

Proyecto Fin de Máster  
Máster Universitario en Ingeniería Industrial

Aplicación del sensor LiDAR en la medición de las  
variables zoométricas en caballos

Autor: Diego Tarrat Martín

Tutor: Sergio Toral Marín

Tutor externo: Manuel Pérez Ruiz

**Dpto. de Ingeniería Electrónica**  
**Dpto. de Ingeniería Aeroespacial y Mecánica de Fluidos**

**Escuela Técnica Superior de Ingeniería**

**Universidad de Sevilla**

Sevilla, 2020





Proyecto Fin de Máster  
Máster Universitario en Ingeniería Industrial

# **Aplicación del sensor LiDAR en la medición de las variables zoométricas en caballos**

Autor:

Diego Tarrat Martín

Tutor:

Sergio Toral Marín

Profesor titular

Tutor externo:

Manuel Pérez Ruiz

Profesor titular

Dpto. de Ingeniería Electrónica  
Dpto. de Ingeniería Aeroespacial y Mecánica de Fluidos  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla  
Sevilla, 2020



Proyecto Fin de Carrera: Aplicación del sensor LiDAR en la medición de las variables zoométricas en caballos

Autor: Diego Tarrat Martín

Tutor: Sergio Toral Marín

Tutor externo: Manuel Pérez Ruiz

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

*A mi familia, en especial a mis  
padres que creyeron en mí.*

*A mis tutores.*

*A mis compañeros.*



# Agradecimientos

---

En primer lugar, quería agradecer a mis padres por sus sabios consejos, su comprensión y su apoyo incondicional durante estos años de estudio.

Me gustaría dar las gracias a mi tutor externo D. Manuel Pérez Ruiz por su valiosa ayuda. Definitivamente me ha brindado todas las herramientas necesarias para completar mi trabajo de fin de máster con éxito.

Además, quería agradecer a mi tutor D. Sergio Toral Marín por su disposición de ayudarme a poder presentar mi trabajo de fin de máster satisfactoriamente.

Finalmente, a mis compañeros Jorge Martínez Guanter, Enrique Apolo Apolo y Fabio Tucci Álvarez. No sólo por haber estado ahí apoyándome en los momentos difíciles, sino también por los diferentes momentos compartidos dentro y fuera de la universidad.

*Diego Tarrat Martín*

*Sevilla, 2020*



# Resumen

---

La medida de la dimensión corporal de ganado de gran tamaño juega un papel significativo en la mejora de la calidad y el mejoramiento genético. En este contexto, las mediciones obtenidas sin contacto por sistemas de detección visualizado por ordenador podrían representar un gran avance ante el método tradicional que precisa de mucha mano obra y provoca situaciones de estrés a los animales. En este trabajo se presenta un nuevo enfoque para la medida del tamaño del cuerpo del animal basado el modelado tridimensional digital. La captura de la serie de datos se lleva a cabo por un sensor LiDAR de Velodyne.



# Abstract

---

The measurement of the body size of large cattle plays a significant role in quality improvement and genetic improvement. In this context, the measurements obtained without contact by detection systems visualized by computer could represent a great advance before the traditional method that requires a lot of workmanship and causes stress situations to the animals. This paper presents a new approach to measure the size of the animal's body based on digital three-dimensional modeling. The capture of the data series is carried out by a Velodyne's LiDAR sensor.

# Índice

---

<b>Agradecimientos</b>	<b>ix</b>
<b>Resumen</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Índice</b>	<b>xiv</b>
<b>Índice de Tablas</b>	<b>xvi</b>
<b>Índice de Figuras</b>	<b>xviii</b>
<b>Notación</b>	<b>xxii</b>
<b>1 Introducción</b>	<b>1</b>
1.1. <i>Antecedentes</i>	1
1.2. <i>Ojetivos y alcance del trabajo</i>	2
1.2.1. <i>Objetivos generales</i>	2
1.2.2. <i>Objetivos específicos</i>	2
<b>2 Introducción a la tecnología LiDAR</b>	<b>3</b>
2.1. <i>Datos LiDAR</i>	3
2.2. <i>Devolución láser de LiDAR</i>	4
2.3. <i>Atributos del punto LiDAR</i>	4
2.4. <i>Nube de puntos</i>	5
<b>3 Materiales y métodos</b>	<b>6</b>
3.1. <i>Velodyne LiDAR Puck VLP-16</i>	6
3.1.1. <i>Caja Interfaz</i>	7
3.1.2. <i>Reflectividad calibrada</i>	8
3.1.3. <i>Modo de retorno de señales</i>	8
3.1.4. <i>SLAM con Velodyne VLP-16</i>	10
3.1.5. <i>Paquete de datos de VLP-16</i>	10
3.2. <i>Raspberry Pi 3 Model B+</i>	12
3.3. <i>OS</i>	13
3.3.1. <i>ROS</i>	14
3.3.2. <i>Estructura básica</i>	15
3.3.3. <i>RViz</i>	22
3.3.4. <i>Velodyne_pointcloud</i>	23
3.3.5. <i>Velodyne_node</i>	23
3.4. <i>Lugar del ensayo</i>	24
3.5. <i>Toma de medidas de forma manual</i>	25
3.5.1. <i>Método de medición manual</i>	29

<i>3.6. Toma de medidas con el sensor LiDAR</i>	<i>33</i>
3.6.1. Configuración de la Raspberry Pi para comunicarse con el Velodyne	33
3.6.2. Instalación de ROS	35
3.6.3. Instalación de dependencias de ROS	36
3.6.4. Instalación del controlador VLP-16	36
3.6.5. Visualización de la nube de puntos	36
3.6.6. Procesamiento de la nube de puntos	39
3.6.7. Método de medición con el sensor LiDAR	41
<b>4 Resultados</b>	<b>43</b>
<b>5 Conclusión</b>	<b>50</b>
<b>Anexo</b>	<b>51</b>
<b>Referencias</b>	<b>55</b>

# ÍNDICE DE TABLAS

---

Tabla 2–1. Atributos LiDAR	4
Tabla 3–1 Conexiones de la caja interfaz	7
Tabla 3–2. Velocidad de los datos del retorno dual	10
Tabla 3–3. Tipo de mensaje	16
Tabla 3–4. Definición de Constantes en Mensajes	16
Tabla 4-1. Medidas zoométricas del caballo ENCANTADO 49 realizadas manualmente y mediante el sensor	49
Tabla A-1. Medidas zoométricas del caballo FESTEJADA 10 realizadas manualmente y mediante el sensor	51
Tabla A-2. Medidas zoométricas del caballo VALIOSA 8 realizadas manualmente y mediante el sensor	52
Tabla A-3. Medidas zoométricas del caballo FESTEJADO 14 realizadas manualmente y mediante el sensor	53
Tabla A-4. Medidas zoométricas del caballo REVOLTOSO 85 realizadas manualmente y mediante el sensor	54



# ÍNDICE DE FIGURAS

---

Figura 2-1. LiDAR aerotransportado	3
Figura 2-2. Retornos LiDAR	4
Figura 2-3. Nube de puntos del sensor LiDAR Velodyne VLP-16	5
Figura 3-1. Sensor LiDAR Velodyne VLP-16	6
Figura 3-2. Visión general del sistema LIDAR VLP-16Imaging 3D	6
Figura 3-3. Caja de interfaz	7
Figura 3-4. Tipos de reflectores.	8
Figura 3-5. Modos de retorno	9
Figura 3-6. Retorno dual con sensor aerotransportado	9
Figura 3-7. Retorno dual con sensor en tierra	10
Figura 3-8. Paquete de datos del VLP-16	11
Figura 3-9. Conversión de esféricas a XYZ	11
Figura 3-10. Logo Raspberry Pi	12
Figura 3-11. Placa y conexión de Raspberry Pi Model b+	13
Figura 3-12. Interfaz gráfica de Ubuntu 18.04.2 (Bionic)	13
Figura 3-13. Relación ROS y sistema operativo	14
Figura 3-14. Comunicación del proceso LIDAR VLP-16 con el Maestro	17
Figura 3-15. Comunicación del proceso RViz con el Maestro	17
Figura 3-16. Comunicación del proceso LIDAR VLP-16 con RViz	18

Figura 3-17. Comunicación nodo a nodo por tópico y servicio (Publicación y Suscripción)	19
Figura 3-18. Localización geográfica de la Yeguada de la Cartuja – Hierro Bocado	24
Figura 3-19. Patio de sementales, lugar del ensayo	24
Figura 3-20. Bastón zoométrico de Aparicio	25
Figura 3-21. Compás de Brocas	25
Figura 3-22. Medidas zoométricas relacionadas con la calidad racial	26
Figura 3-23. Medidas zoométricas relacionadas con el rendimiento funcional	28
Figura 3-24. Medidas zoométricas relacionadas con la calidad racial y el rendimiento funcional	29
Figura 3-25. Medida Alzada a la cruz	29
Figura 3-26. Medida Altura al punto más declive de la cruz	29
Figura 3-27. Medida Alzada a las palomillas	30
Figura 3-28. Medida Longitud de la cabeza	30
Figura 3-29. Medida Anchura de la cabeza	30
Figura 3-30. Medida Longitud del cuello	30
Figura 3-31. Medida Longitud de la espalda	30
Figura 3-32. Medida Longitud del antebrazo	30
Figura 3-33. Medida Diámetro dorso esternal	31
Figura 3-34. Medida Long. Long. dorso+ lomo	31
Figura 3-35. Medida Longitud de la grupa	31
Figura 3-36. Medida Distancia nalga-babilla	31
Figura 3-37. Medida Longitud de la pierna	31
Figura 3-38. Medida Longitud de la nalga	31
Figura 3-39. Medida Ángulo de la espalda	32
Figura 3-40. Medida Ángulo de la grupa	32
Figura 3-41. Medida Anchura de pecho	32
Figura 3-42. Medida Anchura de la grupa	32
Figura 3-43. Medida Diámetro bicostal	32
Figura 3-44. Medida Perímetro del tórax	32
Figura 3-45. Medida Perímetro de la caña	33
Figura 3-46. Medida Diámetro Longitudinal	33
Figura 3-47. Conexiones del hardware	33
Figura 3-48. Configuración de la dirección IP	34
Figura 3-49. Pantalla de configuración del sensor LiDAR Velodyne	34
Figura 3-50. ROS Melodic Morenia	35
Figura 3-51. Interfaz gráfica de RViz y nube de puntos generada en el laboratorio	37
Figura 3-52. Selección de tópicos para convertir a csv	38
Figura 3-53. Mensaje de conversión finalizada	38
Figura 3-54. Matriz de datos obtenida en el ensayo del caballo ENCANTADO 49	38
Figura 3-55. Nube de puntos del ensayo del caballo ENCANTADO 49	39

Figura 3-56. Segmentación en modo de edición poligonal	40
Figura 3-57. Barra de herramientas de la Segmentación Interactiva	40
Figura 3-58. Reconstrucción tridimensional del caballo	41
Figura 3-59. Montaje del sensor LiDAR sobre el trípode	41
Figura 3-60. Etiqueta de distancia entre dos puntos	42
Figura 3-61. Etiqueta de ángulo	42
Figura 4-1. Medida zoométrica Altura al punto más declive de la cruz	43
Figura 4-2. Medida zoométrica Alzada a las palomillas	43
Figura 4-3. Medida zoométrica Longitud escápulo isquial	43
Figura 4-4. Medida zoométrica Longitud de la cabeza	44
Figura 4-5. Medida zoométrica Anchura de la cabeza	44
Figura 4-6. Medida zoométrica Longitud del cuello	44
Figura 4-7. Medida zoométrica Longitud de la espalda	45
Figura 4-8. Medida zoométrica Longitud del antebrazo	45
Figura 4-9. Medida zoométrica Diámetro dorso esternal	45
Figura 4-10. Medida zoométrica Longitud del dorso	46
Figura 4-11. Medida zoométrica Longitud del lomo	46
Figura 4-12. Medida zoométrica Longitud de la grupa	46
Figura 4-13. Medida zoométrica Distancia anca-babilla	47
Figura 4-14. Medida zoométrica Distancia nalga-babilla	47
Figura 4-15. Medida zoométrica Longitud de la pierna	47
Figura 4-16. Medida zoométrica Longitud de la nalga	48
Figura 4-17. Medida zoométrica Ángulo de la espalda	48
Figura 4-18. Medida zoométrica Ángulo de la grupa	48



# Notación

---

GdP	Ganadería de Precisión
LiDAR	Light Detection and Ranging
GPS	Global Positioning System
INS	International Navigation System
RGB	Red, Green, Blue
ASPRS	American Society for Photogrammetry and Remote Sensing
g	Gramo, unidad principal de masa del Sistema Cegesimal de Unidades
mm	Milímetro, unidad de longitud del sistema métrico
W	Vatio, unidad derivada coherente del Sistema Internacional de Unidades (SI) para la potencia
V	Voltio, unidad derivada del Sistema Internacional de Unidades para el potencial eléctrico, la fuerza electromotriz y la tensión eléctrica
Hz	Hercio, unidad de frecuencia del Sistema Internacional de Unidades
m	Metro, unidad coherente de longitud del Sistema Internacional de Unidades
TVS	Transient-voltage-suppression
CC	Corriente Continua
CA	Corriente Alterna
FPGA	Field Programmable Gate Array

TTL	Transistor-Transistor Logic
PLC	Power Line Communications
ROS	Robot Operating System
GdP	Ganadería de Precisión
LiDAR	Light Detection and Ranging
GPS	Global Positioning System
INS	International Navigation System
RGB	Red, Green, Blue
ASPRS	American Society for Photogrammetry and Remote Sensing
g	Gramo, unidad principal de masa del Sistema Cegesimal de Unidades
mm	Milímetro, unidad de longitud del sistema métrico



# 1 INTRODUCCIÓN

# L

a innovación es un elemento clave para impulsar los avances, mejoras y el crecimiento sustentable para el beneficio social y económico de nuestra sociedad. El sector agrícola ha atravesado por cambios significativos a través de la historia, producto de los avances en tecnología y las tendencias del momento. Estos cambios han motivado el desarrollo de nuevos productos y procesos de producción, que suponen, a su vez, un desarrollo en la industria misma. Cuando esto sucede es que se habla de una innovación en el sector agrícola. Gracias a la tecnología y la innovación, la Agricultura y Ganadería Moderna permite aumentar la productividad de sector agrario cuidando los recursos de nuestro planeta.

Actualmente, existen numerosas tendencias en la agricultura que facilitan la optimización de los cultivos, impulsando su desarrollo, como la introducción de nuevas prácticas en los procesos de producción, en la gestión del campo, nuevas ideas para la captura de la información diaria de labores, adopción de las nuevas tecnologías (ejemplo: la tendencia actual de usar drones en el campo), entre otras muchas posibilidades, todo con base en una apropiada investigación, a modo de garantizar la obtención de buenos resultados. Con el análisis avanzado de los datos, los productores cuentan con estadísticas más certeras sobre cómo mejorar sus producciones. La agricultura de precisión planea desde hace años sobre emprendedores y agricultores. El uso de datos y tecnologías de la última generación promete cambiar el futuro del sector agrícola.

El incremento poblacional a nivel mundial y los cambios en los patrones de alimentación han hecho que el consumo de productos de origen animal vaya en aumento año tras año. Sin embargo, los consumidores demandan cada vez alimentos más saludables, que al mismo tiempo respeten cuestiones como el bienestar animal, la sostenibilidad y la trazabilidad. Ante tal escenario, la ganadería de precisión (GdP) se presenta como un nuevo sistema sostenible para producir alimento de origen animal con la ayuda de las tecnologías de la información y comunicación. Su uso permite aumentar la producción y el bienestar animal, reducir costos de inversión e impacto ambiental y resolver problemas mediante el análisis de información.[6]

En este trabajo se aborda una solución que la ganadería de precisión está aportando a las explotaciones animales, los sistemas de monitorización de ganado vivo a través de sistemas de medición y detección de animales mediante un sensor LiDAR, recogiendo datos del modelo tridimensional digital de los animales en tiempo real y sus medidas zoométricas. El animal al cual se le realizará las mediciones es el caballo procedente de la Yeguada de la Cartuja – Hierro Bocado perteneciente al municipio de Jerez de la Frontera, pero que también se puede utilizar para la medición del cuerpo de otros animals grandes

Resulta gran valor conocer estas medidas debido a que la variación en las dimensiones del cuerpo del ganado durante sus periodos de crecimiento se correlaciona con el peso corporal, la evolución en la productividad, la selección y la cría [7]. La medición periódica de las dimensiones del cuerpo del ganado se usa para evaluar la respuesta de crecimiento al suministro de nutrientes y a las anomalías de salud.

Todo ello realizado con el objetivo de obtener información relevante que ayude a los ganaderos en la gestión de sus explotaciones y en la toma de decisiones

## 1.1 Antecedentes

En el 2017, la población humana mundial sumaba más de 7.5 mil millones y se estima que para el 2030 ascenderá a 8.5 mil millones. Esto incrementará la demanda mundial de alimentos y servicios básicos. La producción agropecuaria deberá satisfacer la creciente demanda de alimento y considerar que actualmente se sobreexplotan los recursos naturales (por ejemplo, biodiversidad y agua). El cambio climático podría reducir hasta en 25% la producción agropecuaria dado que el calentamiento global puede incrementar la morbilidad y mortalidad del ganado [6]. La producción pecuaria mundial tendrá que satisfacer la demanda creciente de alimento.

Mientras que en la ganadería tradicional frecuentemente se toman decisiones basadas únicamente en la experiencia del productor. En la GdP éstas se toman con base en una mayor cantidad de información, la cual puede obtenerse en tiempo real. Usa tecnologías de información y comunicación para la recolección/análisis de datos y sistemas de control que ayudan a reducir costos de inversión y aumentar tanto la producción como el bienestar animal.

La medición periódica de las variables zoométricas se mide mensualmente colocando al ganado necesita ser un cepo, lo cual resulta estresante para el ganado, así como mucha mano de obra e intensiva para el agricultor. Por tanto, existe una necesidad de automatizar el proceso de medición del cuerpo, así como hacer que sea más fácil para medir los tamaños de cuerpo siempre que sea necesario.

Existen métodos a partir de imágenes digitales, pero presentan ciertas limitaciones cuando existen cambios en la iluminación o en el fondo, provocando “ruido” que afecta a la precisión de la toma de medidas y puede conducir a errores en los resultados estimados [7]. Sin embargo la aplicación de los sensores LiDAR se ofrece una alternativa a las imágenes digitales, ofreciendo una solución ante las limitaciones mencionadas anteriormente proporcionando modelos digitales tridimensionales de los objetos a partir de imágenes de nubes de puntos en 3D fácilmente manejables.

En este trabajo se presenta un enfoque de medición de las dimensiones del cuerpo del ganado vivo sin contacto para los caballos de la Yeguada de la Cartuja.

## 1.2 Objetivos y alcance del trabajo

El objetivo principal del proyecto es:

*Determinar las variables zoométricas de varios caballos mediante su reconstrucción en 3D mediante el sensor LiDAR Velodyne VLP-16.*

Para alcanzar este objetivo principal se han fijado una serie de objetivos generales y específicos:

### 1.2.1 Objetivos generales

- Llevar a cabo el proyecto utilizando los conocimientos proporcionados durante la fase de estudio.
- Tener conocimiento de las técnicas y herramientas disponibles para la realización del proyecto así como también su uso en cada una de las fases de este.
- Obtener un documento final adecuado para el entendimiento del proyecto ya sea por parte de alguien especializado o cualquier persona ajena al ámbito en el que se centra.

### 1.2.2 Objetivos específicos

- Conocer la tecnología LiDAR y en concreto del LiDAR
- Comprender el funcionamiento del LiDAR Veldoyne VLP-16 para un posterior entendimiento de los resultados obtenidos
- Sincronizar el sensor LiDAR con la Raspberry Pi 3 Model B+ y obtener la nube de puntos del caballo.
- Filtrar la nube de puntos a través del software CloudCompare para obtener de una forma más óptima las variables zoométricas exigidas.

## 2 ESTRUCTURA DEL TRABAJO

**L**iDAR (del inglés Light Detection and Ranging) es una técnica de teledetección óptica que utiliza la luz de láser para obtener una muestra densa de la superficie de la tierra produciendo mediciones exactas de x, y y z. LiDAR, que se utiliza principalmente en aplicaciones de representación cartográfica láser aéreas, está surgiendo como una alternativa rentable para las técnicas de topografía tradicionales como una fotogrametría. Su funcionamiento utiliza los principios del radar, sustituyendo las ondas electromagnéticas por señales ópticas con mayor ancho de espectro. Esta técnica se ha llevado a desde la invención del láser en 1960.

### 2.1 Datos LiDAR

El lidar es un sensor óptico activo que transmite rayos láser hacia un objetivo. El reflejo del láser del objetivo lo detectan y analizan los receptores en el sensor lidar. Estos receptores conformados por un telescopio receptor, filtros y un detector óptico registran el tiempo preciso desde que el pulso láser dejó el sistema hasta cuando regresó para calcular la distancia límite entre el sensor y el objetivo. Combinado con la información posicional (GPS e INS), estas medidas de distancia se transforman en medidas de puntos tridimensionales reales del objetivo reflector en el espacio del objeto. La señal es transmitida al procesador de señales donde es digitalizada y analizada.

Para obtener la nube de puntos del terreno mediante escáner láser aerotransportado, el módulo debe realizar dos movimientos combinados, una barrida longitudinal y otra transversal mediante un espejo móvil que desvía el haz de luz láser emitido por el escáner. Para obtener las coordenadas x, y y z georeferenciadas con alta precisión de la nube de puntos se necesita analizar el rango de tiempo láser, la posición del sensor y ángulo de escaneo láser en cada momento, para esto el sistema se suele apoyar en un GPS diferencial y un sensor inercial de navegación (INS). El resultado son datasets de nube de puntos masivos por segundo que se pueden administrar, visualizar, analizar y compartir.

Los componentes de hardware principales de un sistema lidar incluyen un vehículo de recolección (avión, helicóptero, vehículo y trípode), sistema de escáner láser, GPS (Sistema de posicionamiento global) e INS (sistema de navegación por inercia). Un sistema INS mide la rotación, inclinación y encabecamiento del sistema lidar.

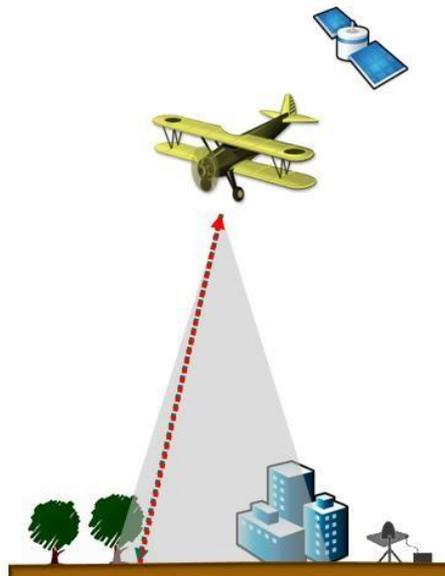


Figura 2-1. LiDAR aerotransportado [5]

## 2.2 Devolución láser de LiDAR

Los pulsos láser emitidos desde un sistema lidar se reflejan desde objetos sobre y por encima de la superficie del suelo: vegetación, edificios, puentes y así sucesivamente. Un pulso láser emitido puede regresar al sensor lidar como uno o muchas devoluciones. Cualquier pulso láser emitido que encuentre varias superficies de reflejo a medida que viaja hacia el suelo se divide en tantas devoluciones como superficies reflectoras existen.

El primer pulso láser devuelto es el más importante y se asociará con la entidad más grande en el panorama como una copa de árbol o la parte superior de un edificio. La primera devolución también puede representar el suelo, en cuyo caso el sistema lidar solo detectará un regreso.

Varias devoluciones pueden detectar las elevaciones de varios objetos dentro de la huella láser de un pulso láser saliente. Las devoluciones intermedias, en general, se utilizan para la estructura de la vegetación, y la última devolución para los modelos de terreno de suelo desnudo.

La última devolución no siempre será de una devolución del suelo. Por ejemplo, considere un caso en donde un pulso golpee una rama gruesa en su camino hacia el suelo y el pulso no llega en realidad al suelo. En este caso, la última devolución no es desde el suelo, pero sino desde la rama que reflejó el pulso láser completo.

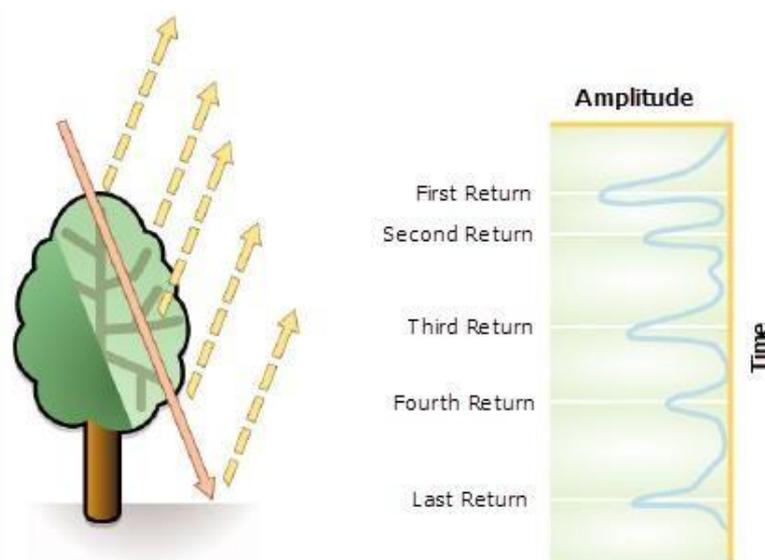


Figura 2-2. Retornos LiDAR [5]

## 2.3 Atributos del punto LiDAR

La información adicional se almacena junto con cada valor posicional x, y, y z. Los siguientes atributos del punto lidar se mantienen para cada pulso láser registrado: intensidad, número de devolución, cantidad de devoluciones, valores de clasificación de punto, puntos que están en el borde de la línea de vuelo, valores RGB (rojo, verde y azul), tiempo del GPS, ángulo de escaneo y dirección de escaneo. La siguiente tabla describe los atributos que se pueden proporcionar con cada punto lidar.

Tabla 2-1. Atributos LiDAR [5]

Atributo LiDAR	Descripción
<b>Intensidad</b>	La fortaleza de la devolución del pulso láser que genero el punto LIDAR.
<b>Número de devolución</b>	Un pulso láser emitido puede tener hasta cinco devoluciones dependiendo de las entidades en las que se refleja y las capacidades del escáner láser que se utiliza para recopilar los datos. La primera devolución se etiquetará como devolución número uno, la segunda como devolución número dos, etc.
<b>Número de devoluciones</b>	El número de devoluciones es el número total de devoluciones por cada pulso.

<b>Clasificación de puntos</b>	Cada punto LIDAR que es postprocesado puede tener una clasificación que define el tipo de objeto que reflejó el pulso láser. Los puntos LIDAR se pueden clasificar en varias categorías que incluyen suelo o terreno desnudo, parte superior de cubierta forestal y agua. Las diversas clases se definen mediante códigos numéricos de enteros en el archivo LAS.
<b>Borde de la línea de vuelo</b>	Los puntos se simbolizarán en base a un valor de 0 o 1. A los puntos etiquetados en el borde de la línea de vuelo se les proporcionará un valor de 1 y a todos los otros puntos se les proporcionará un valor de 0.
<b>RGB</b>	Los datos lidar se pueden atribuir con bandas RGB (rojas, verdes y azules). Esta atribución con frecuencia viene de las imágenes recopiladas al mismo tiempo que la topografía lidar.
<b>Tiempo del GPS</b>	La fecha de registro del GPS en la que se emitió el punto láser desde el avión. El tiempo está en segundos del GPS de la semana.
<b>Ángulo de escaneo</b>	El ángulo de escaneo es un valor en grados entre -90 y +90. A 0 grados, el pulso láser está directamente debajo del avión en nadir. A -90 grados, el pulso láser está en el lado izquierdo del avión, mientras que a +90, el pulso láser está a la derecha del avión en la dirección de la luz. La mayoría de sistemas lidar actualmente tienen menos de $\pm 30$ grados.
<b>Dirección de escaneo</b>	La dirección de escaneo es la dirección en la que estaba viajando el espejo de escaneo en el momento del pulso láser de salida. Un valor de 1 es una dirección de escaneo positiva, y un valor de 0 es una dirección de escaneo negativa. Un valor positivo indica que el escáner se está moviendo del lado izquierdo al lado derecho de la dirección de vuelo en pista y un valor negativo es el opuesto.

---

## 2.4 Nube de puntos

Los datos LiDAR organizados espacialmente postprocesados se conocen como datos de la nube de punto. Se trata de un conjunto de datos de gran tamaño integrado por datos de puntos 3D. Las entidades de superficie específicas que el láser encuentra se clasifican después de que la nube de punto LIDAR inicial es postprocesada. Los escáneres láser aéreos LiDAR son los instrumentos de uso más frecuente para la recopilación de datos de nube de puntos geográficos. Los objetos físicos como exteriores e interiores de edificios, plantas de procesamiento, topografías o elementos fabricados, y todo lo demás que el rayo láser encuentra durante la encuesta constituye los datos de nube de punto. [5]

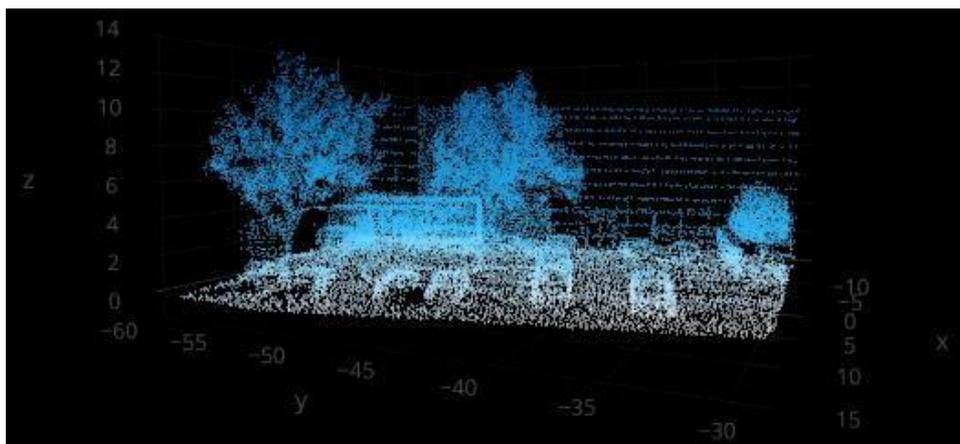


Figura 2-3. Nube de puntos del sensor LiDAR Velodyne VLP-16

Cada punto LiDAR puede tener una clasificación asignada que define el tipo de objeto que reflejó el pulso láser. Los puntos LiDAR se pueden clasificar en varias categorías que incluyen suelo o terreno desnudo, parte superior de cubierta forestal y agua. Las diversas clases se definen mediante códigos numéricos de enteros en el archivo LAS. Estos códigos los definió la Sociedad Americana de Fotogrametría y Teledetección (ASPRS).

## 3 MATERIALES Y MÉTODOS

### 3.1 Velodyne LiDAR Puck VLP-16

El Velodyne VLP-16 es un sensor láser 3D de alta precisión de tipo LiDAR con una cabeza giratoria que contiene un determinado número de láseres semiconductores o diodos láseres. Cada uno de los láseres tiene su propio detector. Se trata del sensor más pequeño (ver Figura 3-1) y avanzado de la gama de productos de 3D LiDAR de Velodyne. Tiene un peso ligero de apenas de 830 g (sin cableado ni caja de interfaz) y un tamaño compacto y reducido (~Ø103 mm x 72 mm).



Figura 3-1. Sensor LiDAR Velodyne VLP-16 [1]

La capacidad de retorno dual lo hacen ideal no solo para vehículos autónomos pero también para aplicaciones robóticas y cartografía móvil terrestre 3D, adaptable a cualquier vehículo. El Velodyne no tiene partes giratorias visibles, lo que lo hace altamente resistente en entornos difíciles (Clasificación IP67) mientras opera en un amplio rango de temperatura (-10°C a + 60°C).

Las características más importantes son:

- 16 canales de adquisición
- El alcance de obtención de puntos es de 100 metros, aproximadamente 300.000 por segundo en retorno sencillo y 600.000 para retorno dual.
- Campo de visión horizontal de 360°
- Campo de visión vertical de 30° con ±15° arriba y abajo (necesita que se tenga visión en todas la direcciones de manera simultánea, procediéndose a realizar el montaje del mismo en una parte libre de obstáculos)
- Frecuencia de rotación variable entre 5 y 20 Hz.
- Bajo consumo: 8 W de potencia [1]

La comunicación se realiza bajo una conexión a través de una caja diseñada para realizar la interfaz la cual se conecta con el ordenador vía Ethernet, permitiendo intercambio de información a 100Mbps entre el sensor y el ordenador.

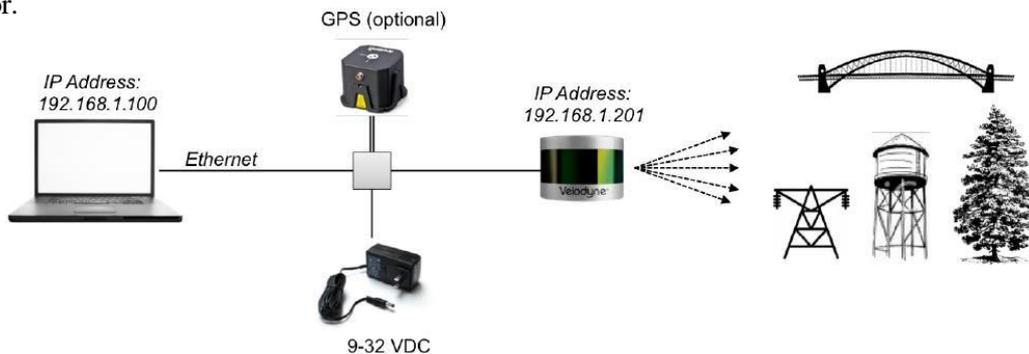


Figura 3-2. Visión general del sistema LIDAR VLP-16 Imaging 3D [1]



### 3.1.2 Reflectividad calibrada

El sensor VLP-16 mide la reflectividad de un objeto con 256 bits de resolución independiente de la potencia del láser y la distancia en un intervalo de 1 m a 100 m.

Los estándares de reflectividad comercialmente disponibles se utilizan para la calibración absoluta de la reflectividad, que se almacenan en una tabla de calibración dentro de la FPGA del VLP-16 (del inglés, Field Programmable Gate Array), dispositivo que permite la ejecución sencilla de funciones para la calibración.

Los parámetros de reflectividad para el VLP-16 tienen dos características principales:

- Los reflectores difusos reportan valores de 0-100 para reflectividad de 0% a 100% (Figuras 3-3a y 3-3b).
- Los retro-reflectores reportan valores de 101 a 255 siendo 255 la reflectividad reportada para un retro-reflector ideal y 101-254 para reflectores parcialmente obstruidos o imperfectos (Figuras 3-3c y 3-3d). [1]

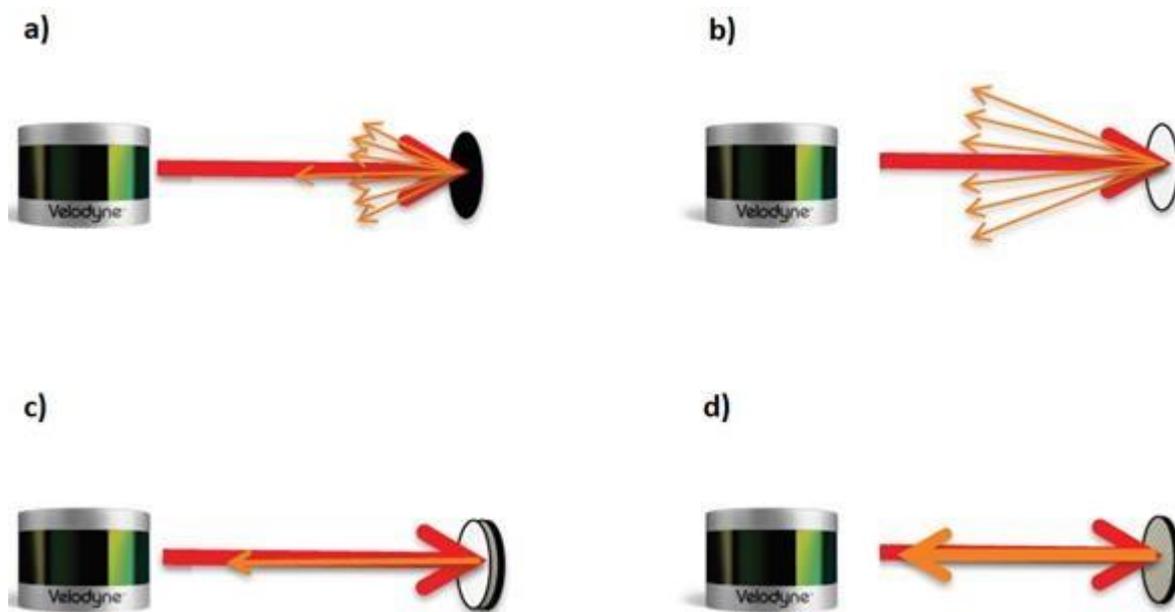


Figura 3-4. Tipos de reflectores. En (a), Negro, reflector difuso absorbente (valor 0). En (b), Blanco, reflector difuso reflectante (valor 100). En (c), Retro-reflector cubierto con una superficie semi-transparente (valor 101). En (d), Retro-reflector sin ninguna cobertura (valor 255) [1]

### 3.1.3 Modo retorno de señales

Gracias a la divergencia del haz del láser, un solo disparo golpea múltiples objetos donde se producen diferentes retornos. El VLP-16 tiene la capacidad de analizar las devoluciones múltiples y reportar el retorno más fuerte (Strongest return), el último retorno (Last return) o ambas (Dual return o dual mode).

En la siguiente figura se observa que el láser golpea la pared cercana mientras su difusión golpea la pared más alejada. Solamente se registrarán ambas lecturas o devoluciones si la distancia entre las dos paredes es mayor a un metro.

En el caso de que el retorno más fuerte sea el último retorno (segunda pared), se registrará el segundo retorno más fuerte. [1]

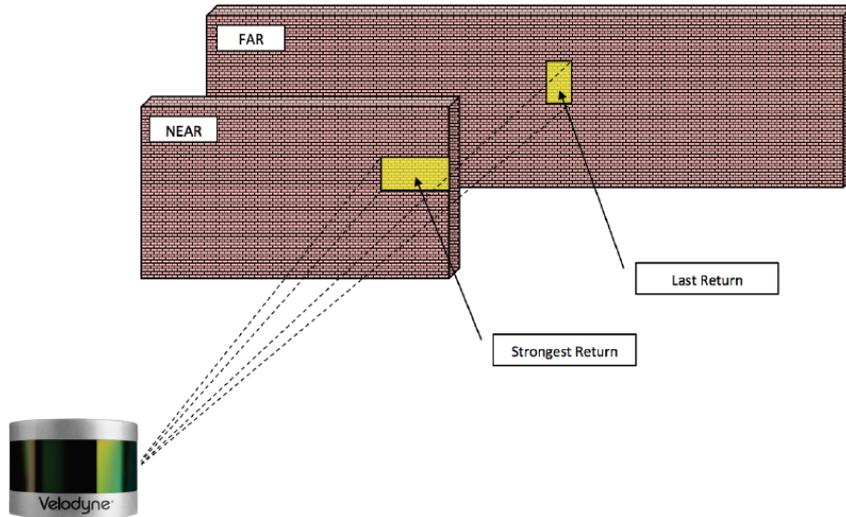


Figura 3-5. Modos de retorno [1]

La función de retorno dual se utiliza a menudo en aplicaciones forestales, donde se necesita determinar la altura de los árboles o también para determinar la altura de edificios de poca altura si se requiere saber el tamaño de estas edificaciones [1]. La figura de a continuación (el haz láser se representa mediante la línea azul) ilustra lo que ocurre cuando el punto de láser golpea la parte exterior del árbol, penetra en las hojas y ramas, y, finalmente, llega al suelo, registrándose diferentes amplitudes para el último retorno y el retorno más fuerte.

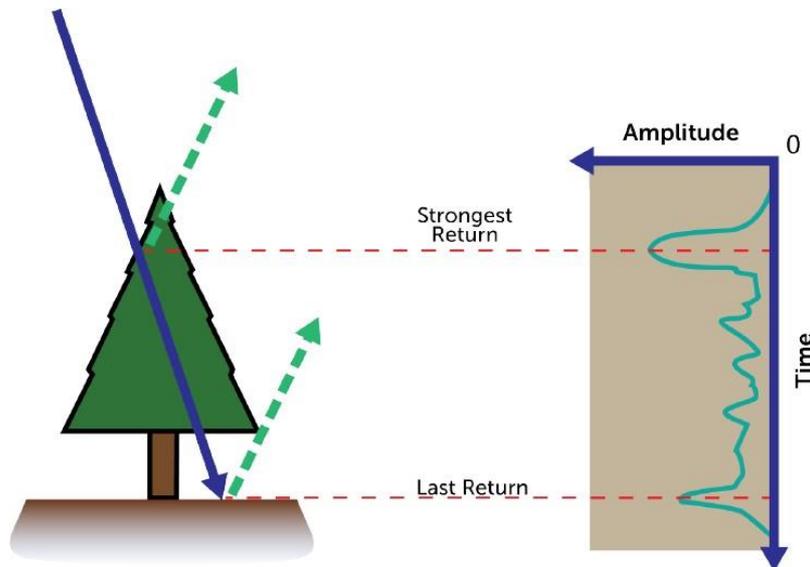


Figura 3-6. Retorno dual con sensor aerotransportado [1]

También se puede utilizar en entornos donde el vehículo que transporta el sensor se encuentre sobre tierra y el haz del láser impacte contra un vidrio o algún elemento que le permita tener un retorno dual. En esta figura siguiente se observa que el retorno más fuerte identificado de color azul es el primer retorno que se refleja contra un cristal, dejando que ciertos puntos de lectura del laser superen este obstáculo y se vean reflejados dentro del edificio en una superficie sólida identificada con las líneas de color rojo.

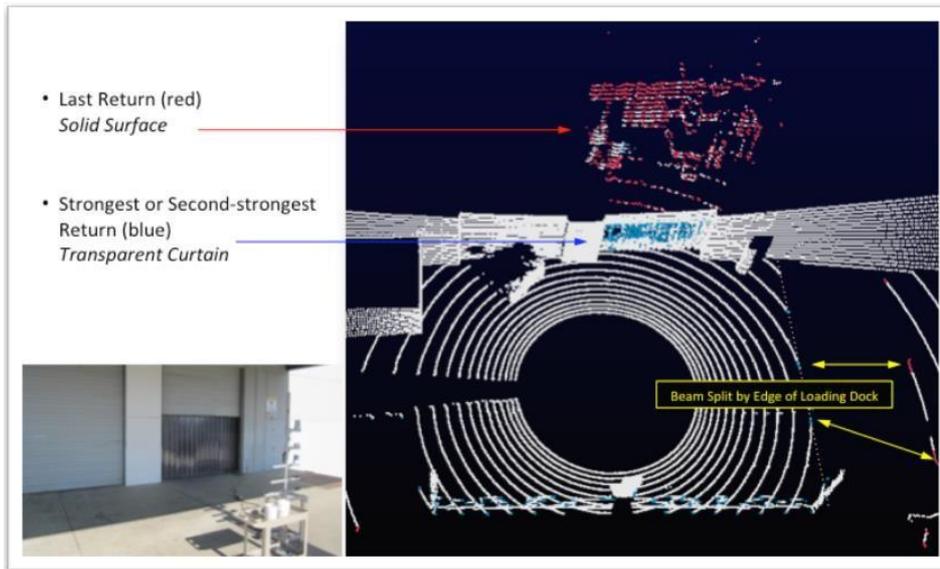


Figura 3-7. Retorno dual con sensor en tierra [1]

En el modo de retorno dual, la velocidad de datos del sensor se duplica. Las velocidades de datos para los dos modos se dan a continuación:

Tabla 3–2. Velocidad de los datos del retorno dual [1]

Modo	Paquetes / segundo	Megabits / segundo
Más fuerte	754	~8
Último	754	~8
Dual	1508	~16

### 3.1.4 SLAM con Velodyne VLP-16

Realizar SLAM está formado por varias partes como el reconocimiento de puntos de referencia del entorno; como la lectura y asociación de los datos, la estimación, actualización de la posición y de los puntos de referencia. En la actualidad existen muchos algoritmos diseñados para cumplir con estas funciones de mapeo y localización, por lo que pueden variar las formas de adquisición y tratamiento de datos de las diferentes partes del proceso. [8]

Para realizar estos reconocimientos de puntos de referencia trabajando con un Velodyne, se debe tener en cuenta la información o tópicos que existen del mismo. Dentro de las especificaciones dadas para ROS y los tópicos que se pueden obtener del dispositivo es que se tiene un paquete de información, llamado *velodyne\_packets*, el cual posee varios nodos que brindan información y a los cuales se debe prestar atención, debido a que es la lectura que realiza el láser. Principalmente lo que interesa es cómo se está reconociendo el entorno y el cambio de posición con respecto al origen. Esto se realiza a través de la odometría y de un tópico llamado nube de puntos, también conocido como PointCloud o PCL que se describirán más adelante.[9]

Cabe destacar que aunque no es objetivo de este trabajo la construcción de mapas de un entorno desconocido y la estimación de la posición dentro del mapa, sería de gran ayuda utilizar esta técnica para mejorar la toma de medidas zoométricas en los caballos.

### 3.1.5 Paquete de datos de VLP-16

El paquete de datos consiste básicamente en un bloque de referencia (Figura 3-8). El paquete de datos es 1248 bytes de longitud y se envía en el puerto 2368. La Figura 3-8 que contiene la información que brinda la lectura de cada haz de láser emitido, que al retornar es organizada en bloques que contienen la siguiente estructura:

- 42 bytes de encabezado
- 1200 bytes de datos divididos en:
  - ↓ 12 canales (bloques) de 100 bytes de datos que dentro de cada uno hay:
    - 2 bytes de identificación de inicio (0xFFEE) O (0xFFEE)
    - 2 bytes de azimut
    - 32x3 bytes de recolección de datos
  - ↓ 4 bytes de bloque de tiempo
  - ↓ 2 bytes de fábrica

Los bytes de fábrica contienen información del modelo de Velodyne con el que se está trabajando y el tipo de retorno de información que tiene (Strongest, Last o Dual). [1]

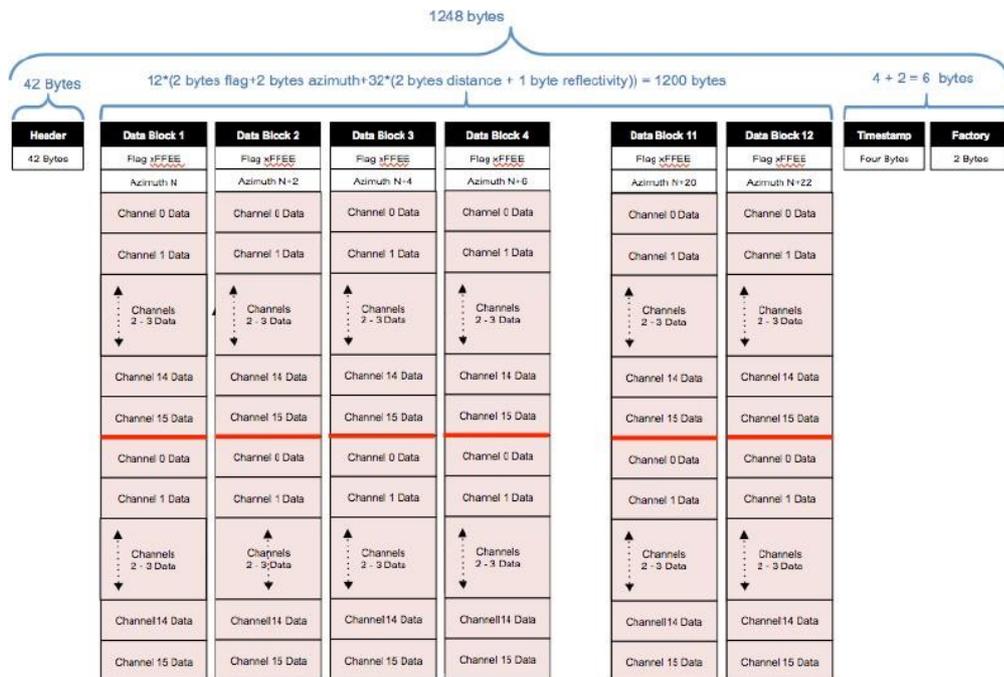


Figura 3-8. Paquete de datos del VLP-16 [1]

El dispositivo reporta coordenadas esféricas en coordenadas  $(R, \omega, \alpha)$ , para que con cálculos matemáticos convertir esta información a coordenadas rectangulares  $(x, y, z)$ . Sabiendo que  $\omega$  es el ángulo de elevación,  $\alpha$  es el ángulo de azimuth y  $R$  la distancia medida por el láser hasta el punto adquirido. Los cálculos se realizan de manera automática en el algoritmo de lectura *velodyne\_points*, como se muestra en la Figura 3-9.

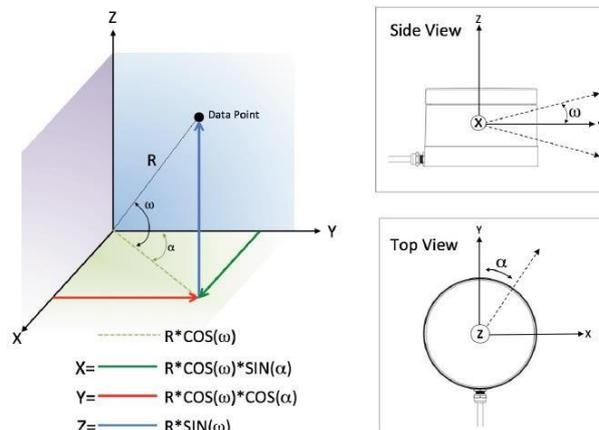


Figura 3-9. Conversión de esféricas a XYZ [1]

### 3.2 Raspberry Pi 3 Model B+

Un mini PC no tiene una definición única, ya que dependiendo de la fuente que se consulte, engloba un mayor o menor número de equipos. Como características comunes tenemos que todos son capaces de ejecutar diversos SO sin problemas, en un formato y tamaño mucho menor que un PC de escritorio y a un coste reducido.

Existe una gran variedad de mini PC, muchos de ellos son productos acabados y para un fin determinado, como pueden ser los smart TV o el intel nuc (equipo de sobremesa pequeño y compacto). Por otro lado, se pueden encontrar una gran variedad de mini PC en formato placa base.

Ante la necesidad de seleccionar un sistema empotrado capaz de soportar ROS con una bajo coste, bajo consumo y fácil aprendizaje, la familia de sistemas empotrados Raspberry Pi es una de las mejores elecciones.

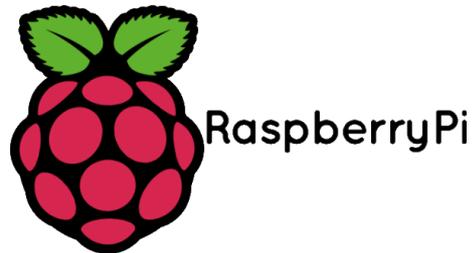


Figura 3-10. Logo Raspberry Pi [2]

Raspberry Pi es un ordenador de placa reducida (SBC) de bajo coste y pequeño tamaño, desarrollado en Reino Unido por la fundación Raspberry Pi, con el objetivo principal de estimular la enseñanza de la informática en las escuelas.

El modelo usado en este trabajo ha sido la Raspberry Pi 3 Model B+. Todas las especificaciones técnicas de este modelo de Raspberry pi 3 son:

- CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- RAM: 1GB LPDDR2 SDRAM
- Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE
- Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)
- GPIO de 40 pines
- HDMI
- 4 puertos USB 2.0
- Puerto CSI para conectar una cámara.
- Puerto DSI para conectar una pantalla táctil
- Salida de audio estéreo y vídeo compuesto
- Micro-SD
- Power-over-Ethernet (PoE) [2]

Los elementos y pines que componen este modelo se muestran en la siguiente figura:

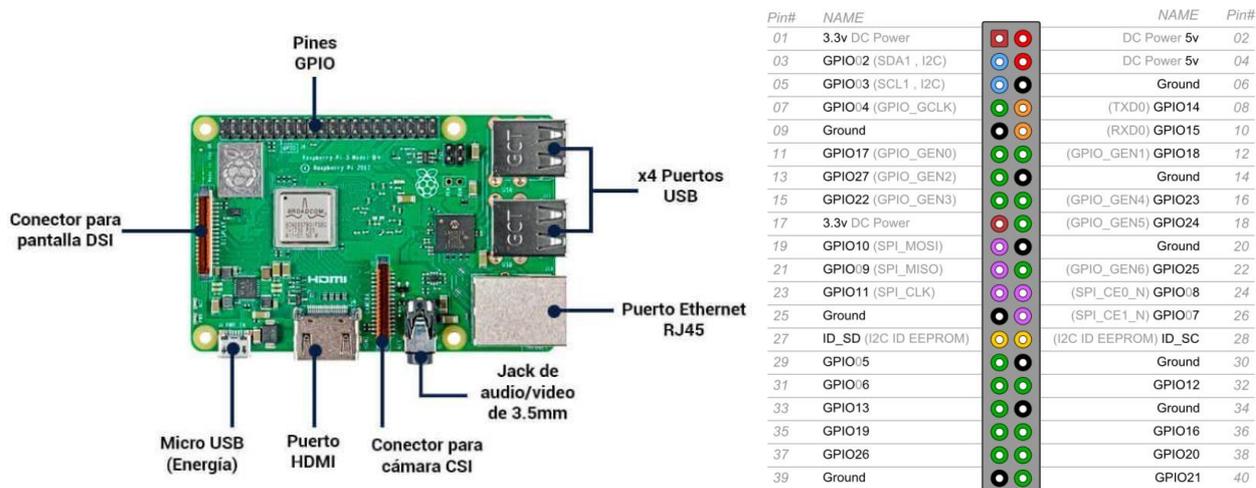


Figura 3-11. Placa y conexionado de Raspberry Pi Model b+ [3]

Como se puede observar en la Figura 4-2 la Raspberry Pi Model B+ cuenta con dos filas de 20 pines cada una. Estos pines sirven de interfaz entre la Raspberry Pi y el exterior. GPIO (General Purpose Input/Output) es un sistema de Entrada/Salida de propósito general que permite dar un uso más amplio, controlando multitud de dispositivos como luces leds, motores o relés. De los 40 pines que contiene la Raspberry Pi, 12 de ellos son de alimentación (5V o 3.3V) o tierra. El resto de pines son GPIO normales y pines GPIO especiales para usar como puerto UART, I2C o SPI

La Raspberry Pi es un sistema muy versátil que permite la instalación de varios sistemas operativos ya sean estos oficiales o no, así como distintos gestores de contenidos en función del uso al que vaya enfocado.

### 3.3 OS

El OS (Sistema Operativo) utilizado es UBUNTU 18.04.2 (Bionic). La razón por la se ha elegido Ubuntu es por que es el sistema operativo abierto o de software libre preferido para desarrollo bajo Linux, especialmente en lo que se refiere a soluciones de inteligencia artificial, aprendizaje automático y aprendizaje profundo, debido a que ofrece la profundidad y amplitud de las bibliotecas, tutoriales y ejemplos de Ubuntu, así como ningún otro sistema operativo ofrece el mismo nivel de soporte para las versiones más recientes de software y plataformas gratuitas de código abierto. Asimismo, con Ubuntu es posible desarrollar con el mismo sistema tanto en el escritorio como en el servidor, en la nube y en dispositivos IoT. A todo esto se une el hecho de que se trata de una de las plataformas más óptimas para que el sistema operativo de robótica ROS funcione de manera correcta y sencilla.



Figura 3-12. Interfaz gráfica de Ubuntu 18.04.2 (Bionic)

### 3.3.1 ROS

ROS (Robot Operating System) es un framework (entorno de trabajo) flexible, con una amplia variedad de herramientas, librerías y paquetes que simplifican considerablemente la creación de aplicaciones para robots. ROS se puede definir de igual manera como un metasisistema operativo, ya que es un OS que se instala sobre otro. Lo más usual es que sea un sistema UNIX (Ubuntu, Debian). Además de proveer funcionalidades de un OS en un cluster (muchos ordenadores conectados entre sí).

Fue creado en el año 2007 bajo el nombre de Switchyard (Subestación), por el Laboratorio de Inteligencia Artificial de Stanford para dar soporte al proyecto del Robot con Inteligencia Artificial de Stanford (STAIR). A partir de 2008, el desarrollo continúa primordialmente en Willow Garage, un instituto de investigación robótico con más de veinte instituciones colaborando en un modelo de desarrollo federado.[10]

Desde su creación, ROS se ha diseñado para facilitar el intercambio de software entre los aficionados y los profesionales de la robótica en todo el mundo debido a su enfoque didáctico y abierto, lo que ha permitido la construcción de una gran comunidad de colaboradores a lo largo de todo el mundo.

Como se ha comentado antes, a pesar de su nombre, ROS no es un sistema operativo propiamente dicho, de hecho, funciona sobre Ubuntu. Lo que hace realmente es proveer los servicios estándar de un sistema operativo, tales como abstracción del hardware, control de dispositivos de bajo nivel, implementación de funcionalidad de uso común, paso de mensajes entre procesos y mantenimiento de paquetes, dejando las labores de procesamiento, manejo de memoria, gestión de interfaces gráficas, etc. a Ubuntu. En la Figura 4-2 puede observarse un esquema de la separación entre los servicios proporcionados por Ubuntu y por ROS.

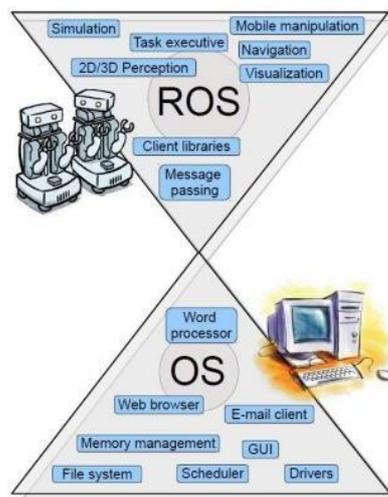


Figura 3-13. Relación ROS y sistema operativo [4]

Trabajar con el sistema operativo Ubuntu, acompañado de ROS nos permite trabajar con una interfaz gráfica simple como lo es RVIZ y obteniendo información a través del terminal con comandos de ejecución, configuraciones del sensor, configuración de la interfaz, tiempo de ejecución, entre otras funciones.

ROS es un software libre bajo términos de Licencia BSD [10] que se especializa en el desarrollo de robótica de manera sencilla y consta principalmente de paquetes, bloques, tópicos y mensajes que son vistos como nodos sencillos. Consta básicamente de publicadores de tópicos de información y a su vez suscriptores a los mismos (Publishers y Suscribers). Los publicadores (por ejemplo, los dispositivos de percepción que se utilizan) transmiten los datos obtenidos, dejando la información disponible para su obtención por otros algoritmos que requieran esta información para realizar alguna tarea, como son los llamamos suscriptores, que son nodos capaces de leer datos y manipularlos, pero nunca modificarlos para crear una interfaz óptima y sencilla con el usuario. Para que el ordenador o sensor reciban ordenes del usuario, se debe crear una interfaz en la cual el usuario sea el publicador y el dispositivo sea el suscriptor de la información.

ROS no solo consta del “sistema operativo” que se ha mencionado si no también de una lista de paquetes (ros-pkg) que son aportados por los usuarios, “que implementan las funcionalidades tales como localización y mapeo simultáneo, planificación, percepción, simulación, etc.” [10]

En primer lugar, se realiza la configuración del sensor y se obtiene una nube de puntos en la cual se pueden apreciar tanto el mapeo del espacio como la localización del Lidar dentro del mismo (en nuestro caso el VLP-16) si se sincronizara el sensor con el GPS. En este caso, esta nube de puntos, al ir variando la ubicación del hardware, va realizando un mapeo continuo bajo una gama de colores previamente configurada, dando conocimiento de las distancias que tenemos entre los diferentes obstáculos y el vehículo.

### 3.3.2 Estructura básica

ROS tiene tres niveles de conceptos: el nivel de archivos del sistema, el nivel de Computación Gráfica y el nivel comunitario. Estos niveles y conceptos se resumen a continuación junto con los diferentes nombres que pueden tener los recursos de ROS.

#### 1. Nivel de archivos de sistema

- a) **PAQUETES:** ROS a nivel de software se organiza en paquetes, los cuales pueden contener nodos, una biblioteca independiente, conjunto de datos, archivos de configuración, softwares para otras aplicaciones o cualquier otro archivo que lógicamente complete el uso lógico de un módulo específico. El objetivo principal de la organización por paquetes es que la útil funcionalidad de fácil acceso para que el software se pueda utilizar de una forma sencilla. Los paquetes son fáciles de crear y cuentan con herramientas útiles para su manipulación. Los paquetes de ROS son simplemente un directorio que nace de la variable principal de entorno que es llamada ROS PACKAGE PATH que contiene un archivo "package.xml" en él. [1]

1) **Archivos y directorios comunes:** Los paquetes de ROS tienden a seguir una estructura básica de los cuales los siguientes son algunos directorios y archivos más notables.

- *include/package\_name:* Incluye las cabeceras de los archivos de los paquetes que nos permiten leer los archivos.
- *msg/:* Contiene los tipos de mensajes utilizados por el paquete.
- *src/package\_name:* incluye los archivos de origen que son necesarios para la ejecución del algoritmo.
- *srv/:* contiene los archivos de servicio del paquete.
- *scripts/:* Secuencias de comandos ejecutables.
- *CMakeLists.txt:* es un archivo creado por y para la construcción de los archivos.
- *package.xml:* es el archivo que define el paquete y contiene los permisos de acceso del paquete descrito o que se está utilizando.
- *CHANGELOG.rst:* es un archivo que es definido por el paquete el cual permite que el paquete sea leído de forma binaria.

2) **Herramientas de la línea de comandos:** Los paquetes se definen como la forma más básica y central de la organización de los archivos en ROS, por ende, existen herramientas para administrarlos y manipularlos, entre ellas podemos encontrar:

- *rospack:* Permite tener acceso para encontrar y recuperar información acerca de los paquetes.
- *catkin create pkg:* permite la creación de paquetes nuevos.
- *catkin make:* construye el espacio de trabajo para el paquete deseado.
- *rosdep:* instala el sistema de dependencias del paquete, configurando las dependencias del mismo con las dependencias que contenga nuestro ROS PACKAGEPATH.
- *rqt:* rqt es una herramienta bastante amplia para la visualización gráfica de la información de ROS, dentro de él existe un plugin llamado "Introspection/PackageGraph", que permite la visualización del paquete como un gráfico completo, donde se ve la información más relevante del mismo.

- b) **METAPAOQUETES:** Son paquetes especializados que juntan un grupo de paquetes relacionados y los representan. Los Megapaquetes son utilizados regularmente como un marcador de posición compatible con versiones anteriores de paquetes convertidos por `roscbuildStacks`.
- c) **MANIFIESTOS DE PAQUETES:** Los manifiestos incluyen información acerca de los paquetes: nombre, versión, descripción, información de licencia, dependencias, como por ejemplo, los paquetes exportados.
- d) **REPOSITARIOS:** Son una colección de paquetes que comparten un sistema VCS (sistema de control de versiones, del inglés *Version Control System*) común, que se refiere a que pueden ser liberados junto con la misma función de automatización de liberación. Pueden ser varios paquetes o uno solo. Regularmente estos repositorios se asignan a pilas de `roscbuild` para facilitar su acceso.
- e) **TIPOS DE MENSAJE:** ROS utiliza un lenguaje simplificado para la descripción de mensajes y a su vez para la descripción de los valores de la data que los nodos publican. Esta descripción es guardada en archivos de tipo `.msg` en el subdirectorio `msg/` del paquete que se este ejecutando.

Existen dos partes dentro de un archivo de mensaje, los campos y las constantes. Los campos son descritos como la data que es enviada dentro del mensaje y las constantes definen valores que pueden ser usadas para interpretar estos campos.

Los tipos de mensaje se refieren a la utilización de los nombres de recursos del paquete.

Por ejemplo, un archivo `geometry_msgs/msg/Twist.msg` se conoce normalmente como `geometry_msgs/Twist` para la simplificación de su comprensión.

- 1) **Líneas de comando:** `roscmsg` es una línea de comando que imprime la denición del mensaje, la información y puede ubicar que archivos utilizan este tipo de mensaje.
- 2) **Descripción del mensaje específico:** los formatos de descripción del mensaje son sencillos. Simplemente constan de una descripción del mensaje en una lista de las descripciones de los campos de datos y definiciones constantes en líneas separadas.

Cada campo consite en un tipo y nombre de mensaje separados por un espacio, por ejemplo:

Tabla 3–3. Tipo de mensaje

Tipo de campo	Nombre de mensaje
fieldname1	fieldtype1
fieldname1	fieldtype2

- 3) **Constantes:** Cada definición de constante es como una descripción del campo, exceptuando que se le asigna un valor. Este valor es indicado por el signo “=”, como se puede observar en las siguientes tablas:

Tabla 3–4. Definición de Constantes en Mensajes

Tipo de campo	Nombre y valor de mensaje
constanttype1	CONSTANTNAME1=constantvalue1
constanttype2	CONSTANTNAME2=constantvalue2

- f) **TIPOS DE SERVICIO:** Las descripciones de los servicios de ROS son guardados en `my_package/srv/MyServiceType.srv`. Definen la petición de estructuras de datos y respuesta de los servicios de ROS.
  - 1) **Líneas de comando:** `roscsrv` imprime las descripciones de los servicios, paquetes que contengan archivos `.srv`, y pueden encontrar archivos fuente que son usados para los tipos deservicio.
  - 2) **Descripción de Especificación de Servicios:** una descripción de un archivo de servicio consiste básicamente en una solicitud, una respuesta de un tipo de mensaje separado por ' '.

Cualquier archivo *.msg* puede ser concatenado con otro con un comando `'_'`.

2. Nivel de Computación Gráfica: La computación gráfica de ROS es la red de procesos que procesa la data en conjunto, es decir, como una red entre iguales o red de pares, P2P (del inglés *peer-to-peer*). Los conceptos básicos de la computación gráfica de ROS que proporciona datos a la gráfica de maneras distintas son los siguientes:

- a) **NODOS:** Los nodos son los procesos principales realizados para la ejecución de los algoritmos. ROS está diseñado para ser modular en una escala de “*fine-grained*” lo que quiere decir que cada proceso lo coloca en un nodo para realizar el intercambio de información. El sistema de control de un robot normalmente comprende muchos nodos, que pueden ser para controlar las ruedas, los sensores que pueda tener, la inclinación de algún brazo articulado, entre otras cosas. Los nodos están escritos con el uso de las bibliotecas de ROS, bien sea en lenguaje C++ (*roscpp*) o Python (*Rospy*).
- b) **MAESTRO:** Es el responsable de proporcionar el registro de los nombres y buscar el resto de la computación gráfica. Se encarga de que todos los nodos creados por los procesos puedan tener comunicación entre ellos. Sin éste, los nodos no serían capaces de encontrarse unos con otros, ni intercambiar mensajes o invocar servicios.

En las figuras siguientes se puede apreciar el proceso a partir del cual se realizan las conexiones nodo a nodo a través del Maestro la comunicación. El proceso se inicia cuando el primer nodo del dispositivo llamado LIDAR VLP-16 solicita enviar información al Maestro como se indica en la Figura 3-14.

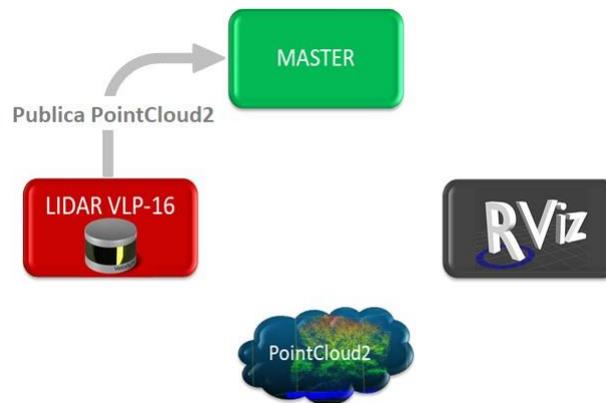


Figura 3-14. Comunicación del proceso LIDAR VLP-16 con el Maestro

A continuación, el proceso se inicia por la herramienta RViz, que solicita al Maestro tener acceso a esta información publicada, mientras que el proceso LIDAR VLP-16, hace que sus mensajes sean públicos para cualquier suscriptor que lo solicite (Figura 3-15).

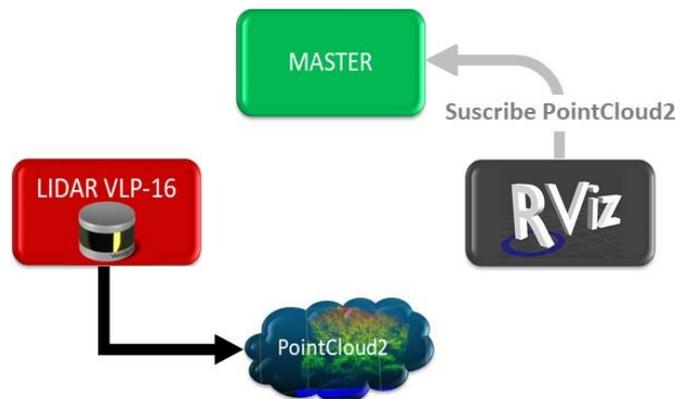


Figura 3-15. Comunicación del proceso RViz con el Maestro

Después de que el proceso RViz tenga permisos para poder acceder a estos mensajes se conecta directamente al mensaje que se requiere, como se muestra en la Figura 3-16.

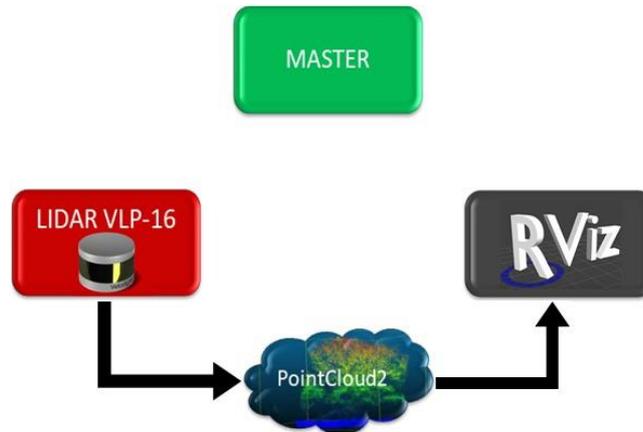


Figura 3-16. Comunicación del proceso LIDAR VLP-16 con RViz

Este proceso descrito anteriormente se realiza para todos los nodos de información que el algoritmo solicite realizar, por lo que se hace una red de información necesaria para el intercambio de datos y una ejecución óptima de todos los algoritmos.

- c) **SERVIDOR DE PARÁMETROS:** El servicio de parámetros es una parte del Maestro que permite que los datos sean almacenados en una localización central. Es regularmente utilizada para guardar parámetros de configuración debido a que no está diseñada para alto rendimiento si no para información estática. Esto quiere decir que es globalmente observable para que las herramientas puedan fácilmente inspeccionar los estados de configuración del sistema y modificarlo en caso de que sea necesario.
- d) **TIPOS DE PARÁMETROS:** Para realizar operaciones con los parámetros, tenemos el comando *rosparam* que permite consultar y establecer parámetros en el servidor usando sintaxis YAML.
- e) **MENSAJES:** Los nodos se comunican entre ellos al transmitir mensajes. Estos son simplemente una estructura de datos que comprende los campos escritos. Están soportados por los tipos clásicos de información (entero, punto flotante, booleano, etc.), así como matrices. Se almacenan en el subdirectorio *msg* de un paquete. Para tener acceso y manipulación de los datos descritos por los mensajes, tenemos un comando llamado **rosmmsg**, que nos permite visualizar la estructura de cierto tipo de *.msg*.
- f) **TÓPICOS:** Los mensajes son enrutados a través de un sistema de transporte con semántica de *publicación/suscripción*. en donde un nodo envía un mensaje por publicación a un tópico determinado. El tópico es un nombre que es utilizado para identificar el contenido del mensaje. Un nodo que está interesado en cierto tipo de datos realiza una suscripción al tema apropiado. Pueden existir varios publicadores y suscriptores concurrentes para un solo tema y un solo nodo que puede publicar y/o suscribirse a varios temas. La herramienta *rostopic*, es un comando que nos permite interactuar con los tópicos, tanto para saber de qué tipo es, como para tener acceso a la información que publica o introducir alguna información que queremos que sea leída por otro tópico conectado a la publicación de éste.
- g) **SERVICIOS:** El modelo de peer-to-peer de publicador/suscriptor es un paradigma de comunicación muy flexible, pero su teoría de comunicación de tópicos a tópicos al mismo tiempo en un solo sentido no es el más apropiado para la interacción de solicitud y respuesta, que usualmente es requerido por la distribución del sistema.

Los servicios son los encargados de realizar la petición de información que se está manipulando. También la respuesta a las solicitudes por los diferentes nodos en un par de estructuras respectivamente.

El nodo que provee la información ofrece un servicio bajo un nombre y el cliente usa ese servicio enviando un mensaje de solicitud y esperando la respuesta.

Las herramientas de servicio son dos:

- 1) **rossrv:** muestra la información acerca de la estructura de datos del servicio solicitado (*.srv*)
- 2) **rosservice:** muestra la lista y las consultas de los servicios de ROS.

- h) **BAGS:** El *.bag* es un formato para guardar y reproducir datos de mensajes de ROS. Es uno de los mecanismos más importantes para almacenar datos de lectura de sensores, que puede ser difícil de

recopilar pero son necesarios para desarrollar y probar algoritmos.

Las herramientas principales de los bags son:

- 1) **roscap:** es una consola unificada para realizar la grabación, reproducción y otras operaciones a los archivos *.bag*.
- 2) **rqt\_bag:** Es una herramienta de visualización gráfica de los datos de la información del archivo *.bag*.
- 3) **rostopic:** Se trata de una herramienta de acceso a la información de los *.bag*, únicamente por las opciones *echo* y *list*, debido a que son más enfocadas a tópicos que a este tipo de archivos.
- 4) **rosrecord:** esta herramienta permite realizar la grabación de los tópicos que están siendo ejecutados por un proceso. Se ejecuta por medio de la herramienta *roscap* bajo el comando *roscap record*.
- 5) **roscap:** esta herramienta permite realizar la reproducción de un proceso grabado anteriormente. Se ejecuta a través de la herramienta *roscap* bajo el comando *roscap play*.

ROS trabaja como un centro de nodos que almacenan tópicos y servicios de información. Estos nodos se comunican con el Maestro para transmitir los datos que contienen. Como estos nodos están comunicados con el Maestro, pueden recibir o enviar datos o mensajes a otros nodos realizando conexiones nuevas entre ellos como se considere necesario. Para que los nodos creen conexiones dinámicamente, el Maestro realiza una llamada a estos nodos cuando la información de registro cambie.

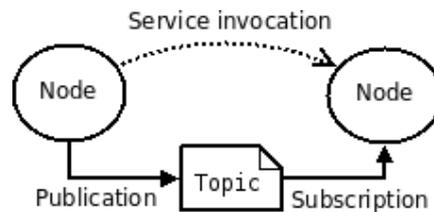


Figura 3-17. Comunicación nodo a nodo por tópico y servicio (Publicación y Suscripción)

3. Nivel comunitario: “ROS Community Level” son recursos que permiten a las diferentes comunidades intercambiar softwares y conocimientos. Dentro de estos recursos tenemos:

- a) **DISTRIBUCIONES:** Las distribuciones ROS son colecciones de paquetes y pilas versionadas que se pueden instalar. Juegan un papel bastante similar a las distribuciones de Linux facilitando la instalación de una colección de software. También mantienen las versiones actualizadas.
- b) **REPOSITORIOS:** ROS está basado en una red de repositorios de código en donde diferentes instituciones pueden desarrollar y liberar sus propios componentes de software de robot.
- c) **ROS WIKI:** La comunidad de Wiki de ROS es el foro principal para documentarse acerca de ROS. Cualquier usuario puede inscribirse en una cuenta y aportar documentación, proporcionar correcciones, actualizaciones, escribir tutoriales entre muchas otras cosas. Como es de esperarse deben estar bajo regulaciones y bajo la inspección de otros usuarios para asegurar la veracidad de las aportaciones
- d) **LISTA DE CORREOS:** Es el principal canal de comunicación entre sus usuarios, en donde se conversa acerca de las actualizaciones de ROS, así como para realizar preguntas sobre el software y fallos. Estos correos están regulados acerca la redacción de mensajes para realizar las consultas.
- e) **ROS ANSWERS:** Es un sitio de preguntas y respuestas acerca de dudas relacionadas con ROS.
- f) **BLOG:** es el lugar en donde encontramos las actualizaciones periódicas con ejemplos gráficos, fotos y vídeos.
- g)

4. Nombres:

- a) **NOMBRE DE RECURSOS GRÁFICOS:** Proporcionan una estructura jerárquica de nomenclatura utilizada para todos los recursos de computación gráfica. Cada recurso está definido dentro de un espacio de nombres, que puede ser compartido con muchos otros recursos. En general, los recursos

pueden crear otros recursos y pueden acceder a otros sobre o dentro de su propio espacio de nombres.

1) **Nombres Válidos:** Los nombres tienen ciertas restricciones y características permitidas.

- El primer carácter puede ser una letra, mayúscula o minúscula, una tilde (´) o una barra inclinada (/).
- Los siguientes caracteres pueden ser caracteres alfanuméricos, guiones bajos (\_) o barras diagonales (/).

2) **Resolving:** Existen diferentes nombres de recursos en ROS, que tienen la siguiente sintaxis

- *base:* Son una subclase de nombres relativos que tienen reglas de resolución. Son utilizados para iniciar el nombre de un nodo.
- *relative/name:* son “apuntadores” a nodos. Son nombres no reales que hacen referencia a otros nodos.
- */global/name:* son considerados como nombres totalmente resueltos. Estos nombres deben ser evitados tanto como sea posible debido a que limitan la portabilidad del código.
- *private/name:* estos tipos de nombre son realmente privados. Convierten los nombres de los nodos en nombres de espacio. Son regularmente utilizados para pasar parámetros a un nodo específico por medio del servidor de parámetros.

3) **Remapping:** Cualquier nombre de un nodo de ROS puede ser reasignado cuando el nodo es ejecutado en una línea de comando.

b) **NOMBRE DE RECURSOS DE PAQUETES:** Los nombres de recursos de paquetes se utilizan en ROS con conceptos de nivel de sistema de archivos para poder simplificar el proceso de referencia a archivos y los tipos de datos. Los nombres de los paquetes de recursos son simplemente el nombre del paquete y el nombre del recurso. Por ejemplo "*std\_msgs/String*" que se refiere al tipo de mensaje *String* en el paquete "*std\_msgs*".

1) **Nombres Válidos:** Los nombres de recursos de paquetes tienen normas estrictas en cuanto a cómo definirlos debido a que son usados muy frecuentemente en códigos autogenerados. Por esta razón los paquetes de ROS no pueden tener caracteres especiales diferentes a los guiones bajos (\_) y deben comenzar por caracteres alfabéticos. Un nombre válido consta de las siguientes características

- La primera letra debe ser un carácter alfabético sin importar si es mayúscula o minúscula.
- Los siguientes caracteres pueden ser alfanuméricos, guiones bajos (\_) o barras diagonales (/).
- Hay por lo menos una barra diagonal (/).

c) **CÓDIGOS API:** Los códigos API contienen funciones que permiten manipular los nombres de ROS. [11]

d) **CONCEPTOS DE ALTO NIVEL:** El Maestro de ROS siempre intenta ser lo más escéptico posible en cuanto a la arquitectura de sus procesos. Aunque no prescribe cómo se usan o cómo son llamados los tópicos, servicios o parámetros de estos, provee diferentes tipos de comunicación a través de ellos. Esta característica permite a ROS ser un sistema operativo de fácil manipulación sin importar el tipo de arquitectura con la que se trabaje, por lo que sus conceptos de alto nivel son necesarios para la compilación de sistemas largos y complejos.

ROS provee conceptos que proveen al alto nivel una comunicación con pilas importantes de información como son los mensajes comunes (*common msgs*), geometría de ubicación por promediación de lecturas (*geometry*) entre otros.

1) **Transformadas/marcos coordinados (*Transforms Frames, TF*) [12] y [13]:** Este concepto proporciona una librería basada en ROS, la cual permite a un marco de referencia calcular las posiciones de diferentes marcos coordinados contra el tiempo transcurrido de lectura de los sensores. La razón para la utilización de esta transformada es debido a que un sistema robótico, ya sea por coordinar diferentes partes de un robot o para realizar la ubicación del mismo en un entorno, se puede realizar a través de la coordinación de tramas que cambian con el tiempo.

Realizar estas correlaciones nos permite saber por dónde se ha movido el vehículo, qué acciones ha realizado, cuál es la posición relativa con respecto a la posición de inicio. Todas estas posiciones, al llevar a cabo un SLAM del vehículo en su entorno por medio del Velodyne VLP-16, nos puede dar desde la distancia relativa al suelo del dispositivo hasta la distancia que lo separa de un objeto predeterminado a 100 metros (alcance máximo del lidar), así como saber en qué dirección y cuánto se ha movido desde su posición inicial.

Todos estos marcos creados por ROS trabajan en sistemas distribuidos, lo que quiere decir, que va a estar disponible para cualquier componente en cualquier computadora conectada al sistema. Estas transformadas o marcos coordinados se configuraron en los algoritmos utilizados para que puedan ser escuchadas por los usuarios y así saber en que posición se encuentra el vehículo con el que se trabaja. Por otro lado, estas transformadas pueden ser transmitidas por el usuario para enviar una posición deseada en donde se quiere que el vehículo se ubique, caso que no es aplicable para este trabajo debido a que sólo se trabaja con la adquisición del entorno para realizar un SLAM.

- 2) **Tareas:** La lista de tareas para estos conceptos de ROS tienen un paquete definido en la librería de paquetes *actionlib* que definen una interfaz basada en tópicos para realizar tareas apropiadas en ROS en cuanto a los movimientos que el vehículo realiza o que se quiere que realice.

La lista de tareas *actionlib* es un paquete que proporciona herramientas para crear servidores que son capaces de ejecutar objetivos de larga duración que pueden ser cancelados con solo una petición del usuario. [14]

- 3) **Ontología del mensaje:** la pila de mensajes *common\_msgs* provee una base de mensajes para sistemas robóticos que es definida por varios tipos de mensaje.

- *actionlib\_msgs*: Mensajes descritos para la librería *actionlib*, que se ha mencionado anteriormente y que permite la interacción entre usuario y servidor.[14]
- *diagnostic\_msgs*: Este paquete de mensajes contiene diagnósticos que proveen al usuario, a través de la interfaz de información, acerca del monitoreo de los sistemas en ROS. El acceso a estos mensajes es bastante sencillo y pueden ser tanto voluntarios como automatizados por el sistema operativo. No pueden ser modificados a menos que sea extremadamente necesario. [15]
- *geometry\_msgs*: Provee un paquete para primitivas geométricas del tipo puntos, vectores y posiciones. Están diseñados para proveer tipos de datos comunes y facilitar la operabilidad del sistema. Son el tipo de mensajes mas importantes para esta investigación, debido a que tienen mucha variedad y dan la información acerca de posiciones y recorridos. [16]
- *nav\_msgs*: Definen los mensajes comunes usados para interactuar con la pila de navegación. Pueden ser tipos de mensajes, servicios y acciones. Entre los mensajes más importantes para esta investigación encontramos los mensajes *OccupancyGrid* y *Odometry*, con los cuales tenemos acceso a la información del entorno y la posición relativa del vehículo con respecto al entorno.
- *sensor\_msgs*: Este paquete define los mensajes más usados por los sensores incluyendo cámaras y láseres de escaneado.  
Un mensaje de primordial importancia es *PointCloud2* o la nube de puntos, debido a que este tipo de mensaje se utiliza para realizar tanto la lectura, reconocimiento y graficación del entorno.

- 4) **Plugins:** la librería *pluginlib* es una biblioteca del lenguaje de programación C++ para cumplir con la demanda de cargas y descargas de plugins en un paquete de ROS. Son útiles para la modificación de comportamiento de la aplicación sin necesidad de tener acceso al código fuente de la aplicación.

- 5) **Modelos de Robots:** Este paquete contiene un analizador programado en C++ para un URDF (del inglés, Unified Robot Description Format), que básicamente es un formato XML para la representación de un modelo de un robot con el cual tras un proceso de revisión puede ser compatible con todas las versiones disponibles del software. [17]

- 6) **Filtros:** La biblioteca de los filtros proporciona una interfaz para el procesamiento de datos. Para poder tener acceso a esta biblioteca es estrictamente necesario tener una vinculación con la librería *pluginlib*.
- 7) **Wiki de ROS:** dispone de varios archivos que pueden ser encontrados en [18] y que traen un resumen muy compacto de todos los comandos de ROS para ejecutar diferentes funciones y opciones.

### 3.3.3 RViz

ROS dispone de una herramienta de visualización 3D, llamada RViz, donde se pueden representar robots, nubes de puntos, datos de sensores, etc. que gracias a estar basada en una arquitectura de *plugins*, nos permite no sólo desarrollar nuestras herramientas, si no reutilizar otras ya existentes.

La forma en que funciona RViz es simplemente leyendo e interpretando los diferentes datos contenidos en mensajes de ROS. Por tanto, es necesario tener un generador externo de estos mensajes, como puede ser un robot o un sensor.

Los comandos más relevantes son:

**rosviz:** Este comando es una herramienta ejecutada por terminal para realizar grabación, reproducción e información, además de otras operaciones de este tipo de archivos, de las cuales sólo se utilizarán las nombradas.

- **Grabación:** permite grabar uno o más tópicos que se están ejecutando durante una adquisición de datos.
  - a) *rosviz record TOPIC1 [TOPIC2 TOPIC3 ...]*: este comando graba los tópicos que especiquemos en el archivo.
  - b) *rosviz record -a*: realiza la grabación de todos los tópicos que están siendo publicados al momento.
 

Para este caso, al ser una cantidad de tópicos notables y de importancia, es recomendado utilizar la opción 2.

Este proceso es de vital importancia para realizar las simulaciones, debido a que con este comando se puede lograr grabar una muestra de alguna prueba que se realice, guardando toda la información requerida para realizar análisis de mejora con posteriores algoritmos y tener puntos de comparación.
- **Reproducción:** Para realizar la lectura del contenido de uno o más archivos tipo *.bag* y reproducirlos se hace a través del comando *rosviz play...* Una vez ubicado nuestro archivo, al ejecutar este comando la reproducción comenzará de inmediato.

Tenemos diferentes opciones de reproducción de las cuales algunas serán utilizadas.

- a) *rosviz play recorded1.bag*: realiza la reproducción del archivo en tiempo real publicando todos los tópicos que fueron grabados al momento en que se tomó la muestra en el archivo *recorded1.bag*.
- b) *rosviz play -d SEC recorded1.bag*: reproduce el archivo *recorded1.bag* luego de SEC segundos que todas las llamadas a los suscriptores han sido realizadas.
- c) *rosviz play -r FACTOR recorded1.bag*: multiplica la tasa de publicación por FACTOR, es decir, aumenta la velocidad de la publicación del archivo *recorded1.bag*.

Existen varias opciones de reproducción bien sea para configurar un atraso en el inicio de la publicación, reproducir sólo un instante de tiempo deseado, entre otras, a las cuales se puede tener acceso a través de la wiki de ROS. [19]

Estos comandos de reproducción son de vital importancia para realizar las simulaciones de pruebas previamente grabadas.

Para hacer que el Velodyne funcione correctamente se debe entender y ejecutar los algoritmos de configuración y adquisición que se encuentran en la web [20]. Con los algoritmos de *velodyne\_pointcloud* y *velodyne\_node*, se controla la cantidad de datos, posición, frecuencia y puntos que transfiere el sensor. Estos algoritmos tienen sus diferencias, ventajas y desventajas, esto hace que dependiendo de nuestras necesidades podamos escoger entre uno de los algoritmos y analizar el desempeño del mismo para nuestro objetivo.

### 3.3.4 Velodyne\_pointcloud

Este algoritmo provee todos los paquetes de las nubes de puntos para los modelos de Velodyne 3D LIDARs. En este trabajo se utiliza el modelo VLP-16, que es un láser de 16 capas con un valor máximo de rotación de 1200 rpm y una tasa de publicación entre 5 y 20 Hz de los bloques de información obtenida del entorno. Consta de suscripción a tópicos de paquetes publicados por el VLP-16 y publica la información en otro tópico llamado `velodyne_points`. El tipo de mensaje se publica en "sensor\_msgs/PointCloud2", el cual brinda información en campos como intensidad y el rango de amplitud, aparte de la posición XYZ de los puntos. El primer campo nos habla del tipo de material el cual está reflejándose y el segundo nos habla de cuál es el rango que se está leyendo. [21]

### 3.3.5 Velodyne\_node

Sirve principalmente para la calibración y configuración del sensor, se puede seleccionar el modelo con el que se está trabajando, número de paquetes por minuto y revoluciones con las cuales se desea trabajar para la adquisición de datos. Es importante destacar que se debe controlar la adquisición de manera muy minuciosa debido al gran tamaño de la información que se está adquiriendo, que está por el orden de los Gb. Por tanto es importante conocer cómo es el procesador del cual disponemos y la cantidad de Memoria RAM con la cual se trabaja, para lograr una configuración adecuada. Por defecto viene configurado con un número de paquetes de 348 y 600 rpm.

Un ejemplo de configuración se describe a continuación:

```
roslaunch velodyne_driver velodyne_node model:=VLP16 npackets:=348 rpm:=600
```

En él se configura el Velodyne de 16 capas para que trabaje a 600 rpm y que brinde 348 paquetes de datos de información por minuto.

Este proceso junto con el algoritmo `velodyne_driver`, son los responsables de la ejecución efectiva de los algoritmos de funcionamiento del LIDAR a través de ROS. Sin ellos, no sería posible que el funcionamiento del dispositivo fuera útil. La ejecución de este algoritmo es primordial para este trabajo. Consta de diferentes archivos de calibración en los cuales se especifica básicamente la frecuencia de publicación que se requiere, la cantidad de datos y cuántas capas se deben habilitar, para una lectura correcta de datos. [22]

### 3.4 Lugar del ensayo

La localización del estudio a realizar está situada en la Yeguada de la Cartuja - Hierro del Bocado, la reserva de caballos de estirpe cartujana más importante del mundo. Está ubicada en la antigua dehesa de La Fuente del Suero, a tan solo 5 kilómetros de donde está emplazado el Monasterio de la Cartuja, en Jerez de la Frontera, municipio de la provincia de Cádiz, perteneciente a la Comunidad Autónoma de Andalucía, en España.



Figura 3-18. Localización geográfica de la Yeguada de la Cartuja – Hierro Bocado

La principal actividad es la salvaguarda, mejora y mantenimiento del Caballo Español de Estirpe Cartujana. Las instalaciones, propiedad del Estado español, se encuentran en la Finca Fuente del Suero, en la Carretera Medina Sidonia - El Portal km 6.5, a las afueras de Jerez de la Frontera. Disponen de unas modernas instalaciones, que incluyen patio de sementales, guarnés de lujo, museo cochera, picadero cubierto, picaderos descubiertos, nave de yeguas, parideras, clínica (sala de exploración, laboratorio y quirófano) y praderas.



Figura 3-19. Patio de sementales, lugar del ensayo

### 3.5 Toma de medidas de forma manual

Los 5 caballos seleccionados para ser escaneados han sido previamente medidos manualmente, es decir, ya se han obtenidos las variables zoométricas con herramientas manuales, las cuales son:

- **Bastón zoométrico de Aparicio:** Utilizado para medir alzadas y diámetros anatómicos.

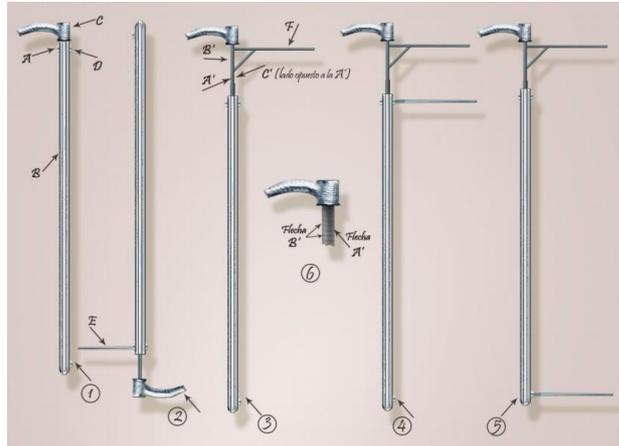


Figura 3-20. Bastón zoométrico de Aparicio

- **Cinta métrica:** Tiene menor precisión. Es útil para medir diámetros longitudinales y perímetros.
- **Compás de Brocas:** Se utiliza para medir pequeños diámetros de longitud y anchura (cabeza)



Figura 3-21. Compás de Brocas

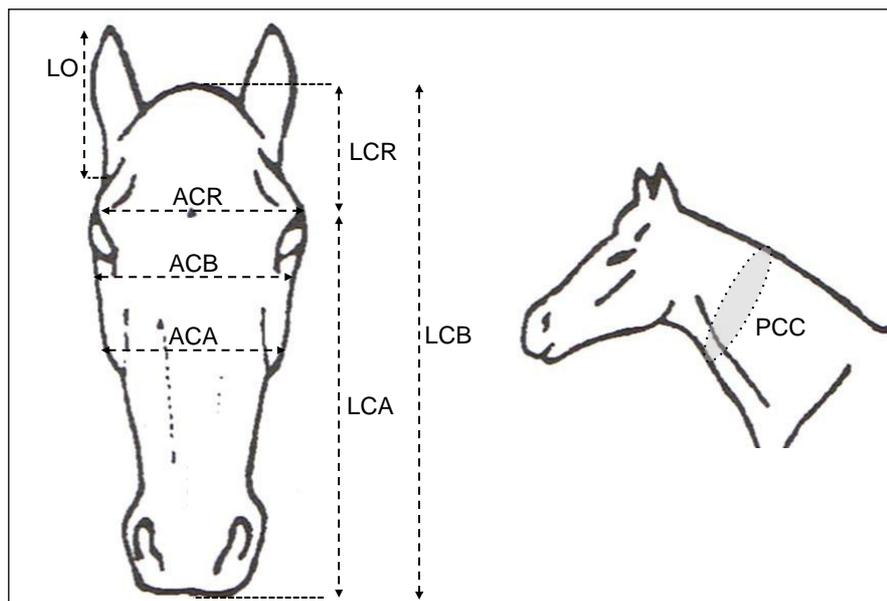
- **Calibre:** Se usa cuando para tomar medidas más precisas.

La toma de medidas de forma manual requiere una cantidad de tiempo considerable (30 min), además de personal experimentado en el uso de estas herramientas. Aunque esta labor puede realizarla una única persona, normalmente se requieren de dos personas (toma y anotación de medidas). A través del sensor láser se pretende reducir los recursos de tiempo y personal.

Las medidas pueden dividirse en 3 grupos:

- I. **Variables raciales:** son medidas zoométricas cuya mayor importancia radica en su efectividad para la descripción de las características de las distintas razas y en su utilidad para evidenciar la existencia de dimorfismo sexual dentro de las poblaciones analizadas (Figura 3-22).
  1. **Longitud de la cabeza:** Longitud del segmento recto comprendido entre el punto más dorsal de la nuca y el más rostral del labio maxilar (LCB).
  2. **Anchura de la cabeza:** Longitud del segmento recto comprendido entre los puntos más salientes lateralmente de los arcos cigomáticos (ACB).
  3. **Longitud del cráneo:** Longitud del segmento recto comprendido entre el punto más culminante del occipital y el punto medio de la unión fronto-nasal (LCR).
  4. **Anchura del cráneo:** Longitud del segmento recto comprendido entre los puntos inmediatamente superiores a las apófisis coronoides de las ramas mandibulares (ACR).

5. Longitud de la cara: Longitud del segmento recto comprendido entre el punto medio de la unión fronto-nasal y el punto más rostral del labio maxilar (LCA).
6. Anchura de la cara: Longitud del segmento recto comprendido entre los puntos más laterales de las crestas malares (ACA)
7. Longitud de las orejas: Distancia recta entre la base y la punta de la oreja (LO)
8. Perímetro central del cuello: Perímetro del cuello tomado en su parte central (PCC).



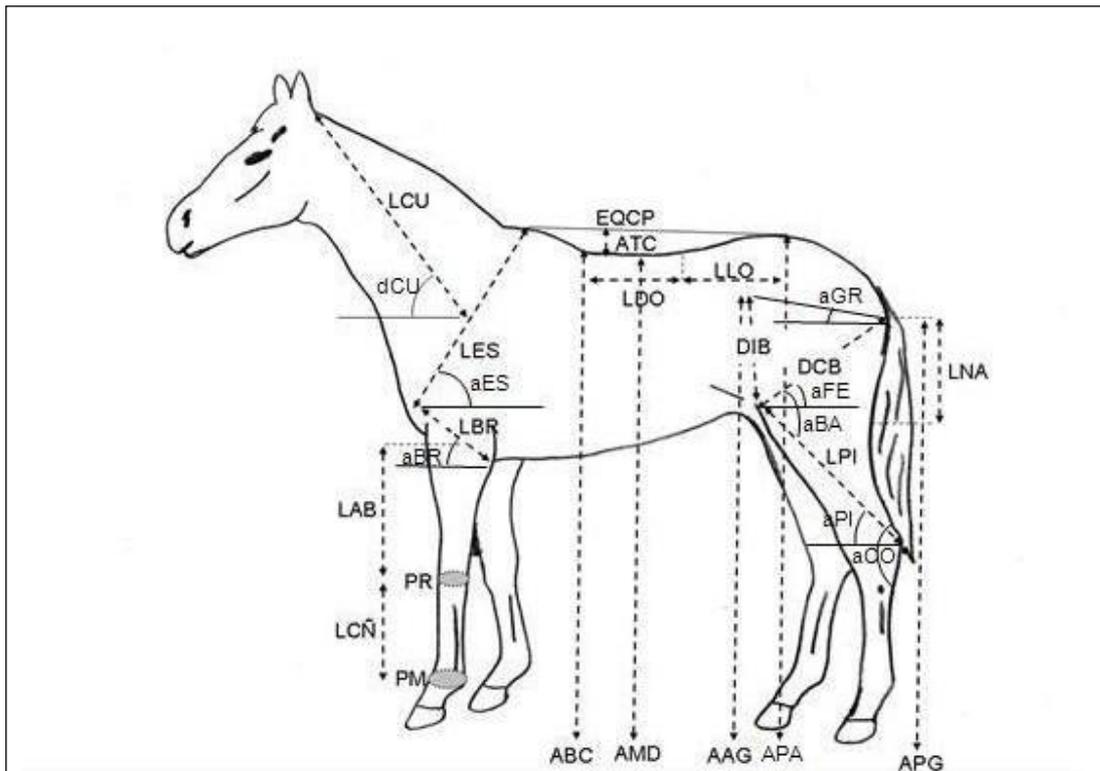
Donde: Longitud de la cabeza es LCB, Anchura de la cabeza es ACB, Longitud del cráneo es LCR, Anchura del cráneo es ACR, Longitud de la cara es LCA, Anchura de la cara es ACA, Longitud de las orejas es LO y Perímetro central del cuello es PCC.

Figura 3-22. Medidas zoométricas relacionadas con la calidad racial

**II. Variables funcionales:** conjunto de medidas zoométricas con gran importancia, directa o indirecta, en la determinación de la aptitud funcional de los animales de una Raza, utilizadas en diferentes poblaciones equinas para realizar una selección indirecta de la funcionalidad (Figura3-23).

9. Altura de la cruz: Diferencia entre la alzada al punto bajo de la cruz y la alzada a la cruz (ATC).
10. Alzada al punto bajo de la cruz: Longitud del segmento vertical comprendido entre el punto más bajo de la cruz y el suelo en el que apoya el animal (ABC).
11. Alzada a la mitad del dorso: Longitud del segmento vertical comprendido entre la parte más dorsal del punto medio del dorso (apófisis espinosa de la 12<sup>a</sup>-13<sup>a</sup> vértebras torácicas) y el suelo en el que apoya el animal (AMD).
12. Alzada a la porción anterior de la grupa: Longitud del segmento vertical comprendido entre la parte cráneo-lateral de la tuberosidad isquiática, y el suelo en el que apoya el animal (AAG).
13. Alzada a las palomillas: Longitud del segmento vertical comprendido entre la parte más dorsal de la tuberosidad sacra y el suelo en el que apoya el animal (APA).
14. Alzada a la porción posterior de la grupa: Longitud del segmento vertical comprendido entre la parte más caudal de la tuberosidad coxal, y el suelo en el que apoya el animal (APG).
15. Equilibrio cruz-palomillas: Diferencia entre la alzada a las palomillas y alzada a la cruz (EQCP).
16. Longitud del cuello: Distancia entre la base de la oreja y la mitad de la espina de la escápula (LCU).
17. Longitud de la espalda: Distancia entre el punto más alto de la cruz y la parte central de la articulación del encuentro (LES).
18. Distancia coxal-babilla: Distancia comprendida entre la porción posterior de la grupa (parte caudolateral de la tuberosidad coxal) y la babilla (DCB).
19. Distancia isquión-babilla: Distancia comprendida entre la porción anterior de la grupa (parte centrolateral de la tuberosidad isquiática) y la babilla(DIB).

20. Longitud de la nalga: Longitud del segmento recto comprendido entre la parte más caudal de la tuberosidad coxal y el final de la masa muscular del miembro pelviano (LNA).
21. Longitud de la pierna: Distancia recta comprendida entre la parte central de la rótula y la punta del corvejón (LPI).
22. Longitud del brazo: Distancia recta comprendida entre el punto más craneal de la articulación del encuentro y el codo del animal (LBR).
23. Longitud del antebrazo: Distancia recta entre la línea paralela al suelo trazada desde el codo hasta la mitad de la articulación de la rodilla (LAB).
24. Longitud de la caña anterior: Distancia comprendida entre la mitad de la articulación de la rodilla del animal y la mitad de la articulación del menudillo (LCN).
25. Longitud del dorso: Distancia desde el punto más bajo de la cruz al final de la región torácica (última vértebra torácica) (LDO).
26. Longitud del lomo: Distancia entre el final de la región torácica del animal (última vértebra torácica) y sus palomillas (parte más dorsal de la tuberosidad sacra) (LLO).
27. Perímetro de la rodilla: Máximo perímetro de la articulación de la rodilla (PR).
28. Perímetro del menudillo: Máximo perímetro de la articulación del menudillo.
29. Dirección del cuello: Ángulo formado por la línea que va desde la base de la oreja al punto medio de la espina de la escápula, con respecto a la horizontal (dCU).
30. Angulación de la espalda: Ángulo formado por la línea que va desde el punto más alto de la cruz a la parte central del encuentro, con respecto a la horizontal (aES).
31. Angulación del brazo: Ángulo formado por la trayectoria del brazo (desde la articulación del hombro hasta el olécranon (codo) con respecto a la horizontal (aBR).
32. Angulación de la grupa: Ángulo formado por la línea que va desde la punta del anca a la punta de la nalga, con respecto a la horizontal (aGR).
33. Angulación de la babilla: Ángulo formado por las líneas que van desde el trocánter mayor del fémur a la babilla y desde ésta al corvejón del animal (aBA).
34. Angulación del fémur: Ángulo formado por la línea que va desde el trocánter mayor del fémur hasta la babilla con respecto a la horizontal (aFE).
35. Angulación de la pierna: Ángulo formado por la línea que une la babilla con el corvejón (maléolo lateral de la tibia), con respecto a la horizontal (aPI).
36. Angulación del corvejón: Ángulo formado por las líneas que unen la babilla con el corvejón y la mitad de la articulación del menudillo con el corvejón (aCO).

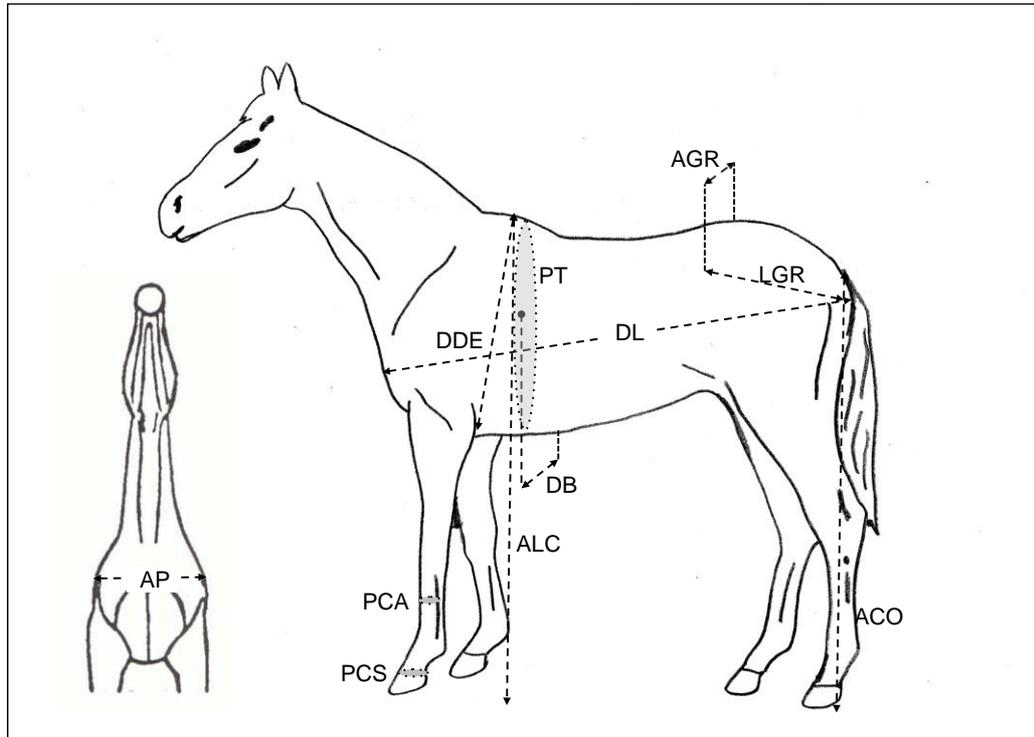


Donde: Altura de la cruz es ATC, Alzada al punto bajo de la cruz es ABC, Alzada a la mitad del dorso es AMD, Alzada a la porción anterior de la grupa es AAG, Alzada a las palomillas es APA, Alzada a la porción posterior de la grupa es APG, Equilibrio cruz-palomillas es EQCP, Longitud del cuello es LCU, Longitud de la espalda es LES, Distancia coxal-babilla es DCB, Distancia isquión-babilla es DIB, Longitud de la nalga es LNA, Longitud de la pierna es LPI, Longitud del brazo es LBR, Longitud del antebrazo es LAB, Longitud de la caña anterior es LCN, Longitud del dorso es LDO, Longitud del lomo es LLO, Perímetro de la rodilla es PR, Perímetro del menudillo es PM, Dirección del cuello es dCU, Ángulo de la espalda es aES, Ángulo del brazo es aBR, Ángulo de la grupa es aGR, Ángulo de la babilla es aBA, Ángulo del fémur es aFE, Ángulo de la pierna es aPI y Ángulo del corvejón es aCO.

Figura 3-23. Medidas zoométricas relacionadas con el rendimiento funcional

**III. Variables raciales y funcionales:** conjunto de medidas zoométricas que se relacionan tanto con la calidad racial de los animales como con su aptitud funcional (Figura 3-24).

37. Alzada a la cruz: Longitud del segmento vertical comprendido entre el punto más alto de la cruz y el suelo en el que apoya el animal (ALC).
38. Alzada al nacimiento de la cola: Longitud del segmento vertical comprendido entre el punto más dorsal de la unión de la cola con la grupa (4ª vértebra coxígea) y el suelo en el que apoya el animal (ACO).
39. Anchura de pecho: Longitud del segmento recto comprendido entre la parte craneal de los tubérculos mayores de los húmeros (AP).
40. Diámetro dorso-esternal: Distancia del segmento recto comprendido entre el punto más culminante de la cruz y el esternón en el plano inmediatamente posterior al codo (DDE).
41. Diámetro bicostal: Longitud del segmento recto comprendido entre los puntos más laterales de los arcos costales de la 5ª costilla (DB).
42. Anchura de la grupa: Longitud del segmento recto comprendido entre los puntos más laterales y craneales de las tuberosidades coxales (AGR).
43. Longitud de la grupa: Longitud del segmento recto comprendido entre el punto más rostral de la tuberosidad coxal y el más caudal de la tuberosidad isquiática (LGR).
44. Diámetro longitudinal: Distancia del segmento recto comprendido entre el encuentro y la punta de la nalga (DL).
45. Perímetro del tórax: Los puntos de referencia son el punto más declive de la cruz (apófisis espinosa de la 7ª-8ª vértebras dorsales) y el esternón en el plano inmediatamente posterior al codo (PT).
46. Perímetro de la caña: Perímetro de la región metacarpiana en su tercio medio (PCA).
47. Perímetro del casco anterior: Perímetro del borde superior del casco anterior (PCS).



Donde: Alzada a la cruz es ALC, Alzada al nacimiento de la cola es ACO, Anchura de pecho es AP, Diámetro dorso-esternal es DDE, Diámetro bicostal es DB, Anchura de la grupa es AGR, Longitud de grupa es LGR, Diámetro longitudinal es DL, Perímetro del tórax es PT, Perímetro de la caña es PCA y Perímetro del casco anterior es PCS.

Figura 3-24. Medidas zoométricas relacionadas con la calidad racial y el rendimiento funcional

### 3.5.1 Método de medición manual

En las siguientes figuras se muestra cómo se realiza la toma de medidas manualmente:



Figura 3-25. Medida Alzada a la cruz



Figura 3-25. Medida Altura al punto más declive de la cruz



Figura 3-27. Medida Alzada a las palomillas



Figura 3-28. Medida Longitud de la cabeza



Figura 3-29. Medida Anchura de la cabeza



Figura 3-30. Medida Longitud del cuello



Figura 3-31. Medida Longitud de la espalda



Figura 3-32. Medida Longitud del antebrazo



Figura 3-33. Medida Diámetro dorso esternal



Figura 3-34. Medida Long. Long. dorso + lomo



Figura 3-35. Medida Longitud de la grupa



Figura 3-36. Medida Distancia nalga-babilla



Figura 3-37. Medida Longitud de la pierna



Figura 3-38. Medida Longitud de la nalga



Figura 3-39. Medida Ángulo de la espalda



Figura 3-40. Medida Ángulo de la grupa



Figura 3-41. Medida Anchura de pecho



Figura 3-42. Medida Anchura de la grupa



Figura 3-43. Medida Diametro bicostal



Figura 3-44. Medida Perímetro del tórax



Figura 3-45. Medida Perímetro de la caña



Figura 3-46. Medida Diámetro Longitudinal

### 3.6 Toma de medidas con el sensor LiDAR

#### 3.6.1 Configuración de la Raspberry para comunicarse con el Velodyne

En primer lugar se deben realizar las conexiones pertinentes de hardware para la alimentación y comunicación de la caja interfaz con el ordenador según lo indicado en el manual de usuario del Velodyne:

- 1) Alimentar el LiDAR a través del adaptador CA/CC incluido.
- 2) Conectar el LiDAR a un puerto Ethernet de la Raspberry.
- 3) Por ahora, deshabilitar la conexión WiFi en la Raspberry.



Figura 3-47. Conexiones del hardware

Es necesario configurar la dirección IP de la Raspberry.

- 1) En "Conexiones de red" elegir Configuración de IPV4 y cambiar el campo "Método" a "Manual".
- 2) Configurar la dirección IP en 192.168.1.100, Netmask en 255.255.255 y Gateway en 0.0.0.0.

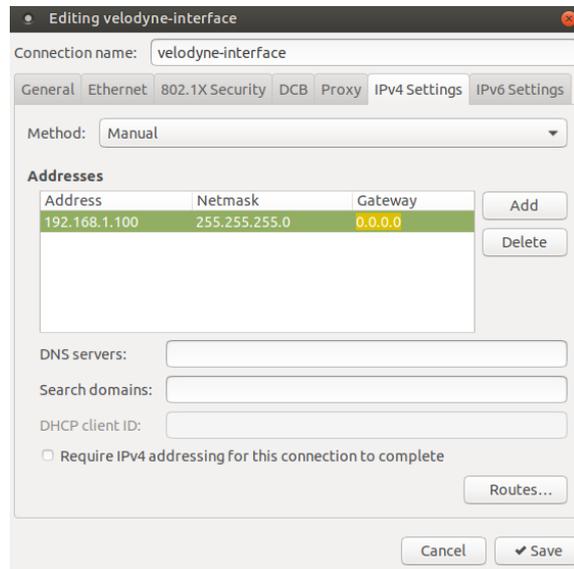


Figura 3-48. Configuración de la dirección IP

Conectado el LiDAR a la Raspberry, hay que asignar estáticamente una dirección IP a este puerto en el rango 192.168.3.x.

```
sudo ifconfig eth0 192.168.3.100
```

Se agrega una ruta estática a la dirección IP del sensor.

```
sudo route add 192.168.1.201 eth0
```

Para verificar la conexión, se accede a la dirección de red '192.168.1.201' del sensor a través del navegador, en la cual aparecerá la pantalla de configuración del LiDAR. Desde aquí se pueden modificar diferentes parámetros del sensor, entre los cuales el ángulo del campo de visión que se ajustará durante la realización del ensayo. [23]

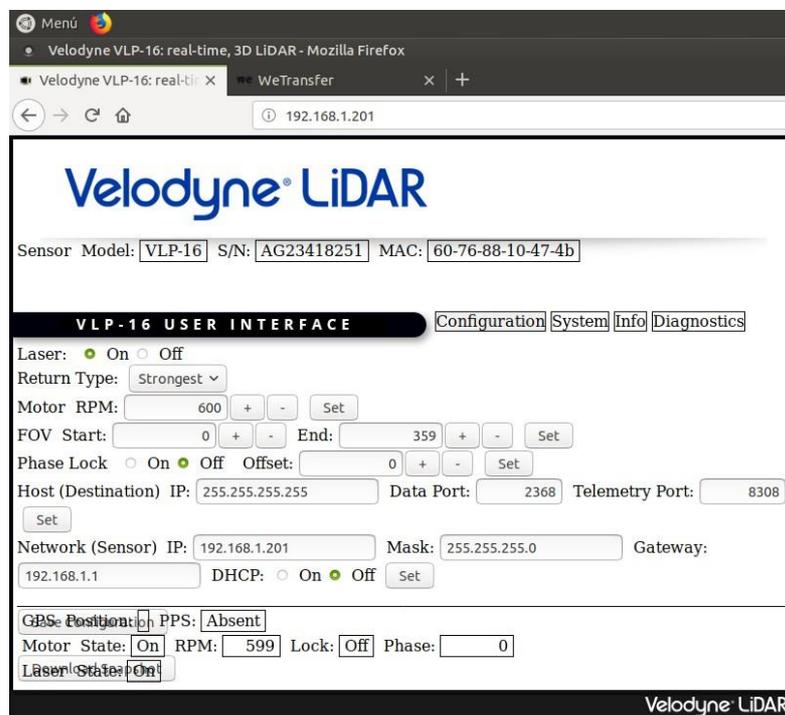


Figura 3-49. Pantalla de configuración del sensor LiDAR Velodyne

### 3.6.2 Instalación de ROS

Para ejecutar funciones básicas del sensor LiDAR Velodyne VLP-16 para realizar un SLAM, se debe realizar la instalación de los sistemas operativos pertinentes, así como los paquetes necesarios para la ejecución de los algoritmos mencionados, lo cual hace necesario instalar ROS en dicho ordenador.

La versión elegida ha sido ROS Melodic Morenia disponible para Ubuntu Bionic 18.04.2 ideal para la Raspberry pi 3. Se recomienda seguir tutoriales de la página oficial del ROS para instalarlo y lograr entender los tópicos de los cuales se hablan.



Figura 3-50. ROS Melodic Morenia

Antes de comenzar con la instalación, es necesaria la configuración de repositorios y dependencias:

- **Repositorios de de ROS:** Para proceder con la instalación es necesario crear un archivo raíz, además de obtener un paquete de autenticación de ROS. Este proceso creará un archivo llamado *ros-latest.list*, el cual contendrá dicha llave de autenticación.

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main"> /etc/apt/sources.list.d/ros-latest.list'

sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com: 80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

- **Dependencias Bootstrap:** Herramientas que permiten la descarga y gestión de los paquetes de ROS y sus dependencias.

```
sudo apt install python-rosinstall python-rosinstall-generator python-wstool build-essential
```

Adicionalmente se puede actualizar *Debian Package*

```
sudo apt update
```

Antes de poder utilizar ROS, se debe inicializar *rosdep*. *rosdep* es una herramienta de línea de comandos que permite instalar fácilmente las dependencias del sistema.

```
sudo rosdep init
rosdep update
```

Después de la instalación de las dependencias necesarias, se puede proceder con la instalación. La instalación está basada en la compilación de los núcleos o paquetes esenciales por lo que se necesita un *Catkin Workspace*:

```
mkdir -p~/catkin_ws
cd~/catkin_ws
```

A continuación se procede a reunir los paquetes esenciales para su compilación, añadiendo todos los paquetes *catkin* o *wet* de la variante deseada (*melodic*) en el entorno de trabajo, específicamente en */ros\_catkin\_ws/src*. Actualmente se ofrecen dos variantes de paquetes a instalar por defecto:

- **ROS-Comm:** ROS package, build y librerías de comunicación.
- **Desktop:** ROS package, rqt, rviz y librerías de robótica.

La diferencia principal entre los dos paquetes es que ROS-Comm no incluye interfaces gráficas de usuario, mientras que Desktop sí las incluye. Debido al alcance de este proyecto, se selecciona Desktop por el tipo de aplicaciones que se van a desarrollar.

```
sudo apt install ros-melodic-desktop-full
```

### 3.6.3 Instalación de dependencias de ROS

Las dependencias de ROS se instalan implementando el siguiente código en la terminal:

```
sudo apt-get install ros-melodic-velodyne
```

### 3.6.4 Instalación del controlador VLP-16

En la terminal, dentro del área de trabajo de ROS, en la carpeta "src" se ejecuta el siguiente comando:

```
cd ~/catkin_ws/src/ && git clone https://github.com/ros-drivers/velodyne.git
```

Después de esto, dentro del área de trabajo, es necesario actualizar todas las dependencias y construir un espacio de trabajo:

```
rosdep update
cd ~/catkin_ws/ && catkin_make
```

Ahora, el paquete Velodyne está listo para ejecutarse. [24]

### 3.6.5 Visualización de la nube de puntos

Se ejecuta el siguiente comando en la terminal:

```
roslaunch velodyne_pointcloud VLP16_points.launch
```

Se pueden ver los mensajes que se publican y se suscriben al siguiente tópico:

```
rostopic echo /velodyne_points
```

Después de esto se inicializa RViz, donde es necesario seleccionar como tópico "Point Cloud2"

```
roslaunch rviz rviz -f velodyne
```

A continuación se muestra la nube de puntos que genera la herramienta de visualización RViz. A través de esta nube de puntos ya podemos conocer la reflectancia de los materiales que escanea. [23]

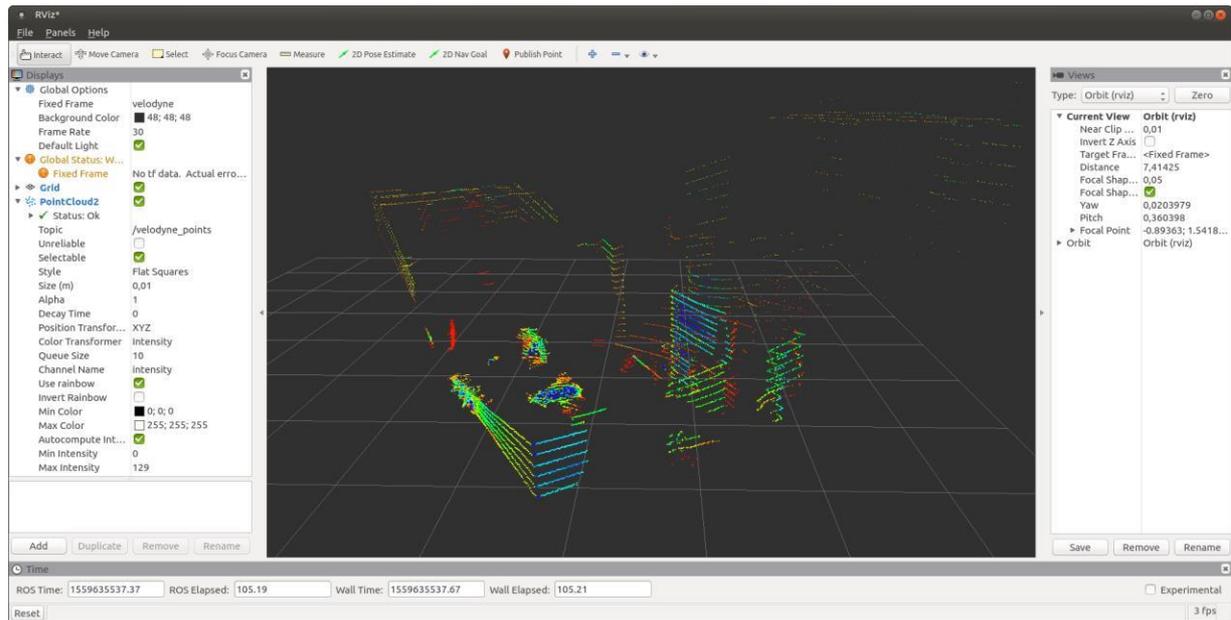


Figura 3-51. Interfaz gráfica de RViz y nube de puntos generada en el laboratorio

Hay dos formas de visualizar datos en RViz: a través del modo de transmisión en vivo y del modo de reproducción. La transmisión en vivo corresponde a la situación en la que un sensor Velodyne está conectado directamente al ordenador ejecutándose al mismo tiempo Rviz. En este caso, el modo de transmisión en vivo permite visualizar la salida actual del sensor (como una cámara web).

El modo de reproducción corresponde a la situación que se ha grabado mientras se transmitía en vivo. Como se ha comentado anteriormente, a través del comando *rosvbag* ejecutado por terminal se realiza la grabación del contenido de todos los tópicos, creándose un archivo *.bag*

```
rosvbag record -a
```

Para poder trabajar con los datos en el software de procesamiento de nube de puntos es necesario convertir el archivo *.bag* grabado en RViz en un formato de archivo separado por comas *.csv*, o en un archivo *.pcd*.

La instrucción para convertir un archivo *.bag* en un archivo *.pcd* que pueda abrirse en el software de procesamiento de nube de puntos, el cual se describirá en el siguiente capítulo, es la siguiente:

```
roslaunch pcl_ros bag_to_pcd "nombre_del_archivo".bag /pointcloud2 ./"nombre_carpeta_del_archivo_pcd"
```

Por otra parte los archivos CSV son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea. Para convertir un archivo *.bag* en un archivo *.csv* es necesario realizar lo siguiente:

```
cd ~ / catkin_ws / src
git clone https://github.com/AtsushiSakai/rosbag_to_csv.git
cd ~ / catkin_ws && rosdep install -r --ignore-src-from-paths src
```

Se instala el nodo:

```
roslaunch rosbag_to_csv rosbag_to_csv.py
```

Se selecciona el archivo bag y los tópicos que se pretente convertir a csv:

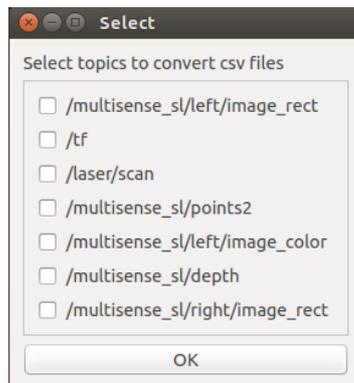


Figura 3-52. Selección de tópicos para convertir a csv

A continuación, se espera unos segundos mientras se muestra un mensaje “Converting...” en la terminal. La conversión finaliza cuando se muestra un cuadro de diálogo de conversión finalizada:



Figura 3-53. Mensaje de conversión finalizada

De esta forma se genera una matriz de datos de cada uno de los frames, correspondientes a la duración del ensayo. En este caso se dispondrá de un solo frame por ensayo y por caballo. La matriz de datos consta de 9 columnas con los siguientes valores: coordenada X, coordenada Y, coordenada Z, intensidad, canal del láser, azimut, distancia tiempo ajustado y tiempo de la captura.

	A	B	C	D	E	F	G	H	I
1	Points_m_XYZ:0	Points_m_XYZ:1	Points_m_XYZ:2	intensity	laser_id	azimuth	distance_m	adjustedtime	timestamp
2	0.005539392586797476	453.405.237.197.876	-1.214.896.559.715.270	38.0	0.0	7.0	4.694	1829404654.0	1829404654.0
3	0.005880163982510567	421.135.425.567.627	0.07350953668355942	5.0	1.0	8.0	4.212	1829404656.0	1829404656.0
4	0.008451621048152447	5.380.465.030.670.160	-1.242.179.751.396.170	20.0	2.0	9.0	5.522	1829404658.0	1829404658.0
5	0.06955049186944962	4.427.718.353.271.480	2.320.471.525.192.260	22.0	3.0	9.0	44.338	1829404661.0	1829404661.0
6	0.01127668935805559	6.461.060.047.149.650	-1.255.904.793.739.310	14.0	4.0	10.0	6.582	1829404663.0	1829404663.0
7	0.1021266281604767	5.319.470.596.313.470	4.653.942.584.991.450	21.0	5.0	11.0	53.398	1829404665.0	1829404665.0
8	0.01698327250778675	8.108.903.884.887.690	-1.284.326.910.972.590	20.0	6.0	12.0	8.21	1829404668.0	1829404668.0
9	0.1184469237923622	5.220.382.308.959.960	6.409.840.106.964.110	26.0	7.0	13.0	52.596	1829404670.0	1829404670.0
10	0.02592102251946926	1.060.830.116.271.970	-1.302.539.587.020.870	12.0	8.0	14.0	10.688	1829404672.0	1829404672.0
11	0.1323591619729996	5.416.861.724.853.510	857.949.161.529.541	39.0	9.0	14.0	54.844	1829404675.0	1829404675.0
12	0.01214819774031639	46.402.587.890.625	-0.4059714376926422	15.0	10.0	15.0	4.658	1829404677.0	1829404677.0
13	0.01394382677972317	4.699.530.124.664.300	-0.2462930083274841	26.0	12.0	17.0	4.706	1829404681.0	1829404681.0
14	0.1718748211860657	5.182.981.872.758.590	-0.904697835445404	15.0	14.0	19.0	51.838	1829404686.0	1829404686.0
15	0.02139346674084663	4.539.801.120.758.050	-1.216.449.499.130.240	40.0	0.0	27.0	4.7	1829404709.0	1829404709.0
16	0.02060004323720932	4.215.307.712.554.930	0.0735793486237526	2.0	1.0	28.0	4.216	1829404711.0	1829404711.0
17	0.0271539893001318	5.364.812.850.952.140	-1.238.580.465.316.770	19.0	2.0	29.0	5.506	1829404714.0	1829404714.0
18	0.2244904190301895	4.435.256.576.538.080	2.324.449.062.347.410	22.0	3.0	29.0	44.414	1829404716.0	1829404716.0
19	0.0339018851518631	6.474.724.292.755.120	-1.258.576.154.708.860	14.0	4.0	30.0	6.596	1829404718.0	1829404718.0
20	0.2924455106258392	5.405.073.928.833.000	4.728.896.141.052.240	22.0	5.0	31.0	54.258	1829404721.0	1829404721.0
21	0.04531059041619301	811.274.528.503.418	-1.284.952.640.533.440	20.0	6.0	32.0	8.214	1829404723.0	1829404723.0

Figura 3-54. Matriz de datos obtenida en el ensayo del caballo ENCANTADO 49

Las tres primeras columnas corresponden a las coordenadas de los puntos expresadas en metros respecto a la posición del LiDAR. El eje Y coincide con el eje longitudinal del caballo; el eje X, con el eje transversal; y el eje Z, con el vertical.

La intensidad hace referencia a la cantidad de radiación reflejada por la superficie. Está relacionada con la reflectividad de la superficie contra la que impacta. En RViz se representa mediante el color de los puntos.

El laser\_id indica el canal del láser por el que se recibe la señal. Cada uno de ellos corresponde a una cierta inclinación vertical, que varía desde -15° hasta 15°.

El azimuth corresponde al ángulo de barrido de cada uno de los puntos. Está expresado en centésimas de radián.

La distancia es aquella que se ha medido entre cada punto y el LIDAR, está expresada en metros. Se observa, que con el valor de estas tres últimas variables es posible calcular las coordenadas de los puntos, y viceversa.

Por último, las dos últimas columnas corresponden al tiempo en el que se ha capturado cada uno de los puntos. La columna de timestamp corresponde al instante de tiempo desde que se inicia en ensayo y la columna ajustedtime corresponde al instante de tiempo ajustado para que coincida con el tiempo de la señal GPS. Ambas están expresadas en microsegundos.

Para este trabajo, no todas estas variables serán de interés, de todas ellas se utilizarán únicamente las tres coordenadas de los puntos.

Obtenida la matriz de datos en formato .csv se exporta a *CloudCompare* que procesa la nube de puntos.

### 3.6.6 Procesamiento de la nube de puntos

*CloudCompare* es un software libre de procesamiento de nubes de puntos 3D y mallas de triangulación. Originalmente ha sido diseñado para llevar a cabo la comparación entre dos nubes de puntos 3D (tales como las obtenidas con un escáner láser), o entre una nube de puntos y una malla triangular. Se trata de un software destinado a hacer frente a enormes nubes de puntos, por lo general, más de 10 millones de puntos y hasta 120 millones con 2 gigabytes de memoria.

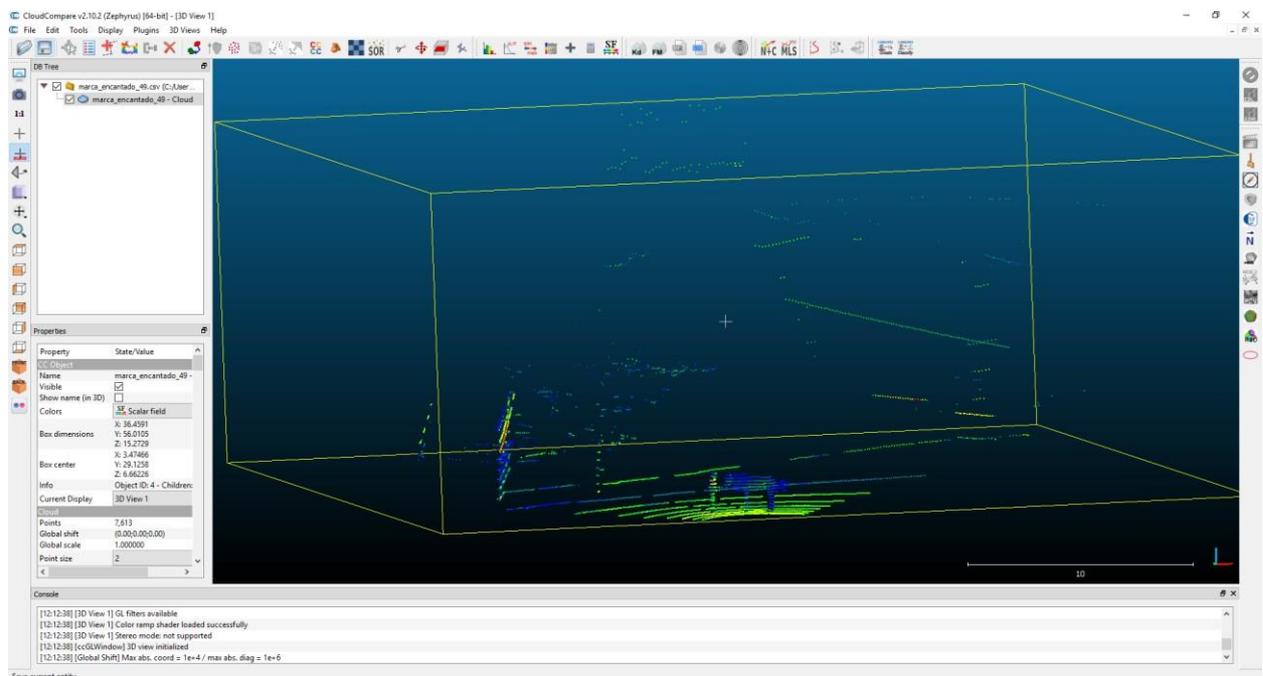


Figura 3-55. Nube de puntos del ensayo del caballo ENCANTADO 49

Debido a que la toma de datos se realizó en uno de los patios de la Yeguada de la Cartuja, es necesario realizar un filtrado manual de la nube de puntos para facilitar la extracción de las medidas de las variables zoométricas. El filtrado consiste en eliminar de la nube de puntos completa todos aquellos objetos que rodean al caballo y que dificultan la toma de medidas, como pueden ser los árboles del patio o el operario que mantiene atado el caballo.

Para filtrar la nube de puntos se utiliza la herramienta de Segmentación Interactiva (en inglés *Interactive Segmentation*). Se puede acceder a esta herramienta mediante el icono  en la barra de herramienta superior o a través del menú 'Edit>Segment'.

Esta herramienta permite segmentar interactivamente los objetos seleccionados definiendo un polígono 2D (o un rectángulo) en la pantalla. Este proceso puede repetirse varias veces, cambiando la orientación de los objetos para segmentar adecuadamente los objetos en 3D. En cada segmentación se puede decidir mantener los puntos que caen dentro o fuera del borde del polígono. El procedimiento es el siguiente:

Seleccione uno o varios objetos y ejecute la herramienta. Aparecerá una nueva barra de herramientas en la esquina superior derecha de la vista 3D.

- **Modo de edición Poligonal (“Polygonal selection”)**

Por defecto, la herramienta comienza en modo de edición poligonal. Esto significa que se puede comenzar a dibujar el polígono de inmediato:

- 1) Clic izquierdo: se crea un nuevo vértice de polígono
- 2) Justo después de crearse el primer vértice, el primer borde del polígono comenzará a "seguir" el cursor del ratón. Se debe definir la posición del segundo vértice (clic izquierdo) para “arreglarlo”. Este proceso comenzará de nuevo con el siguiente borde y así sucesivamente.
- 3) Clic derecho: se detiene la edición del polígono. [25]

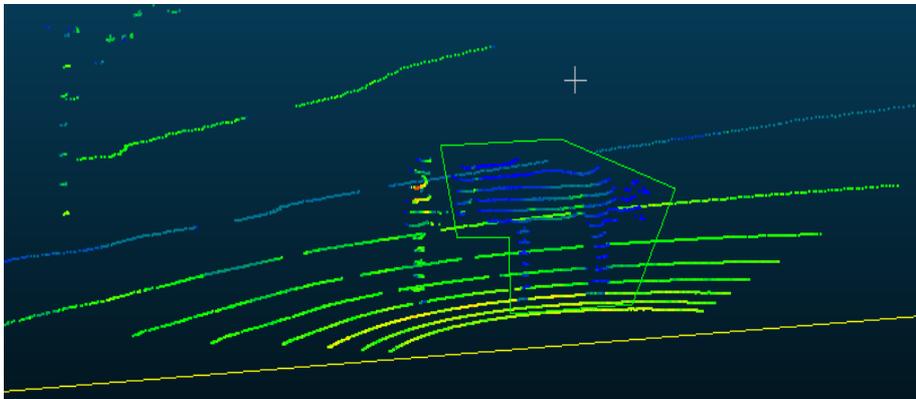


Figura 3-56. Segmentación en modo de edición poligonal

- **Modo de edición Rectangular (“Rectangular selection”)**

Puede cambiar el modo edición poligonal al modo edición rectangular haciendo clic en la flecha hacia abajo junto al icono “polígono”:



Figura 3-57. Barra de herramientas de la Segmentación Interactiva

En el modo edición rectangular, se hace clic con el botón izquierdo para definir la primera esquina del rectángulo, luego se mantiene presionado el botón izquierdo del ratón, se mueve el ratón y se suelta el botón para definir la esquina opuesta. [25]

Una vez que se finaliza la edición del polígono/rectángulo, haciendo clic en el botón izquierdo, el proceso de edición comenzará nuevamente (es decir, el polígono de segmentación actual se descartará).

De lo contrario, dependiendo del objeto seleccionado, se puede elegir si desea mantener los puntos dentro (🔴) o fuera (🔵) del polígono. Una vez hecho esto, los otros puntos desaparecerán (así como el polígono).

Para validar la segmentación actual y crear dos nubes: una con los puntos seleccionados y otra con los otros (hacer clic en el botón ✓) [25]

Finalmente se consigue eliminar el entorno de caballo y aislarlo para obtener las diferentes medidas (ver figura 3-58)

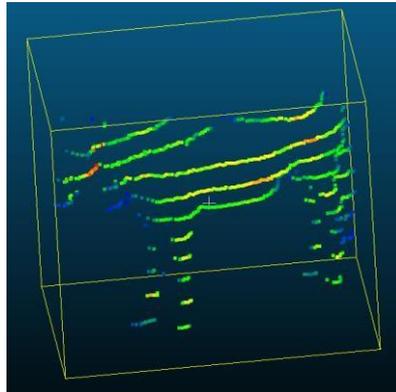


Figura 3-58. Reconstrucción tridimensional del caballo

### 3.6.7 Método de medición con el sensor LiDAR

El LiDAR de Velodyne usa la luz del láser para examinar el entorno. Un haz de luz láser incide en el objeto que se pretende examinar. Como ya se ha mencionado anteriormente, una de las características del LiDAR Velodyne VLP-16 es que tiene 16 rayos láser girando 360° de manera permanente a 600 rpm para monitorizar todo el entorno.

Para estabilizar el sensor, evitar su movimiento y regular su altura se hace uso de un trípode. El sensor atornillado al trípode debe estar colocado a una distancia lo suficientemente alejada del caballo para que pueda ser capturado por completo e incidan sobre su cuerpo el mayor número de canales de adquisición. Además debe colocarse a una altura que favorezca a que los rayos del sensor incidan lo menos posible en el suelo y en la parte superior del caballo.

Como el campo de visión vertical del sensor es de 30°, se ha colocado el sensor verticalmente, perpendicular al suelo, asegurando un barrido completo del caballo.

Desde la pantalla de configuración del sensor LiDAR es posible reducir el campo de visión vertical, evitando que el sensor escanee la parte trasera, centrándose únicamente en la superficie del caballo.



Figura 3-59. Montaje del sensor LiDAR sobre el trípode

A partir de la nube de puntos generada y filtrada a través del programa *CloudCompare* descrito anteriormente (ver figura 3-58 en el capítulo anterior) se obtienen las medidas de las variables zoométricas raciales y funcionales con la herramienta *Point Picking*. Se puede acceder a esta herramienta a través del icono  en la barra de herramientas superior o en el menú 'Tools > Point picking'. El objetivo principal de esta herramienta es permitir elegir uno, dos o tres puntos para obtener diversos datos (sobre todo la distancia entre dos puntos). Se pueden crear diferentes tipos de 'etiquetas' de esta manera.

Al hacer click sobre Point picking aparecerá una barra de herramientas en la esquina superior derecha de la vista 3D. De las diferentes acciones que aparecen, las dos que se utilizarán en este trabajo son las siguientes:

- **Etiqueta de 2 puntos**

Se trata del segundo icono . Consiste en elegir dos puntos y hacer que aparezca una etiqueta con la distancia entre dichos puntos.



Distance: 2128.458536			
dX	12.303223	dXY	2128.429199
dY	2128.393555	dXZ	16.631804
dZ	11.191406	dZY	2128.422852

Figura 3-60. Etiqueta de distancia entre dos puntos

Esta etiqueta mostrará principalmente la distancia entre los dos puntos seleccionados (como título). Se utilizará para sacar todas variables zoométricas excepto los ángulos. Además, se muestran los siguientes valores:

- ↓ dX: distancia algebraica a lo largo de la dimensión X
- ↓ dY: distancia algebraica a lo largo de la dimensión Y
- ↓ dZ: distancia algebraica a lo largo de la dimensión Z
- ↓ dXY: distancia en el plano XY
- ↓ dXZ: distancia en el plano XZ
- ↓ dZY: distancia en el plano ZY

- **Etiqueta de 3 puntos**

Tercer icono . Permite elegir tres puntos y hacer que aparezca una etiqueta de 'Area'.



Area: 2107973.500000			
index.A	3475	AB	4898447.953907
index.B	133333	BC	7202750.731628
index.C	197251	CA	3732664.579328
angle.A	80.385376	Nx	-0.184999
angle.B	45.216079	Ny	-0.217080
angle.C	54.398546	Nz	-0.958463

Figura 3-61. Etiqueta de ángulo

Esta etiqueta mostrará el área del triángulo formado por los tres puntos seleccionados. Además, se muestran los siguientes valores:

- ↓ los índices de los 3 puntos
- ↓ la longitud de los 3 bordes
- ↓ los 3 ángulos.
- ↓ el vector normal a este triángulo / plano [26]

Se obtendrán los ángulos de la espalda y de la grupa con esta etiqueta. A continuación, se muestran todas las medidas zoométricas obtenidas con esta herramienta.

# 4 RESULTADOS

A partir de las herramientas mencionadas en el capítulo anterior, se obtienen las variables zoométricas en cm del caballo ENCANTADO 49. En el capítulo de Anexo se incluirán las medidas del resto de caballos.

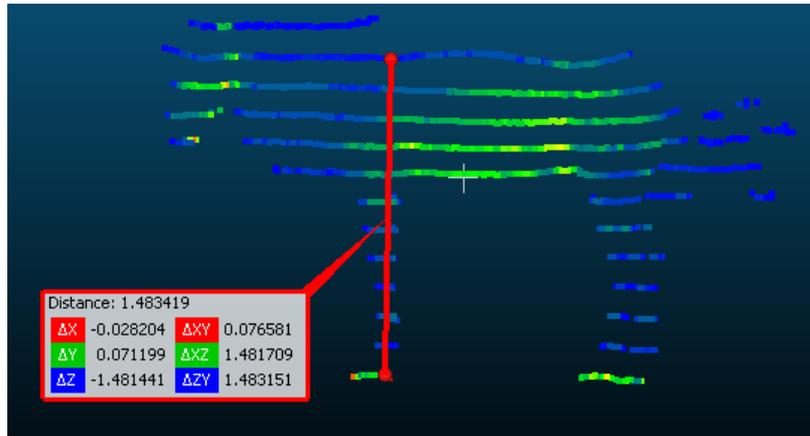


Figura 4-1. Medida zoométrica Altura al punto más declive de la cruz

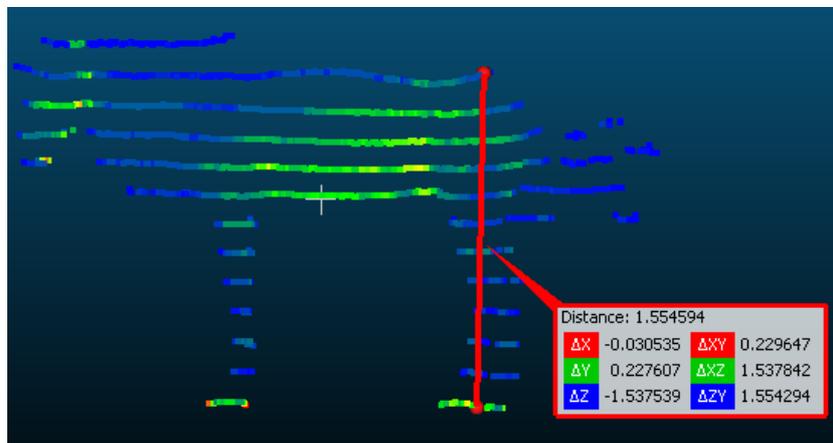


Figura 4-2. Medida zoométrica Alzada a las palomillas

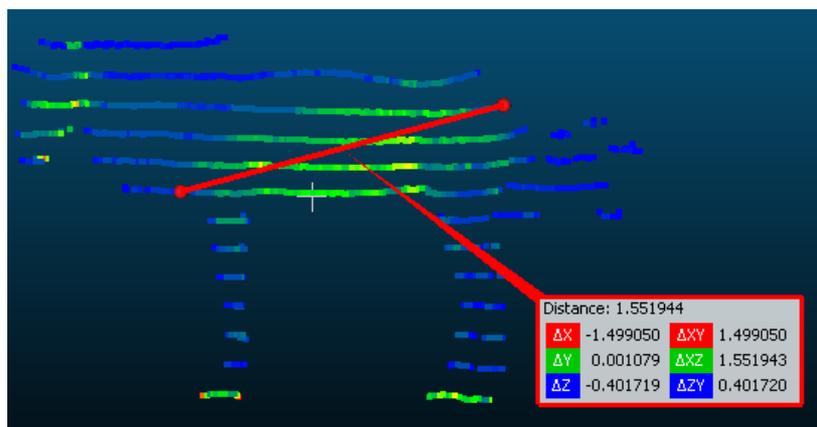


Figura 4-3. Medida zoométrica Longitud escápulo isquial

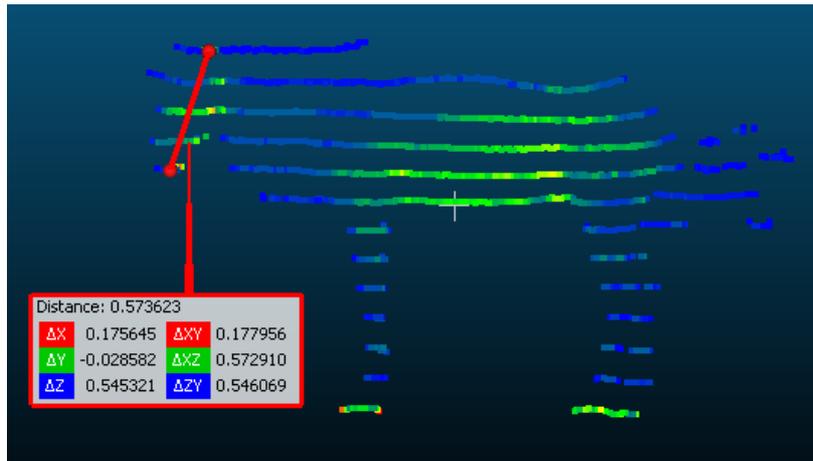


Figura 4-4. Medida zoométrica Longitud de la cabeza

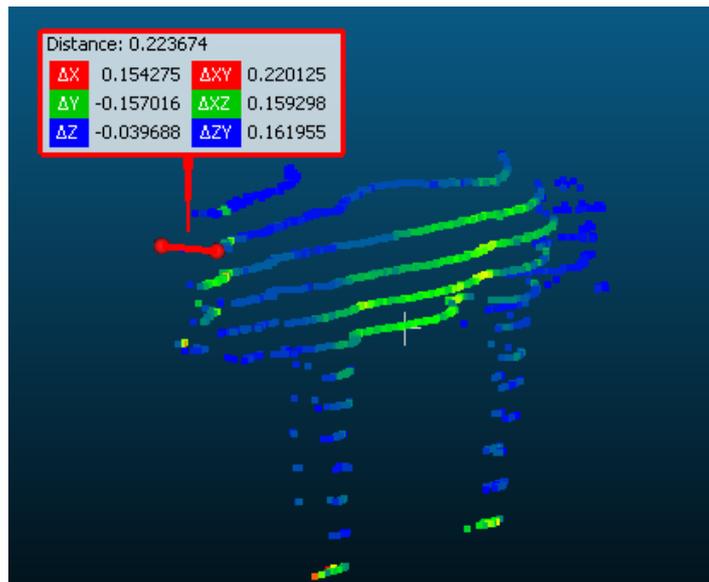


Figura 4-5. Medida zoométrica Anchura de la cabeza

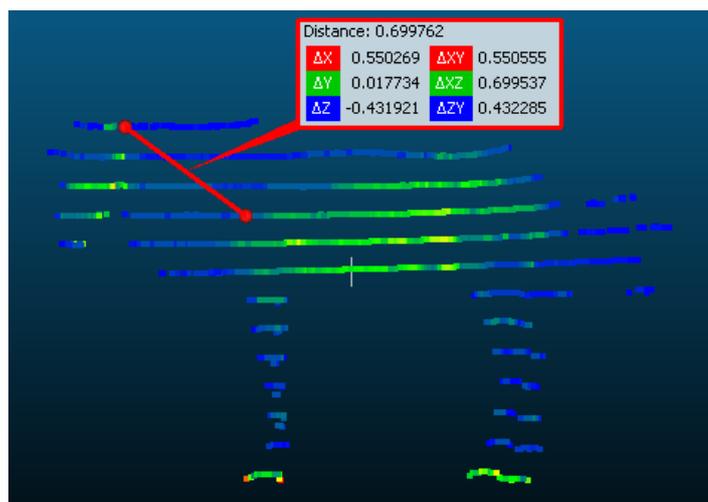


Figura 4-6. Medida zoométrica Longitud del cuello

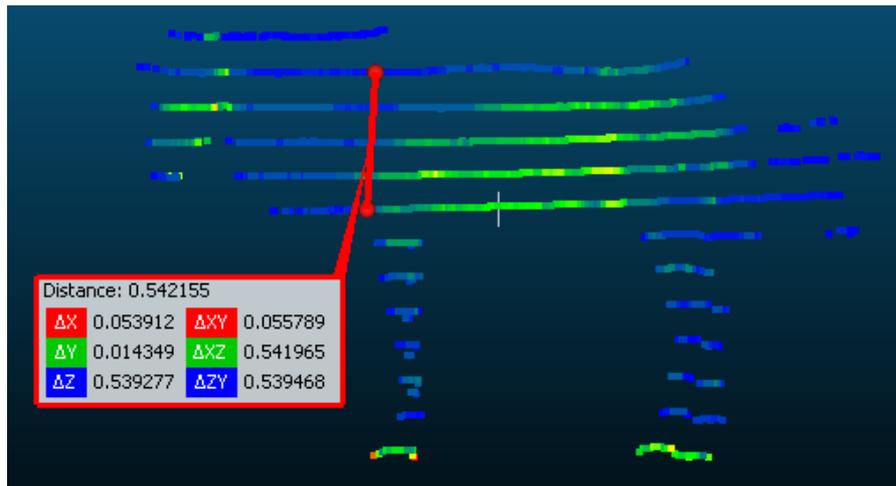


Figura 4-7. Medida zoométrica Longitud de la espalda

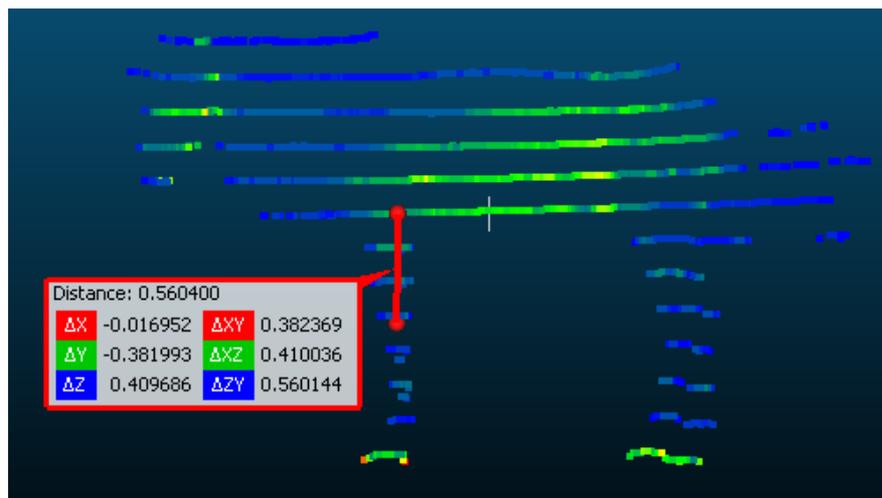


Figura 4-8. Medida zoométrica Longitud del antebrazo

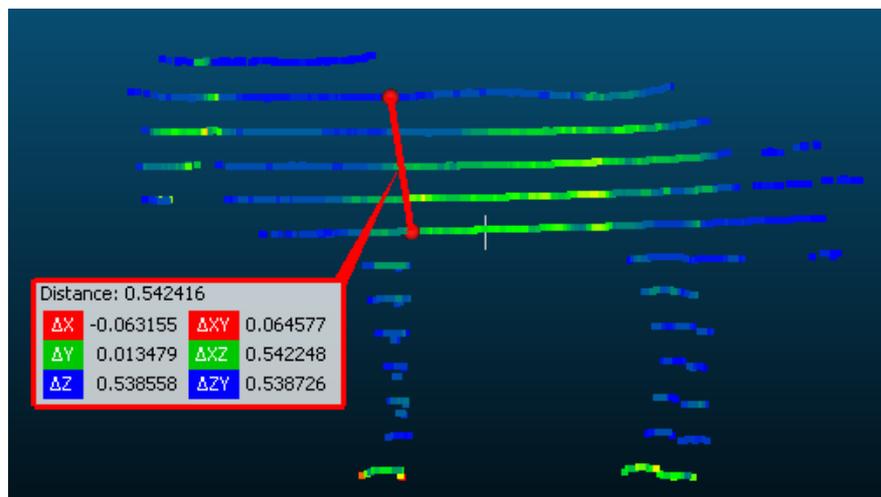


Figura 4-9. Medida zoométrica Diámetro dorso externo

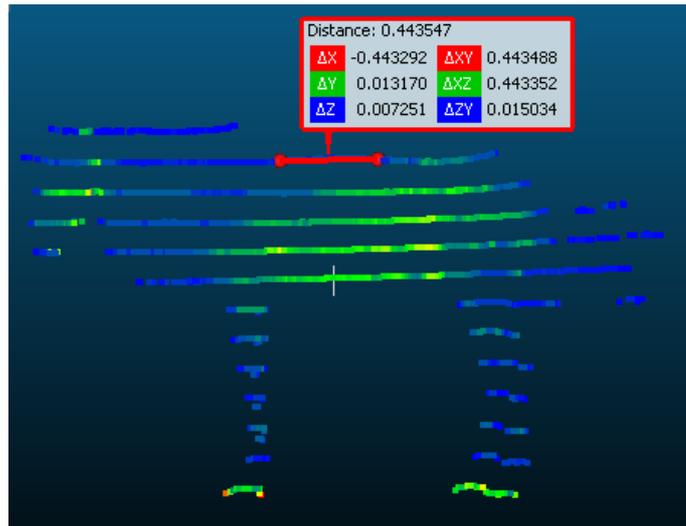


Figura 4-10. Medida zoométrica Longitud del dorso

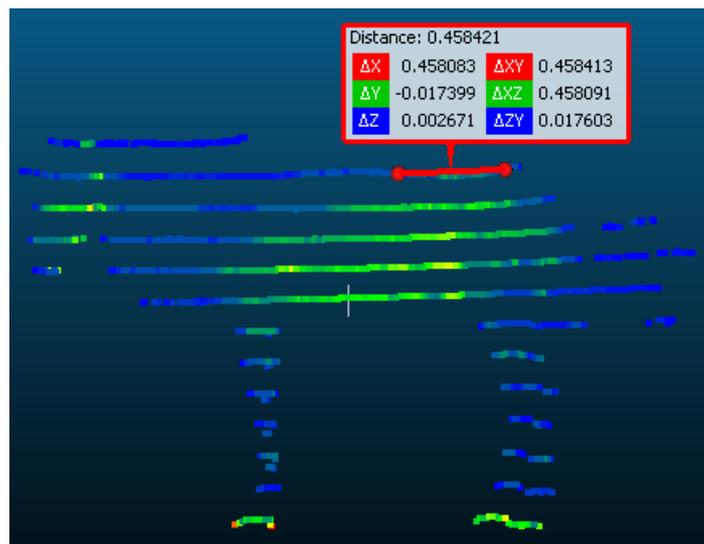


Figura 4-11. Medida zoométrica Longitud del lomo

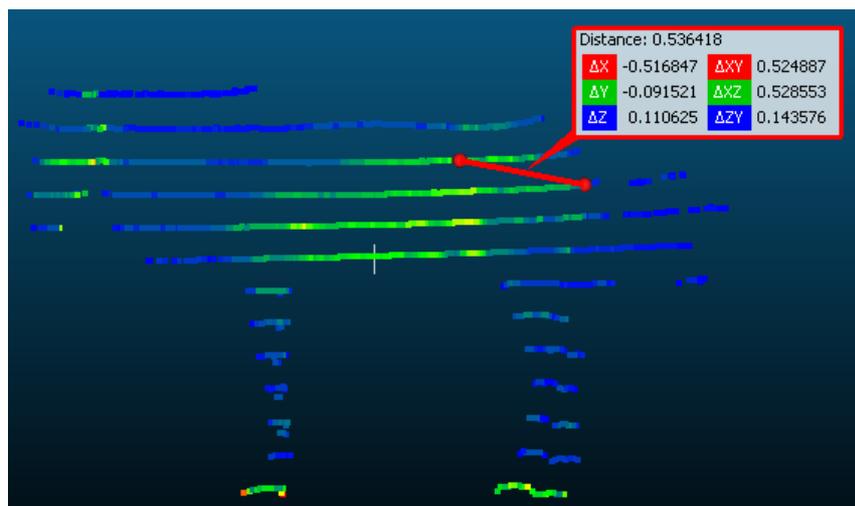


Figura 4-12. Medida zoométrica Longitud de la grupa

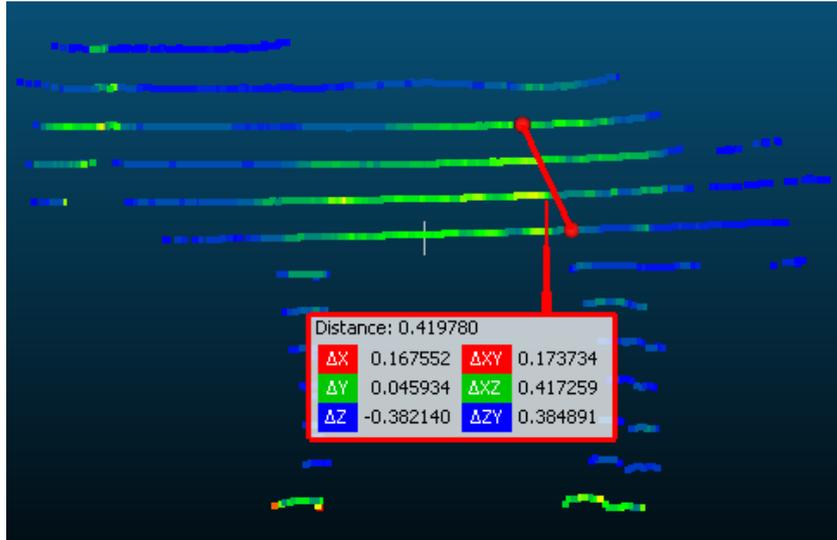


Figura 4-13. Medida zoométrica Distancia anca-babilla

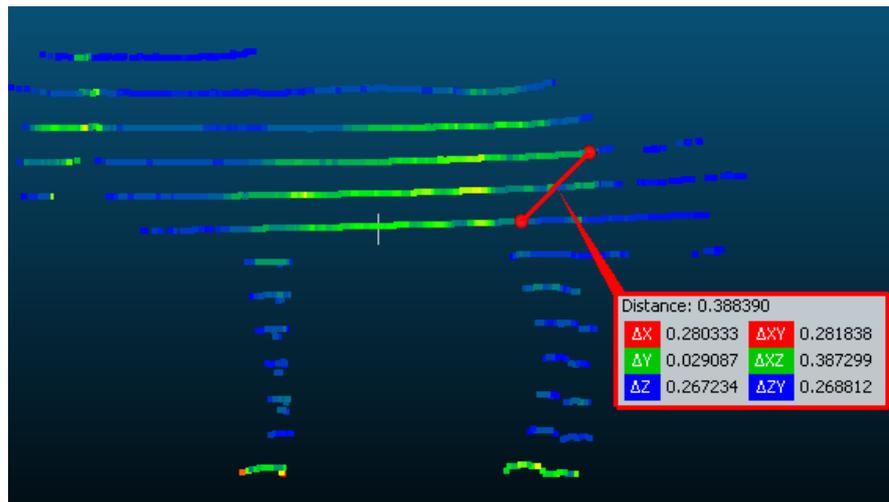


Figura 4-14. Medida zoométrica Distancia nalga-babilla

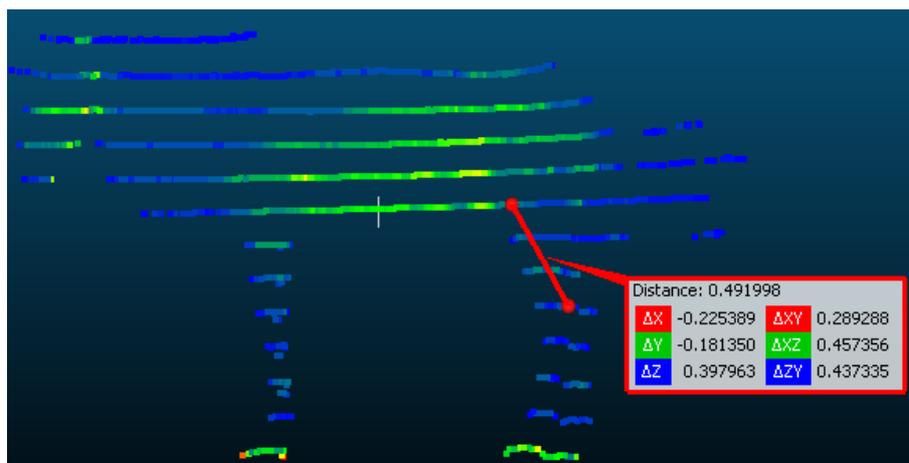


Figura 4-15. Medida zoométrica Longitud de la pierna

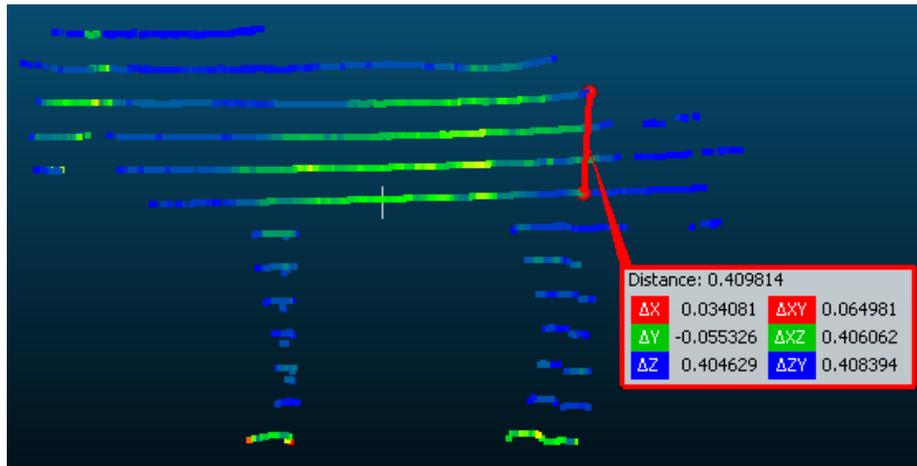


Figura 4-16. Medida zoométrica Longitud de la nalga

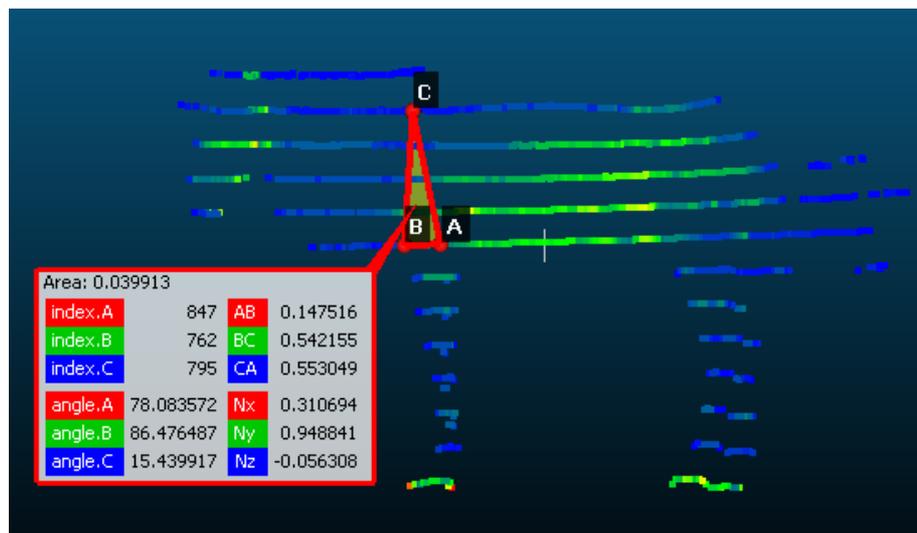


Figura 4-17. Medida zoométrica Ángulo de la espalda

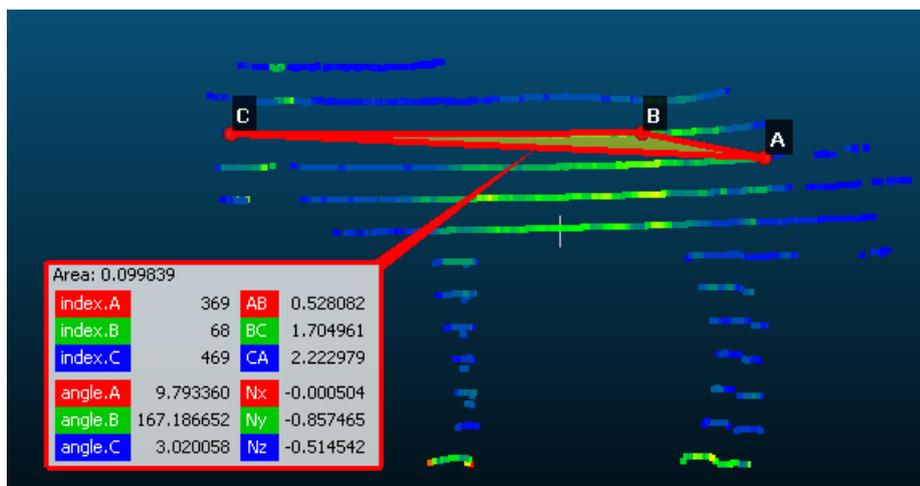


Figura 4-18. Medida zoométrica Ángulo de la grupa

En la siguiente tabla se recogen las medidas zoométricas obtenidas manualmente y mediante el sensor LiDAR de Velodyne. Como se puede observar, algunas celdas correspondientes a las medidas obtenidas con el sensor están vacías debido a que con el método de adquisición de medidas a través de un único disparo con el sensor (*o frame*) el cual solamente se escanea un lado del caballo no es posible obtener información de todos los puntos del caballo.

Tabla 4-1. Medidas zoométricas del caballo ENCANTADO 49 realizadas manualmente y mediante el sensor

<b>ENCANTADO 49</b>			
ZOOMETRÍA (cm)			
Variables zoométricas	Manual	Sensor	Error relativo
Alzada a la cruz	159	149	6,29
Altura al punto más declive de la cruz	150	148	1,33
Alzada a las palomillas	159	155	2,52
Longitud escápulo isquial	155	155	0,00
Longitud de la cabeza	58	57	1,72
Anchura de la cabeza	23	22	4,35
Longitud del cuello	73	70	4,11
Anchura del pecho	73		
Longitud de la espalda	62	54	12,90
Longitud del antebrazo	47	56	19,15
Diámetro dorso esternal	71	54	23,94
Diámetro bicostal	42		
Longitud del dorso	28	44	57,14
Longitud del lomo	32	45	40,63
Anchura de la grupa	52		
Longitud de la grupa	54	54	0,00
Distancia anca-babilla	50	42	16,00
Distancia nalga-babilla	49	39	20,41
Longitud de la pierna	48	49	2,08
Longitud de la nalga	45	41	8,89
Comisura labial	9		
Perímetro torácico	184		
Perímetro de la rodilla	34		
Perímetro de la caña anterior	21		
Ángulo de la espalda	57	86	50,88
Ángulo de la grupa	18	10	44,44
Índice de proporcionalidad			

Se ha utilizado la fórmula del error relativo para comparar las medidas de forma manual y mediante el sensor y conocer el error que se comente con este método basado en el sensor de reconstrucción tridimensional (3D) para mediciones de animales sin contacto.

$$Error\ relativo = \frac{Medida\ manual - Medida\ sensor}{Medida\ manual} * 100$$

## CONCLUSIÓN

---

Una vez presentada la metodología propuesta en este proyecto y analizados los resultados obtenidos a partir de dicho proceso, se ha llegado al desarrollo final de dos conclusiones, una desde el punto de vista genérico y otra desde el punto de vista específico, además de una hoja de ruta para futuros desarrollos

Desde el punto de vista genérico se ha creado una metodología que cumpla con los objetivos de establecer un proceso de evaluación de datos, de manera que en un futuro se puedan desarrollar aplicaciones en las que podamos englobar todo el proceso. Por una parte, se han simplificado la toma de medidas, dando lugar a otro método para obtener resultados más óptimos en un control del crecimiento del ganado un con sensor LiDAR. Y finalmente es una metodología que especifica, es decir, se trata de un proceso claro, preciso y lógico.

Desde el punto de vista específico se obtienen las siguientes conclusiones basadas en los resultados obtenidos al poner en práctica la metodología. El enfoque llevado a cabo en este trabajo verifica la viabilidad de la medición sin contacto de ganado de gran tamaño. El conocimiento del crecimiento saludable y bienestar animal da lugar a una mejora de la calidad de los animales y de la cría genética. Sin embargo, como se ha podido comprobar en el apartado de resultados, debido al constante movimiento de los caballos y a la limitación de analizar un único frame, no es posible analizar completamente la imagen trimidimensional del caballo. La exactitud geométrica de estos sensores está íntimamente relacionada con la calidad de su orientación externa la cual es esencial. Esta orientación externa es la que obtendremos de los sistemas de navegación integrados (GPS e INS). El escáner láser mide solamente el vector orientado desde la abertura del sistema láser a un punto objeto normalmente en el terreno, el caballo. Los puntos tridimensionales solo se pueden calcular si en cualquier instante se conoce la posición y orientación del sensor LiDAR con respecto a un sistema de coordenadas.

Para este problema, en un futuro próximo se va a seguir optimizando este sistema mediante la sincronización del sensor LiDAR y un sistema de navegación integrado que permita el desplazamiento del sensor para así poder rodear al caballo mientras se está escaneando. Esto permite una reconstrucción íntegra del caballo en 3D para una mayor precisión a la hora de obtener las variables zoométricas.

Tabla A-1. Medidas zoométricas del caballo FESTEJADA 10 realizadas manualmente y mediante el sensor

<b>FESTEJADA 10</b>			
ZOOMETRÍA (cm)			
<b>Variables zoométricas</b>	<b>Manual</b>	<b>Sensor</b>	<b>Error relativo</b>
Alzada a la cruz	160	149	6,88
Altura al punto más declive de la cruz	149	146	2,01
Alzada a las palomillas	160	154	3,75
Longitud escápulo isquial	159	158	0,63
Longitud de la cabeza	61	57	6,56
Anchura de la cabeza	21	17	19,05
Longitud del cuello	78	78	0,00
Anchura del pecho	47		
Longitud de la espalda	66	73	10,61
Longitud del antebrazo	44	37	15,91
Diámetro dorso esternal	74	66	10,81
Diámetro bicostal	58		
Longitud del dorso	36		100,00
Longitud del lomo	28		100,00
Anchura de la grupa	59		
Longitud de la grupa	52	54	3,85
Distancia anca-babilla	48	41	14,58
Distancia nalga-babilla	52	35	32,69
Longitud de la pierna	57	62	8,77
Longitud de la nalga	42	39	7,14
Comisura labial	10		
Perímetro torácico	198		
Perímetro de la rodilla	30		
Perímetro de la caña anterior	20		
Ángulo de la espalda	58	83	43,10
Ángulo de la grupa	17	21	23,53
Índice de proporcionalidad			

Tabla A-2. Medidas zoométricas del caballo VALIOSA 8 realizadas manualmente y mediante el sensor

<b>VALIOSA 8</b>			
ZOOMETRÍA (cm)			
<b>Variables zoométricas</b>	<b>Manual</b>	<b>Sensor</b>	<b>Error relativo</b>
Alzada a la cruz	163	150	7,98
Altura al punto más declive de la cruz	153	150	1,96
Alzada a las palomillas	162	161	0,62
Longitud escápulo isquial	170	156	8,24
Longitud de la cabeza	62	60	3,23
Anchura de la cabeza	23	25	-8,70
Longitud del cuello	78	75	3,85
Anchura del pecho	46		
Longitud de la espalda	69	70	-1,45
Longitud del antebrazo	47	51	8,51
Diámetro dorso esternal	77	67	12,99
Diámetro bicostal	60		
Longitud del dorso	56		100,00
Longitud del lomo	32		100,00
Anchura de la grupa	59		
Longitud de la grupa	53	55	3,77
Distancia anca-babilla	45	42	6,67
Distancia nalga-babilla	53	49	7,55
Longitud de la pierna	54	55	1,85
Longitud de la nalga	45	47	4,44
Comisura labial	10		
Perímetro torácico	201		
Perímetro de la rodilla	32		
Perímetro de la caña anterior	21		
Ángulo de la espalda	57	65	14,04
Ángulo de la grupa	18	20	11,11
Índice de proporcionalidad			

Tabla A-3. Medidas zoométricas del caballo FESTEJADO 14 realizadas manualmente y mediante el sensor

<b>FESTEJADO 14</b>			
ZOOMETRÍA (cm)			
<b>VARIABLES ZOOMÉTRICAS</b>	<b>Manual</b>	<b>Sensor</b>	<b>Error relativo</b>
Alzada a la cruz	164	156	4,88
Altura al punto más declive de la cruz	153	155	-1,31
Alzada a las palomillas	159	140	11,95
Longitud escápulo isquial	159	153	3,77
Longitud de la cabeza	59	62	-5,08
Anchura de la cabeza	24	23	4,17
Longitud del cuello	79	73	7,59
Anchura del pecho	40		
Longitud de la espalda	62	67	-8,06
Longitud del antebrazo	49	40	18,37
Diámetro dorso esternal	74	62	16,22
Diámetro bicostal	44		
Longitud del dorso	30		100,00
Longitud del lomo	34		100,00
Anchura de la grupa	52		
Longitud de la grupa	52	51	1,92
Distancia anca-babilla	51	48	5,88
Distancia nalga-babilla	52	53	1,92
Longitud de la pierna	50	50	0,00
Longitud de la nalga	48	47	2,08
Comisura labial	8		
Perímetro torácico	185		
Perímetro de la rodilla	33		
Perímetro de la caña anterior	21		
Ángulo de la espalda	56	62	10,71
Ángulo de la grupa	18	23	27,78
Índice de proporcionalidad			

Tabla A-4. Medidas zoométricas del caballo REVOLTOSO 85 realizadas manualmente y mediante el sensor

<b>REVOLTOSO 85</b>			
ZOOMETRÍA (cm)			
<b>Variables zoométricas</b>	<b>Manual</b>	<b>Sensor</b>	<b>Error relativo</b>
Alzada a la cruz	156	160	-2,56
Altura al punto más declive de la cruz	151	141	6,62
Alzada a las palomillas	158	145	8,23
Longitud escápulo isquial	156	150	3,85
Longitud de la cabeza	60	63	-5,00
Anchura de la cabeza	23	29	-26,09
Longitud del cuello	74	78	-5,41
Anchura del pecho	41		
Longitud de la espalda	62	53	14,52
Longitud del antebrazo	54	54	0,00
Diámetro dorso esternal	72	54	25,00
Diámetro bicostal	42		
Longitud del dorso	27		100,00
Longitud del lomo	38		100,00
Anchura de la grupa	52		
Longitud de la grupa	56	50	10,71
Distancia anca-babilla	53	44	16,98
Distancia nalga-babilla	52	48	7,69
Longitud de la pierna	51	64	25,49
Longitud de la nalga	47	45	4,26
Comisura labial	8		
Perímetro torácico	186		
Perímetro de la rodilla	33		
Perímetro de la caña anterior	20		
Ángulo de la espalda	57	66	15,79
Ángulo de la grupa	14	25	78,57
Índice de proporcionalidad			

# REFERENCIAS

---

- [1] Velodyne LIDAR. User's manual and programming guide.
- [2] Raspberry Pi. Raspberry Pi 3 Model B+: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
- [3] Element14. Raspberry Pi Model B GPIO 40 Pin Block Piout, <https://www.element14.com/community/docs/DOC-73950/1/raspberry-pi-3-model-b-gpio-40-pin-block-pinout>, 29 de febrero del 2016
- [4] SlideShare. Introducción a Ros: <https://es.slideshare.net/pibgeus/introduccion-a-ros>, 15 de diciembre del 2010
- [5] ArcMap. ¿Qué son los datos LIDAR?: <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/las-dataset/what-is-lidar-data-.htm>. 2019
- [6] INCyTU (Oficina de Información Científica y Tecnológica para el Congreso de la Unión). Ganadería de Precisión. N° 23. Disponible en: [https://www.foroconsultivo.org.mx/INCYTU/documentos/Completa/INCYTU\\_18-023.pdf](https://www.foroconsultivo.org.mx/INCYTU/documentos/Completa/INCYTU_18-023.pdf). Agosto del 2018
- [7] Lvwen Huang, Shuqin Li, Anqi Zhu, Xinyun Fan, Chenyang Zhang y Hongyan Wang. Non-Contact Body Measurement for Qinchuan Cattle with LiDAR Sensor: <https://www.semanticscholar.org/paper/Non-Contact-Body-Measurement-for-Qinchuan-Cattle-Huang-Li/c1f1ddfb0db85dde261619a2451f9a8019ff9e68>
- [8] Wikipedia. Localización y modelado simultáneo: [https://es.wikipedia.org/wiki/Localización\\_y\\_modelado\\_simultáneos](https://es.wikipedia.org/wiki/Localización_y_modelado_simultáneos), 29 de agosto de 2019
- [9] Frank Moosmann and Christoph Stiller. Velodyne slam. Publicado en 2011 IEEE (Institute of Electrical and Electronics Engineers) Intelligent Vehicles Symposium (IV), páginas 393–398. IEEE, 2011
- [10] Wikipedia. Sistema Operativo Robótico: [https://es.wikipedia.org/wiki/Sistema\\_Operativo\\_Robótico](https://es.wikipedia.org/wiki/Sistema_Operativo_Robótico), 30 de agosto del 2019
- [11] ROS Wiki. ROS/Concepts: <http://wiki.ros.org/ROS/Concepts>, 21 de junio del 2014.
- [12] ROS Wiki. tf: <http://wiki.ros.org/tf>, 2 de octubre del 2016.
- [13] Tully Foote. tf: The transform library. Publicado en 2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA), páginas 1–6. IEEE, 2013
- [14] ROS Wiki. actionlib: <http://wiki.ros.org/actionlib>, 30 de octubre del 2018
- [15] ROS Wiki. diagnostic\_msgs: [http://wiki.ros.org/diagnostic\\_msgs](http://wiki.ros.org/diagnostic_msgs)
- [16] ROS Wiki. geometry\_msgs: [http://wiki.ros.org/geometry\\_msgs](http://wiki.ros.org/geometry_msgs)
- [17] ROS Wiki. urdf: <http://wiki.ros.org/urdf>, 1 de noviembre del 2019.
- [18] GitHub. Releases – ros/cheatsheet: <https://github.com/ros/cheatsheet/releases>. 11 de mayo del 2015
- [19] ROS Wiki. rosbag/Commandline: <http://wiki.ros.org/rosbag/Commandline>. 27 de marzo del 2019
- [20] ROS.org: <https://www.ros.org/>
- [21] ROS Wiki. velodyne\_pointcloud: [http://wiki.ros.org/velodyne\\_pointcloud](http://wiki.ros.org/velodyne_pointcloud). 25 de mayo del 2016
- [22] ROS Wiki. velodyne\_driver: [http://wiki.ros.org/velodyne\\_driver](http://wiki.ros.org/velodyne_driver). 28 de enero del 2019
- [23] ROS Wiki. velodyne/Tutorials/Getting Started with the Velodyne VLP16: <http://wiki.ros.org/velodyne/Tutorials/Getting%20Started%20with%20the%20Velodyne%20VLP16>. 9 de

enero del 2019

[24] ROS Wiki. melodic/Installation/Ubuntu: <http://wiki.ros.org/melodic/Installation/Ubuntu>. 27 de julio del 2019

[25] CloudCompare. Interactive Segmentation Tool.

[https://www.cloudcompare.org/doc/wiki/index.php?title=Interactive\\_Segmentation\\_Tool](https://www.cloudcompare.org/doc/wiki/index.php?title=Interactive_Segmentation_Tool). 27 de agosto del 2018

[26] CloudCompare. PointPicking. [https://www.cloudcompare.org/doc/wiki/index.php?title=Point\\_picking](https://www.cloudcompare.org/doc/wiki/index.php?title=Point_picking). 20 de febrero del 2015



