La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Segmentación de obstáculos y detección del área transitable de un robot móvil usando cámara de profundidad

Altamirano Soria José Edgar, Ramos-Arreguin Juan-Manuel✉, Aceves-Fernández Marco-Antonio, Gorrostieta-Hurtado Efrén

Universidad Autónoma de Querétaro, Facultad de Ingeniería.
✉jsistdig@yahoo.com.mx

Resumen

En el presente trabajo se utiliza la cámara de profundidad Realsense D435 para detectar la superficie transitable por un robot móvil y los obstáculos que se encuentran en su camino. Para lo que se hace uso de técnicas de visión artificial para procesar las imágenes y la nube de puntos correspondientes a la escena del campo de visión de la cámara. La importancia de la detección del área transitable y obstáculos radica en que esta información es usada por algoritmos de planeación de trayectoria, evasión de obstáculos y sistemas de asistencia al conductor. Una de las ventajas de contar con la nube de puntos de la escena es que se puede modelar con la ecuación de un plano a los puntos pertenecientes a la superficie que representa al área transitable por el robot móvil. Para esto, se aplicó el método RANSAC (Random Sample Consensus) ya que no todos los puntos de la nube pertenecen al modelo buscado. Los puntos que quedan en la nube tras la eliminación de los puntos del área transitable y la selección de una región de interés (ROI), son los que pertenecen a los obstáculos. Mediante la proyección de estos puntos se logra obtener la ubicación de los obstáculos en la imagen correspondiente a la escena. El principal aporte de este artículo es la implementación de la nube de puntos para segmentar los obstáculos de la escena y representarlos, mediante una proyección, en la imagen de profundidad. La máscara binaria que se obtiene tras la proyección, es tratada con operaciones morfológicas para mejorar el resultado obtenido.

Palabras clave: Cámara de profundidad, Imagen de profundidad, Realsense D435, RANSAC, detección de obstáculos, robot móvil.

Abstract

In the present work, the Realsense D435 depth camera is used to detect the surface that can be transited by a mobile robot and the obstacles that are in its path. For the above, artificial vision techniques are used to process both the images and the point cloud corresponding to the scene of the field of view of the camera. The importance of the detection of the traversable area and obstacles lies in the fact that this information is used by trajectory planning algorithms, obstacle evasion and driver assistance systems. One of the advantages of having the point cloud of the scene is that the surface points that represent the traversable area by the mobile robot, can be modeled with the equation of a plane. For this, the RANSAC (Random Sample Consensus) method was applied since not all the points of the cloud belong to the searched model. The points that remain in the cloud after the elimination of the points of the traversable area and the selection of a region of interest (ROI), are those belonging to obstacles. By projecting these points, is possible to obtain the location of the obstacles in the image corresponding to the scene. The main contribution of this article is the implementation of the point cloud to segment the obstacles of the scene and represent them, through a projection, in the depth image. The binary mask obtained after the projection is treated with morphological operations to improve the result.

Keywords: Depth camera, depth image, Realsense D435, RANSAC, Obstacle detection, mobile robot.

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

1. Introducción

Los robots móviles que navegan en ambientes desconocidos, con cierto grado de autonomía, pueden ser utilizados para el transporte de mercancías o personas, incluyendo recolección de objetos, exploración en zonas desconocidas, entre otras aplicaciones. Una problemática importante es que sean capaces de detectar el camino que pueden seguir en su trayectoria. Es decir, que puedan detectar aquellas zonas correspondientes al plano del piso (*ground plane*) que no tienen objetos que impidan su paso, para planear su trayectoria y evitar colisiones. Esta misma problemática, puede ser aplicada a los vehículos comerciales, que cuenten con cierto grado de autonomía, o que cuenten con algún sistema avanzado de asistencia al conductor (ADAS por sus siglas en inglés)[1][2].

Otra aplicación común para la detección del piso es en sistemas de ayuda para personas con discapacidad visual. Se busca indicar a la persona el área libre de obstáculos que es posible transitar, y así evitar que la persona impacte con algún objeto [3][4][5].

En el tipo de aplicaciones mencionadas, se considera que el sensor usado para adquirir los datos de la escena de interés está montado sobre algún soporte que se desliza sobre el piso, y que los obstáculos que pueda haber en el camino se encuentran sobre el piso. De esta forma el área transitable es aquella en la que se logra detectar puntos pertenecientes al piso.

Los sensores más comúnmente usados para la detección del camino y obstáculos son: Lidar 3D, sensores RGB y sensores RGB-D.

La cámara o sensor RGB entrega imágenes a color o escala de grises, con la desventaja de perder información de la profundidad de la escena capturada. Esto debido a que el sensor RGB proyecta la escena o espacio tridimensional en un espacio bidimensional. Este tipo de sensor se utiliza generalmente para detectar carriles, es decir, un área delimitada por líneas punteadas o continuas. La detección de estas líneas se logra implementando técnicas de visión artificial (como la transformada de Hough), con lo que se consigue identificar el camino transitable [6].

Mediante los sensores Lidar y cámaras RGB-D se pueden obtener nubes de puntos de la escena de interés, en donde cada punto tiene tres coordenadas que indican su posición en el espacio tridimensional. La nube de puntos entregada por el Lidar es la más exacta y densa, pero tiene la desventaja de tener un alto costo económico, lo cual eleva de manera considerable el precio del sistema donde sea utilizado.

A diferencia del Lidar 3D, los sensores RGB-D como el Kinect o las cámaras de profundidad Realsense de Intel, también entregan imágenes de la escena. Esto permite tener un sistema más robusto al permitir procesar imágenes y nubes de puntos de la escena a analizar. Otra ventaja de los sensores RGB-D son su bajo costo en comparación al Lidar.

De los sensores RGB-D mencionados anteriormente, se decide usar el Realsense D435 debido a que sus dimensiones y especificaciones eléctricas lo hacen el más indicado para ser implementado en robots móviles.

Se utiliza el algoritmo RANSAC (Random Sample Consensus) [7] para estimar los parámetros que mejor se ajusten al modelo del plano con el que se busca representar al área transitable. Con este modelo se pueden extraer todos los elementos de la nube de puntos quesean parte del piso o área transitable.

En este trabajo, también se aplica una metodología para eliminar el ruido de la información adquirida por el sensor también conocidos como datos atípicos. También se busca clasificar los puntos entre aquellos que pertenecen al *background* (fondo) de la escena, como el piso y el ruido, y aquellos que pertenecen al *foreground* (primer plano), que son aquellos puntos que pertenecen a los obstáculos

que se encuentran en el camino del robot móvil. En el texto se usa los términos en inglés debido a que es el término técnico usado en la literatura.

La salida del sistema propuesto consta de los siguientes elementos:

- Nube de puntos e imagen, correspondientes a la escena de interés, con el piso y *foreground* identificados.
- Máscara o imagen binaria, para segmentar los obstáculos de la imagen proporcionada por el sensor.

2. Estado del arte

En el 2014, se presenta un nuevo algoritmo para detectar el piso que usa la información de profundidad entregada por los sensores RGB-D. El método propuesto se basa en el ajuste de una curva exponencial usada para modelar el piso, debido a que los puntos correspondientes al piso en las imágenes RGB-D muestran un comportamiento creciente en sus valores de profundidad. Cuando el ajuste de la curva modelada no es bueno o muestra un error considerable, la detección del piso es errónea [8].

En el 2015, Qian, busca desarrollar un algoritmo que permita que un robot móvil navegue en pasillos desconocidos, usando un sensor Kinect. Con la información RGB-D obtenida por el sensor, se aplica el método RANSAC a la nube de puntos para obtener la ecuación del plano que describe al piso. De las imágenes se extrae información de las líneas que unen al piso con las paredes para mejorar los resultados obtenidos [9].

En este mismo año, se implementa un sistema que permite detectar objetos usando un sensor RGB-D. Se busca segmentar a los objetos que se encuentran sobre el camino y que puedan obstruir el paso del dispositivo sobre el que se encuentre montado el sensor de profundidad. Se usan técnicas de etiquetado que permiten identificar a cada objeto de acuerdo a su posición, por lo que es necesario segmentar el piso, dado que la profundidad de la intersección entre los objetos y el piso es la misma. Por esto, se necesita segmentar el piso antes de hacer la detección de los objetos. Para segmentar el piso se usa el mapa de disparidad V con cuya información se puede buscar aproximar el modelo del piso mediante una ecuación de segundo orden. Dicha aproximación se hace por medio del método de mínimos cuadrados, aunque los parámetros calculados no siempre predicen de forma correcta los valores del piso. Este trabajo no obtiene la nube de puntos para detectar el piso, por lo que no muestra la escena con el piso segmentado en el plano x-z para visualizar de mejor forma la ubicación de los obstáculos [10].

En el 2016, Shumin Liu, obtiene información de profundidad mediante una cámara estereoscópica para obtener el plano x-z de la escena, en donde segmenta los objetos detectados de acuerdo a su posición con el propósito de obtener sus contornos. Para poder distinguir a los objetos en el plano x-z, se eliminaron los puntos pertenecientes al piso, bajo la suposición de que todos los puntos con una altura mayor a 0.3 metros pertenecen al piso. Para poder hacer esta suposición se debe saber de antemano la altura que hay del piso al sensor, lo cual dificultaría la implementación del sistema en otros robots o vehículos dependiendo de las necesidades de cada situación; además de que se debe tener una alta precisión en los datos adquiridos por el sensor usado [11].

En este mismo año, Pham propone un método para ayudar a las personas ciegas a evadir obstáculos mediante el uso de un sensor Kinect. La información RGB-D del Kinect es adquirida y procesada mediante la lo librería PCL (Point cloud library), que es una de las librerías más comúnmente usadas para el procesamiento de nubes de puntos. Para lograr separar el suelo de los objetos que hay en el camino se busca modelar el piso mediante la ecuación de un plano utilizando el método RANSAC. Una desventaja de este trabajo es que no se detecta el piso dentro de la imagen generada por el sensor (a color o escala de grises) lo que limita la detección de otros objetos dentro de la imagen [12].

En el área de robótica móvil, Aladren presenta un sistema de navegación que permite a su usuario navegar a través de un ambiente desconocido. Se usa un sensor RGB-D del cual se utiliza la información de profundidad y color para detectar el camino libre de obstáculos. Se utiliza el método RANSAC para segmentar el plano de la nube de puntos y el área detectada se mejora o "alarga" haciendo uso de la información de color. Cabe destacar que la mejora del alcance de la detección del piso por medio de la imagen a color era debido a que el alcance en profundidad del sensor que se uso era limitado (3 metros) [4].

Para la detección de obstáculos en robots móviles, Yang propone un sistema para detectar y distinguir entre obstáculos con y sin movimiento utilizando una cámara RGB-D. Se obtiene una nube de puntos por medio de un sensor Kinect y se segmenta el piso para tener una mejor visibilidad de los objetos en el plano x-z. Para extraer los puntos del piso se aplica el método de mínimos cuadrados para calcular los parámetros de la ecuación del plano que mejor se ajusten a *n* cantidad de puntos del piso previamente seleccionados. La principal desventaja de esta propuesta es la previa selección de los puntos del piso para ajustar los parámetros de la ecuación del plano, lo que se tendría que hacer cada que se ajuste la altura del sensor. Los resultados de este articulo solo muestran al piso segmentado en el plano x-z (2d) de la nube de puntos. También se indica que la detección de los objetos no es del todo exacta debido al ruido que tienen los datos del Kinect [13].

En el 2017, se implementa el método RANSAC para distinguir entre obstáculos y piso con la información obtenida por un Kinect montado sobre una plataforma móvil. Para obtener los parámetros del plano que describa al piso, se toman varias muestras con el sensor montado a la plataforma móvil. Además, se da una comparación de la implementación de su algoritmo con y sin la aplicación de filtros [14].

3. Marco teórico

3.1 Visión por computadora y cámaras

La visión por computadora busca entender una escena dada utilizando información visual. Desde el punto de vista de hardware, los sistemas de visión son transductores que miden la intensidad de la luz, produciendo comúnmente imágenes o secuencias de imágenes (video) y, en algunos casos, nubes de puntos.

La cámara es un sensor que captura información de la luz proporcionada por el entorno, la transforma en una cantidad física procesable y visualiza su mapa de medición desde su punto de vista. La cámara es un dispositivo óptico y de no-contacto debido a que aprovecha las propiedades de la luz y no requiere contacto físico con el entorno a sensar [15].

3.2 Métodos para la detección de profundidad

Existen varias técnicas de imagen que pueden proporcionar la coordenada de profundidad *z* que se pierde al proyectar un punto con tres dimensiones en el plano de dos dimensiones de la imagen. En este trabajo, se mencionan los dos principales métodos para la obtención de imágenes ópticas de profundidad conocidos como paradigmas de profundidad: a) Profundidad a partir de triangulación y b) Profundidad a partir del tiempo de vuelo (TOF por sus siglas en inglés) [16].

3.2.1 Profundidad a partir del tiempo de vuelo.

Estas técnicas miden el tiempo que le toma a una señal viajar cierta distancia. Si la señal se envía desde la posición de la cámara, tiene que viajar el doble de la distancia entre la cámara y el objeto que refleja la señal. El tiempo T que le toma a la señal viajar del sensor al objeto y regresar al sensor esta dado por (1).

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

$$T = \frac{2z}{c} \quad (1)$$

Donde:
z= distancia entre la cámara y el objeto reflector.
c= velocidad de la señal
T=Tiempo que le toma a la señal regresar a la cámara.

La figura 1 muestra la configuración del sistema para la adquisición de profundidad por tiempo de vuelo.

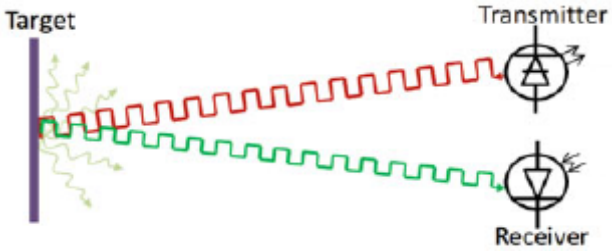


Figura 1.Funcionamiento del tiempo de vuelo [15].

3.2.2 Profundidad a partir de triangulación.

Si se observa un objeto desde dos puntos de vista separados por una distancia *b*, denominada línea base, el objeto será visto bajo distintos ángulos desde la línea base en ambas posiciones. Esta diferencia en el ángulo de visión da como resultado un desplazamiento en el plano de la imagen, conocido como disparidad, a partir de la cual se puede inferir la profundidad del objeto. [16]

Las mediciones de profundidad basadas en la triangulación incluyen una amplia variedad de técnicas diferentes que, a primera vista, no tienen mucho en común, pero siguen basándose en el mismo principio. En este artículo es de principal interés la estereoscopía.

3.2.3 Estereoscopia o visión estéreo

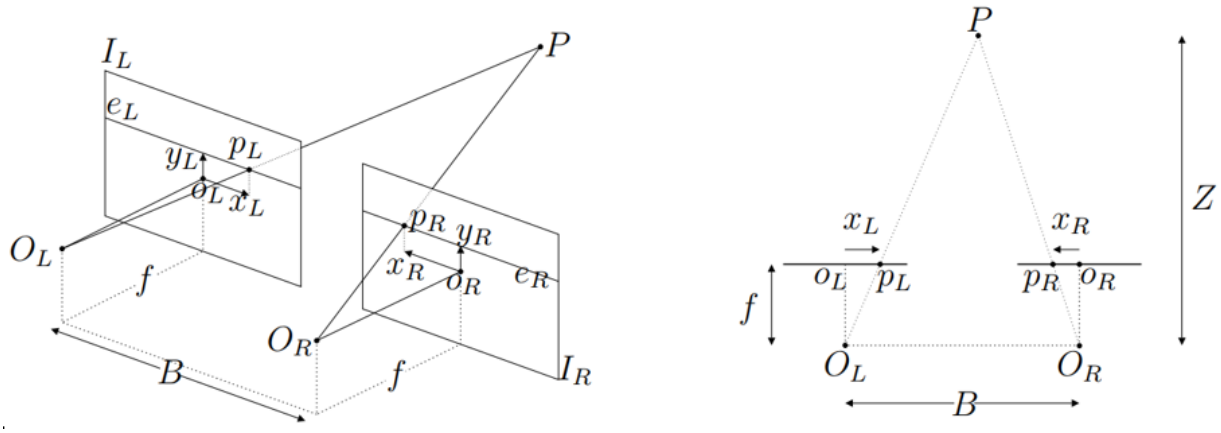
A las configuraciones que constan de dos sensores de imagen (cámaras) separados por una distancia conocida se les conoce como sistemas estéreo o estereoscópicos. El principio de la estereoscopia está basado en la capacidad del cerebro humano para estimar la profundidad de los objetos presentes en las imágenes capturadas por los dos ojos.

En la configuración estereoscópica, dos cámaras se colocan una cerca de la otra con ejes ópticos paralelos. La configuración se muestra en la figura 2a. Ambas cámaras, con centros O_L y O_R separados por una distancia *B* denominada base, tienen la misma distancia focal f, de forma que las imágenes I_L e I_R se encuentran en planos paralelos. Un punto en el espacio tridimensional P se proyectará en diferentes posiciones, p_L y p_R con coordenadas $(x_L, y_L, 1)$ y $(x_R, y_R, 1)$ respectivamente, de los planos de las imágenes porque se ve desde ángulos ligeramente diferentes. A esta diferencia en la posición se le conoce como disparidad y matemáticamente se describe en (2) y es usada para calcular la distancia *z*, mostrada en la figura 2b, que hay entre el punto *P* del espacio tridimensional y el sistema estereoscópico. La ecuación para obtener la profundidad *z* se muestra en (3) [16].

$$disparidad = x_R - x_L \quad (2)$$

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

$$z = \frac{B*f}{disparidad} \quad (3)$$



a) Configuración de la cámara estéreo. b) Relación geométrica para obtención de profundidad

Figura 2. Estructura geométrica de la cámara estereoscópica [17]**.**

Los métodos activos mejoran la adquisición de la profundidad mediante el uso de una fuente de iluminación externa que proporciona información adicional al sistema. El principio activo de estereoscopía es similar al pasivo, pero busca características proyectadas artificialmente que sirven como información adicional para la triangulación [15].

3.3 Cámara 3D

En este artículo son de especial interés las cámaras 3D activas, que extraen mapas de profundidad o mapas de rango, como se muestra en la figura 3, proporcionando información de profundidad y de color. Las cámaras 3D más comunes y recientes, se basan en tecnologías de tiempo de vuelo y estereoscopía activa. Recientemente ganaron interés entre la comunidad de visión por computadora, gracias a sus precios accesibles y a su amplia gama de aplicaciones. [15]

Las principales cámaras 3D que hay en el mercado son el “Kinect v2” y la “Intel Realsense Depth Camera D435”.

3.3.1 Kinect V2

El Kinect V2 es una cámara 3D desarrollada por Microsoft que obtiene la profundidad a partir del tiempo de vuelo. Algunas de las principales características de este sensor se muestran en la tabla 1.

Tabla 1. Características del Kinect v2

Características	**Kinect v2**
Cámara RGB	1920 x 1080
Cámara de profundidad	512 x 424
Distancia máxima	4.5 m
Distancia mínima	50 cm

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

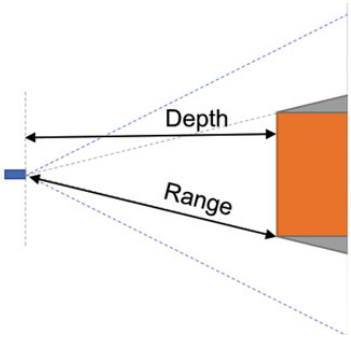


Figura 3. Medidas de profundidad [15].

Las medidas del sensor son: 24.9 x 6.6 x 6.7 cm (largo x ancho x alto) y requiere una conexión a corriente alterna para alimentarse. Las dimensiones y características de alimentación lo hacen una opción poco viable para implementarse en robots móviles o sistemas móviles. En la figura 4 se muestra al Kinect v2.

Figura 4.Kinect v2.

3.3.2 Intel Realsense Depth Camera D435

La cámara 3D de Intel utiliza visión estéreo activa para calcular la profundidad. Cuenta con un proyector de infrarrojos, módulo de visión estereoscópico, procesador de visión y sensor RGB. El caculo de la profundidad se realiza con el procesador de visión D4 que se encuentra dentro de la cámara.

El control de las características de adquisición de la cámara, modos de operación y datos extrínsecos e intrínsecos asociados a esta pueden ser realizados por medio del API desarrollado por Intel. Las principales características en la cámara se muestran en la tabla 2.

Tabla 2. Características de la cámara estereoscópica D435

Características	**Intel Realsense Depth Camera D435**
Cámara RGB	1920 x 1080
Cámara de profundidad	1280 x 720
Distancia máxima	10 m (puede ser más dependiendo de la iluminación)
Distancia mínima	0.2 cm

La alimentación del sensor es vía USB y sus dimensiones son: 90mm x 25 mm x 25mm (largo x ancho x alto). La figura 5 muestra la cámara de profundidad D435.

Figura 5. Cámara de profundidad D435 de Intel.

3.4 Imágenes de profundidad y nubes de puntos.

Las cámaras de profundidad entregan imágenes de profundidad, es decir, imágenes cuyos valores de intensidad representan la profundidad del punto *(x, y)* en la escena. Una nube de puntos es una estructura de datos utilizada para representar puntos con tres dimensiones *(X, Y, Z)* en donde la profundidad está representada por medio de la coordenada *Z*.

Al tener las imágenes de profundidad es posible obtener la nube de puntos haciendo uso de los valores intrínsecos de la cámara con la que fue adquirida la información, a este proceso se le conoce como deproyección. Un punto *P* con coordenadas *(X,Y,Z)* puede ser obtenido de acuerdo a (4), (5) y (6) a partir de la información de profundidad $D_{x,y}$, siendo (x,y) la posición rectificada del pixel en el sensor [15].

$$X = \frac{D_{x,y} * (c_x - x)}{f_x} \tag{4}$$

$$Y = \frac{D_{x,y} * (c_y - y)}{f_y} \tag{5}$$

$$Z = D_{x,y} \tag{6}$$

Las variables c_x, c_y, f_x y f_y son los valores intrínsecos de la cámara usada para adquirir la información, siendo (f_x, f_y) las componentes de la distancia focal y (c_x, c_y) el centro de proyección de la imagen [15]. La proyección es el proceso inverso a la deproyección con el cual se convierten los puntos de la escena en imágenes. Las fórmulas para llevar a cabo este proceso se encuentran en la metodología.

3.5 RANSAC

El método RANSAC (Random Sample Consensus) fue presentado en primera instancia por Fischler y Bolles[7], como un método para estimar los parámetros de cierto modelo utilizando datos con *outliers* (datos atípicos), es decir, que no todos los datos se ajustan al modelo buscado. En caso de aplicar el método de mínimos cuadrados utilizando todos los datos (incluyendo *outliers*), buscando minimizar el error del modelo deseado a los datos, sería contraproducente ya que los *outliers* afectarían de forma significativa a los parámetros calculados.

El porcentaje de *outliers* soportados por este algoritmo para calcular los parámetros de un modelo, es mayor al 50% [18].

Los pasos principales del algoritmo son:

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

1) Se seleccionan muestras de la población al azar, los cuales se denominan MSS (minimal sample sets). El número de elementos de la población seleccionados es el mínimo necesario para calcular el modelo, es decir, en caso de querer modelar una recta basta con dos puntos, en caso de querer modelar un plano se requieren mínimo tres puntos.

   Una vez que se tiene el MSS se procede a calcular los parámetros θ del modelo deseado usando únicamente los datos del MSS por medio del método de mínimos cuadrados. El espacio del modelo μ está definido por (7).

$$\mu(\boldsymbol{\theta}) = \{\boldsymbol{d} \in R^d : f_\mu(\boldsymbol{d}, \boldsymbol{\theta}) = 0\} \tag{7}$$

   Donde:

   d es el vector de los parámetros.
   f_μ es una función cuya igualdad a cero contiene todos los puntos que se ajustan al modelo $\mu(\boldsymbol{\theta})$

2) Una vez que se tiene el modelo, se calcula la distancia del modelo calculado a cada uno de los datos de la base de datos D, también conocida como error, mediante (8) y en caso de que esta distancia sea menor a un valor δ, se considera como parte del modelo (inlier). Al conjunto de *inliers* calculado en este paso se denomina "consensus set" y está definido por (9).

$$e_\mu(\boldsymbol{d}, \boldsymbol{\theta}) = \min_{d' \in \mu(\boldsymbol{\theta})} dist(\boldsymbol{d}, \boldsymbol{d'}) \tag{8}$$

$$S(\boldsymbol{\theta}) = \{\boldsymbol{d} \in D : e_\mu(\boldsymbol{d}, \boldsymbol{\theta}) \leq \delta\} \tag{9}$$

   El set de datos es evaluado con la función costo mostrada en (10). Esta función califica a cada set de parámetros $\boldsymbol{\theta}$ dependiendo del error que presentan los datos con respecto al modelo encontrado [19].

$$C_\mu(D, \boldsymbol{\theta}) = \sum_{i=1}^{N} \rho(d_i, \mu(\boldsymbol{\theta})) \tag{10}$$

   $\rho\big(\boldsymbol{d}, \mu(\boldsymbol{\theta})\big)$ está definido en (11).

$$\rho\big(\boldsymbol{d}, \mu(\boldsymbol{\theta})\big) = \begin{cases} e_\mu(\boldsymbol{d}, \boldsymbol{\theta}) & e_\mu(\boldsymbol{d}, \boldsymbol{\theta}) \leq \delta \\ \delta & de\ otra\ forma \end{cases} \tag{11}$$

En (10) y (11) se aprecia que a los datos que no forman parte de los *inliers* se les asigna un valor constante δ y a los *inliers* se les califica de acuerdo a que tan bien se ajustaron al modelo. Las ecuaciones (10) y (11) representan una modificación al método RANSAC original, y este método lleva el nombre de M-SAC [19].

La cantidad de iteraciones necesaria para encontrar el C.S. (consensus set) apropiado o con mejor calificación, es descrito en [7].

4. Metodología

En la figura 6 se muestra un diagrama de la metodología implementada, la cual se considera hasta el punto de reconocer los obstáculos de forma general, sin llegar a la planeación de trayectorias.

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

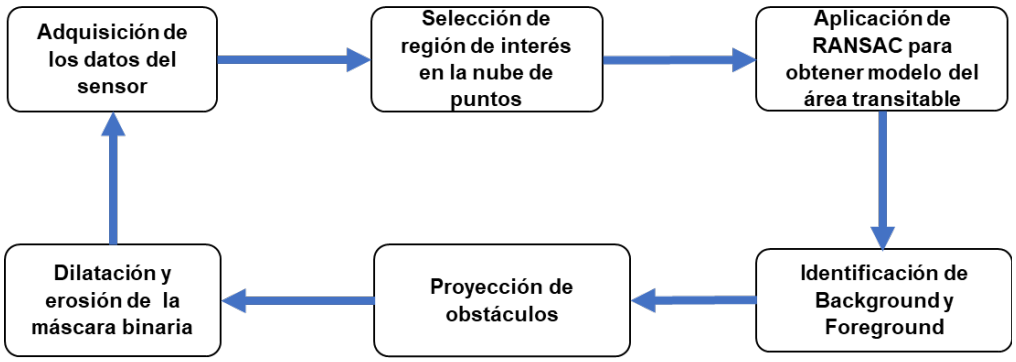


Figura 6. Metodología.

4.1 Adquisición de los datos del sensor

Se utiliza la cámara de visión estéreo "Intel Realsense Depth Camera D435", mencionada en el marco teórico, para capturar la escena a analizar. Por medio de las librerías proporcionadas por Intel para hacer uso de las características de la cámara, se adquirió la nube de puntos texturizada, la imagen de profundidad y una de las imágenes que utiliza el sensor para obtener la información de profundidad. Cabe destacar, que la nube de puntos texturizada se adquiere para obtener la imagen a color que este alineada con la imagen a profundidad, ya que las cámaras no están alineadas físicamente en el dispositivo. La nube de puntos también puede ser obtenida mediante (4), (5) y (6), haciendo la deproyección de la imagen de profundidad.

En este paso también se adquieren los valores intrínsecos de la cámara, los cuales son utilizados para realizar la proyección de la nube de puntos.

Esta metodología puede ser usada con otros sensores de profundidad, como el Kinect, siempre y cuando se tenga la imagen de profundidad y los valores intrínsecos del sensor.

4.2 Selección de región de interés en la nube de puntos.

La región de interés (ROI por sus siglas en inglés) se establece con el objetivo de delimitar la región a procesar en la nube de puntos, considerando todo aquello que este fuera de esta, como parte del *background*, es decir, aquellos puntos que no son de interés para el procesamiento. Esta región de interés se define por medio de valores de umbral, que buscan seleccionar aquellos puntos *p* cuyas coordenadas cumplan con (12).

$$\begin{cases} -x_{max} < p_x < x_{max} \\ p_y < h_{max} \\ p_z < z_{max} \end{cases} \tag{12}$$

Donde x_{max}, h_{max} y z_{max} son umbrales que permiten delimitar los valores de las coordenadas de los puntos. Estos valores de umbral están dados en metros, debido a que las coordenadas de cada punto están dadas en esta unidad.

4.3 Aplicación del método RANSAC para obtener el modelo del área transitable

Siguiendo los pasos del método RANSAC descrito en el marco teórico, se seleccionan tres puntos al azar de la nube de puntos (esto debido a que solo se necesitan 3 puntos para generar un plano). El modelo del plano que contenga a estos tres puntos puede ser descrito mediante el sistema de ecuaciones mostrado en (13).

$$\begin{cases} \theta_1 x^{(1)} + \theta_2 y^{(1)} + \theta_3 z^{(1)} + \theta_4 = 0 \\ \theta_1 x^{(2)} + \theta_2 y^{(2)} + \theta_3 z^{(2)} + \theta_4 = 0 \\ \theta_1 x^{(3)} + \theta_2 y^{(3)} + \theta_3 z^{(3)} + \theta_4 = 0 \end{cases} \quad (13)$$

Si hacemos $\boldsymbol{p^{(N)}} = (x^{(N)}, y^{(N)}, z^{(N)})$ y $\boldsymbol{\theta^T} = (\theta_1, \theta_2, \theta_3, \theta_4)$ entonces podemos expresar el sistema de ecuaciones de (13) como se muestra en (14).

$$\begin{pmatrix} \boldsymbol{p^{(1)}} & 1 \\ \vdots & \vdots \\ \boldsymbol{p^{(N)}} & 1 \end{pmatrix} = A\boldsymbol{\theta} = 0 \quad (14)$$

Se buscan los parámetros $\widehat{\boldsymbol{\theta}}$ que minimicen la distancia del plano a los puntos $\boldsymbol{p}$ por medio del método de mínimos cuadrados de acuerdo a (15).

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in R^4}{\mathbf{argmin}} \|A\boldsymbol{\theta}\|^2 \quad (15)$$

Una vez obtenidos los parámetros, se comprueba que el plano encontrado tenga como normal al vector *(0,1,0),* para asegurarnos de que el plano encontrado sea paralelo al plano x-z; en caso de no ser así, se desechan estos parámetros y se seleccionan al azar otros tres puntos para calcular nuevos parámetros.

Después de obtener los parámetros se obtiene el error e_μ o distancia euclidiana, de acuerdo a (16), de cada punto de la nube al plano descrito por los parámetros $\widehat{\boldsymbol{\theta}}$.

$$e_\mu = \frac{(\theta_1 x^{(N)} + \theta_2 y^{(N)} + \theta_3 z^{(N)} + \theta_4)^2}{{\theta_1}^2 + {\theta_2}^2 + {\theta_3}^2} \quad (16)$$

Con los errores calculados se evalúa la función costo descrita en (10), siendo el valor de δ usado en este trabajo de 1 centímetro; todos los puntos que se encuentren a una distancia δ del plano evaluado, serán considerados como *inliers* o parte del modelo. El resultado de la evaluación costo se almacena y compara contra las evaluaciones de los modelos obtenidos en cada iteración.

Este proceso se repite h veces y al final el modelo final es aquel que tenga una mejor evaluación de acuerdo a la función costo. La cantidad de iteraciones *h* necesaria para encontrar el C.S. (consensus set) apropiado, es descrito en [7].

Una vez obtenidos los parámetros del plano, se procede a identificar a aquellos puntos que se ajustan al modelo y que pertenecen al área transitable (*inliers*) y aquellos que no se ajustan al modelo (*outliers*).

4.4 Identificación de background y foreground.

Se obtiene el valor de la componente en *y*, h_{min}, con menor valor dentro de los puntos pertenecientes a los *inliers*, para después eliminar todos los puntos pertenecientes a los *outliers* cuya componente en el eje *y* sea menor a h_{min}, como se muestra en (17) y (18).

$$h_{min} = \min(Inliers) \quad (17)$$

$$\forall p \in Outliers\ ,\ p_y > h_{min} \quad (18)$$

Lo anterior se hace con el propósito de eliminar valores atípicos en los datos, ya que en ocasiones el sensor arroja valores erróneos que están por debajo del plano que representa al piso. Estos valores

erróneos también se encuentran en puntos con valores elevados en las coordenadas *x* y/o *z*, pero se eliminan en la selección de la región de interés descrita anteriormente.

En este punto se pueden generar las vistas aéreas, para lo que se grafican las coordenadas *(x, z)* en un plano bidimensional de los puntos pertenecientes a los *inliers* y *outliers*. Estas graficas representan una de las salidas de la metodología propuesta, ya que mediante estas se pueden tomar decisiones de evasión y planeación de trayectoria (que no se abordan en este trabajo) debido a que se tienen identificados tanto el área del piso por la que puede transitar el robot como los puntos que representan obstáculos. Cabe destacar que la unidad de medida de las gráficas mencionadas es el metro.

Los *outliers* contienen a la región de interés sin los puntos que pertenecen al área transitable y sin datos atípicos o ruido, los cuales fueron eliminados previamente en esta metodología, por lo que representan al *foreground* de la escena, es decir, aquellos puntos que pertenecen a objetos que representan obstáculos para el robot móvil. El *background* representa entonces a los puntos que no representan un obstáculo, como el piso, ruido y objetos fuera de la región de interés.

4.5 Proyección de obstáculos.

El siguiente paso es representar al *foreground* en una imagen binaria (mascara binaria) que permita eliminar los puntos, que no representan obstáculos, de la imagen adquirida por la cámara de profundidad.

La transformación de las coordenadas en tres dimensiones (p) a la imagen en 2 dimensiones (Im) se logra mediante la proyección de los puntos haciendo uso de los valores intrínsecos f_x, f_y, ppx, ppy, como se muestra en (19) y (20).

$$Im_x = \frac{p_x}{p_z} * fx + ppx \tag{19}$$

$$Im_y = \frac{p_y}{p_z} * fy + ppy \tag{20}$$

Después de hacer los cálculos de proyección, las coordenadas obtenidas de las posiciones de los pixeles son valores flotantes, por lo que se deben aproximar al valor entero más cercano y eliminar los datos que tengan algún valor menor a 1.

En cada una de las coordenadas obtenidas de la proyección, se almacena un valor lógico ‘1’ y en las demás posiciones ‘0’, esto con la finalidad de generar la máscara.

4.6 Dilatación y erosión de la máscara.

La máscara obtenida se erosiona dos veces con el propósito de eliminar líneas delgadas y puntos aislados; después se dilata dos veces para que aquellas zonas que no se eliminaron después de la erosión vuelvan a su forma original y para que se eliminen las zonas vacías dentro de regiones cerradas.

En (21) se muestra el elemento estructurante utilizado para la erosión, mientras que en (22) se muestra el elemento estructurante utilizado para la dilatación.

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{22}$$

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Al multiplicar la máscara por alguna de las imágenes obtenidas en la adquisición de datos, se obtendrá la imagen con los obstáculos segmentados.

5. Resultados

La metodología propuesta es probada utilizando la cámara de profundidad Realsense D435 de Intel. Para esto, se ha trabajado en una zona de pruebas mostrada en la figura 7, adquirida con la cámara. De esta escena, la cámara entrega una imagen de profundidad, que se muestra en la figura 8.

Figura 7. Imagen RGB.

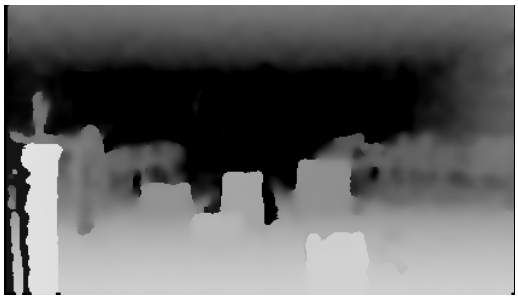


Figura 8. Imagen de profundidad.

La nube de puntos texturizada mostrada en la figura 9, también corresponde a la escena mostrada en la figura 7.

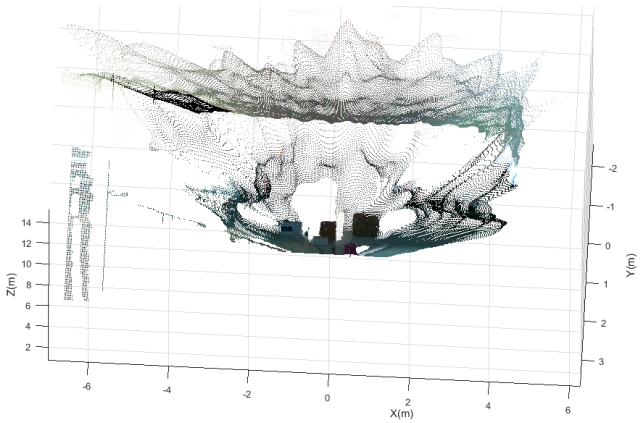


Figura 9. Nube de puntos original.

La figura 10 muestra una de las imágenes del par estereoscópico usado por la cámara para calcular la profundidad. Los puntos blancos que se aprecian en la imagen corresponden a el patrón de la fuente activa de la cámara. Esta imagen se corresponde pixel a pixel con la imagen de profundidad mostrada en la figura 8, contrario a la imagen RGB mostrada en la figura 7, que no está alineada con la imagen de profundidad.

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Figura 10. Imagen izquierda del módulo estéreo.

Al alinear la imagen RGB con la imagen de profundidad se obtiene la figura 11. En esta se puede apreciar que hay una pérdida de información en el contorno de la imagen.

Figura 11. Imagen RGB emparejada con la imagen de profundidad.

La ubicación de la cámara está representada por el origen (0,0,0) de la nube de puntos, cuya unidad de medida es el metro.

La imagen de profundidad de la figura 8 es una imagen a escala de grises de 424x240 pixeles, y 16 bits de profundidad por lo cual toma valores de intensidad en el rango: [1 2^{16}]. Cada incremento en una unidad del valor de intensidad de cada pixel representa 0.0010000000475 metros, que es la escala de profundidad del sensor utilizado. Al deproyectar la imagen de profundidad de acuerdo a (4), (5) y (6) se puede obtener la nube de puntos. Los valores intrínsecos de la cámara de profundidad son:

- ancho: 424, altura: 240, ppx: 214.063, ppy: 120.214, fx: 213.177, fy: 213.177

Ancho y altura describen el número de columnas y renglones de la imagen de profundidad. Los valores fx y fy describen la distancia focal de la imagen como un múltiplo del ancho y altura de la imagen.

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Los valores ppx y ppy describen las coordenadas de los pixeles del centro de proyección (que no necesariamente es el centro de la imagen).

Los puntos pertenecientes a la región de interés se muestran en la figura 12. Se puede observar que la profundidad se acoto a 4 metros, de -4 metros a 4 metros en el eje *x,* y una altura máxima de 1.8 metros, teniendo como punto de origen la posición de la cámara.

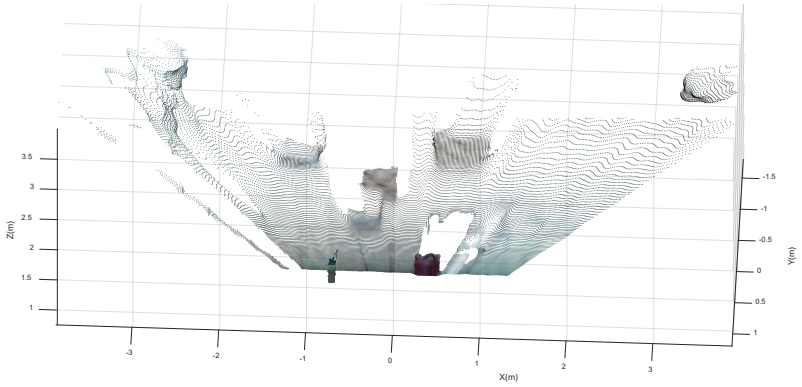


Figura 12. Selección de la región de interés.

Se aplica el algoritmo RANSAC a la nube de puntos resultante de la selección de la región de interés, para la detección del plano que representa al área transitable. En la figura 13 se eliminan todos los puntos que no pertenecen al área transitable (*outliers*) y en la figura 14 se muestra el área transitable (*inliers*) que es el area que puede ser transitada.

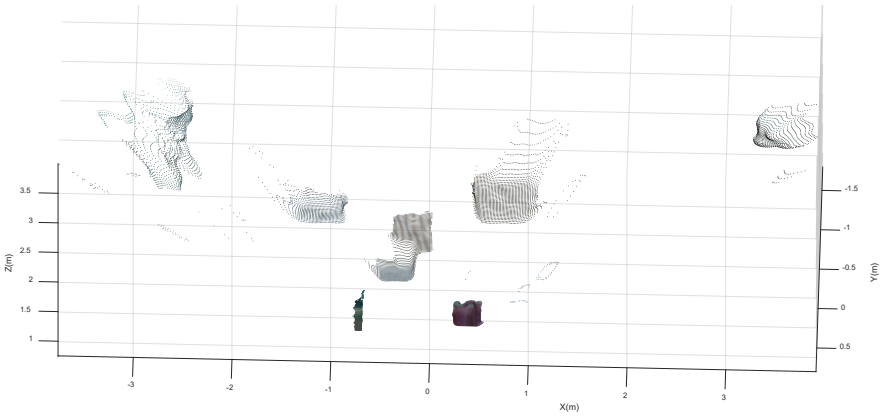


Figura 13. Puntos pertenecientes a los *outliers*.

Las figuras 15 y 16 muestran las vistas aéreas de la escena con y sin el piso después de haber detectado a los *inliers* y *outliers*. Estas tomas pueden ayudar a planificar la trayectoria que debe tomar el robot móvil para no colisionar con los obstáculos.

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

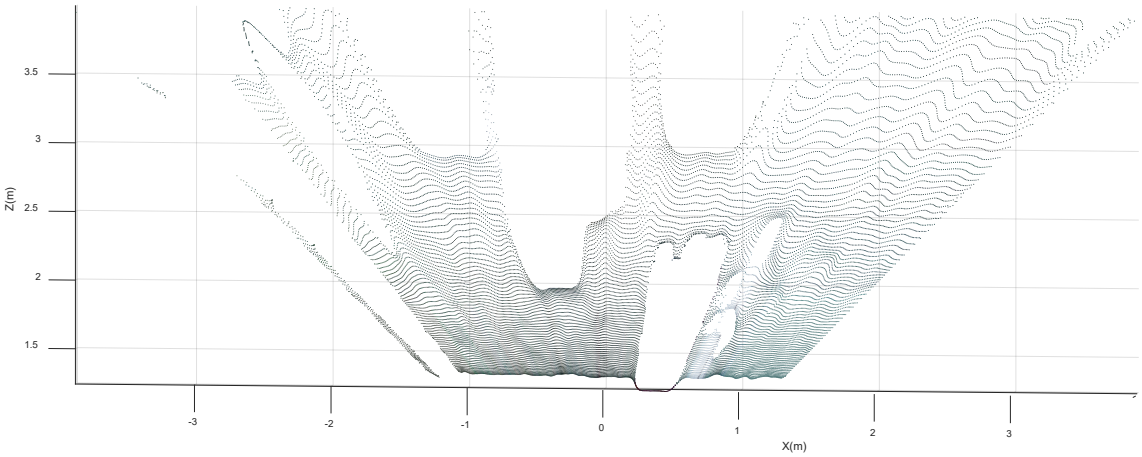


Figura 14. Puntos de la escena pertenecientes al área transitable.

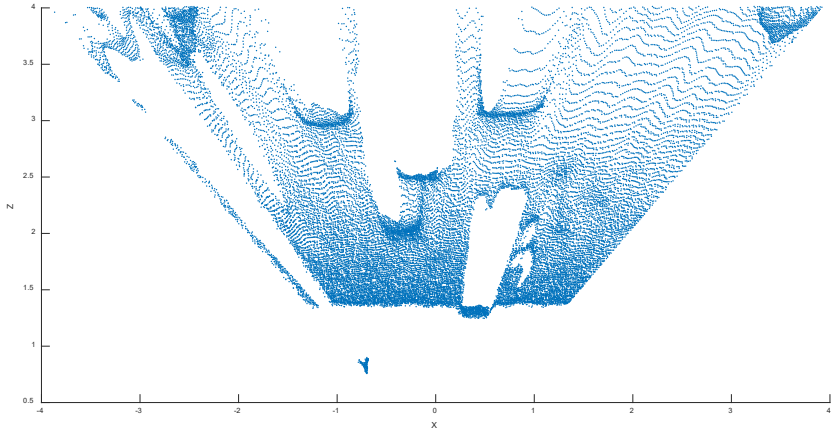


Figura 15. Vista aérea de la región de interés.

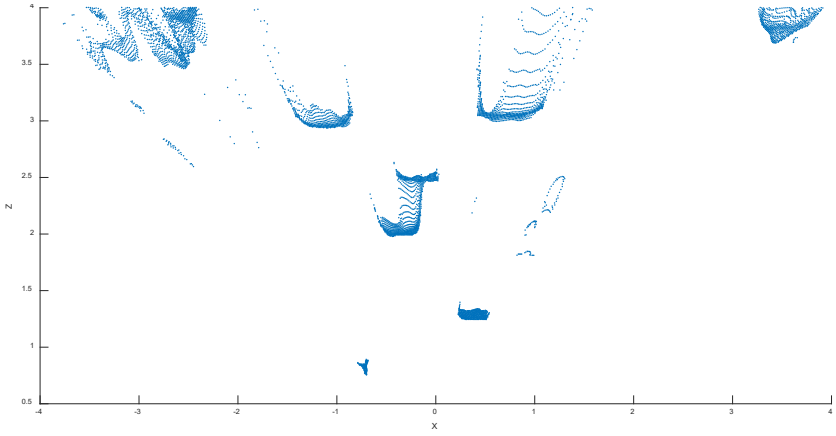


Figura 16. Vista aérea de la región de interés sin el piso.

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Después de eliminar los puntos de los *outliers* que se encuentran por debajo del área transitable, se obtiene el *foreground* de la escena original. La proyección del *foreground* en forma de mascara binaria se muestra en la figura 17. Con esta mascara se busca eliminar el *background*, es decir, aquellos puntos que pertenecen al piso y/o que están fuera de la región de interés en la imagen 2D. La eliminación del *background* de la escena se realiza con el propósito de detectar los obstáculos en la imagen. En la figura 18 se erosiona y dilata la máscara deproyectada originalmente, como se menciona en la metodología, con el propósito de eliminar el ruido generado en la deproyección.

Las figuras 19 y 20 muestran la imagen de profundidad original y el resultado de la aplicación de la máscara mostrada en la figura 18 a la imagen de profundidad. El resultado final obtenido es la eliminación del *background* de la imagen, quedando únicamente los obstáculos.

En la figura 21, se muestran los resultados de la aplicación de la metodología propuesta en este trabajo en dos escenas adicionales a la mostrada en la sección 5. Cada columna representa una escena diferente.

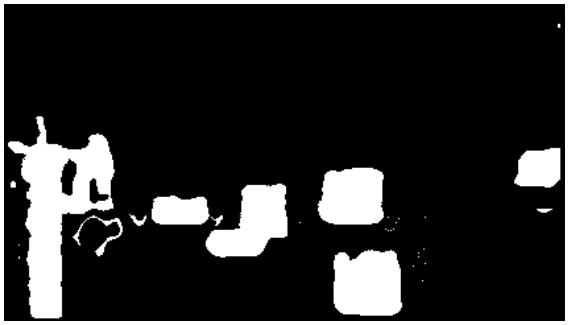


Figura 17. Proyección de la nube de puntos.

Figura 18. Dilatación y erosión de la máscara binaria.

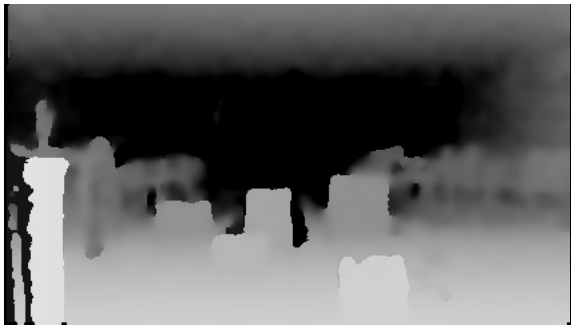


Figura 19. Imagen de profundidad obtenida por el sensor.

Figura 20. Obstáculos detectados en la imagen de profundidad.

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

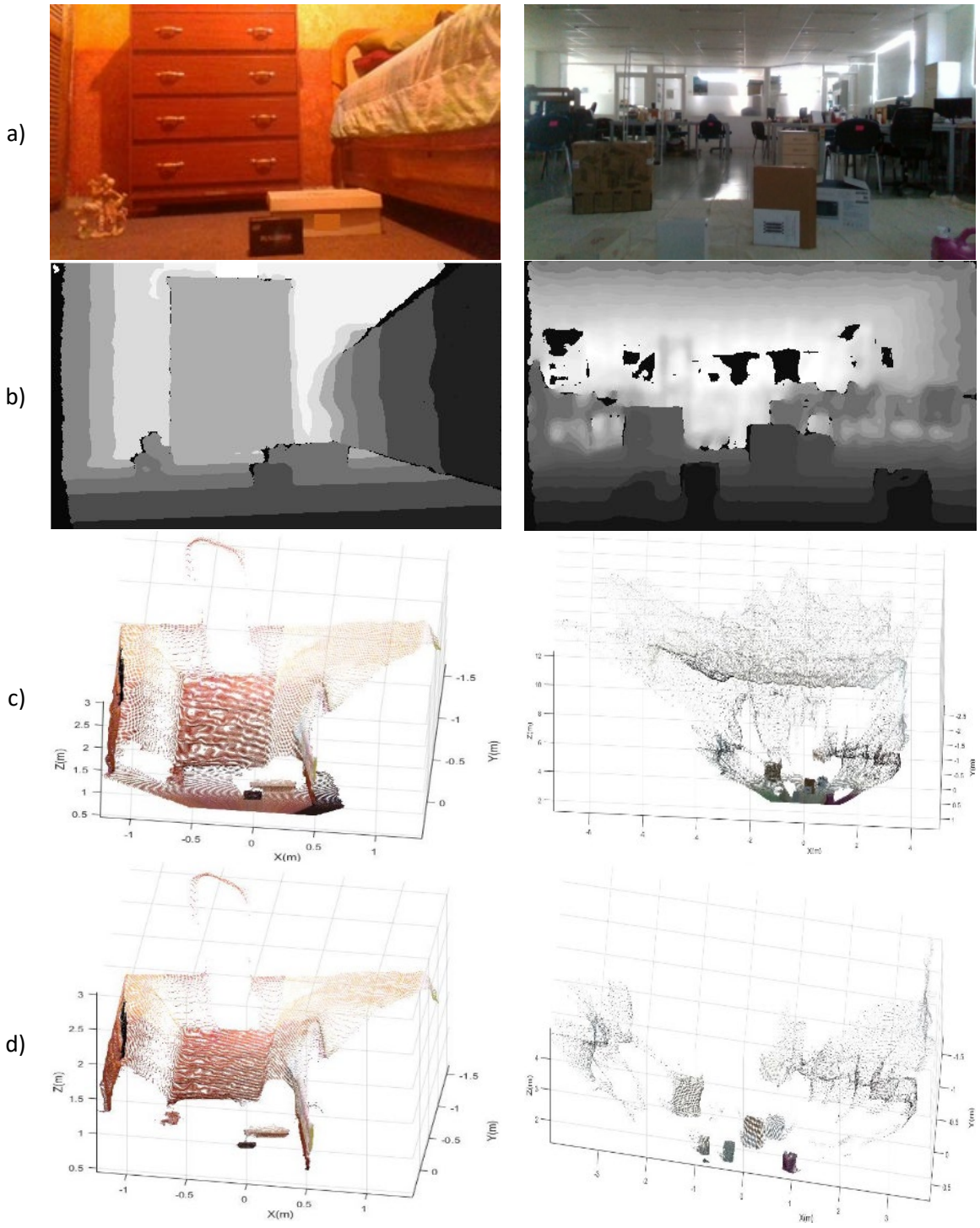


Figura 211. Resultados obtenidos de la aplicación de la metodología propuesta, en dos escenas diferentes. a) Imagen RGB. b) Imagen de profundidad. c) Nube de puntos. d) Nube de puntos de los obstáculos. e) Nube de puntos del área transitable. f) Mascara binaria procesada. g) Obstáculos detectados en la imagen de infrarrojos. h) Obstáculos detectados en la imagen de profundidad a 8 bits.

La Mecatrónica en México, Enero 2019, Vol. 8, No. 1, páginas 1 – 21
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

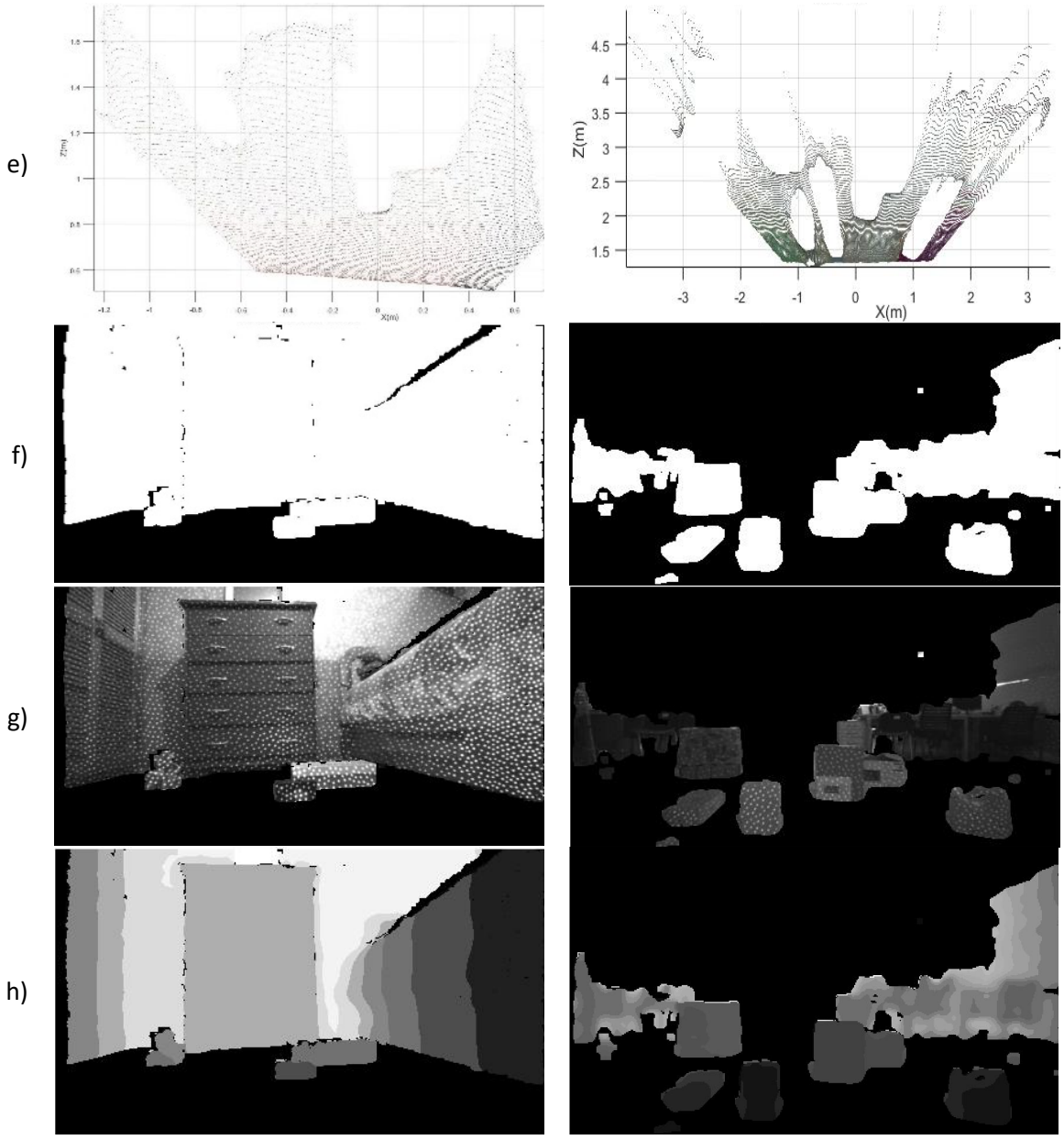


Figura 221 (Continuación). Resultados obtenidos de la aplicación de la metodología propuesta, en dos escenas diferentes. a) Imagen RGB. b) Imagen de profundidad. c) Nube de puntos. d) Nube de puntos de los obstáculos. e) Nube de puntos del área transitable. f) Mascara binaria procesada. g) Obstáculos detectados en la imagen de infrarrojos. h) Obstáculos detectados en la imagen de profundidad a 8 bits.

Los incisos b y h de la Figura 21, muestran a las imágenes de profundidad y a la imagen de profundidad resultante de la aplicación de una máscara binaria, respectivamente. Estas imágenes corresponden a la imagen de profundidad de 16 bits ecualizada. Esto con el propósito de apreciar mejor la imagen, sin embargo, todo el procesamiento se hizo utilizando la imagen de profundidad de 16 bits sin ecualizar.

6. Conclusiones

En este artículo se hace uso de la imagen de profundidad proporcionada por la cámara de profundidad Realsense D435 de Intel. El objetivo es detectar los puntos de la imagen que pertenecen a obstáculos, aquellos que pertenecen al *background*, y aquellos que pertenecen a la superficie o área transitable. Para esto se utiliza la nube de puntos de la escena correspondiente a la imagen de profundidad.

El área transitable se obtiene mediante la implementación del algoritmo RANSAC para encontrar el modelo y los puntos del plano correspondientes a la superficie del piso por el que transita un robot móvil. La dificultad de encontrar dicho modelo radica en la gran cantidad de puntos que hay en la escena que no pertenecen al modelo buscado, es decir, son *outliers*. También se selecciona una región de interés en la nube de puntos, es decir, se acota la región a analizar para la búsqueda de obstáculos.

Los puntos que pertenecen al área transitable y que están fuera de la región de interés, representan al *background* de la escena, es decir, son los objetos detectados por el sensor que no representan un obstáculo. Los puntos restantes son los que pertenecen a los obstáculos que hay en el camino del robot móvil, como se muestra en la figura 13. Para identificar los pixeles de la imagen que pertenecen a los obstáculos, se proyecta la nube de puntos. Con esto se obtiene una máscara binaria que, al multiplicarse por la imagen de profundidad original, segmenta las regiones pertenecientes a los obstáculos. Esta mascara se dilata y erosiona con el propósito de eliminar el ruido que resulta de la proyección.

El principal aporte de este artículo es la implementación de la nube de puntos para segmentar los obstáculos de la escena y representarlos, mediante una proyección, en la imagen de profundidad. La dilatación y erosión de la máscara resultante es también un aporte de este trabajo.

Como trabajo futuro se pretende realizar la detección o localización individual de cada uno de los obstáculos que se detectaron como resultado de la implementación de la metodología propuesta en el presente trabajo. Se define como obstáculo individual a todos aquellos objetos continuos en el espacio tridimensional, por lo que, en la imagen resultante de este artículo, cada uno de los obstáculos detectados deben tener valores de profundidad similares y ser continuos en el espacio.

7. Referencias

[1] P. Amaradi, N. Sriramoju, D. Li, G. S. Tewolde, and J. Kwon, “Lane following and obstacle detection techniques in autonomous driving vehicles,” in *IEEE International Conference on Electro Information Technology*, 2016.

[2] F. García, D. Martín, A. De La Escalera, and J. M. Armingol, “Sensor Fusion Methodology for Vehicle Detection,” *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 123–133, 2017.

[3] Y. H. Lee and G. Medioni, “RGB-D camera based wearable navigation system for the visually impaired,” *Comput. Vis. Image Underst.*, vol. 149, pp. 3–20, 2016.

[4] A. Aladren, G. Lopez-Nicolas, L. Puig, and J. J. Guerrero, “Navigation Assistance for the Visually Impaired Using RGB-D Sensor with Range Expansion,” *IEEE Syst. J.*, vol. 10, no. 3, pp. 922–932, 2016.

[5] H.-H. Pham, T.-L. Le, and N. Vuillerme, “Real-Time Obstacle Detection System in Indoor Environment for the Visually Impaired Using Microsoft Kinect Sensor,” *J. Sensors*, vol. 2016, pp. 1–13, 2016.

[6] V. Viswanathan and R. Hussein, “Applications of Image Processing and Real-Time embedded Systems in Autonomous Cars: A Short Review,” *Vidya Viswanathan Rania Hussein Int. J. Image Process.*, no. 112, pp. 2017–35.

[7] M. A. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Apphcatlons to Image Analysis and Automated Cartography,” vol. 24, no. 6, 1981.

[8] D. Kircali and F. B. Tek, “Ground Plane Detection Using an RGB-D Sensor,” *Inf. Sci. Syst.*, pp. 69–77, 2014.
[9] K. Qian, Z. Chen, X. Ma, and B. Zhou, “Mobile robot navigation in unknown corridors using line and dense features of point clouds,” *IECON 2015 - 41st Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 1831–1836, 2015.
[10] H.-C. Huang, C.-T. Hsieh, and C.-H. Yeh, “An Indoor Obstacle Detection System Using Depth Information and Region Growth,” *Sensors*, vol. 15, no. 10, pp. 27116–27141, 2015.
[11] S. Liu and Y. Huang, “Multi-class obstacle detection and classification using stereovision and improved active contour models,” *IET Intell. Transp. Syst.*, vol. 10, no. 3, pp. 197–205, 2016.
[12] H.-H. Pham, T.-L. Le, and N. Vuillerme, “Real-Time Obstacle Detection System in Indoor Environment for the Visually Impaired Using Microsoft Kinect Sensor,” *J. Sensors*, vol. 2016, pp. 1–13, 2016.
[13] G. Yang, F. Chen, W. Chen, M. Fang, Y. H. Liu, and L. Li, “A new algorithm for obstacle segmentation in dynamic environments using a RGB-D sensor,” *2016 IEEE Int. Conf. Real-Time Comput. Robot. RCAR 2016*, pp. 374–378, 2016.
[14] I. Van Crombrugge, L. Mertens, and R. Penne, “Fast Free Floor Detection for Range Cameras,” *Proc. 12th Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, no. February, 2017.
[15] S. Giancola, M. Valenti, and R. Sala, *A Survey on 3D Cameras : Metrological Comparison of Time-of-Flight , Structured-Light and Active Stereoscopy Technologies*. Springer, 2017.
[16] Jähne Bernd, *Digital image processing*, 6th ed. Springer, 2005.
[17] F. Lecumberry, “Cálculo de disparidad en imágenes estéreo, una comparación.”
[18] M. Zuliani, “Ransac for,” vol. 0, no. 0, pp. 5–7, 2014.
[19] P. H. S. Torr and A. Zisserman, “MLESAC: A new robust estimator with application to estimating image geometry,” *Comput. Vis. Image Underst.*, vol. 78, no. 1, pp. 138–156, 2000.