

DISEÑO E IMPLEMENTACIÓN DE UN ESCANER 3D PARA LA  
RECONSTRUCCIÓN DE UNA SUPERFICIE A TRAVÉS DE UNA NUBE DE  
PUNTOS.

DISEÑO E IMPLEMENTACIÓN DE UN ESCANER 3D PARA LA  
RECONSTRUCCIÓN DE UNA SUPERFICIE A TRAVÉS DE UNA NUBE DE  
PUNTOS.

Diego Arévalo Ovalle

EMILIA DEL PILAR ÁLVAREZ MENESES

JOHN GABRIEL MUÑOZ CRUZ

Trabajo de grado para optar al título de:

ESPECIALISTA EN MATEMÁTICA APLICADA EN SISTEMAS  
COMPUTACIONALES

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO

PROYECTO DE GRADO

BOGOTÁ D.C.

2016

## Contenido

TABLA DE ILUSTRACIONES .....	5
1. PLANTEAMIENTO DEL PROBLEMA.....	6
2. JUSTIFICACIÓN .....	6
3. OBJETIVOS.....	7
3.1 General .....	7
3.2 Específicos.....	7
4. MARCO TEÓRICO.....	7
4.1 Escáner 3D .....	7
4.2 Coordenadas cilíndricas .....	8
4.3 Laser .....	8
4.4 Cámara WEB.....	9
4.5 Motor paso a paso .....	9
4.6 Arduino .....	9
4.7 Octave.....	10
4.8 Nube de puntos y Meshlab .....	10
5. DESARROLLO DEL PROYECTO .....	10
5.1 CONSTRUCCIÓN DEL HARDWARE .....	10
5.1.1 Chasis.....	10
5.1.2 Movimiento base giratoria .....	11
5.1.3 Adecuación del borde del objeto.....	11
5.1.4 Preparación de la cámara.....	12
5.2 CONSTRUCCIÓN DEL SOFTWARE.....	12
5.2.1 Control del motor y el laser .....	12
5.2.2 Adquisición de la imagen .....	12
5.2.3 Toma correcta de los limites .....	13
5.2.4 Procesamiento de la imagen .....	14
5.2.5 Conversión de coordenadas.....	18
5.2.6 Construcción del modelo en tres dimensiones.....	19
6. DELIMITACIONES .....	19
7. DESARROLLO DE PRUEBAS .....	20

7.1 Casos de uso.....	20
8. CONCLUSIONES.....	21
9. BIBLIOGRAFÍA.....	22
10. CIBERGRAFÍA.....	23
11. ANEXOS.....	24
ANEXO 1. Control de los pasos del motor y el láser en ARDUINO. ....	24
ANEXO 2. Adquisición de la imagen.....	25
ANEXO 3. Adecuación de la imagen.....	26
ANEXO 4. Procesamiento de la imagen.....	26
ANEXO 5. Ciclo.....	27

## TABLA DE ILUSTRACIONES

Ilustración 1. Coordenadas cilíndricas y cartesianas relacionadas. ....	8
Ilustración 2 Movimientos motor paso a paso .....	9
Ilustración 3. Proceso de la construcción del chasis. ....	11
Ilustración 4 Modulo Laser Arduino .....	12
Ilustración 5. Posición entre cámara y laser. ....	13
Ilustración 6 Vistas cámara interna y externa .....	13
Ilustración 7. Guía para tomar límites .....	14
Ilustración 8. Imágenes tomadas por la cámara .....	14
Ilustración 9. Coordenadas originales .....	18
Ilustración 10. Coordenadas modificadas.....	18
Ilustración 11 Curva reconstruida. ....	19
Ilustración 12. Tortuga reconstruida. ....	20
Ilustración 13. Perro reconstruido .....	20
Ilustración 14 Oso reconstruido.....	20
Ilustración 15 Pelota reconstruida. ....	21

## 1. PLANTEAMIENTO DEL PROBLEMA

Escanear ha sido una necesidad de la humanidad desde el momento que se precisa convertir alguna información análoga a digital y de esta manera poder recrearla (Beales, Aug 11, 2006). Aunque su mayor auge se encuentra en la ofimática para lograr obtener documentos físicos en un medio digital, también se ha logrado extender a otros usos como códigos bidimensionales para validar información, identificaciones biométricas entre otros, donde incluso objetos como el celular logra escanear diferentes representaciones según las necesidades.

Sin embargo, hay situaciones que exigen conocer el comportamiento de algún objeto físico con todas sus características donde tomar representaciones en dos dimensiones no son suficientes para describirlas. Un ejemplo claro puede ser una superficie irregular como una pieza arquitectónica de un museo (Sebastian, 2005) donde tomar fotografías no da la suficiente información para poder recrearla completamente, o si se usan las técnicas de dos dimensiones para hacerlo puede ser un proceso arduo y posiblemente no albergue todas las características físicas que tiene el objeto.

El modelamiento de objetos en 3D busca desarrollar una representación matemática de los objetos (Plana, 1997) en un software para hacer análisis o representar algún objeto físico a través de una impresora.

## 2. JUSTIFICACIÓN

Un escáner de tres dimensiones es útil porque puede captar características de objetos reales, diferentes a su forma, también puede distinguir colores, profundidades y demás características según el tipo de escáner que se utilice y la necesidad (Ajandro Marambio, Universitat Politècnica de Catalunya, ).

Las capturas de todas esas propiedades se realizan todas al mismo tiempo, realizando aproximaciones cercanas a los objetos reales lo cual ha permitido importantes avances en lugares donde es necesario reconstruir un objeto en tres dimensiones como un museo arquitectónico (Sebastian, 2005) con piezas irregulares donde se pueden hacer estimaciones de su forma a través de un procesamiento. Otros campos de acción donde se usa este tipo de escáner, son áreas como la medicina donde es necesario identificar el comportamiento de algún órgano del cuerpo humano.

Así mismo, el modelamiento de objetos en tres dimensiones, ha permitido simular objetos intentando asemejarse a la realidad, lo cual se puede evidenciar como las

películas en alta definición, ramas tales como video juegos o la reconstrucción de piezas para la industria a tal punto de hacer uso de impresoras en 3D, para poder visualizarlos afuera del computador.

### 3. OBJETIVOS

#### 3.1 General

Diseñar e implementar un dispositivo que logre escanear y reconstruir una superficie a través de una nube de puntos.

#### 3.2 Específicos

- Construir un escáner de distancia el cual permita tomar las curvas de un objeto.
- Apropiar la conversión de coordenadas al objeto para reconstruir cada curva y representar la superficie mediante una nube de puntos.

### 4. MARCO TEÓRICO

#### 4.1 Escáner 3D

Los escáneres 3D son artefactos que permiten obtener imágenes tridimensionales a partir de objetos existentes, se obtienen a partir de una nube de puntos que se adquieren con la recolección de datos del escáner (Ajandro Marambio, Universitat Politècnica de Catalunya, ). Existen varios tipos de escáner, pero principalmente se dividen en 2 categorías: Los de contacto y sin contacto.

Los escáneres 3D de contacto examinan el objeto apoyando el elemento de medida (palpado) sobre la superficie del mismo, típicamente una punta de acero duro o zafiro. Una serie de sensores internos permiten determinar la posición espacial del palpador.

Las ventajas de este tipo de sensores son la precisión que logran al momento de escanear un objeto, sin embargo, sus desventajas son mayores, son más delicados, lentos y valiosos. Se usan en su mayoría en control dimensional en procesos de fabricación.

Los escáneres activos emiten alguna clase de señal y analizan su retorno para capturar la geometría de un objeto o una escena. Se utilizan radiaciones electromagnéticas (desde ondas de radio hasta rayos X) o ultrasonidos. Pueden ser señales de ultrasonido entre otras, que de una u otra manera determinan el contorno del objeto (Dr. Roberto Scopigno , 2010).

## 4.2 Coordenadas cilíndricas

El sistema de coordenadas polares, usa la magnitud y el Angulo de las coordenadas polares, en vez de las coordenadas rectangulares  $x$  y  $y$  en el plano. La coordenada  $z$  es la misma que en plano cartesiano, solo tienen una restricción general que es que la magnitud sea mayor o igual 0, y que el ángulo se encuentre entre 0 y  $2\pi$ , la ilustración 1 hace una representación en el espacio (Purcell, 2007).

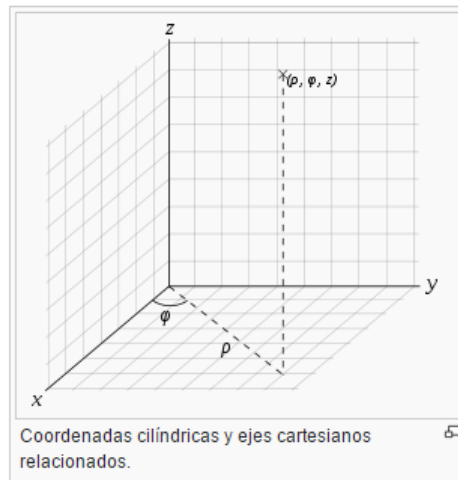


Ilustración 1. Coordenadas cilíndricas y cartesianas relacionadas.

Para convertir de coordenadas polares a rectangulares, se usan las siguientes ecuaciones, donde la magnitud y el ángulo se transforman a coordenadas cartesianas.

$$x = \rho \cos \varphi, \quad y = \rho \sin \varphi, \quad z = z$$

## 4.3 Laser

En general la luz no solo los laser tienen una propiedad que se hace más evidente al usar el láser, cuando la luz pasa por una rendija, bordes ranuras o agujeros ocurre un fenómeno llamado difracción. Cuando el objeto que se pone al frente del láser tiene forma rectangular el equivalente es una línea recta. (Bilmes, 1992)



## 4.4 Cámara WEB

Una cámara WEB, toma una secuencia de imágenes que generalmente tiene su principal uso en aplicativos virtuales, como video conferencias, toma de imágenes o un chat en directo (informatica-hoy, 2011). Según su relación de la cantidad de tomas que puede capturar en un segundo, con el tamaño de las imágenes y la definición de estas mismas puede variar en el mercado.

## 4.5 Motor paso a paso

Los motores paso a paso permiten hacer movimientos muy precisos ya que se pueden mover dependiendo de cada uno de los pulsos que se les aplique a las bobinas, estos pasos pueden ir desde 4 lo cual permitiría  $90^\circ$  por cada paso hasta 200 pasos que equivaldrían a  $1.8^\circ$  en cada uno de ellos, para completar los  $360^\circ$ . En la ilustración 2 se puede observar los posibles movimientos de acuerdo a la excitación de las bobinas.

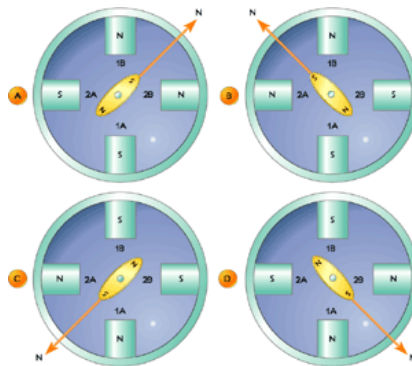


Ilustración 2 Movimientos motor paso a paso

## 4.6 Arduino

Arduino es una plataforma de prototipos libre que trabaja mediante la filosofía de software y hardware fácil de usar, de tal forma que puede ser usado por personas sin conocimientos avanzados de programación y electrónica. Las tarjetas de Arduino poseen entradas para diferentes sensores y salidas para actuadores como motores o leds, los cuales se controlan por un paquete de instrucciones que se programan desde un computador en un software que Arduino desarrolla el cual tiene una comunidad que desarrolla proyectos (Arduino, s.f.).

## 4.7 Octave

Octave es un lenguaje de alto nivel libre, destinado principalmente para cálculos numéricos avanzados. Tiene amplias herramientas para resolver problemas de álgebra lineal numérica comunes, encontrar las raíces de ecuaciones no lineales, la integración de las funciones ordinarias, manipular polinomios, e integrando diferenciales ordinarias y ecuaciones diferenciales algebraicas. Es fácilmente extensible y personalizable a través de las funciones definidas por el usuario escritas en la lengua propia de Octave, o el uso de módulos cargados dinámicamente escritos en C ++, C, Fortran, u otros lenguajes (John W. Eaton, s.f.).

## 4.8 Nube de puntos y Meshlab

La nube de puntos es un conjunto de coordenadas (x,y,z) que forma la superficie del objeto y de donde se extraen características de este según el uso que se desee (Gestión de información georreferenciada y geomática , s.f.). Es la representación más completa ya que tiene toda la información del objeto.

Meshlab es una plataforma libre la cual procesa mallas 3D no estructuradas y proporciona un conjunto de herramientas para la edición, la limpieza, la curación, la inspección, la representación y la conversión de este tipo de mallas (Meshlab, s.f.).

# 5. DESARROLLO DEL PROYECTO

## 5.1 CONSTRUCCIÓN DEL HARDWARE

### 5.1.1 Chasis.

Para la construcción del chasis del escáner se tuvieron en cuenta las limitaciones que iba a tener el proyecto; como el tamaño del objeto, la resolución y distancia de la cámara, el ángulo del láser con la cámara y que permitiera estar totalmente oscuro por tal motivo se pintó de negro su interior. En la parte interna se colocaron 2 cristales filtros para soldadura número 14 para evitar que se reflejara el rayo láser en la parte de atrás del chasis, uno de ellos se colocó en la parte de atrás de la caja y otro en la tapa en la parte superior.

Dado que el láser no era de línea, se usó el cascaron de un esfero por sus propiedades reflectivas y con el que se logró el efecto de una línea de láser con el cual se podía bordear el objeto.

Las medidas que se usaron para el chasis del escáner fueron 51 cm de largo X 24.5 de ancho X 22cm de alto, con una base giratoria de 12cm de diámetro.

En la ilustración 3 se pueden observar algunas imágenes del proceso de construcción del chasis del escáner.

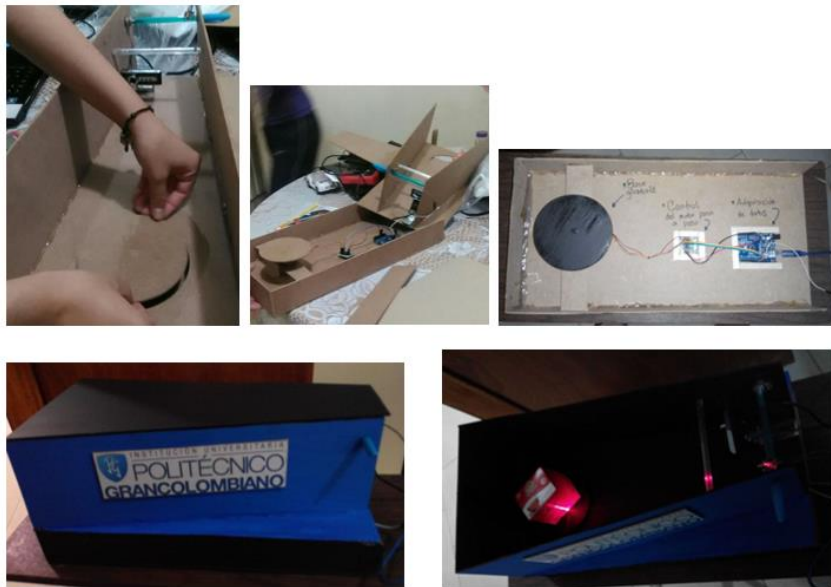


Ilustración 3. Proceso de la construcción del chasis.

### 5.1.2 Movimiento base giratoria

Para la rotación de la base en la cual se encuentran los objetos a escanear se hizo uso de un motor paso a paso 28BYJ-48 con su respectivo adaptador que contienen un puente H integrado que permite el control de las bobinas, separando el voltaje el control por el PWM.

Además de poseer una propia librería para ser programado por Arduino.

### 5.1.3 Adecuación del borde del objeto

Para obtener el borde del objeto se usó el módulo Laser de Arduino KY008 (Ilustración 4) el cual trabaja con un voltaje de 5V y una longitud de onda de 650nm, para reflejar el contorno del objeto.

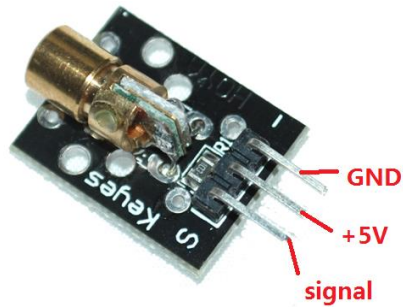


Ilustración 4 Modulo Laser Arduino

IMAGEN TOMADA DE

[http://linksprite.com/wiki/index.php5?title=Advanced\\_Sensors\\_Kit\\_for\\_Arduino](http://linksprite.com/wiki/index.php5?title=Advanced_Sensors_Kit_for_Arduino)

#### 5.1.4 Preparación de la cámara

En esta ocasión se usa el software libre Octave, y su paquete de adquisición de imagen (image-adquisicion), las cámaras en general tienen algunos parámetros que se pueden variar los cuales traen unos valores por defecto, las listas de estos valores se encuentran en el anexo 2.

Una vez se conecta la cámara WEB se usa el comando **imaqhwinfo** para reconocer en qué dirección el computador ubica la cámara y posteriormente con el comando **set** se modifican los parámetros a la máxima resolución y el espacio de color que devuelve la cámara RGB.

Se crea un objeto con el comando VideoInput y con el comando **lsusb** de Linux podemos visualizar en cual puerto reconoció el computador la cámara

## 5.2 CONSTRUCCIÓN DEL SOFTWARE

### 5.2.1 Control del motor y el laser

En este caso se usa un Arduino UNO, donde se usó el convertidor análogo digital para realizar el respectivo control del motor, al igual se usó una salida lógica para el control del láser. Todo está programa por el software propio de Arduino, en lenguaje C en el anexo se adjuntan todos los códigos correspondientes.

### 5.2.2 Adquisición de la imagen

Las imágenes adquiridas tienen una resolución de 400x600 píxeles dadas por la cámara WEB, sin embargo, se debe tomar en cuenta desde la mitad del eje de



La cámara toma imágenes de 640X480 pixeles así que, con un editor de imágenes, en este caso colourpaint con la imagen que se capturo en el interior de la cámara, se toma la posición del pixel hasta donde se debe recortar la imagen para poder tomar la información desde el centro de la imagen. En la ilustración 7 se puede observar el ejemplo acerca de la toma de la guía para establecer los límites.



Ilustración 7. Guía para tomar límites

De esta manera se recortan las respectivas imágenes tomadas por la cámara. En el ejemplo anterior se recortan las imágenes al tamaño 230\*480, para tal fin se usa el comando **imcrop** de Octave. Una vez capturadas las imágenes se inicia el procesamiento de la imagen, a continuación, se pueden mostrar algunas imágenes recortadas (Ilustración 8).

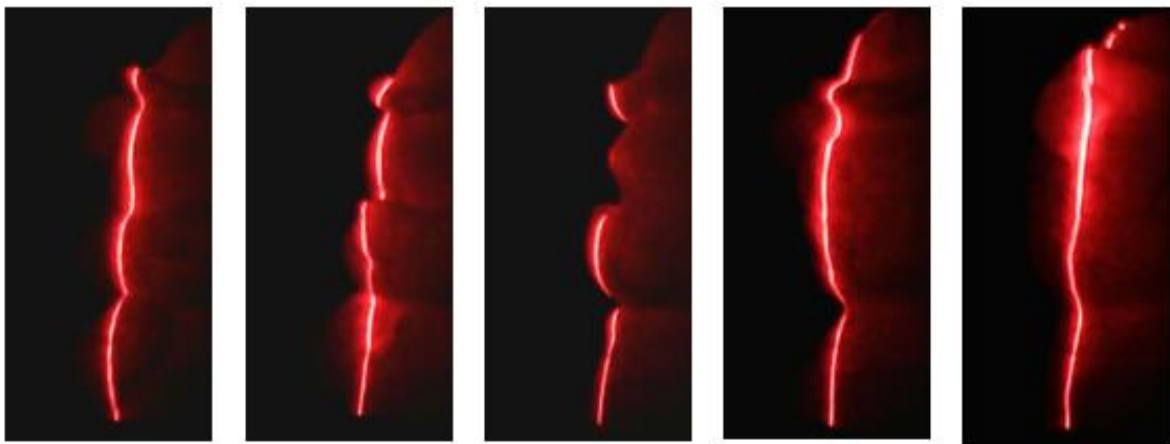


Ilustración 8. Imágenes tomadas por la cámara.

## 5.2.4 Procesamiento de la imagen

En esta etapa entra la imagen sin procesar y sale una matriz de dos coordenadas para hacer al análisis matemático respectivo, inicialmente sobre la imagen original se convierte a una imagen en escala de grises para poder observar su luminosidad y escoger la mejor opción al momento de transformar la imagen.

Aunque solamente escogiendo la forma correcta de promediar se obtienen buenos resultados. El objetivo del filtro no es suavizar los bordes al contrario lo que se necesita es resaltar los bordes y eliminar el ruido producido por la reflexión de la imagen sobre el objeto. La necesidad no es quitar ruido, ni difuminar la imagen, por esta razón se dejan a un lado los filtros lineales y se trabaja con 2 filtros espaciales, el filtro diferencial Laplaciano que permite detectar bordes y un filtro de realce que permite eliminar las zonas borrosas.

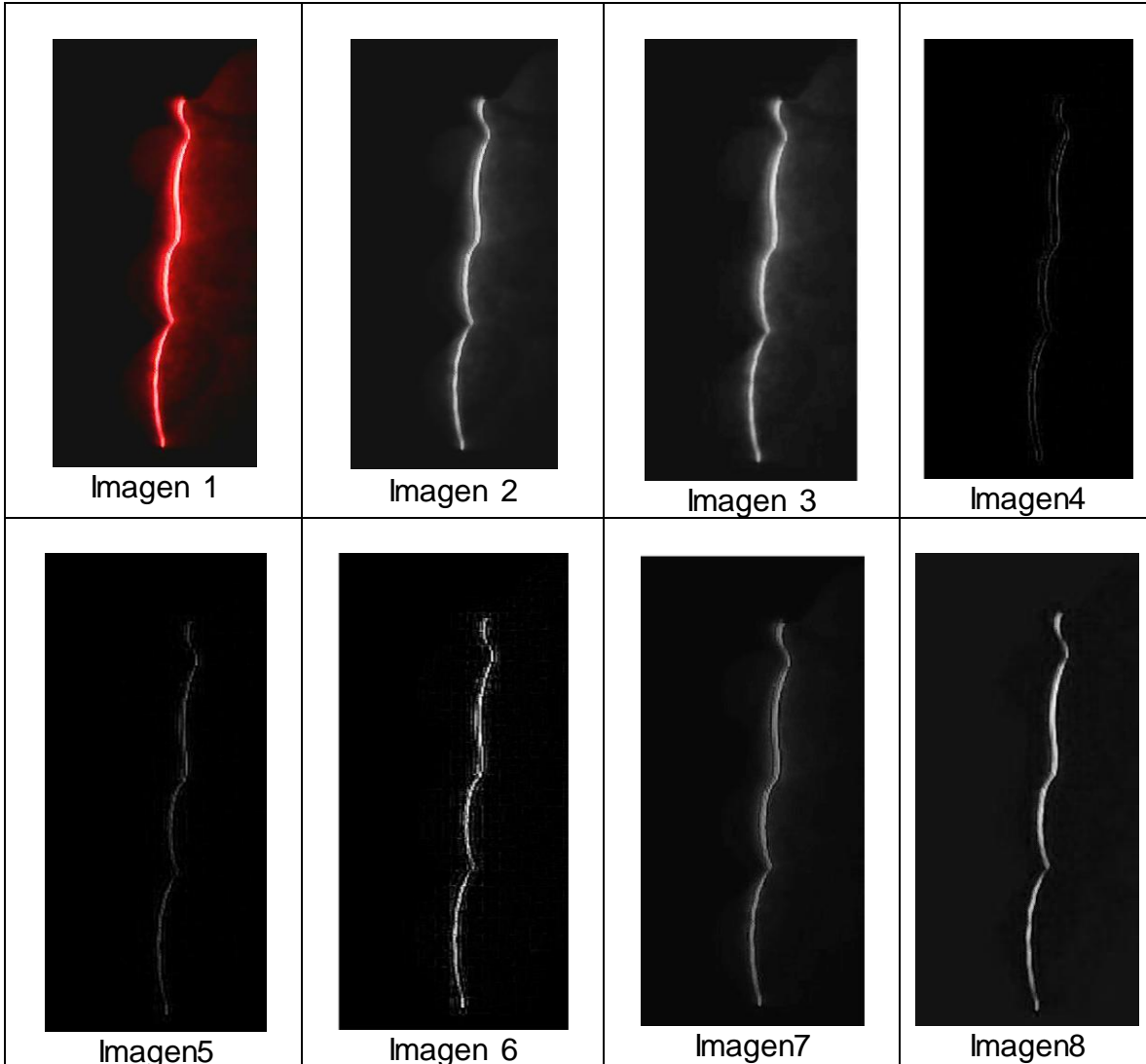


Imagen 1: Imagen Original

Imagen 2: Escalas de grises usando el comando `rgb2gray` donde se promedia:  
 $0.289 * R + 0.5870 * G + 0.1140 * B$

Imagen 3: Escala de grises promediando por igual todos los valores.

Imagen 4: Imagen usando un filtro espacial laplaciano.

Imagen 5: Imagen usando un filtro de realce con la máscara  $\frac{1}{4} * [-2 \ 0 \ -2; 0 \ 8 \ 0; -2 \ 0 \ -2]$ .

Imagen 6: Imagen usando un filtro de realce con la máscara  $[-2 \ 0 \ -2; 0 \ 8 \ 0; -2 \ 0 \ -2]$ .

Imagen 7: Imagen usando un filtro sharpen con la máscara  $\frac{1}{2} * [0 \ -1 \ 0; -1 \ 5 \ -1; 0 \ -1 \ 0]$ ;

Imagen 8: Imagen con canal azul

El objetivo es tener una imagen donde se realcen las curvas del objeto, sin elementos que interfieran. Con los mejores resultados se procede a umbralizar y obtener las imágenes en blanco y negro los resultados son los siguientes:

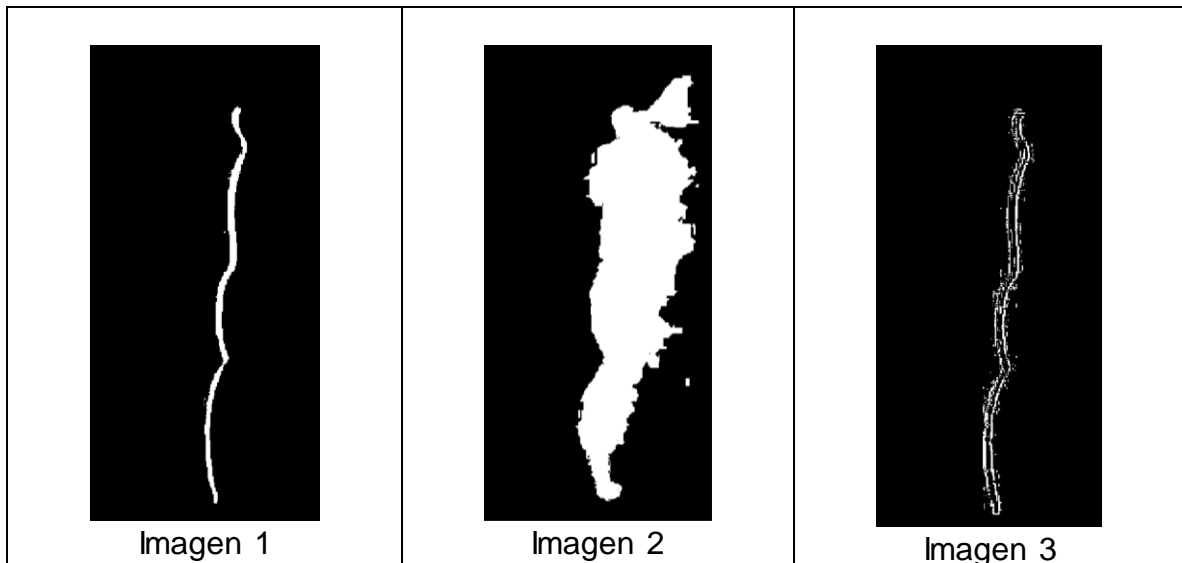


Imagen 1: Imagen en escala de grises promediando los canales por igual.

Imagen 2: Imagen usando el promedio por igual de los canales.

Imagen 3: Imagen usando el filtro Laplaciano.

Una vez se obtiene el mejor resultado se procede a hacerlo más estándar para las demás imágenes y hacer más delgada la línea para que la imagen sea más liviana y se obtengan menos datos al momento de procesar los datos. La imagen umbralizada de la imagen sin un canal es la mejor opción así que con el comando **bwareaopen** se eliminan los puntos pequeños e individuales externos a la imagen y con el comando **bwmorph** se realiza un promedio determinando el “esqueleto” de la figura el resultado es el siguiente:



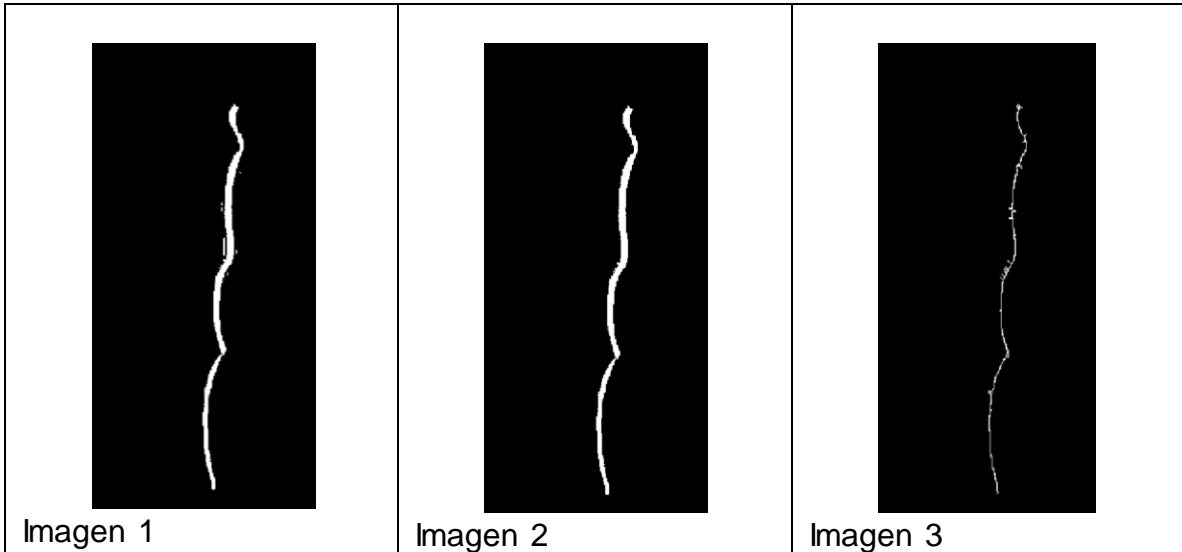
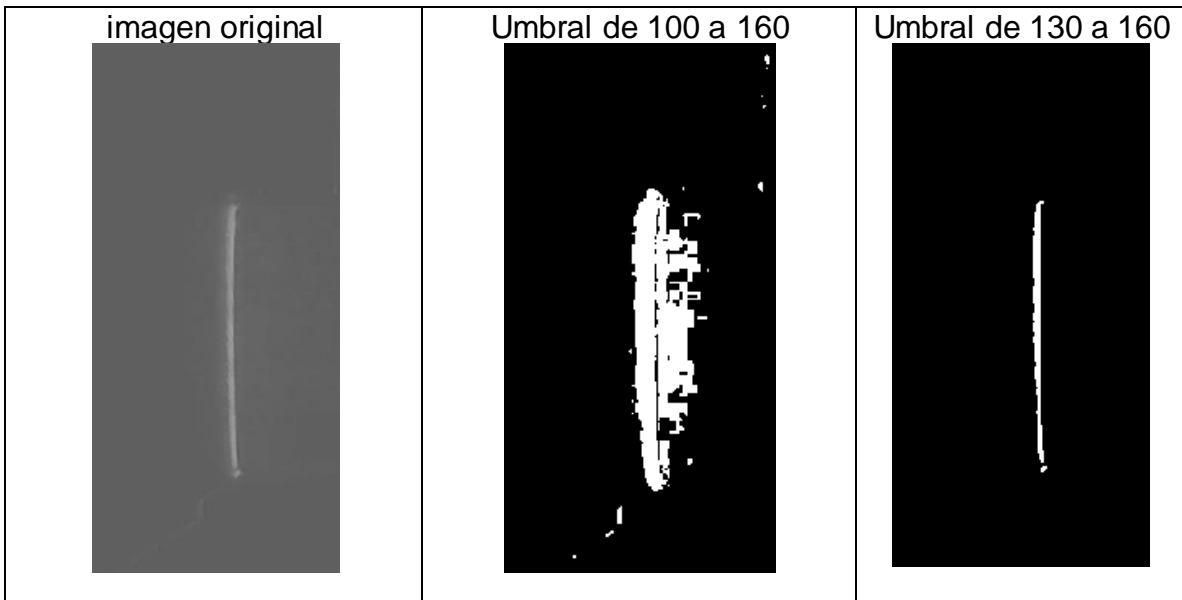


Imagen 1: Imagen original

Imagen 2: Imagen quitando los objetos pequeños.

Imagen 3: Imagen obteniendo solo una línea de la figura.

Posteriormente se escoge un umbral entre estos valores, dependiendo del material se debe seleccionar uno que obtenga la suficiente información. En la siguiente imagen se puede observar la importancia de escoger valores correctos de umbralización, es importante notar que el umbral depende del material o al color porque si es un material que refleja puede causar inconvenientes con el láser, o si es un color que absorbe la luz se va a omitir la luz del láser por la cámara.



### 5.2.5 Conversión de coordenadas.

Una vez se obtiene la respectiva imagen, de cada imagen se obtiene las coordenadas de la siguiente forma (ilustración 9).



Ilustración 9. Coordenadas originales

Sin embargo, para realizar los cálculos correspondientes desde el piso y desde el centro del objeto, se restan a estos valores el valor del ancho y del alto de la imagen obteniendo los siguientes valores:

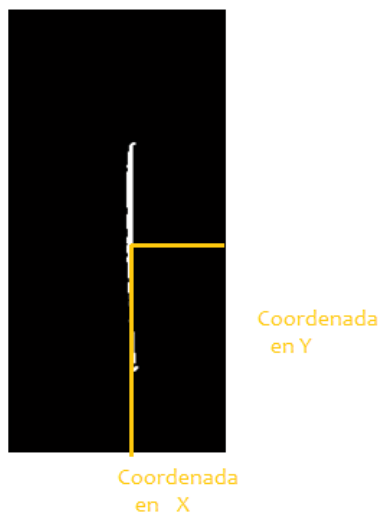


Ilustración 10. Coordenadas modificadas

La coordenada en Z está dada, por el ángulo de giro del objeto, quedando así la respectiva coordenada cilíndrica, así que se aplica la respectiva conversión a

coordenadas rectangulares, se almacena en un vector y se eliminan los valores que no proporcionen información vital con un ángulo de  $190^\circ$  se obtiene en la ilustración 11.

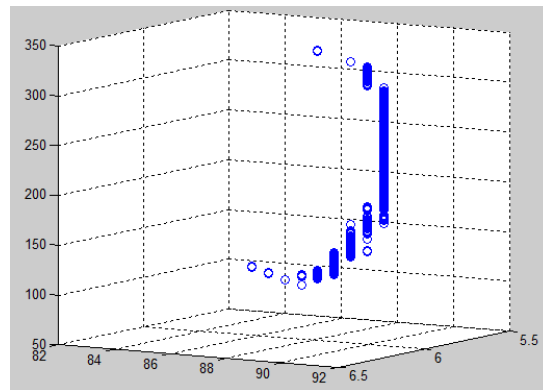


Ilustración 11 Curva reconstruida.

Lo cual implica que el objeto al ser rodeado y estar llegando a  $360^\circ$  se reconstruye la figura total.

#### 5.2.6 Construcción del modelo en tres dimensiones.

Para construir el respectivo modelo en tres dimensiones, el código anterior se encuentra dentro de un ciclo según la cantidad de imágenes que determinan el movimiento completo del objeto, y se almacenan en una variable por ciclo realizado.

Este vector se almacena en un archivo de una hoja de cálculo, que a su vez se modifica para que quede en un archivo.txt y se ejecuta en el programa Mesh LaB para poder reconstruir la imagen.

## 6. DELIMITACIONES

El escáner cuenta con unas medidas de 51 cm de largo X 24.5 de ancho X 22cm de alto teniendo en cuenta que 7cm de alto son destinados a la parte electrónica del proyecto por lo cual tan solo se cuenta con 12cm para colocar el objeto, se recomienda que el objeto sea menor a los 12cm, por lo cual tiene limitaciones en cuanto al tamaño del objeto a escanear el cual tampoco puede ser mayor que la base giratoria que cuenta con 12cm de diámetro, también se debe tener en cuenta el material y el color del objeto el cual debe permitir que el rayo láser se refleje para que pueda ser capturado de manera óptima por la cámara.

## 7. DESARROLLO DE PRUEBAS

### 7.1 Casos de uso

Se desarrollaron pruebas con 4 objetos a continuación se presentan los objetos a escanear y los resultados obtenidos para cada uno de ellos. La superficie de la nube de puntos es generada en el programa Meshlab.

#### CASO 1 - Objeto 1: Tortuga

Objeto original



Nube de puntos

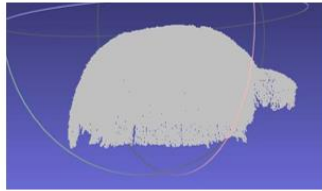


Ilustración 12. Tortuga reconstruida.

#### CASO 2 - Objeto 2: Perro

Objeto original



Nube de puntos



Ilustración 13. Perro reconstruido

#### CASO 3 - Objeto 3: Cabeza oso

Objeto original



Nube de puntos



Ilustración 14 Oso reconstruido.

## CASO 4 - Objeto 4: Pelota

Objeto original



Nube de puntos

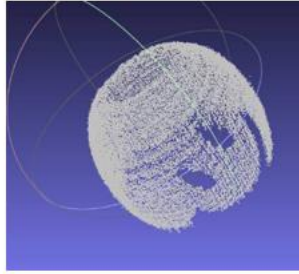


Ilustración 15 Pelota reconstruida.

## 8. CONCLUSIONES

- Para el presente trabajo se hizo uso del software libre OCTAVE el cual es útil para fines académicos, sin embargo, su procesamiento es más lento en comparación con software que no son libres como MATLAB
- La construcción de un escáner con un bajo presupuesto es útil para implementarlo con fines académicos (colegios, universidades.)
- Para proyectos de este tipo sería útil hacer uso de una cámara con una mejor resolución ya que facilita el procesamiento de las imágenes.
- La generación de la nube de puntos es más completa cuando se hace uso de objetos regulares que no tengan demasiadas curvas internas.
- La orientación y distancia del láser es una parte importante del escáner ya que por medio de esto se puede hacer un recorrido apropiado del objeto.

## 9. BIBLIOGRAFÍA

- Ajandro Marambio, D. P. (Universitat Politècnica de Catalunya, ). ESCANER LASER: MODELO 3D Y ORTO IMÁGENES ARQUITECTÓNICAS DE LA IGLESIA. *ACE, arquitectura ciudad y entorno*, 178-187.
- Arduino. (s.f.). *Arduino*. Obtenido de Arduino: <https://www.arduino.cc/en/Guide/Introduction>
- Beales, B. R. (Aug 11, 2006). *PC Systems, Installation and Maintenance*. Routledge: R.
- Bilmes, G. M. (1992). Laser. En G. M. Bilmes, *Laser* (págs. 21-25). Buenos Aires: Ediciones Colihue SRL.
- Dr. Roberto Scopigno . (2010). *Digitalización 3D*. España: Master en desarrollo de Software.
- Gestion de información georreferenciada y geomática . (s.f.). *GIMGEOMATICS*. Obtenido de GIMGEOMATICS: <http://www.gim-geomatics.com/nube-puntos>
- informatica-hoy*. (07 de Julio de 2011). Obtenido de informatica-hoy: <http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-una-Camara-web.php>
- John W. Eaton. (s.f.). *Octave*. Obtenido de GNU Octave: <https://www.gnu.org/software/octave/>
- Meshlab. (s.f.). *Meshlab*. Obtenido de Meshlab: <http://meshlab.sourceforge.net/>
- Plana, C. d. (1997). *VIII encuentros de geometría computacional*. España: Universitat Jaume.
- Purcell, E. J. (2007). *Calculo*. Pearson Educación.
- Sebastian, J. M. (2005). ESCANEADO EN 3D Y PROTOTIPDO DE PIEZAS ARQUITECTONICAS. *Iberia*, 135-158.

## 10. CIBERGRAFÍA

### Cámara

- [https://wiki.archlinux.org/index.php/Webcam\\_setup](https://wiki.archlinux.org/index.php/Webcam_setup)

### Filtrado de imágenes

- <http://informatica.uv.es/doctorado/AIRF/ParteAI/tema3.pdf>
- <http://www.mathworks.com/help/images/ref/im2bw.html>
- <http://www.mathworks.com/help/matlab/ref/rgb2gray.html>
- [http://www2.fisica.unlp.edu.ar/materias/procesamiento de imagenes/Clase2 Imagenes\\_2011.pdf](http://www2.fisica.unlp.edu.ar/materias/procesamiento_de_imagenes/Clase2_Imagenes_2011.pdf)

### Motor paso a paso

- <http://www.prometec.net/motor-28byj-48/>

## 11. ANEXOS

### ANEXO 1. Control de los pasos del motor y el láser en ARDUINO.

```
/* -----
   http://www.prometec.net/motor-28byj-48
   Prog_4_2

   Moviendo un motor paso a paso 28BYJ-48 con Arduino
   -----
*/

/* -----
   CAMBIA EL NOMBRE DE LOS PINES POR UN NOMBRE IN1, IN2, IN3 .....
*/
#define IN0  13  //
#define IN1  12  //
#define IN2  11
#define IN3  10
#define IN4   9

int steps_left=8095;
boolean Direction = true;
int Steps = 0;

/* -----
   Crear la matriz segun el tipo de movimiento que se desee, asi se mueven las
   bobinas.
*/
int Paso [ 8 ][ 4 ] =

    {   {1, 0, 0, 0},
        {1, 1, 0, 0},
        {0, 1, 0, 0},
        {0, 1, 1, 0},
        {0, 0, 1, 0},
        {0, 0, 1, 1},
        {0, 0, 0, 1},
        {1, 0, 0, 1}
    };

/* -----
   inicia los terminos como entrada y salida, ademas se crea la respectiva
   comunicacion.
*/
void setup()
    {
    pinMode(IN0, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
    }

/* -----
   La funcion stepper alcanza un paso del motor, y permite tener control de
   cada cuanto se quiere el movimiento
   -----
*/
```



```

*/
void loop()
{
    digitalWrite (IN0, HIGH); // open the laser head
    while(steps_left>0)
    {
        stepper() ;    // Avanza un paso
        delay (12) ;
        steps_left-- ; // Un paso menos

    }

    steps_left=8095;
}

void stepper()          //Avanza un paso
{
    digitalWrite( IN1, Paso[Steps][ 0] );
    digitalWrite( IN2, Paso[Steps][ 1] );
    digitalWrite( IN3, Paso[Steps][ 2] );
    digitalWrite( IN4, Paso[Steps][ 3] );

    SetDirection();
}

void SetDirection()
{
    if(Direction)
        Steps++;
    else
        Steps--;

    Steps = ( Steps + 7 ) % 7 ;
}

```

## ANEXO 2. Adquisición de la imagen.

```

pkg load image-acquisition;
pkg load image;

imaqhwinfo
%preview(obj) %Util para alistar la imagen
%obj = videoinput ("v4l2", __test__device__);
%obj = videoinput ("v4l2", "/dev/video0");
obj = videoinput ("v4l2", "/dev/video1");

set(obj, "VideoResolution", [640 480]);
set(obj, "VideoFrameInterval", [1 30])
set(obj, "VideoFormat", "RGB24")
start(obj)
img = getsnapshot(obj);
%imshow(img); Probar la camara
imwrite (img, strcat('/home/johnmunoz/Trabajo de
grado/Limite', '.jpg'));
for p=1:360;

```

```

    img=getsnapshot(obj);           % funcion para tomar la
foto
    %pause(0.01);                  % es el tiempo de espera
entre una foto y otra
    imwrite (img, strcat('/home/johnmunoz/Trabajo de
grado/Objeto/', num2str(p), '.jpg'));

end
stop(obj); %detiene el video

    % K=rgb2gray(J);               % convierte la imagen a
escala de grises

```

### ANEXO 3. Adecuación de la imagen.

```

pkg load image;
% El limite es 230
    limite=350
%%
for p=1:360

str = '/home/johnmunoz/Trabajo de grado/Objeto/';
str2 = '.jpg';
n = [str num2str(p) str2] ; % El uno equivale a la primer imagen.

    I = imread (n);
    alto=size(I,1); %ALTO
    ancho=size(I,2);%ANCHO
    rect = [1 10 (limite) alto-11];% por problemas con el laser, se quitan
11 pixeles
    J= imcrop(I,rect);
    %imshow(J)

    imwrite(J,n)

end

```

### ANEXO 4. Procesamiento de la imagen.

```

pkg load image;
% El limite es 230

%%
for p=1:360

str = 'C:\Trabajo de grado\Objeto\';
str2 = '.jpg';
n = [str num2str(p) str2] ; % El uno equivale a la primer imagen.
    G = imread (n);
    I=G(:, :, 2);

    alto=size(I,1); %ALTO
    ancho=size(I,2);%ANCHO

```

```

minVal = 25;      %Rango de umbralizacion 25
maxVal = 255;    %      //      //      //
for j=1:alto
for i=1:ancho

if (I(j,i) >= minVal) && (I(j,i) <= maxVal)
    I(j,i)=255;

else
    I(j,i)=0;
end
end

end

I = bwareaopen(I, 150);%remover areas peque??as
%subplot(1,2,1)
%imshow(G)
%subplot(1,2,2)
%imshow(I)
imwrite(I,n)

end

%BW2 = bwmorph(BW,'skel-pratt',Inf);

```

```
%imshow(BW2)
```

## ANEXO 5. Ciclo.

```
% Umbralizacion y puesta en 3D de una sola muestra
```

```

clc,clear
angulo=0.1;
%teta=20;
e=0;%ciclo con todos los valores
for p=1:360

```

```

str = 'C:\Trabajo de grado\Objeto\';
str2 = '.jpg';
n = [str num2str(p) str2] ; % El uno equivale a la primer imagen.

```

```

I = imread (n);
alto=size(I,1); %ALTO
ancho=size(I,2);%ANCHO
% Extraer las coordenadas

```

```
% Mirar fila por fila, hacer un promedio y guardar en el vector resultado
A

```

```
A = zeros (alto, 3);
```

```
c=0;%suma de posiciones
```

```

m=0;%promedio
for j=1:alto
    A(j,3)=j; % Y
    A(j,1)=angulo;%ANGULO
    for i=1:ancho

        if (I(j,i)== 255)
            c=i+c;
            m=m+1;
            A(j,2)=ancho-(c/m);% X (El seno se da por la separacion de la
camara y el laser)
            % A(j,2)=ancho-(c/m)/sin(teta);% X (El seno se da por la
separacion de la camara y el laser)
            i=ancho; % obligarlo a terminar de buscar y terminar el ciclo para
pasar a la siguiente fila

        end

    end

end
c=0;
m=0;
end

A=round(A); % Matriz con la posicion (Angulo , radio y altura)

%CONVERTIR DE RECTANGULAR A POLAR
M = zeros (alto, 3);

for j=1:alto

    M(j,1)=A(j,2)*cos(deg2rad(A(j,1)));
    M(j,2)=A(j,2)*sin(deg2rad(A(j,1)));
    M(j,3)=A(j,3);
end
%Quitar los ceros
R = zeros (10, 3); %Va a almacenar todos los datos
j=1;
for i=1:alto

    if (M(i,2)~= 0)

        R(j,1)=M(i,1);
        R(j,2)=M(i,2);
        R(j,3)=M(i,3);
        j=j+1;
    end

end

end
%Almacenar todo en una Mega variable para recrear los datos
for d=1:size(R,1)
e=e+1;
Z(e,1)=R(d,1);
Z(e,2)=R(d,2);
Z(e,3)=R(d,3);

```

```
end

    angulo=angulo+1

end

% pkg load io ;
xlswrite('molde.xlsx', round(Z));
```