



Construcción Escáner LIDAR

Tomás de Camino Beck, Ph.D.

Introducción

El escáner LIDAR es un escáner de lecturas del sensor de distancia láser de alta resolución. El escáner rota 360 grados haciendo lecturas de distancia cada ciertos intervalos, y registrando el ángulo con respecto al norte magnético. La distancia máxima de lectura del LIDAR es de 50 metros.

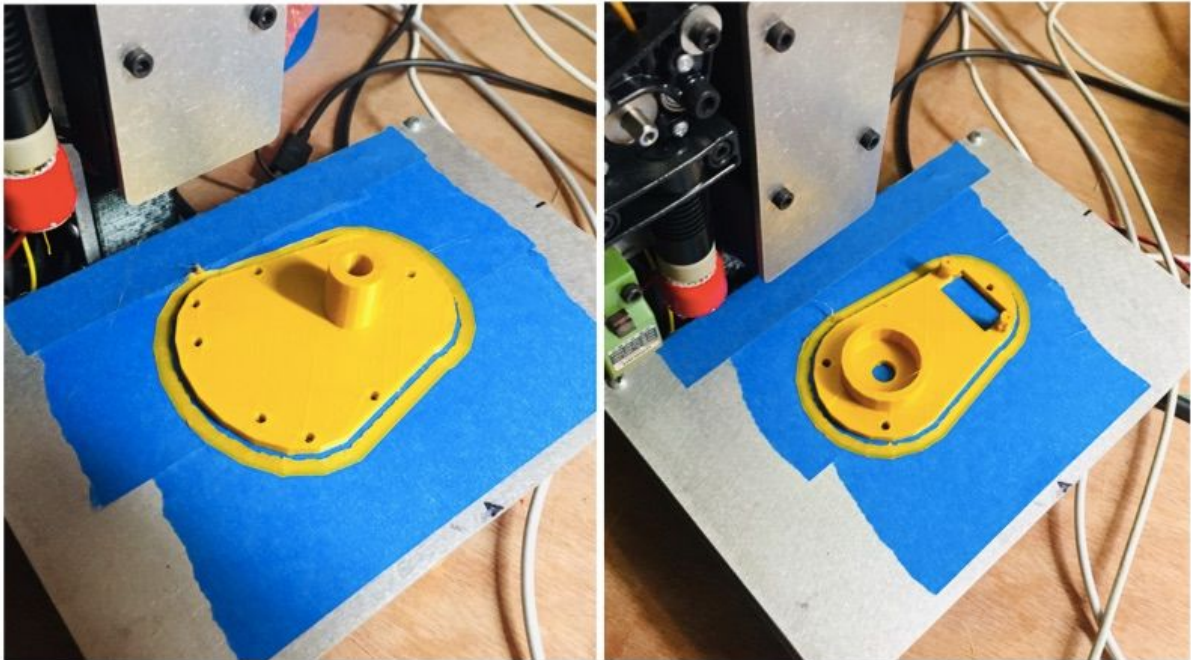
La función del escáner es detectar una nube de puntos para luego determinar la presencia de árboles en un rodal, y posteriormente calcular diámetros y áreas basales. este dispositivo se encuentra en fase de prototipo, por tanto es necesario realizar ensayos de campo para validar su funcionamiento.

Materiales

- Rol genérico de patineta
- Microcontrolador (en este documento se utiliza un S4A)
- Servomotor de rotación continua
- Extensiones (stand off)
- Slip ring de 6 cables
- LIDAR lite
- ADAfruit 9DOF
- Impresora 3D
- Taco anclaje metal 1/2"
- Elásticos varios

Impresión 3D

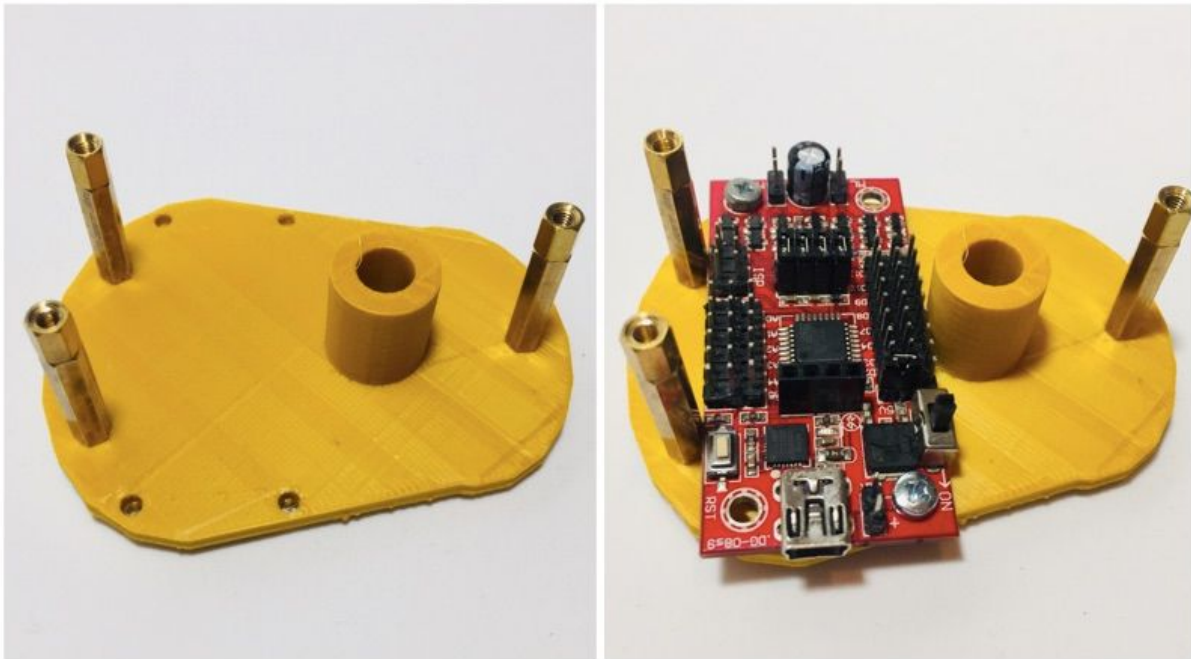
Imprimir en impresora 3D en resolución de 300 micras (0.3mm) las piezas del siguiente repositorio [GitHub](#), como se ve en la fotos.



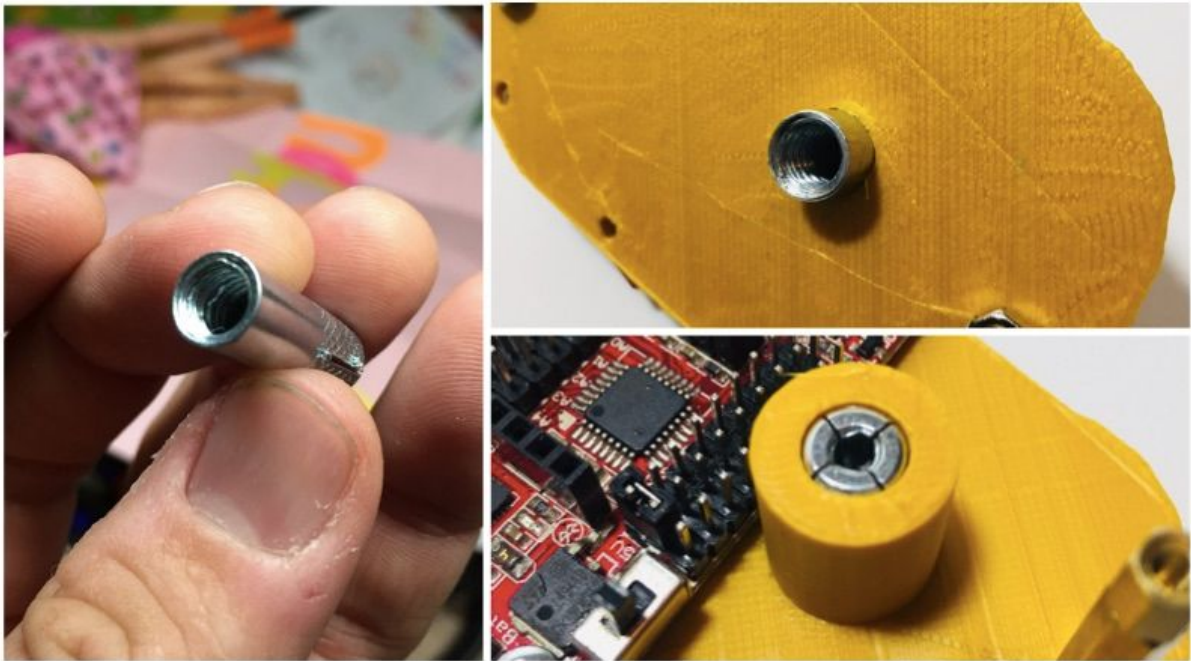
Para la impresión del prototipo se utilizó plástico PLA, con una impresora Printbot Smalls.

Ensamblado de la base

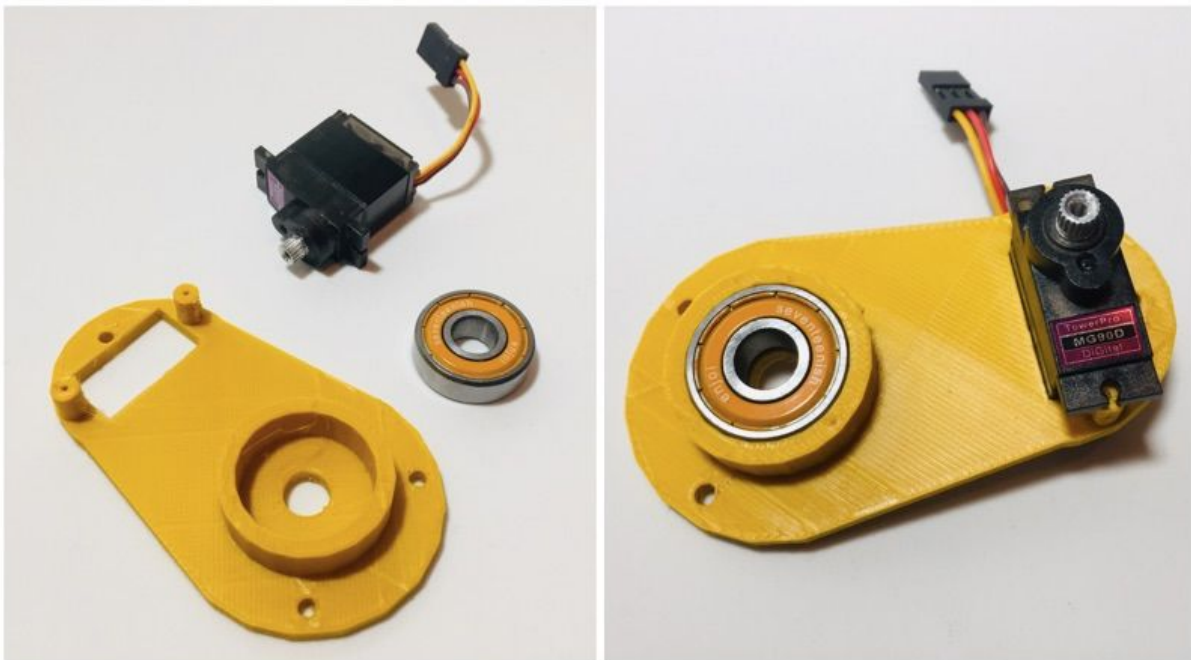
Colocar el microcontrolador y fijarlo con tornillos a la base impresa en 3D. Colocar las extensiones. En este caso se utilizan extensiones de 2cm y una adicional de 6mm. (ver fotos)



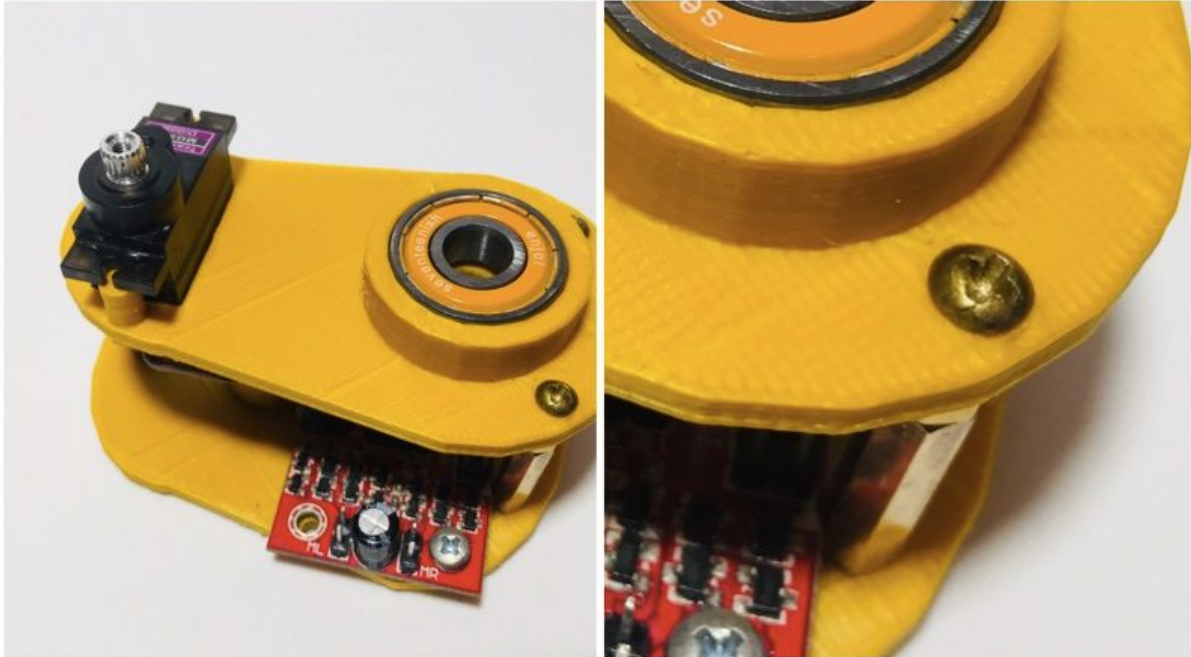
Importante colocar el anclaje de metal. El anclaje permite luego utilizar un trípode fotográfico para fijar el escáner. En la siguiente foto se muestra cómo colocarlo



En la segunda base, montar el servomotor de rotación continua y fijarlo con tornillos como se observa en la foto.

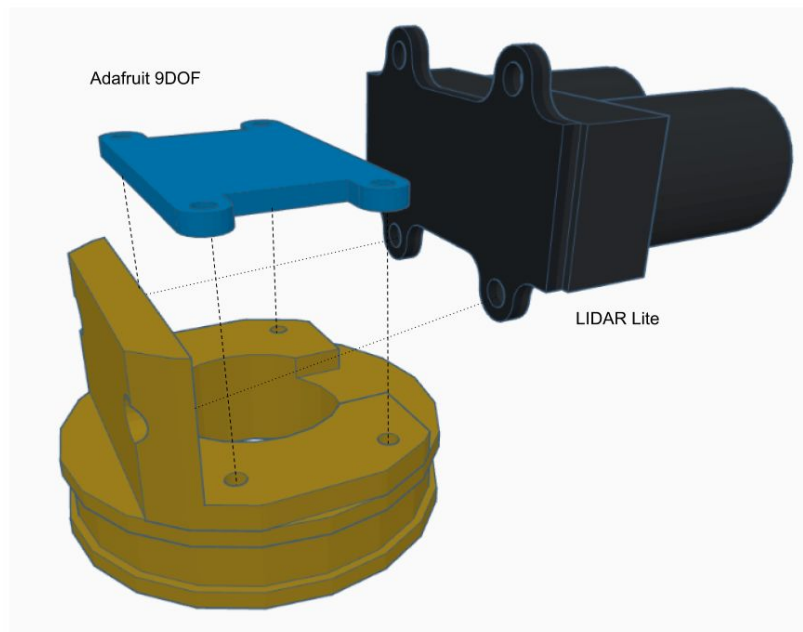


Una vez que las dos bases estén ensambladas, atornillar la base superior a la inferior como se muestra en la siguiente foto

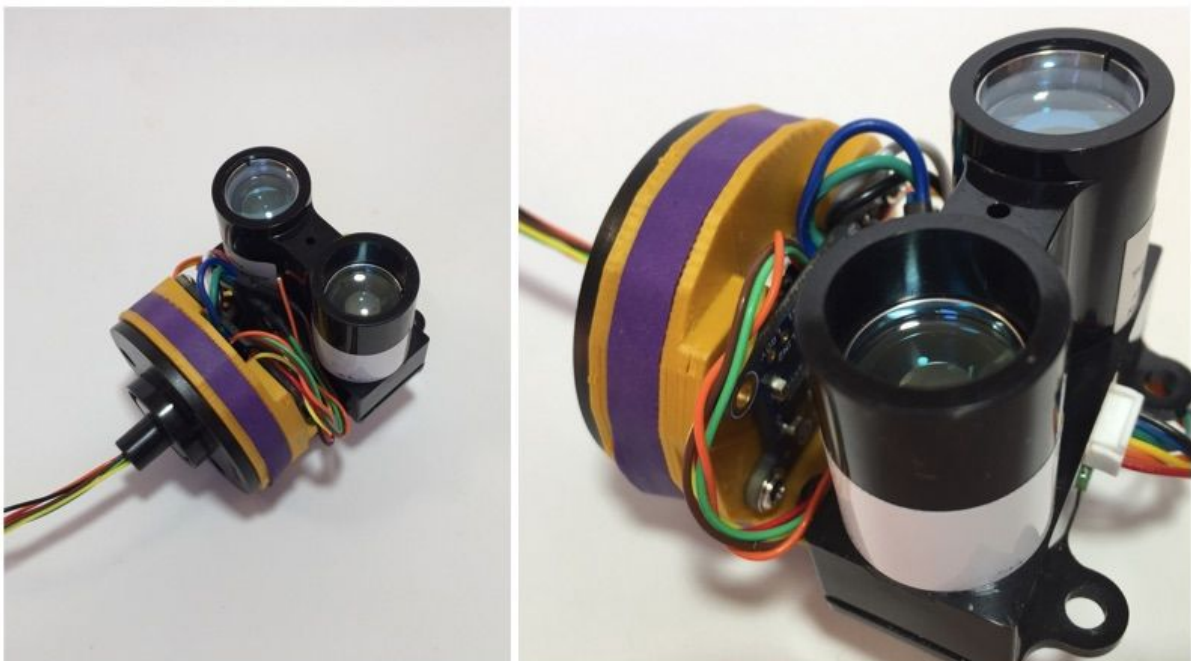


Ensamblado del Lidar y magnetómetro

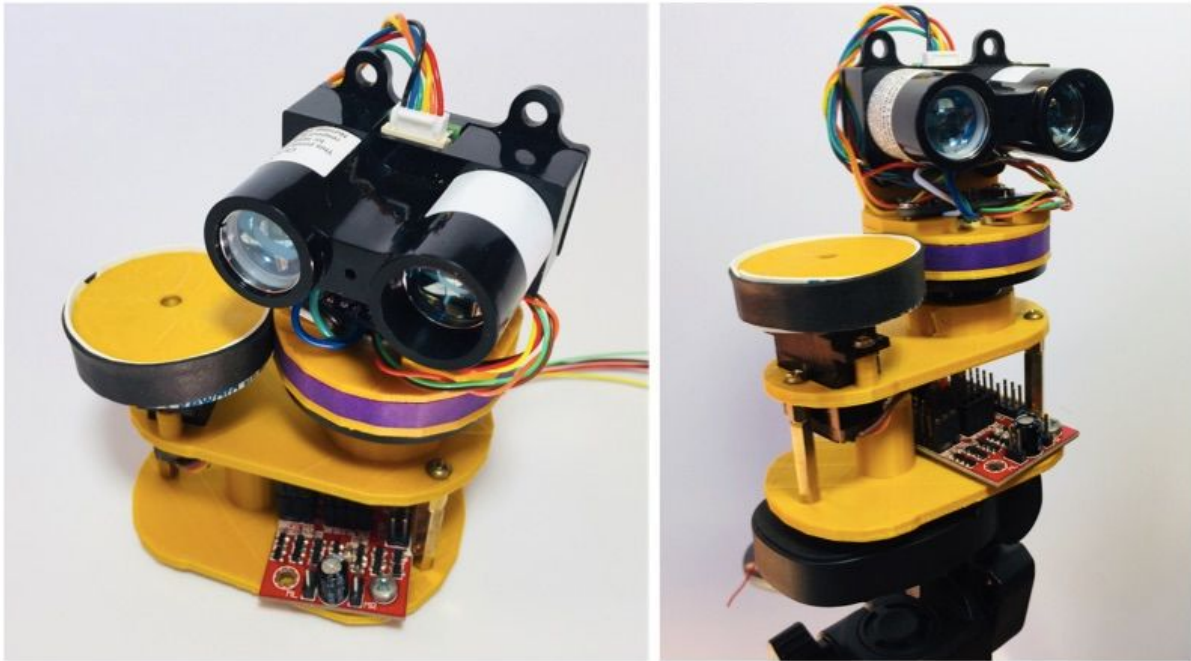
El LIDAR y el magnetómetro van colocados sobre el “slip ring” que permite la rotación de cables y que ambos sensores se muevan libremente. Ambos sensores se colocan sobre la pieza impresa de PLA como se ve en el siguiente esquema



Ensamblar el LIDAR, con el magnetómetro como se observa en la fotografía

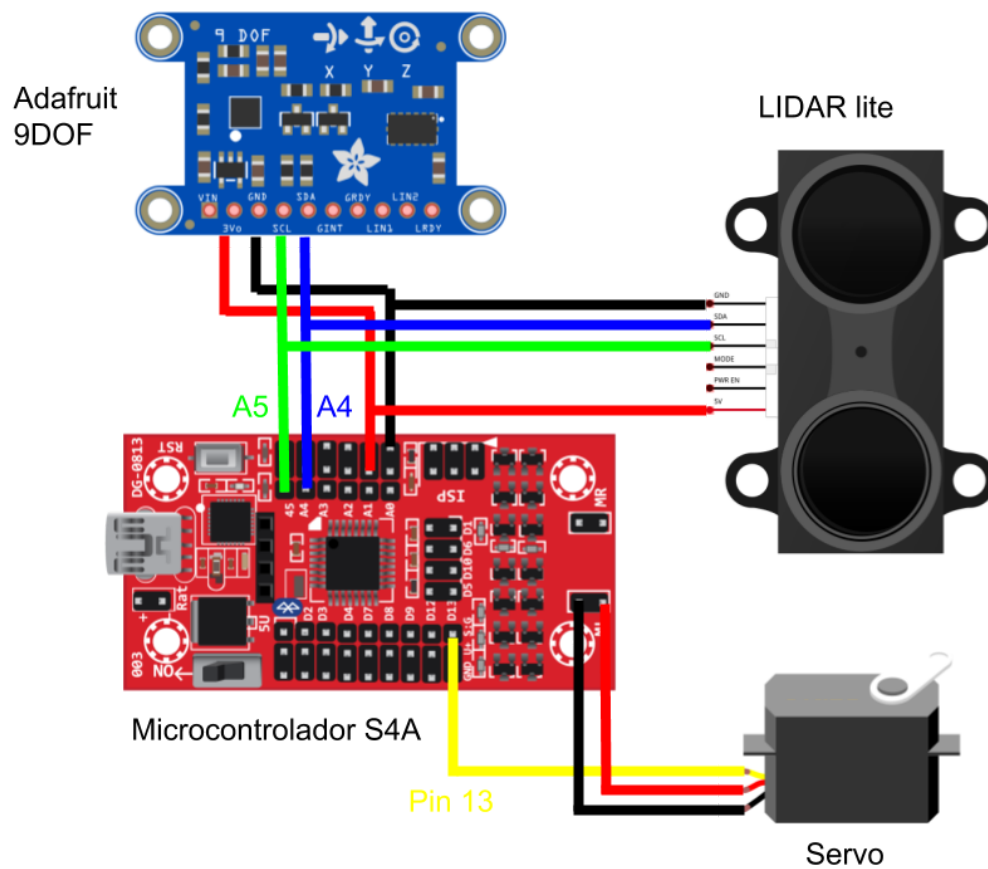


Una vez colocados ambos sensores, se colocan junto con la rueda del servo sobre la base,



Hardware y código

El hardware se conecta de manera bastante simple. Las conexiones SDA, SCL, del LIDAR van conectadas directamente con las conexiones SDA y SCL del magnetómetro 9DOF. De igual manera las conexiones de GND y 3V. Estas se conectan a los cables de “Slip Ring”, y de allí a las conexiones de SDA (A4) y SCL (A5) del controlador S4A. El servomotor se conecta directamente al S4A en el pin número 9, pero la alimentación de poder se conecta a ML en el S4A. El esquema de conexión se puede ver en el siguiente esquema

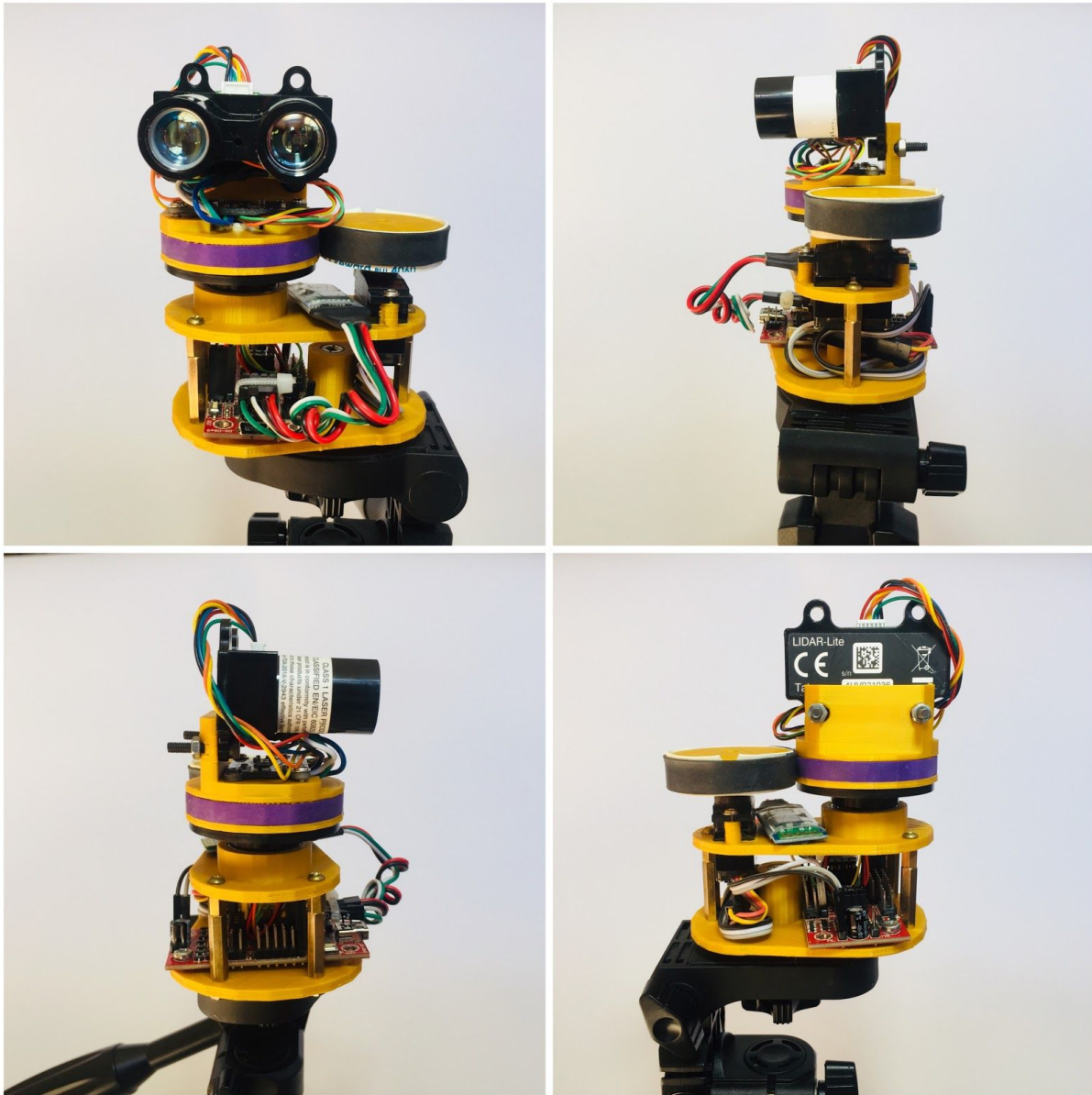


El código se encuentra en el ANEXO, o en el siguiente repositorio de GitHub¹.

Operación

Colocar el escáner sobre el trípode

¹ <https://github.com/FunCR/ProyectoCATIE/tree/master/TreeScanner/Code>



- Conectar el puerto USB del escáner a la computadora. El escáner comenzará a rotar y a tomar datos de forma inmediata. Para ver los datos, abrir el Arduino IDE, y abrir el monitor serial.
- Los datos son arrojados por el escáner en conexión serial a 9600 Baudios. Los datos se leen en formato “ángulo (grados), distancia (en cm)”. El ángulo es la lectura del magnetómetro que corresponde a los grados con respecto al norte magnético.
- El escáner se debe dejar leyenda por varios minutos para que recoja una “nube” de puntos suficientemente densa, para luego detectar la presencia de



árboles en un rodal. El LIDAR hace una lectura de 1, cuando no puede determinar la distancia.

- Los datos recogidos en la terminal serial del Arduino IDE, se puede copiar y pegar en un archivo de texto (sin formato), para luego se procesados (recomendado)
- Del mismo modo, se puede leer los datos de forma visual a través del Lidar Viewer, que se desarrolló como parte de este proyecto y que corre en la plataforma Processing.org.



LIDAR Viewer

Tomás de Camino Beck, Ph.D.

Introducción

Lidar viewer es el prototipo de un visualizador de datos del escáner LIDAR. El software se desarrolló en Processing.org una plataforma de código abierto que permite el desarrollo de prototipos de software rápidamente, con códigos escritos en JAVA. Esta plataforma además permite ir modificando el código y distribuyendolo en forma rápida, pues corre de manera transparente en cualquier versión instalada de Processing.

El LIDAR Viewer, permite la visualización de datos crudos del escáner LIDAR, pero además implementa un algoritmo de agregación que permite identificar estructuras constantes en el escaneo. En el caso de esta aplicación, permite la identificación de árboles. A partir de identificar las estructuras o árboles, es necesario una investigación para establecer la relación entre puntos de borde de estructuras y estimación de diámetro.

El software se conecta al Escáner LIDAR a través del puerto serial con un módulo Bluetooth 2.0 que se encuentra en el escáner. Sin embargo también es posible visualizar los datos a través de cable USB

El código se encuentra en prototipo, y está sujeta a cambios. El código para correr en Processing, se puede descargar en el repositorio GitHub² (Se adjunta como ANEXO).

Recepción de datos

Los datos son recibidos por el LiDAR Viewer, como tuplas de datos,

²

https://github.com/FunCR/ProyectoCATIE/blob/master/TreeScanner/Code/LIDAR_View/LIDAR_Viewer.pde



$$datum = \langle angle, distance \rangle$$

Una nube de puntos lo constituye un conjunto grande de puntos ángulo distancia. Mediante esta nube de puntos es posible reconstruir información espacial en 2 dimensiones.

Cada 50 milisegundos el software recibe los datos. estos datos se pueden visualizar crudos (toda la nube de puntos) o como agregados de datos

Algoritmo

La meta del algoritmo es crear “clusters” de puntos para identificar estructuras permanentes (árboles). La estrategia para lograrlo es agrupar los puntos por proximidad espacial, es decir, se asume que cuando un fuste es detectado, existirá un grupo de puntos muy cercanos, con los bordes marcados por puntos distantes.

Los valores ángulo distancia se convierten a coordenadas cartesianas mediante las siguientes fórmulas,

$$\langle x, y \rangle = \langle \cos(angle) \times distance, \sin(angle) \times distance \rangle$$

Cada par ordenado x, y representa un punto p_i . La distancia entre dos puntos se calcula mediante la distancia euclidiana definida como,

$$|p_1, p_2| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Donde p_1 y p_2 representan dos puntos cualesquiera en la nube de puntos.

Consideremos el siguiente conjunto de datos agrupados,

$$\mathbb{D} = \{\bar{x}, \bar{y}, c\}$$

Donde x y y son un promedio de puntos dentro de un rango de distancia predeterminado (centroide), y c es el conteo de puntos con ese centroide. Las estructuras predominantes o más “sólidas” tendrán un conteo de puntos más alto. Al escanear un bosque, los troncos de árboles serán estructuras predominantes.

El pseudocódigo del algoritmo es el siguiente,

Require: $datum = \langle angle, distance \rangle$

Ensure: $distance > 1$

$x \leftarrow \cos(angle) \times distance$

$y \leftarrow \sin(angle) \times distance$

$D \neq \{\}$

$i = 0$

while $p_i \in D$ **do**

$p \leftarrow \langle x, y \rangle$

if $|p, p_i| < threshold$ **then**

$\langle p_i.x, p_i.y \rangle \leftarrow \langle \overline{x}, \overline{p_i.x}, \overline{y}, \overline{p_i.y} \rangle$

$p_i.c \leftarrow p_i.c + 1$

$i \leftarrow i + 1$

else

$D \leftarrow D + p$

end if

end while

El código completo en JAVA se puede ver en el ANEXO.

Utilización

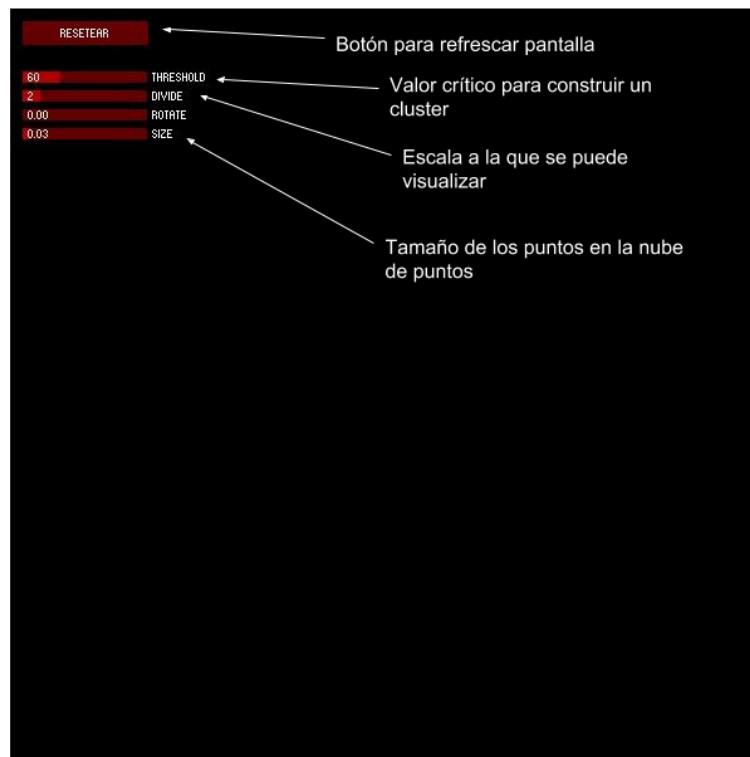
1. Se debe instalar Processing. Para hacerlo, descargar e instalar el programa en el sitio de Processing.org³
2. Descargar el código del repositorio de GitHub⁴.
3. EN processing, abrir el código de Lidar_Viever.

³<https://processing.org/>

⁴ https://github.com/FunCR/ProyectoCATIE/tree/master/TreeScanner/Code/LIDAR_Viever

4. En la línea `String portname = "/dev/tty.HC-06-DevB";` el valor `"/dev/tty.HC-06-DevB"` indica el nombre del puerto a donde se encuentra el bluetooth. Este se indica una vez que hayan hecho el “pairing” con el bluetooth “HC-06” del escáner Lidar.
5. Una vez que se aseguren que el Bluetooth está registrado en la computadora, presionar el botón de play en processing.

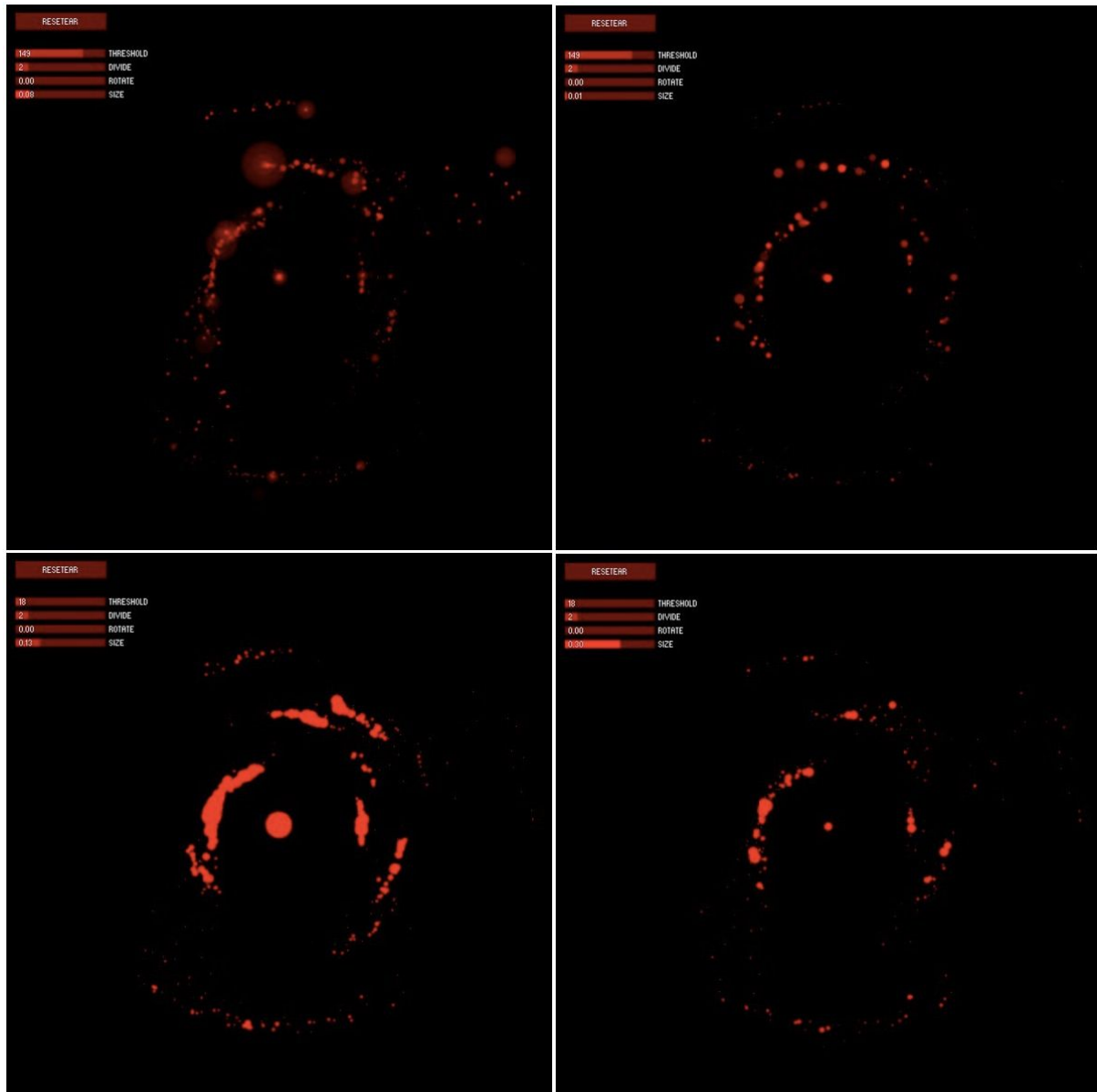
Interfaz



Una vez hecha la conexión con el USB, el programa comenzará a desplegar la nube de puntos. Los puntos de un cluster, se verán como círculos cada vez más grandes. El tamaño de esos círculos se puede cambiar con el slider etiquetado “SIZE”. Si se mueve el slider “THRESHOLD” se puede cambiar la tolerancia de la distancia euclidiana para formar los clusters.

El Slider “DIVIDE” simplemente es la escala a la cual despliega los puntos, y el botón “RESETEAR” permite refrescar la pantalla.

Luego de algunos minutos la pantalla se verá de la siguiente manera,



Cada “banda” que se forma, representa un árbol. En la imagen se muestran varias configuraciones de formación de cluster.

Extensiones Necesarias



- Con los puntos agrupados, es necesario extender el análisis para determinar los diámetros. Este trabajo requiere un ensayo experimental tanto en campo, como en medidas diamétricas exactas para calibrar el algoritmo. Con medidas diamétricas exactas, nos referimos a estructuras cilíndricas de diámetro exacto que son escaneadas por el escáner LIDAR, para determinar los diámetros y validar el algoritmo de reconocimiento de troncos, así como extenderlo para la medición de diámetros y cálculo de área basal y de diámetros.
- La forma de exportar datos todavía no es directa, para ellos hay que grabar los datos en la consola terminal y exportar a archivo separado por comas