



**DISEÑO DE UN SISTEMA EN ENTORNO 3D PARA DETECCIÓN DE OBJETOS
CERCANOS CON UNA INTERFAZ GRÁFICA 2D**

GIOVANNY ALBERTO BENITES TORRES

JONNY FERNEY BERNAL HERNÁNDEZ

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

BOGOTÁ DC

2018

**DISEÑO DE UN SISTEMA EN ENTORNO 3D PARA DETECCIÓN DE OBJETOS
CERCANOS CON UNA INTERFAZ GRÁFICA 2D**

GIOVANNY ALBERTO BENITES TORRES

CÓDIGO: 20082005040

JONNY FERNEY BERNAL HERNÁNDEZ

CÓDIGO: 20082005101

**TRABAJO DE GRADO COMO REQUISITO PARA OPTAR POR EL TÍTULO DE
INGENIERO ELECTRÓNICO**

DIRECTOR:

ING. MSc. IVÁN FERNANDO GONZÁLEZ RODRÍGUEZ

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

BOGOTÁ DC

2018

DEDICATORIA

Dedico este trabajo de grado a mis padres Pedro Emigdio Bernal Delgado y Blanca Nelly Hernández Pachón, a mis hermanos Carlos Alberto Bernal Hernández y Diana Carolina Bernal Hernández por su apoyo, comprensión y amor durante toda mi vida. A mis abuelos Victor (QEPD), Concepción (QEPD) y Rosa por su guía, cariño y sabiduría. A mis primos y familiares Lorena, Julián, Alex, Andrey, Marcela, Eduardo, Stiven, Gloria, Yaneth y demás, por su cariño y compañía. A mi padrino Ramón Arias, por su guía y sabiduría. A Giovanni Benites por su apoyo, conocimientos y el trabajo conjunto. A mis amigos Carlos Patarroyo, Edwin Reyes, Lady Cardozo y demás, por todos estos años de amistad. A mi mascota Luna, gracias por tu compañía y amor incondicional.

Dedicataria de Jonny Bernal

Dedico este trabajo de grado a mi madre Aura Jacqueline Torres Cubillos, quien ha sido el apoyo emocional, económico y familiar más importante en todo mi proceso de formación académica y en toda mi vida. A mi hermano Edward Daniel Benites Torres, por su compañía, cariño y comprensión formando un futuro juntos como ingenieros. A mis abuelos Beatriz Giraldo y Excelino Benitez quienes me entregaron sabiduría, apoyo y amor, de los cuales aprendí bastantes cualidades y valores, por los que estoy eternamente agradecido. A Laura Juliana Caro Ballestas, por acompañarme en todo mi proceso académico y brindarme todo su amor, apoyo, comprensión y entrega, en la espera de un futuro mejor juntos. A Jonny Bernal por ser un gran compañero y ser un gran apoyo desde nuestro encuentro en la universidad. A mi gato Dino por acompañarme todas las noches.

Dedicataria de Giovanni Benites

AGRADECIMIENTOS

A la Universidad Distrital Francisco José de Caldas por la formación académica impartida.

Al ingeniero Iván González, director de este proyecto de grado, por su guía, orientación y todo el apoyo durante el proceso.

A los profesores que nos impartieron sus conocimientos y su experiencia para formarnos como ingenieros de calidad.

A nuestros compañeros por brindarnos su amistad, apoyo y conocimientos en esta etapa universitaria.

A nuestras familias por su amor, comprensión y por su apoyo incondicional en pro de nuestro futuro como ingenieros.

TABLA DE CONTENIDO

1- INTRODUCCIÓN.....	8
2- DESCRIPCIÓN DEL PROBLEMA.....	10
2.1- PROBLEMÁTICA.....	10
2.2- JUSTIFICACIÓN.....	11
3- OBJETIVOS.....	12
3.1- OBJETIVO GENERAL.....	12
3.2- OBJETIVOS ESPECÍFICOS.....	12
4- ALCANCES Y LIMITACIONES.....	13
4.1- ALCANCES.....	13
4.2- LIMITACIONES.....	13
5- MARCO TEÓRICO.....	14
5.1- ESTADO DEL ARTE.....	14
5.2- SENSORES DE ULTRASONIDO.....	17
5.3- SISTEMAS BASADOS EN CÁMARAS ESTÉREO PARA RECONSTRUCCIÓN 3D.....	18
5.4- SISTEMA KINECT PARA RECONSTRUCCIÓN 3D.....	28
6- DESARROLLO.....	31
6.1- SENSORES PARA DETECCIÓN DE OBJETOS CERCANOS.....	32
6.2- SISTEMA DE ROTACIÓN DE LOS SENSORES DE ULTRASONIDO.....	36
6.3- SISTEMA DE ROTACIÓN PARA ADQUISICIÓN DE IMÁGENES.....	37
6.4- SISTEMA DE DETECCIÓN A PARTIR DE SENSORES DE ULTRASONIDO.....	39
6.5- UNIDAD DE CONTROL DEL SISTEMA DE SENSADO Y TRANSFERENCIA DE DATOS.....	44
6.6- SISTEMA DE PROCESAMIENTO Y REPRESENTACIÓN GRÁFICA DE DATOS.....	51
6.7- SISTEMA DE RECONSTRUCCIÓN 3D DEL ENTORNO.....	55
6.8- INTERFAZ GRÁFICA DE USUARIO.....	57
7- RESULTADOS.....	59

7.1- MEDICIONES CON SENSORES DE ULTRASONIDO.....	59
7.2- MATRIZ DE DATOS EN RASPBERRY PI.....	65
7.3- GRÁFICA DE MATRIZ DE PUNTOS.....	67
7.4- RESULTADOS DE LA RECONSTRUCCIÓN 3D.....	69
7.4.1- REPRESENTACIÓN GRÁFICA DEL ENTORNO 3D.....	70
8- ANÁLISIS DE RESULTADOS.....	73
9- CONCLUSIONES.....	78
10- REFERENCIAS.....	80
11- ANEXOS.....	87
11.1- Anexo 1.....	87
11.2- Anexo 2.....	89
11.3- Anexo 3.....	91
11.4- Anexo 4.....	92
11.5- Anexo 5.....	93
11.6- Anexo 6.....	94

ÍNDICE DE ILUSTRACIONES

Figura 1: modelo pin hole, donde el punto representa el centro óptico de la cámara. Fuente [22].....	20
Figura 2: rotación y traslación en el espacio. Fuente [25].....	22
Figura 3: Geometría epipolar.....	24
Figura 4: Cálculo de la distancia al objeto.....	26
Figura 5: Componentes del sensor de profundidad de Kinect. Adaptada de [34].....	28
Figura 6: modelo geométrico del sensor de profundidad de Kinect. Fuente [37].....	30
Figura 7: sensor de ultrasonido HCSR04 y servomotor SG90 utilizados en el proyecto.....	33
Figura 8: estructura física y funcionamiento de un sensor de ultrasonido HS04. Fuente [39].....	33
Figura 9: Rango de medida de un sensor de ultrasonido HRC RS-04.....	35
Figura 10: servomotor utilizado en el proyecto.....	36
Figura 11: instalación del sistema de rotación con dos servomotores.....	37
Figura 12: motor paso a paso Nema 23. Fuente [41].....	38
Figura 13: EasyDriver A3967 para motor paso a paso Nema 23.....	39
Figura 14: esquema básico de conexión. Fuente [42].....	39
Figura 15: área de cobertura por 4 sensores. Las secciones oscuras son las cubiertas por dos sensores, las zonas claras son las cubiertas por un solo sensor.....	40
Figura 16: espacio mínimo de cobertura para este proyecto.....	41
Figura 17: ángulo máximo para cobertura perpendicular de 2 m con respecto al móvil y distancia máxima entre sensores.....	43
Figura 18: dispositivo de procesamiento de datos Raspberry Pi 2 modelo B (caja incluida).....	44
Figura 19: cálculo de la distancia por cada sensor.....	46
Figura 20: enfoque en los objetos más cercanos con respecto a los sensores.....	48
Figura 21: enfoque en los objetos más cercanos con respecto al punto de referencia.....	49
Figura 22: prototipo con conexiones cableadas y elementos adicionales.....	51
Figura 23: batería usada en el prototipo.....	52
Figura 24: conversores DC-DC utilizados para el proyecto.....	53
Figura 25: prototipo sin conexiones cableadas ni elementos adicionales.....	53
Figura 26: router TrendNet 651BR para conexión WiFi.....	54

Figura 27: Interfaz gráfica de usuario.....	58
Figura 28: datos entregados por el dispositivo de procesamiento Raspberry Pi.....	66
Figura 29: puntos encontrados por los sensores de ultrasonido.....	68
Figura 30: puntos encontrados por los sensores de ultrasonido, sin puntos despreciables.....	68
Figura 31: prototipo y objeto de prueba.....	70
Figura 32: representación gráfica del entorno 3D del móvil.....	71
Figura 33: histograma de las coordenadas en x.....	72
Figura 34: histograma de las coordenadas en y.....	72
Figura 35: relación para sensores de ultrasonido.....	74
Figura 36: reflexión de señal de sensores de ultrasonido.....	76
Figura 37: reflexión en mesa con un ángulo diferente.....	76
Figura 38: relación para sensor Kinect.....	77

1 INTRODUCCIÓN

Un móvil es un objeto que tiene la capacidad de trasladarse de un lugar a otro con o sin control por parte de un operador humano. Debido al avance de la tecnología se han desarrollado automóviles, robots y drones que cuentan con sistemas de navegación, percepción del entorno y prevención de accidentes [1].

Actualmente los móviles se utilizan para el ingreso, recorrido y monitoreo de lugares de difícil acceso para un ser humano; por ejemplo, cuevas, minas o la exploración interplanetaria. Para las funciones anteriormente descritas, es necesario tener sistemas que permitan al móvil hacer detección del espacio que lo rodea, encontrando objetos externos que puedan interferir en su desplazamiento o dañar la estructura del móvil si se presenta un choque con estos.

Dentro de los diferentes sistemas usados en la actualidad para la detección de obstáculos se encuentran aquellos basados en sensores de ultrasonido, los cuales son poco exactos por su grado de resolución. Estos dispositivos se enfocan en detectar la cercanía de objetos en un punto específico hasta aproximadamente 2 m y no en la totalidad del espacio tridimensional o 3D [2]; por lo que, la búsqueda de alternativas que solucionen el problema puede complementarse con un sistema que no solo a través del ultrasonido detecte los obstáculos, sino que tenga dispositivos adicionales que permitan encontrar la posición de estos en un espacio 3D.

De acuerdo a lo anteriormente expuesto, este trabajo de grado implementó un sistema que usa las propiedades de los elementos de detección de ultrasonido para encontrar objetos cercanos y adicionalmente un conjunto de cámaras o uno de sensores de producción de patrones láser (Kinect) para generar una representación del espacio 3D en un entorno gráfico 2D.

Mediante cuatro sensores de ultrasonido y un Kinect se obtuvieron diferentes matrices de posiciones de objetos cercanos, las cuales se procesaron para generar una gráfica 2D del entorno 3D, manipulable según lo requiera el usuario. Dentro de este trabajo se muestran los resultados del uso del sistema con estos dos tipos de dispositivos.

2 DESCRIPCIÓN DEL PROBLEMA

En este capítulo se realiza un análisis acerca de la problemática con respecto a la detección de objetos cercanos, donde se explican las necesidades de representar gráficamente el entorno 3D de un móvil. Adicionalmente, se presenta el porqué desarrollar este trabajo de grado enfocado en resolver la problemática expuesta a continuación.

2.1 PROBLEMÁTICA

Los móviles, autónomos o con asistencia, requieren la identificación del entorno que los rodea para que no se presenten dificultades al momento de desplazarse. En los móviles con asistencia la cantidad de objetos detectados está limitada a los que se encuentren dentro del rango visual del operador, mientras que en los autónomos, la detección de los objetos se afecta por las condiciones del ambiente como son la iluminación o el tipo de terreno, o si el móvil se encuentra en espacios confinados. Debido a estas limitaciones, se pueden dar casos de accidentes por choques u obstrucciones con objetos no detectados, teniendo como consecuencias daños en estructuras o mecanismos de los móviles e interrupción de procesos en los que estos intervengan.

2.2 JUSTIFICACIÓN

Este trabajo de grado pretende implementar un sistema que permita la detección de objetos cercanos a un móvil teniendo en cuenta las limitaciones que se pueden dar en el ambiente donde este se encuentre. Por la resolución de los sensores de ultrasonido o infrarrojos se dificulta encontrar la posición en el espacio 3D de un objeto, mientras que en sistemas de captura de imágenes, como las cámaras, las condiciones de iluminación pueden afectar el funcionamiento de estas y por lo tanto no permitir una reconstrucción adecuada del entorno; sin embargo, el uso de los dos elementos combinados para la detección de objetos del entorno permite abarcar las limitantes mencionadas y así tener un sistema que pueda utilizarse en diferentes condiciones.

3 OBJETIVOS

3.1 OBJETIVO GENERAL

- Diseñar un sistema en entorno 3D para detección de objetos cercanos con una interfaz gráfica 2D.

3.2 OBJETIVOS ESPECÍFICOS

- Desarrollar prototipos basados en sensores que permitan determinar el sistema adecuado para la detección y monitoreo de objetos que se encuentren dentro de un área para generar datos y posteriormente procesarlos.
- Generar un sistema de software que procese en forma matricial los datos adquiridos.
- Desarrollar una plataforma de software y hardware que permita generar un ambiente en 3D visualizado en 2D con los datos procesados.

4 ALCANCES Y LIMITACIONES

4.1 ALCANCES

En este proyecto se desarrolla una interfaz de usuario donde se encuentra la representación gráfica del entorno de un móvil. En la interfaz se observan las gráficas con los puntos encontrados por los sensores de ultrasonido, mostrando la distancia a estos. En una parte se encuentra la representación de la matriz obtenida con la reconstrucción 3D, generada con el procesamiento de los datos obtenidos a partir del sistema de adquisición de imágenes, mostrando la distancia de los objetos cercanos a las cámaras. Adicionalmente, se implementa un prototipo en forma de carro o robot en donde se sitúan los sensores y, las cámaras o el sensor Kinect con sus respectivos motores.

4.2 LIMITACIONES

El rango de detección de objetos se encuentra limitado por las características y capacidades de los sensores a utilizar y los algoritmos realizados para la detección con imágenes de las cámaras de video; también, la velocidad y el ángulo de rotación de los motores limita la duración de la captura de datos por lo tanto, no se puede realizar la reconstrucción del entorno en tiempo real. Con las cámaras de video se hace necesario el uso de elementos adicionales para la detección de objetos en espacios con poca o nula visibilidad.

5 MARCO TEÓRICO

En este capítulo se presentan los conceptos básicos de los dispositivos y métodos utilizados en el transcurso de este trabajo de grado. Se realiza una breve descripción del estado del arte de los sistemas de percepción usados en móviles, los sensores de ultrasonido, el arreglo estéreo de cámaras, motores paso a paso, servomotores y el sensor Kinect.

5.1 ESTADO DEL ARTE

En la práctica se busca que los sistemas de percepción del entorno para móviles sean fáciles de implementar, eficientes energéticamente, y con un rango de detección amplio y preciso. Se usan uno o varios tipos de sensores dependiendo de la aplicación y los objetos a detectar, por ejemplo, si es solo la proximidad a un obstáculo o la detección de la totalidad del entorno 3D que rodea al móvil.

Para la detección de objetos cercanos y la medición de distancia a estos se tienen tres enfoques básicos [3], los cuales se basan en los siguientes tipos de sensores:

- Sensores basados en la medida del tiempo de vuelo (en inglés *TOF*), donde un pulso de energía es emitido hacia un objeto reflectante para luego medir el retardo de su eco. Entre

los sensores de este tipo se encuentran los de ultrasonido, infrarrojos y de emisión de pulsos láser.

- Sensores basados en cambio de fase, en los cuales se envía al objeto una señal continua de láser, ultrasonido o infrarrojos, para comparar el cambio de fase de su reflejo con una señal de referencia. Se necesitan varias señales enviadas al objeto para realizar una medición de distancia adecuada.
- Sensores basados en radar de frecuencia modulada (*FM*), con los que se envía una señal triangular, electromagnética, modulada en frecuencia y, al igual que en los sensores de cambio de fase, se compara el corrimiento de fase de su reflejo con una señal de referencia. Estos sensores tienen la ventaja de solo necesitar un pulso enviado para realizar una medición.

Con respecto a la detección del entorno 3D de un móvil, se usan los siguientes tipos de sistemas [4] :

- Sistemas de visión estéreo, principalmente un arreglo de cámaras de características similares.
- *LIDAR* (*light detection and imaging ranging* en inglés), donde se obtiene una nube de puntos con las coordenadas 3D del entorno por medio del envío de pulsos láser.

- Sistemas RGB-D, que están basados en la combinación de cámaras infrarrojas y cámaras a color o RGB; se basan en la emisión de haz láser en forma de patrón geométrico y a través de un elemento sensor, por ejemplo una cámara infrarroja, se capta la deformación presentada por el patrón en la superficie del objeto para obtener un mapa de disparidad. Un ejemplo de este tipo de sistemas es el sensor Kinect de Microsoft.

En los últimos desarrollos de sistemas para la percepción del entorno de móviles se aplica el concepto de fusión de sensores, que es la integración de los datos obtenidos a partir de varios tipos de sensores [5]. Se ha demostrado que los sistemas basados en este enfoque presentan menor error en las detecciones a diferencia de los que se basan en un solo tipo de sensor [6], mejorando su rendimiento en distintos ambientes y extendiendo su cobertura espacial y temporal [7].

La fusión de sensores se utiliza en los drones para generar mapeos de superficies [8] y detección de obstáculos, [9] de carreteras, otros vehículos y peatones en vehículos autónomos [10]; determinación de posición y orientación de robots dentro de edificios [11].

5.2 SENSORES DE ULTRASONIDO

Los sensores de ultrasonido aprovechan las propiedades de las señales acústicas con frecuencias mayores a 20 KHz [12], para realizar la medición de parámetros físicos. El principio de funcionamiento de estos sensores se basa en la propagación de ondas mecánicas, específicamente sonido, y la oposición que presenta una superficie al movimiento inducido por la presión que estas ejercen, lo que se define como impedancia acústica Z [13]. Cuando una onda mecánica pasa de un medio con una impedancia a otro con un valor de impedancia diferente, dicha onda no solo se absorbe sino que se refleja [14]; la ecuación (1) muestra que la reflexión de la onda R es mayor cuando la diferencia entre las impedancias de los medios, Z_1 y Z_2 , es grande.

$$R = \left(\frac{Z_1 - Z_2}{Z_1 + Z_2} \right)^2 \quad (1)$$

Los tipos de sensores de ultrasonido más usados se basan en los siguientes principios [15] :

- Reflexión de pulso o medición de eco, en los cuales se usa el principio de medición de tiempo de vuelo o *TOF*, visto en la sección 5.1 . Los sensores basados en este principio se usan como detectores de proximidad.
- Efecto Doppler, donde se aprovecha esta propiedad de las ondas para medir fluidos y partículas en movimiento.

- Correlación cruzada y tiempo de tránsito, para lo que se utilizan dos sensores y se miden las diferencias de las detecciones hechas por estos; los sistemas basados en estos principios también son usados para la medición de fluidos y partículas en movimiento.

Hay varios diseños de sensores de ultrasonido para la detección de proximidad de objetos, entre los cuales se pueden encontrar en la misma unidad el emisor y el receptor [16]. Para la generación y detección de ultrasonido se utilizan materiales piezoeléctricos, en los que se generan tensiones eléctricas al momento de ser aplicada una presión mecánica [17].

5.3 SISTEMAS BASADOS EN CÁMARAS ESTÉREO PARA RECONSTRUCCIÓN 3D

La reconstrucción de un entorno 3D se puede realizar por medio de un arreglo estéreo, el cual se compone de dos cámaras similares separadas por una pequeña distancia [18]. En la reconstrucción 3D se usa el principio de la visión estéreo o binocular, el cual se basa en la visión humana [19]. Actualmente se utilizan cámaras digitales para los sistemas de visión estéreo, las cuales son dispositivos sensibles a la luz visible que permiten almacenar imágenes en un formato digital [20].

El modelo de cámara en el que se basan librerías como OpenCV para realizar reconstrucción 3D es el modelo pin hole [21], donde se relaciona un punto en el espacio 3D con su representación en el plano 2D de la imagen capturada según las ecuaciones (2) y (3). En la figura 1 se pueden observar las proyecciones de un punto según el modelo, donde la óptica se reduce a los puntos que pasan directamente por la distancia focal f (distancia entre la cámara y el punto 3D); una dimensión se pierde ya que todos los puntos cuya relación entre las coordenadas X, Z y Y, Z sea constante dan la misma coordenada en la imagen [22].

$$x = \frac{f}{Z} X \quad (2)$$

$$y = \frac{f}{Z} Y \quad (3)$$

Donde:

- x, y : coordenadas del punto en el espacio 2D de imagen
- X, Y, Z : coordenadas del punto en el espacio 3D
- f : distancia focal de la cámara

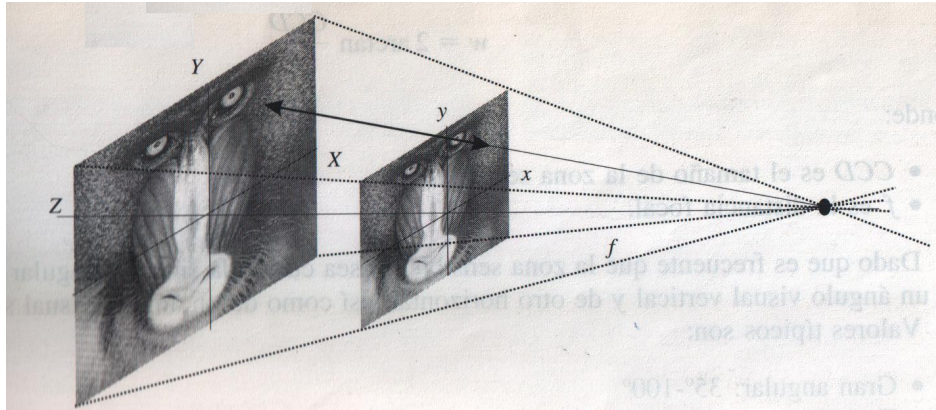


Figura 1: modelo pin hole, donde el punto representa el centro óptico de la cámara. Fuente [22]

La ecuación (4) modela la proyección de un punto 3D Q a un punto 2D q , a partir de las características de cada cámara, usando coordenadas homogéneas con $w = z$.

$$q = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, M = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Donde:

- M : matriz de características intrínsecas de la cámara
- f_x, f_y : distancias focales en los ejes x y y respectivamente

- c_x, c_y : desplazamientos desde el origen en los respectivos ejes
- s : factor de torcimiento, en muchos casos se puede asumir igual a cero [23]

Al ser las imágenes capturadas por una cámara, proyecciones de objetos 3D a un plano imagen 2D se tiene una homografía o transformación de un plano a otro, lo que permite hallar la matriz de rotación R (5) y el vector de traslación t (6); parámetros que dan información acerca de la posición de cada cámara con respecto al origen del sistema de coordenadas. La combinación de R y t genera la matriz extrínseca de la cámara [24].

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (5)$$

$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (6)$$

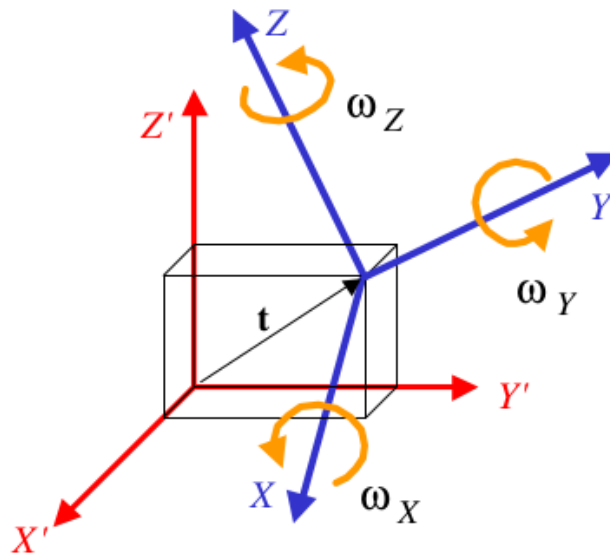


Figura 2: rotación y traslación en el espacio. Fuente [25]

Los factores de la matriz R y el vector t representan el modelo de un sistema que ha tenido una rotación y una traslación en el espacio [25], como se observa en la figura 2. Los elementos del vector t son las coordenadas de este en el espacio. La matriz R se obtiene del producto entre las matrices de rotación de los ejes x, y, z ; las cuales son matrices de transformación de $R^3 \rightarrow R^3$, cada una se muestra en (7), (8) y (9).

$$R_z = \begin{bmatrix} \cos(\omega_z) & \text{sen}(\omega_z) & 0 \\ -\text{sen}(\omega_z) & \cos(\omega_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$R_Y = \begin{bmatrix} \cos(\omega_Y) & 0 & -\text{sen}(\omega_Y) \\ 0 & 1 & 0 \\ \text{sen}(\omega_Y) & 0 & \cos(\omega_Y) \end{bmatrix} \quad (8)$$

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega_X) & \text{sen}(\omega_X) \\ 0 & -\text{sen}(\omega_X) & \cos(\omega_X) \end{bmatrix} \quad (9)$$

Con lo que cada elemento de R es:

$$\begin{aligned} r_1 &= \cos(\omega_Y) \cos(\omega_Z) \\ r_2 &= \cos(\omega_Y) \text{sen}(\omega_Z) \\ r_3 &= -\text{sen}(\omega_Y) \\ r_4 &= \text{sen}(\omega_X) \text{sen}(\omega_Y) \cos(\omega_Z) - \cos(\omega_X) \text{sen}(\omega_Z) \\ r_5 &= \text{sen}(\omega_X) \text{sen}(\omega_Y) \text{sen}(\omega_Z) + \cos(\omega_X) \cos(\omega_Z) \\ r_6 &= \text{sen}(\omega_X) \cos(\omega_Y) \\ r_7 &= \cos(\omega_X) \text{sen}(\omega_Y) \cos(\omega_Z) - \cos(\omega_X) \text{sen}(\omega_Z) \\ r_8 &= \cos(\omega_X) \text{sen}(\omega_Y) \text{sen}(\omega_Z) - \text{sen}(\omega_X) \cos(\omega_Z) \\ r_9 &= \cos(\omega_X) \cos(\omega_Y) \end{aligned} \quad (10)$$

Se necesita realizar calibración en cada cámara para calcular M , los factores de distorsión debidos a defectos en los lentes [26] y la matriz extrínseca. En la técnica [27] se usa un patrón conocido, típicamente un tablero de ajedrez, al cual se toman varias imágenes con la cámara a calibrar para encontrar la relación entre las medidas del patrón y las de sus proyecciones en cada imagen capturada; así, poder realizar la calibración.

Para realizar reconstrucción 3D se toman varias imágenes de un mismo objeto por medio del arreglo de cámaras y se realiza el análisis de cada punto en una de las imágenes buscando puntos correspondientes en las otras. Con el conjunto de correspondencias de los puntos se haya un mapa de disparidad, el cual muestra las diferencias en coordenadas. Conociendo las características de posición de ambas cámaras se obtienen distancias, a partir del mapa, usando triangulación.

Uno de los métodos que se utiliza para encontrar correspondencias de puntos es la geometría epipolar, la cual permite reducir el esfuerzo computacional al momento de buscar correspondencias de puntos entre imágenes [28]. En la Figura 3 se tiene una representación gráfica de esta geometría.

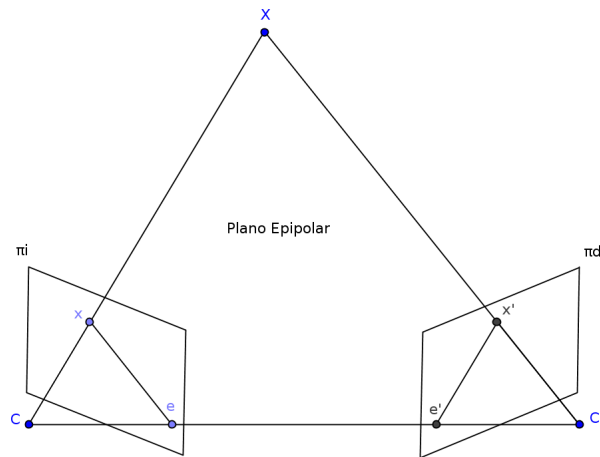


Figura 3: Geometría epipolar.

Para un punto X en el espacio 3D sus proyecciones en los planos imagen de las cámaras izquierda y derecha, π_l y π_d , son los puntos 2D x y x' respectivamente. Desde los centros ópticos de cada cámara, C y C' , al punto X se tienen un par de rayos, la intersección de estos y la línea base, que es el segmento que une los centros de cada cámara, forma el plano epipolar. El centro óptico de una cámara es proyectado como un punto en la imagen que forma la otra; este punto es el epipolo, e y e' para las cámaras derecha e izquierda respectivamente. Suponiendo que solo se conoce al punto x de π_l , el punto correspondiente x' de la otra imagen únicamente debe pertenecer a la línea epipolar, que es el segmento $(e' - x')$ generado por la intersección del plano epipolar con el plano π_d ; esta condición se conoce como restricción epipolar y es lo que permite reducir el esfuerzo computacional ya que la búsqueda de correspondencias pasa de 2D a 1D. La transformación de un punto 2D a la línea epipolar correspondiente es representada por la matriz fundamental F [29] Esta metodología tiene las desventajas de no obtener información de zonas con bajas texturas y/o poca iluminación [30].

Teniendo en cuenta las características de las cámaras, su posición en el espacio y asumiendo que los ejes ópticos son paralelos, se calcula la distancia del arreglo de las cámaras al punto en el espacio 3D, para esto se usa el método de triángulos semejantes [31]. En la figura 4 se observa que la distancia de los centros ópticos a los puntos x y x' es la distancia focal de cada cámara, f y f' ; adicionalmente, se observa que las cámaras se encuentran separadas por una distancia que es representada por la línea base.

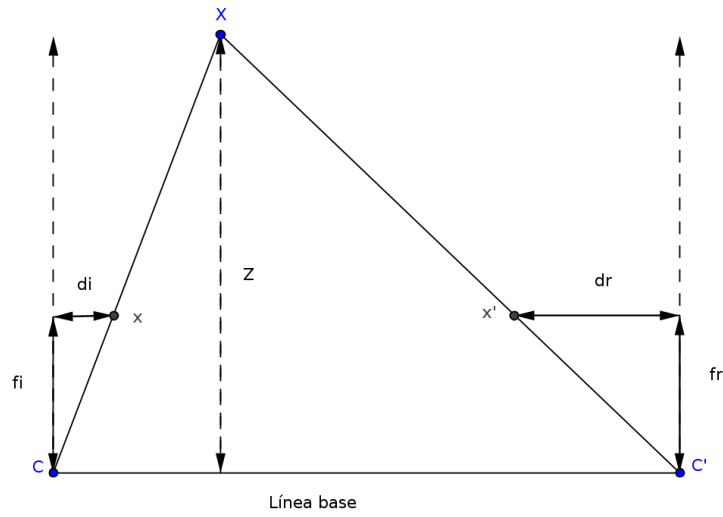


Figura 4: Cálculo de la distancia al objeto.

Como la disparidad es la diferencia de las coordenadas de los puntos, se tiene la ecuación (11) :

$$D = d_l - d_r \quad (11)$$

Donde:

- D : disparidad
- d_l : coordenada horizontal del punto x
- d_r : coordenada horizontal del punto x'

Asumiendo que las cámaras son iguales se encuentra que sus distancias focales también lo son, teniendo en cuenta esta consideración se encuentra la relación (12) al aplicar triángulos semejantes; b representa la línea base.

$$\frac{b-D}{Z-f} = \frac{b}{Z} \quad (12)$$

Despejando Z en (12) se haya la profundidad de un punto en el espacio 3D, ecuación (13).

$$Z = \frac{f \cdot b}{D} \quad (13)$$

Las coordenadas X y Y del punto en el espacio 3D se pueden calcular despejándolas en las ecuaciones (2) y (3) respectivamente, teniendo en cuenta que se presenta un corrimiento en las coordenadas de los píxeles por los factores c_x y c_y , como se observa en (14) y (15).

$$X = \frac{(x - c_x) \cdot Z}{f} \quad (14)$$

$$Y = \frac{(y - c_y) \cdot Z}{f} \quad (15)$$

5.4 SISTEMA KINECT PARA RECONSTRUCCIÓN 3D

El sistema Kinect de Microsoft se presentó como un dispositivo para controlar videojuegos por medio de la captura de los movimientos de una persona [32] , sin embargo se han desarrollado aplicaciones con este sistema en los ámbitos de la robótica, la biomédica y la visión por computador [33]entre otros.

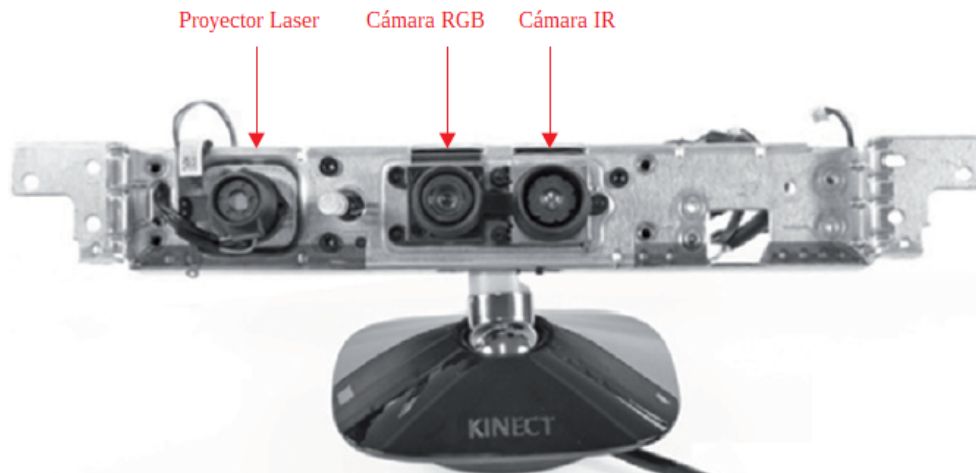


Figura 5: Componentes del sensor de profundidad de Kinect. Adaptada de [34]

El Kinect se compone de un sensor de profundidad, una cámara RGB y un arreglo de micrófonos. El sensor de profundidad consiste en un proyector de láser no visible junto con una cámara de infrarrojos (IR), la cual está fabricada con un sensor CMOS [34] el rango de detección de profundidad está comprendido entre 0.8 m a 3.5 m y el punto de vista angular es de 57° horizontalmente y 43° verticalmente [35]. En la figura 5 se pueden observar los elementos descritos anteriormente.

La tecnología usada en Kinect fue desarrollada por la compañía israelí Primesense [36] y se basa en la emisión de un haz de láser con varios patrones geométricos que son proyectados sobre una superficie; el haz se genera por un emisor infrarrojo y pasa a través de un elemento difusor, que lo expande en forma de un patrón geométrico de partículas. Un sensor de imagen, formado por un arreglo CMOS, detecta la deformación del patrón en la superficie para luego compararla con imágenes de referencia almacenadas en el sistema de procesamiento del Kinect y así, obtener la profundidad representada por un mapa de disparidad.

Para la reconstrucción 3D se aplica el modelo geométrico [37], donde se usan las mismas consideraciones del modelo pin hole con la adición de la ecuación de proyección del sensor de profundidad (16); la representación gráfica del modelo se observa en la figura 6.

$$\begin{bmatrix} X \\ Y \\ d \end{bmatrix} = \begin{bmatrix} u - u_0 \\ v - v_0 \\ \frac{1}{c_1} \frac{1}{Z} - \frac{c_0}{c_1} \end{bmatrix} \quad (16)$$

Donde:

- d : disparidad dada por el Kinect
- u, v : coordenadas de puntos en la cámara IR
- u_0, v_0 : desplazamientos desde el origen en los ejes respectivos
- c_0, c_1 : parámetros internos del sensor de profundidad

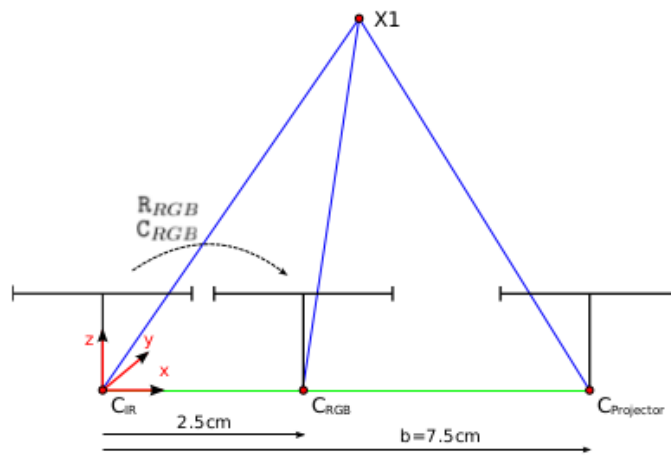


Figura 6: modelo geométrico del sensor de profundidad de Kinect.
Fuente [37]

Existen varias ventajas en el uso del Kinect frente a las cámaras estéreo, entre las cuales se destaca el realizar reconstrucción 3D de objetos sin textura [38], lo cual no puede ser llevado a cabo por cámaras estéreo.

6 DESARROLLO

Para este trabajo de grado se usaron dos sistemas: el primero para la captura de datos y el segundo para el procesamiento y visualización de los mismos. El primer sistema consta de un grupo de dispositivos que permiten tener información del entorno en 360° horizontalmente para lograr encontrar los puntos más cercanos al móvil, brindando la capacidad de generar un mapa de puntos de distancia en 2D .

El segundo sistema se encarga de procesar toda la información recibida y generar la representación en 3D mediante una interfaz 2D del entorno, que consta de un software implementado en el lenguaje Python y una pantalla que permite la visualización de la representación.

Los dispositivos que componen la totalidad de los sistemas son:

- Cuatro sensores de Ultrasonido.
- Cuatro servomotores.
- Motores Paso a Paso.
- Kinect.

- Dispositivo de procesamiento de datos Raspberry Pi.
- Computador personal o PC para procesamiento de datos y visualización de gráficas.

6.1 SENSORES PARA DETECCIÓN DE OBJETOS CERCANOS

En el mercado existen diferentes tipos de sensores que se usan para detectar objetos cercanos como lo son los inductivos, capacitivos y de efecto Hall; con la limitación que solo detectan objetos a distancias cortas pues se basan en el uso de campos magnéticos y eléctricos para percibir cambios de corriente, incluso requiriendo de contacto físico por lo que son ampliamente utilizados como detectores ON/OFF.

En este sentido, los sensores de ultra sonido se han escogido sobre otros tipos de sensores debido a las limitaciones presentadas anteriormente, además de su bajo costo y fácil uso. La figura 7 muestra uno de los sensores de ultrasonido usados en el proyecto sobre el servomotor que le permite el movimiento de rotación.



Figura 7: sensor de ultrasonido HCSR04 y servomotor SG90 utilizados en el proyecto

Los sensores de ultrasonido están compuestos por un arreglo emisor-receptor como se muestra en la figura 8. El emisor genera una serie de pulsos u ondas sonoras que rebotan en cualquier objeto encontrado a su paso, regresando nuevamente al sensor. El receptor recibe estas ondas sonoras y las transforma en una señal eléctrica.

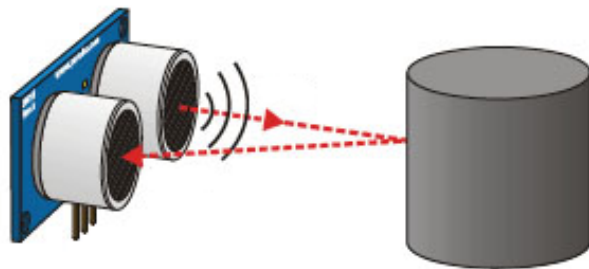


Figura 8: estructura física y funcionamiento de un sensor de ultrasonido HS04. Fuente [39]

Según la hoja de datos [39], los sensores cuentan con las siguientes características:

- Frecuencia de la señal de ultrasonido: 40 KHz
- Rango de Detección de 2 cm hasta 400 cm, con precisión 3 mm
- Rango de medición: 15°

Es posible calcular el rango de cobertura del sensor de ultrasonido para un objeto que se encuentre al máximo de detección, utilizando la ley de senos y cosenos con estructuras que tienen la forma de triángulos rectángulos. Con la ecuación (17) se calcula el rango de cobertura:

$$rg=2 h \operatorname{sen}(\alpha) \quad (17)$$

Dónde:

- rg :rango de cobertura.
- h : hipotenusa.
- α : ángulo de cobertura con respecto al eje central.

Usando el valor máximo de distancia que es de 400 cm y el ángulo de 15° se tiene (18) :

$$rg = 2 * 400 \text{ cm} * \text{sen}(15^\circ) = 207,0552 \text{ cm} \quad (18)$$

Como hay 15° de cobertura con respecto al eje central, se tienen 30° de extremo a extremo; así, el rango de cobertura total es de 207 cm cuando la distancia del objeto al sensor es de 400 cm. En la práctica se presenta un inconveniente como se puede observar en la figura 9, donde hay un objeto en tres posiciones distintas dentro del rango de los 2 m a la misma distancia con respecto al sensor. Por lo anterior, no se puede establecer en qué posición se encuentra el objeto, entonces se requiere de un sistema o dispositivo adicional que mejore la precisión de la medición.

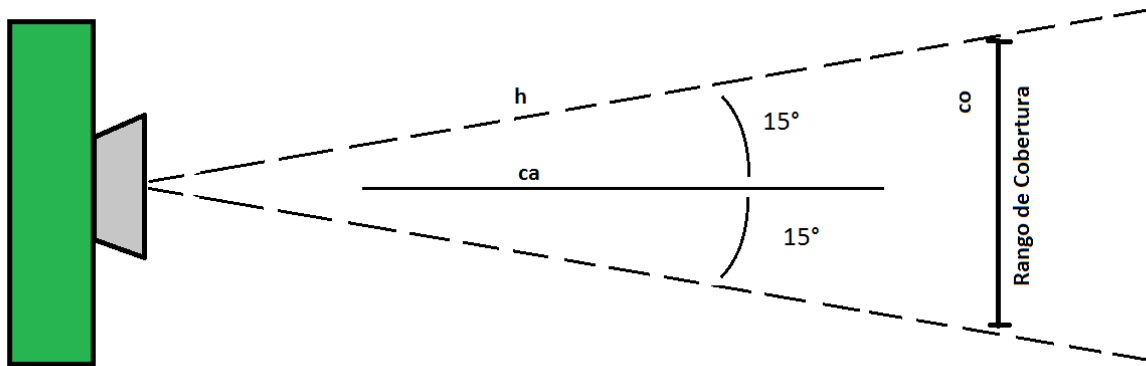


Figura 9: Rango de medida de un sensor de ultrasonido HRC RS-04

A pesar de las limitaciones de los sensores de ultrasonido, estos presentan unas consideraciones apropiadas para este proyecto, ya que como se mostró anteriormente, el rango de detección está entre 2 cm hasta 400 cm; adicionalmente, el voltaje de alimentación es de 5 V por lo que no es necesaria una interfaz para conectarlos al Raspberry Pi.

6.2 SISTEMA DE ROTACIÓN DE LOS SENSORES DE ULTRASONIDO

Los sensores de ultrasonido requieren de desplazamiento para poder dirigir el rango de detección hacia un punto específico, para esto se usan como mecanismo de giro los servomotores como se pueden observar en la figura 10.

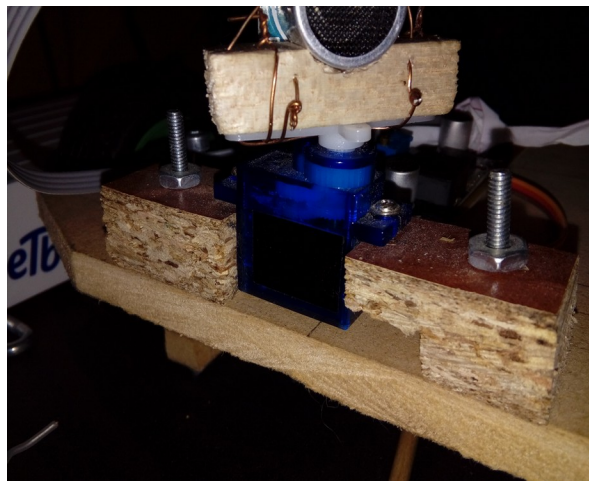


Figura 10: servomotor utilizado en el proyecto

Para este trabajo se utilizaron cuatro servomotores SG90, con las siguientes características [40]:

- Peso: 9 gr
- Dimensiones: 22.2 x 11.8 x 31 mm approx.
- Torque: 1.8 kg x cm
- Velocidad de operación: 0.1 s/60°

- Voltaje de operación: 4.8 V - 5 V

6.3 SISTEMA DE ROTACIÓN PARA ADQUISICIÓN DE IMÁGENES

Se requiere usar un motor paso a paso porque los dispositivos de captura de imágenes, como el Kinect o un arreglo de cámaras, deben rotar 360°. Primero se realizó una prueba uniendo dos servomotores, pero se encontró que esta configuración presenta inestabilidad al momento de girar debido al peso de los dispositivos de captura de imágenes, en la figura 11 se puede observar dicha configuración.

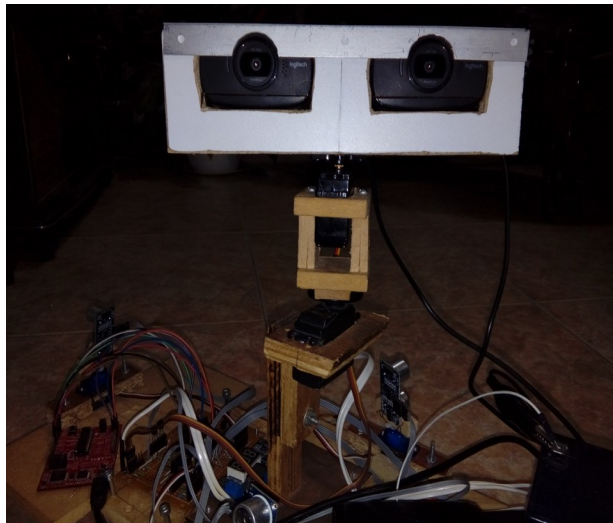


Figura 11: instalación del sistema de rotación con dos servomotores

El motor paso a paso escogido es el *Stepper Motor Nema 23* [41], que se puede observar en la figura 12. Las características del motor paso a paso se describen a continuación:

- Ángulo por Paso: $1,8^\circ$
- Longitud del Motor: 48 mm
- Corriente Nominal: 1,7 A

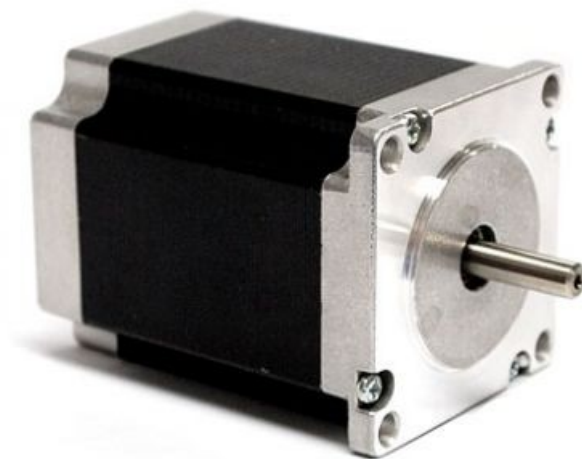


Figura 12: motor paso a paso Nema 23. Fuente [41]

En el proyecto se utiliza el driver *EasyDriver V44 A3967* [42], como se observa en la figura 13 y su esquema de conexiones con el motor paso a paso, que se observa en la figura 14.



Figura 13: EasyDriver A3967 para motor paso a paso Nema 23

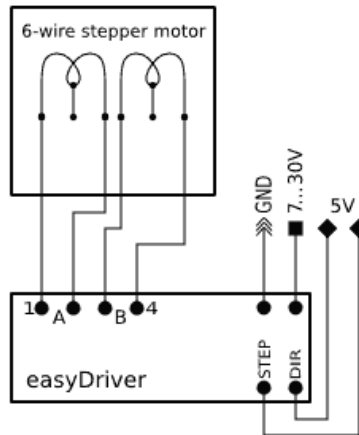


Figura 14: esquema básico de conexión. Fuente [42]

6.4 SISTEMA DE DETECCIÓN A PARTIR DE SENSORES DE ULTRASONIDO

El sistema de detección está compuesto por cada sensor de ultrasonido junto con su respectivo sistema de rotación, que se muestra en la sección 6.2. En el prototipo se usa un sensor de ultrasonido por cada lado, de manera que cada sensor pueda cubrir 180°, teniendo en total una

cobertura de detección de 360° a su alrededor; solo se requiere de un sensor por cada lado debido al tamaño del prototipo (aproximadamente 50 cm de largo, 30 cm de ancho y 20 cm de alto) y al rango de cobertura de los sensores de ultrasonido. La distancia de detección de los sensores es de máximo 400 cm, por lo que se puede cubrir más espacio que el requerido por el móvil, como se puede observar en la Figura 15. De tenerse un prototipo de mayores dimensiones se podría implementar un número de sensores a partir de las consideraciones tratadas a continuación.

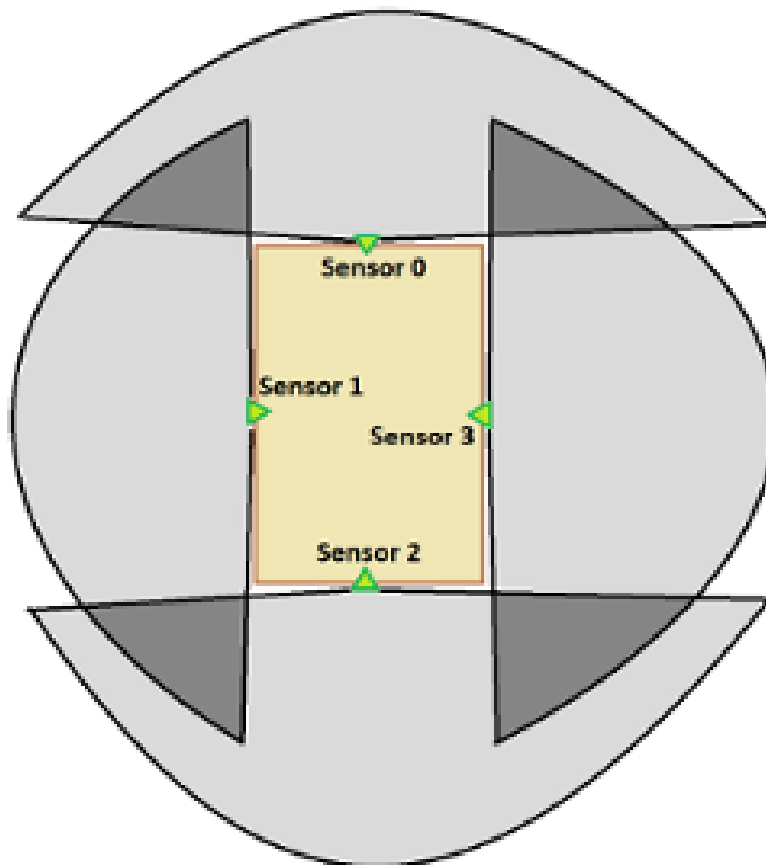


Figura 15: área de cobertura por 4 sensores. Las secciones oscuras son las cubiertas por dos sensores, las zonas claras son las cubiertas por un solo sensor

Los sensores pueden detectar objetos hasta 4 m de distancia como ya se mencionó. Se decide especificar una distancia de medición hasta 2 m en todos los lados del móvil, como se muestra en la Figura 16, en donde se puede observar que perpendicularmente se tiene un espacio de 2 m alrededor, generando cobertura en todo el área gris.

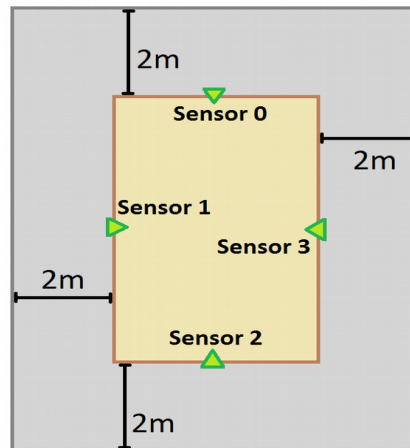


Figura 16: espacio mínimo de cobertura para este proyecto

Es así, que se puede encontrar un ángulo mínimo con el cual el sensor en su distancia máxima de detección, logre realizar el sensado perpendicular de 2 m, como se observa en la Figura 17. Para hallar el ángulo mínimo que un sensor de ultrasonido pueda tener para dar la cobertura requerida se requiere del uso de la ecuación (19).

$$\theta = \text{sen}^{-1}\left(\frac{ca}{h}\right) \quad (19)$$

Dónde:

- ca : distancia de cobertura perpendicular con respecto al móvil
- h : distancia máxima de detección del sensor de ultrasonidos
- θ : ángulo de giro del sensor de ultrasonido

Teniendo como distancia máxima de detección 400 cm y una distancia de cobertura perpendicular de 200 cm, se encuentra el ángulo mínimo del sensor para obtener la cobertura requerida, mostrando el resultado en la ecuación (20):

$$\theta = \text{sen}^{-1}\left(\frac{200 \text{ cm}}{400 \text{ cm}}\right) = 30^\circ \quad (20)$$

Luego de encontrar el ángulo θ , se utiliza la ecuación (21) para hallar la distancia máxima entre sensores sin afectar la medición dentro del espacio de cobertura:

$$ca = \left(\frac{2 \text{ } co}{\tan(\theta)}\right) \quad (21)$$

Donde:

- ca : distancia de cobertura perpendicularmente con respecto al móvil
- co : distancia de un lado del sensor a lo largo del móvil

- θ : ángulo de giro del sensor de ultrasonido

Los valores dados permiten obtener el resultado de la ecuación (22):

$$ca = \left(\frac{2 * 200 \text{ cm}}{\tan(30^\circ)} \right) = 692,82 \text{ cm} \quad (22)$$

La distancia máxima entre sensores permitida sin afectar el rango de cobertura es de 692,82 cm, es decir, 2 veces co según la ecuación (22); tal como se observa en la figura 17.

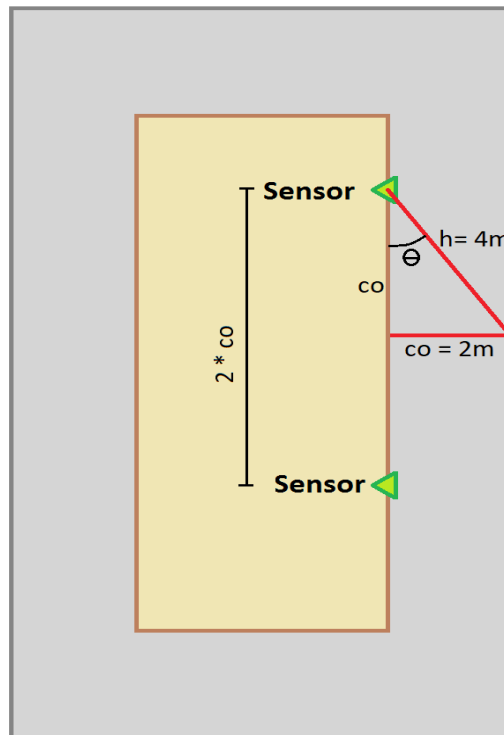


Figura 17: ángulo máximo para cobertura perpendicular de 2 m con respecto al móvil y distancia máxima entre sensores

6.5 UNIDAD DE CONTROL DEL SISTEMA DE SENSADO Y TRANSFERENCIA DE DATOS

La Unidad de Procesamiento Raspberry Pi 2 Modelo B, mostrada en la Figura 18, se encarga de controlar los servomotores, sensores de ultrasonido, motor paso a paso y Kinect; recibir los datos generados por estos sensores, crear matrices con estos datos y transmitirlos hacia el sistema de procesamiento. Los motores y los sensores de ultrasonido se encuentran conectados al Raspberry Pi por medio del puerto GPIO mientras que el Kinect es conectado a uno de los puertos USB.



Figura 18: dispositivo de procesamiento de datos Raspberry Pi 2 modelo B (caja incluida)

El programa implementado en Raspberry Pi comienza con el control del sistema de detección tratado en la sección 6.4, los algoritmos de este programa, implementados en Python, se pueden observar en los diagramas de flujo 1, 2 y 3. Cuando se obtienen los datos de los sensores de ultrasonido se procesan adecuadamente con un patrón de la forma:

“*Sensor,sens,ang,dis*”

Esta secuencia tiene como objetivo transmitir todos los datos necesarios en una sola variable de texto, para poder ser leídos más adelante por el Raspberry Pi; posteriormente, se divide esta variable de texto para encontrar cada uno de los valores que se encuentran separados por un carácter, que para este caso es el símbolo “,” (coma).

El valor contenido en el campo *sensor* indica simplemente texto y se utiliza como condición para el inicio de una variable de texto, el valor contenido en *sens* es numérico e indica el número del sensor que se está utilizando. Dentro de *ang* se tiene una variable numérica que contiene el ángulo en el que se encuentra el sensor; finalmente se tiene *dis*, que es una variable numérica que representa la distancia entre el sensor y el objeto u obstáculo.

Los valores obtenidos permiten realizar el cálculo de la distancia del objeto al sensor en sus componentes x y y , como se muestra en la Figura 19. Se encuentra la posición con respecto al eje x utilizando la ecuación (23) y la posición con respecto al eje y utilizando la ecuación (24):

$$x = h \cos(\alpha) \tag{23}$$

$$y = h \sin(\alpha) \tag{24}$$

Donde:

h : distancia obtenida por el sensor de ultrasonido

- α : ángulo del sensor de ultrasonido
- x : distancia obtenida con respecto al eje x
- y : distancia obtenida con respecto al eje y

Es importante tener en cuenta que para una mayor precisión en la ubicación del objeto se toma el centro del prototipo como punto de referencia, lo que implica que se requiera de los valores de las dimensiones del móvil para poder hallar la distancia y el ángulo real de este al objeto. En la Figura 19, se observa lo expresado anteriormente:

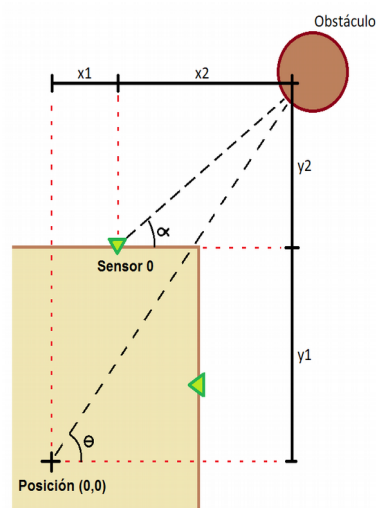


Figura 19: cálculo de la distancia por cada sensor

Donde:

- x_1 : distancia con respecto al eje x entre el centro y el sensor
- x_2 : distancia con respecto al eje x entre el sensor y el objeto
- y_1 : distancia con respecto al eje y entre el centro y el sensor
- y_2 : distancia con respecto al eje y entre el sensor y el objeto
- α : ángulo formado entre sensor y objeto
- θ : ángulo formado entre centro y objeto

Así, la distancia real x_r con respecto al eje x se consigue sumando x_1 y x_2 como se presenta en la ecuación (25), al igual que para la distancia real y_r , ecuación (26).

$$x_r = x_1 + x_2 \quad (25)$$

$$y_r = y_1 + y_2 \quad (26)$$

En el caso del prototipo diseñado en el proyecto de grado, mostrado en la Figura 16, los sensores laterales se encuentran a 0 cm en el eje y con respecto al punto de referencia, mientras que los sensores superior e inferior se encuentran a 0 cm en el eje x con respecto al punto de referencia. Luego de realizar los cálculos requeridos, se crea una matriz de 140 filas por 5 columnas; para efectos prácticos se decide que los servomotores giren desde 5° hasta 175° en pasos de 5° , lo que

implica que cada sensor de ultrasonido obtenga 35 mediciones. Con los cuatro sensores se obtienen un total de 140 mediciones que deben ser almacenadas en la misma cantidad de posiciones. En la primera columna de la matriz se almacena el valor del ángulo del sensor respectivo, en la segunda la distancia en el eje x con respecto al sensor, en la tercera la distancia en el eje y con respecto al sensor, en la cuarta la distancia real con respecto al punto de referencia, y en la quinta el valor del ángulo entre el objeto y el punto de referencia.

Después de almacenar los datos en la matriz creada se procede a obtener las distancias más cortas encontradas en la matriz, con esto se logran enfocar los puntos donde se encuentran los objetos más cercanos con respecto a cada sensor, como se puede observar en la Figura 20 .

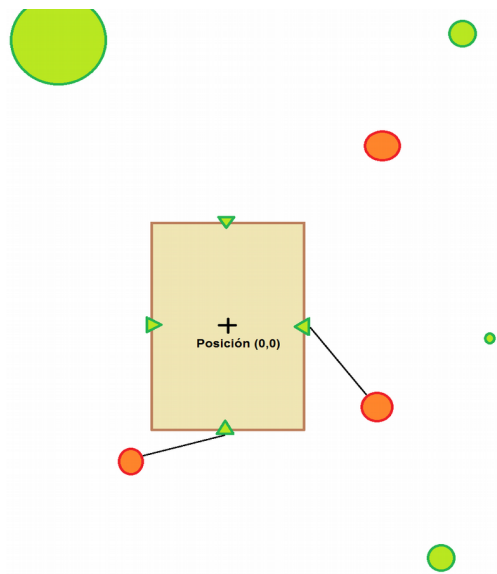


Figura 20: enfoque en los objetos más cercanos con respecto a los sensores

Luego, se encuentran las distancias a los objetos más cercanos con respecto al punto de referencia, haciendo uso de las ecuaciones (25) y (26). En la Figura 21 se puede observar como se enfocan los objetos más cercanos. Con las distancias de los ejes x y y obtenidas es posible calcular el ángulo de ubicación del objeto utilizando la ecuación (27):

$$\theta = \tan^{-1}\left(\frac{co}{ca}\right) \quad (27)$$

Donde:

- θ : ángulo entre el punto de referencia y el objeto
- ca : distancia entre el punto de referencia y el objeto con respecto al eje x
- co : distancia entre el punto de referencia y el objeto con respecto al eje y

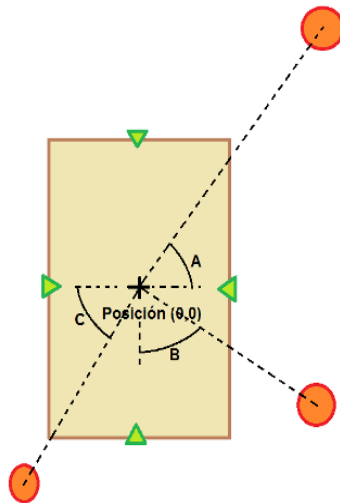


Figura 21: enfoque en los objetos más cercanos con respecto al punto de referencia

El sistema de reconstrucción 3D tiene la capacidad de girar entre 0° y 360° para realizar la captura de la profundidad de los objetos detectados. Este sistema se ubica en las posiciones donde se encuentran los objetos más cercanos y realiza el proceso de captura de profundidad únicamente cuando los servomotores que hacen girar a los sensores de ultrasonido llegan a la posición de 5° o 170° , es decir, en cada uno de los extremos.

La información obtenida por el sistema de reconstrucción 3D es almacenada en el Raspberry Pi y posteriormente es enviada junto con los datos de la matriz de dimensiones 140×5 , a través de un servicio SSH que permite conexión entre diferentes computadores, utilizando claves públicas y privadas. El diagrama de flujo 6 representa el algoritmo implementado en el computador Raspberry Pi.

En el diagrama de flujo 1 se presentan los llamados a las funciones “Captura Foto” y “Gráficas”, que son realizados de forma paralela; estas funciones son mostradas en los diagramas de flujo 2 y 3 respectivamente. En el diagrama de flujo 4 se observa un llamado a la función “Guardar” en el que se transfieren las imágenes obtenidas hacia otro sistema de procesamiento a través de un servicio SSH.

6.6 SISTEMA DE PROCESAMIENTO Y REPRESENTACIÓN GRÁFICA DE DATOS

Es necesario otro sistema de procesamiento, en este caso un computador, para el procesamiento y la representación gráfica de las matrices de datos, debido a que el Raspberry Pi no tiene la suficiente potencia computacional para esta tarea en tiempos razonables; adicionalmente, al utilizar una pantalla dentro del prototipo se tendrían cables y conexiones que pueden llegar a interferir en la toma de medidas por parte de los sensores de ultrasonido. En la Figura 22 se observa una implementación del proyecto en la cual se encuentran cables externos, junto a un teclado y un mouse utilizados para la programación del Raspberry Pi; todos estos elementos pueden presentar interferencia a las señales enviadas y recibidas por los sensores.



Figura 22: prototipo con conexiones cableadas y elementos adicionales

Para efectos prácticos se decide utilizar una batería como la mostrada en la Figura 23, con las siguientes características [43] :

- Voltaje Nominal: 12 V
- Capacidad Nominal: 5 Ah



Figura 23: batería usada en el prototipo

Debido a que existen varios elementos en el prototipo que trabajan a diferentes voltajes, se utilizan convertidores DC-DC, que permiten regular a un voltaje deseado. Estos convertidores se pueden observar en la Figura 24.

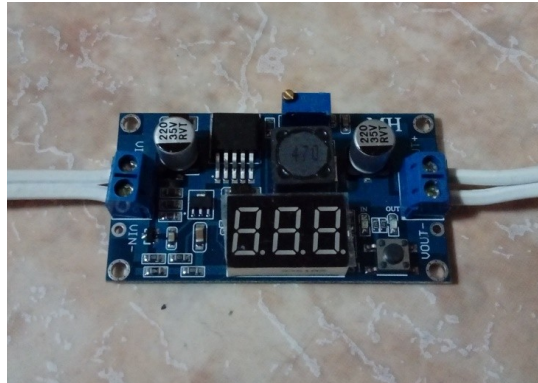


Figura 24: conversores DC-DC utilizados para el proyecto

Con el uso de la batería y los conversores se tiene un sistema completamente independiente de alimentación por cables, como se observa en la Figura 25. Este prototipo ya no presenta interferencia alguna con elementos adicionales.



Figura 25: prototipo sin conexiones cableadas ni elementos adicionales

El computador para procesamiento y representación gráfica tiene las siguientes características:

- Procesador: Intel Core I3 de 2.26 GHz
- Memoria ram: 4 GB DDR3
- Sistema Operativo: Linux Mint 18.3 LTS

Los programas y algoritmos se desarrollaron con el lenguaje de programación Python, versión 2.7, tanto en el Raspberry Pi como en el computador de procesamiento. La transferencia de archivos al computador se realiza mediante conexión WiFi, gestionando una red LAN local por medio de un router TrendNet [44], como el mostrado en la Figura 26, que asigna direcciones IP fijas a los computadores por medio de DHCP.



Figura 26: router TrendNet 651BR para conexión WiFi

6.7 SISTEMA DE RECONSTRUCCIÓN 3D DEL ENTORNO

Para el sistema de reconstrucción 3D, se usó el sensor Kinect de Microsoft, rotando hasta 360 ° gracias al sistema tratado en la sección 6.3, conectado a a la unidad de procesamiento Raspberry Pi por medio del puerto USB. Con el dispositivo anteriormente mencionado, se genera una matriz de disparidad, definiendo disparidad como la representación de la diferencia de la posición relativa de un punto en dos o más imágenes [45]; en el computador de procesamiento y visualización se procesa esta matriz para obtener la reconstrucción del entorno 3D y con esto, su representación gráfica en 2D.

El sensor Kinect genera la matriz de disparidad automáticamente, a diferencia de las cámaras que requieren algoritmos adicionales para esa tarea; esta matriz es de tamaño 640 x 480, con datos de 11 bits [46]. En el Raspberry Pi se usa la librería Freenect [47] para activar el Kinect y obtener la matriz de disparidad. Se requiere convertir el tipo de datos de la matriz de disparidad a 8 bits para su visualización, de ahí que se utilice el método *copy* de la librería Numpy. Después de realizar el procedimiento anterior, el Raspberry Pi envía el par disparidad – imagen de 8 bits al equipo que realiza el procesamiento por medio de la red local.

En el equipo de procesamiento se carga la matriz de disparidad y se calcula la profundidad de cada punto de esta a partir de la ecuación (28), siguiendo la metodología [48], descartando

valores iguales a 2047, que representan puntos donde no se detectó correctamente la profundidad.

$$Z = \frac{1}{c_1 d_u + c_0} \quad (28)$$

Donde:

- Z : profundidad de cada punto
- c_1 y c_0 : parámetros intrínsecos de la cámara infrarroja del Kinect
- d_u : valor de disparidad para cada punto

Los valores para estos parámetros se basaron en la evaluación de métodos de calibración [49], donde se encontró que los menores errores se obtienen con los valores que se encuentran en la tabla 2.

Tabla 2. Valores de los parámetros para el cálculo de profundidad con Kinect

Parámetro	Valor
c_1	3.0946
c_0	-0.0028
f_x	586.6
f_y	577.7
c_x	318.92
c_y	231.46

Con Z se encuentran las coordenadas X y Y , empleando para esto las ecuaciones de proyección (14) y (15), vistas en la sección 5.3, y los valores de las matrices de rotación y traslación, también basados en la metodología [49]. Finalmente, se unen los vectores con las coordenadas Z , X y Y , con el método *hstack*, para crear la matriz de reconstrucción 3D.

6.8 INTERFAZ GRÁFICA DE USUARIO

La interfaz gráfica de usuario se desarrolló usando las clases y métodos de la librería PyQt4 [50], la cual cuenta con un marco en la parte superior donde se muestra la graficación de la reconstrucción 3D y seis botones con las siguientes opciones:

- Graficar: inicio del programa
- Cambiar vista: modificar la perspectiva de la representación gráfica
- Girar a izquierda: rotar la representación gráfica 10° en el plano XY , en sentido horario
- Girar a derecha: rotar la representación gráfica 10° en el plano XY , en sentido antihorario
- Inclinar abajo: rotar a representación gráfica 10° en el plano Z , en sentido horario
- Inclinar arriba: rotar a representación gráfica 10° en el plano Z , en sentido antihorario

Se cuenta que con dos campos de texto que muestran el estado del programa y las mediciones hechas.

La representación gráfica del entorno 3D se realiza por medio de la librería Matplotlib [51], usando los métodos `plot_surface` y `scatter` para graficar el prototipo y los puntos de reconstrucción detectados, respectivamente.

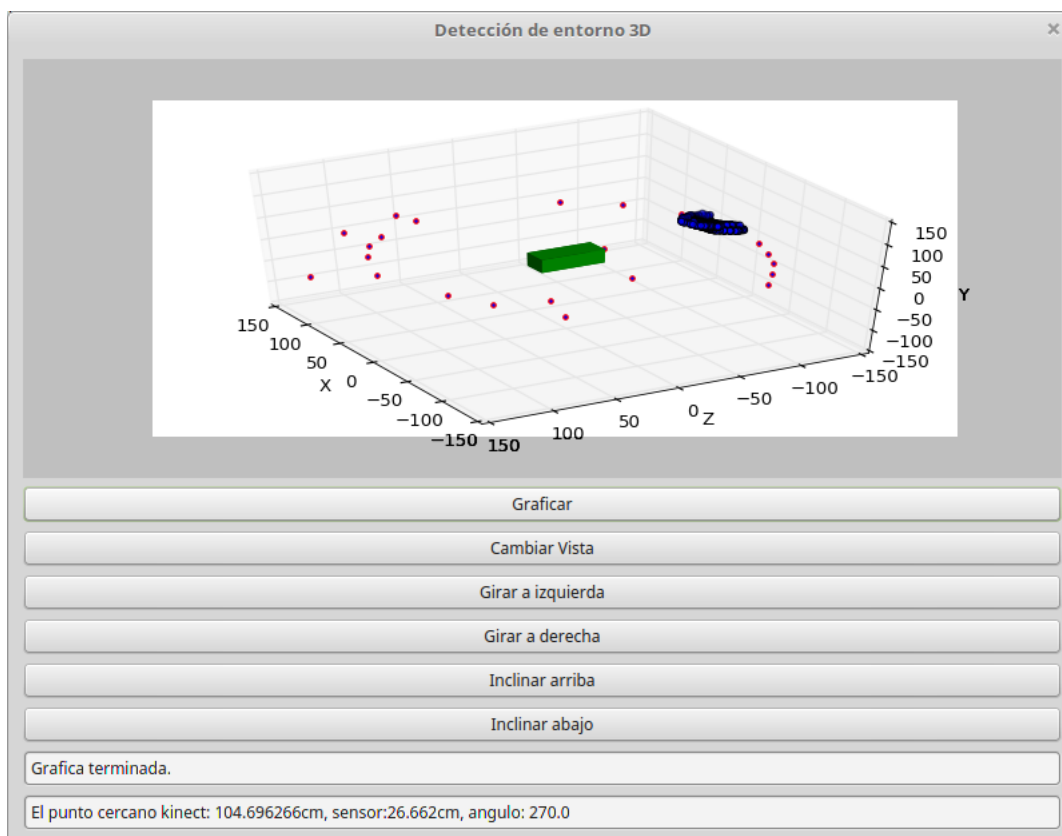


Figura 27: Interfaz gráfica de usuario

7 RESULTADOS

A continuación se presentan los resultados obtenidos con el sistema de detección de objetos cercanos. Los porcentajes de error se calculan de acuerdo a la ecuación (29) , los promedios de error se obtienen usando (30).

$$\text{Porcentaje de Error}(\%) = \left| \frac{\text{Valor Teórico} - \text{Valor Experimental}}{\text{Valor Teórico}} \right| * 100\% \quad (29)$$

$$\text{Error promedio}(\%) = \sum_{i=0}^n \text{Porcentaje de Error}(i) \quad (30)$$

7.1 MEDICIONES CON SENSORES DE ULTRASONIDO

La distancia real se mide mediante el uso de una regla o flexómetro, mientras que la distancia calculada es la obtenida por medio del procesamiento de los datos en los algoritmos implementados en el Raspberry Pi.

En la Tabla 3 se muestran las mediciones de distancia hechas a partir de los datos obtenidos por los sensores de ultrasonido.

Tabla 3. Datos de Distancias Reales y Calculadas por un sensor de ultrasonido.

Distancia Real (cm)	Distancia Calculada (cm)	Porcentaje de Error (%)
10	9.92	0.800
	10.03	0.300
	9.93	0.700
	10.04	0.400
	10.04	0,400
25	25.03	0.120
	24.96	0.160
	25.07	0.280
	25.06	0.240
	24.95	0.200
48	48.02	0.041
	47.97	0.062
	48.00	0.000
	47.98	0.041
	48.03	0.0625
57	57.10	0.175
	57.09	0.157
	57.13	0.228
	57.07	0.122
	57.10	0.175
91	91.44	0.483
	91.23	0.252
	90.56	0.483
	91.01	0.010
	90.68	0.351
110	109.63	0.336
	110.66	0.600
	111.48	1.345
	110.15	0.136
	110.39	0.355
143	142.21	0.552
	144.35	0.944
	143.90	0.629
	142.18	0.573
	142.07	0.650
165	163.78	0.739

	164.01	0.600
	164.25	0.455
	164.42	0.351
	163.54	0.884
183	181.43	0.858
	182.11	0.486
	183.94	0.514
	181.08	1.049
	182.65	0.191
200	198.54	0.730
	198.47	0.765
	199.28	0.360
	199.29	0.355
	198.99	0.505
275	274.87	0.047
	275.25	0.090
	275.72	0.261
	276.25	0.454
	276.16	0.421
347	346.10	0.259
	345.43	0.452
	345.41	0.458
	344.91	0.602
	344.50	0.720
400	400.96	0.240
	400.04	0.010
	400.04	0.010
	400.07	0.017
	400.06	0.015

En la tabla 4 se toman las mediciones del ángulo real que rotan los servomotores del sistema de rotación de los sensores de ultrasonido, el programado en el dispositivo y el porcentaje de error; el ángulo real se mide con un transportador .

Tabla 4 Datos de Ángulos

Ángulo Programado en grados	Ángulo medido en grados	Desfase en grados	Porcentaje de Error (%) con respecto a 180°
10	14.0	4.0	2.2
	14.5	4.5	2.5
	14.5	4.5	2.5
	14.0	4.0	2.2
	14.0	4.0	2.2
30	35.0	5.0	2.7
	34.5	4.5	2.5
	35.0	5.0	2.7
	35.0	5.0	2.7
	35.0	5.0	2.7
50	54.0	4.0	2.2
	53.5	3.5	1.9
	54.0	4.0	2.2
	54.0	4.0	2.2
	53.5	3.5	1.9
70	76.0	6.0	3.3
	76.0	6.0	3.3
	76.0	6.0	3.3
	75.5	5.5	3.0
	75.5	5.5	3.0
90	94.0	4.0	2.2
	94.0	4.0	2.2
	94.0	4.0	2.2
	94.0	4.0	2.2
	94.0	4.0	2.2
110	114.0	4.0	2.2
	114.0	4.0	2.2
	113.5	3.5	1.9
	114.0	4.0	2.2
	114.0	4.0	2.2
130	133.0	3.0	1.6
	133.0	3.0	1.6
	133.0	3.0	1.6

	133.0	3.0	1.6
	133.5	3.5	1.9
150	155.0	5.0	2.7
	154.5	4.5	2.5
	155.0	5.0	2.7
	155.0	5.0	2.7
	155.0	5.0	2.7
170	175.0	5.0	2.7
	175.0	5.0	2.7
	175.0	5.0	2.7
	175.0	5.0	2.7
	175.0	5.0	2.7

En la tabla 5 se puede observar la distancia que es obtenida cuando se tiene un objeto a distintos ángulos con respecto al eje horizontal del sensor, el objeto se encuentra a una distancia fija de 45 cm al prototipo.

Tabla 5. Datos de distancias con respecto a un ángulo del objeto a una distancia de 45 cm

Ángulo (cm)	Distancia	Porcentaje de
	Obtenida (cm)	Error (%)
30	3034.112	100.00
	3049.494	100.00
	3062.562	100.00
	3058.735	100.00
	3073.810	100.00
45	3062.361	100.00
	59.685	32.63
	3063.159	100.00
	59.272	31.71
	3063.367	100
60	3078.304	100.00
	45.368	0.81
	3055.228	100.00

	42.299	0.50
	3091.458	100
80	45.276	0.61
	45.484	1.07
	45.223	0.49
	45.207	0.46
	45.256	0.56
90	45.055	0.12
	45.035	0.07
	45.088	0.19
	45.035	0.07
	45.055	0.12
100	45.258	0.57
	45.387	0.86
	45.243	0.25
	45.301	0.66
	45.109	0.24
120	108.941	100.00
	112.145	100.00
	104.994	100.00
	104.998	100.00
	107.893	100.00
135	104.050	100.00
	108.351	100.00
	108.286	100.00
	100.554	100.00
	107.926	100.00
170	3066.322	100.00
	3069.439	100.00
	3066.058	100.00
	3059.577	100.00
	3070.384	100.00

7.2 MATRIZ DE DATOS EN RASPBERRY PI

De acuerdo a los datos procesados en el Raspberry, realizando toda la metodología propuesta y utilizando las ecuaciones de la sección 7.1, se obtienen como resultado los valores mostrados en la figura 28, la cual muestra los datos calculados y preparados para ser entregados al sistema de procesamiento.

La primer columna de la figura 28 especifica el ángulo del objeto con respecto a cada sensor, la segunda muestra la distancia con respecto al sensor, la tercera indica la posición con respecto al eje x entre el objeto y el punto de referencia, la cuarta columna muestra la distancia en el objeto y el punto de referencia con respecto al eje y , y la última columna indica el ángulo del objeto con respecto al punto de referencia.

	0	1	2	3	4
0	5.000	100.616	33.803	101.000	18.570
1	10.000	100.450	42.712	102.000	23.035
2	15.000	73.410	44.670	76.000	31.321
3	20.000	70.477	50.652	75.000	35.705
4	25.000	67.973	56.696	75.000	39.831
5	30.000	64.952	62.500	75.000	43.898
6	35.000	59.798	66.871	73.000	48.196
7	40.000	61.284	76.423	80.000	51.274
8	45.000	57.276	82.276	81.000	55.157
9	50.000	51.423	86.284	80.000	59.206
10	55.000	46.460	91.351	81.000	63.043
11	60.000	24.500	67.435	49.000	70.033
12	65.000	19.440	66.690	46.000	73.748
13	70.000	15.733	68.226	46.000	77.015
14	75.000	11.906	69.433	46.000	80.270
15	80.000	7.988	70.301	46.000	83.518
16	85.000	4.009	70.825	46.000	86.760
17	90.000	0.000	71.000	46.000	90.000
18	95.000	-7.321	108.680	84.000	-86.146
19	100.000	-0.868	29.924	5.000	-88.338
20	105.000	-1.812	31.761	7.000	-86.735
21	110.000	-10.261	53.191	30.000	-79.082
22	115.000	-2.958	31.344	7.000	-84.608
23	120.000	-3.000	30.196	6.000	-84.326
24	125.000	-3.441	29.915	6.000	-83.437
25	130.000	-4.500	30.362	7.000	-81.570

Format Resize Background color

Cancel OK

Figura 28: datos entregados por el dispositivo de procesamiento Raspberry Pi

7.3 GRÁFICA DE MATRIZ DE PUNTOS

Una de las matrices creadas en el dispositivo de procesamiento Raspberry Pi contiene los puntos de distancias obtenidos con los que es posible realizar un análisis básico de los objetos y obtener una gráfica en 2D.

En la Figura 29 se pueden observar los puntos encontrados por los sensores de ultrasonido. Los puntos se encuentran en un plano 3D pero tienen una altura constante porque los sensores no se desplazan a través del eje z del plano, para una mejor representación visual en esta figura se muestran en color azul los puntos que tienen una distancia menor a 2 m, mientras que los puntos que superan esta distancia se encuentran en color rojo ya que no son tomados en cuenta para las mediciones; el prototipo se muestra en color verde. La figura 30 muestra la gráfica de los puntos detectados por los sensores de ultrasonido, en donde no se visualizan los puntos despreciables. A partir de las figuras anteriores se puede inferir que si todos los puntos fuesen mayores a 200 cm no se graficarían.

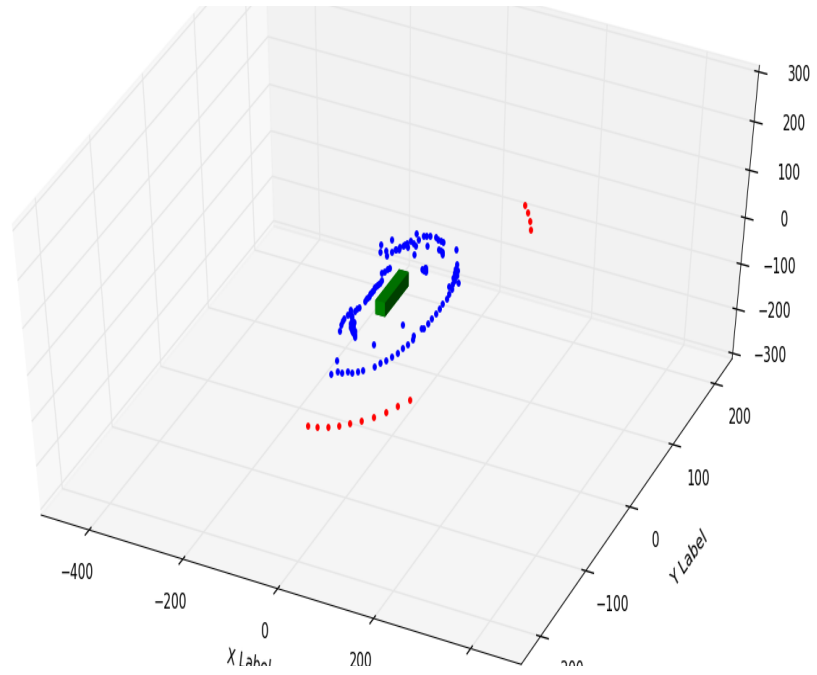


Figura 29: puntos encontrados por los sensores de ultrasonido

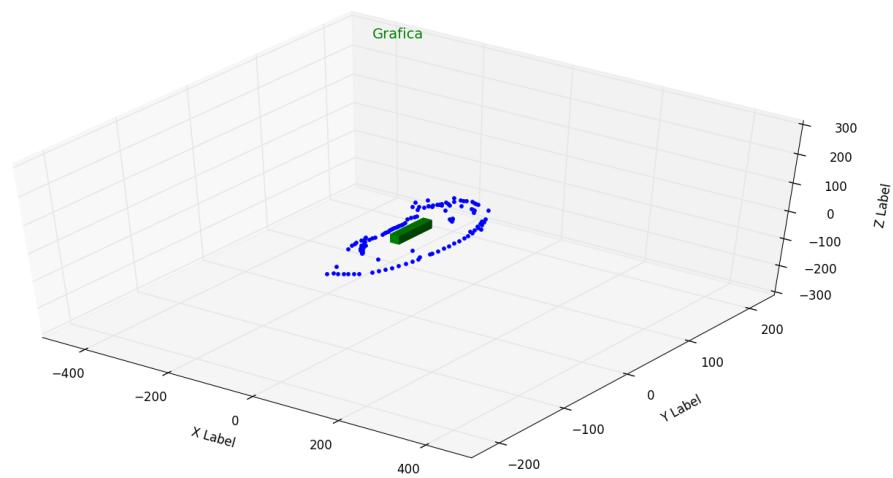


Figura 30: puntos encontrados por los sensores de ultrasonido, sin puntos despreciables

7.4 RESULTADOS DE LA RECONSTRUCCIÓN 3D

Para la medición de distancias a partir de la reconstrucción 3D se utiliza una superficie plana, en este caso una tabla de madera, se captura una escena donde se conoce la distancia del objeto al sensor Kinect y se encuentra la medida de la profundidad del punto más cercano perteneciente a dicho objeto. En la tabla 6 se observan los resultados de las mediciones.

Tabla 6. Medidas de puntos cercanos por medio de Reconstrucción 3D

Distancia (cm)	Distancia Medida (cm)	Porcentaje de Error (%)
48	49.2311	2.564
	49.0313	2.148
	49.0209	2.127
57	57.9744	1.709
	57.7629	1.338
	57.8538	1.497
91	89.3398	1.824
	90.0145	1.082
	90.041	1.053
110	107.312	2.443
	107.488	2.283
	107.723	2.069
143	137.062	4.152
	136.584	4.486
	137.098	4,127
165	155.709	5.630
	156.357	5.238
	157.036	4.826
183	173.065	5.428
	173.068	5.427
	172.254	5.871
200	188.491	5.754
	187.498	6.251

	188.492	5.754
275	245.628	10.680
	242.354	11.871
	247.537	9.986
347	296.6686	14.499
	301.71	13.051
	296.664	14.506
400	342.364	14.408
	339.342	15.164
	342.258	14.355

7.4.1 REPRESENTACIÓN GRÁFICA DEL ENTORNO 3D

La matriz de reconstrucción 3D se grafica junto con una representación del prototipo. En esta prueba se usa un objeto cilíndrico con dimensiones de 29 cm de alto y 8 cm de diámetro, a una distancia de 91 cm de este al prototipo. En la figura 31 se muestra una fotografía de la prueba realizada, en la figura 40 se encuentra la representación gráfica del entorno 3D.

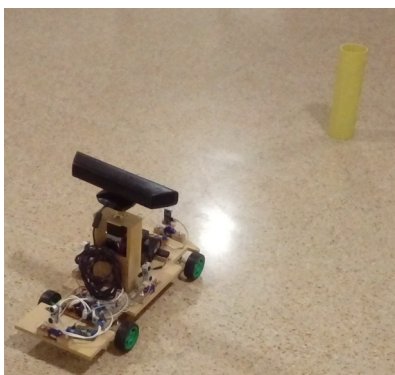


Figura 31: prototipo y objeto de prueba

Reconstrucción del entorno 3D

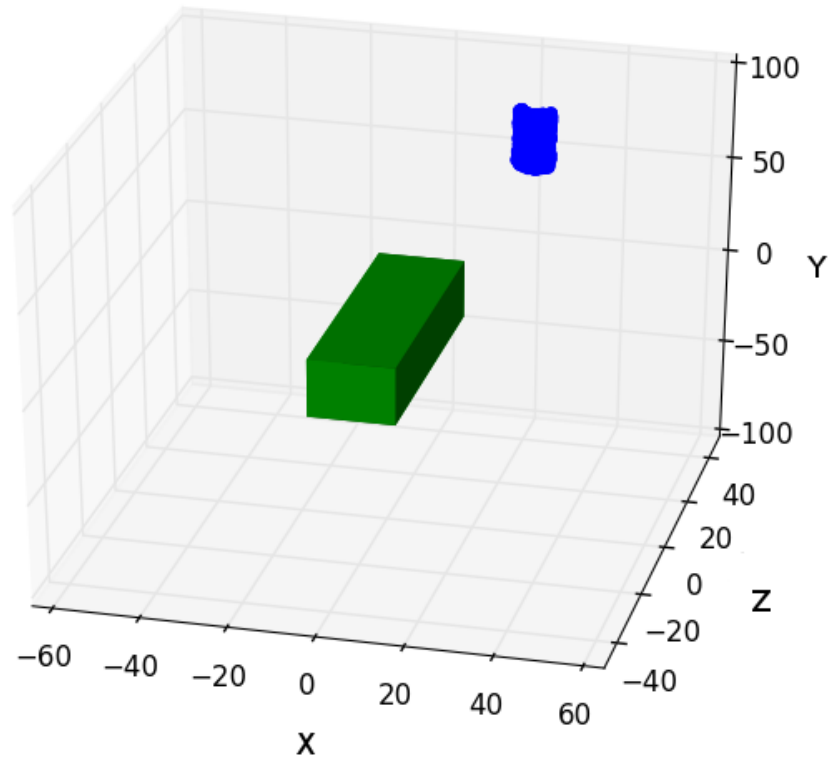


Figura 32: representación gráfica del entorno 3D del móvil

Las dimensiones del objeto de la representación en 3D se calculan a partir de la resta entre los valores máximos y mínimos de las coordenadas x y y . Se usan histogramas en las columnas correspondientes a los ejes dentro de la matriz de reconstrucción 3D, para descartar puntos que estén alejados del promedio ya que estos se deben al ruido en la captura de la escena y por lo tanto, afectan la medición del objeto. En las figuras 33 y 34 se observan los histogramas de los ejes x y y , respectivamente; el eje de las abscisas representa el valor de la coordenada, el eje de las ordenadas representa la cantidad de puntos por valor. De acuerdo a la reconstrucción 3D del

objeto se encuentra que su distancia al móvil es de 89.9 cm y sus dimensiones son de 27.382 cm de alto y 7.909 cm de ancho.

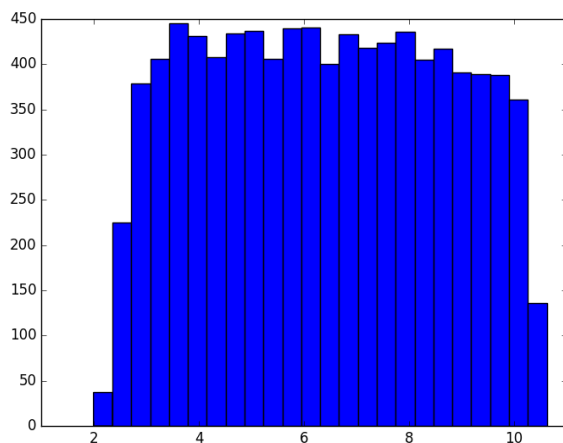


Figura 33: histograma de las coordenadas en x

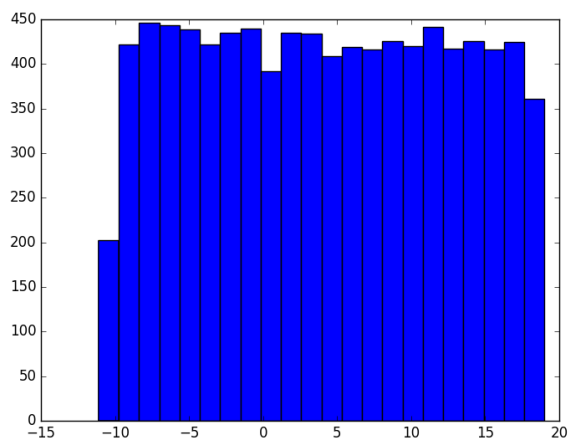


Figura 34: histograma de las coordenadas en y

8 ANÁLISIS DE RESULTADOS

En el prototipo final se descartó el uso del dispositivo Launchpad, ya que se evidenció que se presenta una caída de la corriente que suministra la batería al momento en que los servomotores y el motor paso a paso se activan simultáneamente, teniendo como consecuencia que el motor paso a paso tome toda la corriente y con esto no se permita el funcionamiento correcto de los servomotores. El código implementado en Launchpad fue portado al Raspberry Pi, con la ventaja adicional que no se requieren circuitos adicionales ya que todos los dispositivos trabajan a 5 V.

En el caso de la estructura de rotación del sistema de adquisición de imágenes, se llevó a cabo una prueba instalando dos servomotores para realizar el giro desde 0° hasta 360°, como se observa en la figura 11. Se evidenció que la estructura presenta bastante inestabilidad ya que el centro de masa se encuentra en la parte superior, es decir, el peso de las cámaras dentro de la estructura de madera es mayor en comparación al sistema de rotación; teniendo como consecuencia oscilación en las estructuras, lo que dificulta la captura de imágenes. Otra razón para la inestabilidad es que el punto de anclaje entre la estructura de las cámaras de video y los servomotores es muy pequeño para lograr tener firmeza. Por lo anterior, se decidió cambiar a una estructura más sencilla y firme como la implementada con el motor paso a paso, la cual tiene la capacidad de girar 360°, además de llevar más carga que los dos servomotores.

Las mediciones realizadas utilizando los sensores de ultrasonido, mostradas en la tabla 3 , indican valores de porcentajes de error que no superan el 1% y se evidencia que son aproximadamente constantes, incluso cuando se varía la distancia del objeto a detectar. Se realizó un gráfico mostrado en la figura 35 en la que se relaciona la distancia real con respecto al promedio de las distancias obtenidas, encontrando que la curva de esta relación presenta un nivel casi constante y lineal.

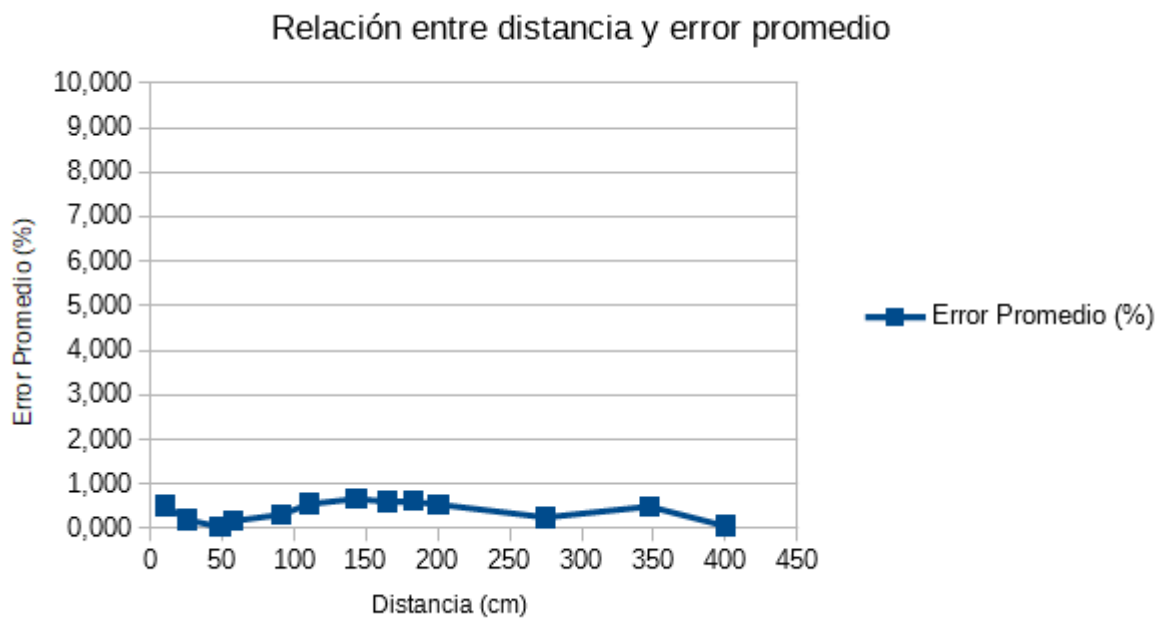


Figura 35: relación para sensores de ultrasonido

Estos porcentajes indican que puede haber un nivel de precisión bastante alto para encontrar las distancias de los objetos cercanos al prototipo; así, la información procesada presentará datos

confiables para que sirvan de base a los que se obtienen con el sensor Kinect. La información obtenida también depende de la precisión del ángulo representado por el movimiento de los servomotores. En la tabla 4 se puede determinar que los servomotores tienen un error aproximadamente constante, con respecto a un valor que es medido a través de un transportador. Estas mediciones presentan error debido a la manipulación del instrumento, adicionalmente, el ángulo medido a través del transportador es obtenido visualmente, lo que aumenta las probabilidades de obtener error en dicha medición.

A pesar que los sensores de ultrasonido tienen un porcentaje de error bajo, tienen sus limitaciones con respecto al ángulo de medida, tanto del objeto como del mismo sensor. Según la tabla 5 se observa que el ángulo de reflexión del objeto con respecto al sensor debe ser cercano a 90° , con un rango entre 80° y 100° con respecto a este; esto es debido a que las ondas de ultrasonido generadas por el sensor pueden cambiar de dirección cuando este ángulo varía, por lo que no llegarían de vuelta al sensor de ultrasonido y por lo tanto, este dispositivo no recibiría la señal.

En la figura 36 se observa la reflexión de una señal cuando el objeto es girado en un ángulo determinado. En la figura 37 se puede observar una mesa que se tomó como obstáculo, en la imagen izquierda se tiene la mesa en un ángulo con el que no es posible obtener una medición de la distancia porque el ángulo no permite que la señal sea retornada al sensor de ultrasonido. En la

imagen derecha la mesa tiene un ángulo diferente que si permite el retorno de la señal al sensor de ultrasonido.

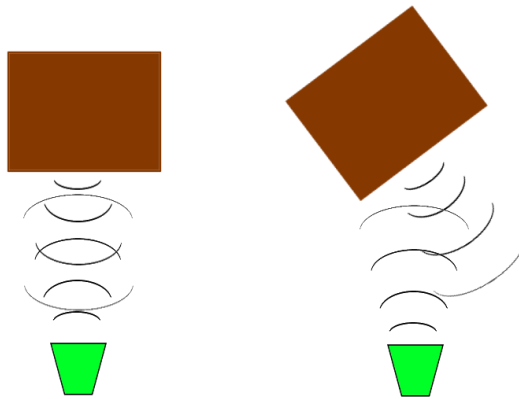


Figura 36: reflexión de señal de sensores de ultrasonido



Figura 37: reflexión en mesa con un ángulo diferente

Con respecto al dispositivo Kinect, se tomaron mediciones de distancias de prueba como las mostradas en la tabla 6. A partir de estas mediciones se encontraron los porcentajes de error, que

para este caso tienden a aumentar con respecto a la distancia de sensado. Se realizó un gráfico mostrado en la figura 38 en la que se relaciona la distancia real con el promedio de las distancias obtenidas; se observa mediante esta figura, que el aumento del porcentaje de error es aproximadamente logarítmico. Se logra determinar que la distancia de los objetos a detectar, con un error tolerable usando el sensor Kinect, se encuentra en el rango de 1.5 m a 2 m.

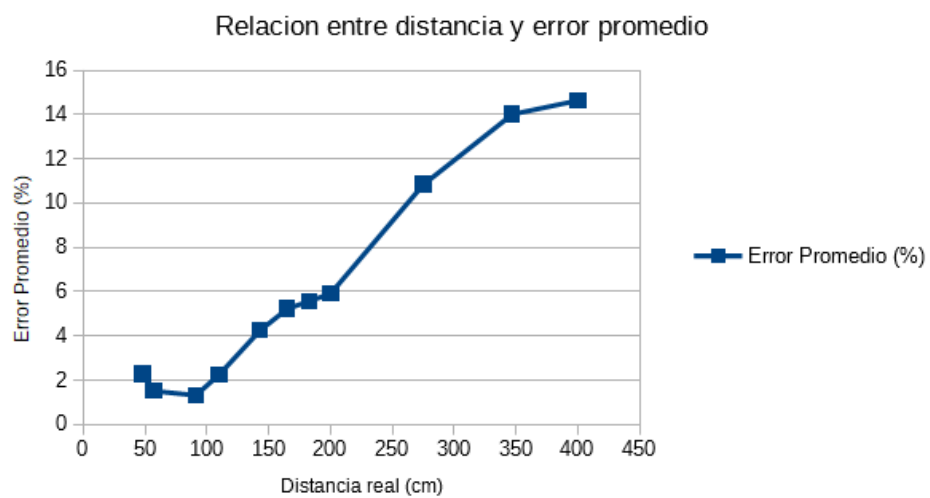


Figura 38: relación para sensor Kinect

9 CONCLUSIONES

El uso de sensores de ultrasonido permite generar entornos 2D si se realiza una integración de varios sensores y a la vez una rotación, esto es útil si se requiere realizar un análisis de los puntos más cercanos a cualquier móvil; es posible generar un ambiente 3D de puntos si se incluye un desplazamiento vertical, lo que permite determinar obstáculos a diferentes alturas o con diferentes tamaños. Para móviles con diferentes dimensiones, es posible utilizar un determinado número de sensores, lo que permite cubrir en su totalidad todo el espacio posible que lo rodea.

Con el modelo de sensor de ultrasonido, utilizado en este proyecto de grado, se tienen las ventajas de bajo costo, fácil adquisición y dimensiones reducidas; adicionalmente, existe la posibilidad de detectar objetos hasta 6 m de distancia, teniendo en cuenta las limitaciones con respecto al ángulo de sensado.

Los dispositivos de sensado Kinect permiten obtener entornos 3D, lo que brinda una recopilación más detallada de los obstáculos. Con Kinect se tiene un rango de trabajo menor que las cámaras de video, pero con más funcionalidades debido a que están integrados con un sistema láser, una cámara de video RGB y una cámara infrarroja, generando automáticamente matrices con las coordenadas de puntos de la escena 3D, a varios niveles de iluminación y texturas.

Ahora, si se integran los sensores de ultrasonido con Kinect, se puede realizar un análisis más detallado del entorno 3D que rodea a un móvil, permitiendo encontrar los puntos más cercanos a este y determinar la forma del obstáculo que se encuentra en esas posiciones del espacio.

El diseño realizado para construir el prototipo sin la necesidad de elementos exteriores de conexión o dependencia a fuentes de energía y/o visualización adicional, lo convierte en un sistema autónomo con la capacidad de transmisión de datos inalámbrica a un segundo sistema y evita el uso de pantallas para visualización, pues en este segundo sistema se presenta el procesamiento y visualización de los datos.

Entre los trabajos futuros aplicando los enfoques presentados en este trabajo de grado se encuentran:

- Detección del entorno 3D de móviles en tiempo real
- Identificación y estructuración de obstáculos
- Toma de decisiones autónomas en base a la reconstrucción 3D
- Generación de ambientes en 3D utilizando únicamente sensores de ultrasonido

10 REFERENCIAS

- [1] M. M. Almasri, A. M. Alajlan, y K. M. Elleithy, «Trajectory Planning and Collision Avoidance Algorithm for Mobile Robotics System», *IEEE Sens. J.*, vol. 16, n.º 12, pp. 5021-5028, jun. 2016.
- [2] H. W. Wehn y P. R. Belanger, «Ultrasound-based robot position estimation», *IEEE Trans. Robot. Autom.*, vol. 13, n.º 5, pp. 682-692, oct. 1997.
- [3] L. Feng, H. R. Everett, J. Borenstein, University of Michigan, Oak Ridge National Laboratory. D&D Program, y United States. Department of Energy, «Where am I?» : *sensors and methods for autonomous mobile robot positioning.*, vol. 3. Michigan: University of Michigan, 1994.
- [4] M. Zhou, H.-X. Li, y M. Weijnen, «Data Processing in Current 3D Robotic Perception Systems», en *Contemporary Issues in Systems Science and Engineering*, Wiley-IEEE Press, 2015, p. 888-.
- [5] M. Kam, X. Zhu, y P. Kalata, «Sensor fusion for mobile robot navigation», *Proc. IEEE*, vol. 85, n.º 1, pp. 108-119, ene. 1997.
- [6] S. Haykin y K. J. R. Liu, «Sensor Data Fusion with Application to Multitarget Tracking», en *Handbook on Array Processing and Sensor Networks*, Wiley-IEEE Press, 2009, pp. 499-531.

- [7] D. Macii, A. Boni, M. D. Cecco, y D. Petri, «Tutorial 14: multisensor data fusion», *IEEE Instrum. Meas. Mag.*, vol. 11, n.º 3, pp. 24-33, jun. 2008.
- [8] M. Nagai, T. Chen, R. Shibasaki, H. Kumagai, y A. Ahmed, «UAV-Borne 3-D Mapping System by Multisensor Integration», *IEEE Trans. Geosci. Remote Sens.*, vol. 47, n.º 3, pp. 701-708, mar. 2009.
- [9] N. Gageik, P. Benz, y S. Montenegro, «Obstacle Detection and Collision Avoidance for a UAV With Complementary Low-Cost Sensors», *IEEE Access*, vol. 3, pp. 599-609, 2015.
- [10] F. Garcia, D. Martin, A. de la Escalera, y J. M. Armingol, «Sensor Fusion Methodology for Vehicle Detection», *IEEE Intell. Transp. Syst. Mag.*, vol. 9, n.º 1, pp. 123-133, Spring 2017.
- [11] Y. Dobrev, S. Flores, y M. Vossiek, «Multi-modal sensor fusion for indoor mobile robot pose estimation», en *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2016, pp. 553-556.
- [12] E. Papadakis, *Reference for Modern Instrumentation, Techniques, and Technology: Ultrasonic Instruments and Devices I: Ultrasonic Instruments and Devices I*. Academic Press, 1998.
- [13] S. W. Rienstra y A. Hirschberg, «An introduction to acoustics», *Eindh. Univ. Technol.*, vol. 18, p. 19, 2004.
- [14] R. P. Areny, *Sensores y acondicionadores de señal*. Marcombo, 2004.
- [15] J. P. Bentley, *Principles of Measurement Systems*. Pearson Prentice Hall, 2005.

- [16] J. A. M. Rodríguez, J. V. Oyaga, y S. D. P. S. Cantor, «Fundamentos teórico-prácticos del ultrasonido», *Tecnura*, vol. 10, n.º 20, pp. 4-18, 2007.
- [17] E. Mandado y A. Murillo, «Sensores de ultrasonidos», presentado en Sensores y acondicionadores, E.T.S. de Ingenieros de Telecomunicación de Vigo.
- [18] A. Goshtasby y W. A. Gruver, «Design of a single-lens stereo camera system», *Pattern Recognit.*, vol. 26, n.º 6, pp. 923-937, jun. 1993.
- [19] F. M. M. Verdú y Á. M. P. Moreno, *Fundamentos de visión binocular*. Universitat de València, 2004.
- [20] E. García, *Visión Artificial*, 1.ª ed. Universitat Oberta de Catalunya, 2012.
- [21] «Camera calibration With OpenCV — OpenCV 2.4.13.3 documentation». [En línea]. Disponible en:
http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html.
[Accedido: 11-ago-2017].
- [22] A. de la E. Hueso, *Visión por computador: fundamentos y métodos*. Pearson Educación, 2001.
- [23] J. Bouquet, «Camera Calibration Toolbox for Matlab», 2015. [En línea]. Disponible en:
http://www.vision.caltech.edu/bouquetj/calib_doc/htmls/parameters.html. [Accedido: 11-ago-2017].

- [24] X. Armangué, J. Salvi, y J. Batlle, «A Comparative Review Of Camera Calibrating Methods with Accuracy Evaluation», *Pattern Recognit.*, vol. 35, pp. 1617–1635, 2000.
- [25] D. Mery, *Visión por computador*. Universidad Católica de Chile, 2004.
- [26] J. Heikkila y O. Silven, «A four-step camera calibration procedure with implicit image correction», en *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 1106-1112.
- [27] Z. Zhang, «A flexible new technique for camera calibration», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, n.º 11, pp. 1330-1334, nov. 2000.
- [28] G. Stockman y L. G. Shapiro, *Computer Vision*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.
- [29] R. Hartley y A. Zisserman, *Multiple View Geometry in Computer Vision*, 2.^a ed. New York, NY, USA: Cambridge University Press, 2003.
- [30] D. A. Revelo-Luna, F. D. Usama, y J. F. Flórez-Marulanda, «3D Reconstruction of Scenes by Means of a Stereoscopic Vision System, Based on Feature Extraction and Developed in OpenCV», *Ing. Univ.*, vol. 16, n.º 2, pp. 485-500, dic. 2012.
- [31] J. M. López-Valles, M. A. Fernández, y A. Fernández-Caballero, «Conceptos y técnicas de estereovisión por computador», *Intel. Artif.*, vol. 9, n.º 27, pp. 35-62, 2006.

- [32] Microsoft, «Kinect - Xbox.com», 01-mar-2012. [En línea]. Disponible en: <https://web.archive.org/web/20120301040716/http://www.xbox.com:80/en-US/Kinect>. [Accedido: 14-ago-2017].
- [33] D. Pagliari y L. Pinto, «Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors», *Sensors*, vol. 15, n.º 11, pp. 27569-27589, oct. 2015.
- [34] Z. Zhang, «Microsoft Kinect Sensor and Its Effect», *IEEE Multimed.*, vol. 19, n.º 2, pp. 4-10, feb. 2012.
- [35] J. Han, L. Shao, D. Xu, y J. Shotton, «Enhanced Computer Vision With Microsoft Kinect Sensor: A Review», *IEEE Trans. Cybern.*, vol. 43, n.º 5, pp. 1318-1334, oct. 2013.
- [36] A. Shpunt y Z. Zalevsky, «Depth-varying light fields for three dimensional sensing», US8050461 B2, 01-nov-2011.
- [37] J. Smisek, M. Jancosek, y T. Pajdla, «3D with Kinect», en *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1154-1160.
- [38] V. Morell *et al.*, «A Comparative Study of Registration Methods for RGB-D Video of Static Scenes», may 2014.
- [39] «HC-SR04 Ultrasonic Sensor Distance Measuring Module Ultra01+». [En línea]. Disponible en: <http://www.electfreaks.com/estore/hc-sr04-ultrasonic-sensor-distance-measuring-module-ultra01.html>. [Accedido: 03-ago-2017].

- [40] Tower Pro, «SG90 Digital | Tower Pro». [En línea]. Disponible en: <http://www.towerpro.com.tw/product/sg90-7/>. [Accedido: 25-ago-2017].
- [41] Schneider Electric, «NEMA size 23 1.8° 2-phase stepper motor». [En línea]. Disponible en: <http://motion.schneider-electric.com/downloads/quickreference/NEMA23.pdf>.
- [42] «Easy Driver stepper motor driver». [En línea]. Disponible en: <http://www.schmalzhaus.com/EasyDriver/>. [Accedido: 05-ago-2017].
- [43] Shenzhen Center Power Tech. Co., Ltd, «Welcome to Vision Battery,VRLA battery,SLA Battery,AGM Battery,UPS Battery». [En línea]. Disponible en: http://www.vision-batt.com/product_view.aspx?pid=1&tid=2#links. [Accedido: 25-ago-2017].
- [44] TRENDnet, «TRENDnet | Products | TEW-651BR | Enrutador inalámbrico N150 Home». [En línea]. Disponible en: https://www.trendnet.com/langsp/products/proddetail?prod=230_TEW-651BR#tabs-solution02. [Accedido: 25-ago-2017].
- [45] F. Lecumberry, «Cálculo de disparidad en imágenes estéreo, una comparación», presentado en XI Congreso Argentino de Ciencias de la Computación, 2005.
- [46] M. J. Landau, B. Y. Choo, y P. A. Beling, «Simulating Kinect Infrared and Depth Images», *IEEE Trans. Cybern.*, vol. 46, n.º 12, pp. 3018-3031, dic. 2016.
- [47] «OpenKinect». [En línea]. Disponible en: https://openkinect.org/wiki/Main_Page. [Accedido: 13-ago-2017].

- [48] C. Raposo, J. P. Barreto, y U. Nunes, «Fast and Accurate Calibration of a Kinect Sensor», en *Proceedings of the 2013 International Conference on 3D Vision*, Washington, DC, USA, 2013, pp. 342–349.
- [49] V. Villena-Martínez *et al.*, «A Quantitative Comparison of Calibration Methods for RGB-D Sensors Using Different Technologies», *Sensors*, vol. 17, n.º 2, ene. 2017.
- [50] «Python Wiki: PyQt4». [En línea]. Disponible en: <https://wiki.python.org/moin/PyQt4>. [Accedido: 13-ago-2017].
- [51] «Matplotlib: Python plotting — Matplotlib 2.0.2 documentation». [En línea]. Disponible en: <https://matplotlib.org/>. [Accedido: 25-ago-2017].

11 ANEXOS

11.1 Anexo 1

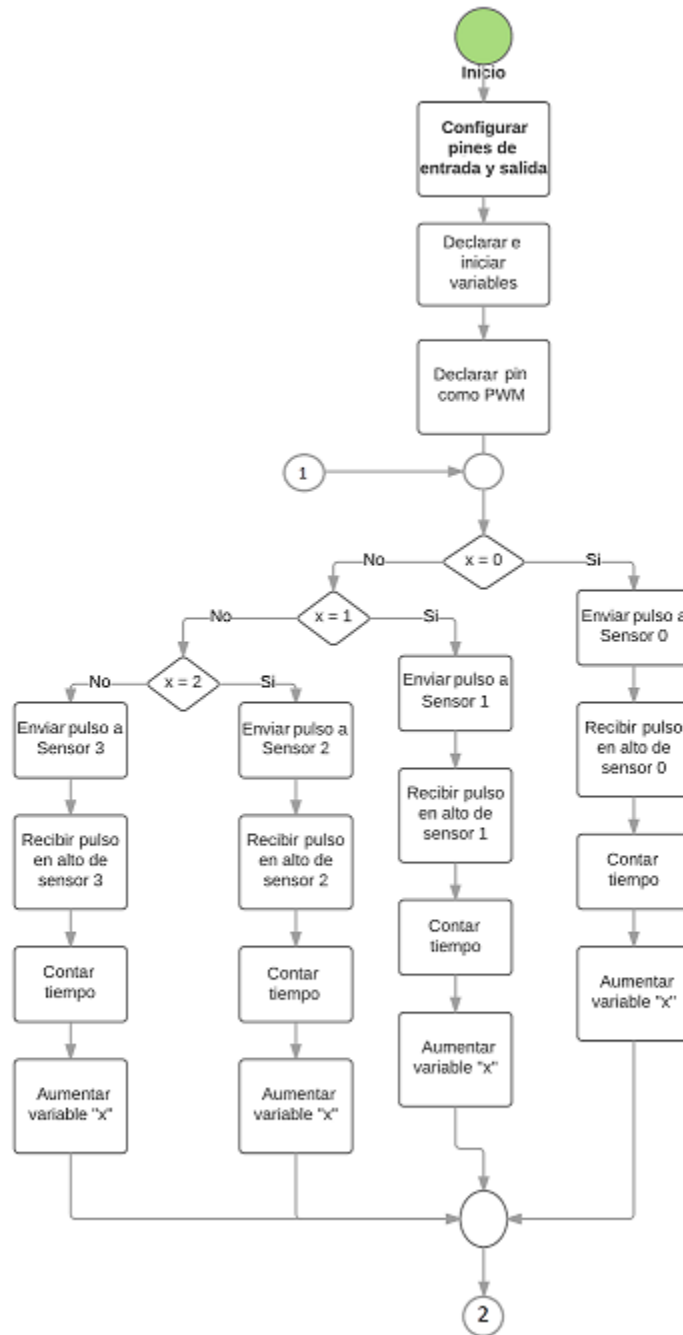


Diagrama 1: control del sistema de detección con sensores de ultrasonido, parte 1

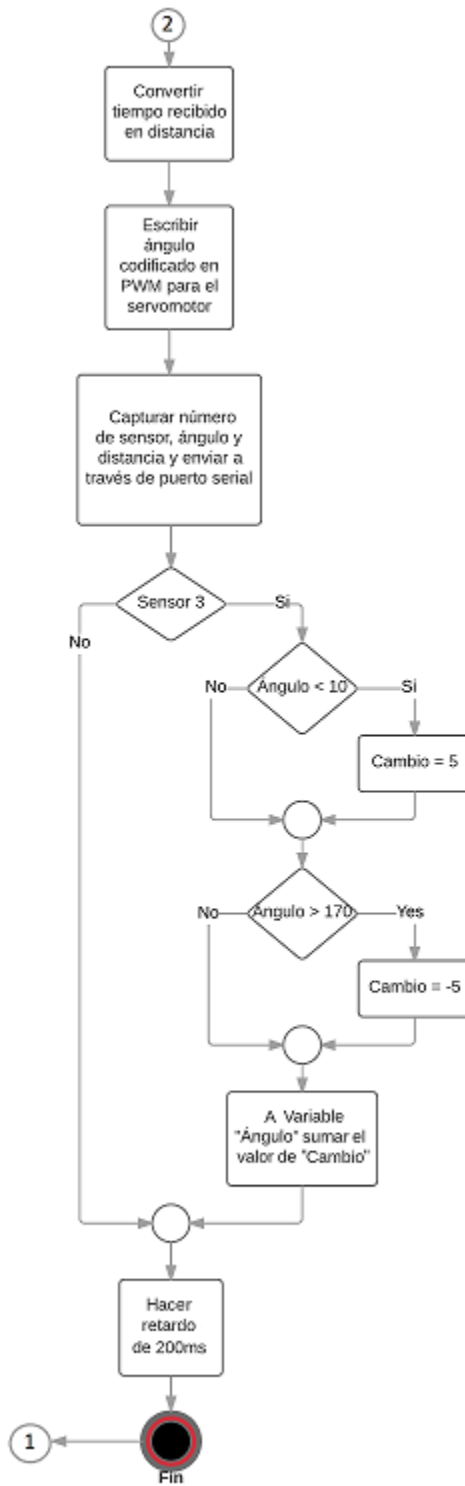


Diagrama 2: control de sistema de detección con sensores de ultrasonido, parte 2

11.2 Anexo 2

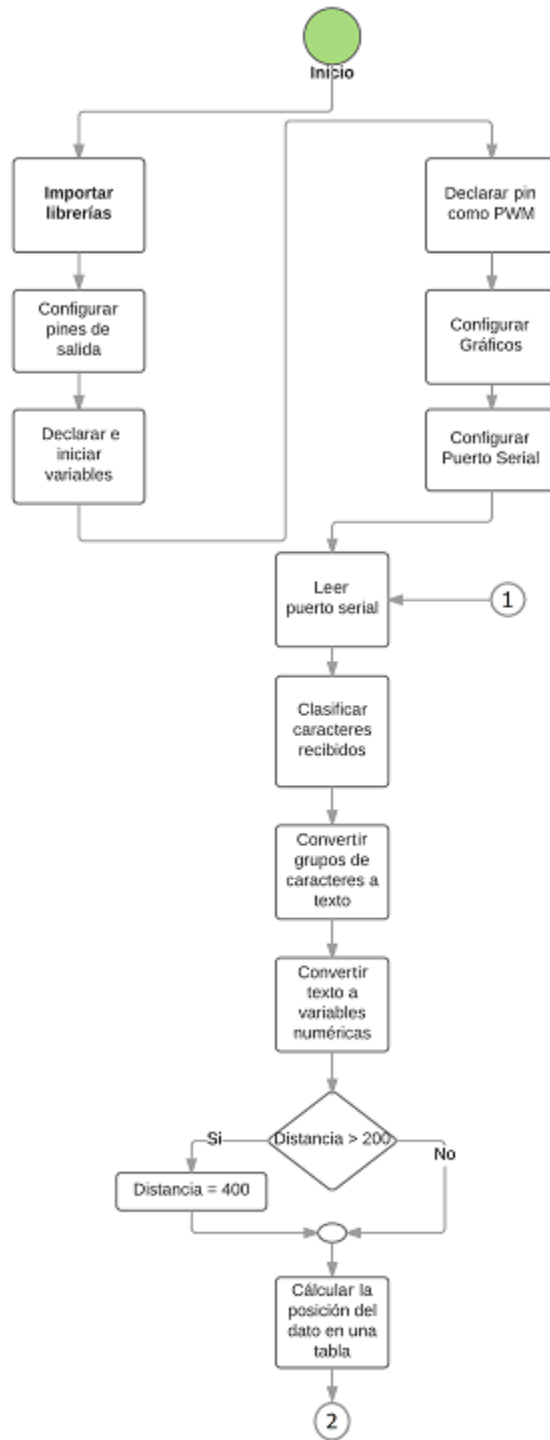


Diagrama 3: Diagrama de flujo de programa en

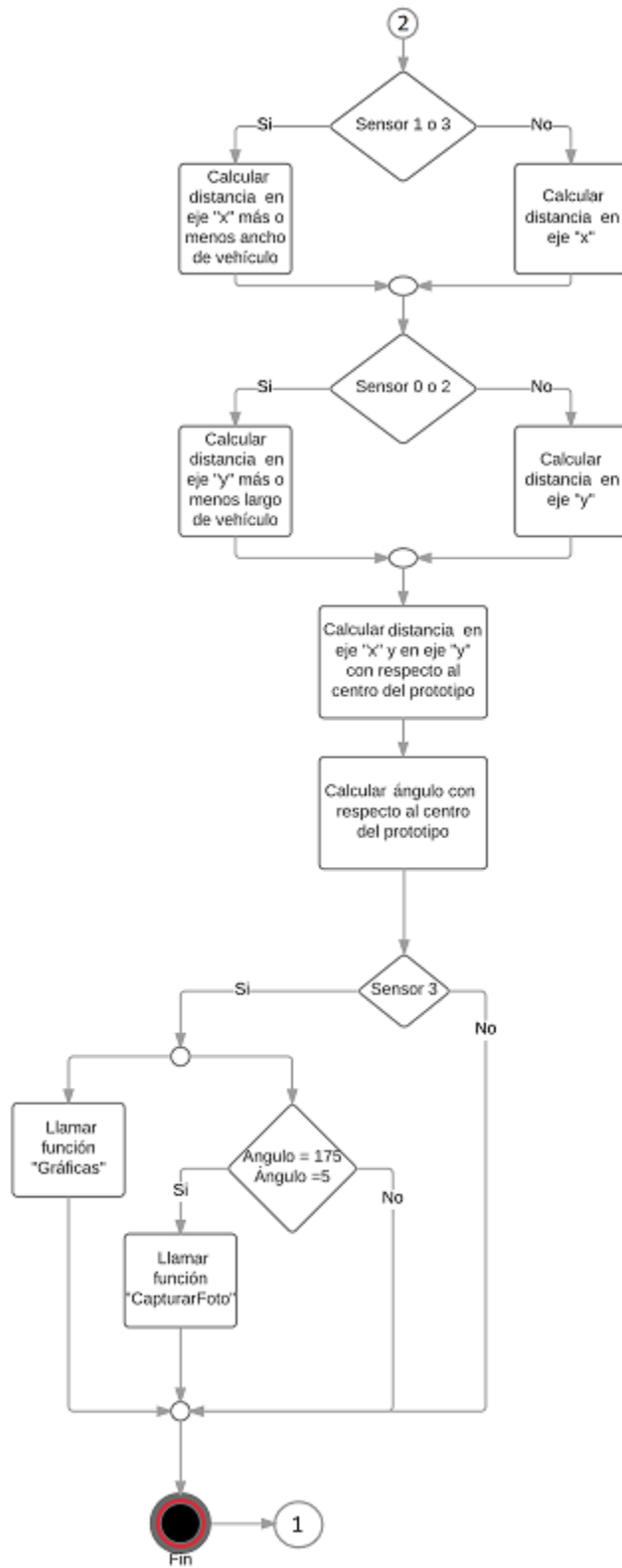


Diagrama 4: Diagrama de flujo de programa en Raspberry Pi,

11.3 Anexo 3

FUNCIÓN CAPTURAR FOTO

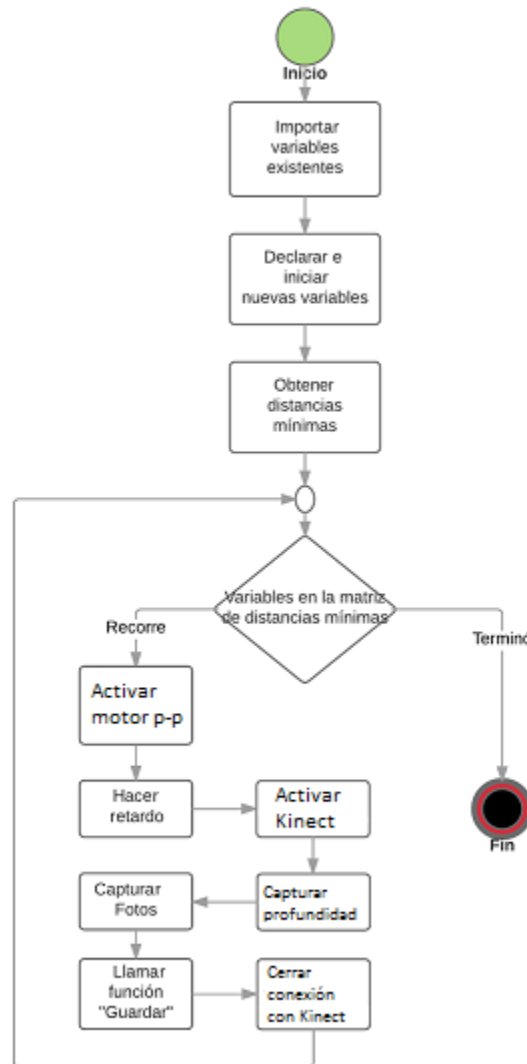


Diagrama 5: Diagrama de flujo de programa para captura de fotos en Raspberry Pi

11.4 Anexo 4

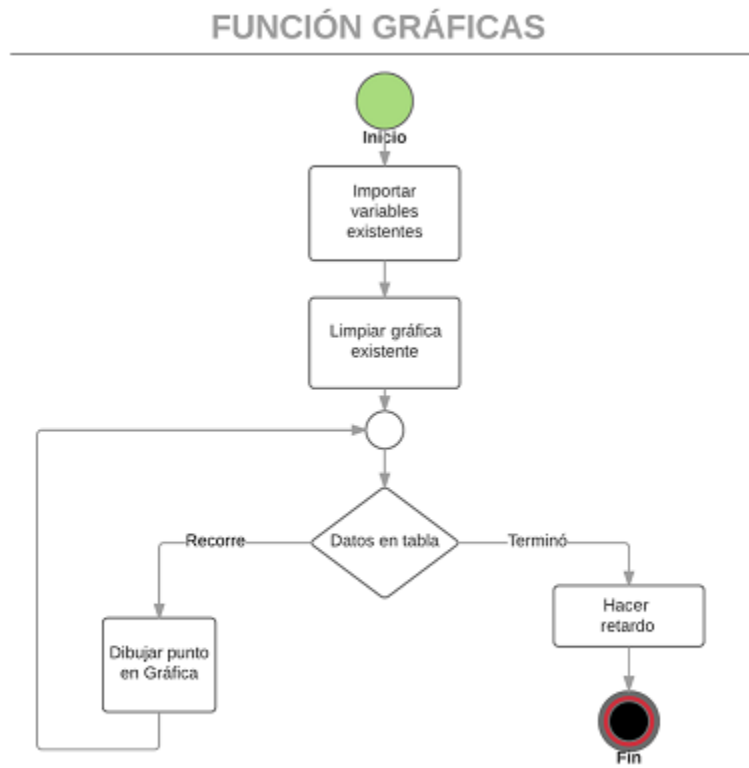


Diagrama 6: Diagrama de flujo del programa de gráficas en Raspberry Pi

11.5 Anexo 5

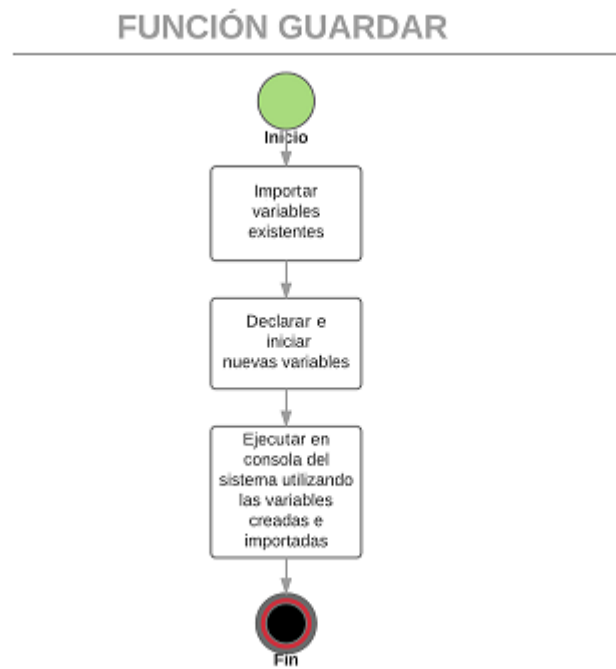


Diagrama 7: Diagrama de flujo del programa de guardado y envío de datos en Raspberry Pi

11.6 Anexo 6

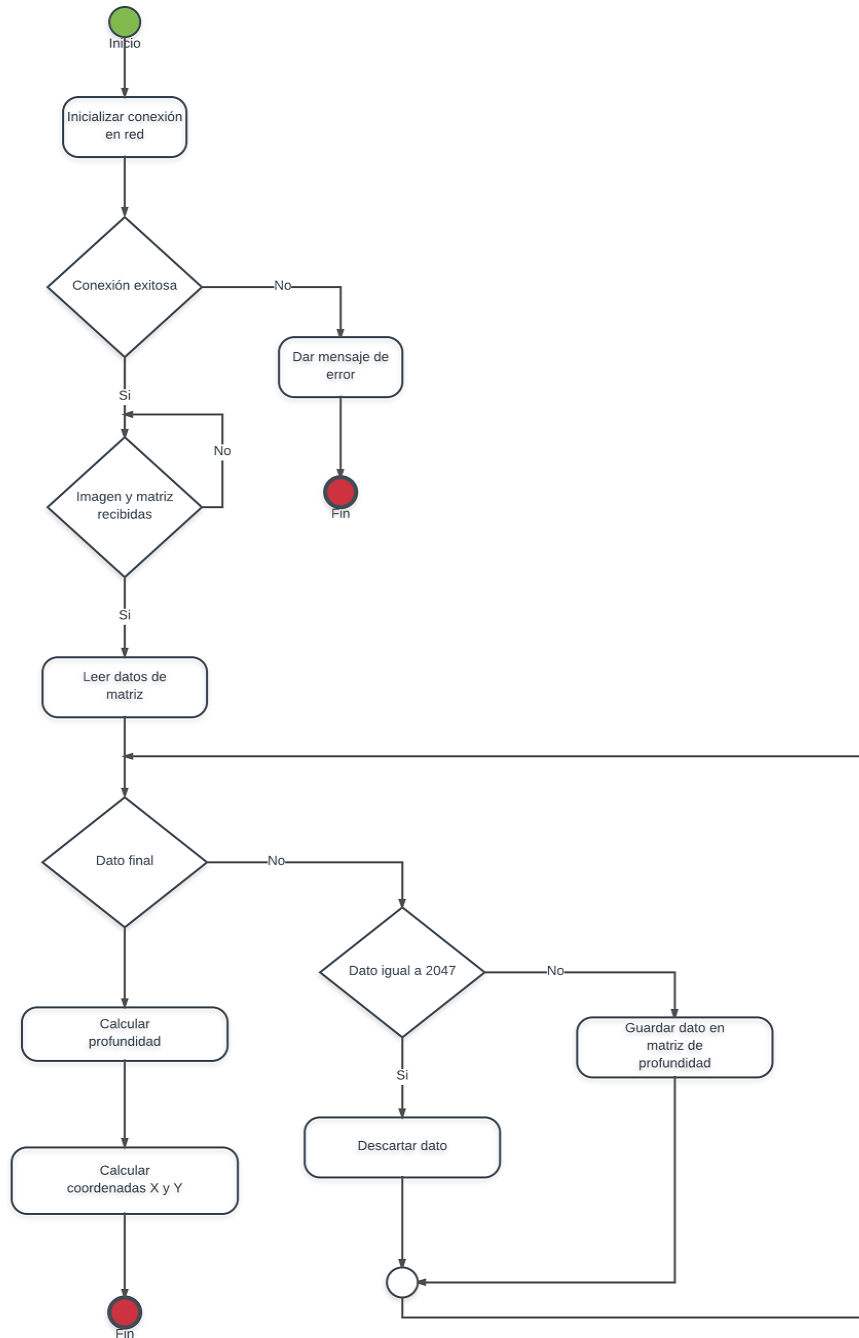


Diagrama 8: Diagrama de flujo del procesamiento de datos del Kinect