, permitiendo de esta manera que se reconozca el espacio y la profundidad de los objetos (StereoLabs, 2018).

Entre las múltiples tecnologías que pueden ser usadas para obtener la profundidad de una imagen se encuentran sensores de ultrasonido (von Ramm, Pavy, & Smith, 1991), sensores de triangulación mediante luz infrarroja (Jin, 2008), sensores láser (Cheng, Jia, Zheng, & Xie, 2002) y sensores infrarrojos mediante el uso de tiempo de vuelo (Schwarz & Navab, 2012).

La tecnología de sensores Kinect v2 desarrollado por Microsoft también permiten la reconstrucción 3D. Mediante el uso de un sensor infrarrojo y una cámara puede llevar a cabo un escaneo de la escena, la imagen obtenida es enviada a un microchip, el cual permite capturar personas y cosas en tres dimensiones. El escáner es capaz de generar mapas de profundidad en tiempo real e incluso a color, llamado “imagen RGB-D” con más de 300 mil puntos y con una tasa de refresco de 30 Hz (Popescu & Lungu, 2014) (Wasenmuller & Sticker, 2017).

En el presente trabajo de titulación se analizará las diferentes técnicas usadas para reconstrucción en tres dimensiones, en especial sus ventajas y configuraciones; los dispositivos que se encuentran actualmente en el mercado; la calidad de la reconstrucción; los diferentes tipos de ambientes en que pueden ser usadas; y su funcionamiento. Al finalizar se procederá con la optimización de los datos obtenidos mediante el uso de filtros y diferentes algoritmos para el correcto manejo de los datos. Finalmente se realizará la reconstrucción de un entorno en tres dimensiones mediante el uso de un Kinect v.2.

1.1 Alcance

Lo que se busca con este trabajo de titulación es la reconstrucción de un entorno en tres dimensiones gracias al uso del sensor Kinect v.2. Este será

implementado sobre una plataforma, la cual rotará en 360 grados sobre su eje, permitiendo al sensor tomar capturas en los diferentes ángulos para una reconstrucción completa del entorno. Para el procesamiento de datos se utilizará el microchip Capri (PS1200), incluido en el sensor Kinect v2, además de herramientas que permitan un alto procesamiento matemático; obteniendo así una imagen clara y definida.

Para llegar a cumplir lo anteriormente mencionado, se elaborará una plataforma a medida para el movimiento del sensor en ángulos exactos, esta plataforma puede ser movida con la finalidad de reconstruir diferentes entornos.

1.2 Justificación

En la actualidad existen grandes avances en la tecnología, los cuales requieren la información del mundo en tres dimensiones para poder ser capaces de la toma de decisiones relacionados a la navegación robótica. Evitando de esta manera que las personas realicen actividades peligrosas, dañinas o delicadas como acceder a lugares en los cuales no existen las condiciones necesarias para la vida. Acceder a puntos que se encuentran a una mayor profundidad que la que el cuerpo humano puede soportar. También se los puede usar en distintos campos como lo es la medicina, construcción y exploración. Estos sistemas generalmente son manejados remotamente por personas, lo cual los vuelve dependientes de su usuario y en muchos casos evita que se lleven a cabo tareas por la distancia entre el operador y el hardware. Este es el motivo por el cual se requiere la creación de un sistema el cual permita obtener visión con profundidad, logrando de esta manera grandes pasos en la autonomía de estos.

1.3 Objetivo General

Implementar un prototipo para reconstruir un entorno 3D mediante el uso de Kinect v2, en base a una nube de puntos.

1.4 Objetivos Específicos

- Estudiar las diferentes tecnologías que existen para la reconstrucción de entornos y sus principales aplicaciones.
- Analizar el funcionamiento del Kinect v2 y adquisición de nube puntos.
- Implementar el prototipo para la reconstrucción del entorno.
- Realizar pruebas de funcionamiento del prototipo y analizar sus resultados.

2. Marco Teórico

2.1 Técnicas de reconstrucción 3D

Para lograr una reconstrucción con profundidad se puede usar diferentes tipos de tecnologías como ultrasonido, triangulación infrarroja, láser o tiempo de vuelo (*time of flight*).

2.1.1 Reconstrucción mediante tecnología de ultrasonido

Este tipo de sistemas fue diseñado para poder realizar un escaneo rápido, proporcionando así la visualización de estructuras dinámicas en tres dimensiones en tiempo real. La reconstrucción mediante ultrasonido usa sonido en altas frecuencias, entre 3 y 15 MHz, con la finalidad de obtener una reconstrucción de estructuras internas del cuerpo. Esto se obtiene mediante la interferencia provocada en los ecos de las ondas transmitidas como se ve en la Figura 1. Generalmente, es usado en diagnósticos clínicos ya que es económico y seguro.

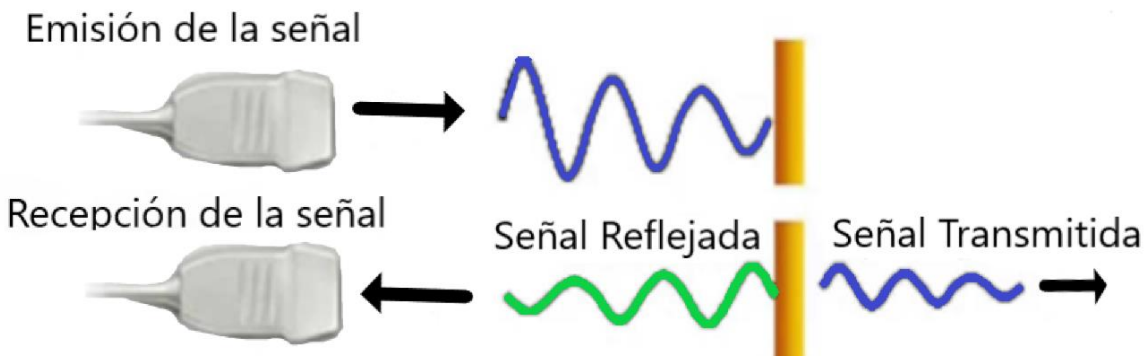


Figura 1. Funcionamiento de reconstrucción mediante ultrasonido. Adaptado de (Shruthi, Renukalatha, & Siddappa, 2015).

Una de sus principales desventajas es su baja calidad y alto ruido dependiente, denominado *Speckle noise* (Shruthi et al., 2015), el cual es un ruido granular que existe inherentemente en este tipo de tecnología; altera los bordes de la reconstrucción obtenida y finos detalles. Y de esta manera limita la resolución del contraste y dificulta el análisis del resultado. *Speckle noise* es considerado como la fuente de ruido predominante en este tipo de imágenes. Asimismo, este tipo de ruido es multiplicativo y removerlo es sumamente complicado (Shruthi et al., 2015) (El-Gwad & Omar, 2017).

2.1.2 Reconstrucción mediante tecnología de triangulación infrarroja

Este tipo de tecnología generalmente usa el algoritmo de reconstrucción *Voronoi-based,* el cual calcula la triangulación de Delaunay de los puntos muestreados. Mediante la emisión de una señal infrarroja y un receptor infrarrojo se obtiene un conjunto de señales (puntos obtenidos) como se ve en la Figura 2, las cuales, mediante el uso de algoritmos, permite obtener la distancia entre cada punto y de esta manera se logra obtener la profundidad de los objetos (Frigerio, Marcon, & Tubaro, 2013).

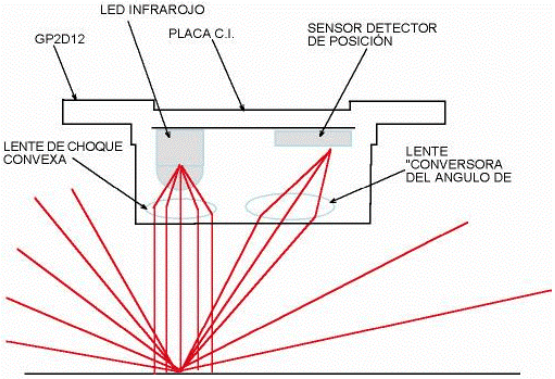


Figura 2. Triangulación mediante el uso de un sensor infrarrojo
Tomado de (Ibarra et al., 2008).

El algoritmo de triangulación Delaunay es sumamente conocido para realizar reconstrucciones en tres dimensiones, mediante el uso de puntos de referencia, se construye una triangulación entre estos y los puntos a su alrededor. Gracias a la diferencia que exista entre los puntos, se establece la profundidad de estos. Obteniendo así, puntos tridimensionales los cuales forman una malla de nube de puntos. Este tipo de reconstrucción contiene una gran cantidad de ruido y errores numéricos en la triangulación. Es por esta razón que se genera una reconstrucción generalmente irregular. Debido a esto la reconstrucción resultante puede variar tanto en forma, como en tamaño (Khoshelham & Elberink, 2012).

2.1.3 Reconstrucción mediante tecnología de tiempo de vuelo

El principio de tiempo de vuelo o *Time of Flight (ToF)* es un método que permite medir la distancia entre un objeto y el sensor, basado en el tiempo que le toma a la señal ser emitida y regresar al dispositivo, como se ve en la Figura 3, después de ser reflejada por un objeto (TERABEE, 2018). Poseen una excelente estabilidad a largo plazo y son sumamente pequeños. Son usados en un sin número de lugares y con una gran cantidad de usos como reconocimiento de usuarios o gestos de las manos; en aspiradoras robots como guías de navegación para reconocer paredes y obstáculos. Incluso son usados en drones para reconocimiento de terreno (Eskow, 2018).

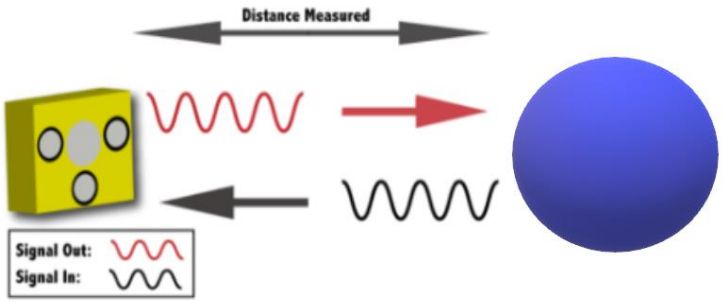


Figura 3. Funcionamiento de la tecnología de tiempo de vuelo (ToF). Tomado de (TERABEE, 2018).

2.1.4 Reconstrucción mediante tecnología de luz estructurada

Este tipo de tecnología permite la reconstrucción 3D mediante la aplicación de un rayo luminoso o patrones emitido hacia un objeto y el posterior análisis de la deformación que haya sufrido en la proyección, mostrado así en la Figura 4. Los datos son obtenidos mediante el uso de diferentes imágenes para mantener una mayor calidad de la reconstrucción (Silva & Zapata, 2015).

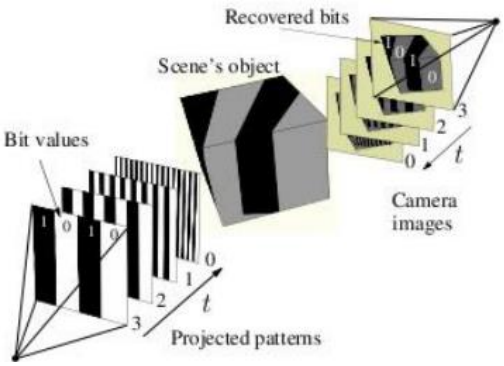


Figura 4. Funcionamiento de reconstrucción 3D mediante luz estructurada Tomado de (Salido, 2013).

2.2 Dispositivos comerciales para reconstrucción en tres dimensiones

2.2.1 Hokuyo YVT-35LX Scanning Rangefinder

Este dispositivo también denominado 3D Lidar permite medir distancias mediante un rayo láser vertical y horizontalmente, el cual usa el principio de *ToF*. Una señal láser es emitida en un amplio campo 3D, siendo su *field of view* de 210° en horizontal y 40° en vertical, como se ve en la Figura 5. Logrando así obtener una nube 2590 puntos, proveyendo de altura, anchura y profundidad de los objetos. Tiene un alcance de detección de 35 metros al frente y 14 metros a los lados. En caso de ser necesario se puede incrementar la densidad de los muestreos hasta 20 veces horizontalmente y 10 veces verticalmente independientemente. Este modo se denomina *High Density* (HD) logrando un total de 518000 puntos o más. (Hokuyo Automatic, 2014).

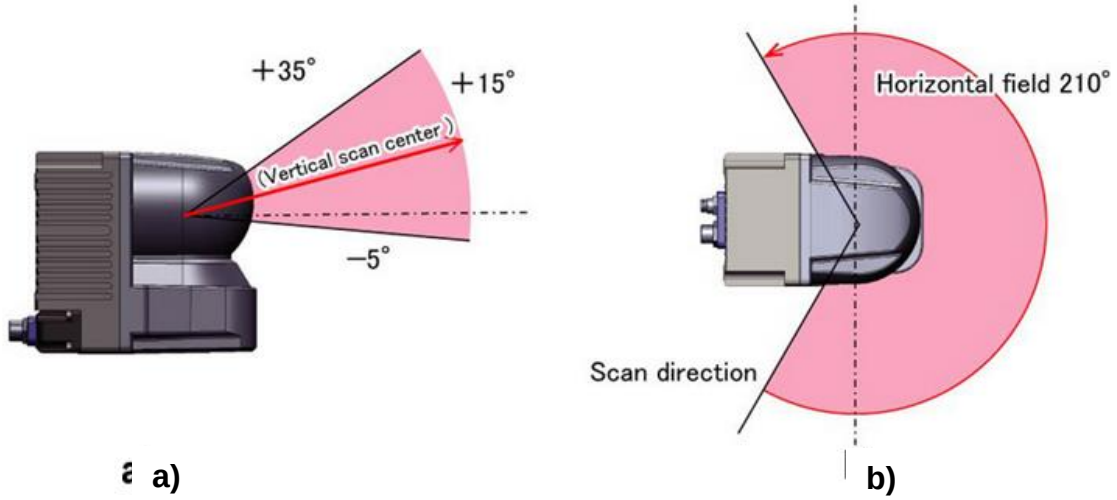


Figura 5. Campo de vista del sensor Hokuyo YVT-35LX
Adaptado de (Hokuyo Automatic, 2014).
a) Vista vertical.
b) Vista horizontalmente.

Cuando se usa en exteriores, la lluvia, el polvo y la neblina pueden generar múltiples señales de regreso. Para lo cual la función de eco múltiple permite separar la lluvia y la niebla y así el sensor informará la distancia de las primeras cuatro señales en ser recibidas, a pesar de que puede procesar hasta un total de 8 (Hokuyo Automatic, 2014).

2.2.2 SlamTec RPLIDAR A2 M8

Es un dispositivo que permite realizar un escaneo láser en 360 grados y en dos dimensiones gracias al uso de la tecnología LIDAR, es capaz de tomar hasta 8000 muestras láser por segundo con una alta velocidad de rotación. Y gracias a su tecnología OPTMAG obtiene un tiempo de vida mucho mayor a cualquier tecnología LIDAR. El sistema es capaz de realizar escaneos con un rango de alcance de 12 metros. Los datos de la nube de puntos obtenida pueden ser usados para mapear, localizar y modelar ambientes u objetos, como se observa en la Figura 6 (SLAMTEC, 2018).

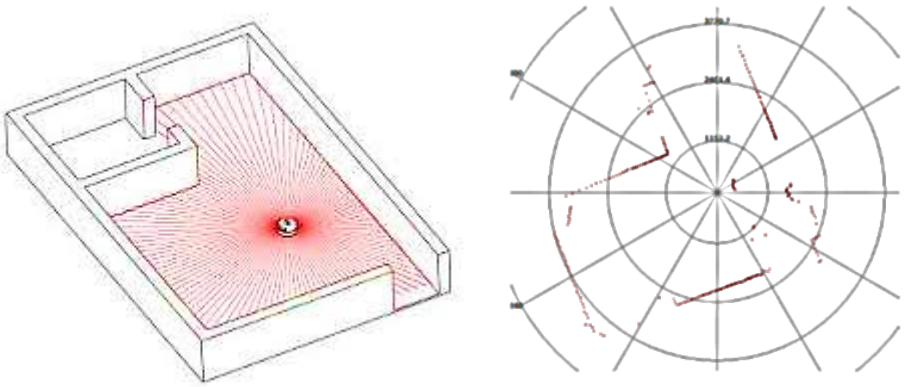


Figura 6. Entorno mapeado mediante el uso del escaneo del RPLIDAR. Adaptado de (SLAMTEC, 2018).

La frecuencia del RPLIDAR A2 es de 10 Hz permitiendo obtener 600 rpm, y así una resolución de alrededor de 0.45°. Su frecuencia puede ser ajustada entre 5 y 15 Hz dependiendo de los requerimientos de los usuarios (SLAMTEC, 2018).

El RPLIDAR A2 adopta el sistema de medición de bajo costo de triangulación láser desarrollado por SLAMTEC, el cual permite obtener un excelente desempeño tanto en ambientes interiores como exteriores sin exposición directa de luz de sol (SLAMTEC, 2018).

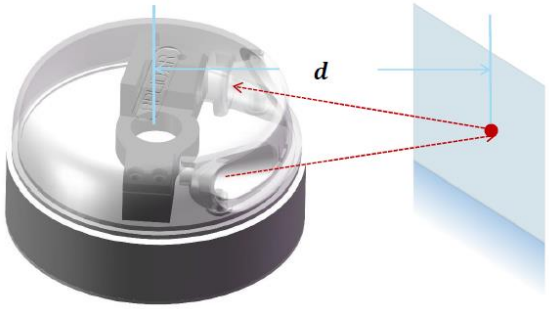


Figura 7. Esquema de trabajo del RPLIDAR
Adaptado de (SLAMTEC, 2018).

Durante el proceso de escaneo el RPLIDAR emite una señal láser infrarroja modulada y así la señal láser es reflejada por el objeto para poder detectarlo, mostrado en la Figura 7. La señal devuelta es procesada por el sistema de adquisición de visión y el DSP integrado en el RPLIDAR, el cual comienza a procesar los datos obtenidos de la muestra y obtiene la distancia y ángulo entre el objeto y el dispositivo (SLAMTEC, 2018).

2.2.3 Waymo System

Es un sistema usado normalmente en vehículos autónomos y cuenta con un total de 5 sensores Lidar, una cámara en 360 grados y un sistema de radar. La combinación de estos sensores permite obtener una reconstrucción en 360 grados en tiempo real con una alta resolución como se ve en la Figura 8 y la identificación de objetos pequeños como piernas, brazos, pelotas, etc. Gracias al uso de la cámara se puede obtener una imagen a color del lugar. Permitiendo identificar diferentes objetos como semáforos y tipos de señales en el camino (Waymo, 2018).

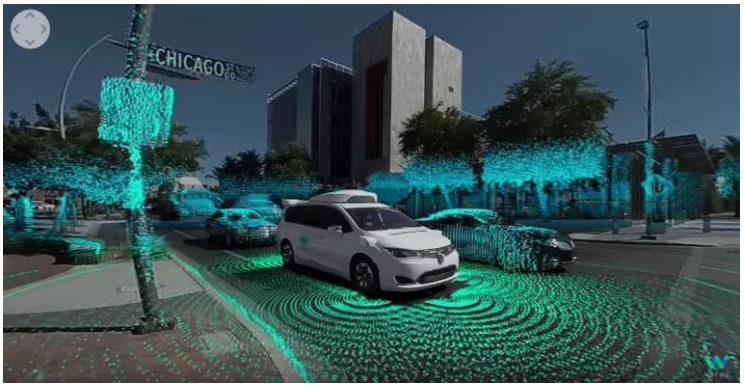


Figura 8. Reconstrucción del entorno obtenido por *Waymo sensors*. Adaptado de (Waymo, 2018).

El uso del radar es contemplado con la finalidad de detectar objetos grandes y a una mayor distancia. Inclusive este sistema cuenta con un limpiador en caso de ser necesarios para la carcasa del dispositivo y así obtener muestras claras del entorno. El uso de software permite la integración de las diferentes tomas en una sola y gracias a la gran cantidad de sensores puede obtener un alcance de detección de hasta 300 metros (Waymo, 2018).

2.2.4 ZED Stereo Camera

ZED es una cámara que reproduce la manera en que trabaja la visión humana. Mediante el uso de sus dos “ojos” y triangulación logra proveer una escena en tres dimensiones de lo observado, existe dos tipos entre ellos ZED y ZED-M los cuales se pueden observar en la Figura 9 (StereoLabs, 2018).

El dispositivo ZED es una cámara con lentes duales, el cual permite capturar una imagen o video 3D en alta definición con un amplio campo de vista. Su percepción de profundidad es la habilidad para determinar distancias entre los objetos y ver así el mundo en tres dimensiones.

a) **b)**

Figura 9. Dispositivos ZED.
Adaptado de (SLAMTEC, 2018).
a) Dispositivo ZED Mini.
b) Dispositivo ZED.

Generalmente, los sensores de profundidad se encuentran limitados por su alcance y pueden generar una gran cantidad de ruido al estar bajo luz solar directa, mientras que, la tecnología de estéreo visión, gracias al uso de cámaras, puede ser usado tanto en interiores como exteriores y con un gran alcance contrariamente a sensores basados en luz estructurada y tiempo de vuelo. Mediante el uso del sensor de profundidad ZED se puede obtener un alcance de 20 hasta 30 metros en su modo ultra. Y puede llegar a capturar entre 15 y 100 FPS, dependiendo de la calidad de la reconstrucción. Además de mantener un campo de visión hasta 90° horizontal x 60° en vertical (StereoLabs, 2018).

ZED inclusive entiende su posición y orientación en el espacio, ofreciendo full *6DOF* o seis grados de libertad. Permite un mapeo espacial el cual es la habilidad de capturar un modelo digital de una escena o un objeto en el mundo físico. ZED continuamente es usado en ambientes para reconstruir mapas en tres dimensiones del mundo real (StereoLabs, 2018).

2.2.5 Sensor Kinect v1.

El sensor Kinect, originalmente denominado proyecto Natal, es un dispositivo desarrollado por Microsoft e introducido al mercado en el año 2010. Con la finalidad de reemplazar la manera convencional con la cual se controla un juego. Entre sus partes se encuentra una cámara de color, un sensor de profundidad y un conjunto de micrófonos. Este dispositivo provee una interfaz natural de

usuario o NUI el cual puede detectar los movimientos del cuerpo, los gestos, comandos hablados y reconocimiento facial (Demitseva, 2015).

Figura 10. Partes del Kinect v.1.
Adaptado de (Demitseva, 2015).
1) emisor infrarrojo
2) cámara RGB
3) sensor de profundidad infrarrojo
4) motor de inclinación
5) arreglo de micrófonos.

Computer Vision Society rápidamente considero que se podría usar el Kinect para mucho más que videojuegos y a un costo mucho menor que las tradicionales cámaras 3D. Siendo así en 2011 Microsoft lanza el *Software Development Kit* para el Kinect permitiendo así que pueda ser usado para herramientas no comerciales. (Andersen et al., 2012).

El Kinect v1 está compuesto por un sensor de imagen a color (CMOS), un sensor Infrarrojo (CMOS), una fuente de luz infrarroja (IR) y un chip PS1080 como se ve en la Figura 10. La fuente de luz infrarroja proyecta imagen de IR codificada en la escena. Cuyo patrón es posteriormente recibido por el sensor IR procesada por el PS1080 gracias al uso de un algoritmo de reconstrucción 3D basado en la imagen (Oliver, Kang, Wünsche, & MacDonald, 2012).

Entre las características de sus componentes se encuentra con una cámara de color de 640x480 pixeles con una tasa de refresco de 30Hz, una cámara de profundidad de 320x240 pixeles con una tasa de refresco de 30Hz. Cuenta con un alcance máximo de 4.5 m y un campo de vista de 57 grados en el campo horizontal y 43 grados en el campo vertical y un motor de inclinación. El Kinect v.1 puede reconocer alrededor de 20 partes del cuerpo y el movimiento de los mismos hasta un máximo de 2 personas (Demitseva, 2015).

2.2.6 Sensor Kinect v.2

Siendo la segunda versión del dispositivo, cuenta con una cámara de 1920x1080 con una tasa de refresco de 30 Hz, un rango de operación entre 0.5 y 4.5 metros, un sensor de profundidad con un resolución de 512x424 a 30Hz, un campo de visión de 70° en horizontal y 60° en vertical como se ve en la Figura 12 (Varanasi & Devu, 2016) (Sell & Connor, 2014). El sensor Kinect v.2 consiste en tres sistemas diferentes trabajando como uno solo en el SoC o (*System-on-a-chip*) denominado PrimeSense Capri (PS1200). El cual ejecuta sofisticados algoritmos computacionales paralelos para producir una imagen con profundidad. Entre estos sistemas se encuentra una cámara RGB y un sensor de profundidad el cual consiste en un láser infrarrojo combinado con un sensor monocromático (CMOS), el cual logra capturar una imagen independientemente de las condiciones de luz que exista en el lugar. (Popescu & Lungu, 2014) (Sell & Connor, 2014).

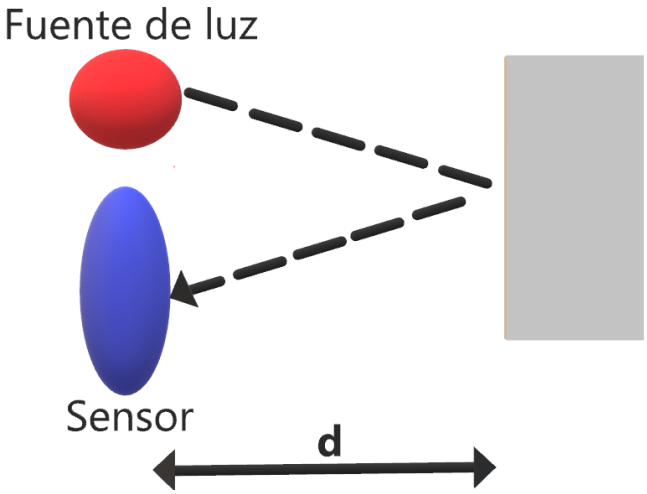


Figura 11. Principio de operación de *Time of Flight*.
Adaptado de (Valgma, 2016).

Este sensor cuenta con 3 fuentes de luz infrarroja, cada una generando una onda modulada con diferente amplitud. Para poder capturar las ondas reflejadas el Kinect v2 cuenta con cámara infrarroja como se ve en la Figura 11. Cada pixel cuenta con dos diodos (A, B) los cuales son controlados por la misma señal de reloj el cual controla la modulación de onda. Estos fotodiodos convierten la luz capturada en corriente, la cual puede ser medida. Estos fotodiodos son manejados por la señal de reloj de la siguiente manera, si $A = [a_i]$ está encendida, $B = [b_i]$ está apagada y viceversa (Valgma, 2016).

- Entonces $([a_i] - [b_i])$ muestra la correlación entre la luz obtenida y la señal de reloj y puede ser usada para obtener la información de fase. (Imagen de profundidad).
- $([a_i] + [b_i])$ provee de una imagen regular en escala de grises iluminada por luz ambiental. (Imagen de ambiente).
- $\sqrt{\sum i\,([a_i] - [b_i])^2}$ provee de una escala de grises la cual es independiente de la luz ambiental (Imagen activa) (Valgma, 2016).

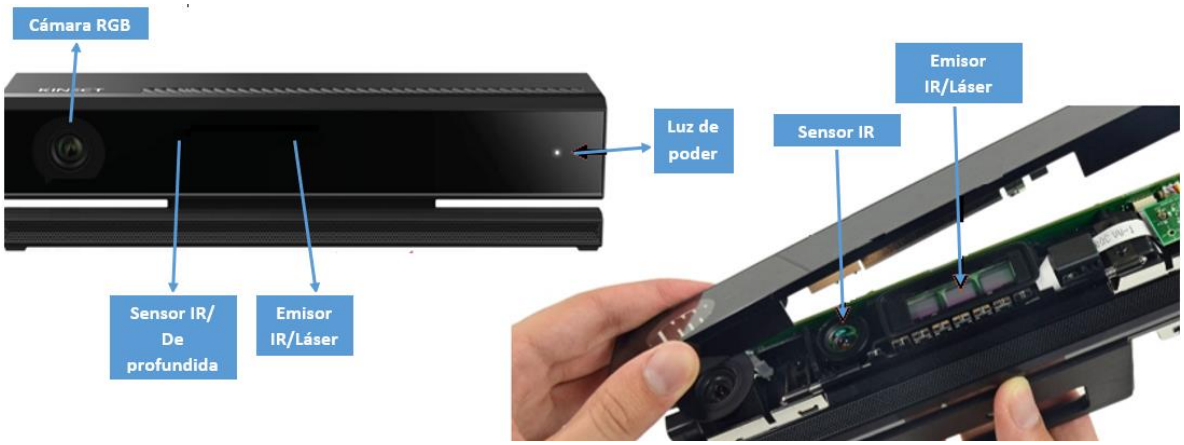


Figura 12. Partes de Kinect v2.
Adaptado de (Valgma, 2016).

El filtro de pasa banda estrecha o *Narrowband-pass filter* es usado para bloquear toda la luz excepto aquella del rango de longitud de onda 860 nm, el cual corresponde a la longitud de onda del sistema de iluminación infrarrojo. Kinect posee un motor de obturador múltiple el cual combina los datos de varios obturadores y elige el mejor valor para cada píxel. Siendo utilizado el tiempo de obturación más largo el cual no cause saturación. Este motor también normaliza todos los valores relativos al tiempo de obturación más largo. (Valgma, 2016).

El Kinect v.2 usa tecnología *ToF* para medir la distancia. El principio de esta operación se basa en medir el tiempo que le toma a la onda de luz viajar desde el emisor hacia el objeto y regresar nuevamente. Es decir, puede ser expresado como se ve en la ecuación 1, donde t_r y t_e representan el tiempo de la emisión y recepción del pulso de luz respectivamente y c es la velocidad de la luz. La distancia es calculada en base a la diferencia de fase de la onda de luz emitida y la onda de luz detectada reflejada por el objeto. La onda de transmisión tiene modulación de frecuencia $\omega = 2\pi f$. Entonces, la distancia recorrida por la onda modulada es $2d$ lo cual produce un cambio de fase φ. El cambio de fase depende de la diferencia en el tiempo, como se ve en la ecuación 2 (Valgma, 2016).

$$d = \frac{t_r - t_e}{2} * c \qquad \text{Ecuación 1}$$

$$\varphi = \omega t_r - \omega t_e = \omega(t_r - t_e)$$ Ecuación 2

Para poder capturar el cambio de fase, se requieren al menos de dos mediciones ya que existen dos parámetros desconocidos. Para esto la señal $\beta\ sin(\omega t - \varphi)$ se mezcla con una versión de fase retrasada de la señal de referencia.

$$R_o(t) = sin(\omega t - \varphi_{offset})$$ Ecuación 3

Lo cual resulta en:

$$\mathrm{R}(t) \cdot R_o(t) = \beta \sin(\omega t - \varphi) \cdot \sin(\omega t - \varphi_{offset})$$ Ecuación 4

$$= 0.5\beta\ (cos(\omega t - \varphi - \omega t + \varphi_{offset}) - cos(\omega t - \varphi + \omega t - \varphi_{offset}))$$
$$= 0.5\beta\ cos(\varphi - \varphi_{offset}) - 0.5\beta\ cos(2\omega t - \varphi - \varphi_{offset}).$$

Este resultado finalmente pasa por un filtro de pasa bajas, lo cual remueve el segundo sumando, finalizando como:

$$LP\ [R(t) \cdot R_o(t)] = 0.5\beta\ cos(\varphi - \varphi_{offset}).$$ Ecuación 5

Usando diferentes valores para φ_{offset}, se puede estimar el cambio de fase φ independientemente de la amplitud de la señal β. El sensor Kinect usa tres cambios de fase (0°,120° y 240°) (Fossati, Gall, Grabner, Ren, & Konolige, 2013)(Andersen et al., 2012) (Sarbolandi, Lefloch, & Kolb, 2015). Dado que obtener la distancia se basa en el cambio de fase de la onda modulada, la distancia máxima medible depende la longitud de onda de la onda modulada. La fase es alrededor de 360° o 2π. Esto implicaría que usando longitud de onda mayor permitiría obtener un mayor alcance. Usar una menor longitud de onda permite obtener una mejor resolución. Con la finalidad de obtener una resolución alta y también medir mayores distancias, Kinect usa tres frecuencias diferentes (16 MHz, 80MHz, y 120 MHz). (Valgma, 2016).

2.3 Fundamentos matemáticos

2.3.1 Matrices de rotación

Estas matrices definen algebraicamente la rotación en un espacio en tres dimensiones considerando el ángulo en el cual está girando. Algunas de sus propiedades son que: sus ejes son dos vectores ortogonales; su determinante es 1; es una matriz unitaria, es decir, el resultado al obtener la normal de cualquier vector que pertenezca a esta matriz será uno; debido a que es una matriz ortogonal su transpuesta será igual a su inversa (Sistemas Robustos, 2018). Este tipo de matrices son el método más extendido para describir la orientación, gracias a la facilidad proporcionada por el álgebra matricial (Barrientos, 2007). Y pueden ser expresadas como se puede ver en las ecuaciones 6, 7, 8 y Figura 13:

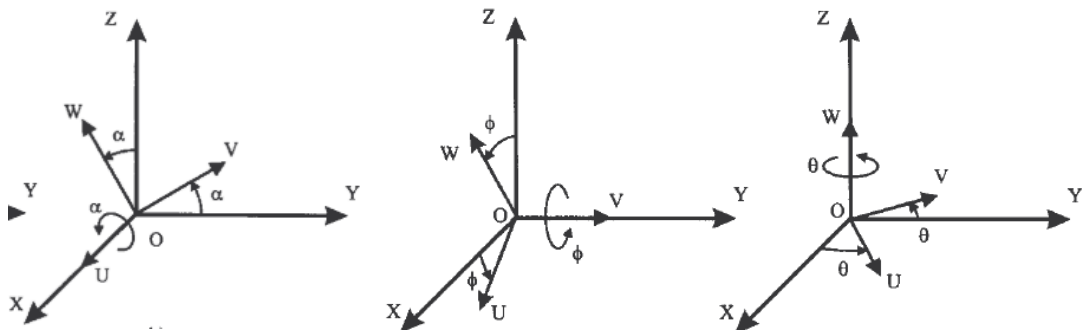


Figura 13. Rotación del sistema OUVW con respecto a los ejes OX, OY y OZ. Tomado de (Barrientos, 2007).

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \qquad \text{Ecuación 6}$$

$$R_y(\phi) = \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix} \qquad \text{Ecuación 7}$$

$$R_Z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$ Ecuación 8

2.3.2 Ángulos de Euler

Para poder representar la orientación en un espacio tridimensional mediante la matriz de rotación se requiere definir elementos nuevos. Todo sistema OUVW puede ser definido como un sistema OXYZ mediante tres ángulos (ϕ, θ, ψ), denominados ángulos de Euler. Al girar repetidamente en el sistema OXYZ sobre unos ejes determinados en los valores de (ϕ, θ, ψ), se obtendrá un sistema OUVW. Es por esta razón que se requiere conocer los ángulos y los ejes en los que se realizan los giros (Barrientos, 2007).

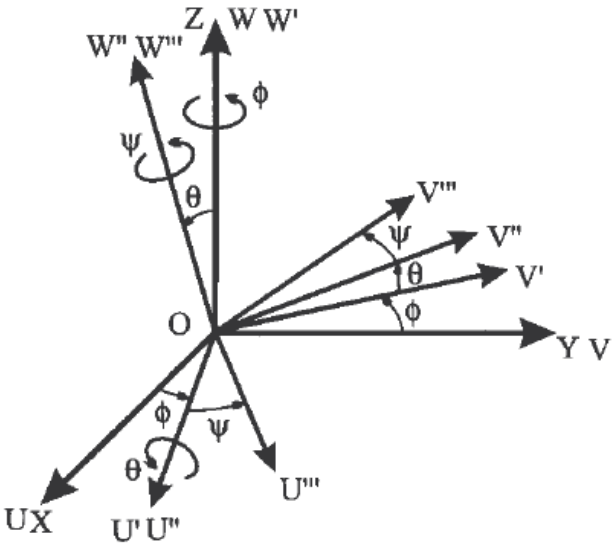


Figura 14. Ángulos de Euler ZXZ.
Tomado de (Barrientos, 2007).

Es conocido como r*oll, pitch y yaw* la representación de los giros en un sistema fijo, en la cual, partiendo de los sistemas OXYZ y OUVW, se puede colocar el sistema OUVW en cualquier orientación siguiendo los pasos a continuación:

1. Girar el Sistema OUVW un ángulo ψ con respecto al eje OX. Es conocido como *Yaw*.
2. Girar el Sistema OUVW un ángulo θ con respecto al eje OY. Es conocido como *Pitch*.
3. Girar el Sistema OUVW un ángulo ϕ con respecto al eje OZ. Es conocido como *Roll*.

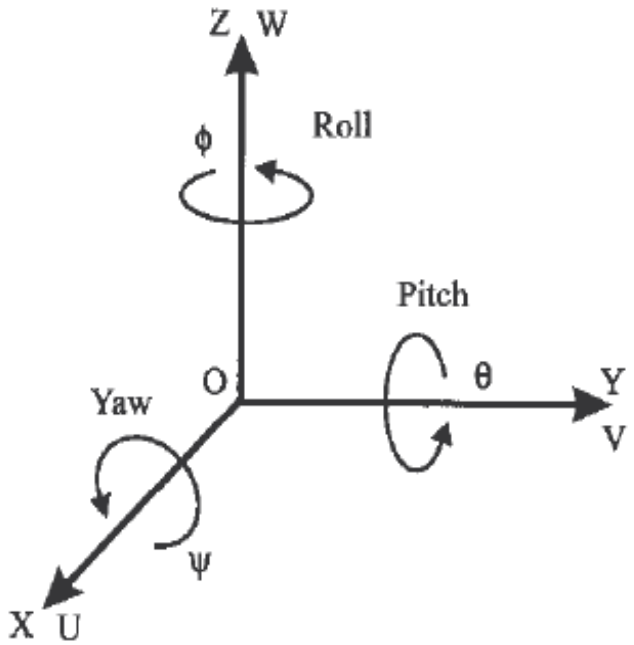


Figura 15. Ángulos de Euler: Roll, pitch, yaw.
Tomado de (Barrientos, 2007).

Para cada uno de ellos es necesario considerar que no se trata de una transformación conmutativa, siendo necesario seguir una secuencia determinada de aplicación de los mismos (Barrientos, 2007).

2.3.3 Coordenadas y matrices homogéneas

La localización de sólidos en un espacio n-dimensional mediante coordenadas homogéneas se realiza a través de coordenadas de un espacio (n+1)-dimensional. Entonces, un espacio n-dimensional se encuentra representado en coordenadas homogéneas por (n+1) dimensiones. De esta manera, un vector

dado $p(x, y, z)$ será representado por $p(wx, wy, wz, w)$, siendo w un valor que representa la escala además de ser un valor arbitrario. Visto de una manera general un vector $p(a\boldsymbol{i}, b\boldsymbol{j} + c\boldsymbol{k}), en\ el\ cual$ i, j, k son lo vectores unitarios de los ejes OX, OY, OZ del sistema de referencia OXYZ, es representado en coordenadas homogéneas mediante el vector columna:

$$p = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} aw \\ bw \\ cw \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix}$$

(Barrientos, 2007).

En base a la definición de las coordenadas homogéneas se obtiene inmediatamente el concepto de matriz de transformación homogénea (T). La cual está definida por una matriz 4x4, la cual representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro. (Barrientos, 2007).

$$T = \begin{bmatrix} R_{3x3} & r_{3x1} \\ f_{1x3} & W_{1x1} \end{bmatrix} = \begin{bmatrix} Rotación & Traslación \\ Perspectiva & Escalado \end{bmatrix}$$

Una matriz homogénea se encuentra compuesta por cuatro submatrices de diferente tamaño: una submatriz R_{3x3} y de r_{3x1} considerándose que las componentes de f_{1x3} es 0 y la de W_{1x1} la unidad. Al ser una matriz 4x4, los vectores en cuales será aplicada deberán contar con 4 dimensiones, que serán las coordenadas homogéneas del vector tridimensional (Barrientos, 2007).

Siendo de esta manera la matriz homogénea resultará de la siguiente forma:

$$T = \begin{bmatrix} R_{3x3} & r_{3x1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Rotación & Traslación \\ 0 & 1 \end{bmatrix}$$

Esta representa la orientación y posición de un sistema O'UVW rotado y trasladado con respecto al sistema de referencia OXYZ. Esta matriz permite conocer las coordenadas (r_x, r_y, r_z) del vector r en el sistema OXYZ a partir de sus coordenadas (r_u, r_v, r_w) en el sistema O'UVW (Barrientos, 2007).

$$\begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} = \boldsymbol{T} \begin{bmatrix} r_u \\ r_v \\ r_w \\ 1 \end{bmatrix}$$

Una matriz de transformación homogénea puede ser usada para:

- Representar la posición y orientación de un sistema girado y trasladado O'UVW con respecto a un sistema fijo de referencia OXYZ, que es lo mismo que representar una rotación y traslación realizada sobre un sistema de referencia.
- Transformar un vector expresado en coordenadas con respecto a un sistema O'UVW, a su expresión de coordenadas del sistema de referencia OXYZ.
- Rotar y trasladar un vector con respecto a un sistema de referencia fijo OXYZ.

(Barrientos, 2007).

2.4 Procesamiento de datos

2.4.1 Nube de puntos

Una nube de puntos o también denominado *Point Cloud* es la representación de un conjunto de puntos en un espacio tridimensional de los elementos físicos captados por un dispositivo. Estos puntos representan la posición X, Y, Z de la superficie del objeto. La cantidad de puntos en la escena será responsable de la resolución de la representación. En el caso del Kinect v.2 cada punto obtenido del entorno es asociado con un valor RGB el cual, proviene de la cámara de color (Navarro, 2017). Existen varios algoritmos para manejar nubes de puntos los cuales difieren de la cantidad de datos obtenidos. Estos métodos pueden variar por el tipo de correspondencia usado, método de cálculo de movimiento y robustez (Valgma, 2016). Un punto en tres dimensiones es representado de la siguiente manera:

$$P_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}$$

Siendo una nube de puntos la representación de puntos relacionados:
$P = \{pi \mid i = 1, \ldots, N\}$, donde N es el número de puntos.

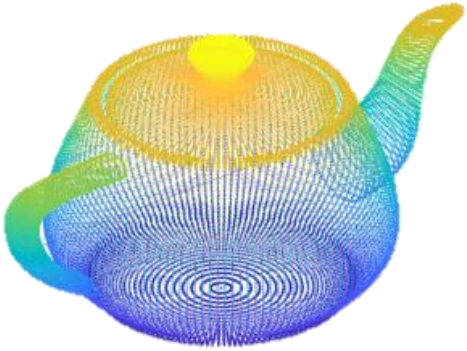


Figura 16. Ejemplo de nube de puntos.
Adaptado de (MATLAB, 2018).

2.4.2 Técnicas de filtrado para nubes de puntos

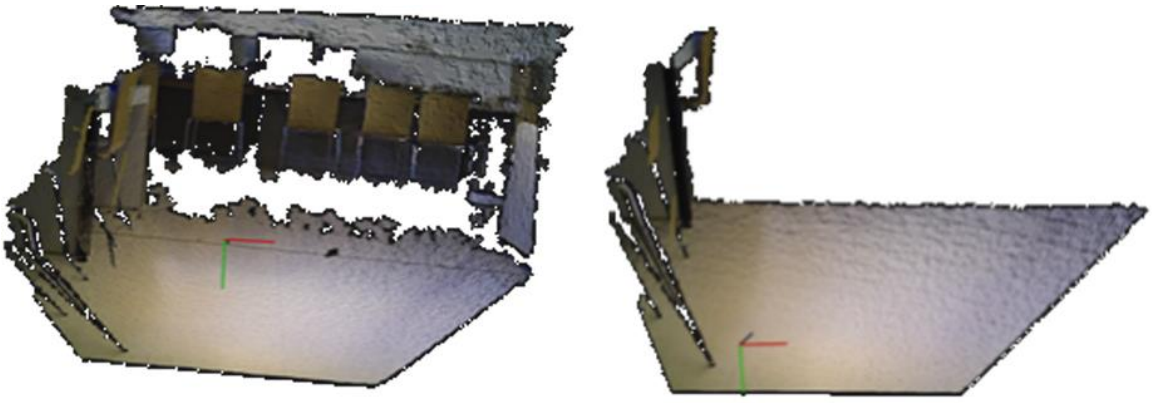
Al tratar con nubes de puntos existen muchas razones por las cuales se requieren filtros, entre ellas se encuentra una densidad irregular, oclusiones, grandes cantidades de datos y ruido. Para ello existen distintos tipos de filtros con la finalidad de remover o modificar puntos de un *Point Cloud,* los cuales serán descritos a continuación.

2.4.2.1 *PassThrough filter*

Este tipo de filtro remueve cualquier punto de la nube cuyos valores no se encuentren dentro de un rango específico proporcionado por el usuario. Como por ejemplo en el caso de que se requiera eliminar cualquier punto a una distancia mayor a 3 metros se filtrara la nube dentro del plano Z entre 0 y 3 como se puede observar en la Figura 17. Este filtro puede ser sumamente útil para desechar objetos innecesarios de la nube de puntos. Siempre y cuando sea

modificado dependiendo de cada sensor y el plano en que se requiera (Grupo de robótica Universidad de León, 2015).

Adentro de este filtro se puede encontrar el algoritmo *Random SAmple Consensus* o *RANSAC*, es un algoritmo matemático que permite separar un conjunto de datos que contienen *outliers* o bordes, tratando a todos los datos como iguales, solo despreciando aquellos que se encuentren fuera en base a los límites dados (Flores & Braun, 2011).



a) **b)**

Figura 17. Filtro *PassThrough.*
Adaptado de (Grupo de robótica Universidad de León, 2015).
a) Nube de puntos original.
b) Nube de puntos después de aplicarse el filtro *passthrough* en el plano Z entre 0 y 2 metros.

2.4.2.2 *Downsampling using a VoxelGrid filter*

Muchas veces se requiere disminuir la cantidad de puntos obtenidos debido a que tiene una alta densidad, lo cual lo hace difícil de procesar. Para eso se crean filtros que permiten disminuir la cantidad de datos de las muestras. Entre los distintos tipos de filtros que pueden ser usados se encuentra VoxelGrid. Este filtro divide la muestra en pequeños cuadrados como se ve en la Figura 18; para disminuir la cantidad de puntos elimina los datos excepto aquellos puntos que se

encuentren en el centro de cada cuadro como se observa en la Figura 19 (PCL, 2011).

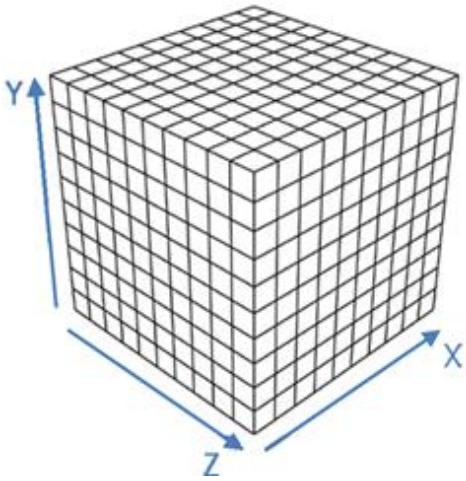
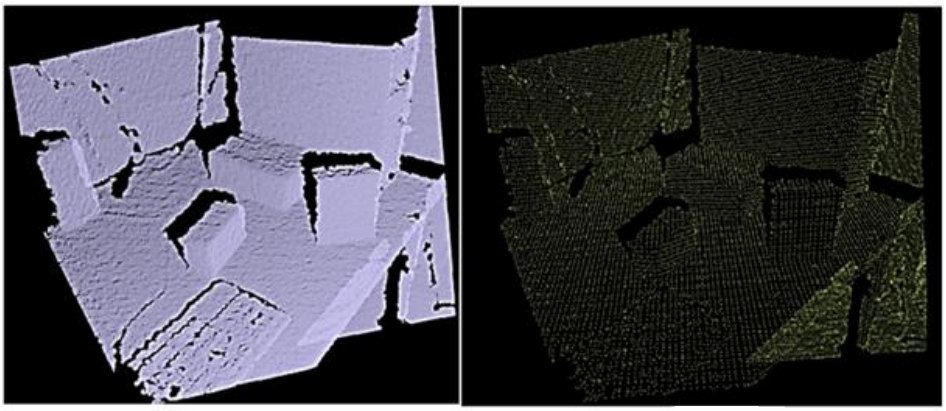


Figura 18. División de Voxel.
Adaptado de (PCL, 2011).



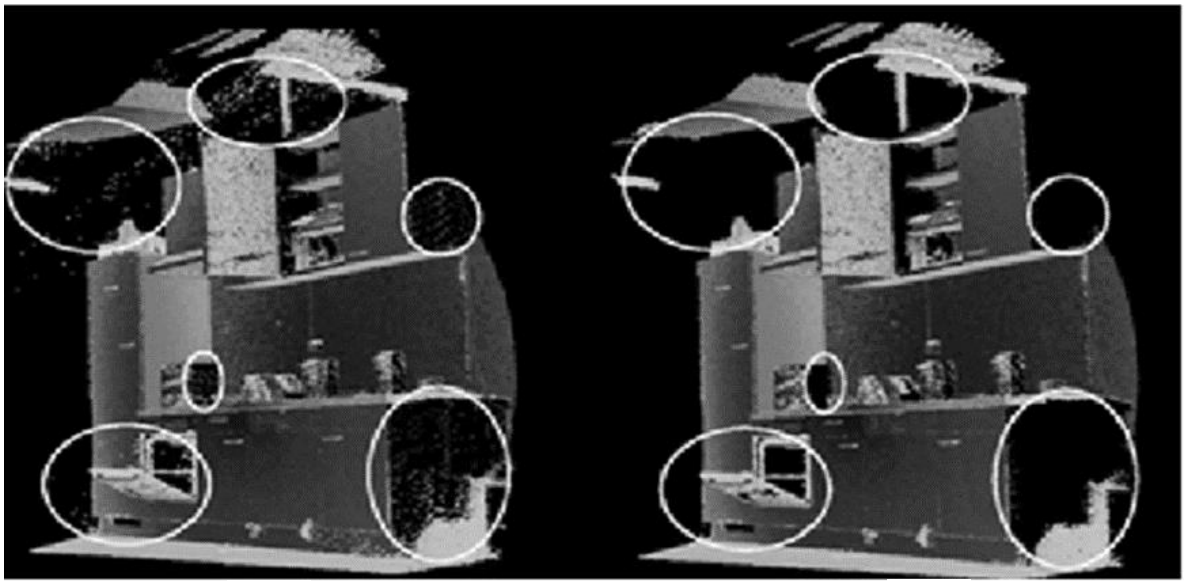
a) **b)**

Figura 19. Downsampling using a VoxelGrid filter.
Adaptado de (Grupo de robótica Universidad de León, 2015).
a) Nube de puntos original.
b) Nube de puntos después de realizar un *Downsampling* con *VoxelGrid.*

2.4.2.3 *StatisticalOutlier Removal filter*

El filtro *StatisticalOutlier Removal* es usado para remover puntos atípicos repartidos en la nube. Generalmente son el resultado de la inexactitud del sensor

al registrar datos en lugares que no debería haber ninguno. Es considerado como ruido no deseado ya que puede causar errores en los cálculos y resultar en datos no confiables, además aumenta el requerimiento de procesamiento debido a tener una mayor cantidad de puntos.



a) **b)**

Figura 20. StatisticalOutlier Removal filter

Tomado de (PCL, 2011).

a) Nube de puntos original.

b) Nube de puntos después de aplicarse el filtro S*tatisticalOutlier Removal.*

Para lograr eliminar estos puntos se toma en cuenta la distribución de los puntos y sus vecinos. En cada uno se calcula la distancia media con sus vecinos. Suponiendo que la distribución resultante es gaussiana con una media y desviación estándar, todos los puntos que se encuentren fuera del intervalo definido por la media global y la desviación estándar pueden ser considerados como valores atípicos y recortados del conjunto de datos como se ve en la Figura 20 (PCL, 2011).

2.4.2.4 *RadiusOutlier Removal*

Al igual que el anterior, en este filtro se busca el número de vecinos mínimo dentro de un radio proporcionado por el usuario, con el que debe contar cada punto para evitar ser tomado como un punto atípico. Uno de los principales problemas de este filtro es que requiere una gran cantidad de procesamiento y tiempo en caso de que haya una gran cantidad de datos (PCL, 2011).

Como se puede ver en la Figura 21 en caso de que el usuario quiera eliminar cualquier punto que no tenga ningún vecino dentro del radio determinado, se eliminará únicamente el punto amarillo. Mientras que, en caso de requerirse que se elimine cualquier punto con menos de 4 vecinos en su radio se procederán a eliminar el punto amarillo y verde.

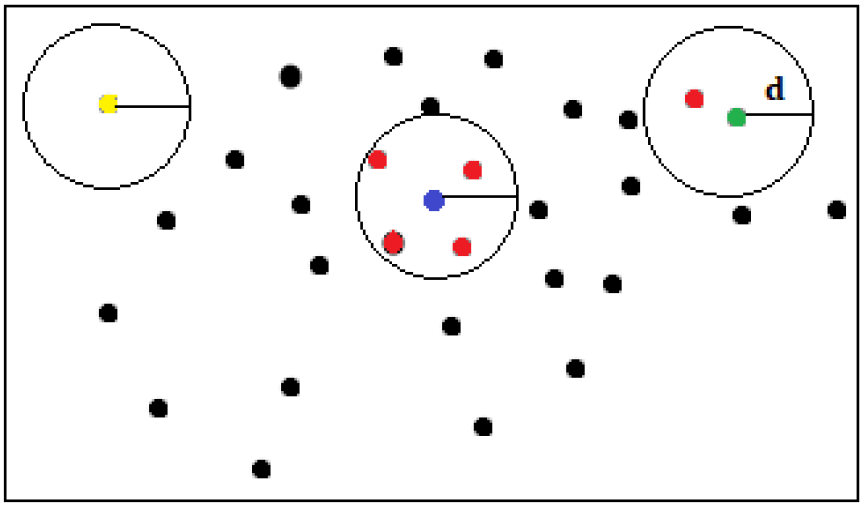


Figura 21. Funcionamiento de *RadiusOutlier Removal*.
Adaptado de (PCL, 2011).

3. Diseño e implementación

3.1 Metodología

Este proyecto se realiza mediante el uso de la metodología experimental, ya que se realizará el diseño y construcción de un prototipo que permita la

reconstrucción completa de un entorno en tres dimensiones mediante la obtención de una nube de puntos gracias al Kinect v.2.

Para lograr el cometido se usará: un motor a pasos, permitiendo que la plataforma gire; una tarjeta Arduino nano con un microcontrolador ATMega328, el cual recibirá las señales enviadas, un driver a4988 para controlar el movimiento del motor; un sistema mecánico que permitirá a la plataforma girar con mayor facilidad, una impresora 3D para crear la base; un Kinect v.2 para tomar las muestras del entorno; un adaptador de Kinect v.2 a PC con USB 3.0; un computador el cual permitirá el uso los datos obtenidos por el Kinect v.2; la herramienta Matlab r2018b la cual logrará la manipulación de las nubes de puntos obtenidas.

El sensor Kinect v.2 será montado sobre la plataforma donde procederá a adquirir los datos de la escena. Estos datos serán procesados para obtener una nube de puntos x, y, z en el sistema de coordenadas local del sensor. Posterior a esto se realiza una transformación de la nube de puntos en el sistema de coordenadas local al sistema de coordenadas global el cual puede ser apreciado en la Figura 22.

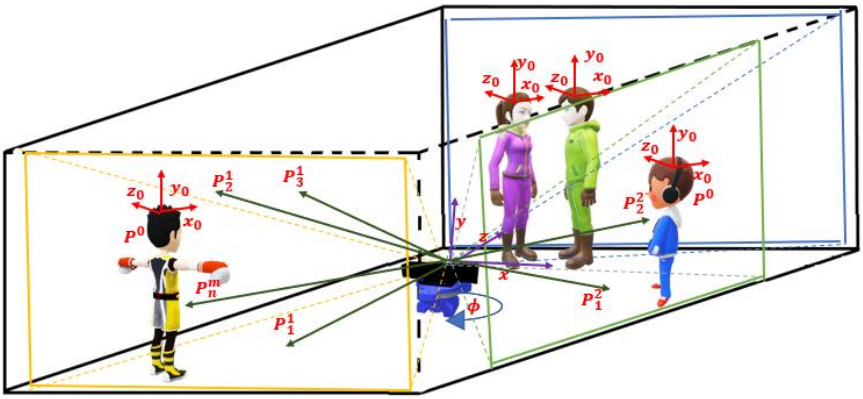


Figura 22. Ubicación de las coordenadas locales en función del Kinect v.2 (P_n^m) y coordenadas globales (P^0).

Una vez finalizada la transformación, la plataforma girará en el ángulo proporcionado por el usuario sobre su propio eje para obtener de esta manera

los datos de la nueva escena, este proceso se repetirá hasta completar los 360° de rotación. Los datos adquiridos procederán a unirse con los datos del sistema de coordenadas global, con la finalidad de obtener la reconstrucción del entorno en tres dimensiones, esto se puede observar de mejor manera en la Figura 23.

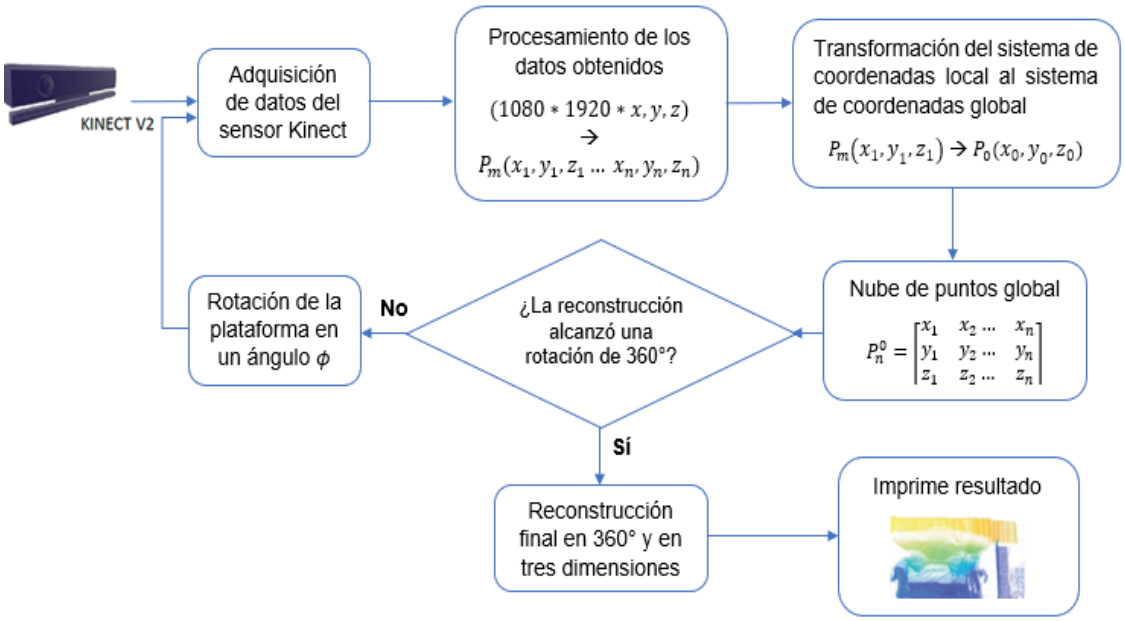


Figura 23. Diagrama del sistema de reconstrucción en tres dimensiones en 360° del sensor Kinect v.2.

3.2 Diseño de plataforma giratoria

3.2.1 Diseño e implementación

La reconstrucción en tres dimensiones se llevará a cabo gracias al uso de una plataforma la cual sea capaz de girar en 360 grados, gracias al uso de un motor a pasos. La plataforma debe ser capaz de soportar el peso del sensor Kinect y permitir la rotación de este en distintos ángulos según sea requerido.

Figura 24. Diseño de plataforma con sus diferentes partes.

Figura 25. Diseño e implementación de plataforma.
a) el montaje de estructura en la fase de diseño.
b) montaje de estructura final.

Como se puede analizar en la Figura 24, la plataforma consta de 6 partes diferentes las cuales serán creadas gracias al uso de una impresora 3D; un motor a pasos nema 17, el cual tiene un ángulo de paso de 1.8° o 200 pasos por vuelta, es bipolar y tiene un torque de 3.2 kg/cm; y un sistema mecánico el cual reduce la fricción entre el eje y las piezas conectadas a la plataforma. En la Figura 25 se puede observar el resultado final tanto en el diseño como en la implementación de la plataforma.

3.2.2 Manejo de la plataforma

La plataforma será controlada mediante el envío de una señal serial desde el software de Matlab hacia un microcontrolador ATMega328, el cual permite que el motor rote en 45°, 60° o 90°, dependiendo de las necesidades de la reconstrucción, gracias al uso de un driver a4988 como se puede observar en la Figura 26.

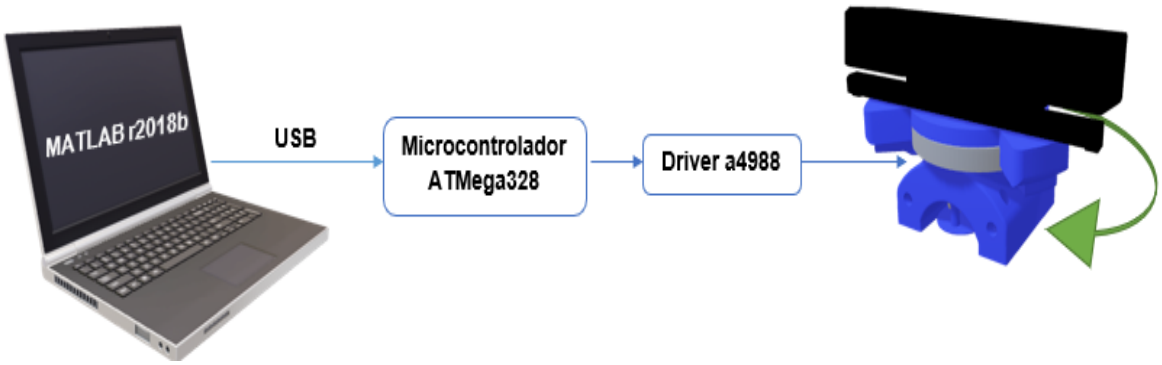


Figura 26. Diagrama de manejo de la plataforma.

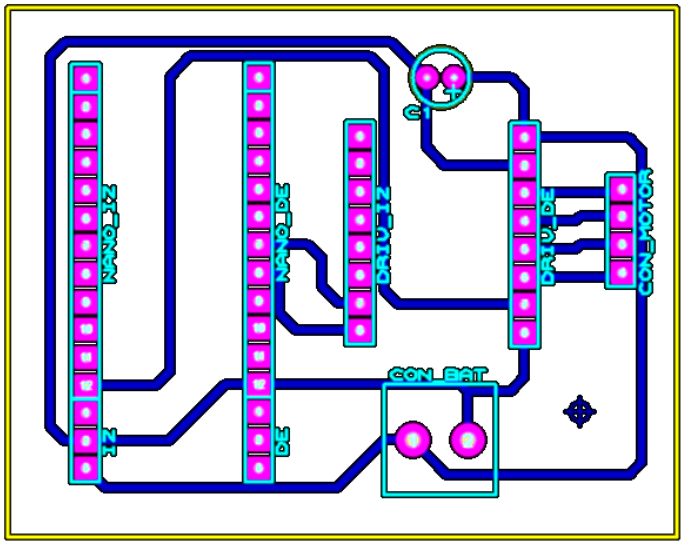


Figura 27. PCB de los dispositivos mencionados.

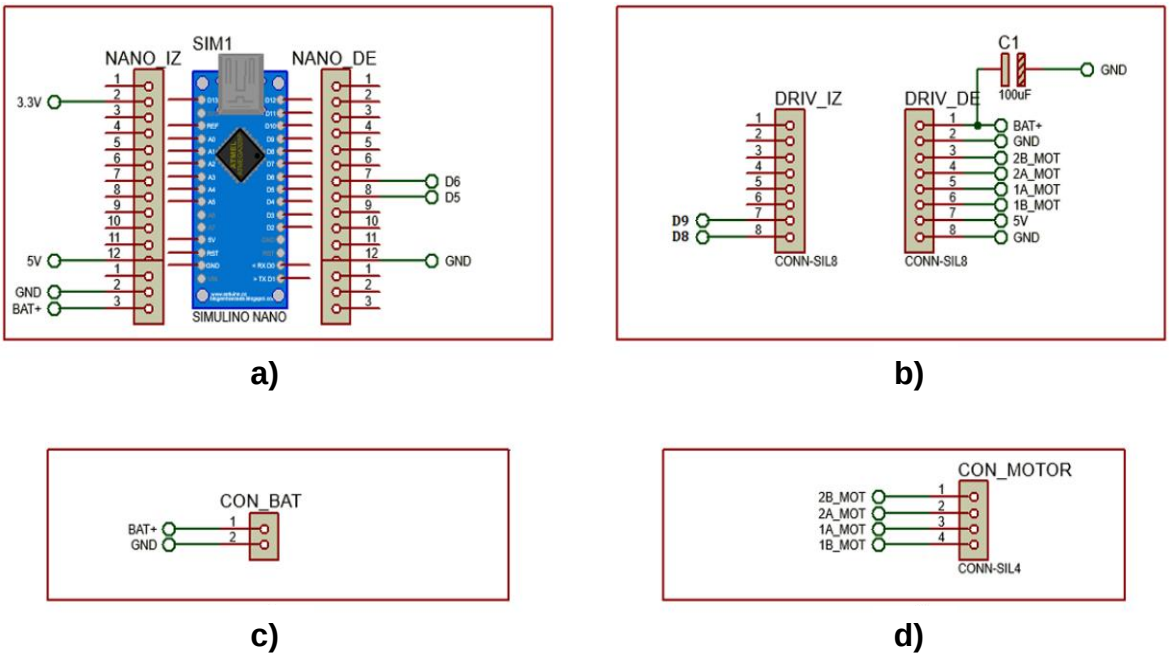


Figura 28. Diseño de diagramas de conexiones

a) Diagrama de conexiones de la tarjeta Arduino nano.

b) Diagrama de conexiones del driver a4988.

c) Diagrama de conexiones de la alimentación.

d) Diagrama de conexiones del motor.

3.3 Obtención y manejo de datos

3.3.1 Matlab, librerías y software

Matlab es un ambiente de computación numérica y lenguaje de programación desarrollado por *Mathworks*. Este sistema permite la implementación de algoritmos, creación de interfaces de usuario y funcionar junto otros lenguajes de programación como lo son C++, C#, Java, fortran y Python (MATLAB, 2018).

La plataforma de Matlab permite resolver problemas científicos y de ingeniería. Este lenguaje de programación se basa en matrices, y así, expresar las matemáticas computacionales es la forma más natural del mundo. Puede ser

usado para aprendizaje automático, procesamiento de señales, diseño de control, robótica, etc. (MATLAB, 2018).

Con el fin de obtener una nube de puntos del dispositivo Kinect v.2 se requiere de software y librerías. Una vez instalado el software *Matlab*, se procederá con la instalación de las librerías: *Image Acquisition Toolbox* la cual provee de funciones y bloques que permiten conectar diferentes dispositivos de obtención de imagen a Matlab. Inclusive permite detectar y configurar las propiedades del hardware. Esta librería habilita los modos de adquisición como *processing in-the-loop, hardware triggering, background acquisition y* adquisición sincronizada a través de múltiples dispositivos. (MATLAB, 2018). Al finalizar esta instalación procedemos a instalar la herramienta denominada *Image Acquisition Toolbox Support Package for Kinect For Windows Sensor.* Esta herramienta permite obtener los datos de la imagen del sensor directo en Matlab (MATLAB, 2018).

Para finalizar la instalación de la herramienta *Image Acquisition Toolbox Support Package for Kinect For Windows Sensor,* se debe llevar acabo la instalación de dos sistemas: *Kinect for Windows SDK v2.0*, este permite a developers crear aplicaciones que soporten gestos y reconocimiento de voz (Microsoft, 2014); y *Kinect for Windows Runtime, el* cual provee de drivers y un entorno de ejecución lo cual se requiere por diferentes aplicaciones que se encuentren usando la tecnología del sensor Kinect (Microsoft, 2014).

3.3.2 Tool para reconstrucción en tres dimensiones

Con la finalidad de obtener una reconstrucción en tres dimensiones, se creó una nueva herramienta capaz de procesar las diferentes capturas realizadas por el Kinect v.2 y desplegarlos como una sola nube de puntos, siendo esta la reconstrucción del entorno. El principal problema de esta herramienta es que el usuario debe tener una noción de programación en Matlab para poder hacer uso de esta. Con la finalidad de que cualquier usuario sea capaz de manejar la herramienta, teniendo o no conocimiento previo de programación, se realizó una nueva interfaz gráfica partiendo desde 0. Esta interfaz facilita el uso de la

herramienta hacia los usuarios; permite que cualquier persona que requiera usarla pueda hacerlo sin mayor problema e inclusive previene que los usuarios afecten de alguna manera el código del programa. La interfaz se diseñó en cinco partes las cuales serán descritas a continuación.

3.3.2.1 *Descripción general de la interfaz gráfica de la herramienta*

Abrir y guardar archivos

Se encuentra compuesta por dos botones: el botón “Guardar” y el botón “Cargar Archivo”. El primer botón desplegará una ventana que permitirá al usuario guardar la reconstrucción original obtenida previamente en un archivo “.mat” con el nombre proporcionado por el usuario e inclusive se puede exportar el resultado final como un archivo “.ply”; en caso de que no se haya realizado o cargado una reconstrucción se desplegará el mensaje “El archivo se encuentra vacío” como se puede ver en la Figura 29.

Figura 29. Mensaje de error al guardar un archivo sin que exista una reconstrucción en el sistema.

El segundo botón proporciona al usuario la capacidad de cargar el archivo de un entorno previamente guardado, con la resolución requerida por el usuario como se puede apreciar en la Figura 30. Para lograr esto se carga la reconstrucción guardada previamente y mediante el uso de la función “filtrar Resolución” la calidad de esta será modificada. Una vez finalizado el proceso se procede a mostrar el resultado; siendo el lado izquierdo la reconstrucción en tres dimensiones (x, y, z) y al lado derecho la reconstrucción del entorno con profundidad y color proporcionado por la imagen RGB obtenida.

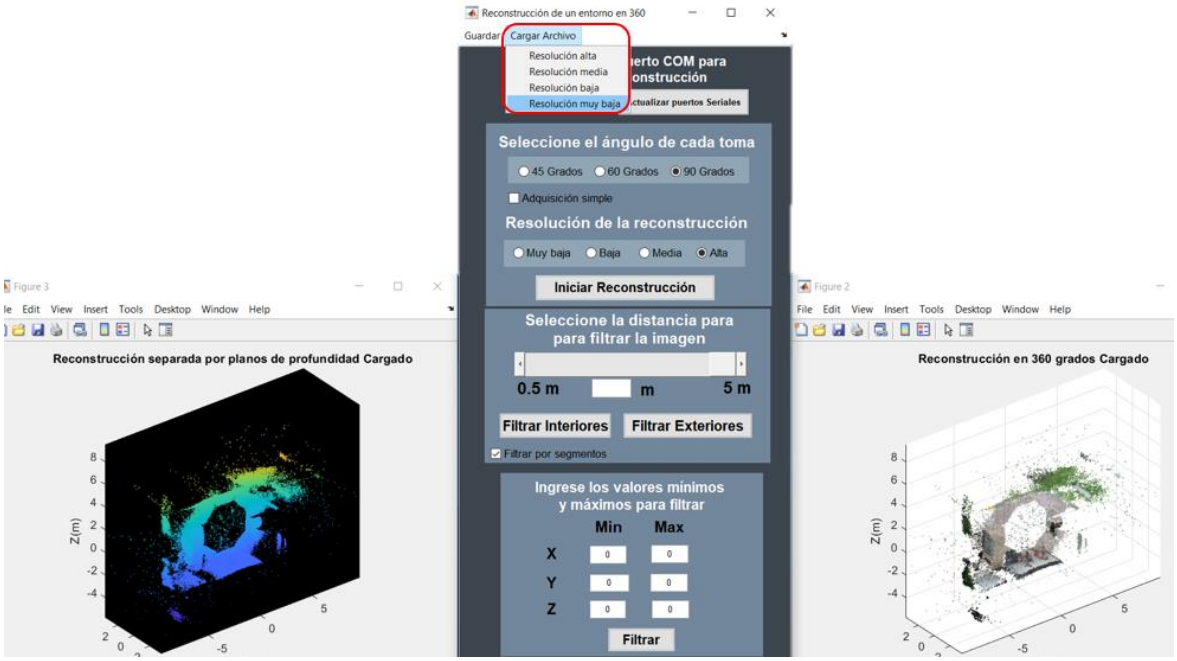


Figura 30. Cargar una reconstrucción previa, con una resolución muy baja.

Selección de puerto serial

Existe una lista en la cual se encuentran los puertos activos para la comunicación serial con el motor. En caso de que no se encuentre el puerto buscado se puede actualizar la lista mediante el uso del botón “Actualizar puertos seriales” como se observa en la Figura 31. Una vez seleccionado el puerto serial se habilitará la tercera parte de la interfaz gráfica.

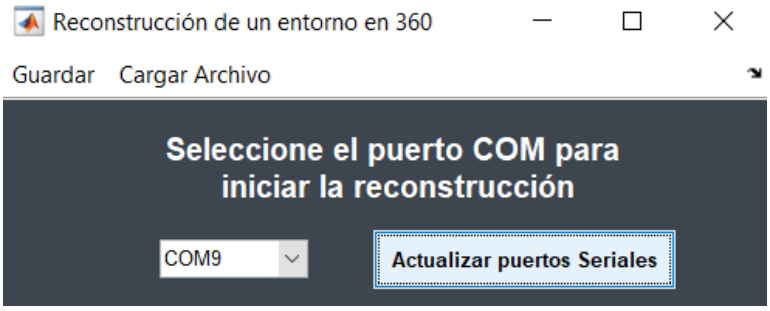


Figura 31. Actualización de los puertos seriales disponibles.

Preferencias de la reconstrucción

Aquí el usuario puede ingresar las preferencias con las cuales se realizará la reconstrucción. Inicialmente el usuario podrá decidir el ángulo en el cual rotará la plataforma para llevar a cabo la reconstrucción (45°, 60° o 90°), es decir, el número de tomas que se llevarán a cabo. Este número será dado al dividir 360° para el ángulo en que se decida realizar la toma. En caso de que el usuario lo requiera puede marcar la casilla “Adquisición simple” la cual realizará una sola toma del entorno frente al sensor. Al igual que al cargar el archivo, al iniciar la reconstrucción también se puede seleccionar la resolución de la reconstrucción entre alta, media, baja, muy baja. Una vez que el usuario haya seleccionado estas preferencias, se procede seleccionar el botón “Iniciar reconstrucción”. Este botón procede a llamar a una función denominada “fusión Kinect Global” con el valor del ángulo seleccionado y el puerto de comunicación seleccionado en la segunda parte. El resultado de la función es una nube de puntos en 360°, la cual contiene las coordenadas x, y, z y la matriz de color RGB resultante. Esta nube de puntos entonces será enviada a una función denominada “filtrar Resolución” junto con la resolución seleccionada por el usuario. Esto reducirá significativamente el número de puntos que tiene la reconstrucción final, sin afectar la misma. Al finalizar estos procesos satisfactoriamente se desplegarán los resultados a ambos lados de la interfaz, siendo el lado izquierdo la nube de puntos y al lado derecho la nube de puntos con la imagen RGB y a continuación el mensaje “Reconstrucción realizada con éxito”. Como se muestra en la Figura 32.

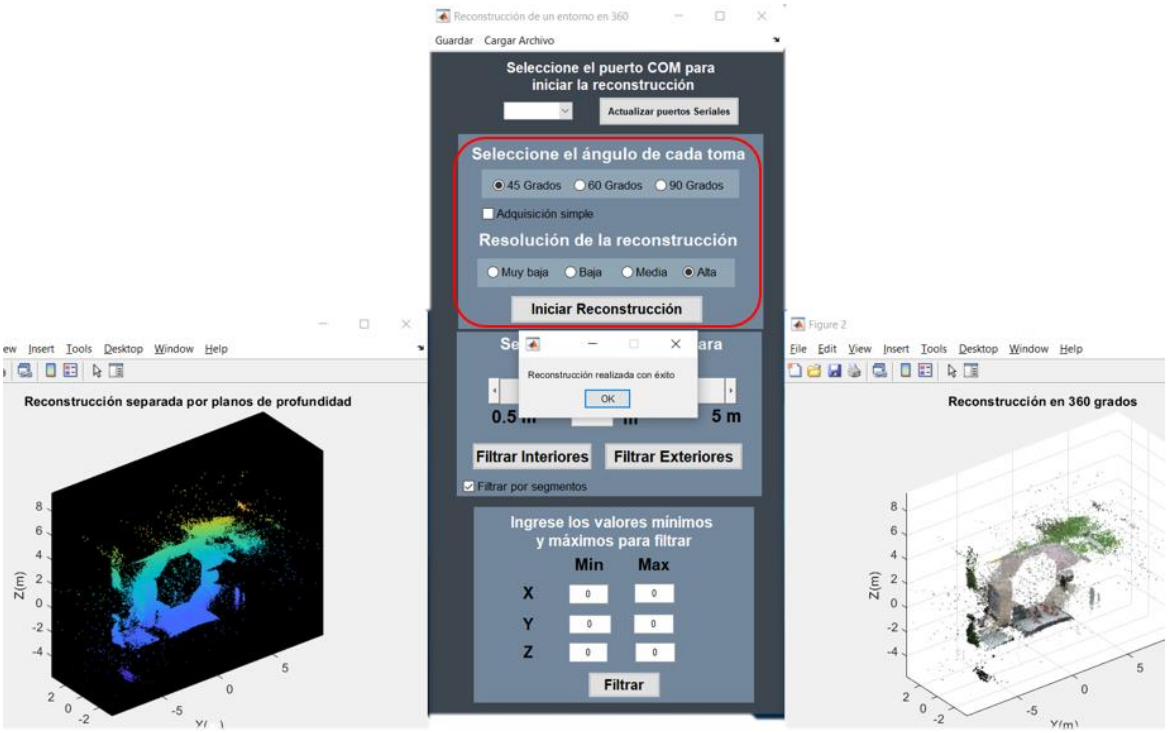


Figura 32. Reconstrucción realizada mediante la selección de las preferencias para iniciar la reconstrucción del entorno.

Filtrar exteriores e interiores

Esta sección se habilita únicamente al finalizar o cargar una reconstrucción satisfactoriamente. Contiene dos botones “Filtrar Exteriores” y “Filtrar interiores”. El primero de los botones permitirá proporcionar una distancia máxima a la nube de puntos obtenida, entre 0.5 y 5 metros mediante la función “filtrar Distancia Máxima”, es decir, eliminará todos los puntos que tengan una distancia mayor a la proporcionada desde el punto de vista del sensor. El segundo botón eliminará todos los puntos que sean menores a la distancia mínima proporcionada por el usuario, la cual puede estar entre 0.5 y 5 metros y lo hará mediante la función “filtrar Distancia Mínima. Para establecer la distancia, ya sea la distancia mínima o máxima, se puede desplazar un *slide* o inclusive se puede introducir el valor numérico en metros en el campo de texto. Estableciendo así un filtrado externo e interno de la nube de puntos, como se ve en la Figura 33. Al finalizar el filtro solicitado se procederá a desplegar dos imágenes al igual que en la tercera

parte, al lado izquierdo superior la imagen con profundidad a) y en el lado derecho superior b), la reconstrucción con profundidad y color proporcionado por la imagen RGB.

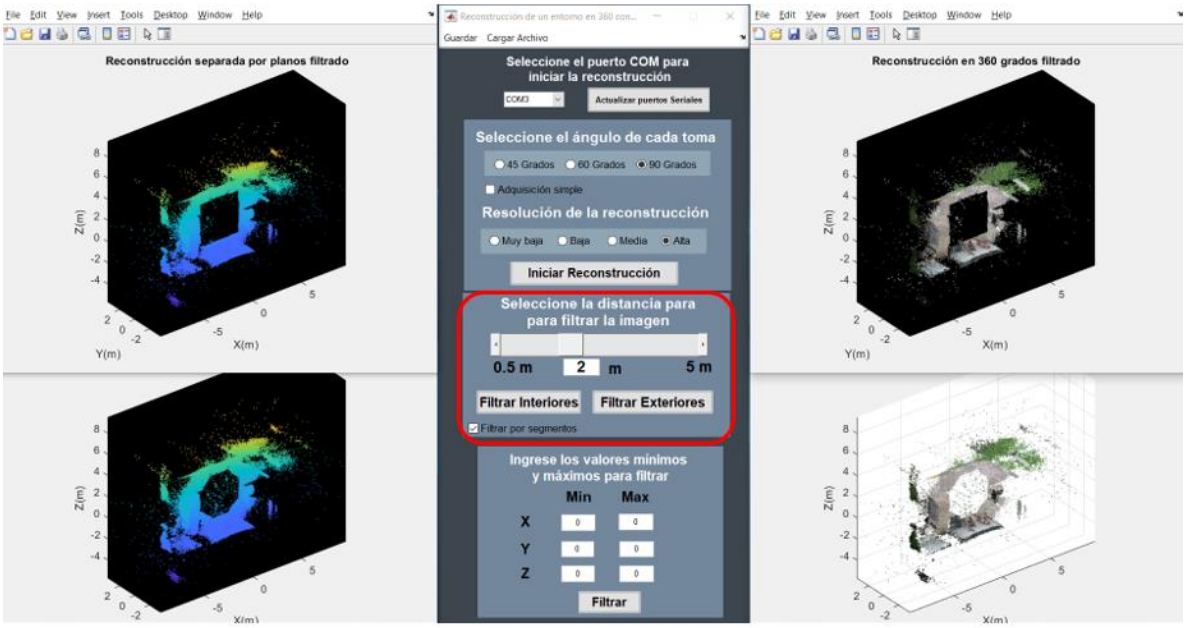


Figura 33. Resultado al filtrar con límites de distancia máxima de 5 metros y mínima de 2 metros.

Filtrado por segmentos

Finalmente, la quinta parte, la cual es habilitada al marcar la casilla “Filtrar por segmentos”, se podrá filtrar entre rangos máximos y mínimos en cada plano por separado como se puede observar en la Figura 34. Estos rangos no tienen límite por lo que dependerá completamente de los requerimientos del usuario. Los resultados serán mostrados en lugar de los resultados de la cuarta parte.

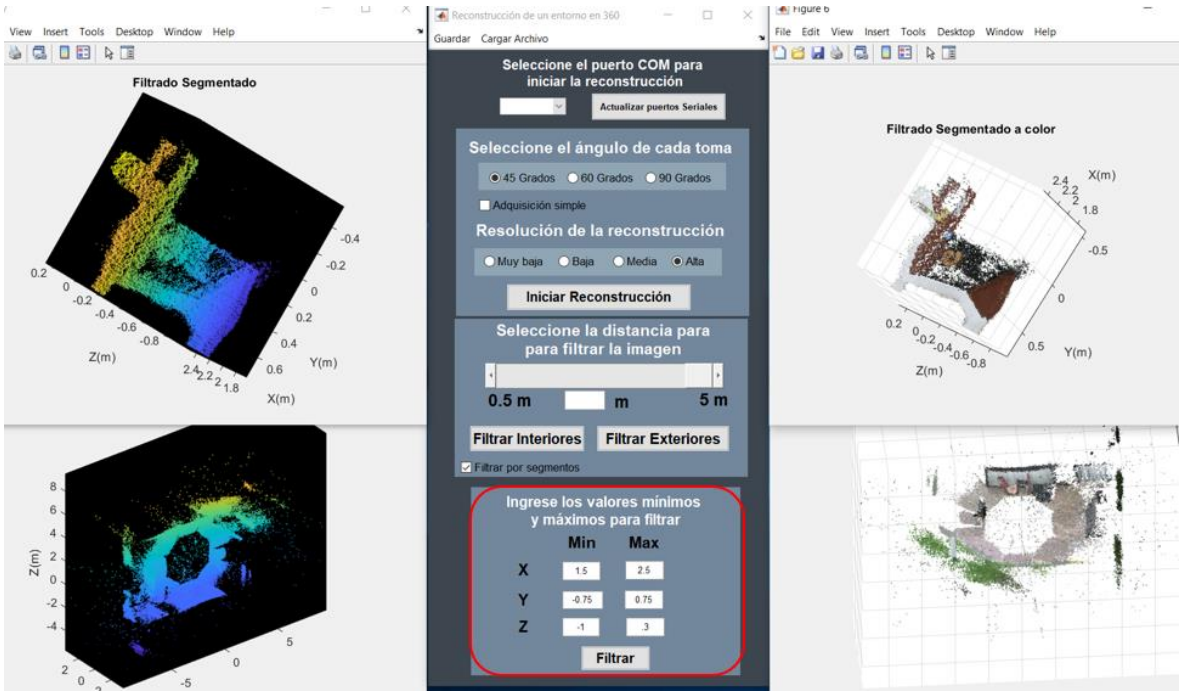


Figura 34. Filtrado de nube de puntos por rangos en cada plano.

3.3.2.2 *Función Fusión Kinect Global*

Esta función iniciará abriendo la comunicación con el puerto, el cual fue enviado desde la interfaz gráfica al igual que el ángulo de rotación. Una vez establecida la comunicación, el número de iteraciones o tomas que se realizarán será determinado por el ángulo de rotación seleccionado en la interfaz gráfica. En cada iteración se procederá a llamar a la función *"tomar Muestra"*. La función devuelve una nube de puntos con una matriz de coordenadas de los puntos obtenidos $P_n(x, y, z)$, dado de la siguiente manera: $P_n^m(x_1, y_1, z_1 \dots x_n, y_n, z_n)$, donde n es un punto obtenido y m la iteración en la que fue realizada la captura de los datos como puede ver en la Figura 37; y una matriz con los valores RGB en cada punto $C_n(R, G, B)$, representada de forma $C_n^m(R_1, G_1, B_1 \dots R_n, G_n, B_n)$ para realizar una representación de la nube de puntos con color. A continuación, se procede con la separación de las dos matrices de la nube de puntos para ser tratadas como dos variables separadas (localización y color), de las cuales se obtiene la transpuesta y se obtiene como resultado la ecuación 9.

$$P_n^m \begin{bmatrix} x_1 & x_2 \dots & x_n \\ y_1 & y_2 \dots & y_n \\ z_1 & z_2 \dots & z_n \end{bmatrix}; C_n^m \begin{bmatrix} R_1 & R_2 \dots & R_n \\ G_1 & G_2 \dots & G_n \\ B_1 & B_2 \dots & B_n \end{bmatrix};$$ Ecuación 9

Es necesario realizar la transformación del sistema de coordenadas del sensor Kinect v.2 al sistema de coordenadas Global, para así obtener una reconstrucción del entorno en 360°. Este proceso puede ser llevado a cabo gracias al uso de la ecuación 10. Donde P_0 es el sistema de coordenadas global; T_m^0 es la matriz homogénea como se ve en la ecuación 11; y P_m es la nube de puntos obtenida en la iteración dada. Al ser la matriz homogénea una matriz 4x4 se requiere que la nube de puntos con coordenadas x, y, z tenga un cuarto valor por lo cual se agrega un valor de 1, a cada conjunto de coordenadas (x, y, z) de la nube de puntos P_n^m, obteniendo como resultado la ecuación 12. Para obtener la matriz homogénea (T_m^0) se requiere de la matriz de rotación en el eje y $(R_{y(\phi)})$ la cual se encuentra representada en la ecuación 13, y de la matriz de traslación.

$$P_0 = T_m^0 * P_m$$ Ecuación 10

$$T_m^0 = \begin{bmatrix} R_{y(\phi)} & r_{3x1} \\ 0_{3x1} & 1 \end{bmatrix} = \begin{bmatrix} Rotación & Traslación \\ 0 & 1 \end{bmatrix}$$ Ecuación 11

$$P_n^m \begin{bmatrix} x_1 & x_2 \dots & x_n \\ y_1 & y_2 \dots & y_n \\ z_1 & z_2 \dots & z_n \\ 1 & 1 \dots & 1 \end{bmatrix}$$ Ecuación 12

$$R_y(\phi) = \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix}$$ Ecuación 13

Una vez reemplazados estos valores en la ecuación 11, obtenemos como resultado una matriz homogénea como se observa en la ecuación 14. El resultado de esta matriz varía en función del ángulo de rotación.

$$T_m^0 = \begin{pmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ Ecuación 14

Una vez que se transforman las coordenadas de un sistema de coordenadas al otro $(P_m \rightarrow P_0)$ el resultado es almacenado en una nueva variable de tipo *pointcloud*, junto con la matriz RGB como parte del color de la nueva variable, el valor de esta nube de puntos incrementará proporcionalmente con cada nueva iteración.

Al finalizar el proceso anterior, el motor recibe una señal lo que permite que rote en el ángulo determinado, en caso de que al finalizar el movimiento se logre completar la rotación de 360° se enviará la nube de puntos del entorno en 360° y en tres dimensiones a la interfaz gráfica; caso contrario se procederá a realizar una nuevamente el proceso desde la llamada a función “tomar Muestra”. Para una idea más clara del proceso se puede observar la Figura 23.

3.3.2.3 *Función Tomar Muestra*

La función “*tomar Muestra”* permite obtener los datos del sensor, estos datos son adquiridos mediante el principio *ToF* para adquirir una nube de puntos del entorno y una cámara que permite obtener los valores RGB para cada punto obtenido como se puede observar en la Figura 35. Los datos adquiridos contienen los siguientes parámetros: *Location* (es una matriz de 1080*1920*3 de variable tipo single la cual contiene la posición de cada punto tomado)*; Color (*es una matriz de 1080*1920*3 de variable tipo uint8 conteniendo los valores RGB en cada conjunto de coordenadas*); Count (el cual contiene el total de puntos en la nube); y XLimits, YLimits, ZLimits* siendo una matriz con los valores mínimos y máximos de cada plano. Estos datos se pueden observar en la Figura 36 a) y graficados mediante el uso de un *plot* 3D como se ve en la Figura 37 gracias al uso de pcshow.

Gracias al uso de la función *pcfitplane,* la cual usa el algoritmo *M-estimator SAmple Concensus (MSAC),* siendo una variación del algoritmo *RAMSAC* se permite obtener una nube de puntos en tres dimensiones con matrices bidimensionales como se puede observar en la Figura 36 b) y de igual manera eliminará los puntos que se encuentren vacíos, dejando una matriz limpia para ser procesada.

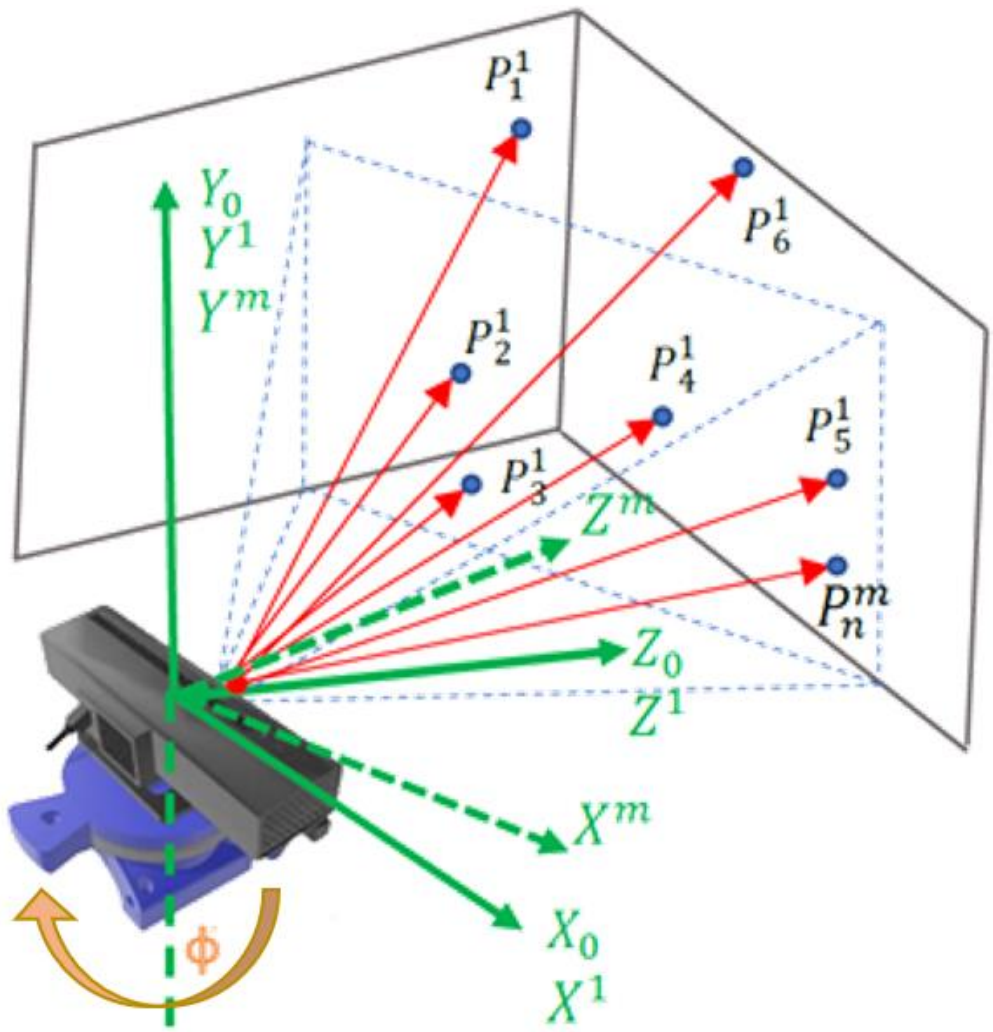


Figura 35. Adquisición de datos del Kinect v.2.

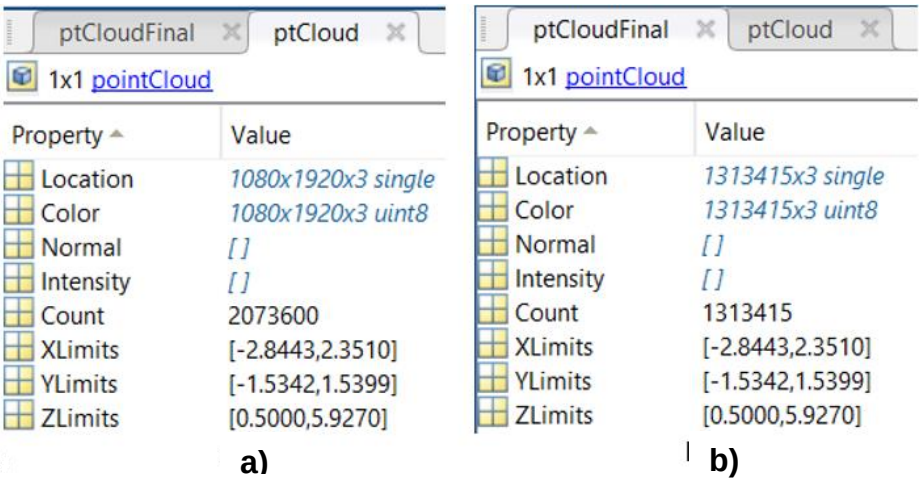


Figura 36. Uso de la función *pcfitplane.*

a) Nube de puntos original obtenida directamente del sensor Kinect v.2

b) Nube de puntos posterior al uso de la función *pcfitplane*.

Figura 37. Reconstrucción de una nube de puntos

a) Entorno real.

b) Reconstrucción de una nube de puntos obtenida por el sensor Kinect v.2.

3.3.2.4 *Filtros para nubes de puntos*

Función Filtrar Resolución

Al ser llamada, la función toma la nube de puntos y la resolución seleccionada por el usuario y gracias a un filtro denominado *Downsampling* que permite eliminar puntos con la finalidad de facilitar el procesamiento de los datos. Dependiendo de la resolución deseada se procederán a eliminar cada cierto

número de puntos, es decir, en el caso de que la resolución sea alta, la resolución de la imagen se mantendrá igual; en el caso de que se requiera cambiar la resolución a “media” se elimina 1 de cada 4 puntos obtenidos; en la resolución “baja” se elimina 1 de cada 2 puntos; mientras que en la resolución “muy baja” se eliminarán 3 de cada 4 puntos. De esta manera se cambia la resolución de la nube de puntos y permite un mejor manejo de esta. Un ejemplo de esto se puede apreciar en la Figura 38.

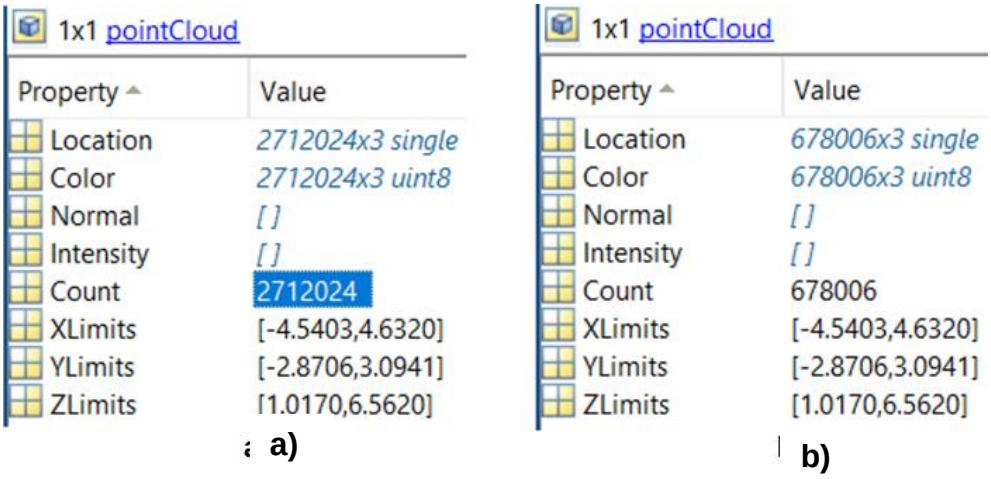


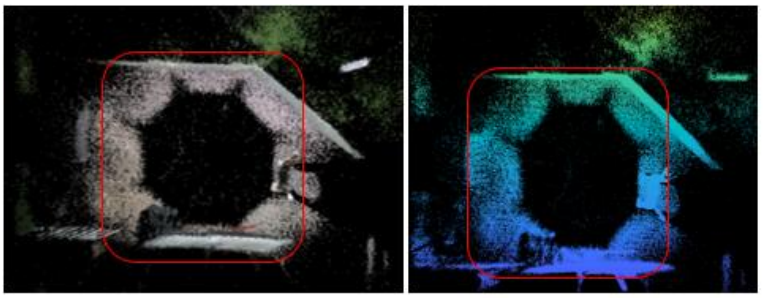
Figura 38. Función filtrar resolución.

a) Nube de puntos original.

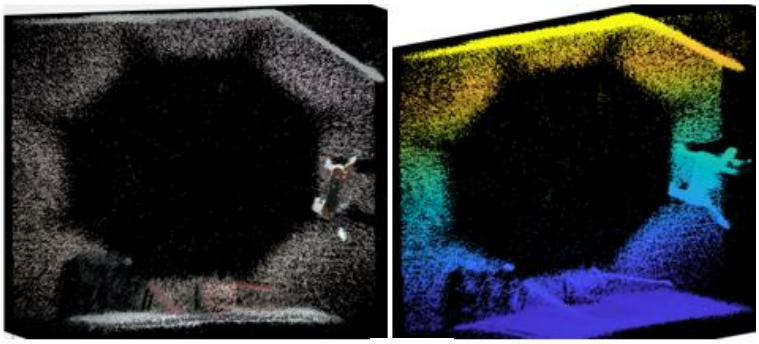
b) Cambio de resolución de nube de puntos a “muy baja”. Se eliminan 3 de cada 4 puntos.

Función Filtrar Distancia Máxima

Cuando se llama a esta función junto con la nube de puntos y un valor de la distancia máxima proporcionada por el usuario, se analiza cada punto y la distancia a la que se encuentra del sensor Kinect v.2. En el caso que esta distancia sea mayor a la distancia límite el punto será eliminado, la distancia es analizada en los 3 planos (x, y, z). Este tipo de filtro es conocido como *PassThrough filter.* Este filtro permitirá eliminar cualquier dato que se encuentre fuera de la distancia de análisis como se puede apreciar en la Figura 39.



a)



b)

c)

Figura 39. Función Filtrar Distancia Máxima

a) Nube de puntos original, en la cual se encuentra marcado el segmento que se requiere mantener.

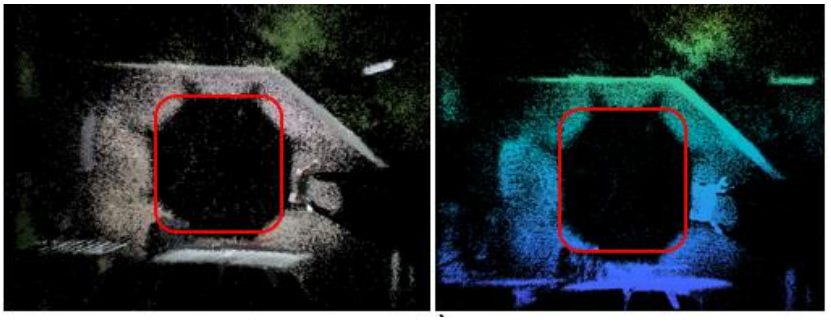
b) Nube de puntos filtrada a una distancia máxima de 3 metros.

c) Entorno real.

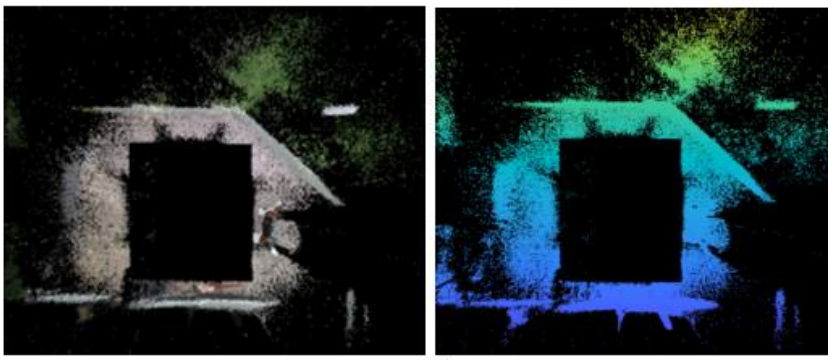
Función Filtrar Distancia Mínima

Con un funcionamiento similar a la función Filtrar Distancia Máxima, esta función recibirá una nube de puntos y un valor de la distancia mínima que fue seleccionada previamente por el usuario. Nuevamente se analiza la distancia a la que se encuentra cada punto del sensor y en caso de que esta distancia sea

menor a la proporcionada por el usuario el punto será eliminado. También conocido como *PassThrough filter*. El filtro permitirá eliminar cualquier objeto que se encuentre cercano al sensor, generalmente objetos muy cercanos que no logra reconstruir completamente generando ruido, como se puede ver en la Figura 40.



a)



b)

c)

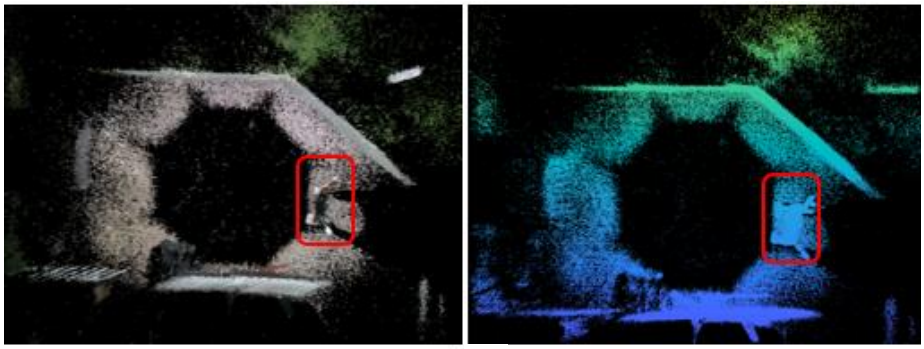
Figura 40. Función Filtrar Distancia Mínima

a) Nube de puntos original, en la cual se encuentra marcado el segmento que se requiere eliminar.

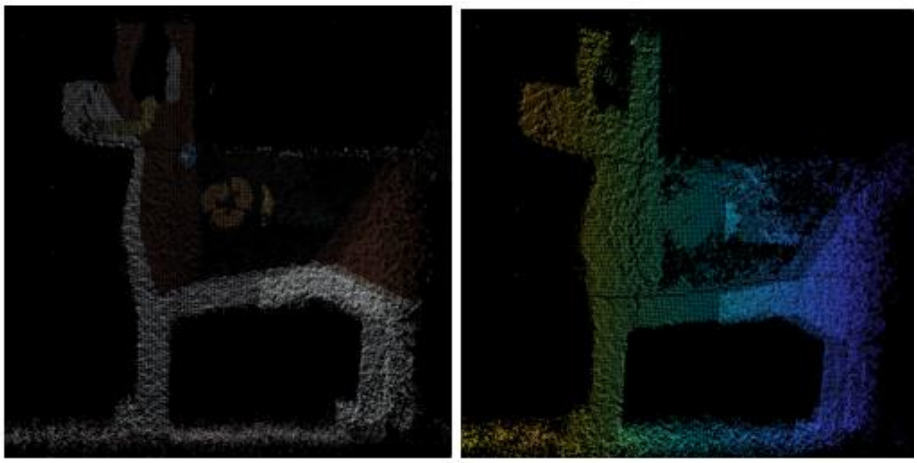
b) Nube de puntos filtrada a una distancia mínima de 2 metros.

c) Entorno real.

Función Filtrar Segmentos



a)



b)

c)

Figura 41. Función Filtrar Segmentos

a) Nube de puntos original, en la cual se encuentran marcados los segmentos que se mantendrán.

b) Nube de puntos filtrada por valores máximos y mínimos en cada eje, con los siguientes valores $x\ (2, 2.5), y\ (-0.5, 1.5), (-1, 0.5)$.

c) Entorno real.

Esta función será llamada junto con una nube de puntos y los valores máximos y mínimos en cada plano (x, y, z) en los que se requiere filtrar la nube de puntos. Este filtro es considerado *PassThrough filter* y permite que se elimine cualquier punto que se encuentre fuera del rango de valores dado; es decir, se establece un valor mínimo y máximo para cada plano x, y, z por separado, en caso de que el punto sea mayor al límite mínimo y menor al límite máximo, el punto no se eliminará. En caso de que no cumpla las condiciones dadas, el punto será descartado. Este proceso se repetirá en los tres planos y en cada punto. Obteniendo de esta manera una nube de puntos reducida, enfocada en el área de análisis como se muestra en la Figura 41.

4. Pruebas y resultados

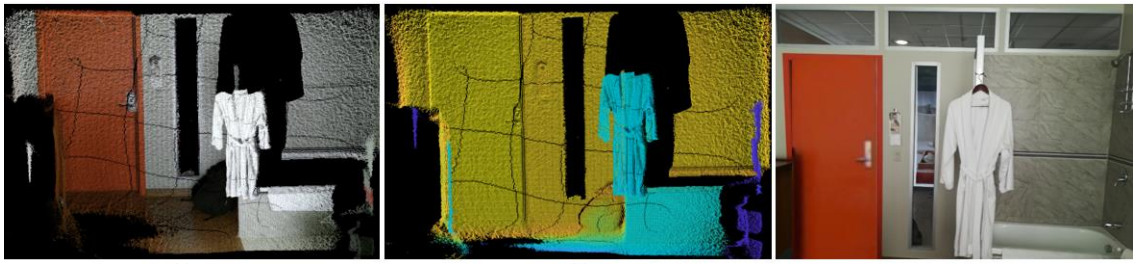
Una vez definido el funcionamiento del *tool* para reconstrucción de entornos en tres dimensiones, se procede a realizar pruebas en distintos ambientes para comprobar el correcto funcionamiento de la herramienta.

Las siguientes pruebas fueron realizadas en un ambiente controlado. Sin exposición de luz solar directa, variando la cantidad de luxes entre 65 y 215. Este ambiente se encuentra representado en la Figura 42.

Figura 42. Ambiente para pruebas de funcionamiento en interiores.

En las siguientes Figuras se puede apreciar una reconstrucción con un ángulo de rotación de 90° y una resolución “Alta”. En la Figura 43, 44, 45 y 46 se puede

observar la reconstrucción del entorno separada por secciones para poder ser apreciado correctamente. Se puede apreciar que este ángulo de rotación no es suficiente para obtener una reconstrucción completa del entorno, ya que el ángulo de visión del Kinect v.2 es de 70°. Esto causa una gran cantidad de vacíos en la reconstrucción final como se puede ver en la Figura 47, en la cual se encuentran la reconstrucción completa del entorno desde un punto de vista superior.



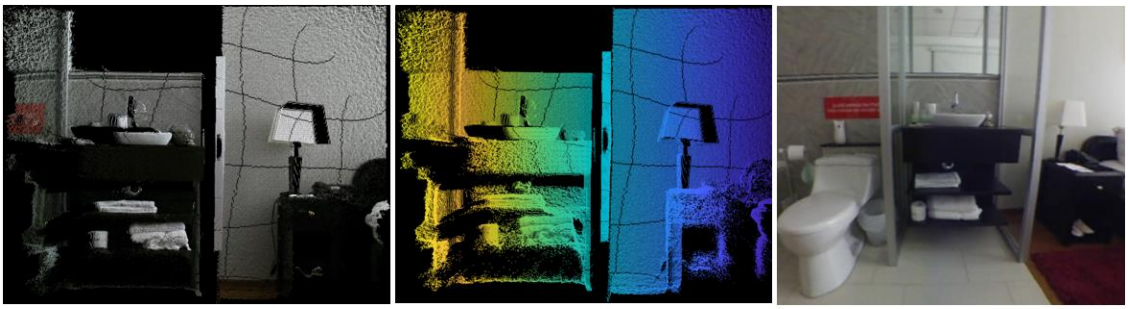
a) **b)** **c)**

Figura 43. Parte de la reconstrucción del entorno de pruebas, con rotación de 90°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.



a) **b)** **c)**

Figura 44. Parte de la reconstrucción del entorno de pruebas, con rotación de 90°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.

a)

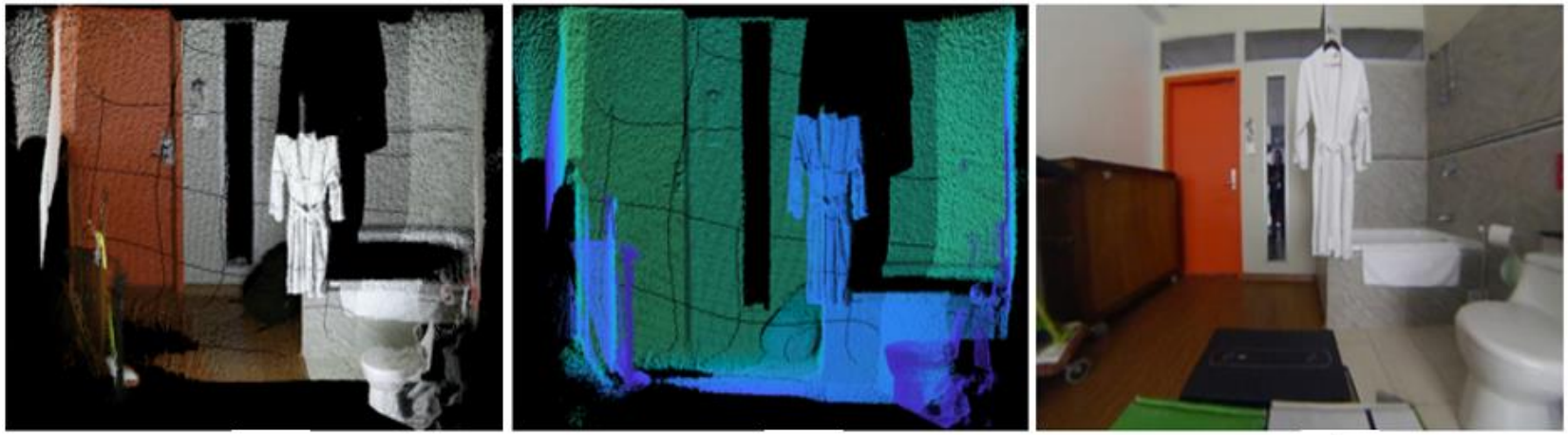
b)

Figura 47. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 42, con una rotación de 90°.

a) Reconstrucción completa del entorno en el cual se encuentran marcados los lugares no reconstruidos debido al ángulo de rotación.

b) Entorno real.

A continuación, se puede apreciar una reconstrucción con un ángulo de rotación de 60° y una resolución “Alta”. En la Figura 48, 49, 50 y 51 se puede apreciar la reconstrucción del entorno separado por secciones para poder ser apreciado correctamente. En la Figura 52 se encuentra la reconstrucción completa del entorno desde un punto de vista superior en la cual se aprecia que, a diferencia de la reconstrucción en 90° esta reconstrucción permite una reconstrucción completa del entorno, gracias a que el campo de visión del Kinect v.2 es menor al ángulo de rotación



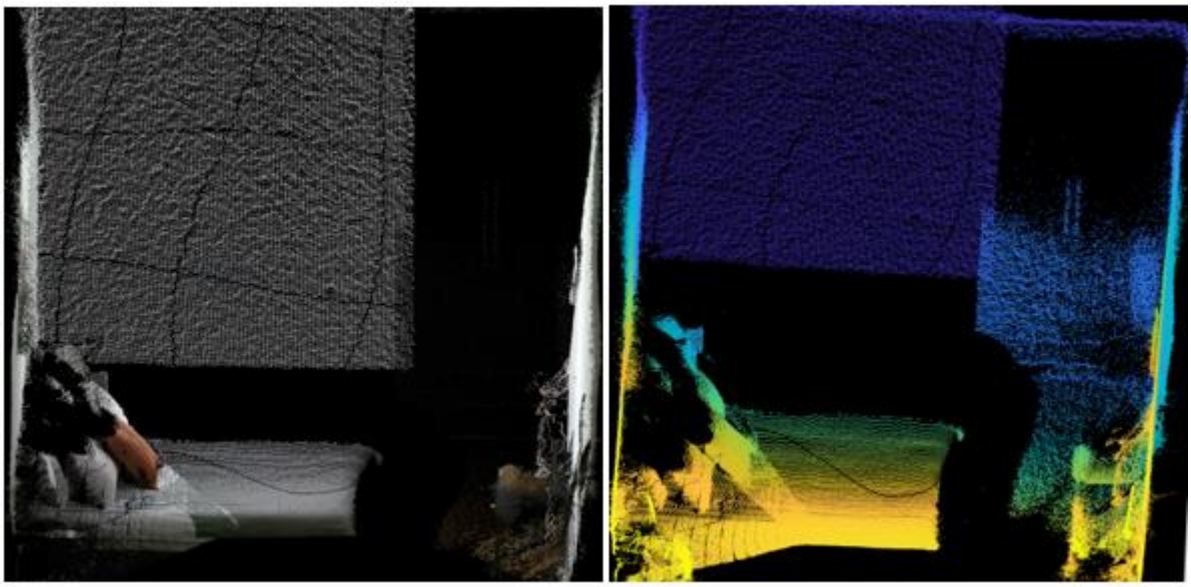
a) **b)** **c)**

Figura 48. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 42, con una rotación de 60°.

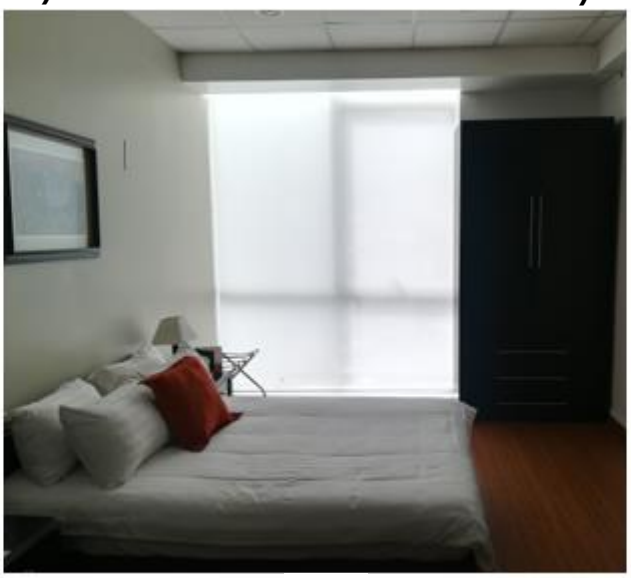
a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.



a) b)



c)

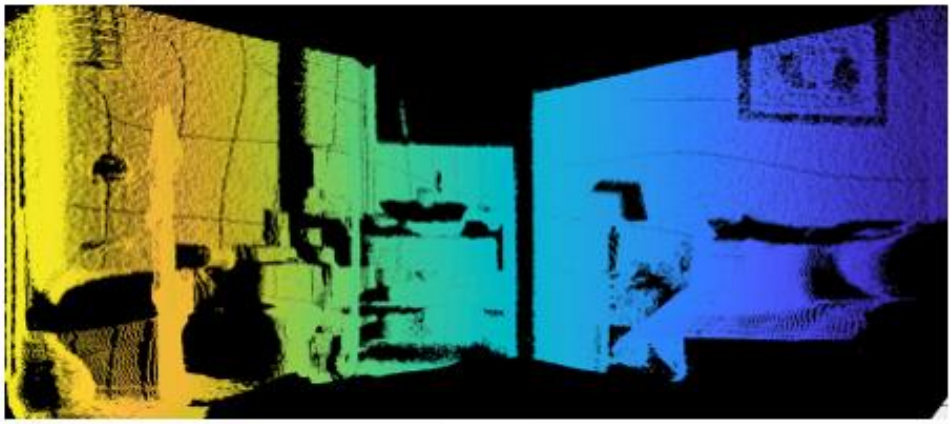
Figura 49. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 42, con una rotación de 60°.

a) Reconstrucción de nube de puntos con RGB.

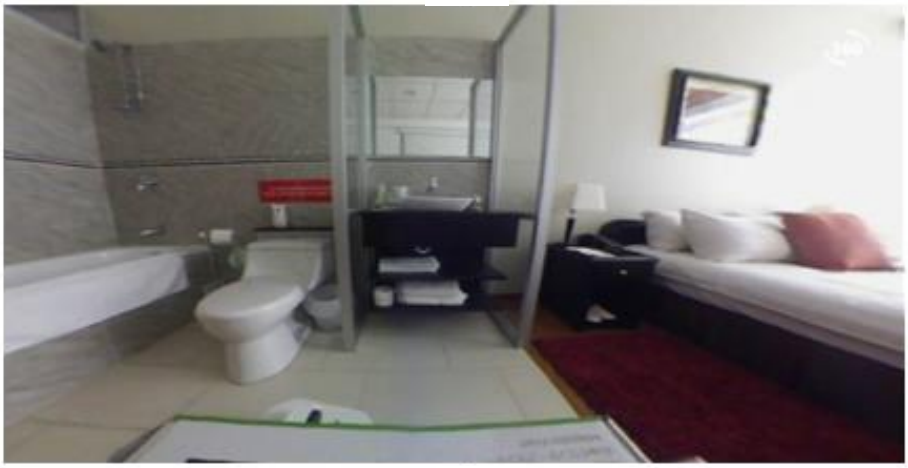
b) Reconstrucción de nube de puntos.

c) Entorno real.

a)



b)



c)

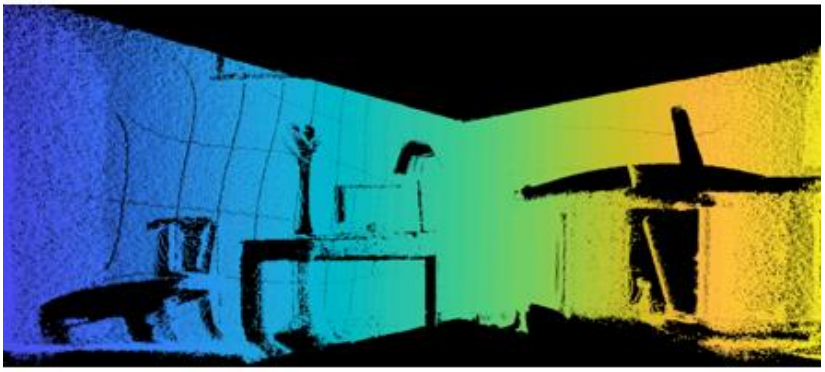
Figura 50. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 42, con una rotación de 60°.

a) Reconstrucción de nube de puntos con RGB.

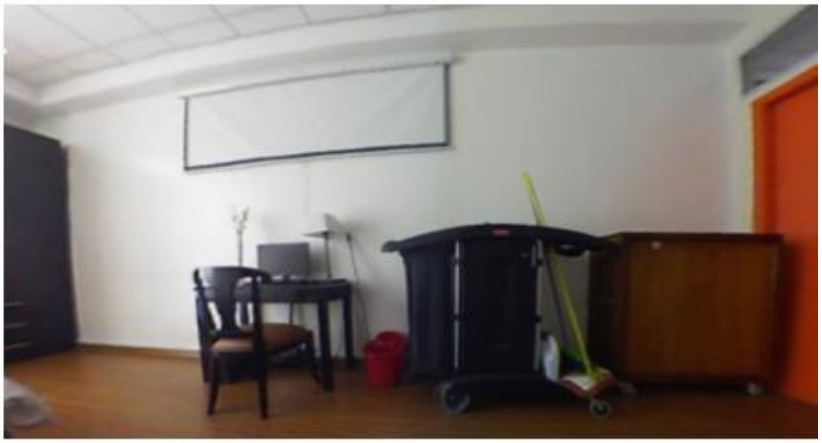
b) Reconstrucción de nube de puntos.

c) Entorno real.

a)



b)



c)

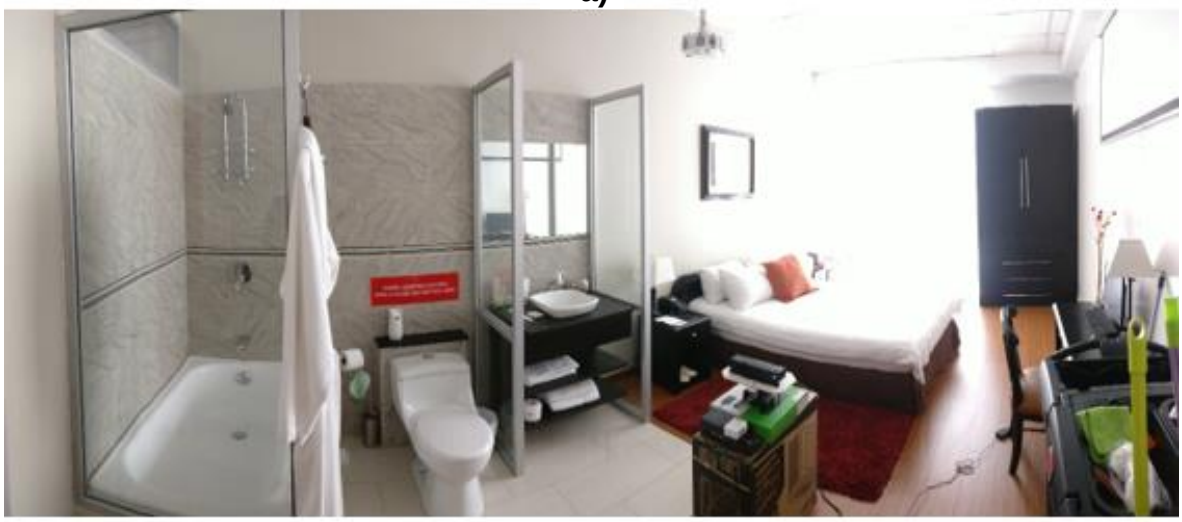
Figura 51. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 42, con una rotación de 60°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.

a)



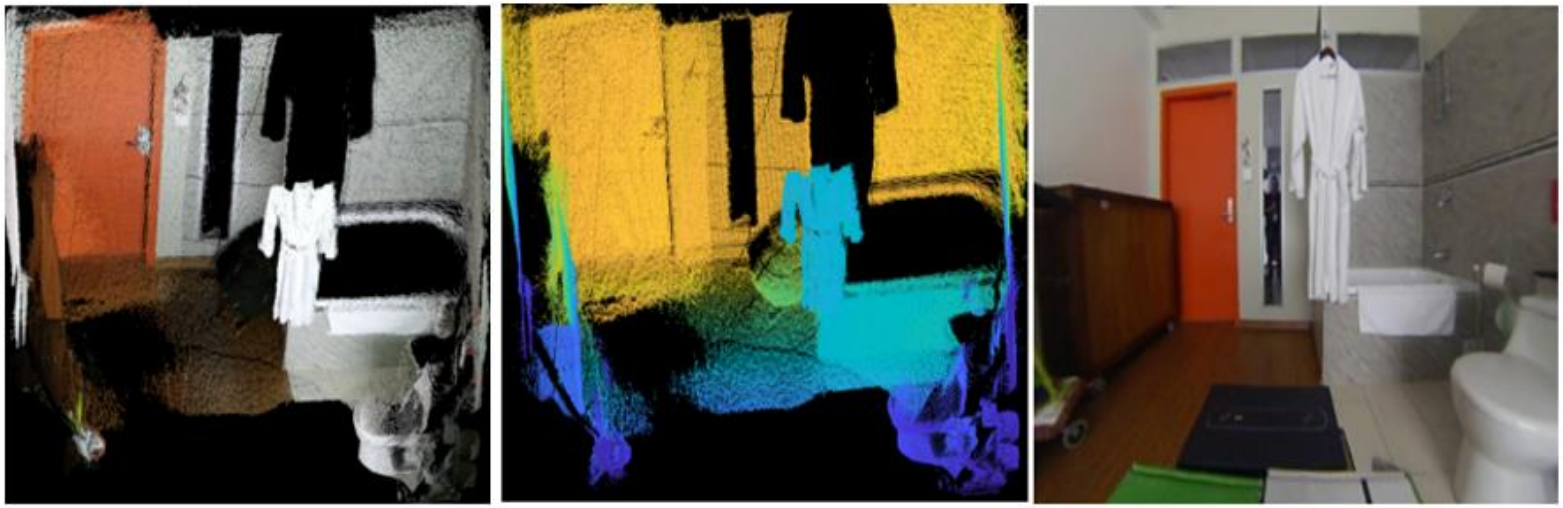
b)

Figura 52. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 42, con una rotación de 60°.

a) Reconstrucción completa del entorno en el cual se encuentran marcados los faltantes.

b) Entorno real.

En las siguientes Figuras se puede apreciar una reconstrucción con un ángulo de rotación de 45° y una resolución "Alta". Se puede apreciar un mayor detalle de objetos en el entorno. Gracias a que tiene un mayor número de iteraciones. En la Figura 53, 54, 55 y 56 se puede apreciar la reconstrucción del entorno separado por secciones para poder ser analizada de mejor manera. Al igual que la reconstrucción con 60°, se obtuvo una reconstrucción completa como se puede ver en la Figura 57.



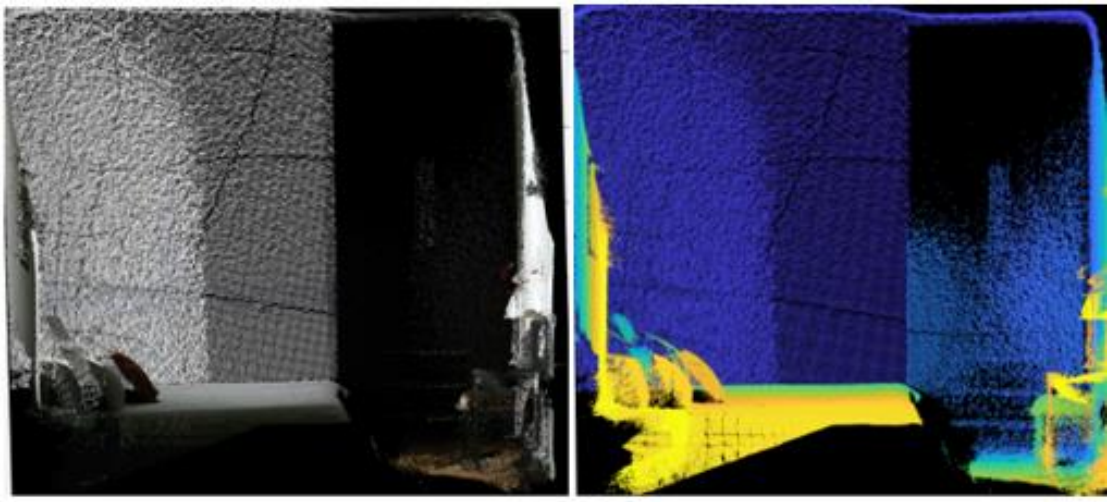
a) **b)** **c)**

Figura 53. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 42, con una rotación de 45°.

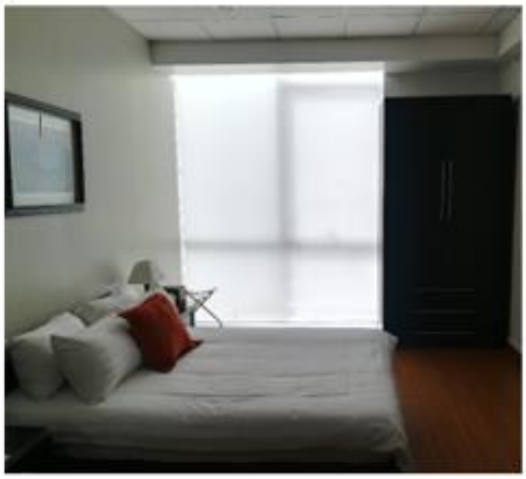
a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.



a) b)



.

c)

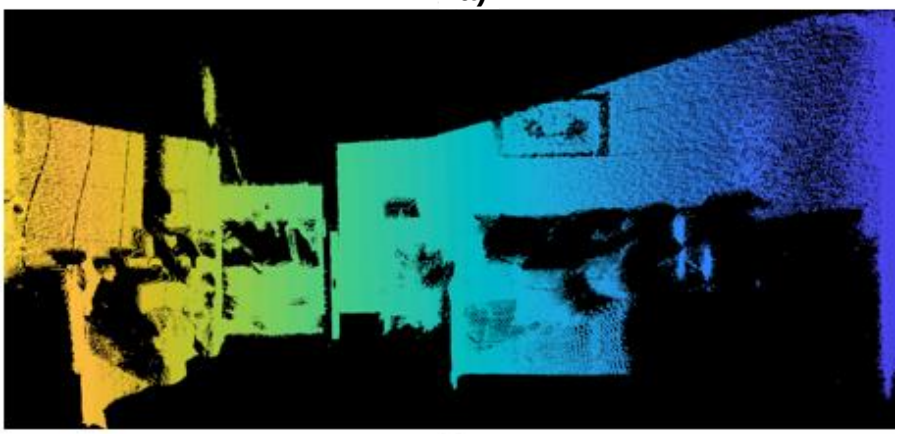
Figura 54. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 42, con una rotación de 45°.

a) Reconstrucción de nube de puntos con RGB.

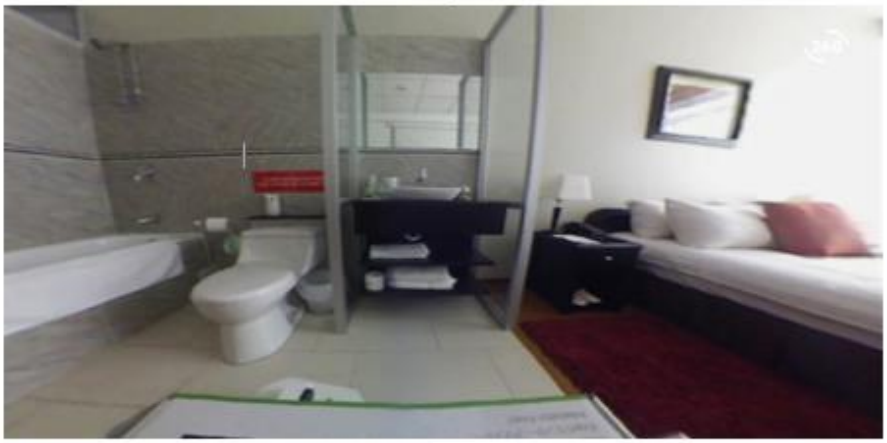
b) Reconstrucción de nube de puntos.

c) Entorno real.

a)



b)



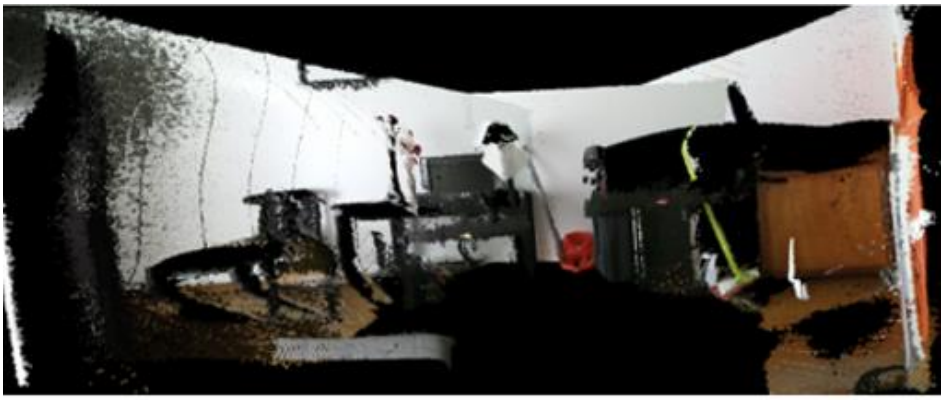
c)

Figura 55. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 42, con una rotación de 45°.

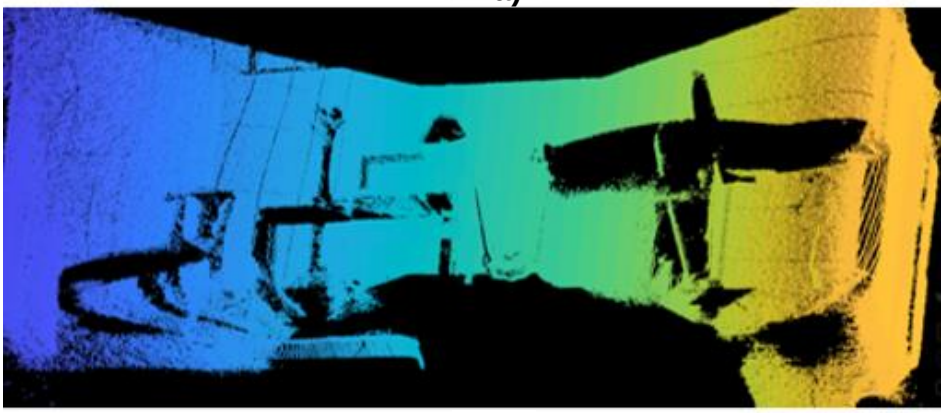
a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

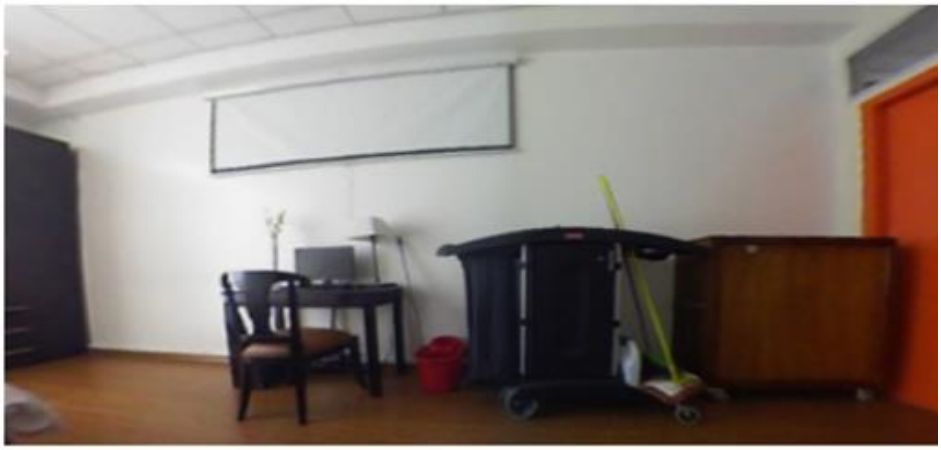
c) Entorno real.



a)



b)



c)

Figura 56. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 42, con una rotación de 45°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.

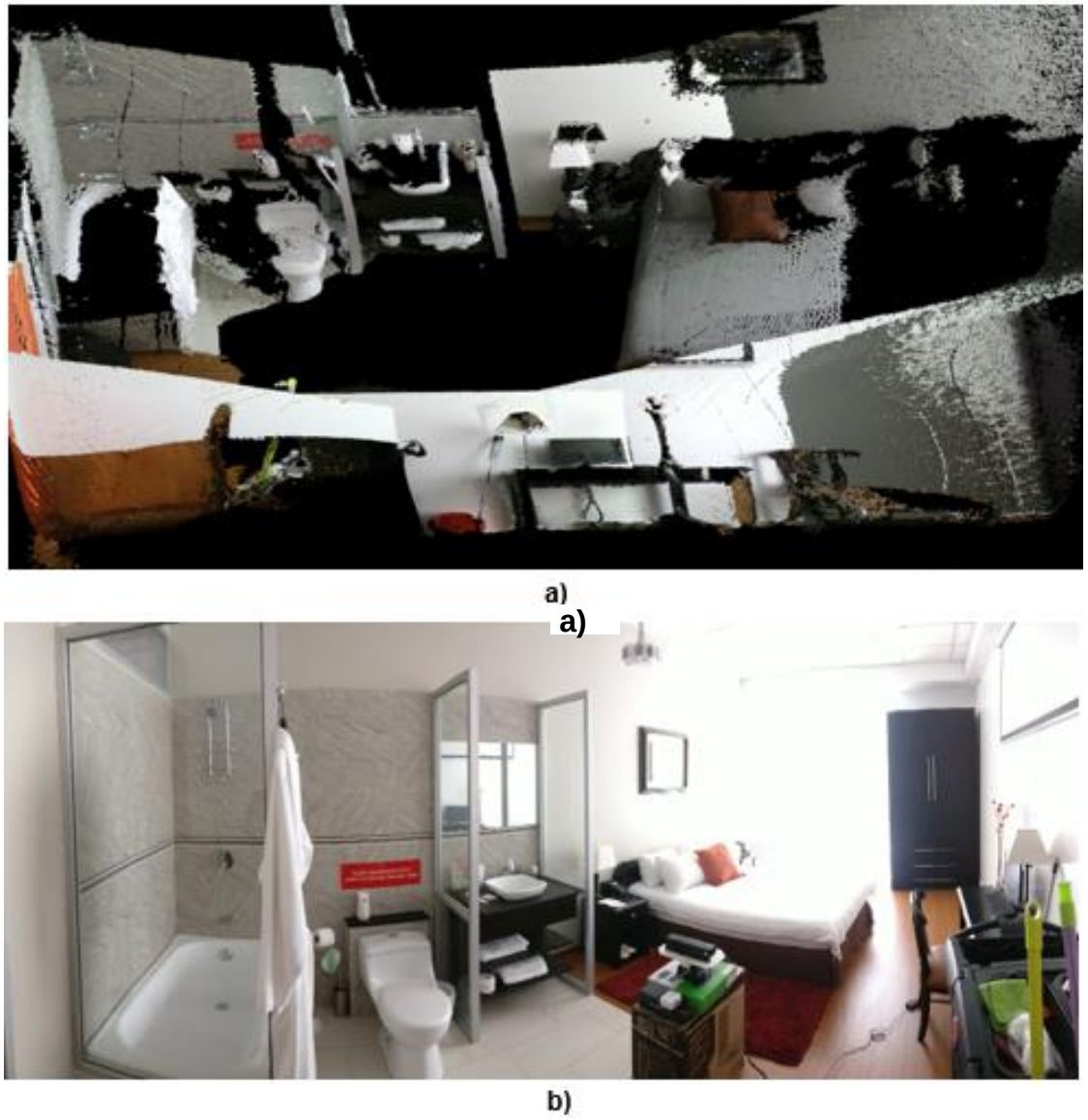


Figura 57. Parte de la reconstrucción co fundidad del entorno de pruebas de la Figura 42, con una rotación de 45°.

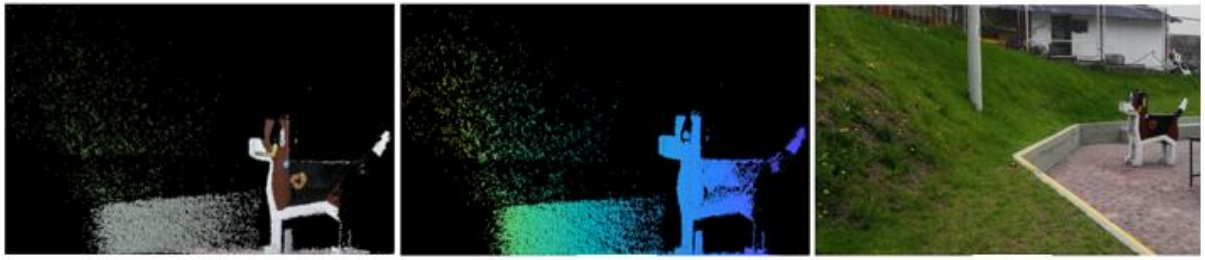
a) Reconstrucción completa del entorno en el cual se encuentran marcados los faltantes.

b) Entorno real.

A continuación, se realizaron pruebas en un ambiente externo, directamente bajo la exposición de luz solar directa. En el lugar se pudo obtener una medición dentro del rango de 40000 y 250000 luxes. En la Figura 58 se puede apreciar el entorno real obtenido mediante el uso de una cámara digital. Al igual que en las pruebas anteriores se puede apreciar dos reconstrucciones. Una de las cuales será una basada únicamente en la localización de cada punto del pointcloud y contará con segmentos de puntos de colores en función de la profundidad, como se puede ver a continuación. La segunda reconstrucción contará tanto con la variable de localización de la nube de puntos, como con variable que contiene la matriz RGB, permitiendo una reconstrucción con color.

Figura 58. Ambiente para pruebas de funcionamiento externo.

En las siguientes Figuras se puede apreciar una reconstrucción con un ángulo de rotación de 90° y una resolución “Alta” En las Figuras 59, 60, 61 y 62 se puede apreciar la reconstrucción del entorno separado por secciones para ser analizado con mayor facilidad. Como se puede observar el alcance del sensor disminuye y la presencia de ruido aumenta. Al igual que en las anteriores pruebas se puede comprobar que una rotación de 90° no es suficiente para obtener una reconstrucción total del entorno como se observa claramente en la Figura 63.



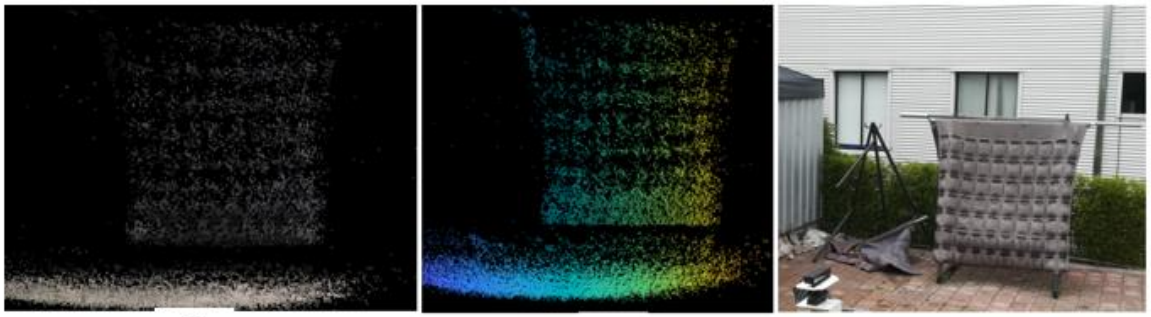
a) b) c)

Figura 59. Parte de la reconstrucción del entorno de pruebas Figura 58, con rotación de 90°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.



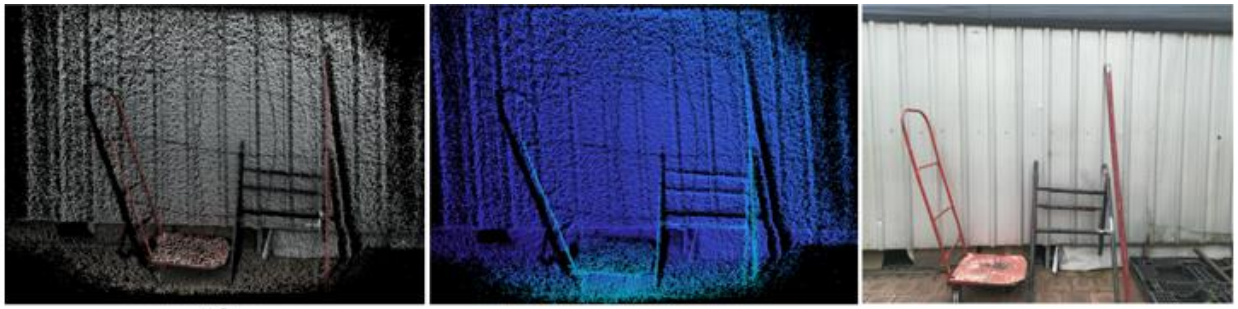
a) b) c)

Figura 60. Parte de la reconstrucción del entorno de pruebas de la Figura 58, con rotación de 90°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.



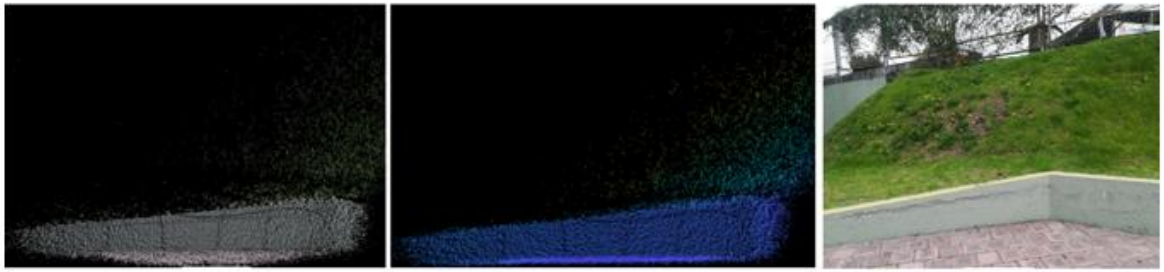
a) b) c)

Figura 61. Parte de la reconstrucción del entorno de pruebas Figura 58, con rotación de 90°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.



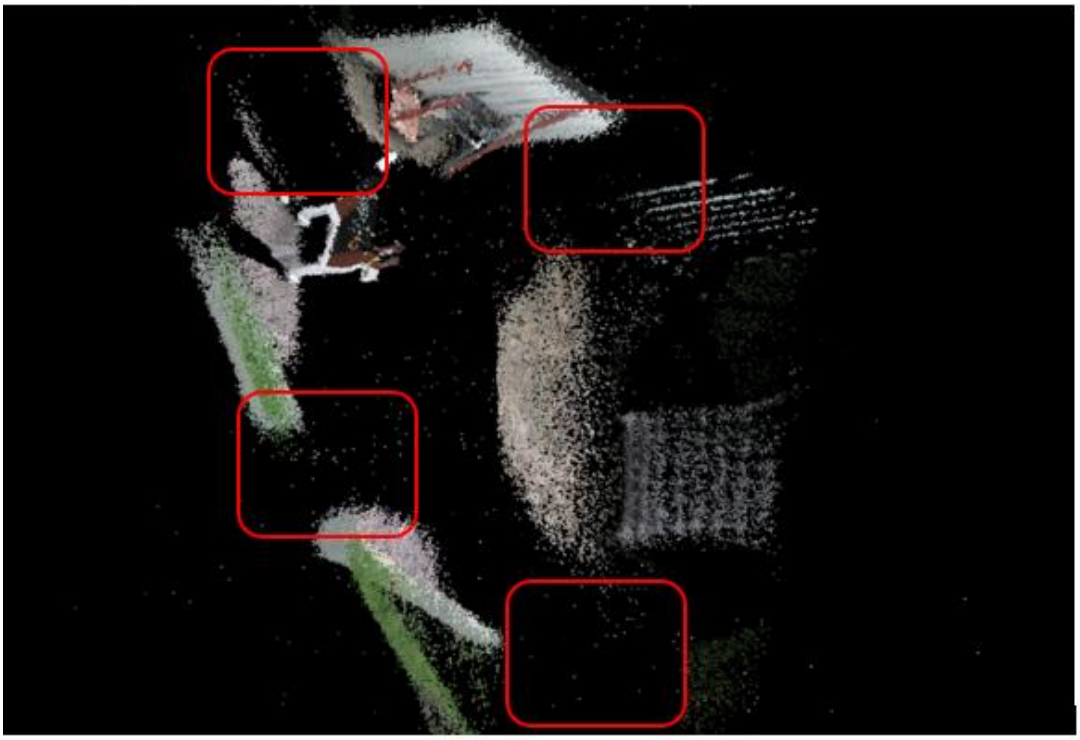
a) b) c)

Figura 62. Parte de la reconstrucción del entorno de pruebas Figura 58, con rotación de 90°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.



a)

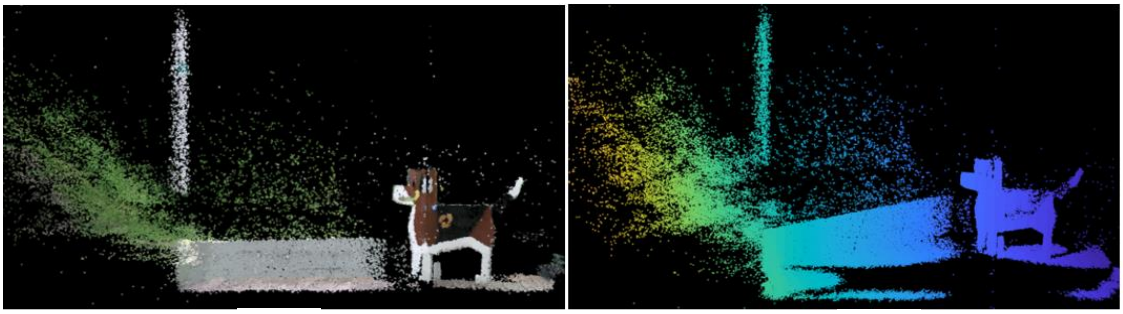
b)

Figura 63. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 58, con una rotación de 90°.

a) Reconstrucción completa del entorno en el cual se encuentran marcados los lugares no reconstruidos debido al ángulo de rotación.

b) Entorno real.

A continuación, se muestra una reconstrucción con un ángulo de rotación de 60° y una resolución “Alta”. El uso de esta rotación permite obtener una mejor reconstrucción del entorno, mas no logra obtener una reconstrucción completa como se puede observar en la Figura 66. Esto se debe a la disminución del alcance del sensor gracias al efecto de la luz solar directa. En la Figura 64 y 65 se puede observar la reconstrucción del entorno separado por secciones.



a) **b)**

c)

Figura 64. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 58, con una rotación de 60°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.

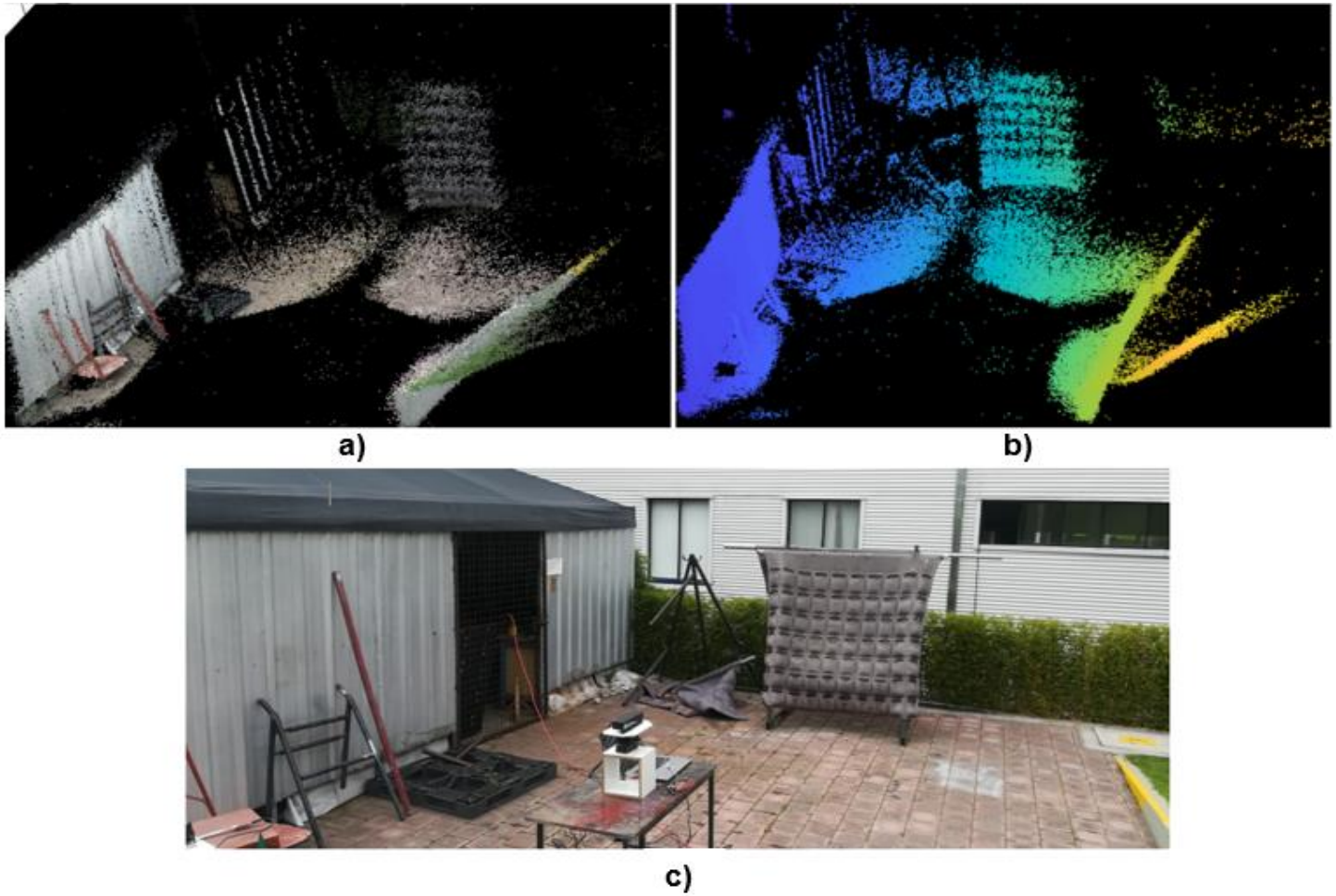
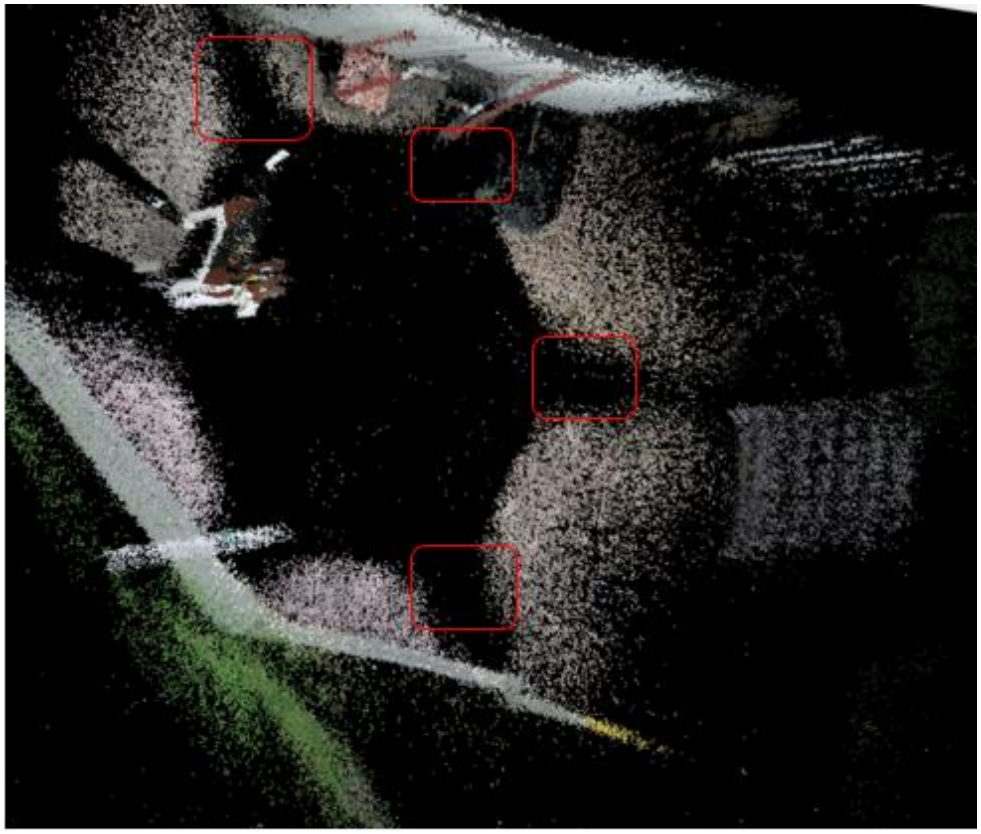


Figura 65. Parte de la reconstrucción del entorno de pruebas de la Figura 58, con una rotación de 60°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.



a)

b)

Figura 66. Reconstrucción completa del entorno con una rotación de 60°

a) Reconstrucción completa del entorno en el cual se encuentran marcados los lugares no reconstruidos debido al ángulo de rotación.

b) Entorno real.

En las siguientes Figuras se muestra una reconstrucción con un ángulo de rotación de 45° y una resolución “Alta”. Al usar este ángulo de rotación se puede obtener una reconstrucción completa del entorno como se puede apreciar a continuación. En la Figura 67, 68 se puede apreciar la reconstrucción del entorno separado por secciones y en la Figura 69 se puede apreciar el entorno completo reconstruido.

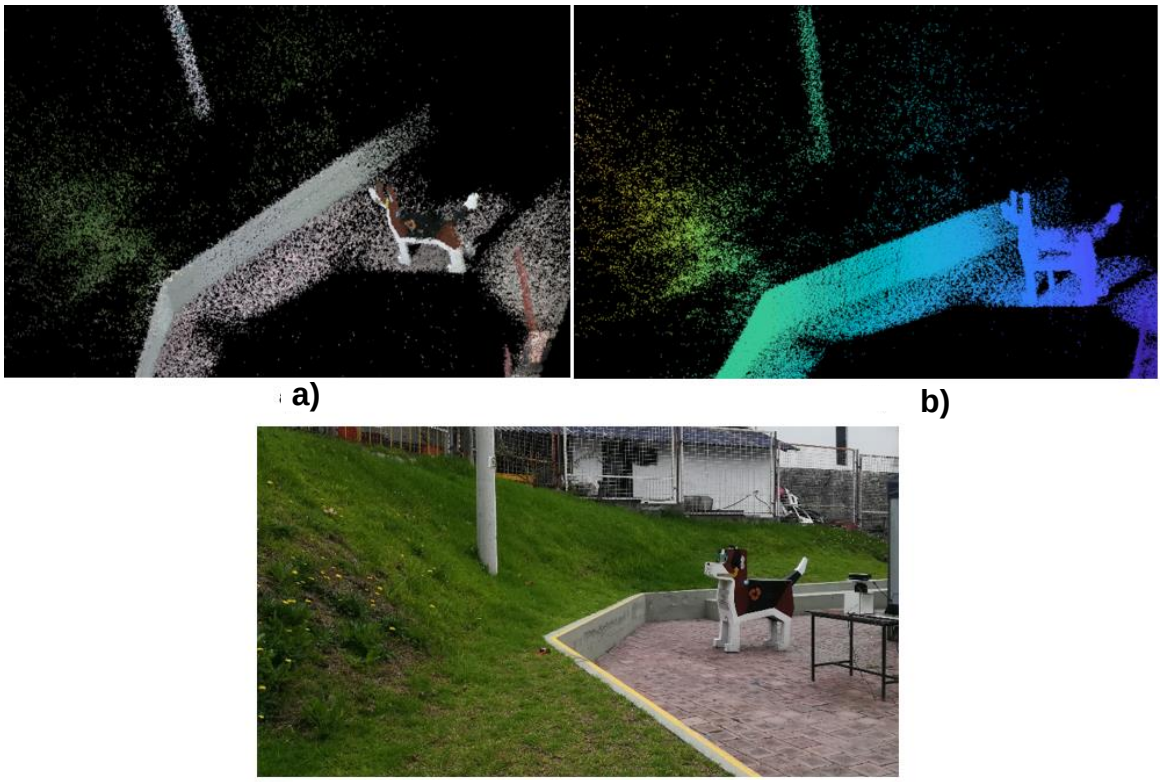


Figura 67. Parte de la reconstrucción del entorno de pruebas de la Figura 58, con una rotación de 45°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.

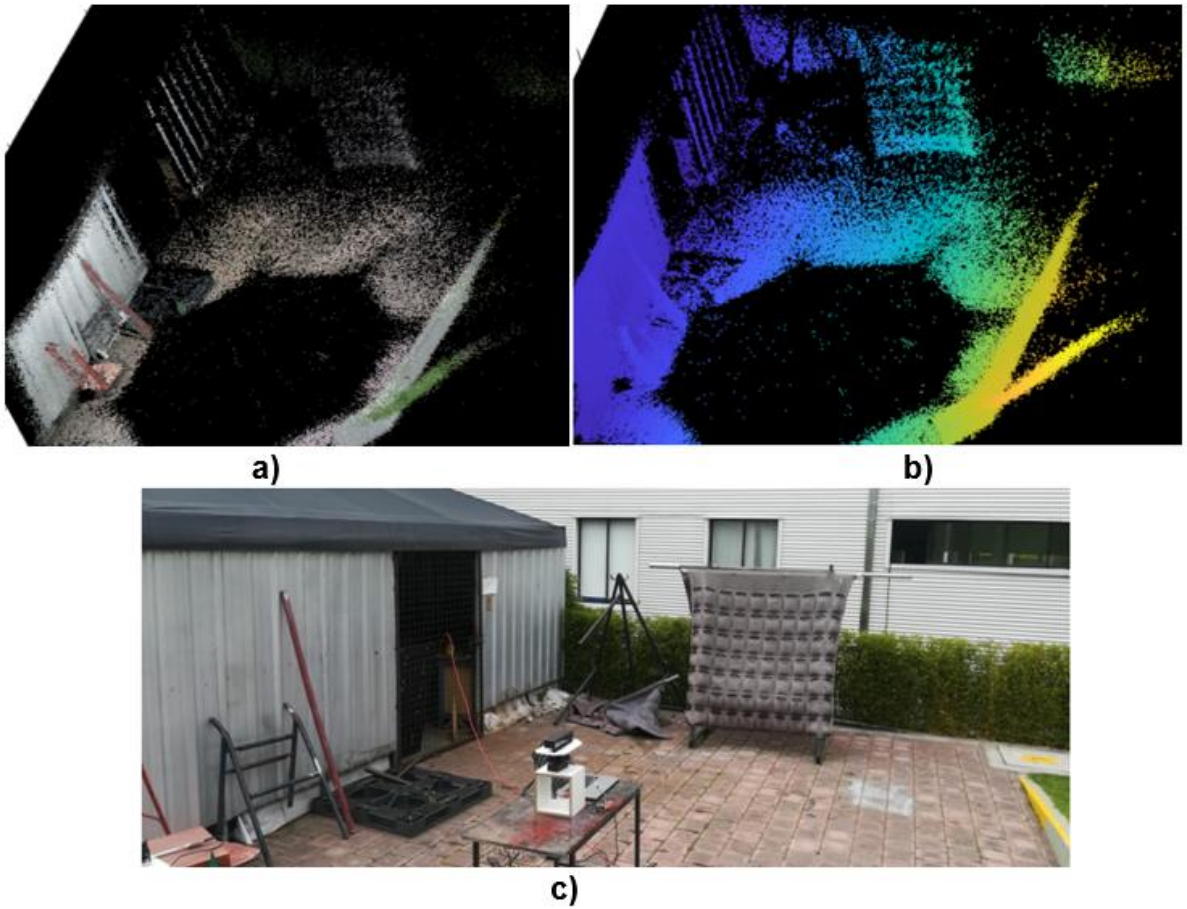
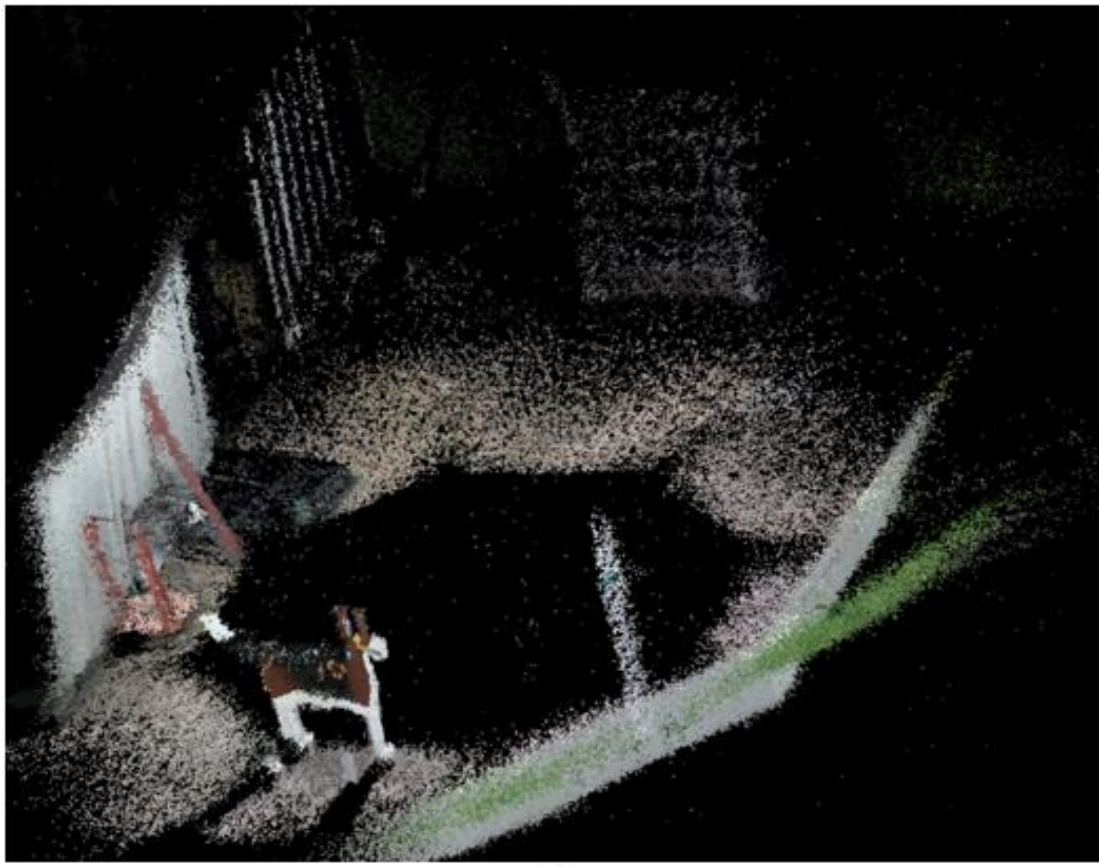


Figura 68. Parte de la reconstrucción con profundidad del entorno de pruebas de la Figura 58, con una rotación de 45°.

a) Reconstrucción de nube de puntos con RGB.

b) Reconstrucción de nube de puntos.

c) Entorno real.



a)

b)

Figura 69. Reconstrucción completa del entorno con una rotación de 45°

a) Reconstrucción completa del entorno.

b) Entorno real.

Mediante la opción “captura simple” del *tol*, se lleva a cabo pruebas para medir la diferencia entre realizar una reconstrucción con luz y en ausencia de ella. La prueba con luz se realizó con una medida entre 130 y 170 luxes, mientras que en ausencia de luz se obtiene una medida entre 2 y 6 luxes. Estos resultados pueden observarse en la Figura 70. Demostrando así que la presencia o ausencia de luz no interfiere en la calidad de la reconstrucción.

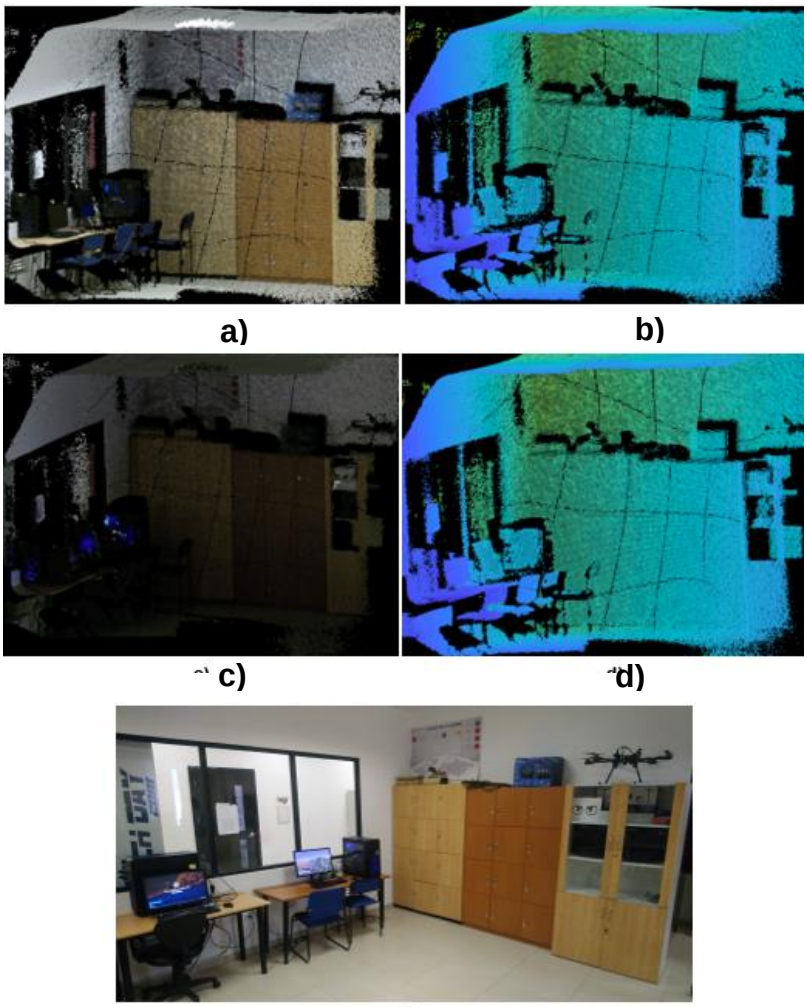
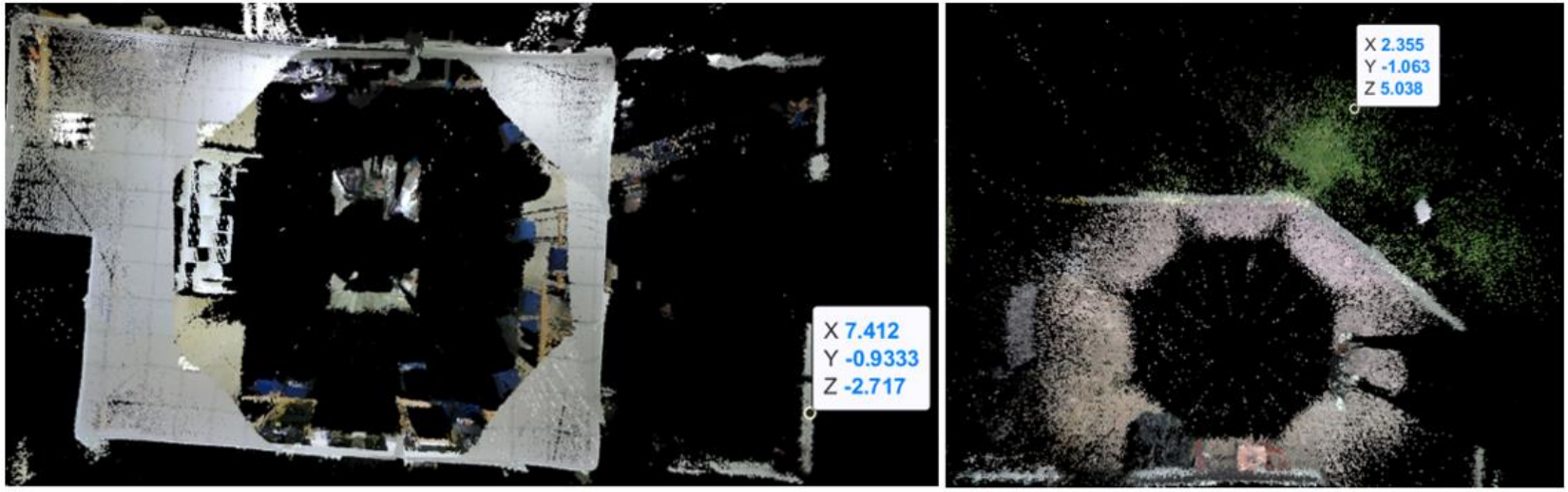


Figura 70. Reconstrucción simple con presencia (a, b) y ausencia de luz (c-d).
a-c) Reconstrucción de nube de puntos con RGB.
b-d) Reconstrucción de nube de puntos.
e) Entorno real obtenido mediante una cámara digital.

Con la finalidad de analizar las diferencias que existen al realizar una captura del entorno con un sensor Kinect v.2, tanto en ambientes controlados como exteriores con luz solar directa, se procede a realizar una comparativa en la Figura 71. Donde ambas tomas se realizaron con un ángulo de 45°. Los resultados de estas diferencias serán mostrados en las conclusiones del trabajo de titulación.



a) **b)**

Figura 71. Comparación de la reconstrucción de entornos en interiores y exteriores.

a) Reconstrucción de un entorno controlado, con una rotación de 45°.

b) Reconstrucción de un entorno exterior bajo luz solar directa, con una rotación de 45°.

5. Conclusiones y recomendaciones

5.1 Conclusiones

El sensor Kinect v.2 permite una reconstrucción del entorno de gran calidad en presencia o ausencia de luz en un ambiente controlado, gracias a que su funcionamiento se basa en la tecnología *ToF*. Esto quiere decir que, a pesar de que no exista ninguna fuente de luz en el lugar, el sensor seguirá obteniendo una resolución de gran calidad; cómo se puede observar en la Figura 70. Mientras que, al estar bajo luz solar directa se reduce significativamente el alcance y la calidad obtenidos en las reconstrucciones, como se puede apreciar en la Figura 71. En ambientes controlados se logra obtener una distancia máxima de 7.4 metros, mientras que en un ambiente exterior se obtiene una distancia máxima de 5.1 metros. Inclusive, al analizar la reconstrucción se puede observar que existe una mayor cantidad de ruido en la reconstrucción del entorno externa, nuevamente debido a la cantidad de luz solar.

Gracias a la variación del ángulo de rotación según sea requerido por el usuario se puede obtener reconstrucciones con mejor calidad que otras. Un ejemplo de esto son las reconstrucciones con un ángulo de 90°. Este ángulo no permite obtener una reconstrucción completa del entorno, esto se debe a que el ángulo de visión del Kinect es de 70°, por lo cual existen espacios vacíos en la reconstrucción. A pesar de esta desventaja puede ser usado para una reconstrucción del entorno más rápida ya que requiere de un menor número de giros y datos para procesar.

Entre las principales ventajas del sensor Kinect v.2 se encuentra que: puede obtener reconstrucciones de gran calidad con un presupuesto limitado; el análisis del entorno se facilita gracias a la imagen RGB proporcionada por el sensor; el procesamiento que se requiere para obtener una reconstrucción es mínimo en comparación de otros dispositivos como ZED o tecnologías de estéreo visión. Por otro lado, entre sus desventajas se encuentra que: el cable

de datos del Kinect v.2 es sumamente pesado y complicado de manipular; las dimensiones del sensor no permiten su portabilidad.

5.2 Recomendaciones

Al realizar las distintas pruebas de reconstrucción de entornos se pudo observar que: en caso de que exista oclusión en la línea de vista del sensor causará una gran cantidad de vacíos y sombras en la reconstrucción; la precisión de la rotación del motor permitirá una reconstrucción limpia y con gran resolución; el uso de filtros permite que se pueda analizar la reconstrucción con una mayor facilidad; cuanto mayor sea el número de tomas la calidad de la reconstrucción mejorará; mediante el análisis de los entornos se puede determinar la distancia a la que se encuentran los objetos del sensor; cuanto más se reduzca el ángulo de rotación mayor será la calidad de la reconstrucción, pero mayor será el requerimiento de procesamiento.

Referencias

Andersen, M., Jensen, T., Lisouski, P., Mortensen, A. K., Hansen, M. K., Gregersen, T., & Ahrendt, P. (2012). Kinect Depth Sensor Evaluation for Computer Vision Applications. *Electrical and Computer Engineering*, *Technical*, 37. https://doi.org/Technical Report ECE-TR-6

Barrientos, A. (2007). *Fundamentos de Robotica*. (S.A. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA, Ed.) (2ª ED).

Cheng, J. X., Jia, Y. K., Zheng, G., & Xie, X. S. (2002). Laser-scanning coherent anti-Stokes Raman scattering microscopy and applications to cell biology. *Biophysical Journal*, *83*(1), 502–509. https://doi.org/10.1016/S0006-3495(02)75186-2

Chuya, J., Cuenca, M., & Delgado, K. (2013). Diseño E Implemetación De Un Sistemas Para El Análisis Del Movimeinto Humano Usando Sensores Kinect, 113.

Demitseva, S. (2015). Gesture based computer controlling using kinect camera. *Journal of applied microbiology*, *119*(3), 859–867.

El-Gwad, G. N. H. A., & Omar, Y. M. K. (2017). Selection of the best despeckle filter of ultrasound images. *Proceedings - 2017 2nd International Conference on Multimedia and Image Processing, ICMIP 2017*, *2017–Janua*, 245–249. https://doi.org/10.1109/ICMIP.2017.46

Eskow, C. (2018). Optical Time Of Flight Sensors. Recuperado el 23 de octubre de 2018, de https://www.ecnmag.com/article/2018/06/optical-time-flight-sensors-overview

Flanagin, M., Grenotton, A., Ratcliff, J., Abdelguerfi, M., & Shaw, K. (2005). Automated terrain generation using LIDAR and waterbody survey data. *Proceedings of the 2005 ACM symposium on Applied computing - SAC '05*, 1185. https://doi.org/10.1145/1066677.1066946

Flores, P., & Braun, J. (2011). Algoritmo RANSAC : fundamento teórico, (2), 2–3.

Fossati, A., Gall, J., Grabner, H., Ren, X., & Konolige, K. (Eds.). (2013). *Consumer Depth Cameras for Computer Vision*. Springer.

https://doi.org/10.1007/978-1-4471-4640-7

Frigerio, E., Marcon, M., & Tubaro, S. (2013). Surface reconstruction using 3D morphological operators for objects acquired with a multi-Kinect system. *Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications - MIRAGE '13*, 1. https://doi.org/10.1145/2466715.2466719

Grupo de robótica Universidad de León. (2015). PCL/OpenNI tutorial 2: Cloud processing (basic). Recuperado el 12 de diciembre de 2018, de http://robotica.unileon.es/index.php/PCL/OpenNI_tutorial_2:_Cloud_processing_(basic)#Passthrough_filter

Hernández, I. (2015). métodos fotogramétricos .

Hokuyo Automatic. (2014). Distance Data Output/YVT-35LX. Recuperado el 12 de diciembre de 2018, de https://www.hokuyo-aut.jp/search/single.php?serial=224

Ibarra, J. M., Cisneros, R., García, L., Lavín, J. ., & Yañez, I. (2008). Evasión de obstáculos basada en visión artificial y en sensores infrarrojos para un humanoide. *Sometido AI*, (October).

Ilbay, L. (2014). Reconstrucción activa de objetos 3D mediante escaneo láser y generación de la vista por medio del software matlab, 1–126.

Jin, G. (2008). Creating 3D images of objects by illuminating with infrared patterns, *2*(12).

Khoshelham, K., & Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, *12*(2), 1437–1454. https://doi.org/10.3390/s120201437

MATLAB. (2018a). Descripción del producto MATLAB. Recuperado el 4 de noviembre de 2018, de https://la.mathworks.com/help/matlab/learn_matlab/product-description.html

MATLAB. (2018b). Image Acquisition Toolbox. Recuperado el 20 de noviembre de 2018, de https://la.mathworks.com/help/imaq/index.html

MATLAB. (2018c). Image Acquisition Toolbox Support Package for Kinect For

Windows Sensor. Recuperado de https://la.mathworks.com/help/supportpkg/kinectforwindowsruntime/index.html?searchHighlight=kinect&s_tid=doc_srchtitle

Microsoft. (2014a). Kinect for Windows Runtime 2.0. Recuperado el 20 de noviembre de 2018, de https://www.microsoft.com/en-us/download/details.aspx?id=44559

Microsoft. (2014b). Kinect for Windows SDK 2.0.

MIT. (2012). Mechanical engineers develop an 'intelligent co-pilot' for cars. Recuperado el 9 de octubre de 2018, de http://news.mit.edu/2012/mechanical-engineers-develop-intelligent-car-co-pilot-0713

MoBiVAP Group Research. (2013). Reconstrucción 3D, 15.

Navarro, F. (2017). Aplicaciones de la Reconstrucción 3D: odometría visual e Integración con la realida virtual, 1–64.

Navas, D., Vargas, J., & Morales, L. (2017). The nearest object localization through 3D lidar reconstruction using an embedded system. *XXVII Jornadas de Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional*, *27*, 32–38.

Oliver, A., Kang, S., Wünsche, B. C., & MacDonald, B. (2012). Using the Kinect as a navigation sensor for mobile robotics. *Proceedings of the 27th Conference on Image and Vision Computing New Zealand - IVCNZ '12*, 509–514. https://doi.org/10.1145/2425836.2425932

PCL. (2011). PCL Filtering. *PCLFiltering*. Recuperado de http://www.pointclouds.org/assets/iros2011/filtering.pdf

Popescu, C. R., & Lungu, A. (2014). Real-Time 3D Reconstruction Using a Kinect Sensor. *Computer Science and Information Technology*, *2*(2), 95–99. https://doi.org/10.13189/csit.2014.020206

Salido, J. A. (2013). Reconstrucción densa de mapas de profundidad por stereo-visión con la ayuda de un proyector. *Journal of Chemical Information and Modeling*, *53*, 1689–1699. https://doi.org/10.1017/CBO9781107415324.004

Sarbolandi, H., Lefloch, D., & Kolb, A. (2015). Kinect range sensing: Structured-

light versus Time-of-Flight Kinect. *Computer Vision and Image Understanding*, *139*, 1–20. https://doi.org/10.1016/j.cviu.2015.05.006

Schwarz, L., & Navab, N. (2012). Depth Imaging with Time-of-Flight Cameras and the Kinect. Recuperado de http://campar.in.tum.de/twiki/pub/Chair/TeachingWs11Cv2/3D_CV2_WS_2011_ToF-Imaging.pdf

Sell, J., & Connor, P. O. (2014). THE XBOX ONE SYSTEM ON A CHIP AND KINECT SENSORsor, 44–53.

Shruthi, B., Renukalatha, S., & Siddappa, M. (2015). Speckle Noise Reduction in Ultrasound Images- A Review. *International Journal of Advanced Research in Computer Science and Software Engineering*, *4*(9), 34–43. https://doi.org/AGE1875 [pii]\r10.1111/j.1365-2052.2009.01875.x

Silva, J., & Zapata, A. (2015). Reconstruccion Entornos 3D.

Sistemas Robustos. (2018). Matrices de Rotación y sus transformaciones. Recuperado el 2 de noviembre de 2018, de http://www.multirotorguides.com/aprendamos/conceptos-matematicos/matrices-de-rotacion/

SLAMTEC. (2018). RPLIDAR A2.

StereoLabs. (2018). ZED Stereo Vision. Recuperado el 24 de octubre de 2018, de https://docs.stereolabs.com/overview/getting-started/introduction/

TERABEE. (2018). Time-of-Flight Principle. Recuperado el 1 de noviembre de 2018, de https://www.terabee.com/time-of-flight-principle/

Valgma, L. (2016). 3D reconstruction using Kinect v2 camera. *Bachelor's thesis*.

Varanasi, S., & Devu, V. K. (2016). 3D Object Reconstruction Using XBOX Kinect V2.0, (September).

von Ramm, O. T., Pavy, H. G., & Smith, S. W. (1991). High-Speed Ultrasound Volumetric Imaging System-Part II: Parallel Processing and Image Display. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, *38*(2), 109–115. https://doi.org/10.1109/58.68467

Wasenmuller, O., & Sticker, D. (2017). Comparison of Kinect V1 and V2 Depth Images, *10118*(September). https://doi.org/10.1007/978-3-319-54526-4

Waymo. (2018). Waymo. Recuperado el 12 de diciembre de 2018, de https://waymo.com/tech/

Wolff, K., Kim, C., Zimmer, H., Schroers, C., Botsch, M., Sorkine-Hornung, O., & Sorkine-Hornung, A. (2016). Point cloud noise and outlier removal for image-based 3D reconstruction. *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, 118–127. https://doi.org/10.1109/3DV.2016.20