VOLUMEN 21
XXVIII Verano De la Ciencia
ISSN 2395-9797
www.jóvenesenlaciencia.ugto.mx

Image understanding for autonomous vehicles using depth information

Comprensión de imágenes usando información de profundidad para la navegación de vehículos autónomos

Francisco Javier Moreno Vazquez[1], Hector Guevara Mosqueda[2], Alberto Franco Mendez[3], Dora Luz Almanza Ojeda[4]
[1]Licenciatura en Ingeniería en Mecatrónica
[2]Licenciatura en Ingeniería en Sistemas Computacionales
[3]Licenciatura en Ingeniería en Comunicaciones y Electrónica
[4]Departamento de Ingeniería Electrónica
División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato
fj.morenovazquez@ugto.mx[1], h.guevaramosqueda@ugto.mx[2], a.francomendez@ugto.mx[3], dora.almanza@ugto.mx[4]

Resumen

En este proyecto se probarán diferentes módulos de reconstrucción 3D para el análisis de contenido de imágenes capturadas desde un vehículo en un ambiente de interior y exterior. El análisis de la imagen color y profundidad del ambiente serán capturadas por cámaras RGBD. Los módulos utilizados forman parte de la librería Open3D. Las imágenes para probar serán capturadas desde un vehículo móvil supervisado para navegación en el interior del laboratorio y en el exterior sobre el estacionamiento. La reconstrucción de la escena apoya para identificar las zonas libres de obstáculos, a fin de que el vehículo pueda navegar de forma autónoma. Adicionalmente, se hace la captura de los datos 3D para objetos encontrados comúnmente en un escenario interior y con ello generar una base de datos que permita el entrenamiento de modelos convolucionales. Se proporciona la representación de la escena vista desde el robot en un video demostrativo.

Palabras clave: imagen color; imagen profundidad; reconstrucción 3D; modelos de puntos.

Introducción

El crecimiento y evolución de los sistemas de cómputo y las comunicaciones ha permitido desarrollar nuevas tecnologías para mejora de procesos. En particular, el área de los vehículos autónomos se encuentra en creciente popularidad, gracias a todos los diferentes sensores que pueden incluirse y desde los cuales es posible construir módulos inteligentes que permiten la conducción del vehículo. Muchas propuestas describen y han demostrado su eficiencia en modelos de vehículos eléctricos, incluso algunos ya en funcionamiento en algunos países. Sin embargo, aumentar el uso estos vehículos para la gran mayoría de la población es, por el momento, inalcanzable por los altos costos de producción que involucra.

La navegación de vehiculos autónomos requiere integrar diferentes sensores que entreguen información de todo el entorno. Los sensores más comunmente instalados en los robots móviles son: sensores de posicionamiento global (GPS), Unidades de Medición Inercial (IMU), sensores de proximidad, dispositivos laser tipo LIDAR y por supuesto cámaras color y profundidad. Las cámaras color profundidad conocidas como RGBD son un tipo de cámara de profundidad que combina información de color (RGB) y profundidad (D) para capturar datos tridimensionales de una escena. Dentro de la gamma de estas cámaras encontramos el Azure Kinect, el cual según Microsoft es un kit de desarrollo, compuesto por 1 megapíxel para la cámara de profundidad, un micrófono 360 y la cámara de color con 12 megapíxeles. Una alternativa son las cámaras Intel RealSense SR305 y D415, las cuales son cámaras de profundidad estereoscópica. Estas cámaras son muy utilizadas debido a su bajo costo, gran cantidad de información que proporciona y la velocidad de captura. Si bien, no es la solución única para la navegación de un vehículo autónomo, es posible entregar una buena representación de la escena bajo condiciones controladas. Otros sensores 3D muy utilizados son el Velodyne y LIDAR que son sensores los cuales mediante un rayo de luz escanean la superficie creando un modelo tridimensional de la escena. Las Unidades de Medición Inercial (IMU) son sensor no necesariamente para la reconstrucción tridimensional, pero este funciona para medir la velocidad y aceleración angular estimando la posición y la orientación. Los sensores de posicionamiento global o GPS brindan una referencia sobre la ubicación global (marco referencia global) del vehículo o robot.

La reconstrucción de un escenario desde la imagen de color y profundidad es un mecanismo similar al de utilizar dos cámaras RGB convencionales, en configuración conocida como estereovisión. Sin embargo, como estas cámaras generan una imagen desde un haz de luz infrarroja, es posible hacer la reconstrucción mediante la proyección de ese patrón de luz estructurado y midiendo su tiempo de rebote sobre los objetos en la escena [1]. Añadir esta información de profundidad a las imágenes color permite que estas cámaras sean útiles para una variedad de aplicaciones, como la robótica, la realidad aumentada y la visión por computadora [2]. De forma similar, combinando una imagen en color y una imagen de profundidad permite la representación de la información visual del color con la información espacial de la profundidad. Esta técnica de fusión de datos se modela en librerías especializadas para procesamiento de imágenes formando objetos de tipo RGBDImage que representan de manera conjunta la apariencia visual y la información de la profundidad de la escena capturada.

Por otra parte, la información 3D nos permite generar nubes de puntos. Una nube de puntos es una representación tridimensional de una escena u objeto capturado mediante un sistema de sensores de luz o camaras color-profundidad. Consiste de un conjunto de puntos en el espacio 3D, donde cada punto tiene una ubicación y posiblemente también información adicional, como el color o la intensidad. Estos puntos se generan a partir de datos de profundidad. En el caso de utilizar cámaras modelo pinhole, se requiere una calibración que entregue los parámetros de funcionamiento de la cámara, conocidos como parámetros intrínsecos y extrínsecos. Los parámetros intrínsecos desempeñan un papel fundamental en el modelado y la calibración de la cámara. Estos parámetros describen características internas de la cámara, como la distancia focal, el punto principal y la relación de aspecto, que son esenciales para la proyección y mapeo precisos de las imágenes capturadas.

Así la metodología propuesta en este proyecto consiste en realizar el análisis de imágenes color y profundidad capturadas desde robots móviles en escenarios de interior y exterior. Se prueban módulos para la reconstrucción de la escena lo que permita dar una interpretación del espacio navegable y la zona con obstáculos. Los módulos que se utilizan emplean las imágenes color para hacer la proyección de los puntos dentro de la librería open3D. El escenario se reconstruye con caracteristicas de color y textura, conocido como de forma densa y también se propone la reconstrucción dispersa a través de los puntos más representativos. Los objetos más comunes que se pueden encontrar en escenarios de interior son reconstruidos en modelos basados en nubes de puntos para generar una base de datos de sus diferentes vistas. La propuesta es etiquetar los objetos en la imagen a través de un modelo de red convolucional entrenado basado en el modelo PointNet. Los resultados demuestran el funcionamiento de los modulos para la reconstrucción del escenario tanto de exterior como de interior, además del modelado de los objetos.

Metodología

El proceso de navegación del robot se lleva a cabo en dos partes: 1) la programación del robot para navegar en espacio de interior o de exterior, y 2) programación de los módulos para la adquisición de las imágenes y preparar la reconstrucción del escenario. La Figura 1 muestra el diagrama a bloques de la metodología realizada para hacer la proyección de la escena desde las imágenes de color y profundidad hacia el espacio 3D. El primer paso de la metodología consiste en la programación de la rutina del robot para que avence en línea recta y dé giros de 90° o de 180° según lo requiera. Una vez programado el robot, la cámara se inicializa mediante el toolbox de calibración y configuración proporcionado por el fabricante[REF]. El módulo captura la imagen color y profundidad, a las cuales se selecciona un área de interés a procesar. El área de interés permite solo procesar la parte más representativa de la imagen, sin tener que procesarla por completo. El área de interés de las imágenes color y profundidad se funsionan para genera una imagen RGBD, en un proceso similar a un empalme de las imágenes. Para ello, se requieren los parametros intrinsecos de la camara y la distancia entre el sensor de visión (cámara color) y el sensor infrarrojo para la captura de la imagen de profundidad. El cálculo de los parametros intrinsecos de la cámara es posible mediante la función de calibración que proporciona la librería. Con los parámetros intrínsecos es posible generar la reproyección de los puntos al espacio 3D, este proceso se conoce como generación de la nube de puntos. El número de puntos generado puede ser muy elevado, por lo que se realiza un proceso de submuestreo para reducir el tamaño del conjunto de puntos. La nube de puntos se combina imagen por imagen a lo largo de la secuencia de video capturada, así, de forma incremental se enriquece el modelo de la nube de puntos permitiendo la reconstrucción del escenario, a lo cual se le agrega información de color de la escena. La escena de exterior, a pesar de las dificultades que puede presentar, como iluminación, gran variedad de texturas, formas estructuradas y no estructuradas, entre otras, de cualquier modo es posible lograr una reconstrucción muy

similar a la real, lo que permite la interpretación del ambiente y las zonas libres de obstáculos para la navegación del robot.

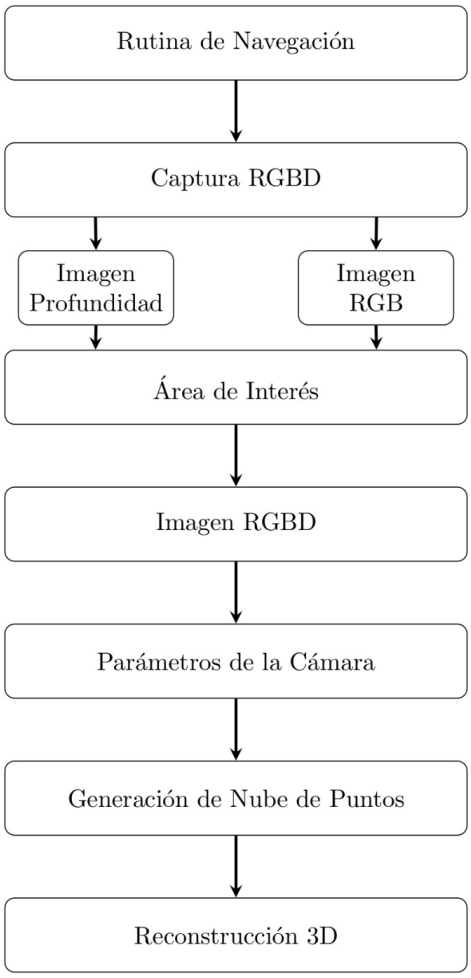


Figura 1. Diagrama general de la metodología propuesta.

Robot móvil y cámara

El robot utilizado para la navegación y captura de imágenes se muestra en la Figura 2. El robot es ensamblado con piezas del kit Tetrix Max de la compañía Pitsco. Está equipado con un controlador PRIZM, similar a una tarjeta Arduino Uno, como unidad de procesamiento. La programación del robot se lleva a cabo mediante la interfaz de la tarjeta Arduino en lenguaje C, con instrucciones que permiten controlar la velocidad, giros y la cinemática del robot [3]. Para programar directamente desde el ambiente de desarrollo (IDE) de Arduino se necesita la biblioteca PRIZM.h. Esta librería contiene las funciones que son como los comando especiales para controlar cada uno de los sensores que integran al robot, en nuestro caso solo utilizaremos los motores y por ello sus encoders para conocer su velocidad. La instrucción "*prizm.setMotorPowers(50,50)*" permite arrancar los motores a una potencia de 50, sin embargo este valor puede ser entre 0 y 100. La rutina de navegación implementada en el robot utilizado para la captura de imágenes y navegación se basa en una secuencia de pasos que le permite moverse de ida y vuelta hacia su punto de partida. A continuación, se describen los pasos de la rutina:

1. El robot comienza moviéndose hacia adelante con ambos motores activados a un 50% de potencia durante un periodo de tiempo específico, en nuestro caso es 10 segundos.

2. Luego, el robot realiza dos giros a la derecha, donde se activan los motores de manera asimétrica para lograr el giro, esto le permite girar 180°.
3. Después de completar el giro, el robot se detiene utilizando el frenado, y se espera un corto período de tiempo antes de continuar.
4. Para regresar a su punto de destino, el robot repite el paso 1.
5. Al completar la secuencia de pasos, el robot hará un recorrido de regreso a su posición de inicio, lo que demuestra su capacidad de navegar en una ruta predeterminada.

a) b)

Figura 2. a) Robot pitsco utilizado para adquirir las imágenes. b) Cámara Intel Real Sense SR305 utilizada.

Como se muestra en la Figura 2a, se coloco la cámara Intel RealSense SR305 en la parte superior del robot. Para una mejor visualización, la cámara se muestra en la Figura 2b. El módulo de adquisición de la cámara está programado en Python y permite la captura de imágenes de color y de profundidad a velocidad de 30 imágenes por segundo con resolución de 640x480 en formato png. El procesamiento de las imágenes adquiridas se realiza mediante la librería Open3D. Esta es una librería de Python de código abierto para procesamiento de datos 3D. Esta librería contiene una gran cantidad de algoritmos de visión por computadora y gráficos 3D, como reconstrucción de mallas 3D, registro de nubes de puntos, visualización de datos 3D, entre otros. Además, ofrece una interfaz de programación para vincular C++ que permite su interacción con este lenguaje [4].

Como se menciono en la introducción, los parámetros intrínsecos de una cámara, describen características internas de la cámara, como la distancia focal, el punto principal y la relación de aspecto, que son esenciales para la proyección y mapeo precisos de las imágenes capturadas. En este proyecto se utilizó la clase *PinholeCameraIntrinsic* de la librería Open3D para representar y gestionar estos parámetros intrínsecos. Al emplear los valores adecuados, como *PrimeSenseDefault* en nuestro caso, configuramos los parámetros intrínsecos según las características específicas de la cámara Intel.

Generación de la nube de puntos

En visión por computadora, una nube de puntos es una representación tridimensional de una escena o objeto capturado mediante una cámara o un sistema de sensores. Consiste en un conjunto de puntos en el espacio 3D, donde cada punto tiene una ubicación y posiblemente también información adicional, como el color o la intensidad. Estos puntos se generan a partir de datos de profundidad. Empleando la imagen RGBD (que combina información de color y profundidad) y los parámetros intrínsecos de la cámara utilizada, la función *create_from_rgbd_image* de la librería Open3D permite generar la nube de puntos. Este proceso implica convertir los datos de la imagen RGBD en una representación tridimensional, donde cada punto de la nube corresponde a una posición en el espacio 3D. Al utilizar la información de profundidad y los parámetros intrínnsecos de la cámara, se logra mapear de manera precisa cada pixel en la imagen a su correspondiente ubicación espacial. La Figura 3a muestra la nube de puntos de una escena de exterior.

Posteriormente, se realiza un submuestreo, con el fin de reducir la cantidad de datos en una nube de puntos. Este proceso implica disminuir la resolución o densidad de la nube de puntos original. Se utiliza para aligerar la carga computacional durante el procesamiento, ya que una nube de puntos densa puede requerir más recursos y tiempo para su análisis. El submuestreo permite simplificar la nube de puntos al eliminar puntos

redundantes o mantener solo una muestra representativa de los datos originales. La Figura 3b muestra el submuestreo de la nube de puntos para la Figura 3a.

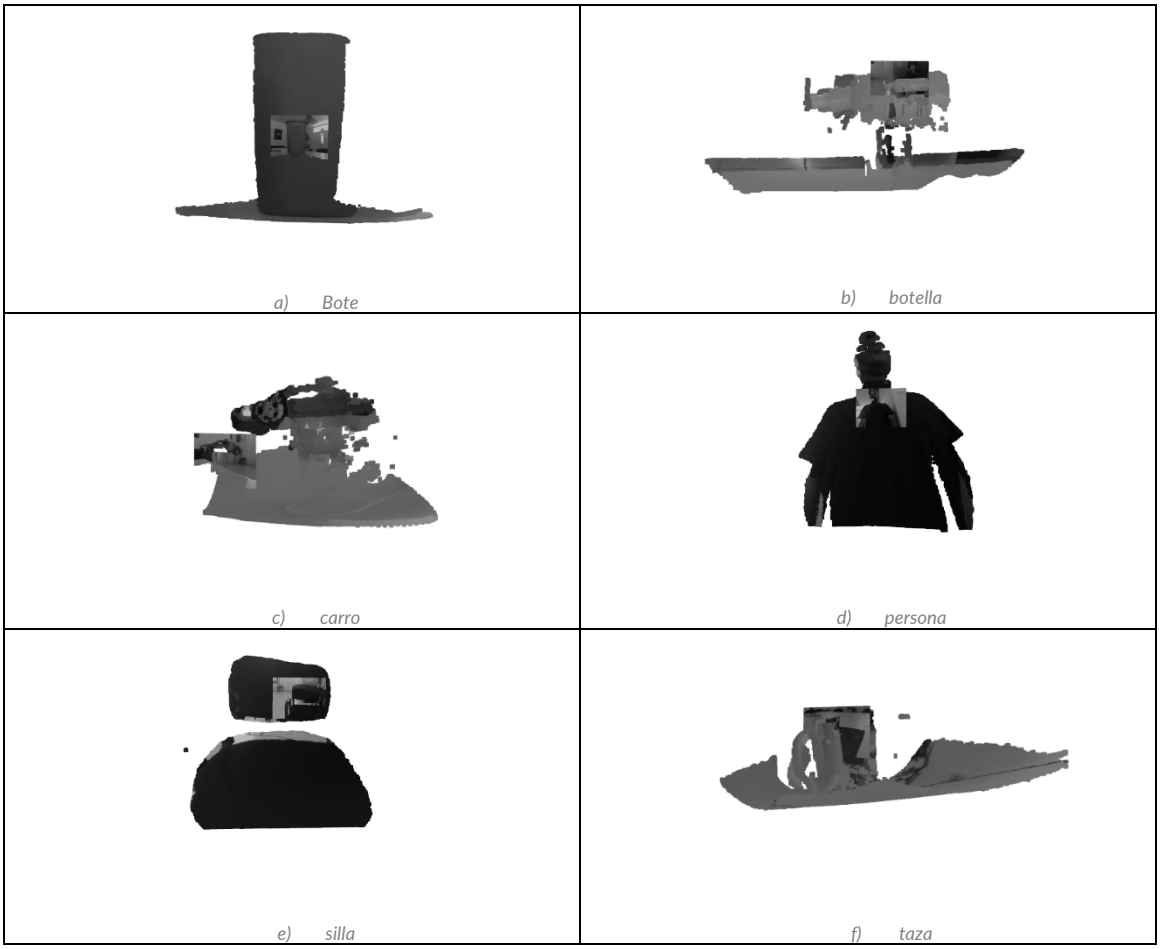


Figura 3. Nubes de puntos de los objetos utilizados para generar las categorías de la base de datos.

Reconstrucción de la escena

La reconstrucción 3d es una tarea importante dentro de la robótica autónoma y distintas áreas. Dentro de la robótica se utilizan distintos algoritmos de esta área y otras técnicas con el fin de obtener un mapa del entorno. Para lograr este objetivo, el automóvil o robot autónomo debe tener alguna forma de observar su entorno, en este desarrollo se obtiene una imagen profundidad-color y la función conocida como Función de signo truncado (TSDF por sus siglas en inglés). Este algoritmos de procesamiento está diseñado para procesar imágenes de profundidad-color con alto ruido. De forma general, el algoritmo TSDF se compone de dos módulos: 1) de activación, 2) de integración. La figura 4 muestra una descripción a bloques del proceso de reproyección de puntos, lo que permite la reconstrucción 3D de la escena.

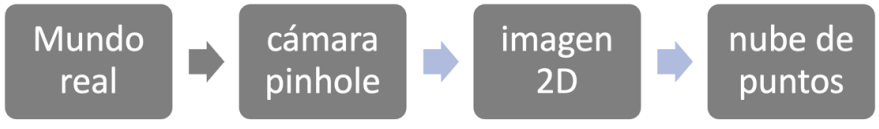


Figura 4. Metodología para la reconstrucción 3D de la escena desde el modelo pinhole de la cámara.

En el proceso de activación, primero se localizan los bloques o zonas sin puntos sin proyectar sobre la imagen de profundidad actual. En otras palabras, encuentra los bloques activos en la imagen actual que se está procesando. Internamente esto logra mediante un hash-map, optimizando que no haya duplicados con las mismas coordenadas. Una vez con este mapa de proyección en la imagen 2D, para obtener las coordenadas del mundo se realiza el siguiente procedimiento. Primero, desde el ambiente de navegación que etiquetamos como mundo real, se obtiene las señales del entorno de navegación. Después, con el modelo pinhole de la cámara, que consiste en los parámetros que controlan la geometría de la cámara, es posible tener la proyección de los puntos reales 3D hacia el plano imagen. La matriz de parámetros intrínsecos contiene los valores mostrados en la ec. 1.

$$p_{intr} = \begin{pmatrix} f_x & S & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \qquad \text{ec. 1}$$

donde f_x y f_y es el foco en x y y, respectivamente, c_x y c_y son las coordenadas de centro en (x,y) y S es el coeficiente de deformación. La imagen de dos dimensiones es la proyección de un punto del mundo real (Tridimensional) a un punto de dos dimensiones. Finalmente, para obtener la nube de puntos se ejecutan las ecs 2 a 4, donde d_s es el factor de escala de profundidad:

$$z = \frac{d}{d_s} \qquad \text{ec. 2}$$

$$x = \frac{(u - c_x)\, z}{f_x} \qquad \text{ec. 3}$$

$$y = \frac{(u - c_y) z}{f_y} \qquad \text{ec. 4}$$

Por otra parte, el módulo de integración se refiere al proceso mediante el cual se actualiza la representación geométrica 3D, en función de la nueva información recibida desde los sensores o datos de escaneo. Esto implica fusionar la información recién obtenida con la representación existente del objeto para mantener una descripción coherente y precisa. El proceso de integración en TSDF generalmente se realiza en un volumen 3D discretizado, conocido como "volumen TSDF". Cada celda o voxel dentro de este volumen contiene un valor que representa la distancia con signo al objeto más cercano en esa ubicación. Estos valores iniciales se establecen típicamente a un valor especial que indica que la celda está vacía o desconocida.

Resultados

Como se menciono en la metodología, el proceso de navegación del robot se lleva a cabo en dos partes: 1) la programación del robot para navegar en espacio de interior o de exterior, y 2) captura y preparación de las imágenes color profundidad para la reconstrucción de la escena y modelado de objetos.

Los resultados obtenidos al programar el robot es que realiza una rutina donde avanza alrededor de 15 metros en el laboratorio. Antes de programar el robot se estimaron parámetros de velocidad, ya que el robot solo recibe parámetros de potencia, en nuestro caso al 50%, por lo que se realizaron pruebas para conocer la velocidad en metros por segundo y tener mejor análisis de su navegación. Para calcular la velocidad se consideraron distintas métricas, como las revoluciones por segundo del motor y el diámetro de las llantas. Las pruebas para calcular la velocidad consistieron en mover el robot durante un tiempo fijo de 10 segundos y se midió la distancia que recorrió. Este mismo tiempo se considero para medir la velocidad a una potencia del motor de 50%, 75% y 100%. Los resultados obtenidos de velocidad se muestran en la Tabla 1.

Tabla 1. *Cálculo aproximado de la velocidad del robot en m/s considerando un tiempo de t=10 segundos*

Potencia	Rev/seg	Distancia recorrida (m)	Velocidad (m/s)
50%	1.43	4.50	0.45
75%	1.59	5.00	0.50
100%	1.75	5.50	0.55

Resultados de la navegación en interior

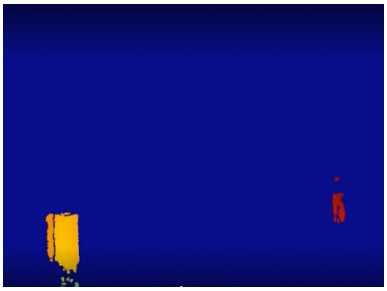
Una vez programado el robot y conociendo su velocidad sabemos que el robot navega a 0.45m/s, entonces se hicieron las capturas con la cámara desde el escenario al interior del laboratorio. Algunas imágenes adquiridas al tiempo t=11 y t= 14 segundos de haber iniciado el recorrido de navegación, se muestran en la figura 5.

Note que en el caso de la figura a) y b), a pesar de que el bote café que se ve al centro de la imagen es muy grande y en el campo de visión de la camara, la imagen de profundidad no es capaz de detectarlo en la imagen b) debido a que se encuentra a una distancia mayor del rango de detección de la cámara. Sin embargo, dos objetos mas cercanos color blanco y de forma cilindrica alcanzan a percibirse en la imagen de profundidad, el primero en color amarrillo denotando que es el más próximo a la cámara y el segundo en una región más pequeña se muestra en rojo, denotando que es un objeto que se encuentra más alejado de la cámara. Esta detección de la percepción es una de las desventajas que encontramos en el presente trabajo de investigación y por lo cual utilizamos la velocidad media del robot, es decir a potencia 50% para que no pase tan rapido en el escenario y haya posibilidad de detectar los objetos antes de que ocurra alguna colision del robot con ellos. El caso de las figuras c) y d) podemos ver que el objeto del bote es capturado en su totalidad en la imagen de profundidad, ya que ahora se encuentra en el campo de visión de la cámara.

Resultados de la navegación en interior

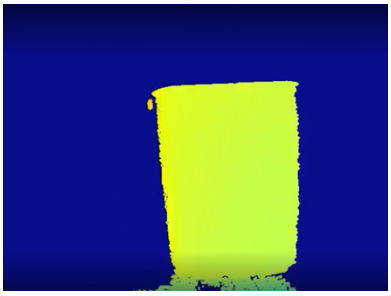
En el caso de la navegación en exterior, el campo de visión que se captura del escenario de navegación es mucho mas amplio y con muy diversas regiones en la escena. Por ejemplo una gran parte de la imagen es cielo, camino, la cerca, árboles, paredes de los edificios cercanos al estacionamiento. Por ello se decidio, no realizar el modelado de los objetos para escenas de exterior y únicamente se realizó la proyección de puntos de las imágenes. La Figura 6 muestra algunas de las proyecciones de los puntos realizados en el escenario 3D. La imagen a) muestra una parte del recorrido, se puede distinguir la estela de puntos en linea recta que denotan los puntos de interes más representativos capturados enfrente del robot. Note la estela de puntos graficados sobre el piso y algunas otras nubes que representan algunos de los objetos como botes y conos colocados a un lado del camino. En todo momento el robot tiene el escenario de navegación libre y solo se captura un escenario estático, es decir, ningún objeto se mueve solo el robot con la cámara. Los resultados obtenidos de la navegación en interior y exterior se muestran en un video complementario a este trabajo de investigación.

a) t=11 seg



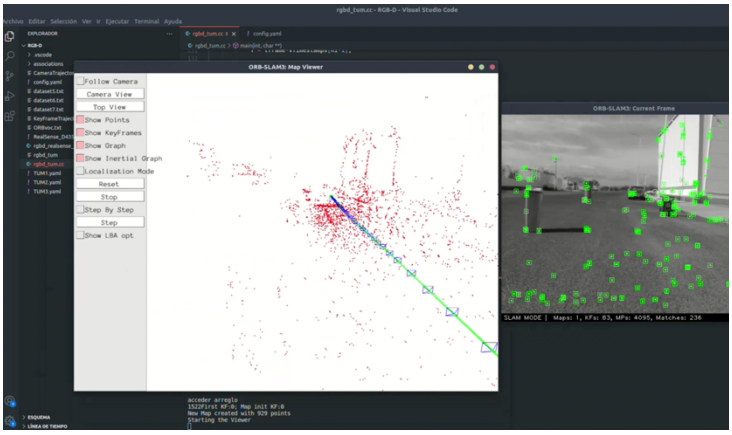
b) t=11 seg

c) t=13 seg

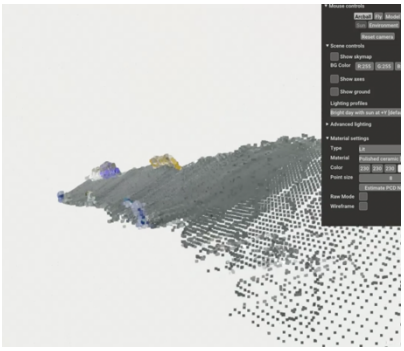


d) t=13 seg

Figura 5. a) y c) Imágenes color capturadas durante la navegación del robot. b) Imágenes de profundidad que muestran los objetos detectados.



a)



b)

Figura 6. a) Reproyección de los puntos de interés mostrados en verde sobre la imagen de la derecha, la línea verde representa la posición de la camara para cada imagen que capturo y los puntos rojos en el plot son las proyecciones 3D. b) Reconstrucción del escenario con objetos.

Conclusiones

En este proyecto se logró la reconstrucción 3D usando imágenes color y profundidad capturadas desde un vehículo en un ambiente de interior y exterior. Los módulos que se utilizaron son de la librería Open3D y apoya la identificación de las zonas libres de obstáculos, a fin de que el vehículo pueda navegar de forma autónoma. Se genero una base de datos con la cual se puede entrenar modelos convolucionales para clasificación. Se obtiene la representación de la escena vista desde el robot en un video demostrativo. En nuestra experiencia en este proyecto fue grata ya que no conocíamos antes la función de los sistemas autónomos que se desarrollan y se utilizan hoy día.

Bibliografía/Referencias

[1] P. Kumar. (2021) What is a tof sensor? What are the key components of a tof camera? [Online]. Available: https://www.e-consystems.com/blog/camera/technology/what-is-a-time-of-flight-sensor-what-are-the-key-components-of-a-time-of-flight-camera/

[2] (2022) What are rgbd cameras? why rgbd cameras are preferred in some embedded vision applications? [Online]. Available:https://www.e-consystems.com/blog/camera/technology/what-are-rgbd-cameras-why-rgbd-cameras-are-preferred-in-some-embedded-vision-applications/

[3] Pitsco, Inc. (2018). TETRIX® PRIZM® Robotics Controller Programming Guide. Pitsco, Inc. Pittsburg, KS.

[4] Q.-Y. Zhou, J. Park, and V. Koltun. (2018) “Open3D: A modern library for 3D data processing,” arXiv:1801.09847