

DISEÑO E IMPLEMENTACIÓN DE UN ESCÁNER 3D PARA PROTOTIPADO Y
MODELADO GEOMÉTRICO DE OBJETOS

CLAUDIA MARCELA CRESPO JULIO
ERNESTO JAVIER FERNÁNDEZ ZÚÑIGA

TRABAJO DE GRADO COMO REQUISITO PARA OPTAR AL TÍTULO DE
INGENIERO MECATRÓNICO.

UNIVERSIDAD AUTÓNOMA DEL CARIBE
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA MECATRÓNICA
BARRANQUILLA

2014

DISEÑO E IMPLEMENTACIÓN DE UN ESCÁNER 3D PARA PROTOTIPADO Y
MODELADO GEOMÉTRICO DE OBJETOS

CLAUDIA MARCELA CRESPO JULIO
ERNESTO JAVIER FERNÁNDEZ ZÚÑIGA

TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE INGENIERO
MECATRÓNICO

Asesor disciplinar

M.Sc. JULIE STEPHANY BERRÍO PÉREZ

Asesor metodológico

M.Sc. MAURICIO ANDRÉS BARRIOS BARRIOS

UNIVERSIDAD AUTÓNOMA DEL CARIBE
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERIA MECATRÓNICA
BARRANQUILLA

2014

Nota de Aceptación

Firma del Presidente del jurado

Firma del jurado

Firma del jurado

Agradecimientos

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo, en especial a nuestra asesora Stephany Berrio. Por la orientación, el seguimiento y la supervisión continúa de la misma.

Especial reconocimiento merece el interés mostrado por nuestro trabajo y las sugerencias recibidas del profesor Mauricio Barrios.

Quisiera hacer extensiva mi gratitud a mis compañeros del programa de ingeniería Mecatrónica, especialmente a Steffen Cantillo por su amistad y colaboración.

Un agradecimiento muy especial merece la comprensión, paciencia y el ánimo recibidos de mi familia y amigos. En especial a mi mama Nayibe Julio y a mi segundo padre Carlos Dávila por brindarme su ayuda en todo los momentos que lo necesitaba.

A todos ellos, muchas gracias.

Agradezco a Dios en primer lugar por este logro alcanzado, gracias por mis victorias, porque en ellas disfruté el fruto de mi esfuerzo, también por los fracasos que tuve afrontar, porque forje mi carácter y me volví más fuerte. Gracias le doy a mi madre, porque siempre creyó en mí, me acompañó cuando más lo necesitaba y me brindo su apoyo incondicional. Gracias le doy a mi padre, por los consejos prácticos que me dio y por enseñarme tantas cosas que me ayudaron en mi formación profesional.

Les debo mi gratitud a los profesores del programa de Ingeniería Mecatrónica, por la motivación que me dieron y por las impartidas, en especial a nuestra asesora Stephanie Berrio, por brindarnos su amistad, apoyo, orientación y guía para la realización de este proyecto. También al actual director de programa Mauricio Barrios, porque siempre nos motivo a ser mejores y a aspirar a la excelencia.

Agradezco a mis amigos y compañeros de estudio, porque con ellos trabajé durante gran parte de mi etapa de formación profesional, y puedo asegurar que de ellos aprendí cosas valiosas.

Muchas gracias a todos.

Tabla de contenido

INTRODUCCIÓN	1
1. PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN	2
1.1 SÍNTOMAS	2
1.2 POSIBLES CAUSAS	2
1.3 VARIABLES DEL PROBLEMA	2
1.4 DESCRIPCIÓN DEL PROBLEMA	3
1.5 PLANTEAMIENTO DEL PROBLEMA	3
1.7 SOLUCIÓN PLANTEADA	3
2. DELIMITACIONES	5
3. OBJETIVOS	6
3.1 OBJETIVO GENERAL	6
3.2 OBJETIVOS ESPECÍFICOS	6
4. MARCO CONCEPTUAL	7
4.1 ESCÁNER 3D	7
4.2 LÁSER DE LÍNEA	8
4.3 NUBE DE PUNTOS	8
4.4 C++	9
4.5 PROCESAMIENTO DIGITAL DE IMÁGENES	9
4.6 OPENCV	9
4.7 OPENGL	10
4.8 QT	11
4.9 ARDUINO	12
4.10 SUPERFICIE DE BÉZIER	12
5. ESTADO DEL ARTE	14
6. ELEMENTOS DE ILUMINACION Y CAPTURA	16
6.1 LÁSER DE LINEA	16
6.1.1 SEGURIDAD VISUAL	17
6.1.2 DISTRIBUCIÓN DE INTENSIDAD	18
6.2 CÁMARA	19
6.3 LUZ BLANCA	20
6.4 UBICACIÓN CÁMARA Y LÁSER	22

7. SISTEMA ELECTRONICO	23
7.1 MODULO DE ROTACIÓN	24
7.3 CIRCUITOS ELECTRONICOS.....	25
8. PROCESAMIENTO DE IMAGENES.....	28
9. CARACTERIZACIÓN DEL SISTEMA	30
11. SOFTWARE.....	41
11.1 DIAGRAMAS DE FLUJO.....	41
11.1.1 PROGRAMACION ARDUINO	41
11.1.2 PROGRAMACION PROCESAMIENTO DE IMÁGENES.....	43
11.1.3 PROGRAMACION GRAFICACIÓN	44
11.2 INTERFAZ	45
12. CHASIS.....	49
13. RESULTADOS.....	50
14. CONCLUSIONES	54
15. RECOMENDACIONES.....	55
16. REFERENCIAS	56

Tabla de ilustraciones

Figura 1. Ejemplo aplicación superficie de Bézier.....	13
Figura 2. Imagen a escala del láser utilizado	16
Figura 3. Avisos de seguridad.....	17
Figura 4. Haz de láser Gaussiano.....	18
Figura 5. Leds de luz blanca de alta potencia utilizados	20
Figura 6. Ubicación Leds de haz blanca	21
Figura 7. Posicionamiento cámara web láser	22
Figura 8. Diagrama de flujos de circuitos electrónicos.....	23
Figura 9. Disco rotatorio.....	24
Figura 10. Circuito fuente 5V y 12V	25
Figura 11. Diagrama esquemático conexión drive L293D y motor paso a paso ...	26
Figura 12 Circuito Alimentación laser y Leds de luz blanca.....	27
Figura 13. Perfil de un objeto en RGB y en escala de grises.....	28
Figura 14. Imagen binaria y esqueletizada,	29
Figura 15. Principio de triangulación.....	30
Figura 16. Patrón utilizado para calibración.....	31
Figura 17. Patrón iluminado por el láser a 165mm de distancia	31
Figura 18. Patrón iluminado por el láser a 100mm de distancia	32
Figura 19. Patrón iluminado por el láser a 55mm de distancia	32
Figura 20. Esqueleto de los perfiles superpuestos	33
Figura 21. Gráfica de relación de pixeles con milímetros	34
Figura 22. Gráfica de relación de pixeles con milímetros	35
Figura 23. Graficación nube de puntos OpenGL.....	37
Figura 24. Pseudocódigo algoritmo de graficación	38
Figura 25. Puntos de control superficie Bézier.....	38
Figura 26. Algoritmo Graficación método de Bézier.....	40
Figura 27. Diagrama de flujo programación arduino	41
Figura 28. Diagrama de flujo programación procesamiento de imágenes	43
Figura 29. Diagrama de flujos programación graficación.....	44
Figura 30. Interfaz sistema de escaneo tridimensional.....	45
Figura 31. Interfaz para el escaneo y procesamiento de imagen	46
Figura 32. Interfaz Graficación.....	47
Figura 33. Vista interior del sistema de escaneo tridimensional	49
Figura 34. Vista exterior del chasis.....	49
Figura 35. Objeto Tridimensional representado en una nube de puntos	50
Figura 36. Objeto Tridimensional representado en forma de malla	51
Figura 37. Objeto escaneado sin color	51
Figura 38. Objeto escaneado a color.....	52
Figura 39. Objeto Escaneado a color.....	52
Figura 40. Objeto escaneado fuera de las delimitaciones	53

INTRODUCCIÓN

Desde los inicios de la era industrial, e incluso mucho antes, se han utilizado los prototipos como la etapa final de un producto para probar su funcionamiento antes de ser producidos en masa. Pero antes de la creación del prototipo físico, se necesita diseñarlo, teniendo en cuenta todas las características con las que va a contar el producto. En la actualidad, el diseño se hace de forma computacional y se hace mucho más sencillo realizarlo. Pero cuando se desea modelar físicamente un objeto o producto ya existente, se hace complicado debido a que se debe realizar algo conocido como ingeniería inversa, es decir, obtener la información de un dispositivo o producto existente para conocer la forma en la que fue hecho.

Esta tarea ahora es más sencilla gracias a la implementación de escáneres 3D, los cuales han reducido la dificultad y el tiempo de trabajo con el que antes se realizaba la ingeniería inversa para obtener información de la geometría de un objeto. Para realizar un modelado es necesario utilizar dispositivos electrónicos, tales como computadores, un láser de línea, sensores, motores, entre otros. La unión de estos dispositivos, permite la creación de un sistema de escaneo tridimensional, del cual se puede obtener un modelo CAD para su fácil manipulación por computador.

1. PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN

La perspectiva del ser humano ha evolucionado con respecto al desarrollo de éste frente a las adversidades que lo acomplexan. Entre las herramientas de las que se ha valido para sobrevivir existe el registro de imágenes para futura interpretación por sí mismo. Desafortunadamente, el formato bidimensional ha dominado tal materia, con tal de solo recurrir a ilusiones ópticas para lograr una percepción más cercana a la psique humana: tridimensional.

1.1 SÍNTOMAS

- Falta de creación rápida de prototipos modelados por computador
- Falta de software para la creación de éste mismo

1.2 POSIBLES CAUSAS

- Alto costo del software existente.
- Poca eficiencia y dificultad de adquisición de métodos alternativos en Latinoamérica.

1.3 VARIABLES DEL PROBLEMA

- Dispositivo emisor de rayos láser.
- Software: equipamiento lógico o soporte lógico de un sistema informático
- Detector: dispositivo pasivo electrónico usado para la identificación de señales electrónicas.

1.4 DESCRIPCIÓN DEL PROBLEMA

Actualmente, en Latinoamérica, no se cuentan con gran variedad de medios o métodos de creación rápida de prototipos modelados por computador. La escasez de estos métodos con respecto al desarrollo general y particular de la tecnología de detección por láser conlleva a una necesidad.

1.5 PLANTEAMIENTO DEL PROBLEMA

¿Cómo se desarrollará la creación rápida de prototipos y modelados por computador al público en Colombia a través de un sistema de escaneo tridimensional?

1.6 SISTEMATIZACIÓN DEL PROBLEMA:

- ¿Cuál es la incidencia del láser para modelamiento de objetos en 3D?
- ¿Qué software es necesario para responder este problema?
- ¿Cuáles son los detectores necesarios para esta implementación?

1.7 SOLUCIÓN PLANTEADA

Nuestra idea de investigación es un sistema de escáner tridimensional con interfaz de usuario por computador, el cual podrá escanear cualquier tipo de objeto cuyo tamaño este dentro de lo estipulado en el diseño, donde su geometría superficial será tomada por un láser y esta información será transferida al computador mediante una cámara, obteniendo un modelado tridimensional del mismo. Esta idea de investigación surgió por la necesidad de realizar modelados rápidos de objetos tridimensionales a un bajo costo para el mercado latinoamericano. Este sistema de escaneo tridimensional de objetos funcionará emitiendo un rayo láser lineal que incide sobre el objeto, esto es percibido por una cámara, y a continuación ingresará los datos obtenidos a un computador para su posterior procesamiento. También es posible tener en cuenta en la toma de datos los colores presentes en el objeto. Este dispositivo presenta las siguientes características: Rápido muestreo, un sistema de medición (contador) que se reinicia al alcanzar el objetivo, alta precisión (submilimétrica), trabajos de alta precisión en momentos o elementos constructivos (para el análisis de las

deformaciones), generación de una alta densidad de puntos, frecuencia oscilante entre los 10.000 - 100.000 puntos. Para la mayoría de las situaciones, un solo escaneo no será suficiente para producir un modelo completo del objeto, indicando que pueden requerirse múltiples tomas, incluso centenares, desde muchas direcciones diferentes para obtener información de todos los lados del objeto. Estos escaneos deben que ser integrados en un sistema común de referencia mediante un proceso llamado alineación, y que transforma las coordenadas locales de cada toma en coordenadas generales del modelo. El proceso completo que va de las tomas individuales a un modelo completo unificado define el flujo de captura de modelo 3D. La interfaz con el usuario será un programa compilado con el lenguaje de programación C++, utilizando en mayor medida las librerías OpenCV y OpenGL, para el procesamiento gráfico de los datos.

El objetivo del estudio es desarrollar un nuevo dispositivo que sea capaz de hacer un modelo tridimensional de objetos mediante láser, cuyo programa será un software libre que les permitirá a los usuarios mejorarlo. Actualmente, el avance de la tecnología ha demostrado que los prototipos y su modelamiento es la base para la creación y diseño de cualquier objeto. Además, la tecnología láser ayuda a ubicar objetos en el espacio y con ayuda de un computador se podrá modificar el prototipo si es necesario. Para iniciar el desarrollo de este proyecto se deben estimar los costos de los circuitos electrónicos, el láser, la estructura y las cosas necesarias para la investigación. El conocimiento necesario para este proyecto de investigación puede ser encontrado en internet, bibliotecas u otros trabajos de investigación. Y cada desarrollo en este campo científico se puede considerar como un avance, dado que planeamos desarrollar un software libre esto les dará fácil acceso a diversos usuarios los cuales podrán implementar y mejorar el software.

2. DELIMITACIONES

- Se requiere un alto conocimiento en campos como álgebra lineal, programación, diseño industrial, electrónica, control automático y procesamiento digital de imágenes.
- Dado a que nuestro software será desarrollado en español sus limitaciones será Iberoamérica.
- Puede ser utilizado en aplicaciones industriales (metrología, automóvil), arqueología, arquitectura, ingeniería y entretenimiento (en la producción de películas y videojuegos).
- Está dirigido a todo público que tenga cierto grado de interés en el modelamiento de objetos en 3D.
- Los objetos a escanear no pueden tener un volumen mayor a: alto=100mm, ancho=200mm, profundidad=200mm
- El software será soportado por sistemas operativos Windows.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Desarrollar un dispositivo que escanee objetos mediante la implementación de un sistema de rayos láser y procesamiento de imágenes para que realice un modelado gráfico 3D vía computador; el cual poseerá un software en el cual se podrán modificar algunas características de este mismo.

3.2 OBJETIVOS ESPECÍFICOS

- Diseñar un dispositivo de recepción y emisión de rayos láser.
- Implementar un dispositivo de recepción y emisión de rayos láser.
- Desarrollar un software que procese las señales emitidas por el dispositivo para realizar un modelado 3D.

4. MARCO CONCEPTUAL

4.1 ESCÁNER 3D

Es un dispositivo que analiza objetos o entornos y obtiene los datos de sus formas y su apariencia. Los datos obtenidos son usados para reconstrucción digital del objeto, crear un modelado tridimensional que puede ser utilizado diversas aplicaciones: industriales, entretenimiento, videojuegos, ingeniería inversa, control de calidad, arqueología, entre otros.

Se puede utilizar muchas clases de dispositivos y tecnologías para construir un escáner 3D, los cuales son seleccionados dependiendo de las características y ventajas que se necesitan, además de disponibilidad de los recursos económicos para su diseño. Estos dispositivos pueden presentar algunas limitaciones, tales como: para la captación de señales ópticas de objetos brillantes, reflejantes o transparentes, aunque estos pueden ser cubiertos por una capa delgada de polvo blanco para solucionar el problema.

El objetivo de un escáner 3D es crear una nube de puntos de muestras geométricas sobre la superficie del objeto. Mediante estos puntos puede extrapolarse la forma del objeto mediante un proceso de reconstrucción digital, además de obtener los colores del objeto si se tuvieron en cuenta en la implementación.

La construcción básica de un escáner 3D está conformada principalmente por un sensor, un emisor de señales y un sistema de procesamiento de datos. Los dispositivos láser son los más utilizados para emitir la señal que se va capturar y procesar, siendo los láseres de línea los más populares. Se utilizan cámaras como sensores gracias a que son muy eficientes para la captura de ondas electromagnéticas, ya que transforma la luz que captura en datos. Por último, el sistema de procesamiento de datos es un software diseñado para transformar y procesar las señales provenientes de la cámara, las cuales vienen en forma de puntos de nubes de puntos.

4.2 LÁSER DE LÍNEA

El láser es una forma de radiación electromagnética que utiliza un efecto de la mecánica cuántica, la emisión inducida o estimulada, para generar un haz de luz coherente de un medio adecuado y con el tamaño, la forma y la pureza controlados. El primer dispositivo láser con resultados satisfactorios fue diseñado y patentado por Theodore H. Maiman en 1960, el cual consistía en un rubí bombeado por una lámpara de flash.

Generalmente los rayos láser son emitidos de forma unidimensional. La utilización de un dispositivo láser de este tipo para el diseño de un escáner 3D es muy ineficiente, haciendo lento el proceso de captura de datos del objeto a escanear. Por esa razón se han diseñado láseres de línea, los cuales despliegan los rayos láser en un determinado ángulo formando un plano luminoso, cuya área dependerá del ángulo de su abertura y su distancia con respecto al objeto, pasando de cubrir un punto en el espacio a una línea, con lo cual se reduce la dificultad y el tiempo de toma de datos.¹

4.3 NUBE DE PUNTOS

Una nube de puntos es un conjunto de puntos ubicados en un sistema de coordenadas tridimensional. Tales vértices se encuentran identificados en un sistema de coordenadas cartesianas X, Y, y Z y representan la superficie externa de un objeto. Un escáner láser 3D puede crear nubes de puntos midiendo automáticamente los puntos que se encuentran en la superficie, almacenando en una base de datos la información.

Con la información obtenida por la nube de puntos, se trata de recrear la forma original del objeto empezando por un proceso de triangulación, donde se interpolan las distancias entre los puntos mediante la generación de triángulos, para luego ser renderizados.

¹Fuente: <http://www.rp-photonics.com/linewidth.html>
Fecha de consulta: 10/09/2013

4.4 C++

Es un moderno lenguaje de programación de alto nivel adoptado por millones de programadores a nivel mundial. Es uno de los más populares para diseñar aplicaciones para PC y el más popular para diseñar juegos de computador. Entre las ventajas que tiene C++ con respecto a otros lenguajes de programación encontramos: su velocidad, un programa bien escrito puede llegar a ser extremadamente rápido, además si se necesita mejorar la ejecución del programa, C++ permite usar lenguaje ensamblador, el cual es un lenguaje de bajo nivel que se comunica directamente con el software. Su flexibilidad, ya que C++ es un lenguaje que tiene soporte para diferentes estilos de programación, incluida la programación orientada a objetos, y a diferencia de otros lenguajes modernos, C++ no obliga a sus programadores a utilizar un estilo en particular. Sus buenos soportes, ya que tiene una gran cantidad de recursos que pueden ser aprovechados por los programadores, tales como gráficas en 2D y 3D, motores de sonido, muchas librerías, entre otros.²

4.5 PROCESAMIENTO DIGITAL DE IMÁGENES

Como su nombre lo indica, es utilizado para procesar y modificar imágenes mediante recursos tecnológicos que captan y manipulan grandes cantidades de información en forma matricial, con el objetivo de mejorar la calidad de la imagen. Este proceso se inicia con la adquisición de datos, los cuales serán analizados y modificados para la realización de las siguientes operaciones: eliminación del ruido, suavización de imagen, realce y detección de bordes.

4.6 OPENCV

Es una librería de visión por computador de código abierto. Está escrita en C y C++ y funciona bajo Windows, Linux y Mac OS X. También tiene un desarrollo activo sobre interfaces para Python, Matlab, Ruby, entre otros lenguajes de programación. Fue diseñado para la eficiencia computacional y con un gran enfoque en aplicaciones en tiempo real. Está escrito en C y puede tomar ventaja de procesadores de varios núcleos.³

² Fuente: <http://www.cplusplus.com/>

Fecha de consulta: 15/09/2013

³ Learning OpenCV, Gary Bradski y Adrian Kaehler

La librería de OpenCV contiene más de 500 funciones que giran alrededor de muchas áreas, tales como visión, inspección de productos, imágenes médicas, seguridad, interfaz de usuario, calibración de cámaras, visión estéreo y robótica

La licencia de código abierto de OpenCV ha sido estructurada de tal manera que cualquier desarrollador pueda construir un producto comercial sin restricciones por parte de OpenCV. Además ha sido usada en muchas aplicaciones, como unión de imágenes satelitales, reducción de ruido en imágenes médicas, análisis de objetos, aplicaciones militares, entre otras.

La librería OpenCV es una herramienta poderosa para diseñar programas de capturas de datos mediante dispositivos de visión, tales como las cámaras. Gracias a los comandos y funciones que tiene disponible, se puede diseñar un programa eficiente para el procesamiento de los datos provenientes de la cámara utilizada en el escáner, donde se podrá configurar la velocidad de escaneo, la calidad de la reestructuración de la nube de puntos obtenida y la densidad de datos que pueden ser capturados.

4.7 OPENGL

Es un entorno de desarrollo portátil para diseño de aplicaciones interactivas con gráficos en 2D y 3D, que brinda una gran variedad de aplicaciones usos a las diferentes plataformas de computadora, sean Windows, Mac OS X o Linux. Promueve la innovación y acelera el desarrollo de aplicaciones gracias a la implementación de un amplio juego de funciones de renderizado, mapeo de texturas, efectos especiales, entre otras funciones de visualización.⁴

Entre las ventajas con las que cuenta OpenGL encontramos: su estabilidad, ya que ha estado disponible por más de 7 años desde su creación, siendo utilizado por diferentes tipos de plataformas, y gracias a sus actualizaciones que mejoran la compatibilidad con los requerimientos lo protege de la obsolescencia. Su confiabilidad, ya que pueden diseñarse aplicaciones con buenos resultados, con cualquier hardware que lo soporte, sin importar el sistema operativo. Su facilidad de uso, ya que su diseño ha sido bien estructurado, cuyas rutinas ocupan pocas líneas de código, en comparación con otras librerías, además de capturar la información concerniente al hardware, para que el desarrollador sea libre de diseñar para una configuración de hardware determinada. Su buena documentación, debido a que muchos códigos de muestras son fáciles de obtener.

⁴ Gráficos con computadora por OpenGL , Donald Hearn y M.Pauline Baker

OpenGL es una librería que complementa a OpenCV, ya que OpenCV se encarga de la captura de datos, su procesamiento y reconstrucción, mientras que OpenGL será utilizado para editar el objeto final generado, permitiendo modificar el modelo de acuerdo a las necesidades del usuario, donde se podrá personalizar características tales como colores, volumen, densidad, entre otras cosas. El archivo será guardado como un documento tipo CAD, donde se podrá modificar utilizando otros programas más acordes a la utilización final del modelo.

4.8 QT

Es un entorno y biblioteca multiplataforma con herramientas diseñadas para agilizar la creación de aplicaciones e interfaces de usuario para escritorio, sistemas embebidos y plataformas móviles. Con este útil programa se puede utilizar el código para eficientemente apuntar diferentes sistemas operativos con un solo código base.

Al ser un software libre es decir su código base puede ser modificado por sus usuarios y desarrolladores oficiales de Nokia esto ayuda a crear mejores actualizaciones y permite lanzar las aplicaciones desarrolladas en el mencionado entorno si el usuario lo consiente con código abierto normalmente utilizando como lenguaje nativo de programación C++ aunque este pueda ser utilizado en varios otros lenguajes a través de bindings.

Este se encuentra disponible para las siguientes plataformas:

Qt se encuentra disponible para sistemas tipo Unix con el servidor gráfico X Windows System (Linux, BSDs, Unix), para Apple Mac OS X, para sistemas Microsoft Windows, para Linux empotrado (en inglés Embedded Linux), para sistemas empotrados como PDA, Smartphone, etc. y para dispositivos que utilizan Windows CE32

Adicionalmente también está disponible QSA (Qt Scripts for Applications), que, basándose en ECMAScript/JavaScript, permite introducir y crear scripts en las aplicaciones creadas con Qt.

Hay tres ediciones de Qt disponibles en cada una de estas plataformas, llamadas:

GUI Framework – edición con nivel reducido de GUI, orientado a redes y bases de datos.

4.9 ARDUINO

Es una plataforma de hardware libre que consta de un microcontrolador y un entorno de desarrollo diseñado para facilitar el uso de la electrónica en diseños multidisciplinarios el cual puede ser descargado gratuitamente desde su página oficial, sus placas normalmente constan de salidas y entradas tanto análogas como digitales las cuales son útiles para controlar distintos tipos de actuadores. El microcontrolador encontrado en la placa se programa mediante el lenguaje de programación de arduino y su entorno de desarrollo consta de distintas librerías y ejemplos predefinidos.

El microcontrolador y el número de entradas y salidas varían de acuerdo al modelo de la placa.

4.10 SUPERFICIE DE BÉZIER

Las Curvas de Bézier son un método de definición de una curva en serie de potencias. Este método consiste en definir algunos puntos de control, y de ahí se calculan los puntos de la curva. Las superficies de Bézier tienen las mismas propiedades que las curvas y proporcionan un método conveniente para aplicaciones interactivas de diseño.

Existen muchas variantes útiles para construir superficies, la más sencilla de comprender es el producto cartesiano o tensorial de curvas de Bézier o de NURBS. Para definir una superficie como un producto cartesiano, debemos asumirla como una función de dos parámetros (u,v) donde P es la función y B es la curva de Bézier .

$$P(u, v) = \sum_i B_{in}(u) \sum_j B_{jm}(v) P_{ij} = \sum_i B_{in} \left(\sum_j B_{jm} P_{ij} \right) = \sum_j B_{jm} \left(\sum_i B_{in} P_{ij} \right) \quad (1)$$

Donde i, j y m, n son las dimensiones de la superficie de las matrices y la curva de Bézier. Para un dado valor del parámetro v fijo se obtiene una curva isoparamétrica con u variable, y lo mismo sucede si se fija u y se varía v . Las alternativas equivalentes de la derecha indican que la superficie puede pensarse

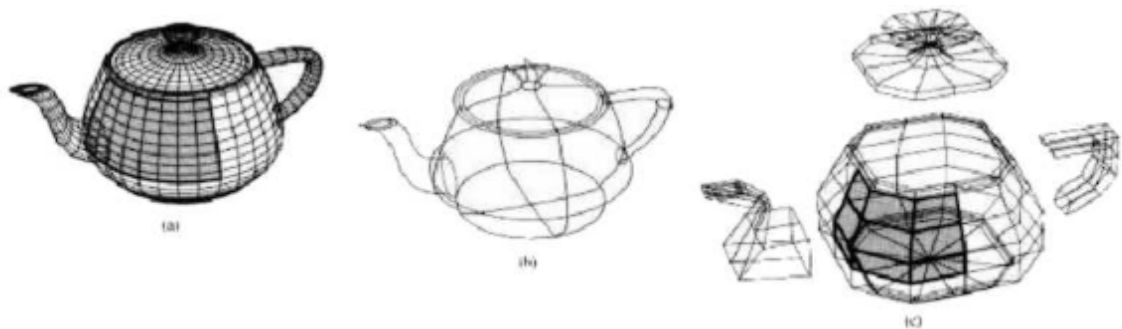
⁵ Fuente: Qt Project qt-project.org
Fecha de consulta: 21/01/2014

como combinación de “curvas de control” (término inexistente), en el mismo sentido en que las curvas son combinaciones de puntos de control.

Dichos puntos de control crean una superficie formando una grilla rectangular de dimensiones $(n+1)$ por $(m+1)$ los cuales se pueden ubicar libremente en el espacio pero están conectados como en una cuadrícula. También es necesario definir un vector de nudos en cada dirección.⁶

Si las splines son rectas se define solo puntos de control de las dos curvas extremas, si ambas son rectas es bilineal. Si estas son círculos coaxiales se unen en una nube de puntos se copian en un array circular alrededor de un eje.

Figura 1. Ejemplo aplicación superficie de Bézier.



Fuente: de http://www.cimec.org.ar/~ncalvo/curvas_doc.pdf

Una superficie de Bézier compuesta construida con dos secciones de dos secciones de Bézier, unidas por la línea indicada, las líneas discontinuas unen los puntos de control, la continuidad de primer orden se establecen haciendo que la relación entre la longitud y la longitud z.

⁶ Tomado de Documento Referente curvas y superficies para modelado geométrico www.cimec.org.ar/~ncalvo/curvas_doc.pdf

5. ESTADO DEL ARTE

Con el pasar de los años las tecnologías de adquisición de imágenes por medio de computador han obtenido un surgimiento mayor que les ha permitido incorporarse aún más en el desarrollo de diversos software y distintos campos de aplicación como lo son la bioingeniería, la resistencia de materiales, producción de videojuegos, entre otros.

Entre estas diversas tecnologías se puede encontrar los sistemas de escaneo 2D, que permiten digitalizar imágenes planas y estos conllevaron a los sistemas de escaneo tridimensionales, que permiten escanear objetos con volumen, es decir analizarlo sus diversas características como lo es su geometría y profundidad más allá del plano principal.

Con respecto a este campo se han visto aportes interdisciplinarios tales como:

Según informa Albert (Jul,2007) con su producto NextEngine Inc, utiliza un set de ojos de láser y junto con la tecnología de cámaras digitales y un sensor de imágenes, pudo escanear datos desde múltiples vistas triangulares; el software ensambla y manipula la información para crear un modelo tridimensional de este para su uso en sistemas CAM. Junto a esto, se puede comprender que la adquisición de estos parámetros arrojarán un resultado apreciable y entendible para futuro procesamiento, cuyos procesos puedan realizarse con facilidad y ajustándose a las necesidades del cliente. Claro está que las digitalizaciones de los objetos en cuestión se limitarán a los parámetros del sistema mismo.

No obstante el alcance de este sistema no se limita a los sistemas CAM, si no que este también se puede aplicar en campos como la biomedicina como nos indica el autor Beverly (Bradley 2008) en su paper "*The design, implementation, and validation of a three-dimensional láser light-sectioning scanner for biomedical purposes*". En el cual se ve reflejado su trabajo sobre el valor del modelamiento de imágenes en el campo de la salud para una variedad de aplicaciones, que requieren la características externas de la partes anatómicas y otros objetos biológicos, y como sería de gran ventaja el mejoramiento de esta tecnología para expandir su aplicación en dicho campo y un costo menos elevado para su uso mediante algoritmos lineares y dándole mayor importancia a su calibración para eliminar el ruido. Entre otros alcances para el escaneo tridimensional dentro de este campo podemos encontrar a los autores Kovacs, L; Zimmermann, A; Brockmann, G; Gühring, M; Baurecht, H; con su paper "*Three-dimensional recording of the human face with a 3D láser scanner*." En el cual se refiere a la grabación de la superficie del cuerpo humano, específicamente la cara. Este estudio se llevó a cabo para mejorar las técnicas de reconocimiento y

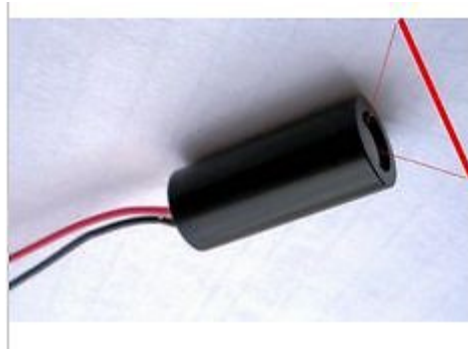
reconstrucción facial, en el cual se escaneo la cara de cinco individuos diferentes para así obtener un mejor ángulo la posición de la cabeza y el impacto de estos parámetros para un propio escaneo de este mismo.

Entre otros campos de usos los escáneres 3D también se pueden utilizar para ingeniería inversa como explica la empresa Metalworking Production (Jan 31,2008) con su producto "QUALITY: *Laser Design and GKS Inspection Services expand 3D láser*" el cual es un sistema de escaneo tridimensional para propósitos de inspección de materiales que puede realizar escaneos en alta velocidad de figuras complejas que incluyen electrónica, aeroespacial y automotriz. Con tales inspecciones, el detallar de los objetos de estudio apremia la eficiencia de detección de problemas difíciles a simple vista, colaborando con una solución más efectiva y prospera para con el problema a resolver. Esto, sin tener en cuenta las limitaciones de alcance, permite una visión un tanto equiparable a la de rayos X, sin los riesgos de radiación que ésta última conlleva. Continuando en los aportes al mismo campo, el siguiente autor, Cheng Hui Zhou muestra una técnica para el mejoramiento de el mismo en "*Three-dimensional laser color scanner hardware subsystem design*" Este escáner adquiere el color y la información tridimensional de los objetos escaneados, con varias técnicas integradas a este mismo como lo son ópticas, mecánicas, control automático, captura de video, procesamiento de imágenes y graficas de computador, todos esto realizado por medio de luz estructurada el cual permite un sistema de calibración más eficiente y también resultado más consistente a su nivel de deformidad

6. ELEMENTOS DE ILUMINACION Y CAPTURA

6.1 LÁSER DE LINEA

Figura 2. Imagen a escala del láser utilizado



Fuente:

[http://i.ebayimg.com/00/s/MTc0WDIzNQ==/z/Ys8AAMXQiFxR72Cc/\\$T2eC16JHJHcFFkSM25\(7BR72Cbtw7!~60_35.JPG](http://i.ebayimg.com/00/s/MTc0WDIzNQ==/z/Ys8AAMXQiFxR72Cc/$T2eC16JHJHcFFkSM25(7BR72Cbtw7!~60_35.JPG)

Se utilizo un láser de línea roja con dimensiones de $\phi 10 \times 23\text{mm}$; $10 * 30\text{MM}$ con una longitud de onda de $650\text{nm} \pm 5\text{nm}$, un ángulo de proyección de 88° , potencia menor de 100mw , un rango de 30 metros y hasta 5000 horas de uso.

Este necesita un voltaje de alimentación de $3\text{V} \sim 5\text{v}$ y con una corriente menor a los 150 mA , la temperatura de operación va desde los $-10\text{ }^\circ\text{C}$ a $70\text{ }^\circ\text{C}$ y temperatura de almacenamiento de $-20\text{ }^\circ\text{C}$ a $\sim +80\text{ }^\circ\text{C}$.

Entre sus aplicaciones se puede encontrar visión para posicionamiento, herramientas de prueba por visión láser, concentricidad, equipos médicos y de belleza, señales entre otros

6.1.1 SEGURIDAD VISUAL

Figura 3. Avisos de seguridad



Fuente:

http://www.laser2000.de/fileadmin/Produktdaten/PPL/Datenblaetter/Compact_Line_635nm_gesamt.pdf.

La luz emitida por estos dispositivos ha sido acordada con el IEC60825, sin embargo mirar directamente o indirectamente debe ser evitado, esta clasifica el láser en tres tipo de peligro de pendiendo de la longitud de onda, la luz emitida y la seguridad de los ojos

CLASE II

Precaución, el haz de luz láser visible menos a de 1.0mw, se considera seguro para los ojos, exposición normal a este tipo de rayo no causara ningún daño permanente a la retina.

CLASE IIIIR

Peligro, la luz láser visible entre 1.0 mW y 5.0 mW. Es considerada segura para los ojos con precaución apuntar esta luz a los ojos puede causar algún daño.

CLASE IIIIB

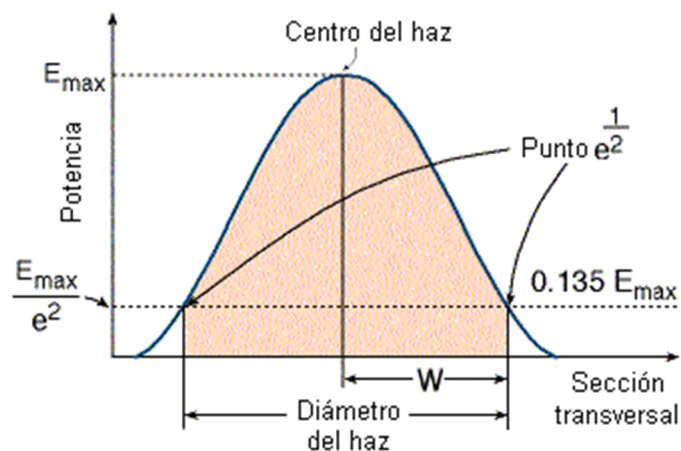
Peligro, infrarrojo y láseres de alto voltaje son peligrosos si la retina es expuesta.

6.1.2 DISTRIBUCIÓN DE INTENSIDAD.

Cuando la radiación láser es emitida en un modo lineal único normalmente se trata del modo básico transversal en este modo la forma de la radiación en un plano perpendicular es un círculo en el cuál la intensidad en el centro es mayor que en los bordes. Esta curva tiene forma de campana, y se denomina Gaussiana- Por ello, el modo básico transversal es llamado Modo Gaussiano.

El generador produce una línea láser con distribución de intensidad Gaussiana, significa que la línea aparece más brillante en el centro y se opaca en los límites. Por este motivo se utiliza la información del centro de la línea, es decir, los píxeles con mayor intensidad luminosa.

Figura 4. Haz de láser Gaussiano



Fuente: <http://www.um.es/leq/láser/Ch-7/F7s1p4.htm>

6.2 CÁMARA.

Se utilizó una cámara Logitech modelo C525; dicha cámara fue escogida por sus características necesarias para el correcto desarrollo del proyecto, como lo son el autoenfoco para evitar distorsión en la imagen mientras el objeto se encuentra en movimiento, por su reducido tamaño el cual es óptimo para su ubicación dentro del chasis, su excelente resolución y la función de corregir automáticamente la iluminación de una toma proporcionando así mejores imágenes para su respectivo procesamiento.

- Videoconferencias HD (1280 x 720 píxeles) con sistema recomendado
- Captura de vídeo HD: Hasta 1280 x 720 píxeles
- Tecnología Logitech Fluid Crystal™
- Enfoque automático
- Fotos: Hasta 8 megapíxeles (mejora por software)
- Micrófono integrado con tecnología Logitech® RightSound™
- Certificación USB 2.0 de alta velocidad (se recomienda)
- Clip universal para monitores LCD, CRT o portátiles⁷

⁷ Fuente: <http://www.logitech.com/es-es/product/hd-webcam-c525>
Fecha de consulta: 10/01/2014

6.3 LUZ BLANCA

La luz blanca se puede definir como la superposición de diferentes colores, cada uno de los cuales presentan una longitud de onda y una frecuencia específicas, y por lo tanto se utilizaron tres Leds de luz blanca de alta potencia (Figura 5), lo cual es indispensable para obtener una toma real del color del objeto.

Figura 5. Leds de luz blanca de alta potencia utilizados

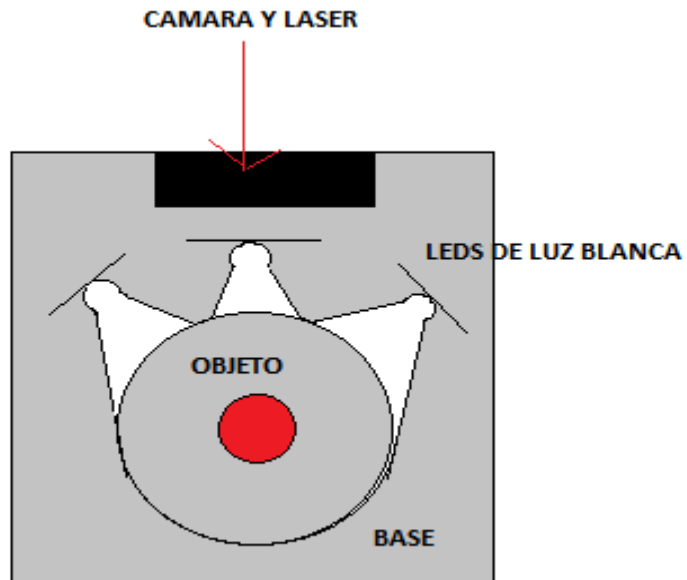


Fuente: <http://img.focalprice.com/94x73/LX/LX562/LX562X-1.jpg>

Dichos Leds trabajan con un voltaje de 3.0 a 5V, un consumo corriente de 700mA cada uno, una potencia de 3Watts y un flujo luminoso de 180-220LM.

6.3.1 UBICACIÓN LEDS DE LUZ BLANCA

Figura 6. Ubicación Leds de haz blanca

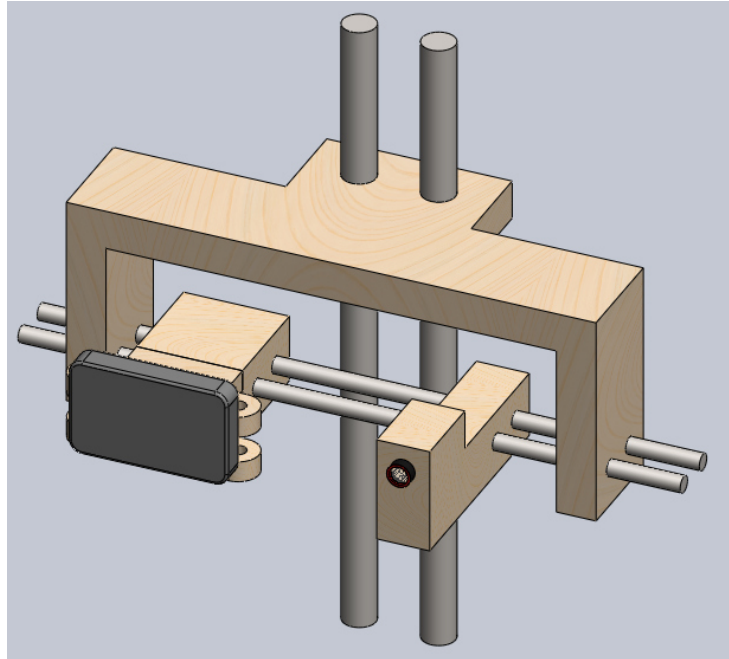


Fuente: Propia.

Tres Leds de luz blanca que generan rayos se distribuyen en forma cónica y han sido ubicados para la iluminación completa del perfil ha escanear iluminando una parte del objeto (Figura 6).

6.4 UBICACIÓN CÁMARA Y LÁSER

Figura 7. Posicionamiento cámara web láser

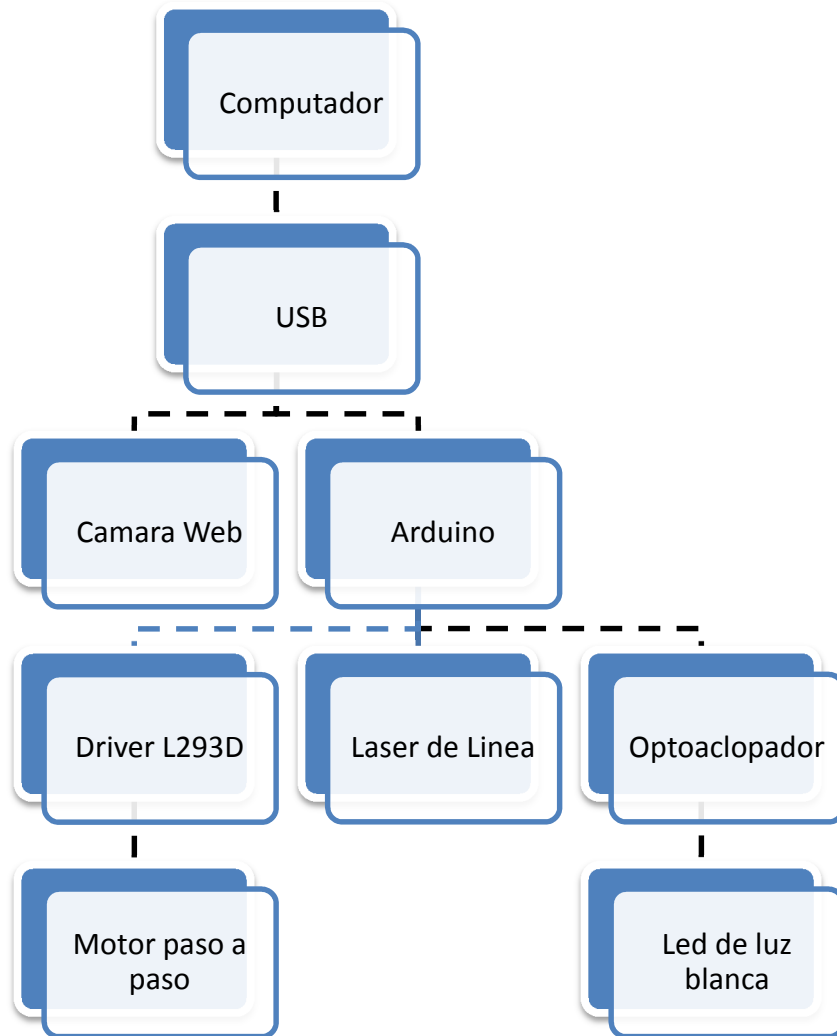


Fuente: Propia.

El mecanismo consta de dos tornillos sin fin, que permiten el movimiento tanto vertical como horizontal del láser y la cámara, permitiendo así una rápida calibración y posicionamiento de los mismos.

7. SISTEMA ELECTRONICO

Figura 8. Diagrama de flujos de circuitos electrónicos



Fuente: Propia.

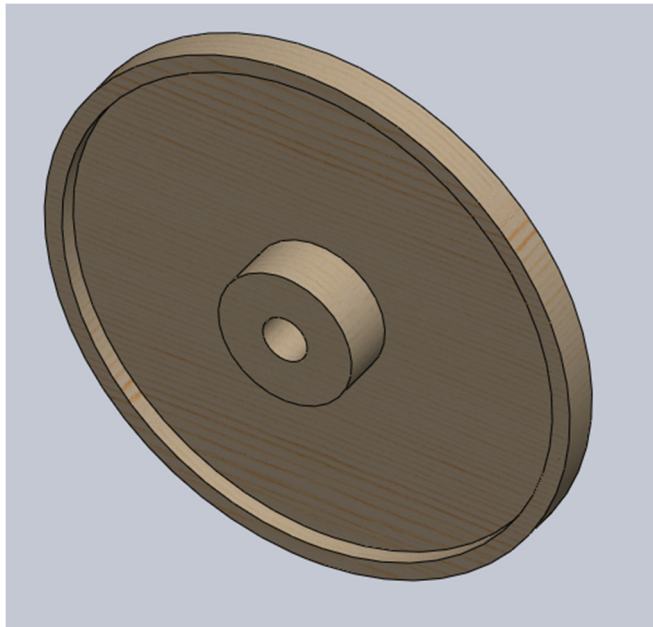
El sistema electrónico (Arduino) recibe las señales emitidas por el computador vía USB en código ASCII y se encarga de decodificarlas, logrando que el actuador (motor paso a paso) se mueva de manera adecuada a través del driver L293D, con el fin de obtener observaciones de la totalidad del objeto a escanear, creando así el modelo tridimensional.

7.1 MODULO DE ROTACIÓN

El módulo de rotación está compuesto por:

- Disco rotatorio
- Motor paso a paso
- Driver L293D

Figura 9. Disco rotatorio

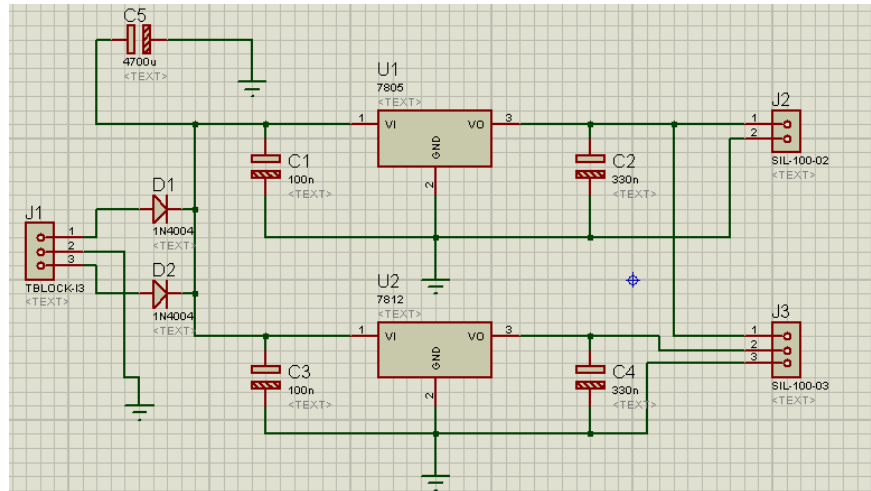


Fuente: Propia.

El modulo de rotación es el encargado de darle un desplazamiento angular al objeto desde 0° hasta 360° , para así obtener todos los perfiles de este. El módulo consta de un disco (Figura 8) que se encuentra acoplado mecánicamente al motor paso a paso, dicho motor es controlado por el driver L293D, el cual recibe información del Arduino que decodifica las órdenes recibidas directamente del software diseñado.

7.3 CIRCUITOS ELECTRONICOS

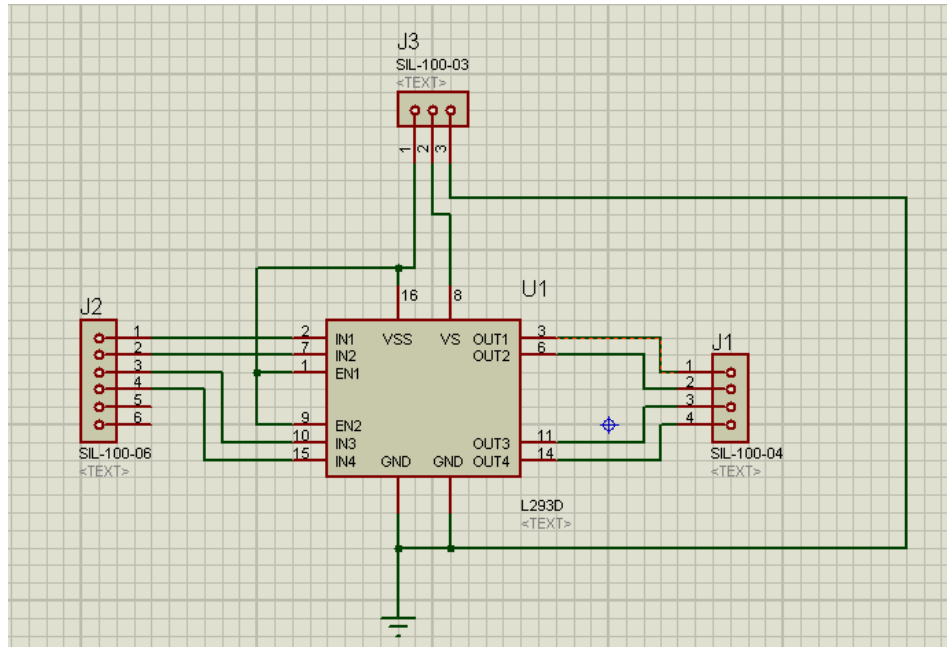
Figura 10. Circuito fuente 5V y 12V



Fuente: Propia.

Se realizó una fuente con salidas de 5V y 12V para suministrar voltaje a los circuitos utilizados para la rotación e iluminación del sistema de escaneo tridimensional. El circuito está conformado por dos reguladores, el LM7805 y el LM7812, los cuales reciben corriente continua de un puente de diodos, que a su vez reciben corriente alterna suministrada por un transformador, de igual forma se agregaron capacitores para eliminar el rizado de la señal en DC.

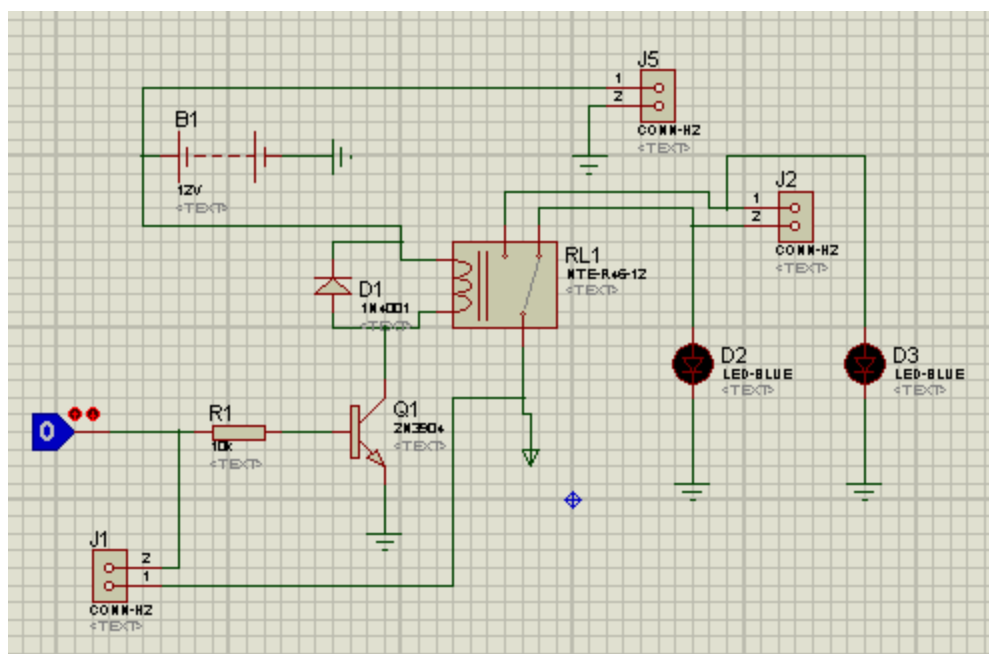
Figura 11. Diagrama esquemático conexión drive L293D y motor paso a paso



Fuente: Propia.

El circuito del L293D es el circuito encargado de controlar el motor es un puente-H dual, incluye cuatro circuitos internos para manejar cargas de potencia media, en especial pequeños motores y cargas inductivas, con la capacidad de controlar corriente hasta 600 mA en cada circuito y una tensión entre 4,5 V a 36 V. Las salidas actúan cuando su señal de habilitación está en alto. Cuando la señal de habilitación del par de circuitos de manejo está en bajo, se desconectan las salidas en un estado de alta impedancia. Por medio de un control apropiado de las señales de entrada y conectando el motor a ciertas salidas de potencia del driver L293D, se obtiene el movimiento deseado.

Figura 12 Circuito Alimentación laser y Leds de luz blanca



Fuente: Propia

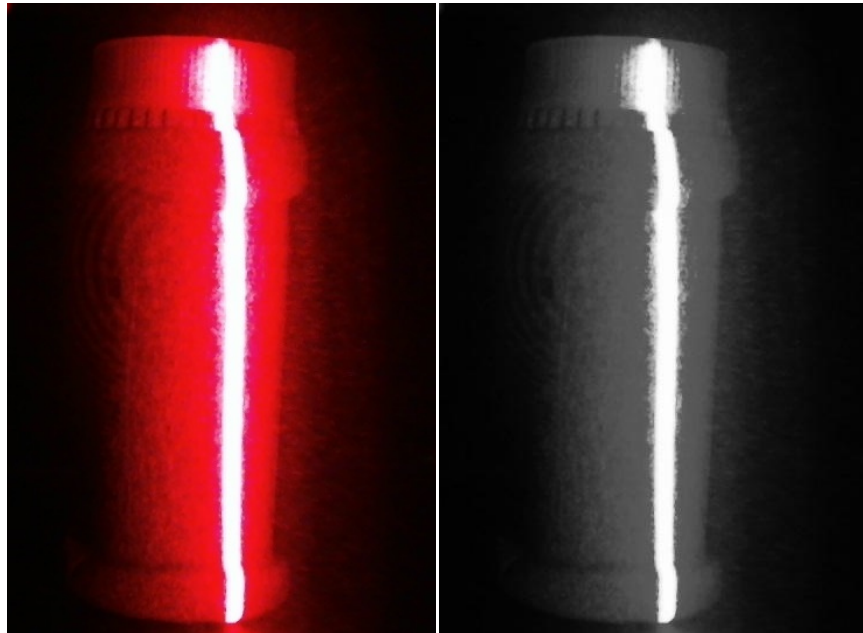
El circuito funciona como un switch para los Leds de luz blanca y laser, Este fue necesario para la propia alimentación de estos, dado a que necesitan más corriente que la proveída por el arduino.

Un pin del Arduino normalmente no alcanzará a manejar correctamente la bobina de un relevador, requerimos una etapa intermedia basada en un transistor BJT o Mosfet que a su vez activará la bobina. Un transistor BJT es un dispositivo controlado por corriente, es decir, requiere una corriente de base para comenzar a conducir. Dicha corriente es muy pequeña a comparación de la corriente que controla en sus otras dos terminales. Para conectarlo al arduino hace falta una resistencia que limite la corriente de base y sin embargo, proporcione la corriente justa para llevar el transistor a un estado de "saturación". Como en esta aplicación en concreto estamos conectando una carga inductiva, también debemos colocar un diodo que protege al transistor de los picos de voltaje generados por la bobina del relé.

8. PROCESAMIENTO DE IMAGENES

Cada perfil que se captura con la cámara es iluminado por el rayo láser, cuya imagen se encuentra en el formato RGB. La imagen debe ser transformada a escala de grises para obtener solamente los valores de intensidad, diferenciando las partes del objeto que están iluminadas por el láser y las que no lo están.

Figura 13. Perfil de un objeto en RGB y en escala de grises



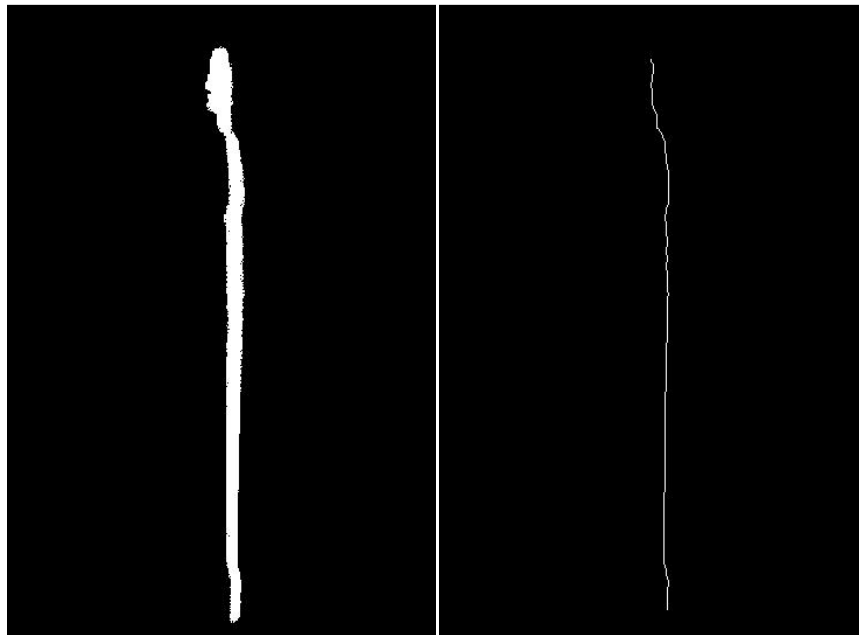
Fuente: Propia.

Habiendo transformado la imagen a escala de grises, se realiza el proceso de binarización, que consiste en dividir los valores de intensidad de la imagen en dos grupos, tomando como referencia un valor de umbral, donde todo valor que sea mayor o igual al umbral es igual a 1, y si el valor de intensidad es menor que el umbral es igual a 0.

$$I_b(x, y) = \begin{cases} 1, & \text{si } I(x, y) \geq U \\ 0, & \text{si } I(x, y) < U \end{cases} \quad (2)$$

$I(x,y)$ es la intensidad de un pixel de la imagen del perfil en escala de grises, mientras que $I_b(x,y)$ es la imagen binaria resultante.

Figura 14. Imagen binaria y esqueletizada,



Fuente: Propia.

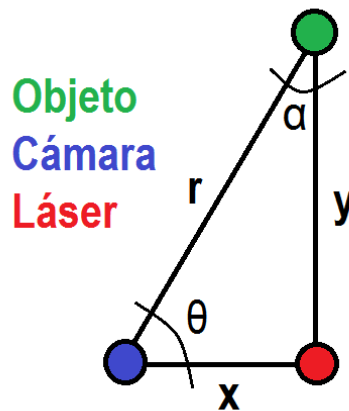
El siguiente paso a realizar es el de transformar el perfil binario a uno de esqueleto binario, que tendrá la misma forma del perfil binario, pero con ancho de un pixel, exhibiendo el centro del perfil original, cuyos valores de distancia horizontal y vertical con respecto a un referencia dada serán utilizados para su posterior análisis.

9. CARACTERIZACIÓN DEL SISTEMA

Para conocer la información geométrica del objeto y procesarla a través del computador es necesario calibrar el sistema, relacionando los pixeles con unidades de longitud físicas. Para poder caracterizar el sistema es necesario aplicar el principio de triangulación en los dispositivos de medida y objetos involucrados en el proceso.

La triangulación consiste en relacionar de forma geométrica la ubicación de los dispositivos y el objeto, los cuales son los vértices de un triángulo imaginario.

Figura 15. Principio de triangulación

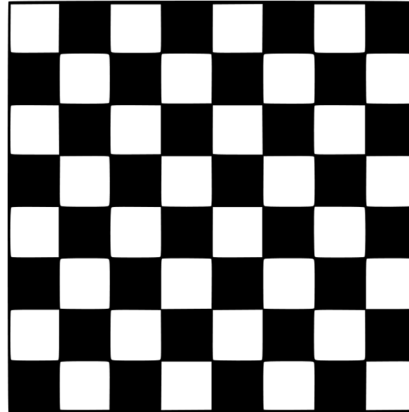


Fuente: Propia.

El diseño implementado ubica la cámara a una distancia de 55mm con respecto al láser, mientras que el láser se encuentra a una distancia Y de 165mm del centro de la base donde serán colocados los objetos a escanear. La distancia X entre el láser y la cámara es fija, pero la distancia Y puede variar ya que depende de la forma del objeto. Como aún no se conoce el valores de R , θ y α , se hallan aplicando las funciones e identidades trigonométricas, donde se obtuvieron los valores de 173.92mm, 71.56° y 18.44 ° respectivamente.

Ahora que se conocen las distancias y ángulos entre los objetos y dispositivos, se procede a la realización de la calibración utilizando un patrón en blanco y negro, el cual se ubicará en el centro de la base rotatoria, con el cual hallaremos los datos necesarios para caracterizar el sistema.

Figura 16. Patrón utilizado para calibración

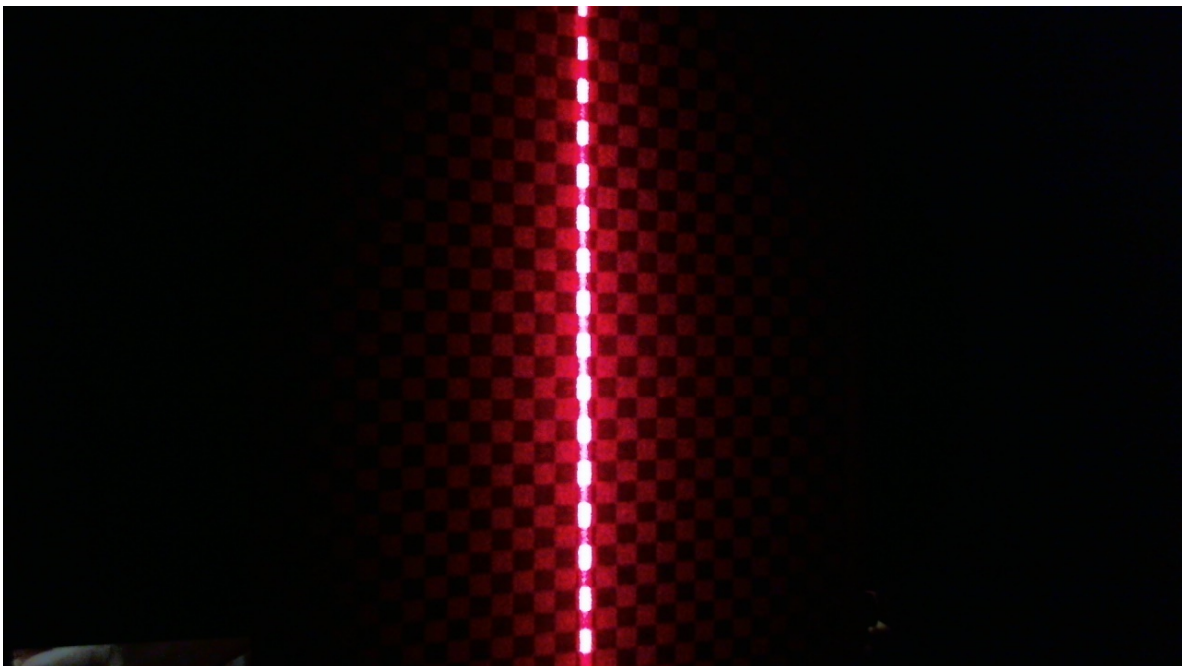


Fuente: Propia.

El patrón utilizado tiene cuadrados de tamaño 5mm*5mm, el cual fue impreso y colocado en una superficie de cartón paja lisa y recta, el cual fue luego colocado ante el láser a una distancia de 165mm.

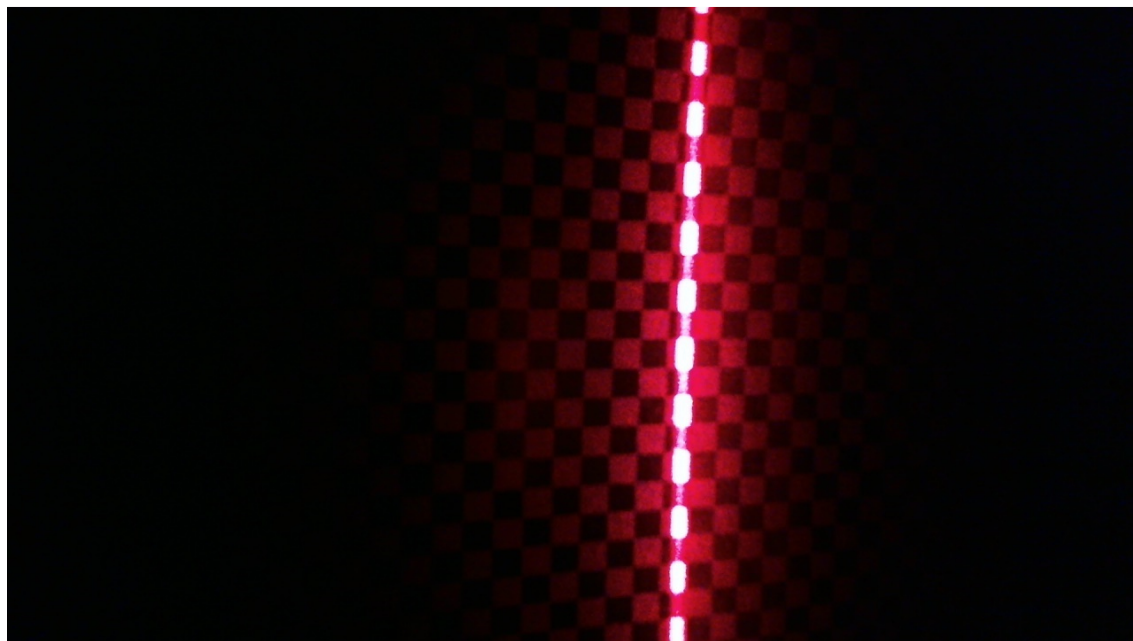
Para la realización de la calibración se acercaba la plataforma que tiene la cámara y el láser desde una distancia de 165mm hasta 55mm, con un paso de 2.5mm, tomando fotos del perfil a medida que se avanzaba.

Figura 17. Patrón iluminado por el láser a 165mm de distancia



Fuente: Propia.

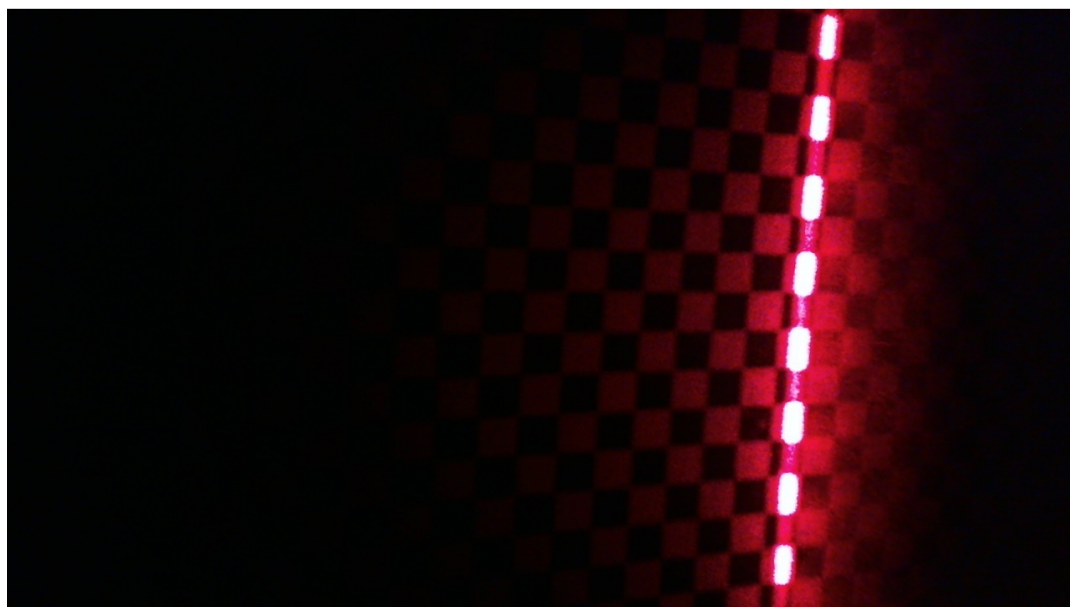
Figura 18. Patrón iluminado por el láser a 100mm de distancia



Fuente: Propia.

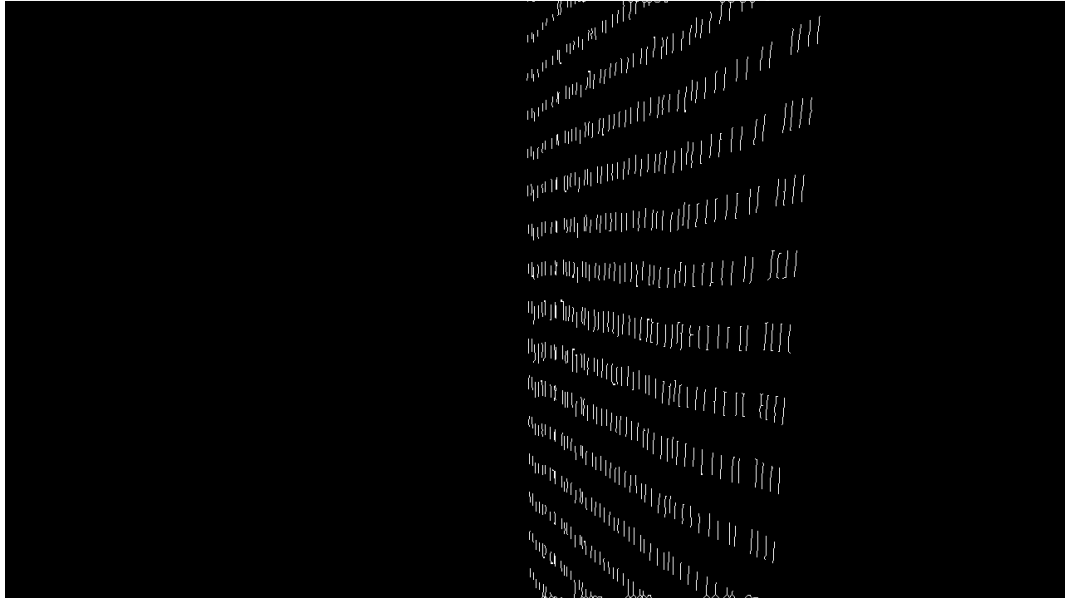
A cada perfil se le realiza el debido procesamiento, convirtiendo primero la imagen de RGB a escala de grises, luego esa imagen en gris se transforma en una imagen binaria en blanco y negro, y finalmente se transforma la imagen binaria en un esqueleto de la misma.

Figura 19. Patrón iluminado por el láser a 55mm de distancia



Fuente: Propia.

Figura 20. Esqueleto de los perfiles superpuestos



Fuente: Propia.

El objetivo es relacionar el desplazamiento hacia la derecha con la profundidad real del objeto, es decir, el radio y la distancia vertical entre los fragmentos de línea y un factor de conversión de pixel a milímetros.

Para conocer el radio de los diferentes puntos que conforman un objeto, los perfiles deben ser medidos a partir de una línea de referencia, la cual corresponde al perfil tomado a una distancia de 165mm. Esto se hace hallando el centroide de cada una de las líneas fragmentadas de las imágenes de muestra superpuestas, ya que al restarse las distancias horizontales de los pixeles con la referencia, se halla una relación entre la profundidad (mm) con la diferencia de pixeles (pixel).

$$Px_d = \frac{\sum_{i=1}^m Px_i}{m} \quad (3)$$

$$X_d = Px_d - P_{Ref} \quad (4)$$

Donde d es la distancia entre el centro del objeto y el dispositivo láser, la cual varía desde 165mm hasta 55mm, a un ritmo de 2.5mm, P_{XD} corresponde a los pixeles del perfil a distancia d y P_{REF} a los pixeles del perfil de referencia, que se encuentra a una distancia de 165mm.

Sabiendo que la distancia entre centroide vertical de cada fragmento de línea que pertenece a un perfil es de 10mm, se halla la relación vertical entre pixeles y

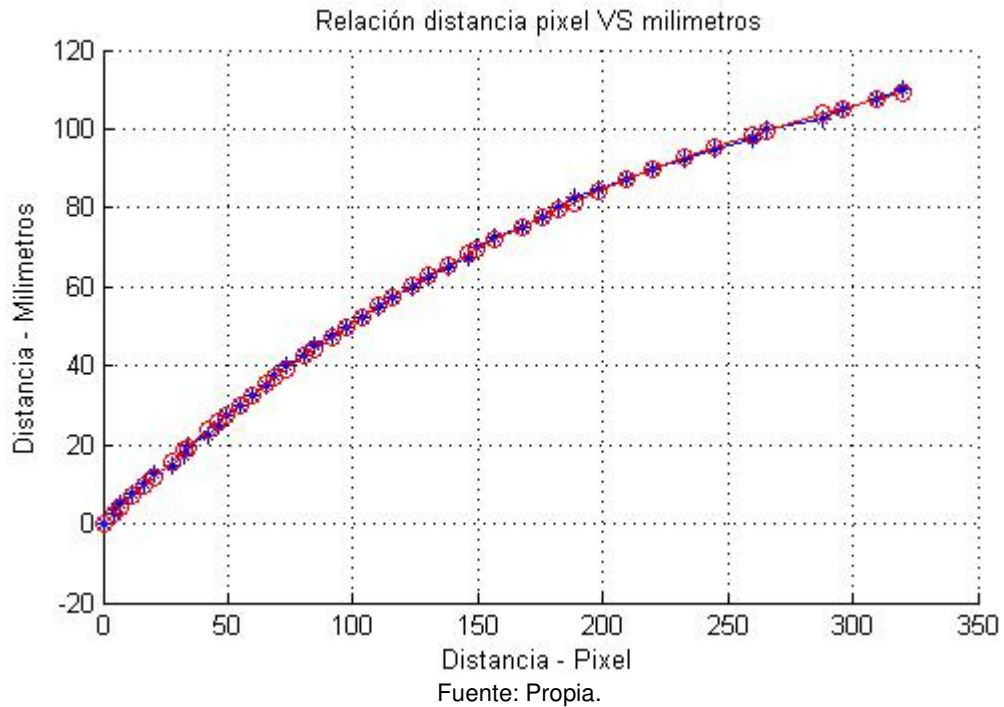
milímetros para cada muestra, restando el centroide cada fragmento con el que le antecede, y luego se saca un promedio de los valores de pixeles obtenidos.

$$P_{yd} = \frac{\sum_{i=2}^n (Py_i - Py_{i-1})}{n} \quad (5)$$

$$F_d = \frac{10mm}{P_d} \quad (6)$$

El valor de P_{yD} (Pixeles promedio en vertical a una distancia d) es el equivalente en pixeles de una distancia vertical de 10mm, por tal razón existe un factor de conversión F_D que relaciona los pixeles con milímetros que depende de la distancia a la que se encuentra cada punto del perfil.

Figura 21. Gráfica de relación de pixeles con milímetros



Los datos de distancia horizontal de pixel a milímetro fueron procesados para hallar un polinomio que definiera dicha relación. El polinomio resultante fue definido de 3° orden en la siguiente ecuación:

$$R = (-7.795 * 10^{-7})P_x^3 - (1.094 * 10^{-4})P_x^2 + 0.611P_x - 0.069 \quad (7)$$

Se compararon los valores de este polinomio con los datos tomados durante la calibración y se halló que la correlación que hay entre ellos es de 0.9998, lo cual nos da un alto grado de confiabilidad.

Hay que tener en cuenta que la función $R(P_X)$ solo entrega la distancia en milímetros del objeto con respecto al láser, por eso de dividirse entre el coseno del ángulo α de 18.44° para hallar la distancia con respecto a la cámara.

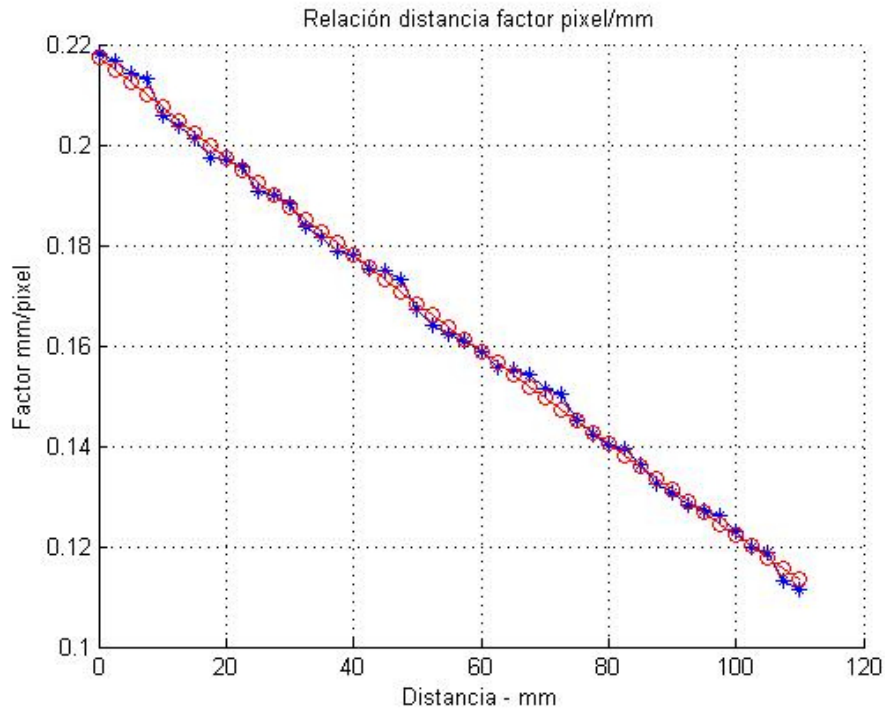
$$R_R = \frac{R}{\cos(18.44^\circ)} \quad \text{donde } R_R : mm \quad (8)$$

Los datos para el factor de conversión de pixeles a milímetros fueron igualmente procesados y se obtuvo un polinomio de 2° orden definido en la siguiente ecuación:

$$F = (6.004 * 10^{-7})R_R^2 - 0.001R_R - 0.217 \quad \text{donde } F : mm/pixel \quad (9)$$

Como se puede evidenciar, el factor de conversión depende del radio real del objeto, además la correlación entre los datos obtenidos por la calibración y los entregados por el polinomio entregan un valor de correlación de 0.9799, lo cual nos permite confiar en la ecuación obtenida.

Figura 22. Gráfica de relación de pixeles con milímetros



Fuente: Propia.

La altura del objeto se obtiene multiplicando el factor de conversión con la posición vertical del pixel:

$$A = P_x * F \quad (11)$$

Ahora que se tienen las ecuaciones que transforman los datos de un perfil de pixeles a milímetros, es posible hallar las coordenadas de cada punto en el espacio, teniendo los parámetros en coordenadas cilíndricas podemos transformarlas a coordenadas cartesianas de la siguiente manera, asumiendo que x es el ancho, y el alto y z la profundidad:

$$x = R_R \cos \theta \quad (12)$$

$$y = A \quad (13)$$

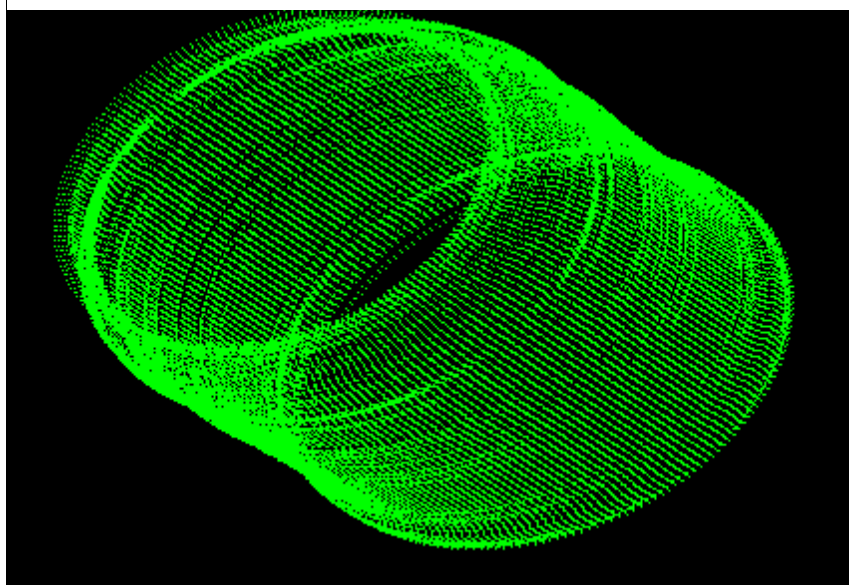
$$z = R_R \sin \theta \quad (14)$$

$$\theta = 1.8n^\circ \quad (15)$$

El valor de θ depende de un factor n , que es el número de perfiles que procesará el sistema, establecido de 1 a 200 y el paso angular que será de 1.8° .

10. GRAFICACIÓN

Figura 23. Graficación nube de puntos OpenGL



Fuente: Propia.

El sistema de graficación se hizo en OpenGL, una herramienta que permite generar gráficas y formas de diferente tipo.

Para poder hacer la graficación se necesitan los datos de forma del objeto, los cuales se encuentran almacenados en una estructura que contiene la ubicación de los puntos que conforman la geometría del objeto y los colores de los mismos, la cual se encuentra definida en la siguiente ecuación:

$$Perfil[n] \begin{cases} Forma[m] \begin{cases} x \\ y \\ z \end{cases} \\ Color[m] \begin{cases} r \\ g \\ b \end{cases} \end{cases} \quad (16)$$

El valor n corresponde a un determinado perfil del objeto, cuyo valor se encuentra entre 1 y 200, mientras que m es el número de puntos que hay por perfil.

Para graficar una nube de puntos con los datos del objeto, solo es necesario aplicar el siguiente pseudocódigo:

Figura 24. Pseudocódigo algoritmo de graficación

```
Para  $i=1$  hasta 200  
  Para  $j=1$  hasta  $m$   
    Graficar  $Perfil[i].Forma[j]$   
  Fin  
Fin
```

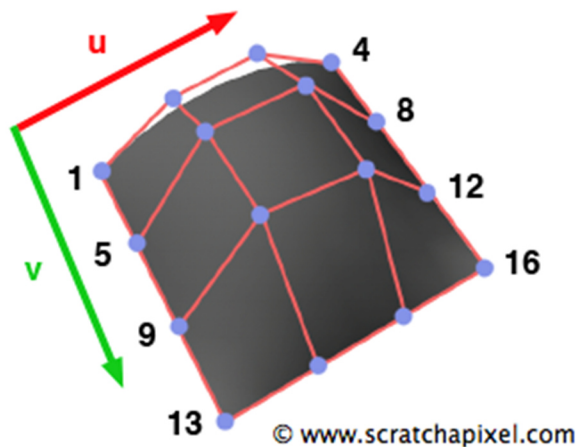
Fuente: Propia.

Como se pudo observar en la Figura 22, para realizar la reconstrucción del objeto se debe hacer una aproximación con los datos obtenidos en el escaneo. Esta aproximación se hace procesando los datos mediante el método de superficies de Bézier, con el cual se pueden crear superficies a partir de una cierta cantidad de puntos.

OpenGL tiene una serie de funciones en sus librerías que hacen esto posible de manera simple, donde para ello se debe tener identificadas las siguientes variables: puntos de control y número de puntos a aproximar.

Los puntos de control en una superficie de Bézier son organizados en una matriz tridimensional tipo $[s][t][3]$, donde s representa los puntos en dirección u , t representa los puntos en dirección v , y el 3 son las coordenadas en x, y, z , tal como se puede observar en la siguiente ecuación y figura:

Figura 25. Puntos de control superficie Bézier



Fuente <http://www.scratchapixel.com/assets/Uploads/Lesson011/I011-Béziersurface.png>

$$Pts. de control \rightarrow forma(s, t, 3) = \begin{bmatrix} xyz_{11} & xyz_{12} & xyz_{13} & \dots & xyz_{1s} \\ xyz_{21} & xyz_{22} & xyz_{23} & \dots & xyz_{2s} \\ xyz_{31} & xyz_{32} & xyz_{33} & \dots & xyz_{3s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ xyz_{t1} & xyz_{t2} & xyz_{t3} & \dots & xyz_{ts} \end{bmatrix} \quad (17)$$

Esta aproximación con superficies de Bézier puede hacerse también con los colores que hay en cada punto, donde los puntos de control quedan de la misma manera, a excepción de la tercera dimensión: $[s][t][4]$. En la tercera dimensión se definen la gama de colores rojo, verde, azul y un valor alfa, que corresponde a la intensidad, representada en la siguiente ecuación:

$$Pts. de control \rightarrow color(s, t, 4) = \begin{bmatrix} rgba_{11} & rgba_{12} & rgba_{13} & \dots & rgba_{1s} \\ rgba_{21} & rgba_{22} & rgba_{23} & \dots & rgba_{2s} \\ rgba_{31} & rgba_{32} & rgba_{33} & \dots & rgba_{3s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ rgba_{t1} & rgba_{t2} & rgba_{t3} & \dots & rgba_{ts} \end{bmatrix} \quad (18)$$

OpenGL tiene un inconveniente a la hora de graficar la superficie de Bézier, el cual es la cantidad de puntos de control que se pueden utilizar, ya que grandes cantidades de puntos de control incrementan la carga computacional haciendo imposible la graficación del objeto. Por eso motivo el objeto fue graficado por partes, teniendo cada parte aproximadamente la misma cantidad de puntos de control, donde cada matriz de forma es de dimensiones $[s \leq 25][t \leq 25][3]$, y las matrices de color son de dimensiones $[s \leq 25][t \leq 25][4]$.

Po lo tanto, para graficar el objeto se debe determinar primero el número de partes en el que se dividirá, teniendo en cuenta que el objeto tiene 200 perfiles con m cantidad de puntos cada uno, siendo 200 un valor que pertenece a la dirección u , y m un valor que pertenece a la dirección v .

El número de partes es determinado bajo las siguientes condiciones:

$$Puntos en u \rightarrow numPtsCtrl_u = \frac{200}{P_u}, \quad \text{donde } numPtsCtrl \leq 25 \quad (19)$$

$$Puntos en v \rightarrow numPtsCtrl_v = \frac{m}{P_v}, \quad \text{donde } numPtsCtrl \leq 25 \quad (20)$$

La cantidad de partes de que se graficarán dependen de los valores elegidos para P_u y P_v . Ahora la graficación del objeto es posible, siendo realizado a través del siguiente algoritmo:

Figura 26. Algoritmo Graficación método de Bézier

```
NPCu = 200/Pu
NPCv = m/Pv
Para n=1 hasta Pv
  PasoV = n*NPCv - 1
  PasoU = 0
  Para k=1 hasta Pu
    Para i=1 hasta NPCv+1
      Para j=1 hasta NPCu+1
        PtsCtrlF[j][i] = Perfil[j + PasoV].Forma[i + PasoU]
        PtsCtrlC[j][i] = Perfil[j + PasoV].Color[i + PasoU]
      Fin
    Fin
    Graficar Bézier Forma (PtsCtrlF)
    Graficar Bézier Color (PtsCtrlC)
    PasoU = PasoU + NPCu
  Fin
Fin
```

Fuente: Propia.

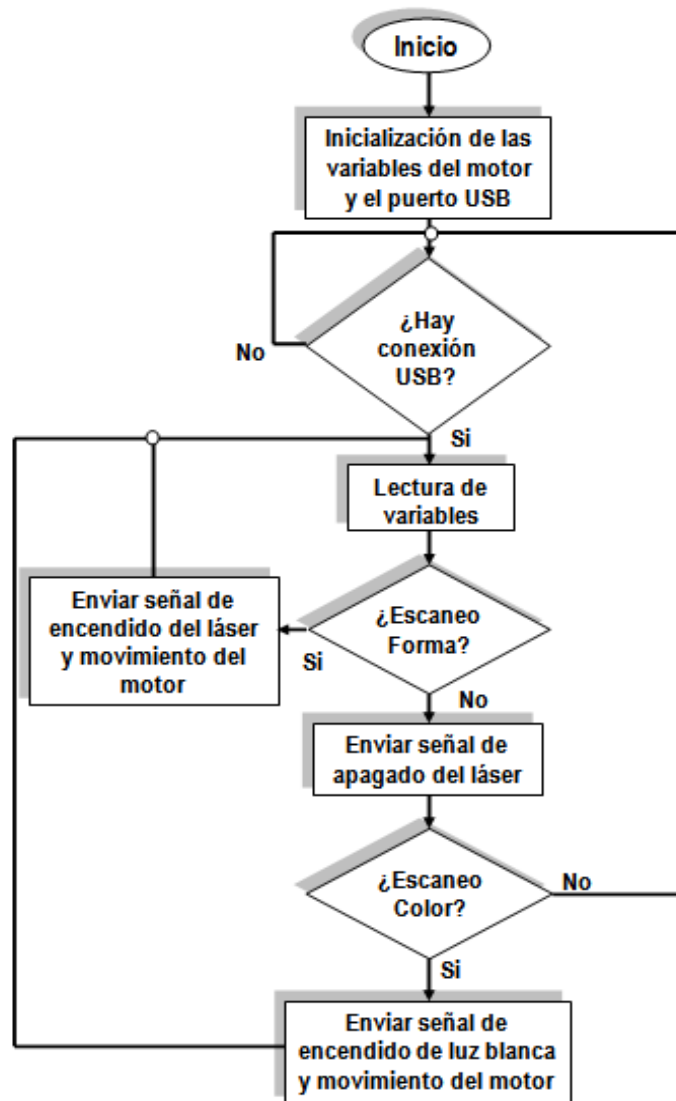
Este algoritmo puede ser utilizado para graficar cualquier objeto por el método de superficies de Bézier en cualquier lenguaje de programación.

11. SOFTWARE

11.1 DIAGRAMAS DE FLUJO

11.1.1 PROGRAMACION ARDUINO

Figura 27. Diagrama de flujo programación arduino

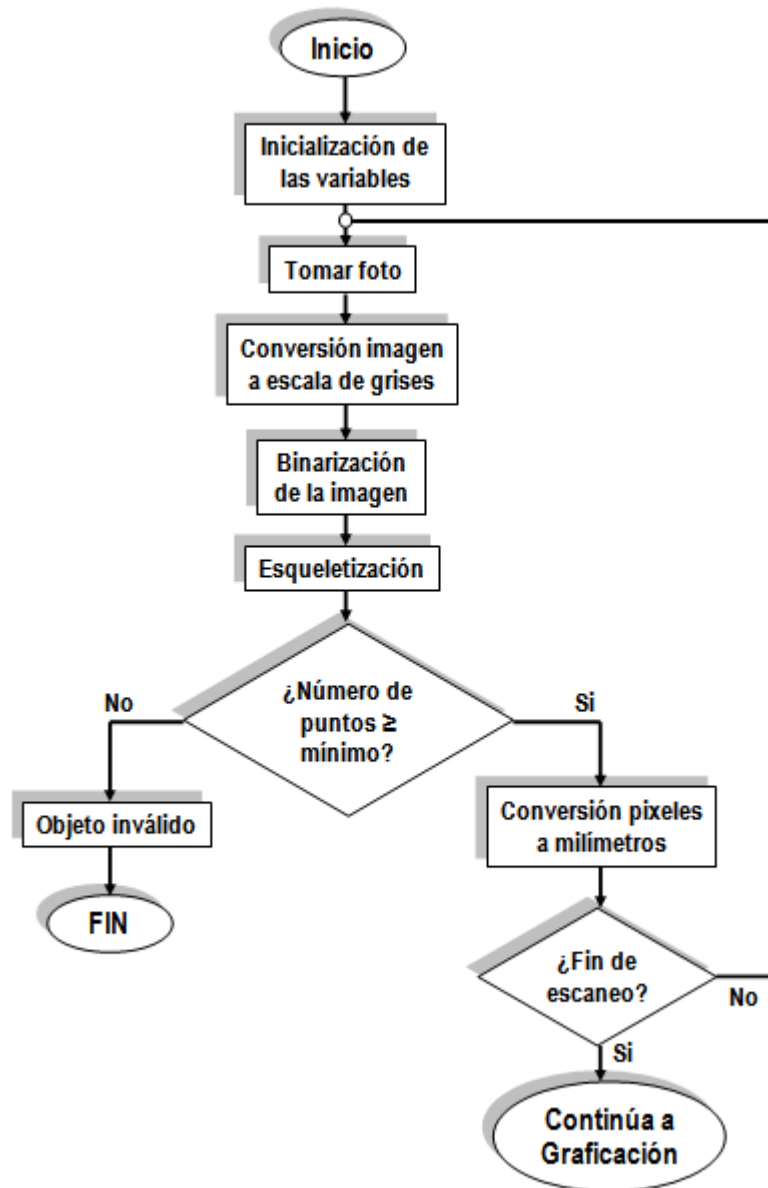


Fuente: Propia.

El software del arduino se encuentra en un ciclo siempre y cuando haya conexión USB, este verifica si las señales son de procesamiento de la forma del objeto o del color del objeto. En el primer caso se manda señales al driver y de ahí al motor para que realice el respectivo movimiento, de la misma forma se le mande un alto para que el láser sea en encendido. En el caso del procesamiento de color es necesario mandarle un bajo para el apagado del láser y el encendido de los leds de luz blanca y al motor el cual gira nuevamente para procesar así el color objeto.

11.1.2 PROGRAMACION PROCESAMIENTO DE IMÁGENES

Figura 28. Diagrama de flujo programación procesamiento de imágenes



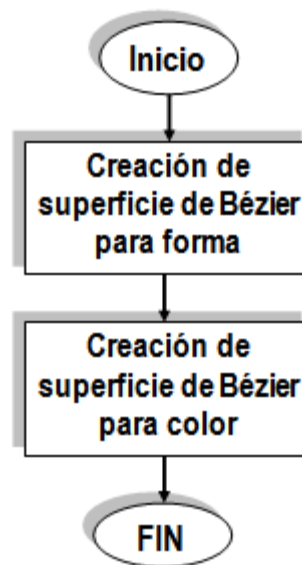
Fuente: Propia

En este diagrama de flujos podemos connotar los pasos y ciclos necesarios para el correcto procesamiento de las imágenes. Para empezar se toma la foto, esta es

almacenada en una variable para luego ser transformada a escala de grises, con el fin de obtener solamente los valores de intensidad de la imagen, luego los niveles de intensidad de esta son pasados a binario, separando así los niveles de intensidad mediante un umbral, donde si es mayor o igual al umbral, los valores de intensidad serán uno y si es menor serán cero, luego se le extrae el esqueleto al perfil binario el cual representa el núcleo del perfil.

11.1.3 PROGRAMACION GRAFICACIÓN

Figura 29. Diagrama de flujos programación graficación



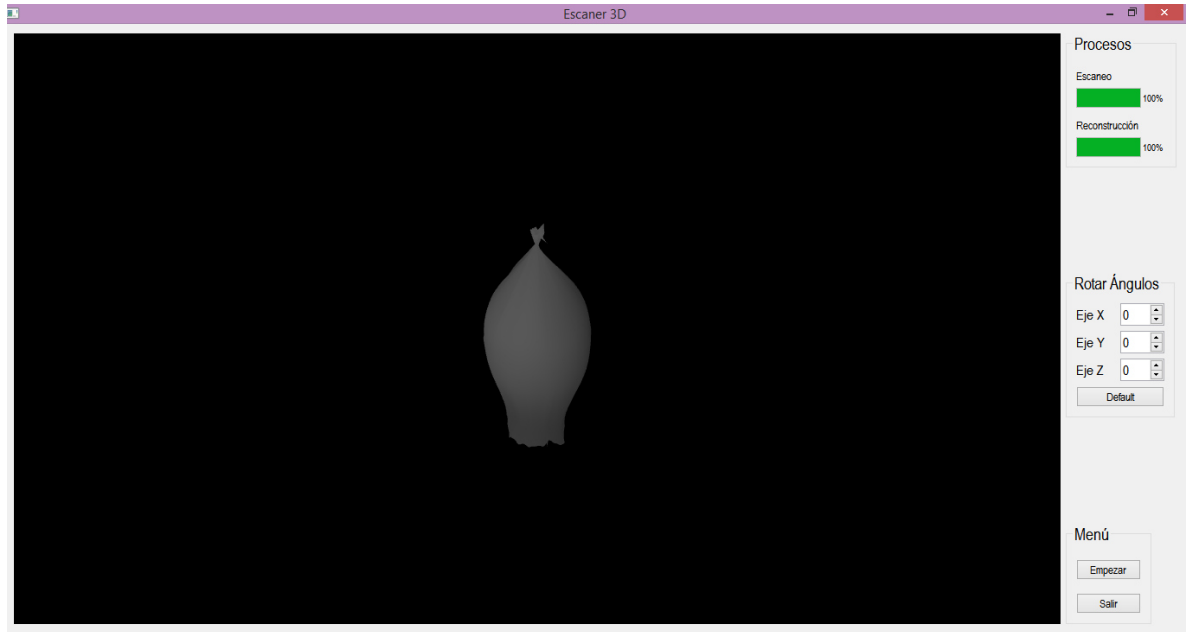
Fuente: Propia.

En este segmento del programa al esqueleto del perfil se le extrae específicamente una cantidad determinada de puntos, separados equitativamente entre si y luego es tomada otra foto al mismo perfil pero a color y se extraen los valores de color específico de cada punto.

Los puntos obtenidos han de ser procesados para la graficación del objeto, implementado el método de las superficies de Bézier, donde estos puntos pasan a ser transformado en puntos de control, con los cuales se hace una reconstrucción del objeto a través de una aproximación mediante curvas

11.2 INTERFAZ

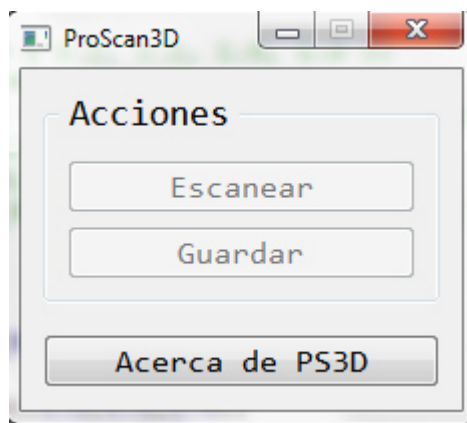
Figura 30. Interfaz sistema de escaneo tridimensional



Fuente: Propia.

La primera versión del software (Figura30) del sistema de escaneo tridimensional constaba de 2 barras de progreso indicadoras del estado en el que encuentra el escaneo, Un menú de rotación de ángulos para mejorar la visualización del objeto, un botón para iniciar el escaneo y otro para salir del programa. Esta primera versión fue descartada por dificultades encontradas en el proceso de envío de señales y la graficación dado a una sobre carga en la memoria dinámica del computador, por lo cual fue necesario separar el software en dos etapas, un segmento para escaneo y procesamiento de imágenes, y otro para graficación.

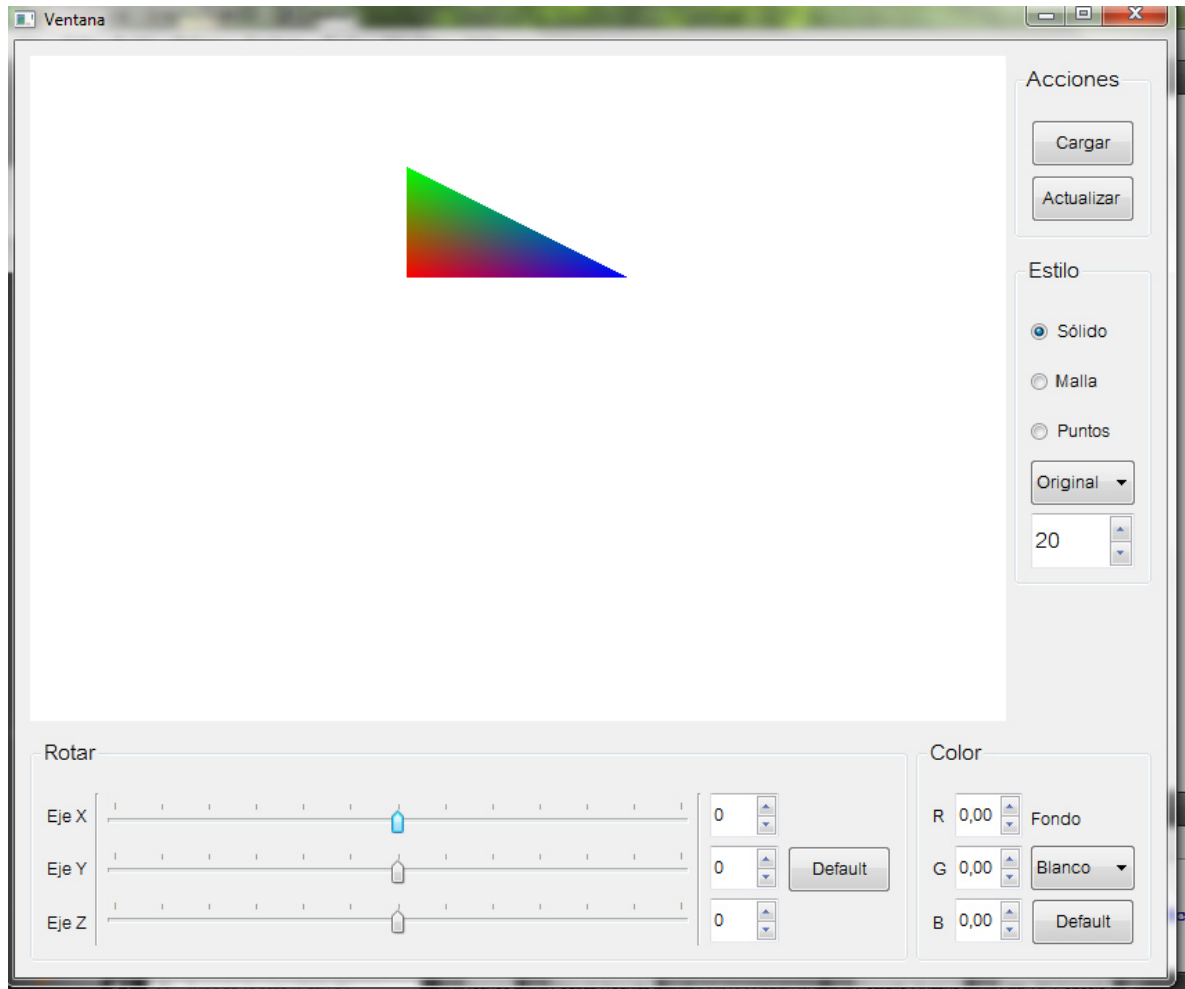
Figura 31. Interfaz para el escaneo y procesamiento de imagen



Fuente: Propia

En la de procesamiento y escaneo tenemos una interfaz sencilla (figura 31) la cual consta de tres botones cada cual con validaciones necesarias. El botón Escanear solo estará disponible cuando el arduino se encuentre conectado, el botón Guardar solo será activado al final del procesamiento de la imagen, y el botón acerca de PS3D (ProScan 3D) despliega una ventana que muestra información acerca del sistema de escaneo tridimensional.

Figura 32. Interfaz Grficación



Fuente: Propia

La interfaz de la graficación en su estado inicial muestra un triángulo multicolor para demostrar al usuario como puede interactuar con la figura, esta cuenta con distintas secciones.

El menú Rotar cuenta con tres barras de desplazamiento las cuales permiten al usuario rotar en distintos ángulos el objeto con respecto a los tres ejes. El botón Default llevara los ángulos de rotación a su estado inicial.

El menú Estilo consta de tres modos de visualización: Sólido, el cual muestra el objeto en su estado original; Malla muestra el objeto visualizado en forma de redes y Puntos lo visualiza en forma de nube de puntos. La cantidad de puntos puede ser modificada con el cuadro de texto el cual muestra un valor por defecto de veinte puntos.

El menú de Acciones consta de dos botones: Cargar y Actualizar. Cargar muestra una ventana de Windows que permite seleccionar un archivo guardado, que contiene los datos de un objeto escaneado y procesado anteriormente, mientras que el botón Actualizar muestra el objeto cargado en el visor de gráficos.

El menú color consta de tres cuadros de texto, donde cada uno controla un color en el espacio RGB, dando así la oportunidad al usuario de obtener una gráfica en diferentes colores. También consta de un menú de despliegue que permite cambiar el fondo de blanco al negro o viceversa del visor de gráficos.

12. CHASIS

El sistema de escaneo tridimensional fue hecho en madera, fue diseñado para tener una forma cubo con unas dimensiones de 40cmx35cmx50cm (figura 34) y un espesor de 0.5cm. Este se encuentra pintado de negro en su totalidad para su propio funcionamiento. Consta de un disco rotatorio con radio de 15cm, consta con las bases necesarias para los leds de luz blanca y una base para la cámara y el láser(figura 33).

Figura 33. Vista interior del sistema de escaneo tridimensional



Fuente: Propia.

Figura 34. Vista exterior del chasis

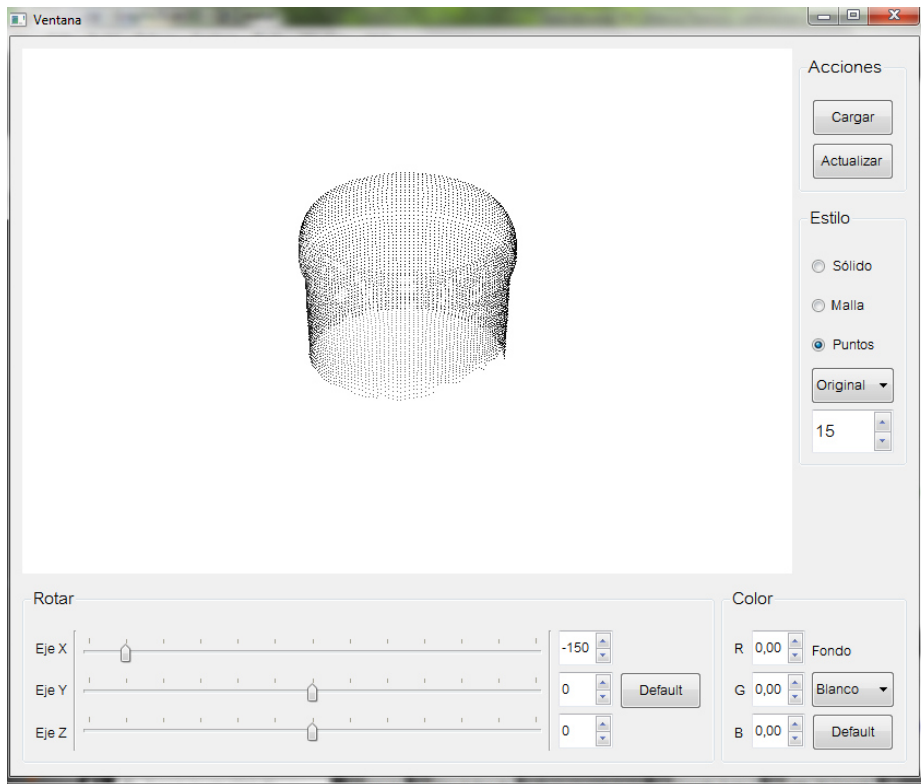


Fuente: Propia

13. RESULTADOS

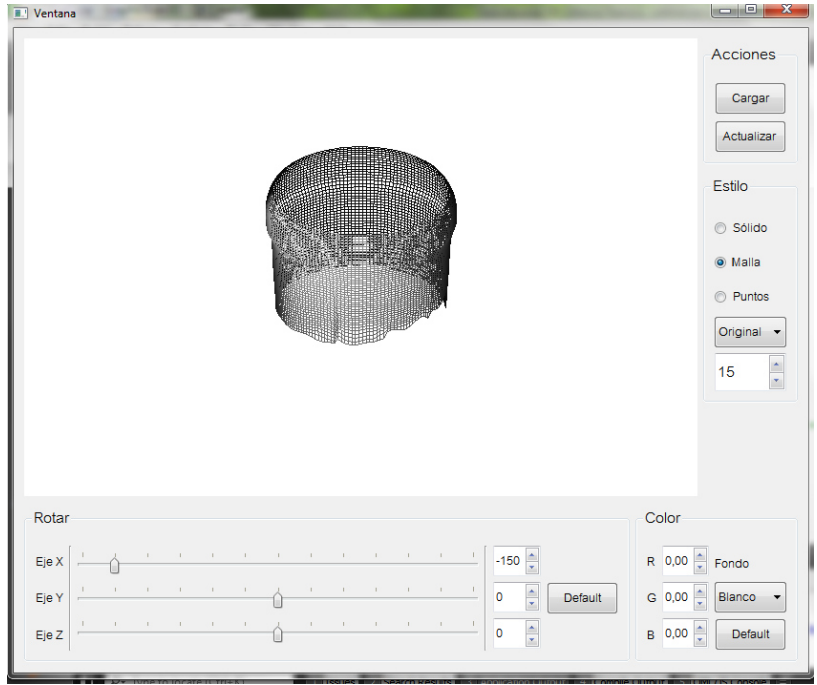
Los primeros resultados del sistema de escaneo tridimensional (figura 35) se realizaron únicamente para obtener su forma para poseer constancia de su propio funcionamiento, A partir de estos se realizaron algunas mejoras en el software y se corrigieron algunos detalles en el posicionamiento del objeto, mejorando así su desempeño y precisión.

Figura 35. Objeto Tridimensional representado en una nube de puntos



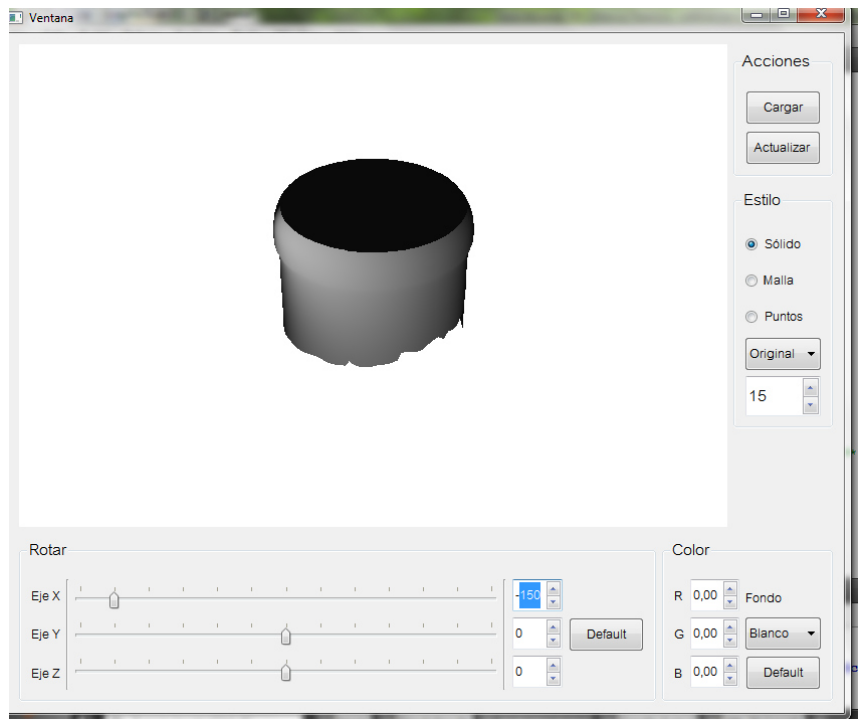
Fuente: Propia

Figura 36. Objeto Tridimensional representado en forma de malla



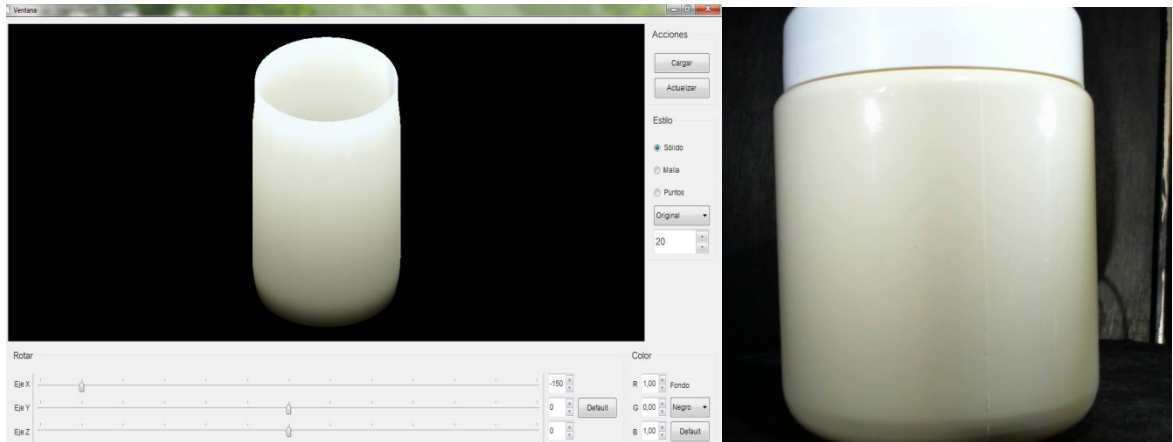
Fuente: Propia

Figura 37. Objeto escaneado sin color



Fuente: Propia

Figura 38. Objeto escaneado a color

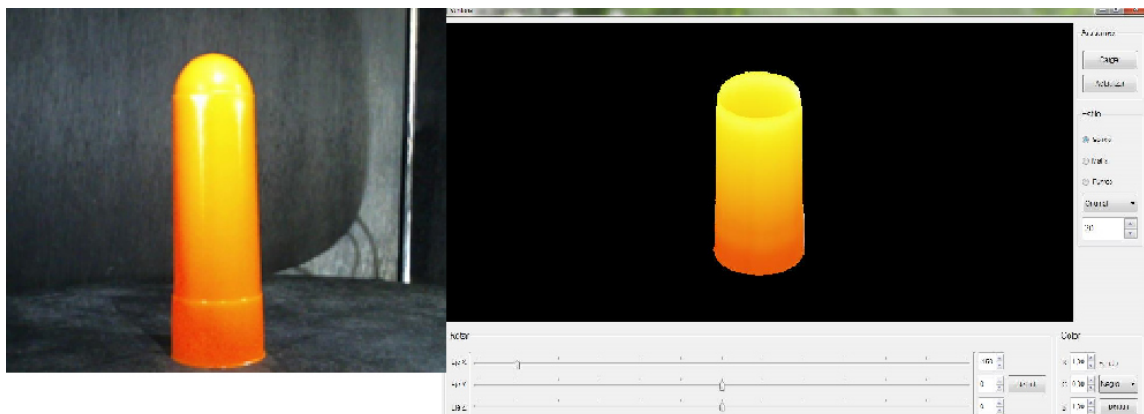


Fuente: Propia.

En las siguientes pruebas se realizaron escaneos de objetos a color que se encontraban dentro del rango de las delimitaciones. Como se puede observar los resultados (Figura 36, Figura 37) son equivalentes a su forma y a su color mostrando así la efectividad del software y un grado de precisión elevado. El tiempo de escaneo del mismo es menor a 5 minutos logrando así un consumo mínimo de energía y una rápida graficación.

La calidad de los resultados depende de factores como la iluminación, la reflectividad del objeto y su rugosidad.

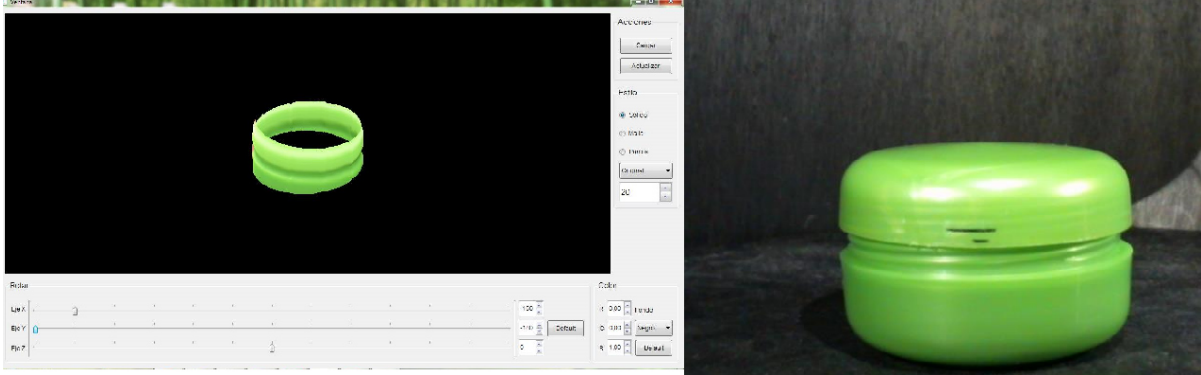
Figura 39. Objeto escaneado a color



Fuente: Propia.

Algunos resultados del sistema de escaneo tridimensional se mostraron de forma errónea (figura 40), estos resultados se pueden dar por su concavidad, rugosidad y dimensiones que evitaban así que el rayo láser alcanzara el objeto en su totalidad distorsionando los resultados obtenidos.

Figura 40. Objeto escaneado fuera de las delimitaciones



Fuente: Propia

14. CONCLUSIONES

Para lograr un sistema de escaneo tridimensional, fue necesario utilizar el método de triangulación para su calibración, el procesamiento de imágenes se realizó con la librería OpenCV, que provee herramientas importantes que permitieron llevar cada toma a una escala de grises y obtener un perfil esqueletizado de cada una de estas, para así conseguir los puntos necesarios para su graficación, la cual es lograda a través de la librería OpenGL, donde se crean superficies de Bézier para obtener la forma y el color del objeto.

El sistema de escaneo tridimensional permite escanear objetos con una altura de 10cm, una profundidad y ancho de 16 cm, un rango de medida amplio y una densidad de la nube de puntos que puede ser variada en el software, dándole así un alto grado de exactitud y confianza. Sin embargo, existen limitantes en objetos con concavidades o agujeros, debido al principio en el cual se basa. La calidad depende de factores como la iluminación, la reflectividad del objeto y su rugosidad. Hay materiales que reflejan gran cantidad de luz, otros que absorben gran parte del haz, incluso otros que esparcen o distorsionan el haz por su superficie debido a la rugosidad de ésta. Por lo tanto, en la calidad de la reconstrucción influye el grado de opacidad del objeto, suavidad de la superficie y color de ésta. Así, los mejores resultados se obtuvieron para objetos totalmente opacos, de colores claros y superficies suaves.

Se trabajó con imágenes en escala de grises porque hay una buena diferenciación de intensidad entre el fondo de la imagen y la proyección del plano láser sobre el objeto, mostro ser necesario tener un láser con una buena intensidad y una fuente de voltaje que suministrara correctamente la tensión adecuada a cada uno de los circuitos integrados.

El sistema realizado va encaminado a la digitalización de objetos abarcando 360° que permite obtener más cantidad de información debido al mayor campo de acción, haciéndolo muy útil en la animación tridimensional y la reproducción de obras de arte.

15. RECOMENDACIONES

Para un proyecto de este tipo se recomienda utilizar un sistema de control para poder realizar la calibración del sistema de forma más precisa, ya que al realizarse de forma manual, se presentaron errores causados por factores humanos. Dicho sistema de control debe ser diseñado para tener márgenes de error despreciables, menores a 1mm.

Se sugiere que la construcción del chasis y las partes mecánicas del dispositivo sean hechas por expertos o personas con habilidad, para asegurar que todas las piezas encajen de acuerdo a lo diseñado y no causen errores de medición y procesamiento.

La base giratoria donde se ubica el objeto debe estar más alejada de la cámara y el láser, para poder escanear objetos con volúmenes mayores a los límites establecidos en este proyecto. Además, el láser y la cámara deben estar bien alineados con el centro de la base giratoria, para evitar errores de medición durante la calibración y el escaneo de un objeto.

La implementación de fuentes de alimentación que generen niveles de corrientes mayores a los que se utilizan, asegura que no habrá caídas de tensión en alguno de los circuitos electrónicos utilizados. Colocar juntos en una misma placa los circuitos más compatibles, como el de control de motor, la fuente y el circuito aislador, después haber sido comprobado su funcionamiento, evitan que haya falso contacto y ruido por la conexión de señales y tierras.

La implementación de dispositivos láser con proyección de multilínea o círculos concéntricos, permitiría procesar más datos por cada captura que se hace del objeto, obteniendo datos más precisos.

Un buen control de la iluminación de la franja láser y las luces blancas, ayudan a tener un mejor procesamiento de forma, color y textura del objeto. Además la utilización de más puntos de control, permite generar gráficas más detalladas y de mayor calidad del objeto escaneado.

16. REFERENCIAS

ANDREWS, Larry y PHILLIPS, Ronald. *Laser beam propagation through random media*. 2 ed. Washington: SPIE, 2007. Pp 4-6, 15.

BAKER, M. Pauline y HEARN, Donald. *Gráficos por computador con OpenGL*. Madrid: Pearson Education, 2006.

BAURECHT, H; BIEMER E; BROCKMANN, G; GÜHRING, M; KOVACS, L; PAPADOPULOS NA; SADER R; SCHWENZER-ZIMMERER K; ZEILHOFER HF y ZIMMERMANN, A;. *Three-dimensional recording of the human face with a 3D laser scanner*. Technical University Munich: Munich, 2006.

BRADLEY, Beverly. *The design, implementation, and validation of a three-dimensional laser light-sectioning scanner for biomedical purposes*. D. Carleton University (Canada)

BRADSKI, Gary y KAEHLER, Adrian. *Learning OpenCV Computer Vision with the OpenCV Library*. Sebastopol: O'Reilly, 2008.

CASTILLO, Herman y DIAZ, Andrés, *Diseño de un sistema para captura de geometrías 3d con técnicas de visión y luz estructurada*. Trabajo de grado Ingeniero Electrónico. San Juan de Pasto: Universidad de Nariño. Facultad de Ingeniería. Departamento de Ingeniería Electrónica, 2009. 131 p.

CORDERO, Juan y CORTES, José, *Curvas y superficies para modelado geométrico*. Madrid: RA-MA editorial, 2002.

DAWSON, Michael. *Beginning C++ game programming*. Boston: Thomson, 2004. Pp 1-3.

DEITEL, Harvey y DEITEL, Paul. *Como programar en C++*. México: Pearson Education, 2009.

ESQUEDA, José y PALAFOX, Luis. *Fundamentos de procesamiento de imágenes*. Mexicali: Universidad de Baja California, 2005.

KHRONOS GROUP, *OpenGL overview* [artículo web]. OpenGL. [Consultado 22 abr. 2013]. Disponible en: www.opengl.org/about/

LU, Zheming; LUO, Hao; WANG, Pinghui y YU, Faxin. *Three-dimensional model analysis and processing*. Hangzhou: Springer, 2010. Pp 17.

TEUTSCH, Christian. *Model-based analysis and evaluation of point sets from optical 3D laser scanners*. Shaker Verlag, 2007.

YE, Hui Jian y ZHOU, Cheng Hui. *Three-dimensional laser color scanner hardware subsystem design*.