
INSTITUTO TECNOLÓGICO DE CHIHUAHUA

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

**TALLADO EN MADERA CON
TECNOLOGÍA LÁSER DE MODELOS
TRIDIMENSIONALES DE OBJETOS DE
TAMAÑO REDUCIDO**

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRA EN INGENIERÍA MECATRÓNICA

PRESENTA:

ING. MARIANA SÁENZ PADILLA

DIRECTOR DE TESIS:
DR. ISIDRO ROBLEDO VEGA



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



CHIHUAHUA, CHIH., MEXICO, ENERO 2020

Chihuahua, Chih., a 20 de Enero del 2020

C. MARIANA SÁENZ PADILLA

P r e s e n t e .

Por este conducto le comunico que esta División de Estudios de Posgrado e Investigación le ha concedido la autorización, para la impresión de su Tesis de Grado de Maestría, cuyo título es:

"TALLADO EN MADERA CON TECNOLOGÍA LÁSER DE MODELOS TRIDIMENSIONALES DE OBJETOS DE TAMAÑO REDUCIDO"

Con el siguiente contenido de capítulos:

- I. ANTECEDENTES
- II. MARCO TEÓRICO
- III. MODELADO
- IV. SISTEMA DE TALLADO CON TECNOLOGÍA LÁSER
- V. EXPERIM
- VI. CONCLUSIONES

"LA TECNICA POR EL ENGRANDECIMIENTO DE MEXICO"

A t e n t a m e n t e

MF. LUIS CARDONA CHACÓN.
JEFE DIVISION DE ESTUDIOS DE POSGRADO.

Chihuahua, Chih., a 28 de enero de 2020

M.F. Luis Cardona Chacón.
Jefe de la División de Estudios de
Posgrado e Investigación

PRESENTE.

Por medio de la presente notificamos a Usted que, en cumplimiento parcial de los requerimientos para la obtención del Grado de Maestría en Ingeniería Mecatrónica, ha sido aprobado y aceptado para su impresión el documento de tesis del C. **MARIANA SÁENZ PADILLA**. Cuyo título es:

**"TALLADO EN MADERA CON TECNOLOGÍA LASER DE MODELOS
TRIDIMENSIONALES DE OBJETOS DE TAMAÑO REDUCIDO"**

Agradeciendo de antemano la atención prestada a la presente, quedamos de Usted:

Dr. ISIDRO ROBLEDO VEGA.
Director de tesis.

M.C. CARMEN LETICIA GARCIA MATA.
Miembro del comité revisor

Dr. PEDRO MARQUEZ
Miembro del comité revisor

Dr. ROGELIO ENRIQUE BARAY ARANA.
Miembro del comité revisor
c.c.p. - Interesado.
- Comité revisor.
- Archivo.

Chihuahua, Chih., a 28 de enero de 2020

Dra. Elena Álvarez-Buylla
Director de CONACYT.

Atención a Mtra. María José Rhi Sausi Garavito

Dirección Adjunta de Posgrados y Becas

P r e s e n t e.

Por este conducto aprovecho la ocasión para saludarlo e informarle que a la fecha he obtenido el Grado de Maestría en Ingeniería Mecatrónica en la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua. Motivo por el cual agradezco todo el apoyo brindado por esta Institución que Usted representa, el otorgamiento de esta beca-crédito permitió dedicarme de tiempo completo a la realización de mis estudios de Posgrado y de esta manera lograr el cumplimiento del objetivo principal del convenio establecido.

Sin otro particular por el momento, me es grato quedar de Usted como su seguro servidor, no sin antes reiterar mi agradecimiento. Muchas Gracias!!!

A t e n t a m e n t e

MARIANA SAENZ PADILLA
Ex becario CONACYT CVU: 811544

c.c.p M.F. Luis Cardona Chacón.
Jefe de la División de Posgrado e Investigación

AGRADECIMIENTOS

A Dios, que sabe que esta tesis es un milagro.

A toda mi familia que fue mi mecenas y mayor apoyo de todas las maneras posibles, animándome de inicio a fin.

A mis amigos que se preocupaban conmigo, los que me hacían preocuparme y a los que me ayudaban a olvidar mis preocupaciones.

A mis profesores, por todas sus enseñanzas, especialmente a mi asesor por su paciencia, dedicación y toneladas de correcciones.

A CONACyT, por haber depositado más que solo su confianza.

“El estudioso es el que lleva a los demás a lo que él ha comprendido: la verdad”

Santo Tomas de Aquino

RESUMEN

TALLADO EN MADERA CON TECNOLOGÍA LÁSER DE MODELOS TRIDIMENSIONALES DE OBJETOS DE DIMENSIONES REDUCIDAS

Mariana Sáenz Padilla
Maestra en Ingeniería Mecatrónica
División de Estudios de Posgrado e Investigación del
Instituto Tecnológico de Chihuahua
Chihuahua, Chih. 2019
Director de Tesis: Dr. Isidro Robledo Vega

La ingeniería inversa nos permite hacer la recreación de modelos tridimensionales y puede ser muy útil en la industria si mediante información obtenida se elabora una pieza en físico. En muchas ocasiones la recreación de piezas se realiza mediante un sistema CNC y herramientas de corte distintas para cada material, o bien, son creadas por medios aditivos como impresoras 3D que no equiparan las propiedades mecánicas del material en una sola pieza. El tallado en madera con tecnología láser es un proceso secuencial y se realiza con una única herramienta de trabajo.

Se desarrolló el prototipo de un sistema de tallado en madera con tecnología láser de modelos tridimensionales de objetos de dimensiones reducidas. En la primera etapa se analizaron diferentes métodos para generar modelos tridimensionales de objetos, en el primero método se usó un escáner tridimensional para capturar datos de objetos reales, en el segundo método se generan datos tridimensionales por medio de código y en el último método se utiliza un programa de CAD para el diseño del modelo tridimensional. Se desarrolló un procedimiento que permite procesar los datos del modelo tridimensional del objeto para ser tallado en madera. Este procedimiento obtiene rebanadas de las nubes de puntos tridimensionales y se genera el Código G de cada rebanada que es procesado por un sistema CNC. En la segunda etapa se desarrolló el sistema de tallado de madera con tecnología láser, se utilizó una impresora 3D como sistema CNC y se sustituyó el extrusor con un módulo láser.

Se realizaron experimentos de tallado en madera de objetos, se generó el modelo tridimensional de un objeto con superficies irregulares utilizando un escáner tridimensional, se obtuvo el Código G de las rebanadas de la nube de puntos y se realizaron pruebas con el sistema de tallado. También se generó el modelo tridimensional de una pirámide de base rectangular por código, se obtuvieron las rebanadas de la nube de puntos y se obtuvo el Código G para cada rebanada. Con los datos de la pirámide se realizaron varias pruebas que permitieron determinar el ajuste de parámetros de funcionamiento del sistema de tallado utilizando diferentes configuraciones y tipos de madera.

ÍNDICE

Capítulo I. ANTECEDENTES.....	2
Capítulo II. MARCO TEÓRICO.....	6
1.1 Adquisición de Datos Tridimensionales.....	6
1.1.1 Adquisición por Contacto	6
1.1.2 Técnicas Fotográficas	7
1.1.3 Representación de Datos Tridimensionales	10
1.2 Procesamiento Digital de Datos Tridimensionales	11
1.2.1 Filtrado de nubes de puntos.....	11
1.2.2 Reconstrucción de Superficies	13
1.2.2.1 Mallas Triangulares	13
1.2.2.2 Mallas poligonales	17
1.2.2.3 NURBS	18
1.2.2.4 Refinado de mallas tridimensionales	20
1.3 Tecnología Láser	21
1.3.1 Tipos de Láser	22
1.3.1.1 Gas.....	22
1.3.1.2 Líquido.....	22
1.3.1.3 Sólidos aislantes.....	23
1.3.1.4 Semiconductores	23
1.3.2 Maquinado con Rayos Láser.....	23
1.4 Clasificación de las Maderas.....	25
1.4.1 Las Maderas y el Láser	27
1.5 Máquinas de Control Numérico Computarizado CNC	28
1.5.1 Sistema de control	28

1.5.2 Control digital	29
1.5.2.1 Actuadores	29
1.5.2.2 Sensores	32
1.5.2.3 Controlador.....	33
1.5.3 Dispositivos de Control Numérico	33
1.5.3.1 Sistema Cartesiano	34
1.5.3.2 Sistema de programación	34
1.5.3.3 Recorrido continuo	35
1.6 Impresoras 3D	36
1.7 Grabadoras láser	39
Capítulo III. MODELADO TRIDIMENSIONAL DE OBJETOS.....	41
1.8 Generación del Modelo por Software	41
1.9 Generación de la Pieza con Escaneo Tridimensional	42
1.10 Generación del Modelo Tridimensional del Objeto	45
1.11 Rebanado de la Nube de Puntos	46
1.12 Generación del Código G.....	49
1.13 Ajustes al Código G	53
Capítulo IV. SISTEMA DE TALLADO CON TECNOLOGÍA LÁSER.....	59
1.14 Sistema CNC	59
1.15 Módulo Láser	61
1.16 Adaptación del Módulo Láser en la Impresora 3D.....	62
Capítulo V. EXPERIMENTOS DE TALLADO EN MADERA.....	65
Capítulo VI. CONCLUSIONES.....	71
Capítulo VII. BIBLIOGRAFÍA	73
Anexo 1	77
Anexo 2.....	79

Anexo 3...	83
Anexo 4...	84

LISTA DE FIGURAS

Figura 1.1 Etapas del sistema desarrollado a) Adquisición datos tridimensionales del objeto. b) Procesamiento digital de los datos tridimensionales. c) Tallado de la figura en un bloque de madera por medio del rayo láser.....	3
Figura 2.1 Escáner de contacto “FaroArm Quantum”.	7
Figura 2.2 Escáneres de contacto Kreon de varios tamaños.	7
Figura 2.3 Adquisición de imágenes mediante un sistema LIDAR aéreo.	8
Figura 2.4 Sistema de Visión en estéreo con 3 cámaras.	8
Figura 2.5 Sistema de adquisición por medio de patrones de luz estructurada.	9
Figura 2.6 Sistema de adquisición por medio de la medición de tiempo de vuelo.....	9
Figura 2.7 Nube de puntos del objeto, mallado de la superficie y el objeto generado dándole color a las superficies.....	10
Figura 2.8 Imagen de profundidad visualizada en niveles de gris.	12
Figura 2.9 Nube de puntos tridimensionales (a) con ruido y (b) filtrada.	12
Figura 2.10 De diagrama Voronoi.	14
Figura 2.11 Triangulación Delaunay de diagrama de Voronoi de la Figura 2.10.	14
Figura 2.12 Superposición de Figura 2.10 y la Figura 2.11.....	16
Figura 2.13 a) No se cumplen las condiciones de Delaunay y en b) si se cumplen.....	16
Figura 2.14 Demostración de cómo se hace el cambio del vértice interior buscando que las condiciones Delaunay se cumplan.....	17
Figura 2.15 Malla poligonal.....	18
Figura 2.16 Diseño de barcos usando splines.	18
Figura 2.17 Refinado de mallado tridimensional.	20
Figura 2.18 Diferencias entre las propiedades del láser y la luz natural.....	21
Figura 2.19 (a) Diseño común de máquinas de tallado láser con sus partes principales. (b) Máquina láser realizando tallado en madera.	24

Figura 2.20 Micrografías de la sección transversal de diferentes tipos de madera.	26
Figura 2.21 Configuración de lazos de control: abierto y cerrado.	29
Figura 2.22 Servomotor con sus partes principales.	31
Figura 2.23 Diagrama a bloques de un servomotor.	31
Figura 2.24 Una pieza dimensionada en el modo incremental (a) y en el modo del sistema absoluto (b).....	35
Figura 2.25 De derecha a izquierda: <i>Anet A8</i> , <i>Beethefirst</i> , <i>Replicator+</i>	37
Figura 2.26 Tarjeta de control de la impresora Ender 3.	38
Figura 2.27 Tipos principales de máquinas de gravado láser: A) Piezas de trabajo cilíndricas B) Mesa X-Y y C) Espejos galvanométricos.	40
Figura 3.1 A la izquierda el código en Matlab y a la derecha la nube de puntos generada.	41
Figura 3.2 Esfera realizada en Solid Works.....	42
Figura 3.3 Escáner 3D Sense 1 usado para la digitalización de las piezas.....	42
Figura 3.4 Nube de puntos sin procesar adquirida con el escáner 3D Sense 1.	43
Figura 3.5 Objeto escaneado: a) Aplicación de rotación a los ejes, b) Región de interés y c) Filtrado de la nube de puntos.	44
Figura 3.6 Resultado del sub-muestreo de la nube de puntos filtrada.	45
Figura 3.7 Modelo tridimensional generado por el algoritmo de pivoteo de pelota.	46
Figura 3.8. 1) Rebanada de una nube de puntos, 2) puntos de la rebanada proyectados sobre el plano Z y 3) unión de puntos en la frontera.	48
Figura 3.9 Rebanadas de la nube de puntos a lo largo del eje Y de la pieza 1... ..	49
Figura 3.10 Rebanado de la nube de puntos de una pirámide a lo largo del eje Z de la pieza 2.....	49
Figura 3.11 Panel de opciones de la extensión de <i>Endurance</i> en <i>Inskape</i> y previsualización de la imagen con las líneas de trayectoria del láser.	51
Figura 3.12 Líneas de trayectoria del láser generadas con la extensión <i>J-Tech Photonics</i>	51
Figura 3.13 Interfaz de usuario de la aplicación GRBL Laser.	52

Figura 3.14. Distancia de trabajo del módulo láser.	54
Figura 3.15. Primeras líneas del Código G original.	55
Figura 3.16. Ajuste de las primeras líneas del Código-G.	56
Figura 3.17. Ajustes al Código-G para mover, encender y apagar el módulo láser.	57
Figura 3.18. Líneas finales del Código-G original.	57
Figura 3.19. Archivos de Código G ajustado para cada rebanada de la pirámide. En el último archivo se une el código de todas las rebanadas.	58
Figura 4.1. Impresora Ender 3 Pro semi-ensamblada.....	60
Figura 4.2. Ensamble final de la impresora Ender 3 Pro.	60
Figura 4.3. Módulo Láser de 10 Watts fabricado por Endurance.	61
Figura 4.4. Cabezal del láser y tablilla o placa de control del láser.	62
Figura 4.5. Tablilla principal de la impresora Ender 3 Pro.....	62
Figura 4.6. Cabezales del extrusor y láser con las piezas de montaje intercambiable.	63
Figura 4.7. Procedimiento para determinar la distancia óptima de trabajo del cabezal láser.	64
Figura 4.8. Sistema de tallado de madera con tecnología láser.....	64
Figura 5.1. Dibujo grabado en madera tipo MDF con el Código G de prueba.	65
Figura 5.2 Código con modificaciones. Aun se muestran fallas señaladas en rojo	66
Figura 5.3. Primeras pruebas de tallado en madera.	66
Figura 5.4. Pirámide de 4 lados tallada en a) madera de pino, b) madera de nogal, c) MDF y d) triplay.....	67
Figura 5.5. Parámetros de configuración del programa <i>GRBL Laser</i>	68
Figura 5.6. Prueba de tallado con dirección “ <i>Diagonal</i> ”.	68
Figura 5.7. Prueba de tallado de la pirámide en (a) la pieza terminada y en (b) la pieza terminada separada del bloque de madera.	69
Figura 5.8. Pirámide tallada limpiada con lija. (a) Vista frontal y (b) vista lateral.	70
Figura 5.9 Pirámide tallada en madera de pino. (a) Vista frontal y (b) vista lateral.	70

LISTA DE TABLAS

Tabla 4.1. Especificaciones técnicas de la impresora Ender 3 Pro.	60
Tabla 4.2. Especificaciones técnicas del módulo láser de 10 Watts de Endurance.	61

Capítulo I. ANTECEDENTES

La creación de modelos tridimensionales a partir de piezas ya existentes es de utilidad para un amplio rango de aplicaciones; también es conocido como ingeniería inversa [1] y ha llamado la atención debido a las formas complejas que se pueden producir y al amplio campo de materiales al que se puede llegar. [2]

La ingeniería inversa ha tenido un gran auge en la creación o recreación de piezas. Se puede hacer un modelo sin necesidad de planos y tener excelente precisión sin la necesidad del vernier. Esta ingeniería inversa tiene una aplicación muy útil en la industria si se es capaz de obtener en base a esa información, algo tangible como lo es la reproducción de una pieza y para esto existen varios métodos, por ejemplo, la impresión en 3D que se basa en la representación digital y posteriormente mandar las instrucciones para que comience a crear la pieza desde cero. La desventaja que posee dicho método es que cada impresora está hecha para determinado tipo de material. Cada Impresora 3D es específica para cierto tipo de plástico o incluso se debe de considerar que cada plástico o cartucho recargable tiene propiedades diferentes y no son insumos intercambiables.

El láser ha sido considerado hasta hace poco solo como un método de corte, pero ya es posible obtener gradientes de profundidad mediante el control de las frecuencias de pulsación, la velocidad y la potencia del haz. Además, si se toma en cuenta la absorción del material y se ajustan los parámetros, con una misma herramienta se pueden manipular distintos materiales. En este trabajo se planteó la construcción del prototipo de un sistema con tecnología láser para el tallado secuencial en capas de un tipo específico de madera sin la necesidad de emplear diferentes herramientas, como lo hace una fresadora CNC, para realizar la recreación de piezas en tres dimensiones. Para lograr esto, primero se modela tridimensionalmente el objeto que se desea tallar en madera. El proceso de modelado tridimensional requirió de hardware de captura de datos, por lo que se evaluaron diferentes opciones, como el escáner de profundidad, los patrones de luz estructurada, la reconstrucción fotogramétrica o la generación del modelo por software. Una vez generado el modelo tridimensional del objeto fue necesario procesarlos digitalmente para generar el código que pudiera ser procesada por el

prototipo de tallado de madera por medio de rayos láser. La Figura 1.1 presenta un diagrama donde se muestran gráficamente las etapas del proyecto desarrollado.

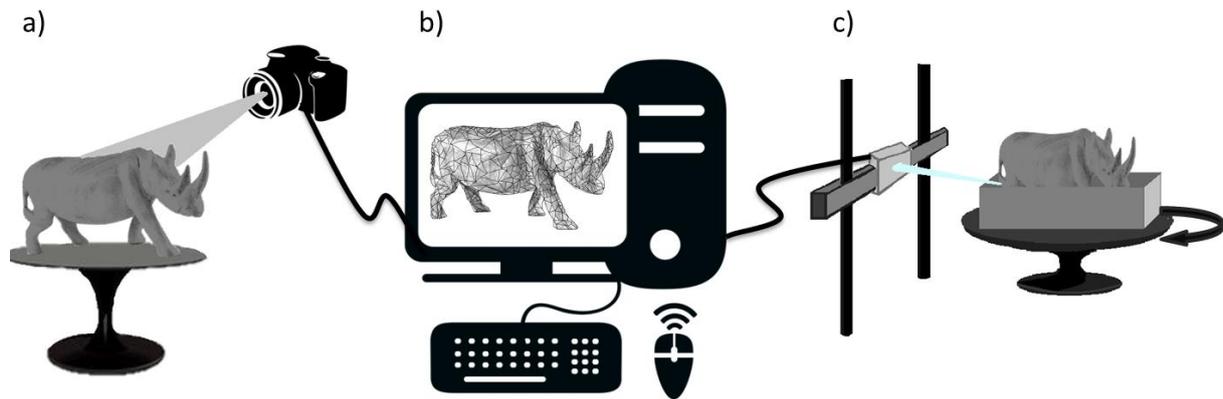


Figura I.1 Etapas del sistema desarrollado a) Adquisición datos tridimensionales del objeto. b) Procesamiento digital de los datos tridimensionales. c) Tallado de la figura en un bloque de madera por medio de rayo láser.

De acuerdo a lo anterior se planteó la hipótesis de que mediante la técnica de fotogrametría se puede obtener la mayor cantidad de detalles en el modelado tridimensional de una figura de geometría complicada de tamaño medio (20x20x20cm) y así generar los archivos de datos en el formato adecuado para ser procesados por el dispositivo de tallado de madera con rayo láser. De esta forma se obtiene una réplica en madera de la figura escaneada de igual tamaño y con buen nivel de detalle. El nivel de detalle final depende directamente de dos factores: el primero es la reconstrucción tridimensional y el segundo es el control correcto del láser para poder dar un buen acabado final a la pieza.

En este proyecto se realizó un proceso completo de ingeniería inversa desde el escaneo hasta la elaboración de la pieza final. Las pruebas se realizaron con madera debido a la facilidad de conseguir el material y a que la energía que el láser requiere es menor para fundir, quemar o vaporizar el material que se desea tallar, erosionar o grabar.

En la actualidad el láser es mayormente usado para hacer cortes o grabado en diferentes materiales y apenas se comienza a experimentar haciendo tallados en materiales en cerámicos y madera, pero no es considerado como grabado tridimensional ya que es más parecido a una figura en relieve sobre un solo plano.

Con el prototipo desarrollado se logra un avance en ese sentido, ya que se pudo hacer una figura volumétrica con relieve los 360°.

El tallado de madera con rayo láser podría ser de gran valor artístico ya que se puede reproducir una gran cantidad de piezas elaboradas por artesanos expertos de su época y con trabajos invaluable por su valor histórico que debido a las características de la madera su deterioro es rápido comparado con otros materiales, los cuidados que se les den, pueden retardar, pero no detener su desgaste. También existen artesanos con habilidades magníficas, pero pueden durar años antes de terminar sus obras, lo que hace que la pieza elaborada tenga un gran valor pero que pocos pueden adquirir y, por lo tanto, la venta de una pieza puede no ser tan significativa considerando el tiempo del proceso. En este caso la reproducción de piezas podría convertirse en una opción económica viable para la comercialización de sus obras.

Debido a la belleza y propiedades mecánicas propias de la madera es un material muy buscado. Además, sus características naturales no pueden ser obtenidas con la fabricación aditiva (conocida como impresión 3D) debido a que sería necesario que se triturara y se usara otro polímero para contenerla y así ser impresa, lo cual no da resultados equiparables a la madera natural.

Mediante el análisis correcto de las piezas a elaborar, se obtuvo un modelo de dimensiones casi idénticas al original y se espera que el campo de la ingeniería inversa vea como factible el dimensionamiento final o total en la creación de piezas con una mejora de tolerancias, esperado que debido a la naturaleza más precisa del corte con láser y al no-contacto con el material se eliminen las distorsiones comunes de los sistemas por contacto. En la elaboración de piezas, un correcto análisis dimensional puede marcar la diferencia para evitar problemas de desgaste o generación de esfuerzos ya sea en el mecanismo e incluso en la estructura interna (esfuerzos internos).

Nuestros objetivos fueron (1) generar modelos tridimensionales de objetos de dimensión reducida, (2) desarrollar el software para el procesamiento del modelo tridimensional del objeto y generación de Código G y (3) desarrollar el sistema de tallado de madera con tecnología láser. Para dar como resultado el prototipo de un

sistema de tallado en madera con tecnología láser de modelos tridimensionales de objetos de dimensiones reducidas.

El presente documento de tesis está organizado de la siguiente manera: En el Capítulo 2 se presenta el Marco Teórico que contiene los conceptos en los que se basa el desarrollo del proyecto y un análisis del estado del arte. El Capítulo 3 presenta como fue generado el archivo del modelo tridimensional y convertido en Código G, en el Capítulo 4 se detalla cómo fue construido el sistema de tallado con tecnología láser, en el Capítulo 5 se muestran los resultados obtenidos del tallado en madera de diferentes objetos y, por último, en el Capítulo 6 se mencionan las conclusiones del desarrollo del proyecto de tesis.

Capítulo II. MARCO TEÓRICO

1.1 Adquisición de Datos Tridimensionales

En general, las herramientas existentes de recolección de datos pueden agruparse en dos categorías principales: Técnicas no fotográficas (también conocidas como de contacto) y las fotográficas (o sin contacto). Las técnicas no fotográficas se basan en estudios convencionales para producir planos, elevaciones y detalles arquitectónicos, también son usadas en la industria, aunque con el inconveniente de que pueden causar daño o modificaciones a la superficie que está siendo escaneada.

1.1.1 Adquisición por Contacto

Las herramientas usadas en las técnicas de contacto van desde el topógrafo de cinta de acero o fibra de vidrio hasta el teodolito óptico o cámara de "estación total". Este tipo de adquisición permite obtener decenas o incluso cientos de coordenadas para objetos pequeños, pero es lenta y consume mucho tiempo cuando se requiere modelar objetos grandes o con muchos detalles lo que tomaría incluso días para procesar la información, agregando que el nivel de sensibilidad depende de la experiencia del inspector, ya que a veces las mediciones se realizan de una forma semiautomática, y del tipo del equipo utilizado.

En la Figura 2.1 se muestra un escáner de contacto. Se puede apreciar en la Figura 2.1(a) la colocación de una nueva punta en el aparato debido a que es necesario que estén calibradas, en la Figura 2.1(b) se ve que, aunque el escáner se puede mover, las piezas a dimensionar requieren un lugar espacioso. En la Figura 2.1(c) se observa como un operador es el encargado de tomar las mediciones en la mayor cantidad de regiones posibles.

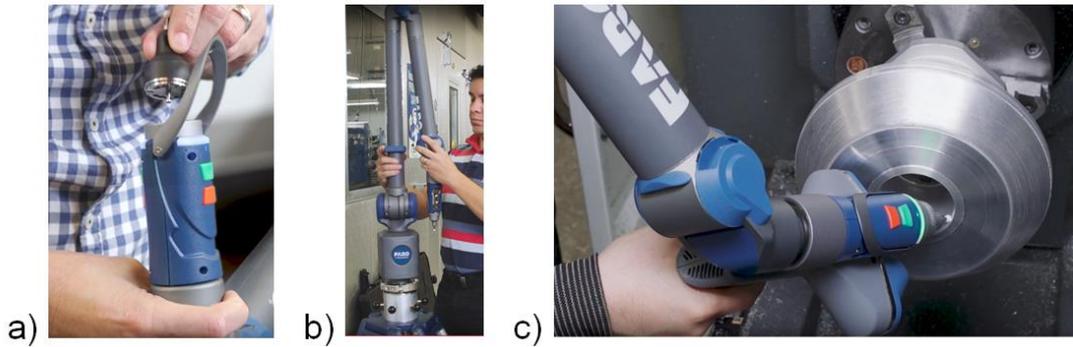


Figura II.1 Escáner de contacto “FaroArm Quantum”.

La compañía Kreon ofrece el brazo mostrado en la Figura II.2, que complementa la medición precisa realizada con el método de contacto, con un escaneo digital y además cuenta con varios tamaños.



Figura II.2 Escáneres de contacto Kreon de varios tamaños.

1.1.2 Técnicas Fotográficas

Dentro de las técnicas fotográficas, podemos encontrar los digitalizadores de proyecciones de franjas de luz, prototipos de sistemas automatizados de fotogrametría de video estéreo y una serie de escáneres basados en láser; que van desde la triangulación hasta el desplazamiento de fase y el radar láser o LIDAR. La Figura 2.3 muestra gráficamente el funcionamiento de un sistema LIDAR aéreo. Este sistema permite conocer la distancia desde el emisor láser al objeto utilizando un haz láser pulsado. La distancia se determina midiendo el tiempo de retraso entre la emisión del pulso y su detección a través de la señal reflejada. La aeronave está equipada con sistema GPS que recibe información de posición, velocidad y rumbo. El haz puede ser de luz ultravioleta, visible o infrarrojo cercano.

Todo sistema tiene sus propias limitaciones, pero los anteriores son mucho más rápidos que los sistemas de contacto. Este tipo de dispositivos son utilizados en la preparación de representaciones 3D complejas utilizadas en realidad virtual [3] [4]. La Figura 2.4 muestra un sistema de visión estéreo de reciente modelo. Como se puede apreciar, cuenta con un sistema de tres cámaras, conocido como visión trinocular, esto debido a que el método de visión binocular tiene dificultad para encontrar una correspondencia de rasgos entre dos imágenes para la reconstrucción de formas tridimensionales.

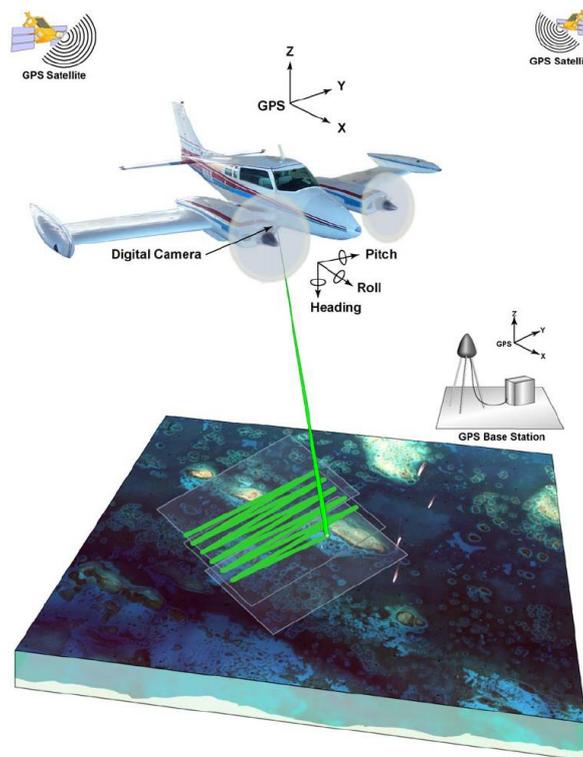


Figura II.3 Adquisición de imágenes mediante un sistema LIDAR aéreo.



Figura II.4 Sistema de Visión en estéreo con 3 cámaras.

Otro de los métodos de no-contacto es el de luz estructurada, que hace uso de una fuente de luz para proyectar un patrón sobre la escena a escanear y una cámara que lo captará. El patrón proyectado tiene una serie de variaciones de

acuerdo a la cantidad de líneas de luz o los patrones formados por la luz transmitida, por ejemplo, los códigos binarios, en grises y corrimiento en tres fases, entre otros [5]. La Figura 2.5 muestra el *Scan in a Box-FX* que es un sistema de adquisición por medio de patrones de luz estructurada, al centro se puede observar el sistema que proyecta los patrones de líneas y a sus lados las cámaras que van adquiriendo la imagen con los cambios provocados por el objeto a escanear. Este escáner cuenta con una precisión de 0.04mm y su distancia de trabajo es de 190mm hasta 1355mm.



Figura II.5 Sistema de adquisición por medio de patrones de luz estructurada.

Al igual que la luz estructurada, el método de medición de tiempo de vuelo o “*time-of-flight*” no hace contacto con la pieza. Este mide el retardo de fase de la luz infrarroja reflejada. Al medir el desplazamiento de fase entre las ondas infrarrojas (IR) radiadas y reflejadas, podemos calcular la distancia al objeto y mientras mayor sea el tiempo de integración permite una mayor precisión de la medición de profundidad [6]. La Figura 2.6 muestra la cámara *Basler ToF*, es un ejemplo de un dispositivo de adquisición por medición de tiempo de vuelo de rayos IR.

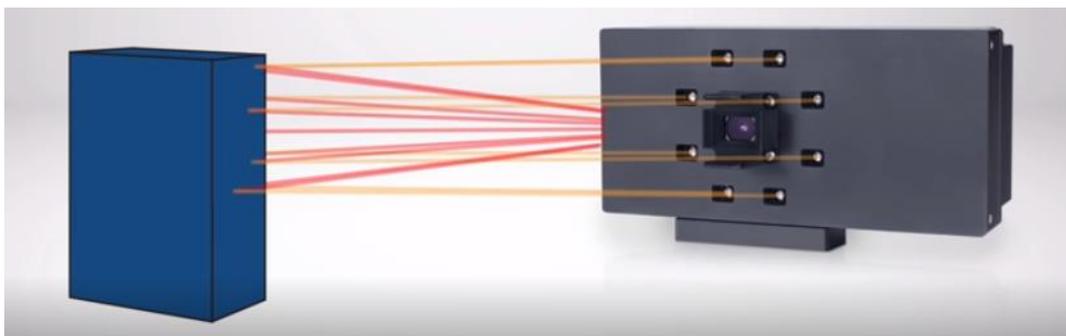


Figura II.6 Sistema de adquisición por medio de la medición de tiempo de vuelo.

Los diferentes métodos conducen a una precisión diferente en la medición de profundidades. Esto lleva a la conclusión de que ningún escáner específico puede satisfacer todas las demandas en diferentes proyectos de digitalización [7] por lo que es necesario considerar los resultados deseados y determinar prioridades.

1.1.3 Representación de Datos Tridimensionales

El paso siguiente a la captura de datos es la representación de la información. Las imágenes de rango y las nubes de puntos son las estructuras de datos más utilizadas para almacenar y procesar la información tridimensional de los objetos escaneados. Luego se requiere la generación de mallas poligonales a partir de las nubes de puntos calculadas sobre regiones del objeto, de modo que cada región pueda ser aproximada por una superficie. Este proceso se muestra en la Figura 2.7.

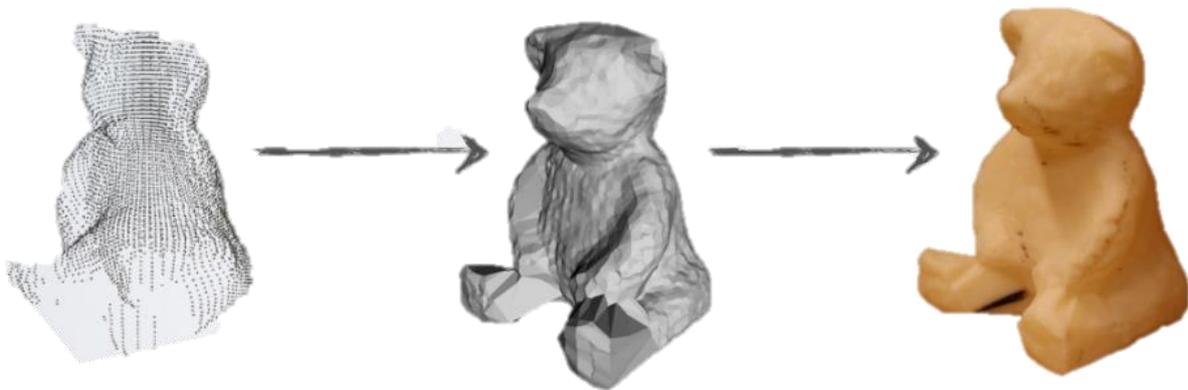


Figura II.7 Nube de puntos del objeto, mallado de la superficie y el objeto generado dándole color a las superficies.

La primera representación de un modelo tridimensional es una nube de puntos, que son un conjunto de puntos individuales usualmente con un acomodo sin relación con la posición y el color definidos. Una nube de puntos puede ofrecer visualizaciones de la estructura tridimensional del objeto escaneado de forma inmediata. Las nubes de puntos contienen un gran volumen de información que no se ha interpretado y de la que a veces resulta difícil extraer datos específicos, lo que sí se puede hacer es realizar un proceso de filtrado, eliminando los puntos muy alejados del objeto o puntos muy cercanos entre sí, para después seguir con el proceso de mallado [8].

1.2 Procesamiento Digital de Datos Tridimensionales

En el procesamiento digital de datos tridimensionales están involucrados un gran número de elementos, tanto de *hardware* como de *software*. Primero se debe de adquirir la imagen mediante un sensor, luego los datos capturados son transferidos a una computadora donde un *software* procesará la información para así poder observar la imagen digitalizada [9]. Las nubes de puntos adquiridas por los escáneres tridimensionales requieren ser procesadas digitalmente para obtener un modelo tridimensional renderizado que puede ser utilizado para generar modelos de CAD en 3D para la manufactura de partes, metrología e inspección de calidad, además de un sin número de aplicaciones de visualización, animación, realidad virtual, medicina, etc.

1.2.1 Filtrado de nubes de puntos

En la adquisición de objetos tridimensionales, una forma de guardar los datos son las nubes de puntos. Dependiendo del sistema de adquisición y de cómo interprete el entorno, muchas veces se pueden capturar puntos que no forman parte del objeto de interés, ocasionando que el objeto tenga deformaciones. Estos puntos fuera del objeto son considerados ruido.

La acción del filtrado es muy importante para la conservación de las características geométricas, por lo que el estudio de diferentes métodos de filtrado ha resultado crucial para lograr una reconstrucción tridimensional más exacta. Aunque es posible usar diferentes tipos de filtrados subsecuentes o individuales, también hay que considerar el tiempo de ejecución y requerimientos de memoria.

Si durante la adquisición no se consideraron las áreas específicas a escanear, la fase del filtrado se alarga. Por eso, mientras se está adquiriendo la figura, hay que procurar que se digitalice lo más cercano a la geometría deseada, para evitar el ruido creado por figuras alrededor, y en caso de ser figuras muy grandes, que se escaneen por partes, organizando los datos de antemano. Esto reduce la cantidad de trabajo en la fase de procesamiento.

Para reducir el ruido presente en las nubes de puntos se pueden usar algoritmos de suavizado de datos. Estos métodos se dividen en dos grupos principales:

- Métodos utilizados con algoritmos tradicionales para eliminar ruido en imágenes 2D adaptados a imágenes de profundidad. La Figura II.8 muestra una imagen de profundidad visualizada como una imagen en niveles de gris donde los puntos más claros son más cercanos al sensor.

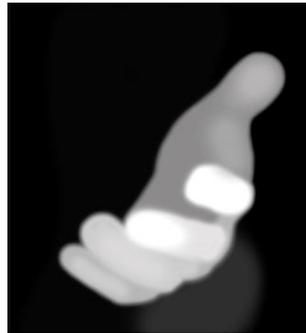


Figura II.8 Imagen de profundidad visualizada en niveles de gris.

- Métodos que trabajan con nubes de puntos tridimensionales. La Figura 2.9 muestra una nube de puntos tridimensionales (a) con ruido y (b) filtrada. Estos pueden involucrar filtros que llevan a cabo la eliminación de ruido en los descriptores calculados de cada punto o en las estimaciones de mínimos cuadrados para superficies alisadas.



Figura II.9 Nube de puntos tridimensionales (a) con ruido y (b) filtrada.

1.2.2 Reconstrucción de Superficies

La reconstrucción de superficies a partir de datos dispersos, como lo son las nubes de puntos, es un problema que es resuelto por medio de la geometría computacional mediante distintos algoritmos para convertir los puntos o datos dispersos en un modelo de malla al que se le puede asignar color, texturas, sombras y colocarlo en una escena tridimensional. Según Hilton et al. [10], el objetivo principal es estimar una superficie múltiple S' que se aproxima a una superficie del objeto desconocido S utilizando una muestra de puntos $x = (x, y, z)$ en el espacio euclidiano en 3D, $X = \{X_0, \dots, X_{N-1}\}$, combinado con el conocimiento sobre la resolución de muestreo Δx , el error de medición c y la confianza en la medición $p(S'/xi)$. A continuación se describen algunos de los algoritmos que se utilizan para la creación de estas superficies.

1.2.2.1 Mallas Triangulares

Los diagramas de Voronoi registran o documentan información acerca de la proximidad de un conjunto de puntos. Un diagrama de Voronoi se puede definir considerando $P = \{p_1, p_2, \dots, p_n\}$ como un conjunto de puntos en un plano Euclidiano bidimensional. A cada p_i se le denomina "sitio". Se divide el plano asignando cada punto en el plano a su sitio más cercano. Todos esos puntos asignados a p_i forman la *región de Voronoi* $V(p_i)$, que consiste en todos los puntos más cercanos a p_i que a cualquier otro punto p_j .

$$V(p_i) = \{x: |p_i - x| \leq |p_j - x| \forall j \neq i\} \quad (2.1)$$

Ha de notarse que se definen como un conjunto cerrado. Algunos puntos no tienen un sitio único más cercano o vecino más cercano. El conjunto de todos los puntos que tienen más de un vecino más cercano forman el diagrama Voronoi para el conjunto de sitios. En la Figura 2.10 se muestra un ejemplo de un diagrama Voronoi, se observa que las regiones de Voronoi se forman con las intersecciones de las líneas bisectores entre cada par de vecinos más cercanos.

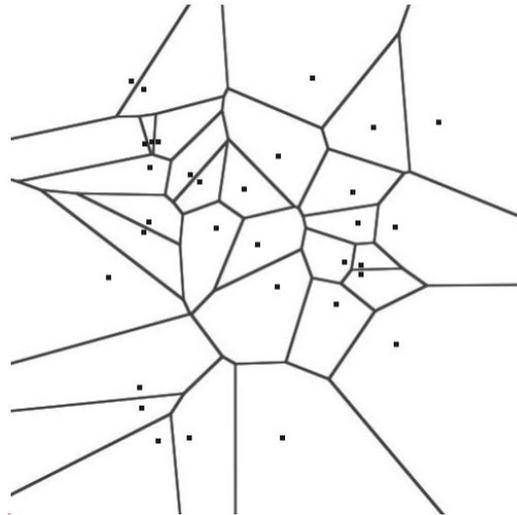


Figura II.10 Diagrama de Voronoi.

En 1934 Delaunay demostró que cuando el gráfico se dibuja con líneas rectas, produce una triangulación planar de los sitios P de Voronoi (no habiendo cuatro o más sitios co-circulares) llamada la Triangulación de Delaunay $\mathcal{D}(P)$. En la Figura 2.11 se muestra la transformación del diagrama de Voronoi con regiones poligonales a su respectiva Triangulación de Delaunay.

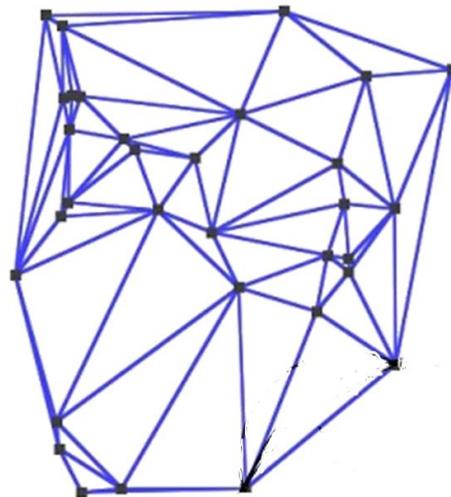


Figura II.11 Triangulación de Delaunay del diagrama de Voronoi de la Figura 2.10.

Sabiendo que la Triangulación de Delaunay \mathcal{D} y el diagrama de Voronoi \mathcal{V} son estructuras equivalentes y que contienen la misma información, pero representada en una forma diferente, se presentan las condiciones que requieren cada una. Es importante tener una comprensión profunda de las relaciones entre una Triangulación de Delaunay y su correspondiente diagrama de Voronoi. A

continuación se encuentran varias propiedades de Delaunay (\mathcal{D}) y después una lista mayor de propiedades de Voronoi (\mathcal{V}). Dado un conjunto de sitios P fijo.

- 1) $\mathcal{D}(P)$ es una línea recta dual a la de $\mathcal{V}(P)$. Esto es por definición.
- 2) $\mathcal{D}(P)$ es una triangulación con la condición de que no existan cuatro puntos co-circulares. Cada cara es un triángulo. Esto es el Teorema de Delaunay. Las caras de $\mathcal{D}(P)$ también son llamadas Triángulos de Delaunay.
- 3) Cada cara (triángulo) de $\mathcal{D}(P)$ corresponde a un vértice de $\mathcal{V}(P)$.
- 4) Cada borde de $\mathcal{D}(P)$ corresponde a un borde de $\mathcal{V}(P)$.
- 5) Cada nodo de $\mathcal{D}(P)$ corresponde a una región de $\mathcal{V}(P)$.
- 6) La frontera de $\mathcal{D}(P)$ es el casco convexo de los sitios.
- 7) El interior de cada (triángulo) cara de $\mathcal{D}(P)$ no contiene ningún sitio.

Y dentro de las propiedades de los diagramas de Voronoi:

- 1) Cada región de Voronoi $V(p_i)$ es convexa.
- 2) $V(p_i)$ no está delimitada si y sólo si p_i está en el casco convexo del conjunto de puntos.
- 3) Si v es un vértice de Voronoi en la intersección de $V(p_1)$, $V(p_2)$ y $V(p_3)$, entonces v es el centro del círculo $C(v)$ determinada por p_1, p_2 y p_3 . Esta afirmación se generaliza a los vértices de Voronoi de cualquier grado.
- 4) $C(v)$ es la circunferencia circunscrita (circunferencia que cubre todos los puntos que estamos analizando) del triángulo Delaunay correspondiente a v .
- 5) El interior de $C(v)$ no contiene otros sitios. (vértices)
- 6) Si p_j es el vecino más cercano a p_i , entonces (p_i, p_j) es un borde de $\mathcal{D}(P)$.
- 7) Si hay algún círculo a través de p_i y p_j que no contiene otros sitios, entonces (p_i, p_j) es un borde de $\mathcal{D}(P)$. Lo contrario se mantiene: Para cada borde de Delunay existe un círculo vacío.

En la Figura 2.12 se muestra una superposición del diagrama de Voronoi de la Figura 2.10 y de la Triangulación Delaunay de la Figura 2.11. Aquí podemos observar el cumplimiento de las propiedades de ambos diagramas; en azul se muestra un ejemplo de triangulación de Delaunay y en gris un diagrama de Voronoi.

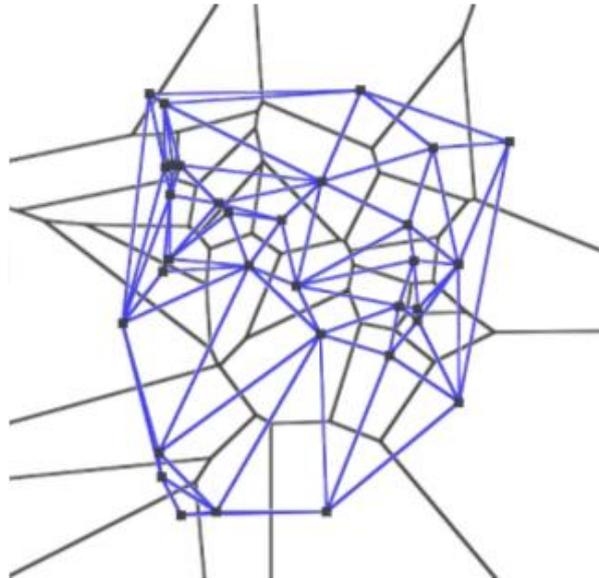


Figura II.12 Superposición de la Figura 2.10 y la Figura 2.11.

En la Figura 2.13(b) se observa cómo se cumple la condición de Delaunay, es decir, que dentro del círculo se encuentran solo 3 vértices o un solo triángulo; al contrario de la Figura 2.13(a) donde el polígono tiene más de tres vértices dentro de la circunferencia creada.

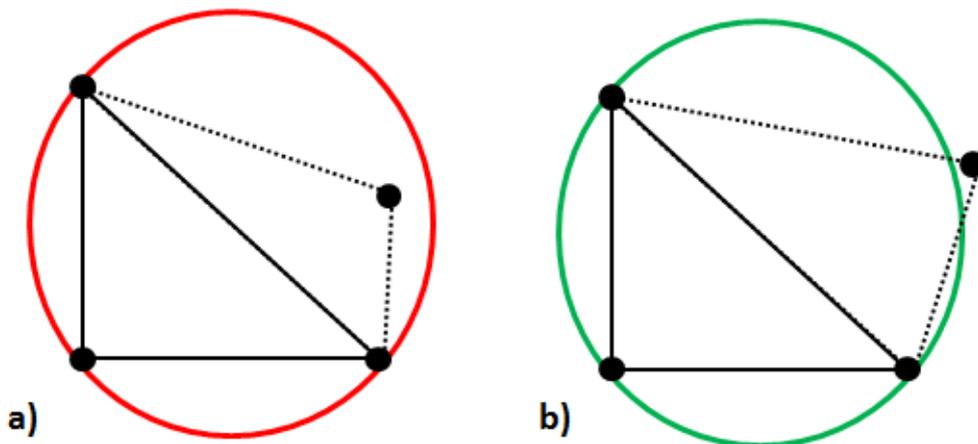


Figura II.13 En a) no se cumplen las condiciones de Delaunay y en b) si se cumplen.

Una posible solución para este problema es girar el vértice interior y tomar ese como nueva referencia del triángulo. Esto permite que, a la hora de crear la circunferencia de cualquiera de los dos nuevos triángulos, no encierre otro vértice, como se demuestra en la Figura 2.14.

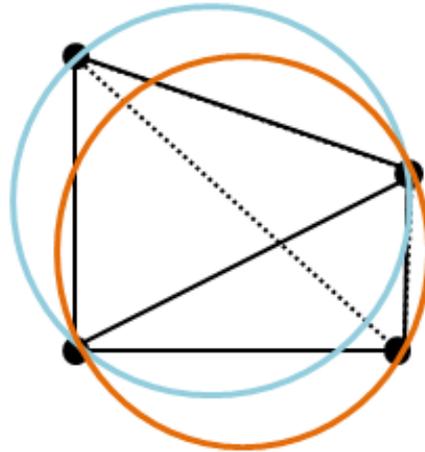


Figura II.14 Cambio del vértice interior buscando que las condiciones de Delaunay se cumplan.

1.2.2.2 Mallas poligonales

Gran parte de la geometría computacional realiza sus cálculos usando objetos geométricos conocidos como polígonos. Estos permiten una representación conveniente para muchos objetos del mundo real, además de que son fácilmente manipulables computacionalmente hablando. En ocasiones los polígonos pueden ser objetos bastante complicados, por lo que surge la necesidad de descomponerlos en piezas más simples.

Un polígono es la región de un plano que es delimitada por un número finito de segmentos de línea formando una curva cerrada simple. Se puede definir considerando $v_0, v_1, v_2, \dots, v_{n-1}$ como n puntos en un plano e implica tener un orden cíclico de los puntos con v_0 después de v_{n-1} ya que $(n - 1) + 1 \equiv n \equiv 0$. Siendo $e_0 = v_0v_1, e_1 = v_1v_2, \dots, e_i = v_iv_{i+1}$ los n segmentos que conectan cada par consecutivo de puntos. Entonces esos segmentos limitan un polígono si y solo si:

- La intersección de cada par de segmentos contiguos en el orden cíclico es el único punto compartido entre ellos: $e_i \cap e_{i+1} = v_{i+1}$ para todo $i = 0, \dots, n - 1$.
- Los segmentos no adyacentes no se intersectan entre sí: $e_i \cap e_j = \emptyset$ para todo $j \neq i + 1$.

Los puntos v_i son considerados los vértices del polígono y los segmentos e_i se denominan aristas. La Figura 2.15 presenta un ejemplo de una malla poligonal.

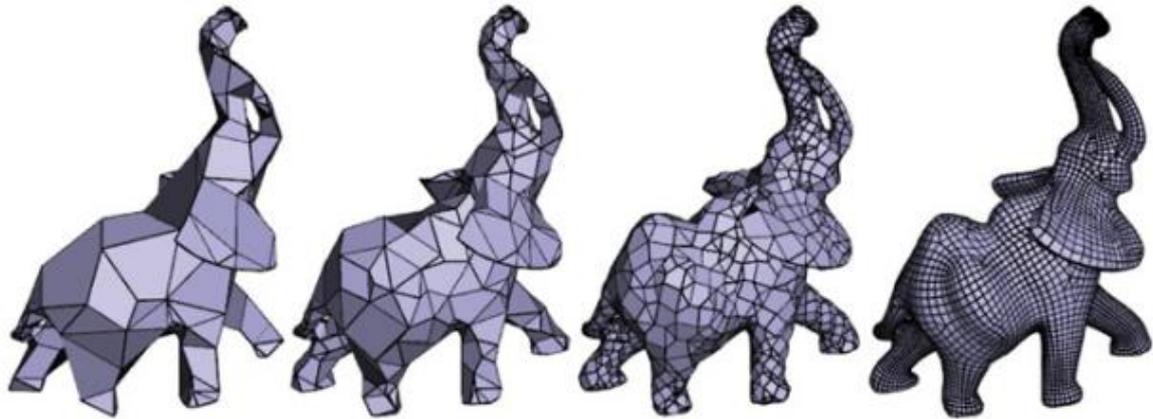


Figura II.15 Malla poligonal.

1.2.2.3 NURBS

Proviene del inglés *Non-Uniform Rational B-splines* podría traducirse “curvas tipo B, racionales no-uniformes.

No-uniforme se refiere a la parametrización de la curva. Las curvas no uniformes permiten, entre otras cosas, la presencia de “nudos” múltiples, que son necesarios para representar las curvas de Bezier.

Se considera Racional por la representación matemática subyacente. Esta propiedad permite que NURBS haga representaciones cónicas exactas (como círculos de cerdas parabólicas y elipses) además de curvas de forma libre.

Los “*splines*” son tipos de curvas, desarrolladas originalmente para la construcción de barcos antes del modelado por computadora. Los arquitectos navales necesitaban una forma de dibujar una curva suave a través de un conjunto de puntos. Su solución fue colocar pesos de metal también llamados “*nudos*” en los puntos de control y doblar una delgada viga de metal o madera (*spline*) a través de los pesos. La Figura 2.16 muestra el proceso de diseño de un navío usando *splines*.

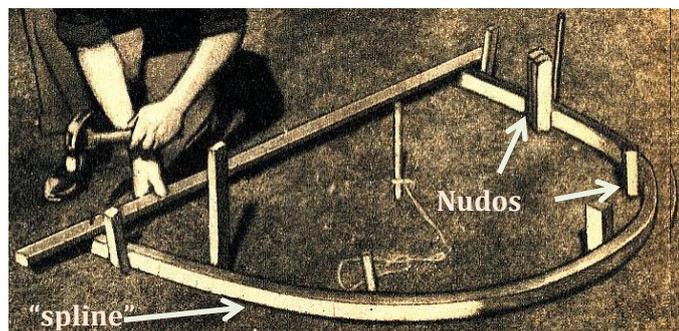


Figura II.16 Diseño de barcos usando splines.

La física de la flexión de la curvatura indica que la influencia de cada peso es mayor en el punto de contacto y disminuye suavemente a lo largo de la *spline* o curva. Para tener más control sobre cierta región de la curva, el dibujante simplemente agrega más pesos. Este esquema rudimentario, tiene un problema obvio al querer intercambiar información para replicar. Es necesaria una forma matemática para describir la forma de la curva. [11]

Las *b-splines* son curvas polinomiales inteligentes que tienen una representación paramétrica. La definición matemática de una curva NURBS es:

$$P(u) = \frac{\sum_{i=0}^n B_{i,k}(u)W_iV_i}{\sum_{i=0}^n B_{i,k}(u)W_i} \quad (2.2)$$

Donde V_i son los puntos de control; W_i son los coeficientes de esos puntos; $B_{i,k}(u)$ es la función base de "B-Spline". Esta misma puede definirse como

$$B_{i,0}(u) = \begin{cases} 1, & u_i \leq u \leq u_{i+1} \\ 0, & u < u_i \cup u > u_{i+1} \end{cases}$$

$$B_{i,k}(u) = \frac{u-u_i}{u_{i+k}-u_i} B_{i,k-1}(u) + \frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} B_{i+1,k-1}(u), \quad k \geq 1 \quad (2.3)$$

$$\frac{0}{0} = 0$$

Si observamos, el $0/0=0$ significa que si el numerador y el denominador son cero al mismo tiempo, el resultado es cero; u es el parámetro.

$$u \in \mathbf{U} = \left[\underbrace{a, a, \dots, a}_{k+1}, u_{k+1}, \dots, u_n, \underbrace{b, b, \dots, b}_{k+1} \right]$$

Y para representar una superficie NURBS:

$$P(u, \omega) = \frac{\sum_{i=0}^n \sum_{j=0}^m B_{i,k}(u) B_{j,l}(\omega) W_{i,j} V_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m B_{i,k}(u) B_{j,l}(\omega) W_{i,j}} \quad (2.5)$$

$$u \in \mathbf{U} = \left[\underbrace{a, a, \dots, a}_{k+1}, u_{k+1}, \dots, u_n, \underbrace{b, b, \dots, b}_{k+1} \right]$$

$$u \in \mathbf{V} = \left[\underbrace{c, c, \dots, c}_{l+1}, u_{k+1}, \dots, u_m, \underbrace{d, d, \dots, d}_{l+1} \right]$$

Las superficies NURBS están definidas por una forma bi-paramétrica por lo que los puntos de superficie pueden construir una superficie que se puede calcular con los dos parámetros en términos de algunos algoritmos. La superficie construida de esta manera tiene mayor precisión que el método de rejilla. Además, cuando se calcula una superficie puntiaguda, se puede obtener el vector normal de la superficie en la punta [12].

1.2.2.4 Refinado de mallas tridimensionales

El uso de mallas para la representación tridimensional de objetos permite omitir puntos individuales (proveniente usualmente de una nube de puntos) que permite un perfeccionamiento o detallado de la superficie para obtener una superficie continua. Los modelos que realizan el mallado pueden y deben ser reestructurados en busca de un consumo de memoria menor mediante un menor uso de puntos, pero sin sacrificar precisión en dimensión con respecto a la forma original del objeto. La Figura 2.17 muestra cómo va de izquierda a derecha una triangulación más fina, con una mayor definición.

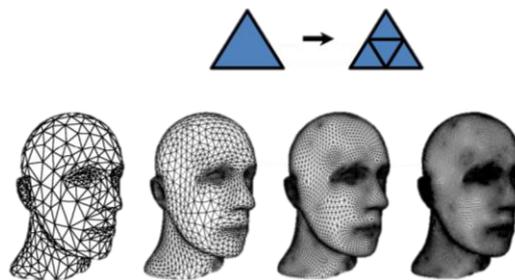


Figura II.17 Refinado de mallado tridimensional.

La segmentación por mallas tridimensionales es una técnica utilizada para el análisis de formas tridimensionales, se ha desarrollado usando diferentes metodologías tomando en cuenta aspectos claves o necesidades específicas, como la segmentación por superficies o por partes de los objetos. La primera maneja al objeto como una superficie y emplea sólo atributos relacionados con la superficie, mientras que la segunda emplea atributos volumétricos y superficiales para lograr su objetivo. Panaguitis *et al.* [13] han re-categorizado el tipo de malla de acuerdo a objetivos, tipo de criterios geométricos, grado de aprendizaje, participación del usuario, número de objetos utilizados como fuentes de información, entre otras. Su

finalidad es entender que dependiendo del objeto se le dará prioridad a una u otra característica.

Yiqun Wang et al. [14] menciona que al hacer un re-mallado, se debe cuidar la *fidelidad* que consiste en no cambiar las dimensiones respecto de la pieza original. La *simplicidad* señala que debe de haber un número de vértices óptimos para evitar disminuir la eficiencia computacional. Por último, se habla de la *calidad del elemento*, en donde se busca que el triángulo sea lo más equilátero posible. Mientras más fina sea la malla, mayor fidelidad y costo computacional.

1.3 Tecnología Láser

La palabra láser, proviene de un acrónimo que significa luz amplificada por emisión estimulada de radiación, por lo que es un emisor de luz, pero cuenta con propiedades debido al mecanismo físico que produce la emisión conocido como “emisión estimulada” para lo cual es necesario entender la interacción de los átomos con los fotones. Lo anterior le confiere al láser propiedades como: alta potencia, direccionalidad (emisión en forma de rayos), frecuencia de emisión bien definida (color de la luz), capacidad de emitirse en pulsos de muy corta duración y una propiedad llamada coherencia que significa que las ondas electromagnéticas que forma el haz de luz marchan "al paso". La Figura 2.18 muestra de manera gráfica las diferencias entre las propiedades del láser y la luz natural.

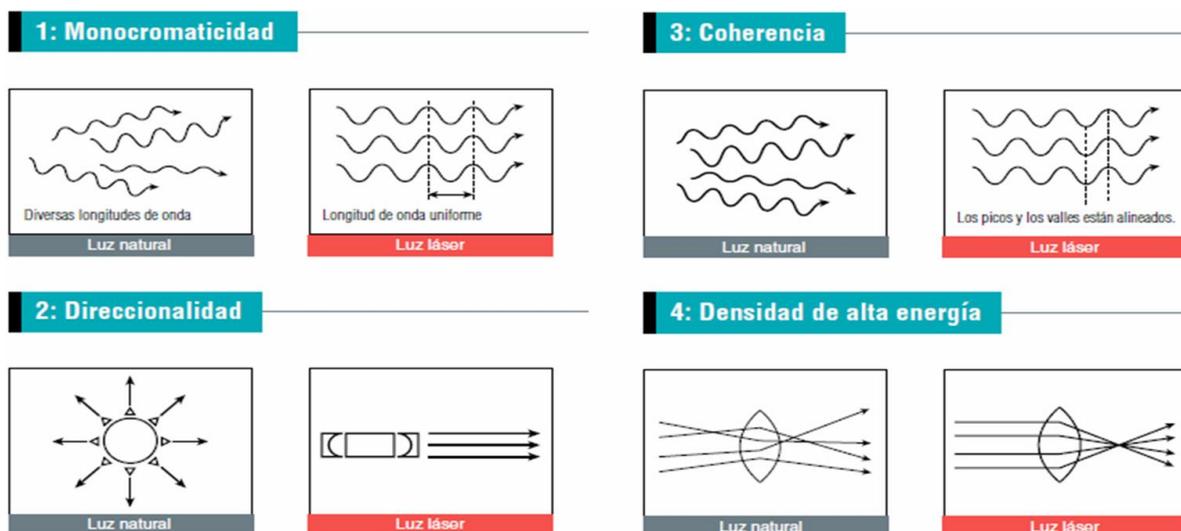


Figura II.18 Diferencias entre las propiedades del láser y la luz natural.

Se dice que el láser es una herramienta muy flexible. Produce un rayo con propiedades únicas y puede ser controlada con precisión, enfocada en un punto muy pequeño, proporcionando una energía tan intensa que atraviesa materiales o hace que el haz sea difuso y trata superficies con el patrón de calentamiento proporcionado.

1.3.1 Tipos de Láser

Existen diferentes tipos de clasificación de láser, aquí tomaremos en cuenta el medio activo para clasificarlos:

1.3.1.1 Gas

En este tipo de láser una corriente eléctrica se descarga a través de un gas para producir el haz de luz. Los láseres de gas son de diferentes tipos: helio (He), neón (Ne), ion de argón, dióxido de carbono (CO₂), monóxido de carbono (CO), excímeros, nitrógeno, hidrógeno y otros.

El tipo de gas utilizado para construir el medio láser puede determinar la longitud de onda o la eficiencia del láser. La cantidad de propiedades que posee el láser de gas le ha dado una gran popularidad en la industria debido a que puede ser excitado directamente con corriente eléctrica, es homogéneo, proporciona una alta flexibilidad en el diseño del resonador que permite una fácil escalabilidad y es relativamente barato.

1.3.1.2 Líquido

Se compone de un colorante orgánico mezclado con un disolvente. Estos láseres generan un haz de luz a partir de los estados de energía excitados de tintes orgánicos disueltos en solventes líquidos. Produce un haz de luz láser en el ultravioleta (UV) cercano a la región del infrarrojo cercano (NIR) del espectro electromagnético. Combinando los colorantes, las longitudes de ondas producidas pueden llegar a cubrir el espectro de luz visible. Su vida de uso puede durar desde horas hasta meses dependiendo del colorante y del medio de excitación. Este tipo de medio tiene la ventaja de que puede ser preparado fácilmente, debido a que los

sólidos requieren de un alto grado de homogeneidad y que contienen una mayor densidad de átomos a diferencia del medio gaseoso.

1.3.1.3 Sólidos aislantes

Este tipo de medio activo no incluye a los semiconductores. Comprende a materiales dopados con iones que se introducen como impurezas en el material huésped, que idealmente son materiales cristalinos estables, duros, ópticamente isotrópicos y con alta fuerza tensil. El proceso de agregar impurezas a la sustancia se llama dopaje. Los elementos de tierras raras como el cerio (Ce), el erbio (Eu) y el terbio (Tb) son los más comunes. El medio activo sólido permite construir láseres relativamente pequeños y sin el mantenimiento que requiere el flujo del gas, pero el calentamiento limita la potencia que se puede generar y la calidad del haz es relativamente pobre a altas potencias.

1.3.1.4 Semiconductores

Ocupan amplias bandas de niveles de energía en donde cada banda comprende una serie de niveles estrechamente espaciados [15]. La longitud de onda puede cambiar al variar la temperatura de funcionamiento del diodo. Por ello se coloca al láser en contacto térmico con un disipador y un enfriador controlado que estabiliza la temperatura de funcionamiento para alcanzar una emisión estable a determinada longitud de onda. Los láseres semiconductores son dispositivos extremadamente sensibles al ruido eléctrico y se pueden dañar fácilmente, aunque están dentro de los más pequeños y eficientes que existen en la actualidad. [16]

1.3.2 Maquinado con Rayos Láser

El maquinado por medio de rayos láser es una técnica nueva, debido a sus características de corte y a que es aplicable a una amplia gama de materiales como metales, cerámica y madera, se sabe que hay una creciente atención enfocada en este campo [17]. Mediante esta técnica es posible hacer cortes geométricos de gran dificultad usando sistemas automatizados y que ya no dependen de habilidades manuales. Gracias a los avances, el grado de precisión de la forma, la velocidad de eliminación y la calidad de la superficie durante el proceso de maquinado

dependerán de las propiedades de los materiales, de las características de la fuente láser y de los parámetros del proceso. [18]

El tallado por láser 3D es un grabado en capas de forma secuencial [19]. La repetición de este proceso permite trabajar zonas en donde exista material que a lo largo de muchas capas se fue acumulando (no pasó el láser, o se apagó) y en otras se fue retirando (láser encendido) hasta que pueda ser observado el modelo completo. Los tipos de láser tienen alcances diferentes y es importante la selección del más adecuado de acuerdo al material a tratar, de manera general se dividen de acuerdo al medio activo, ya sea gas o sólido [20]. Es importante elegir parámetros de proceso optimizados como longitud de onda, diámetro del punto, frecuencia de repetición del pulso, velocidad y potencia del haz tomando en cuenta la absorción del material y la calidad de la superficie requerida [18]. Conforme se va realizando el tallado, la distancia entre el láser y el material se va haciendo mayor, por lo que se alejará gradualmente del plano focal y esto tiene como consecuencia que la densidad de potencia del láser disminuya e igualmente la calidad, por lo que se deberá buscar que la mesa de trabajo y el avance del láser tengan un movimiento coordinado y de la misma magnitud para que la distancia de trabajo se conserve, o bien, ir ajustando el plano focal con el avance en la pieza. [19]. En la Figura 2.19(a) se muestra una ilustración de una máquina láser encontrada en el mercado y en la Figura 2.19 (b) se muestra una máquina casera realizando el tallado en madera.

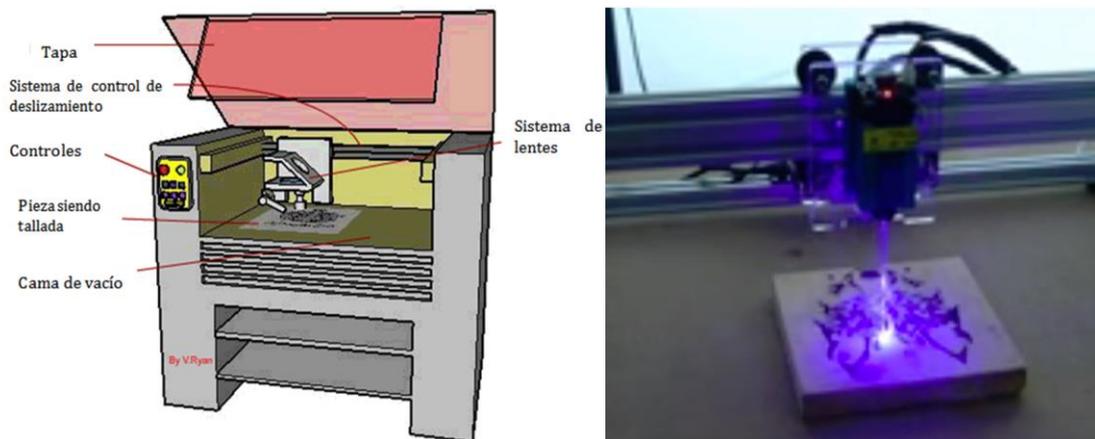


Figura II.19 (a) Diseño común de máquinas de tallado láser con sus partes principales. (b) Máquina láser realizando tallado en madera.

1.4 Clasificación de las Maderas

Desde el punto de vista de la botánica, hay dos grandes clases en las que se dividen las maderas: maderas suaves y maderas duras. Esta clasificación se refiere directamente a la forma de reproducción del árbol y no a sus características de dureza. La madera dura proviene de angiospermas que podemos generalizar como caducifolias, crecen en climas más templados y tropicales y su semilla se encuentra en el ovario de la flor. La madera suave proviene de gimnospermas o coníferas y en su mayoría son de hoja perenne. Anatómicamente no son porosas a diferencia de las duras que poseen poros o *vesse/s* abundantes.

La Figura 2.20 muestra una serie de micrografías de bajo aumento de la sección transversal de diferentes tipos de madera. Las micrografías tienen un aumento aproximado de 15x. Su color es una consecuencia de las manchas utilizadas para acentuar las características anatómicas y no es indicativo del color real de la madera.

El álamo amarillo es generalmente de grano recto y de textura comparativamente uniforme; es de crecimiento lento, moderadamente ligera en peso y moderadamente baja en resistencia a la flexión, moderadamente blanda y moderadamente baja en resistencia al choque. El cedro español se considera resistente a la descomposición y aunque no tiene una alta resistencia, la mayoría de las otras propiedades son similares a las de la caoba americana, se usa para todos los propósitos que requieren una madera duradera, liviana, de grano recto y fácil de trabajar.

La madera más suave para tallar es conocida como *Basswood* o Tilo americano y es de las favoritas para su tallado [21] [22]. La madera más usada en la estructura de muebles, por ser resistente, es la de pino, pero de acuerdo a la región y al subtipo, presentan diferentes propiedades así como vetas con una densidad distinta lo que modifica el avance del tallado en esas zonas por lo que se busca que sea lo más homogénea posible. [23]

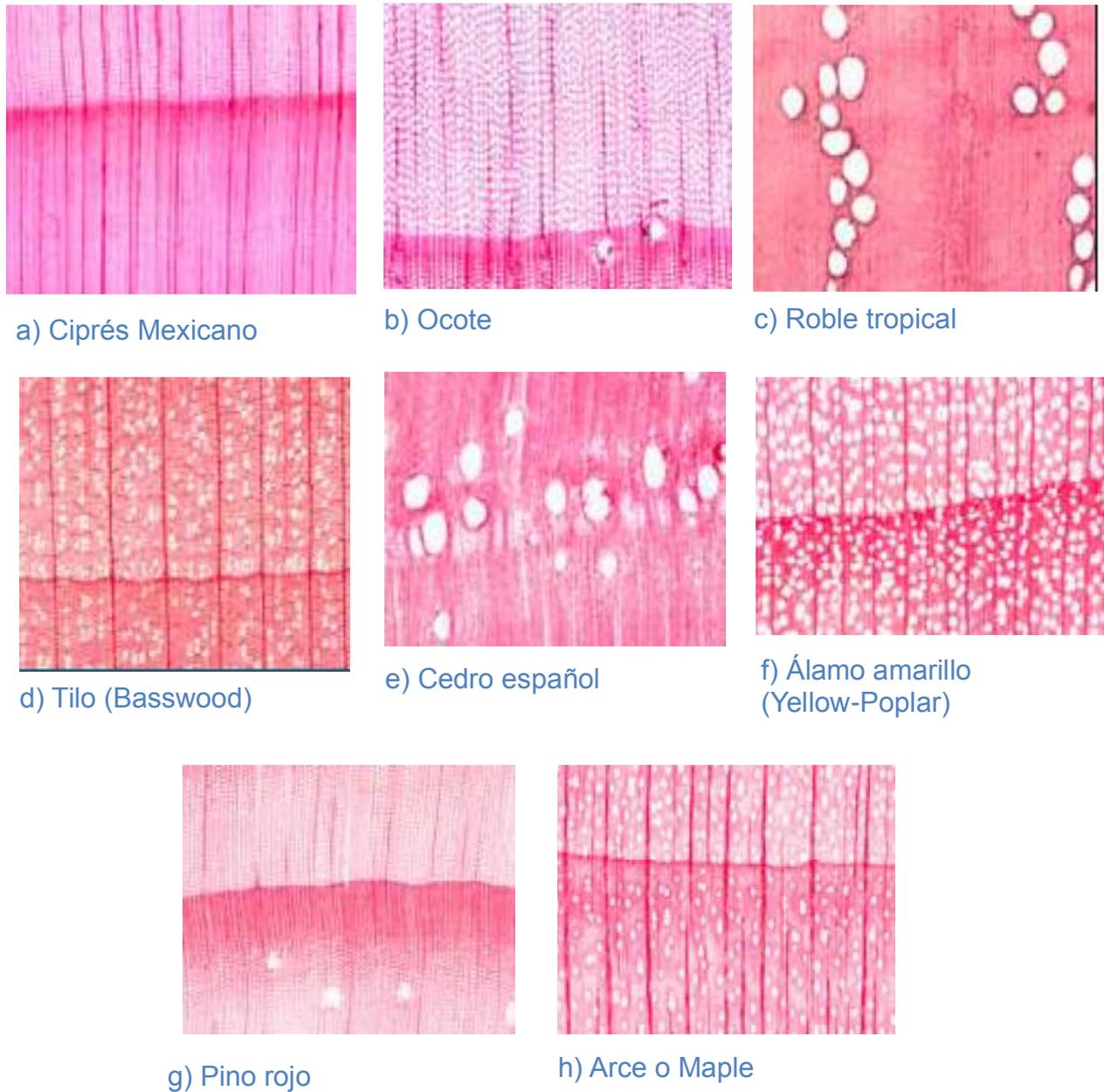


Figura II.20 Micrografías de la sección transversal de diferentes tipos de madera.

Una consideración por agregar, es que deberá de ser deseable la búsqueda de tablones a trabajar con los anillos del crecimiento de la madera lo más lejano al centro debido a que son más espaciados, lo que permite tener una mayor uniformidad.

1.4.1 Las Maderas y el Láser

La madera se descompone y carboniza alrededor de los 300 ° C y tiene un carácter exotérmico. Se forman gases y vapores, y cuando el oxígeno reacciona con los productos de destilación, se produce la combustión.

La interacción del rayo láser con la madera está determinada principalmente por el tipo de láser que se utiliza. El rayo láser actúa como una fuente de calor localizada, que calienta la madera por encima de la temperatura de descomposición, por lo que se lleva a cabo la descomposición localizada lo que vaporiza algo de carbón residual, y en la mayoría de los casos, dejando una capa superficial de carbón sobre la pieza.

Se puede especificar una velocidad óptima para un corte de buena calidad con respecto al grado de carbonización y la suavidad de la superficie de corte. Para obtener la máxima eficiencia, la combinación adecuada de velocidad de alimentación y potencia dependerá del grosor de la pieza de trabajo, el ancho de corte deseado y la densidad de la madera. Cuanto mayor es la velocidad de corte, menos tiempo hay para que el calor se difunda lateralmente a través de las superficies cortadas y más estrecho es el ancho de la ranura. Si la velocidad de trabajo de una máquina CNC se establece demasiado alta, el frente de corte no tendrá suficiente tiempo de exposición para penetrar el material por completo y el material de desecho no dejará la zona de interacción fuera del fondo del corte.

Cuando se corta perpendicularmente a la orientación de la fibra, el material se ve más gravemente afectado que cuando se corta en paralelo a la orientación de la fibra. Dado que la fibra de carbono tiene una conductividad media mucho más alta (casi dos órdenes de magnitud más alta) que la matriz polimérica, la conducción de la corriente es a lo largo de la dirección de la fibra.

Ahora bien, si se usa madera laminada, o MDF, hay pruebas que demuestran variaciones potenciales en velocidad de corte. La capacidad de corte puede estar relacionada con las uniones del material orgánico, el agente de unión utilizado y el residuo que consiste en madera carbonizada, algunos hidrocarburos, alquitrán, etc.

1.5 Máquinas de Control Numérico Computarizado CNC

En el campo de la ingeniería, con el paso de los años, los sistemas de control han asumido un papel fundamental en el desarrollo y avance de la tecnología a nivel mundial, las máquinas de control numérico computarizado (CNC) son un ejemplo donde se observa la aplicación del avance en los sistemas de control.

En su forma más sencilla un sistema de control produce una salida para una entrada dada. Si queremos que el sistema sea mejor, se pueden conectar los sistemas y procesos para controlar las salidas de mejor manera. Cabe mencionar que difícilmente la respuesta real será instantánea, también hay factores como el inicio desde reposo o desde alguna velocidad o posición. Aquí también podemos tener un error conocido como *error en estado estable* que es la diferencia entre la entrada deseada y la salida obtenida incluso cuando se considere que el sistema ha llegado a la respuesta deseada. Este error se puede encontrar en sistemas de control defectuosos y en ocasiones puede ser tolerable, dependiendo el área de aplicación en que se encuentra y la magnitud del mismo. [24]

1.5.1 Sistema de control

Existen básicamente dos configuraciones de sistemas de control: de lazo abierto y de lazo cerrado, las cuales se ilustran en la Figura 2.21. La característica distintiva de lazo abierto es que no puede compensar ninguna perturbación que ocurra entre el controlador y el proceso, lo que hará que la señal entre con ruido al proceso y perturbará la información que entre, haciendo que la información de salida también se encuentre corrompida, por lo tanto, este tipo de sistemas solo se controlan por la entrada.

El sistema de lazo cerrado, en cambio, si puede compensar las perturbaciones mediante la medición de la respuesta de salida, alimentando esa medida a una trayectoria de realimentación y comparando esa respuesta con la entrada en el punto de suma. Si hay alguna diferencia entre ellas, el sistema acciona la planta, de lo contrario, si no hay diferencia, la planta no se acciona, debido a que la respuesta deseada ya ha sido obtenida. Por lo tanto, aunque los sistemas cerrados puedan ser menos sensibles al ruido, a perturbaciones o cambios en el entorno, son más precisos que los de lazo abierto. Y son más flexibles a la hora de hacer

modificaciones o realizar alguna compensación, aunque su complejidad a la hora de diseñarlos los hace más costosos. [24]

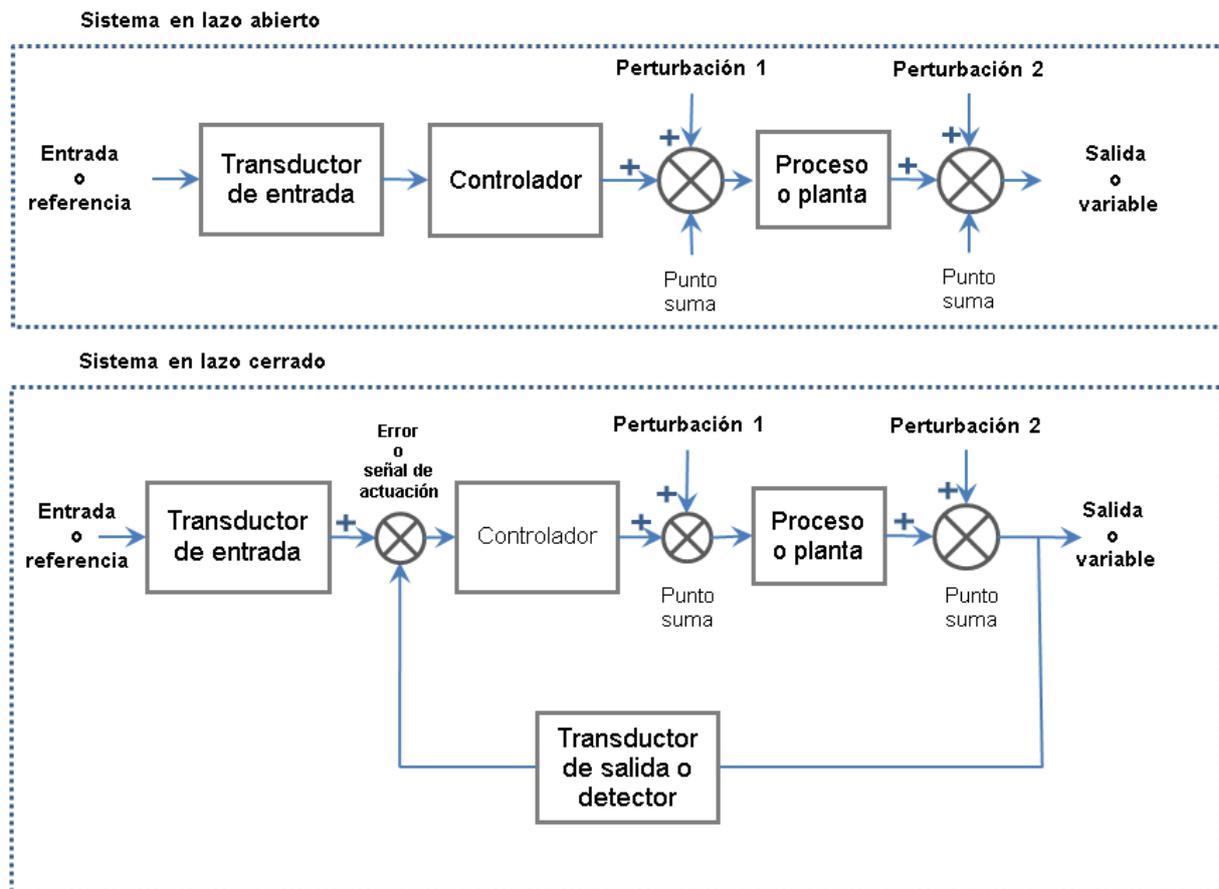


Figura II.21 Configuración de lazos de control: abierto y cerrado.

1.5.2 Control digital

El control digital es una implementación de un sistema de control empleando lógica programada. Los controladores digitales se usan buscando alcanzar el máximo desempeño en cuanto a productividad y beneficio buscando el uso eficiente de energía y mínimos costos. Un sistema de control digital se diseña utilizando tres componentes básicos: Actuadores, Sensores y Controlador.

1.5.2.1 Actuadores

Hay muchas clasificaciones de los actuadores. Aquellos que operan directamente un proceso (carga, planta) se llaman actuadores del proceso. En aplicaciones de control de proceso en particular, los actuadores se utilizan para

operar componentes de controlador (elementos de control final), tal como servo-válvulas, llamados actuadores de control. Se llaman servo-actuadores si usan automáticamente señales de error de la respuesta en un proceso de retroalimentación para corregir la operación del proceso (es decir, manejar el proceso de acuerdo con una respuesta deseada). En particular, los motores que utilizan la posición, velocidad y tal vez mediciones de par de carga y armadura actual o corriente de campo en realimentación, para manejar una carga según un movimiento especificado, son llamados servo-motores. [25]

Los motores de pasos pueden clasificarse como actuadores, han estado en uso durante más de sesenta años, pero sólo después de la incorporación de circuito de estado sólido y dispositivos lógicos en sus sistemas de impulsión, los motores de pasos han surgido como alternativas rentables para servomotores de corriente directa en aplicaciones de alta velocidad y control de movimiento.

Hay tres tipos básicos de motores de pasos:

1. Reluctancia Variable (VR), que tienen rotores de magneto blando
 2. Imán permanente (PM), que tienen rotores magnetizados
 3. Motores de pasos híbridos, que tienen dos pilas de dientes del rotor formando los dos polos de un imán permanente situado sobre el eje del rotor.
- [25]

En los motores de corriente directa (CD), la inercia del rotor se ha hecho muy pequeña, por lo que hay motores con relaciones muy altas entre el par y la inercia. Los servomotores de CD que manejan corriente mediana y grande se usan en sistemas robóticos y en máquinas de fresado controladas numéricamente.

Un servomotor de CD es manejado por un controlador de movimiento electrónico, denominado servo-controlador, como una combinación motor-controlador. El servo-controlador determina el movimiento de un servomotor de CD y opera en diversos modos. Algunas de las características son el posicionamiento de punto a punto y la programación de perfiles de velocidad y aceleración. El uso de un manejador de movimiento electrónico mediante un manejador de modulación por ancho de pulso para controlar un servomotor de CD es común en los sistemas de

control de robots, numérico y de posición y/o velocidad. El control de movimiento (control de posición y velocidad) implica el control indirecto del torque del motor ya que este es el que causa movimiento. [25] La Figura 2.22 muestra un ejemplo de servomotor y algunas partes que lo conforman.

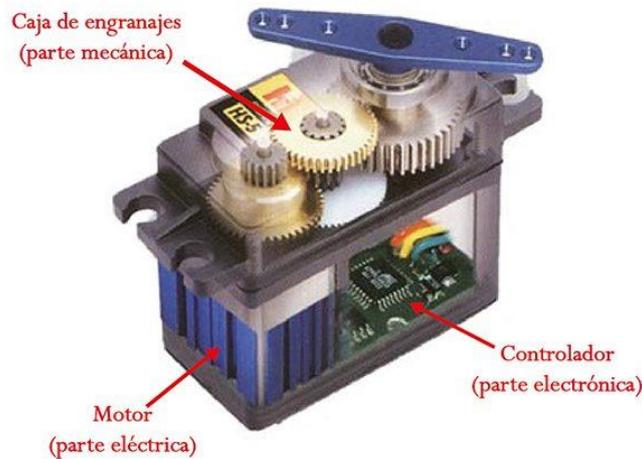


Figura II.22 Servomotor con sus partes principales.

El diagrama de bloque del servomotor en la Figura 2.23 representa de forma visual el servomotor como un sistema. El circuito electrónico es el encargado de recibir la señal PWM y traducirla en movimiento del motor de CD. El eje del motor está acoplado a un potenciómetro, el cual permite formar un divisor de voltaje. El voltaje en la salida del divisor varía en función de la posición del eje del motor de CD. [26]

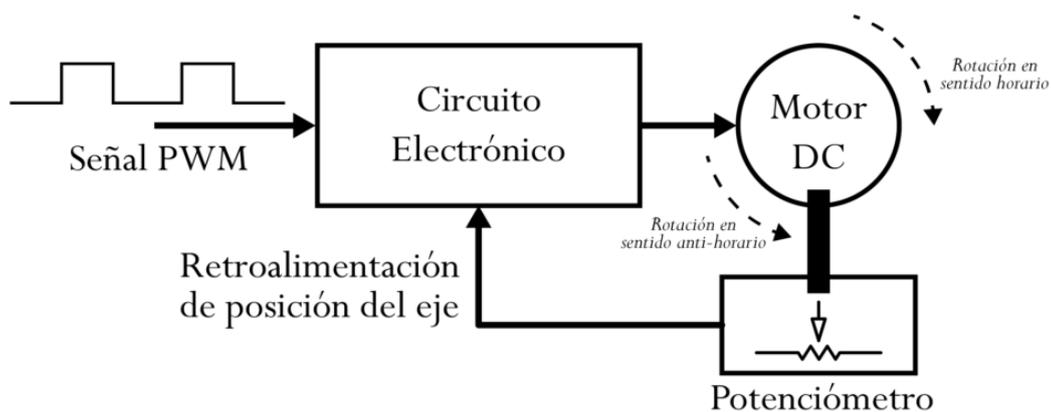


Figura II.23 Diagrama a bloques de un servomotor.

Existen servomotores para muy diferentes usos. Los servomotores de modelismo, son los utilizados en prototipos de robótica, estos operan a voltajes bajos en corriente directa, típicamente entre 4 y 6 voltios. Los servomotores industriales operan tanto en CD como en CA (monofásico o trifásico).

Dentro de los diferentes tipos de servomotores, éstos se pueden clasificar según sus características de rotación.

- Servomotores de rango de giro limitado: Son el tipo más común de servomotor. Permiten una rotación de 180 grados, por lo cual son incapaces de completar una vuelta completa.
- Servomotores de rotación continua: Se caracterizan por ser capaces de girar 360 grados, es decir, una rotación completa. Su funcionamiento es similar al de un motor convencional, pero con las características propias de un servo. Esto quiere decir que podemos controlar su posición y velocidad de giro en un momento dado.

Los servomotores de rango de giro limitado se pueden adecuar para que funcionen como servomotores de rotación continua. Sin embargo, si requerimos un servo de 360 grados es mejor usar uno que haya sido diseñado para este tipo de uso. [26]

1.5.2.2 Sensores

Es considerado un elemento de medición que afecta la indicación del valor real de la variable de salida, lo que lo convierte en un elemento relevante para el desempeño del sistema de control. Por lo general, el sensor determina la función de transferencia en la trayectoria de realimentación en los sistemas de lazo cerrado; es decir, pueda usarse para comparar la salida con la señal de entrada de referencia. El punto de ajuste del controlador debe convertirse en una entrada de referencia con las mismas unidades que la señal de realimentación del sensor o del elemento de medición.

En el diseño de un sistema de control, debe decidirse el tipo de sensor a usarse (electrónico, neumático o hidráulico) considerando el sistema completo. Si

existen fluidos inflamables, los componentes neumáticos eliminan la posibilidad de generación de chispa. De lo contrario, los más usados son los electrónicos. [24]

1.5.2.3 Controlador

Estos sistemas, tienen la capacidad de toma de decisiones y de ser flexibles, tanto para su configuración, como para su toma de decisiones. Los sistemas de control moderno pueden hacer uso desde una computadora de escritorio hasta un micro-controlador y es una parte esencial del sistema.

Para que el controlador dé su máximo, se apoya en sensores, que recogen la información del entorno de forma que el sistema de control entienda y en base a eso realice la toma de decisiones. También cuenta con actuadores que, mediante la decisión tomada, realizan la acción deseada. Visto desde el análisis matemático, en el uso de los controles digitales, la transformada de Laplace se reemplaza con la transformada Z donde una señal de tiempo discreta en la forma de una secuencia de números complejos o reales es convertida en una representación de dominio de frecuencia compleja.

1.5.3 Dispositivos de Control Numérico

Los dispositivos de control numérico son sistemas integrados por un sistema de control, actuadores y sensores, en ocasiones estos dos últimos pueden estar unidos. Bajo el control de una computadora dirigen la posición de sus herramientas. Estas órdenes incluyen las tareas y sus velocidades, así como algunas especificaciones en torno a aspectos, tales como la temperatura, la vibración, las características específicas de los materiales utilizados, el desgaste de las herramientas, etc., de manera que, comparando la situación real del sistema con una programada previamente, pueden realizarse los ajustes necesarios para evitar desviaciones en la producción. Son utilizadas para la fabricación de piezas que precisan de una elaboración compleja.

El control numérico permite a un operador comunicarse con la máquina-herramienta a través de una serie de números y símbolos. El control numérico, rápidamente se convirtió en Control Numérico Computarizado (CNC) y ha permitido a la industria producir de manera consistente piezas con altísima precisión y en

grandes volúmenes. Los comandos operativos que controlan las máquinas CNC se ejecutan automáticamente con gran velocidad, precisión, eficiencia y repetitividad. El uso de las máquinas CNC ha ido en aumento, esto requiere personal con capacidad para preparar los programas que guían a las máquinas para producir piezas con la forma y precisión requeridas.

1.5.3.1 Sistema Cartesiano

El sistema de coordenadas Cartesiano o rectangular nos permite describir cualquier punto específico en términos matemáticos a lo largo de tres ejes perpendiculares. Este concepto se adapta perfectamente a las máquinas CNC ya que su construcción generalmente se basa en tres ejes de movimiento (X, Y, Z) más un eje de rotación; que permite producir piezas 3D en una máquina CNC. Los movimientos de la máquina CNC utilizados en la elaboración de un producto son de dos tipos básicos: punto a punto (movimientos en línea recta) y recorrido continuo (movimientos de contorneado). Los sistemas CNC se basan en gran medida en el uso de coordenadas rectangulares de manera que el programador puede localizar cada punto en el área de trabajo con precisión.

1.5.3.2 Sistema de programación

Se usan dos sistemas de programación: el incremental y el absoluto. Ambos tienen aplicaciones en la programación CNC y no se puede decir cuál de ellos es el mejor. En la figura 2.24 se muestra un ejemplo de una pieza dimensionada con el método incremental en (a) y con el método absoluto en (b). Las ubicaciones de los programas incrementales siempre se dan como la distancia y la dirección desde el punto inmediatamente anterior. El comando G91 indica a la computadora y a la UCM (Unidad de Control de la Máquina) que la programación está en el modo incremental.

Las ubicaciones absolutas del programa siempre se dan desde un solo cero fijo o punto de origen. El cero o punto de origen puede ser una posición en la máquina, como una esquina de la mesa de trabajo o en cualquier punto específico de la pieza. En el dimensionamiento y la programación absolutos, cada punto o ubicación en la pieza se da a una cierta distancia del punto de referencia o cero. El

comando G90 indica a la computadora y a la UCM que la programación está en el modo absoluto.

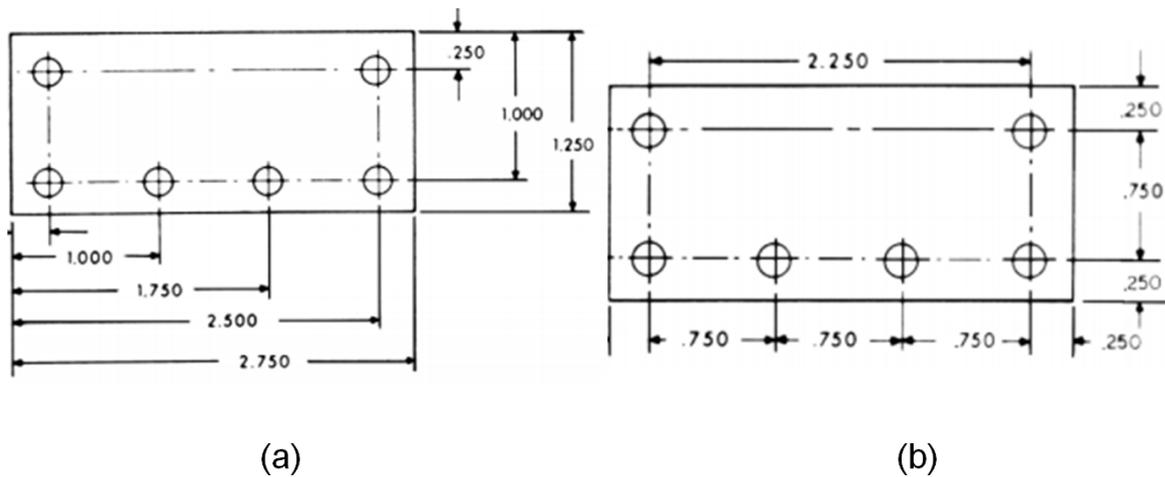


Figura II.24 Pieza dimensionada. (a) Modo incremental y (b) Modo absoluto del sistema.

1.5.3.3 Recorrido continuo

El mecanizado de trayectoria continua o contorneado implica un trabajo donde la herramienta de corte está en contacto con la pieza a medida que se desplaza de un punto programado al siguiente. El posicionamiento continuo de la trayectoria es la capacidad de controlar los movimientos en dos o más ejes de la máquina simultáneamente para mantener una relación constante entre el cortador y la pieza. La información contenida en el programa CNC debe posicionar con precisión la herramienta de corte de un punto al siguiente y seguir una trayectoria precisa predefinida a una velocidad de avance programada para producir la forma o el contorno requerido. El método por el cual las máquinas CNC de recorrido continuo se mueven de un punto programado al siguiente se llama interpolación. Existen cinco métodos de interpolación: lineal, circular, helicoidal, parabólica y cúbica. Todos los controles de contorneado proporcionan interpolación lineal y la mayoría de los controles son capaces de interpolación tanto lineal como circular. La interpolación helicoidal, parabólica y cúbica es utilizada por industrias que fabrican piezas complejas.

La interpolación lineal consiste en cualquier punto programado unido entre sí por líneas rectas, ya sea que los puntos estén juntos o alejados. Las curvas se pueden producir con interpolación lineal dividiéndolas en segmentos cortos y rectos.

Este método tiene limitaciones, ya que se tendría que programar una gran cantidad de puntos para describir la curva y así producir una forma de contorno. Un contorno programado en interpolación lineal requiere las posiciones de coordenadas XY del inicio y el final de cada segmento de línea. Por lo tanto, el punto final de una línea o segmento se convierte en el punto de inicio del siguiente segmento, y así sucesivamente, a lo largo de todo el programa.

El desarrollo de UCMs capaces de interpolación circular ha simplificado enormemente el proceso de programación de arcos y círculos. Para programar un arco, la UCM requiere solo las posiciones de coordenadas del centro del círculo, el radio del círculo, el punto de inicio y el punto final del arco que se está cortando y la dirección en la que el arco debe cortarse, en sentido horario o anti-horario. La información requerida puede variar para diferentes UCM.

1.6 Impresoras 3D

La impresión 3D crea objetos sólidos 3D de cualquier forma o geometría a partir de un archivo digital. La creación se logra estableciendo capas sucesivas de un material específico hasta que se crea todo el objeto. Cada una de estas capas representa una sección transversal horizontal en rodajas finas del objeto final. [27]

Los procesos de impresión 3D para crear objetos físicos a partir de diseños digitales puede variar, se pueden diferenciar estos procesos en la forma en que van creando las capas y en los materiales que se utilizan para la impresión. Las impresoras 3D son construidas con los siguientes componentes básicos:

1. **Cama:** Es una base donde se deposita la pieza mientras es impresa.
2. **Alimentador:** También llamado **extrusor**, que alimenta el filamento al cabezal.
3. **Cabezal:** Incluye la boquilla (la zona por la que se dosifica el material), en los plásticos se requiere que cuente con un sensor de temperatura y la fuente de calor.
4. Material de entrada o **filamento** (PLA, ABS, Nylon, PTEG, TPU-FLEX)
5. **Sistema de control** o tarjeta madre (embebido, genérico o dedicado)

6. **Sistema cartesiano:** Es el sistema de movimiento del cabezal, incluye los motores, bandas y/o rodamientos, ejes de deslizamiento y sensores.
7. **Fuente de poder**

Algunas impresoras 3D cuentan con una pantalla para visualizar los controles, el estado de la impresora y hacer la manipulación de los archivos de entrada sin la necesidad de una conexión directa a una computadora.

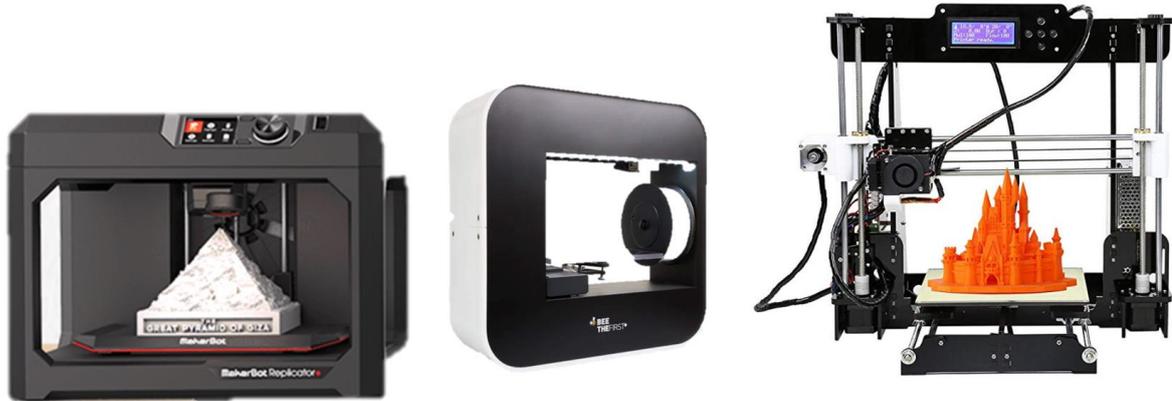


Figura II.25 De derecha a izquierda: Las impresoras *Anet A8*, *Bee-the-first*, *Replicator+*

En el mercado se manejan distintos modelos de impresoras láser. Las mostradas en la Figura 2.25 se muestran de izquierda unas de las más populares del momento, de la marca MAKERBOT que cuenta con sistemas novedosos como cámara integrada para vigilar tu impresión desde tu celular con software propio de la compañía que permite hacer muchas adecuaciones; de la compañía *Bee-very-creative* una impresora que es más básica pero con un modelo estético y que además tiene su propio software; la tercera es una impresora china llamada *Anet A8* que viene desarmada y que requiere que el usuario la ensamble esta es considerada para principiantes que no desean invertir mucho, pero que sienten capacidad técnica suficiente para armarla, e incluso hacerle mejoras. Esta última no cuenta con software de la empresa, si no que se apoya de otras que son abiertas al público como *cura* o *Repetier Host*.

El proyecto *RepRap* [28] es una iniciativa creada con el propósito de diseñar una máquina de prototipado rápido que sea capaz de replicarse a sí misma , es decir, generar las partes necesarias para construir otra máquina igual a ella, para lo cual proporciona toda la información en la elaboración de una impresora 3D, proporcionando los archivos a imprimir, los componentes electrónicos necesarios y

una lista de proveedores, así como el procedimiento de ensamblaje y detalle de errores comunes.

La tarjeta de control debe de contar con un microprocesador, *drivers* para controlar los motores del sistema cartesiano y puertos de entrada y salida (E/S) de datos. La Figura 2.26 muestra una tarjeta de control usada para la impresión 3D.

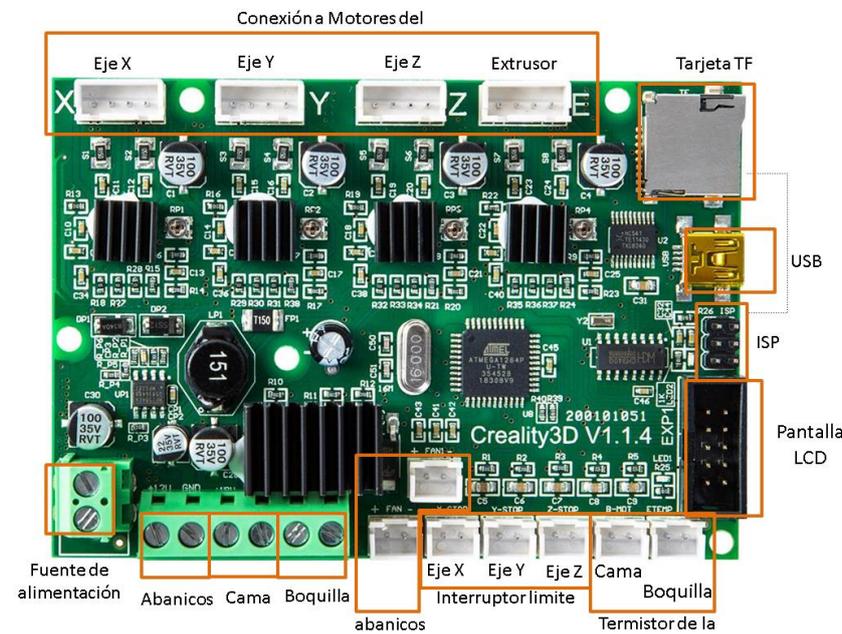


Figura II.26 Tarjeta de control de la impresora Ender 3.

La tarjeta madre creada por la compañía *Creality* [29] es una de las más sencillas y proporciona buenos resultados de impresión. Tiene un microcontrolador de 8 bits con capacidades de lectura y escritura, Puerto serie SPI, un contador en tiempo real, tres temporizadores/contadores flexibles con modos de comparación y PWM utiliza el módulo *CH340G* para convertir rápida y fácilmente las señales serie a USB y los drivers *A4988* que manejan los motores de pasos.

El Arduino Uno [30] es una tarjeta de control con un microcontrolador de código abierto basado en el microchip *ATmega328P*. La tarjeta está equipada con puertos de E/S digitales y analógicos que pueden conectarse a varias tarjetas de expansión y otros circuitos. Utiliza el módulo *Atmega16U2* programado como un adaptador USB a serie. El *shield CNC* permite manejar 3 motores de pasos de forma independiente (X, Y, Z) y 1 motor adicional (A) como duplicado de alguno de los anteriores. Es compatible con los drivers para motores de pasos como el *Pololu A4988* de *Allegro* o los *DRV8825* de Texas

Instruments. El módulo *A4988* puede manejar motores de pasos de hasta 2A por bobina y *microstepping* de 1/16 de pulgada; el driver *DRV8825* trabaja con motores de hasta 2.5A por bobina y *microstepping* de hasta 1/32 de pulgada, pudiendo configurar de forma independiente la resolución del *microstepping* de cada driver con los 3 *jumpers* correspondientes. El *Shield CNC GRBL* con un *Arduino Uno*, Drivers *A4988* o *DRV8825* y una fuente de alimentación, permite construir una máquina CNC de una manera rápida y sencilla.

1.7 Grabadoras láser

El proceso de grabado láser es una técnica no convencional para modificar la superficie de un material mediante calentamiento y vaporización de material en la zona deseada utilizando un módulo emisor de rayos láser. Este proceso requiere del control de parámetros tales como el diámetro de punto al enfocar el rayo láser, potencia del láser, frecuencia del láser, longitud de onda, tasa de eliminación de material, espacio del grabado y otros. Se puede considerar que una máquina grabadora láser consta de 3 partes esenciales: un módulo emisor de rayos láser, un controlador y una superficie. El rayo láser emitido permite al controlador trazar patrones en la superficie a trabajar. El controlador, da la dirección, intensidad, y velocidad de movimiento. Hay tres tipos principales de máquinas de grabado láser:

- a. Piezas de trabajo cilíndricas o piezas de trabajo planas montadas alrededor de un cilindro donde el láser atraviesa efectivamente una hélice fina y el pulso de encendido/apagado del láser produce la imagen deseada sobre una base entramada (Ver Figura 2.27A).
- b. Mesa X-Y con tres posibles configuraciones: 1) pieza estacionaria y el láser se mueve en direcciones X e Y dibujando vectores, 2) el láser es estacionario y la pieza de trabajo se mueve y 3) el láser se mueve en el eje X y la pieza en el eje Y (Ver Figura 2.27B).
- c. Los espejos galvanométricos mueven el rayo láser, la pieza y la mesa son estacionarios (Ver Figura 2.27C).

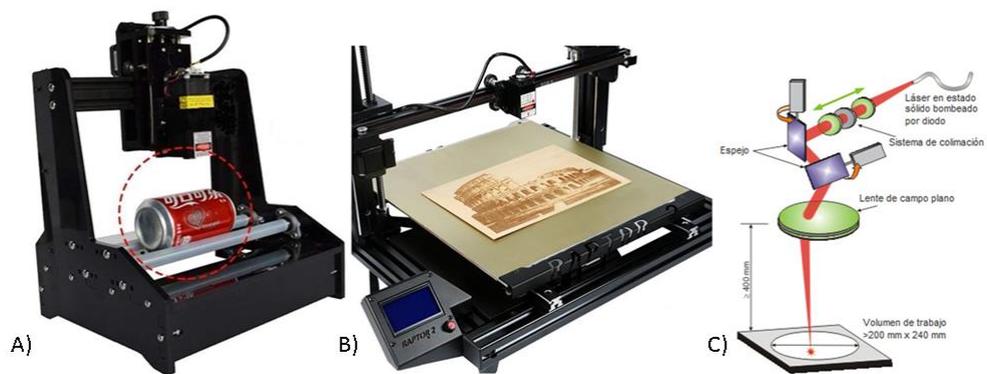


Figura II.27 Tipos principales de máquinas de gravado láser: A) Piezas de trabajo cilíndricas B) Mesa X-Y y C) Espejos galvanométricos.

Capítulo III. MODELADO TRIDIMENSIONAL DE OBJETOS

La creación de modelos de objetos en tres dimensiones puede hacerse por distintos métodos, teniendo toda una serie de ventajas y desventajas a considerar a la hora de seleccionar alguno. En este proyecto se consideraron dos formas de generar los datos tridimensionales de objetos para tallarlos en madera utilizando tecnología láser. Primero se consideró la generación de datos de objetos de forma artificial por medio de código, usando Matlab, o con un programa de diseño asistido por computadora (CAD). La otra forma es el uso de un escáner 3D para la captura de datos.

1.8 Generación del Modelo por Software

Cuando nos referimos a la generación de modelos tridimensionales de piezas por software consideramos dos métodos: por código con Matlab y con un sistema de CAD. Usando Matlab como entorno de desarrollo se creó el modelo 3D de una figura por medio de código, esta es una pirámide de cuatro lados. Para generar el modelo 3D se requiere crear una nube de puntos, por lo que se generaron las coordenadas XYZ de los puntos mediante ciclos *for* anidados. La pieza resultante se muestra en la Figura 3.1. El archivo con los datos de la pieza es guardado en formato de nube de puntos para ser procesado posteriormente por el software del sistema de tallado láser.

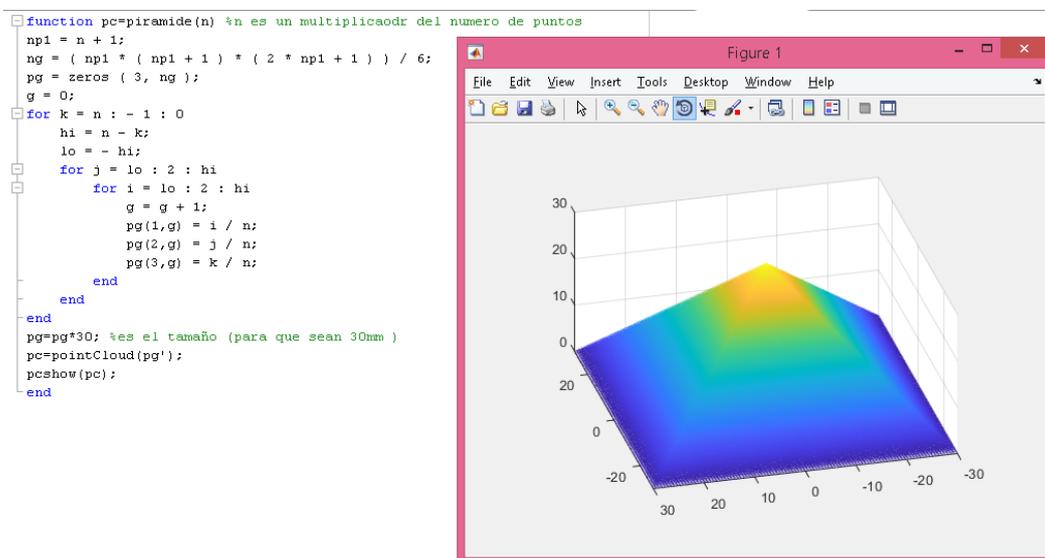


Figura III.1 A la izquierda el código en Matlab y a la derecha la nube de puntos generada.

Se utilizó el sistema de CAD denominado *Solid Works* para generar la esfera mostrada en la Figura 3.2. De esta forma se muestra otra opción para la generación de objetos 3D artificiales. El archivo con los datos de la pieza es guardado en formato STL para ser procesado posteriormente por el software del sistema de tallado láser.

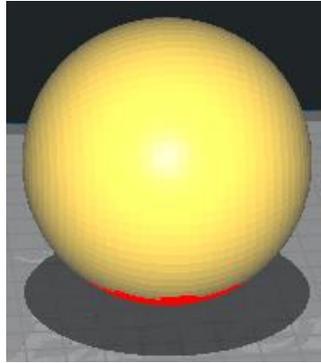


Figura III.2 Esfera realizada en Solid Works.

1.9 Generación de la Pieza con Escaneo Tridimensional

Si la intención es implementar un proceso de ingeniería inversa lo más completo posible, el primer paso a considerar es la adquisición tridimensional de la figura a trabajar. Para esto se considera el uso de un escáner tridimensional portátil que utiliza un emisor de rayos infrarrojos (IR) y cámaras para obtener datos tridimensionales mediante la medición del tiempo de vuelo, tiempo que tarda el rayo IR en ser emitido, reflejado por el objeto y captado por la cámara. El escáner utilizado en este proyecto es el *3D Sense 1* de la compañía 3D Systems, mostrado en la Figura 3.3.



Figura III.3 Escáner 3D Sense 1 usado para la digitalización de las piezas.

Este escáner es de corto alcance con resolución Full HD que captura y procesa los datos a color en tiempo real para generar una malla poligonal completa, ya que cuenta con su propio software de mallado. Para este proyecto se almacenaron los datos sin procesar en archivos de puntos “.ply” y “.obj”, con la intención de trabajar formatos de uso común, además de la necesidad de procesar los datos para realizar modificaciones geométricas y eliminar puntos no deseados.

En este proyecto se seleccionaron dos objetos con diferentes geometrías, una compleja y otra sencilla, esto con la finalidad de evaluar la calidad de los datos adquiridos. La Figura 3.4 muestra el archivo a partir del cual se empezó a trabajar. La visualización de la nube de puntos se realizó en Matlab haciendo uso funciones para el procesamiento de nubes de puntos. La geometría de este objeto se considera compleja.

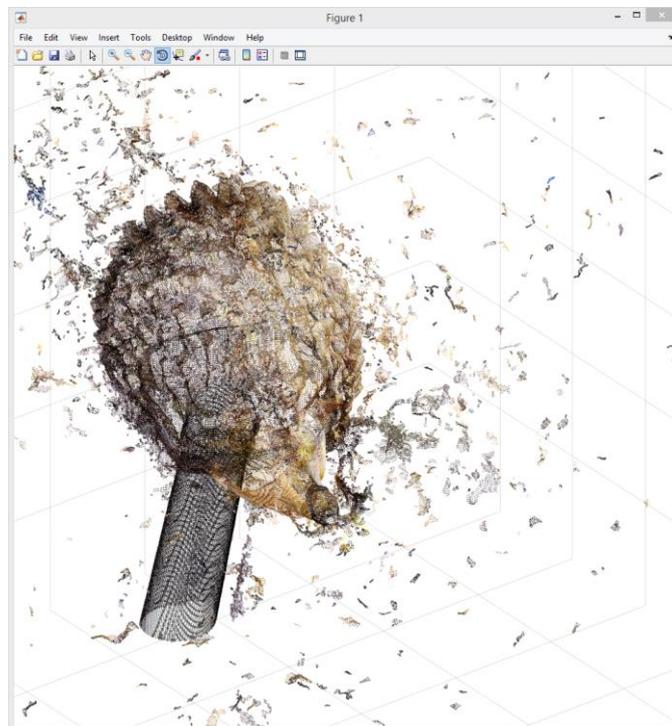


Figura III.4 Nube de puntos sin procesar adquirida con el escáner 3D Sense 1.

La nube de puntos resultado del escaneo tridimensional presenta ruido producido durante el proceso de adquisición. Fueron necesarias varias pruebas para mejorar los resultados, cabe destacar que la iluminación es un aspecto importante debido a que las sombras generan gran parte del ruido. Para eliminar el ruido es necesario aplicar un proceso de filtrado sobre los datos.

Como se puede observar en la figura, el objeto escaneado se encuentra inclinado en los tres ejes y con una base ajena que lo sostiene. Lo primero que se hizo fue medir los ángulos de inclinación para determinar la matriz de rotación que haga el eje X horizontal y el eje Y vertical. Se utilizó la función *pctransform* de Matlab para rotar los ejes del objeto escaneado. Luego se procede a seleccionar manualmente una región de interés que encierre la nube de puntos del objeto escaneado y así poder eliminar el ruido fuera de esa región. Los datos del puerco espín escaneado permiten ver la importancia de hacer un buen escaneo.

En la Figura 3.5 se muestra otra pieza escaneada que se considera de geometría sencilla y no muestra demasiado ruido. Se le aplicó el mismo procedimiento de rotación y eliminación de ruido que a la pieza de geometría compleja.

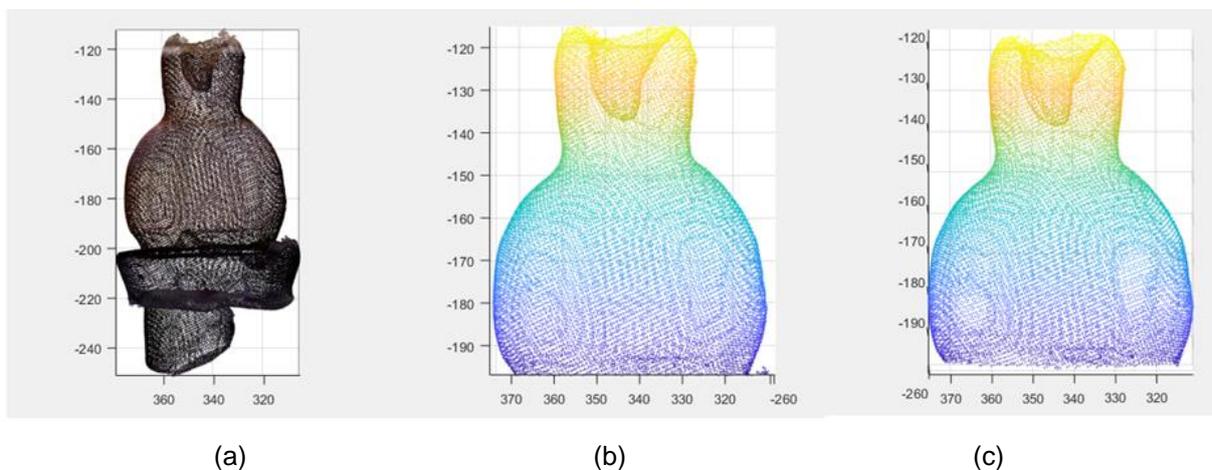


Figura III.5 Objeto escaneado: a) Aplicación de rotación a los ejes, b) Región de interés y c) Filtrado de la nube de puntos.

Para el filtrado de la nube de puntos, se hizo uso de una función llamada *pcdenoise* de Matlab de la siguiente manera:

```
ptCloudOut = pcdenoise(ptCloudN1, 'NumNeighbors', 10, 'Threshold', 10)
```

En donde *ptCloudN1* es la nube de puntos de entrada, *NumNeighbors* nos indica la forma en la que se llevará a cabo el filtrado, que con esta indicación es mediante un valor positivo y entero que se usa para estimar la media de la distancia promedio a los vecinos de cada punto evaluado. Disminuir este valor hace que el filtro sea más sensible al ruido, pero incrementa el número de cálculos. El valor de

umbral o *Threshold* considera un punto fuera de rango o descartable si la distancia promedio a sus vecinos más cercanos está por encima del umbral especificado.

1.10 Generación del Modelo Tridimensional del Objeto

La nube de puntos filtrada fue procesada para generar el modelo tridimensional del objeto escaneado realizando el enmallado por medio de funciones de triangulación de puntos tridimensionales. Debido a que fue demasiado el tiempo para procesar la nube de puntos se utilizó la función *pcdownsample* para reducir el número de puntos. Se usó el método de *gridStep* que define una rejilla cúbica en la que todos los puntos dentro de los cubos se reducen a un solo punto de salida con coordenadas en el centro del cubo. De esta forma también se eliminan puntos repetidos que impiden llevar a cabo el proceso de reconstrucción tridimensional. A comparación de los métodos aleatorios (*random*) y de rejilla de muestreo no-uniforme (*nonuniformGridSample*), el método de *gridStep* seleccionado preserva mejor la geometría de la nube de puntos. La Figura 3.6 muestra el resultado de aplicar el sub-muestreo a la nube de puntos de la Figura 3.5c.

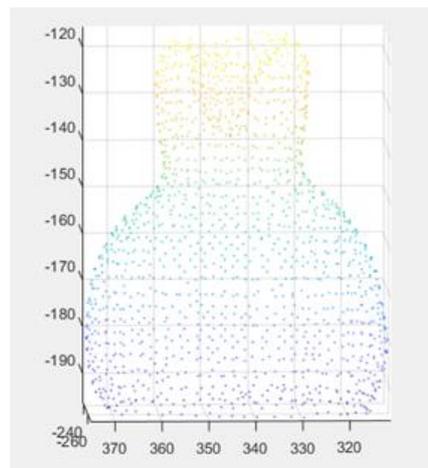


Figura III.6 Resultado del sub-muestreo de la nube de puntos filtrada.

Se pudo realizar en menor tiempo el proceso de mallado de la nube de puntos sub-muestreada sin notorias modificaciones a la geometría del objeto tridimensional. Se utilizó el método de triangulación de Delaunay y la generación del diagrama de Voronoi implementados en las funciones de Matlab, pero no se obtuvieron resultados de mallas bien definidas debido a que estos métodos solo procesan regiones convexas. Se recurrió al algoritmo realizado por Luigi Giaccari que usa la

técnica de pivoteo de pelota (*Ball-Pivoting Algorithm*) [31] para sólidos, ya que rellena huecos en la malla, ayudando a cubrir la parte inferior de la pieza que fue retirada. La Figura 3.7 muestra el modelo tridimensional resultado del mallado de la nube de puntos sub-muestreada.

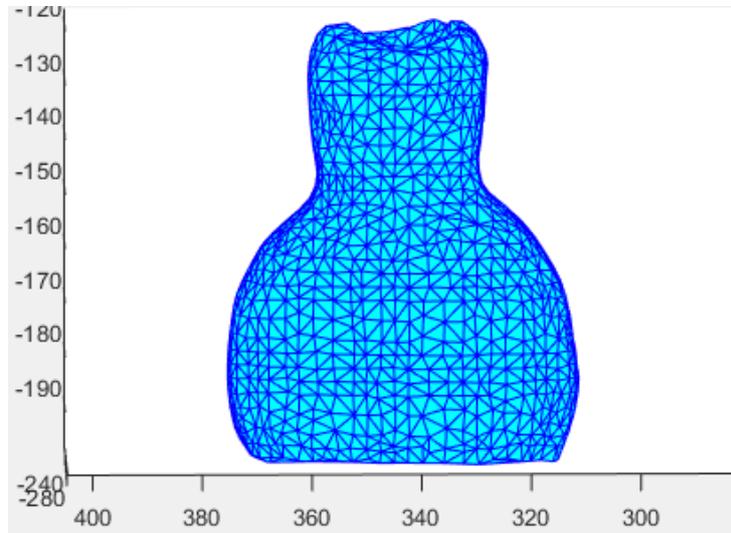


Figura III.7 Modelo tridimensional generado por el algoritmo de pivoteo de pelota.

Los datos generados por el proceso de mallado son:

- Lista de puntos con sus coordenadas XYZ (matriz de $n \times 3$).
- Lista de cómo fueron unidos los puntos para formar los triángulos o caras de la malla (matriz de $m \times 3$).
- La normal de cada uno de los triángulos o caras de la malla ($m \times 3$).

Estos datos fueron almacenados en un archivo con formato de estero-litografía o STL, que es uno de los formatos más aceptados por el software de las impresoras 3D, siendo necesario cambiar la información de ASCII a binario, ya que así se maneja la información en este tipo de archivos para obtener un menor tamaño.

1.11 Rebanado de la Nube de Puntos

Las impresoras 3D requieren de un software que les permita procesar los datos de los archivos STL y convertirlos en Código G, siendo éste el formato estándar usado por las máquinas de control numérico. Este software toma los datos del modelo tridimensional y los va seccionando o rebanando en uno de los tres ejes, lo que da como resultado rebanadas en dos dimensiones (X, Y), pero ordenadas de

acuerdo a su posición en el eje Z. Para cada rebanada se generan los comandos de movimiento del cabezal de la impresora y los códigos especiales para el control del extrusor, abanicos y otros elementos.

En este proyecto se desarrolló un proceso para tallado de madera con tecnología láser en el cual se lleva a cabo el rebanado de los datos tridimensionales directamente de la nube de puntos, sin necesidad de realizar el proceso de reconstrucción tridimensional y la generación del archivo STL.

Es necesario definir el concepto de *avance* en término de este proyecto, que es la forma análoga en que las impresoras 3D hacen una capa de impresión. Para determinar el *avance* se debe calcular el ancho de la rebanada de datos tridimensionales que se extraerá de la nube de puntos para proyectarlos sobre uno de los ejes y formar una imagen en dos dimensiones.

$$\text{Avance} = \text{ancho de rebanada}$$

$$\text{Número de rebanadas} = \left| \frac{\text{mínimo de la pieza} - \text{máximo de la pieza}}{\text{Avance}} \right| \quad 3.1$$

$$\text{Avance} = \left| \frac{\text{mínimo de la pieza} - \text{máximo de la pieza}}{\text{Número de rebanadas}} \right| \quad 3.2$$

El avance es importante y depende de la densidad de la nube de puntos. A mayor densidad, menor puede ser el ancho de la rebanada. Los valores mínimo y máximo se refieren a las coordenadas mínima y máxima sobre el eje que se va a seccionar o rebanar la nube de puntos. Si se toma como referencia el eje Z para obtener las rebanadas, entonces se seleccionan todos los puntos dentro del rango especificado por el avance, iniciando en la coordenada máxima en Z y avanzando rebanada por rebanada. Al tomar las coordenadas (X, Y) de los puntos dentro de la rebanada se obtiene su proyección sobre el plano Z, después se define el contorno de la rebanada uniendo los puntos en el borde obtenidos por medio de la función *boundary* de Matlab. La Figura III.8-1 muestra una rebanada de la nube de puntos tridimensionales, la Figura 3.8-2 muestra los puntos seleccionados en una rebanada proyectados sobre el plano Z y la Figura 3.8-3 muestra la unión de los puntos en la frontera obtenidos mediante la función *boundary*.

Si se desea que el avance sea a lo largo del eje Y , se tomarán los puntos con coordenadas (X,Z) para proyectarlos sobre el plano Y . Se proporciona una opción en el programa para poder seleccionar el avance a lo largo de cualquiera de los 3 ejes. Lo importante es seleccionar el eje de avance de acuerdo a la forma de la pieza, dejando la parte más angosta o pequeña en la parte superior y lo más grueso en la parte inferior, evitando superficies cóncavas que son difíciles de procesar.

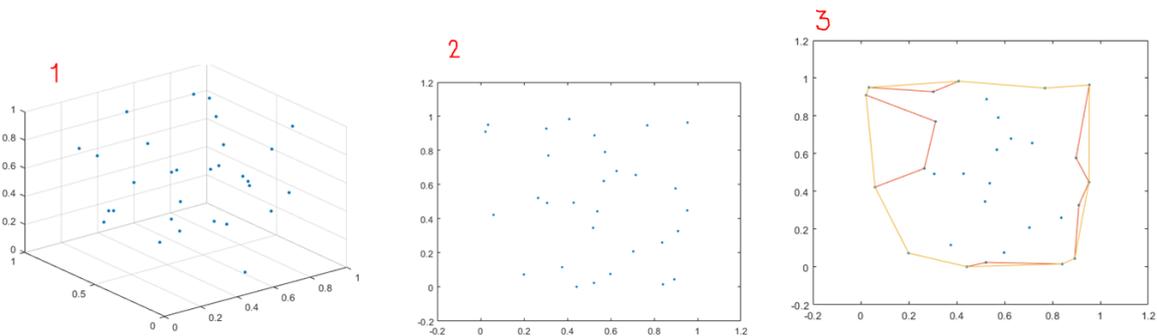


Figura III.8. 1) Rebanada de una nube de puntos, 2) puntos de la rebanada proyectados sobre el plano Z y 3) unión de puntos en la frontera.

Para no depender del factor de contracción en las piezas con huecos interiores o formadas solo por paredes de puntos se fueron acumulando los puntos de las rebanadas anteriores, así la densidad es mayor y la precisión en la definición del contorno es mejor. Una vez definido el contorno de los puntos en cada rebanada se procede a generar una imagen con fondo negro, el tamaño de esta imagen será definido por las dimensiones máximas de la pieza a tallar, es decir, el ancho y el largo del bloque de madera que será utilizado. Después se usa la función *patch* de Matlab para dibujar el contorno de la rebanada en color blanco sobre el fondo negro de la imagen. Se genera entonces una imagen binaria por cada rebanada. El código en Matlab del programa que realiza el rebanado de la nube de puntos para generar las imágenes binarias se incluye en el Anexo 2. Se pueden hacer cortes sobre el eje Y , contemplando la mitad de la pieza con un ciclo de corrección para evitar ciertos retrocesos en los puntos. En el Anexo 3 aparece una versión simplificada del programa para el rebanado de la nube de puntos.

En la Figura III.9 vemos las imágenes binarias generadas para las rebanadas en el eje Y de la pieza de la Figura III.6. El color oscuro indica la zona por la que se desea pase el láser encendido para que haga un desgaste progresivo hasta llegar a

la última rebanada que tiene predominantemente blanco, lo que significa que solo está procesando las orillas de la pieza, dándole profundidad.

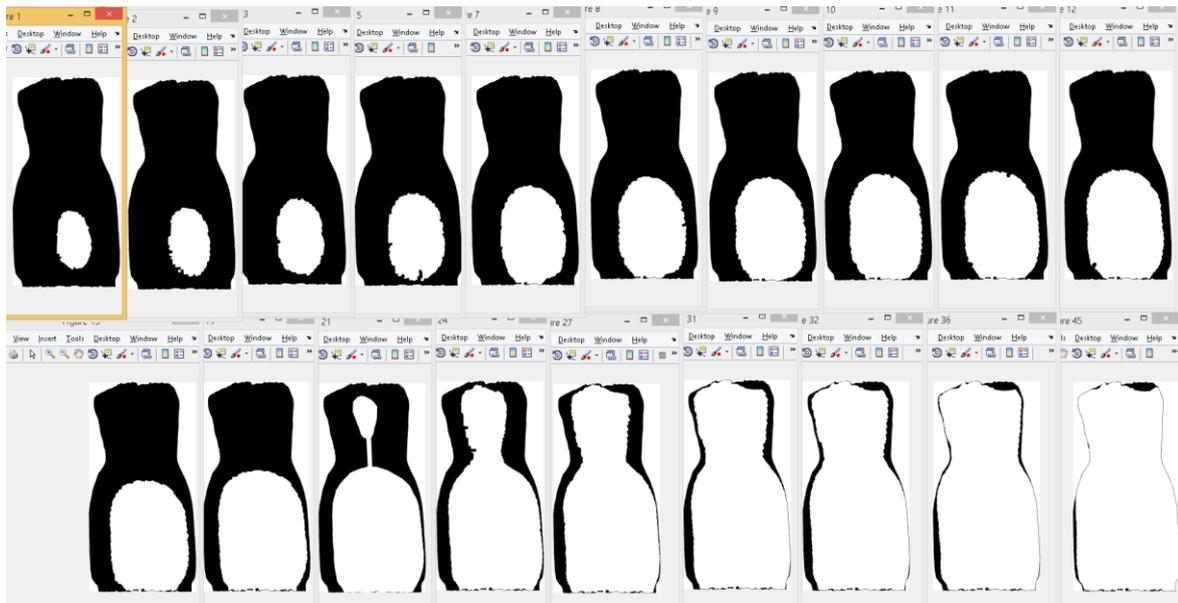


Figura III.9 Rebanadas de la nube de puntos a lo largo del eje Y de la pieza 1.

En la Figura III.10 vemos cómo se inicia el corte por la parte más alta sobre el eje Z de la nube de puntos de la pirámide generada por software (Figura III.1), donde la primera imagen (izquierda superior) muestra solo la punta de la pirámide y la última imagen (derecha inferior) muestra la base de la pirámide.

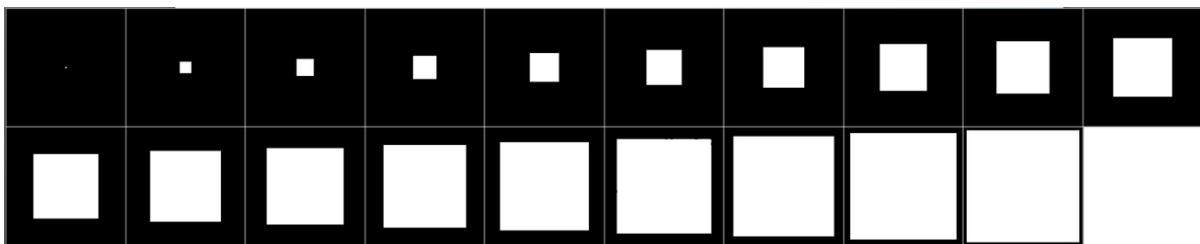


Figura III.10 Rebanado de la nube de puntos de una pirámide a lo largo del eje Z de la pieza 2.

1.12 Generación del Código G

El Código G es el nombre de un lenguaje de descripción de operaciones para máquinas de control numérico (CNC) que se almacena en formato texto, lo que permite la lectura y edición, usualmente ayudado de una aplicación que permite visualizar la instrucción y luego modificarla. Las aplicaciones de software utilizadas para la generación del Código G fueron *Inkscape* y *GRBL Laser*. Estas aplicaciones

toman las imágenes PNG generadas por las rebanadas de la nube de puntos y generan las trayectorias que guiarán al módulo láser en la máquina CNC por medio de Código G, estas trayectorias pueden ser modificadas de acuerdo a diferentes parámetros que pueden ser definidos dentro de la aplicación. *Inkscape* funciona con programas especializados para determinados tipos de hardware denominados extensiones, en este proyecto se utilizaron las extensiones desarrolladas por *Endurance* y por *J-Tech Photonics*. Estas permiten generar Código G con comandos específicos para el control de los módulos láser que fabrican.

Para generar el Código G en *Inkscape* es necesario hacer algunas modificaciones a la imagen PNG para poder usar alguna de las extensiones disponibles. En el caso de *Endurance* tiene la opción para generar Código G para grabar figuras o letras y programa el recorrido del láser solo por los contornos. Para grabar una imagen en la que se desea que el láser se encienda en las áreas negras y se apague en las áreas blancas, el programa coloca líneas a lo largo de toda la imagen, la separación entre líneas es especificada por el usuario, con más líneas se obtiene mayor precisión pero más tiempo de procesamiento. Estas líneas simulan la trayectoria que seguirá el módulo láser. En las transiciones de tono coloca ondas sinusoidales, cuadradas o triangulares de acuerdo a la selección del usuario. La figura 3.10 muestra el panel de selección de opciones de la extensión de *Endurance* para la generación de las líneas de trayectoria del módulo láser y la imagen pre-visualizada.

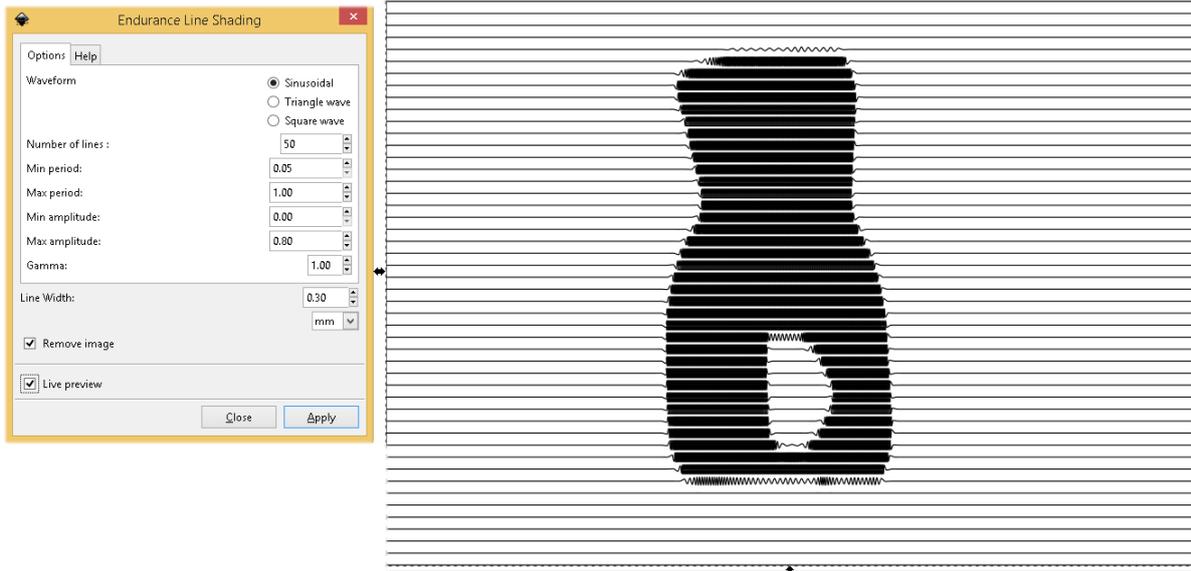


Figura III.11 Panel de opciones de la extensión de *Endurance* en *Inkscape* y pre-visualización de la imagen con las líneas de trayectoria del láser.

El código generado en *Inkscape* con la extensión de *J-Tech Photonics* realiza algo similar, es decir, las líneas generadas son recorridas por el láser, incluso cuando la imagen tiene áreas blancas, aunque no sea necesario que el láser pase, debido a que necesita regresar de nuevo a casi el mismo punto de donde se apagó y no generó ningún cambio. La diferencia es que la transición entre líneas se hace en forma triangular y no cuadrada como en la extensión del *Endurance*. Se genera un movimiento rápido hacia el otro extremo, sin embargo, aún sigue recorriendo las áreas blancas, aunque con mayor velocidad.

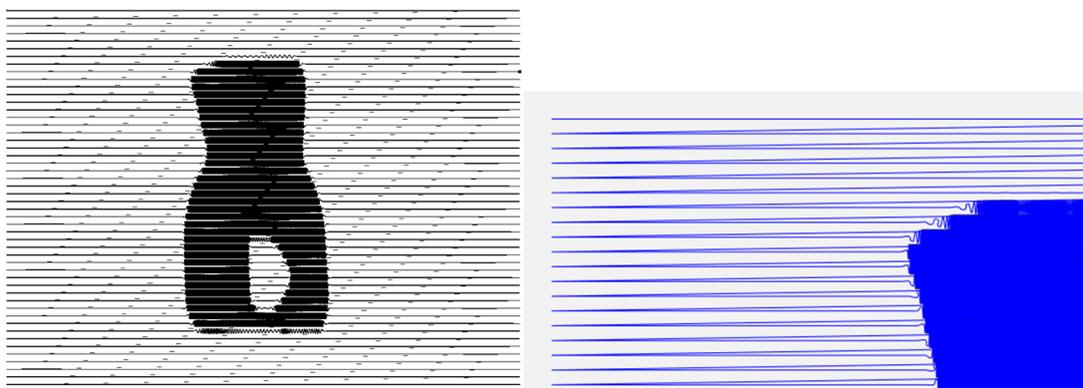


Figura III.12 Líneas de trayectoria del láser generadas con la extensión *J-Tech Photonics*.

La aplicación *GRBL Laser* es mucho más sencilla, solo se carga la imagen y se deben ajustar unos pocos parámetros. La desventaja es que este sistema está diseñado para máquinas que son exclusivamente cortadoras laser. En este proyecto

se utiliza una impresora 3D en la que se sustituyó el extrusor con el módulo láser, por lo que fue necesario hacer algunas adaptaciones al Código G generado. En la Figura III.13 se muestra la interfaz de usuario de la aplicación *GRBL Laser* con las dos ventanas que se abren después de cargar la imagen a procesar. En la ventana titulada “Importar Imagen Rasterizada” se pueden modificar todos los valores a consideración del usuario, solo la “Dirección” (señalada en amarillo) debe ser Horizontal. En la ventana titulada “Target image” la velocidad de tallado será determinada por la velocidad máxima de la máquina CNC. Si se requiere eliminar más material para obtener una mayor profundidad en el tallado de la pieza la velocidad tendrá que ser más lenta. En la misma ventana hay una opción para definir manualmente el tamaño final de la pieza. Si se usa la opción de tamaño automático (*autosize*) se deberán definir los puntos por pulgada (DPI) con los que la imagen fue guardada. Los mejores resultados se obtuvieron definiendo el tamaño manualmente. Al finalizar se guarda el archivo con el Código G.

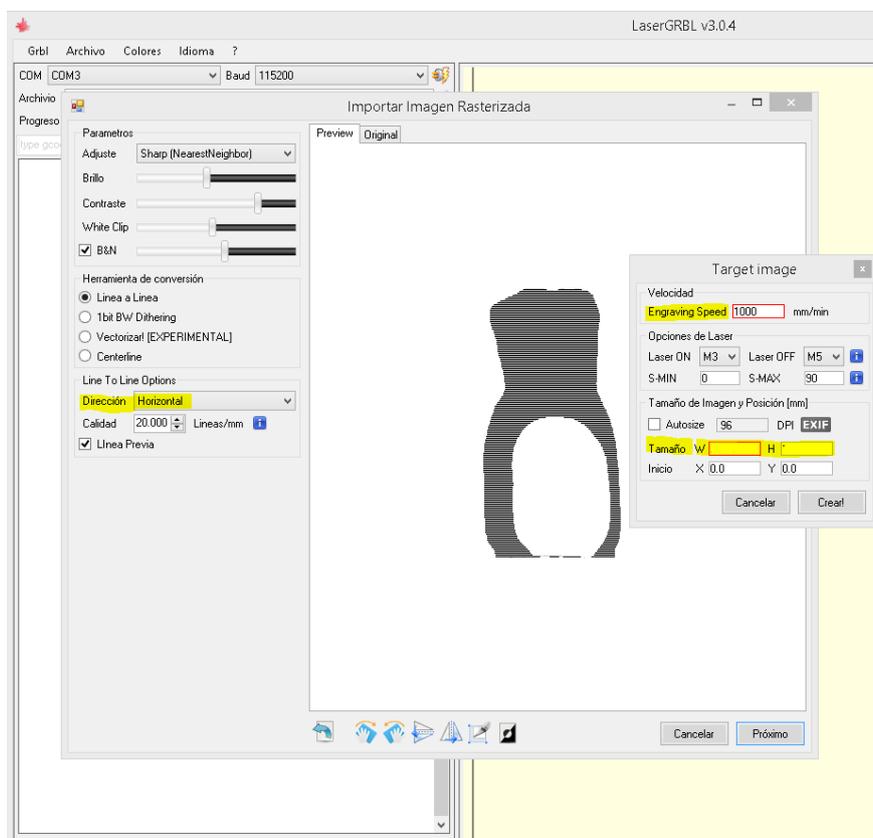


Figura III.13 Interfaz de usuario de la aplicación GRBL Laser.

1.13 Ajustes al Código G

Usar una impresora 3D sustituyendo el extrusor por el módulo láser requiere hacer adaptaciones a la tarjeta de control para generar la señal que encienda y apague el láser utilizando el puerto que controla el abanico del extrusor. Esta modificación obliga a realizar cambios en el Código G para enviar los comandos requeridos y activar el puerto utilizado para controlar el láser. Para hacer los ajustes al Código G se utilizó un script en lenguaje *Python* desarrollado y ejecutado desde el entorno de trabajo de *Visual Studio*. El código se puede ver en el Anexo 6.

A continuación se enlistan los datos que se deben de conocer:

1. Espesor del material con el que se va a trabajar.- Se refiere al tamaño físico del bloque de madera que se va a tallar, en este caso es el alto del bloque.
2. Distancia de trabajo del láser.- Distancia ideal del láser hacia la superficie del bloque de madera a tallar.
3. El formato del archivo de entrada.- Depende del programa usado para la generación del Código G, en este caso solo se consideran los archivos con extensión “.nc”.
4. El avance estimado.- Ancho de las rebanadas de la nube de puntos de la ecuación 3.2.
5. Repeticiones.- Se refiere al número de imágenes a procesar o nube de rebanadas de la nube de puntos.

Para calcular la distancia final del láser, es necesario tener presente las medidas (ver Figura III.14) del ancho del material, la distancia de trabajo del láser, el punto cero de la máquina CNC (H_{cero}) y la altura de la placa de aluminio (H_{placa}) colocada en la base de trabajo de la impresora 3D. Se aplica la siguiente fórmula:

$$inicZ = ancho\ del\ material + dist.\ trab.\ del\ láser - (H_{cero} + H_{placa}) \quad 3.3$$

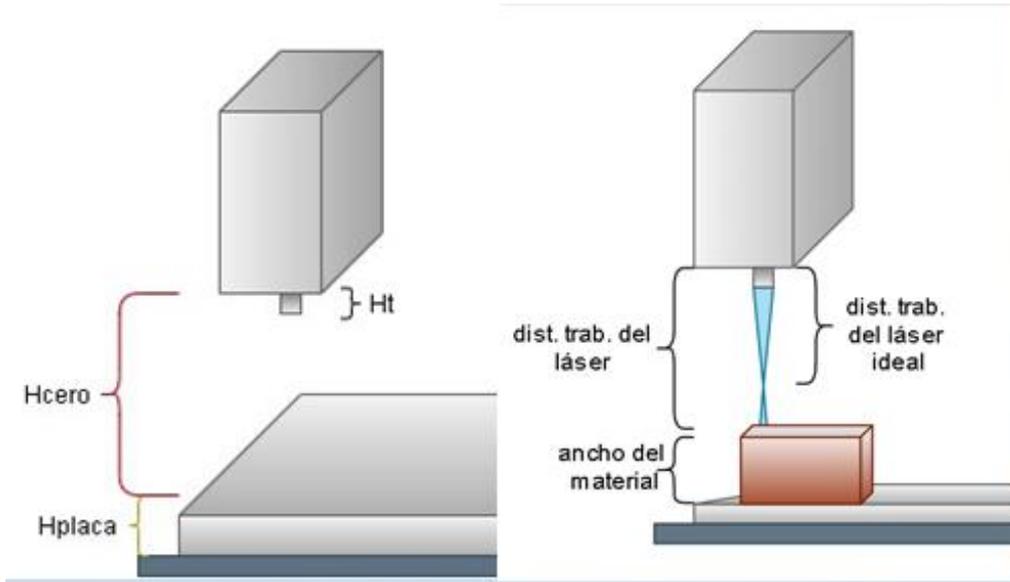


Figura III.14. Distancia de trabajo del módulo láser.

Aunque en la Figura III.14 la distancia *Hcero* del módulo láser a la superficie de la placa parece algo grande, es la distancia en que se ubica el cabezal cuando se le indica a la máquina CNC que se coloque en el punto cero. Fue necesario modificar la posición del interruptor de seguridad colocado en el límite del eje Z de la impresora 3D para establecer la distancia de trabajo adecuada.

La variable llamada *inicZ* de la ecuación 3.3 es el punto donde se ubica el cabezal del láser para que inicie con el desgaste de la madera y es muy importante debido a que si no se toman en cuenta las distancias mencionadas, el desgaste generado puede variar. En base a esta distancia y a las dimensiones del bloque de madera que se va a tallar, se va moviendo hacia abajo el cabezal sobre el eje Z de acuerdo al avance estimado. El avance real se estima de acuerdo al tipo de madera que se está usando, por lo que se recomienda realizar pruebas experimentales y así poder estimar los parámetros necesarios para la ecuación:

$$Repeticiones = \frac{(Hd)(Rp)}{Ap} \quad 3.4$$

Donde:

Hd = Altura deseada(mm)

Rp = Repeticiones de la prueba

Ap = Avance obtenido en la prueba(mm)

$$Avance\ Estimado = \frac{Hd}{Número\ de\ Rebanadas} \quad 3.5$$

Los valores calculados serán utilizados para modificar el Código G generado mediante los programas mencionados en la sección anterior. A continuación se describen los ajustes realizados al Código G original para que funcione de manera correcta de acuerdo a las características físicas del sistema CNC utilizado. La Figura 3.15 muestra las primeras líneas de un archivo de Código G.

```
G90
G0 X0 Y0
M3 S0
F1000
G0 X50.65 Y0.05 S0
```

Figura III.15. Primeras líneas del Código G original.

El comando G90 indica Modo Absoluto, es decir, que todos los valores de las coordenadas X,Y serán referidos al punto de origen. En el segundo renglón el comando G0 indica desplazamiento rápido y directo para mover el cabezal a la posición cero en X (X0) y cero en Y (Y0).

El comando M3 ordena encender el láser, pero también sirve para girar el eje en sentido horario. La velocidad de giro se especifica con el comando S0, esto significa que no comenzará a girar, pero si después se le da una velocidad, esta se establece por encima del cero y comenzará a girar; esto significa que la potencia del láser cambiará hasta que se especifique en una instrucción posterior. El comando F1000 indica Velocidad del movimiento del cabezal en milímetros por minuto y el comando de la última línea indica que se mueva rápidamente (G0) a la coordenada (X=50.65, Y=0.05) con velocidad de giro en cero (S0), es decir con el láser apagado. La Figura III.16 muestra los ajustes realizados a las líneas iniciales del Código G.

```

G90

G21
M106 S0
G0 Z79.8000

F1000
G0 X50.65 Y0.05
    
```

Figura III.16. Ajuste de las primeras líneas del Código-G.

Se utiliza el comando G21 para definir las dimensiones en milímetros, además se agrega el comando M106 que indica el encendido del abanico de la impresora, este es el puerto donde se conectó el láser, por lo tanto, significa que se enciende el abanico a una velocidad cero (S0), supliendo el comando M3.

El comando G0 Z79.8 mueve el cabezal a la coordenada especificada sobre el eje Z y es agregada por cada imagen generada, es decir, para cada rebanada de la nube de puntos. El programa que realiza los ajustes al Código G (ver Anexo 6) calcula la coordenada en Z en cada ciclo. Al inicio del código de cada rebanada se coloca la coordenada en Z que debe de tener el cabezal del láser y al final se asegura de apagar el láser para poder regresar hasta el inicio con el láser encendido.

La Figura III.17 muestra los ajustes realizados en las demás líneas del código G que se encargan de encender, apagar y mover el módulo láser, se observan tres líneas del código original y las siete líneas que las sustituyen en el código modificado. Para modificar la línea F es necesario que el comando M106 se encuentre en una línea nueva, es decir, que la línea inicie con esta instrucción y enseguida tenga la velocidad de giro, que indica la potencia del láser. La línea G del código original, indica que un movimiento a la coordenada (0.1,0.1) y luego apagar el abanico, mientras que en el código modificado primero se le da la instrucción de apagarse (M106 S0), luego con el comando G4 se detiene el movimiento y con la instrucción P200 se le indica que se mantenga detenido por 200 milisegundos, este tiempo es necesario para apagar el láser debido a que no lo hace de forma instantánea, finalmente se mueve a la coordenada (0.1,0.1). En la línea H se

modifica de manera similar a la línea F. Estos comandos para encender, apagar y mover el láser se repiten en el Código G de cada rebanada.

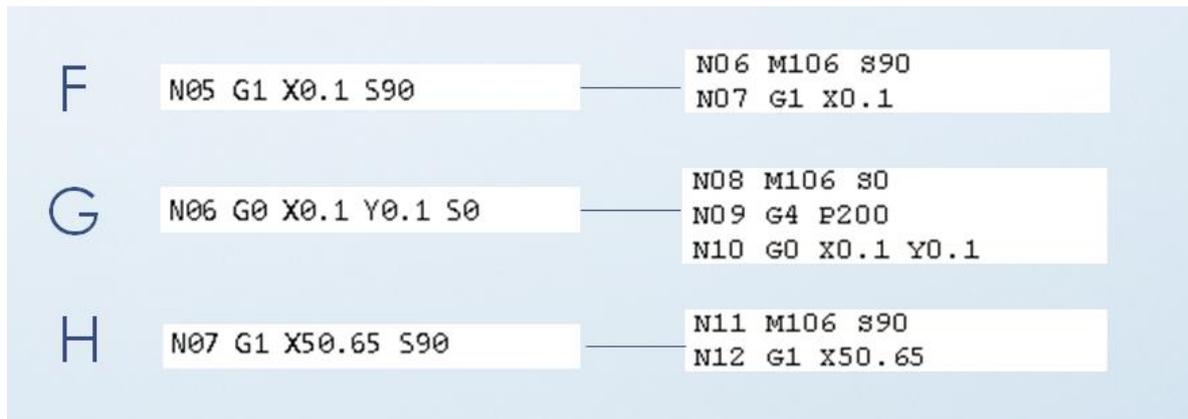


Figura III.17. Ajustes al Código-G para mover, encender y apagar el módulo láser.

Las líneas finales el Código G original en la Figura III.18 muestran el comando M5 que apaga el abanico y se sustituye por el comando M106 S0. Este comando permite encontrar el final del código de cada rebanada.

```
M5
G0 X0 Y0
```

Figura III.18. Líneas finales del Código-G original.

La Figura III.19 muestra el listado de archivos de Código G ajustado para cada rebanada de la pirámide generada por software que deben ser cargados uno a uno en el sistema CNC para tallar la pieza. En la parte inferior de la Figura III.19 aparece un archivo donde se une el código de todas las rebanadas para ser procesado por el sistema CNC en una sola carga.

 pruebapiramide_1M106	GCODE model file	57 KB
 pruebapiramide_2M106	GCODE model file	62 KB
 pruebapiramide_3M106	GCODE model file	64 KB
 pruebapiramide_4M106	GCODE model file	67 KB
 pruebapiramide_5M106	GCODE model file	70 KB
 pruebapiramide_6M106	GCODE model file	73 KB
 pruebapiramide_7M106	GCODE model file	75 KB
 pruebapiramide_8M106	GCODE model file	79 KB
 pruebapiramide_9M106	GCODE model file	81 KB
 pruebapiramide_10M106	GCODE model file	85 KB
 pruebapiramide_11M106	GCODE model file	42 KB
 pruebapiramide_12M106	GCODE model file	44 KB
 pruebapiramide_13M106	GCODE model file	45 KB
 pruebapiramide_14M106	GCODE model file	46 KB
 pruebapiramide_15M106	GCODE model file	47 KB
 pruebapiramide_16M106	GCODE model file	49 KB
 pruebapiramide_17M106	GCODE model file	50 KB
 pruebapiramide_18M106	GCODE model file	51 KB
 pruebapiramide_19M106	GCODE model file	53 KB
 pruebapiramide_20M106	GCODE model file	111 KB
 PruebaUnSoloArchivo_de_piramide_1M106	GCODE model file	11,958 KB

Figura III.19. Archivos de Código G ajustado para cada rebanada de la pirámide. En el último archivo se une el código de todas las rebanadas.

Capítulo IV. SISTEMA DE TALLADO CON TECNOLOGÍA LÁSER

Para la construcción del sistema de tallado con tecnología láser, se utilizó como base una impresora 3D marca *Creality* modelo *Ender 3 Pro*. Se eligió esta plataforma debido a que su estructura y sistema de control permiten hacerle modificaciones para sustituir el extrusor con un módulo láser.

El módulo láser que se usó es del tipo de semiconductor con una potencia de 10 Watts fabricado por la compañía *Endurance*. Este módulo contiene un diodo láser de los más potentes disponible en el mercado.

1.14 Sistema CNC

Se requiere el uso de un sistema CNC con movimiento en 3 ejes para el tallado de madera con tecnología láser de modelos tridimensionales de objetos de dimensiones reducidas. La mayoría de las máquinas de grabado láser disponibles en el mercado solo trabajan en 2D, son muy pocas las que mueven el cabezal hacia arriba o abajo y ese movimiento está limitado a no más de 5 centímetros.

Se decidió usar una impresora 3D como base del sistema CNC que llevará montado el módulo láser en su cabezal ya que estas impresoras tienen un rango más amplio de movimiento en el eje Z para mover el cabezal y variar la profundidad con que se devasta la madera. Otra ventaja del uso de una impresora 3D es que el diseño de estos sistemas tiene el respaldo tecnológico de la compañía que los fabrica, algunas de estas compañías proporcionan la información técnica de sus impresoras 3D, tanto mecánica como electrónica, lo que permite adaptarlos para diferentes usos. Entre diferentes opciones se seleccionó la impresora *Ender 3 Pro* ya que proporciona una buena calidad de impresión, bajo costo y es de código abierto. Una desventaja es que se vende semi-ensamblada (Figura 4.2), además de que la placa base se tiene que nivelar a mano.

El ajuste de la placa base o cama de trabajo es un proceso complicado debido a que tiene cuatro perillas que se giran manualmente, estas sobresalen de la parte inferior de la cama de impresión y se ajustan mientras se mueve el cabezal a lo largo y ancho de toda la cama.



Figura IV.1. Impresora Ender 3 Pro semi-ensamblada.

La tabla 4.1 muestra las especificaciones técnicas de la impresora 3D usada como sistema CNC. La figura 4.2 muestra el ensamble final de la impresora 3D.

Tabla IV.1 Especificaciones técnicas de la impresora Ender 3 Pro.

Característica	Especificación
Volumen de impresión	220 x 220 x 250 mm
Máx. velocidad de impresión	180-200 mm/s
Máx. resolución de capa	0.1 mm
Precisión de impresión	+/-0.1 mm
Conectividad	USB, tarjeta SD
Pantalla LCD	2in x 3in dicromática



Figura IV.2. Ensamble final de la impresora Ender 3 Pro.

1.15 Módulo Láser

El módulo láser usado en este proyecto tiene una potencia máxima de operación de 10 Watts y es fabricado por la compañía Endurance (figura 4.3). El diseño de la carcasa del módulo láser permite adaptarlo a diferentes máquinas, es fabricada en aluminio para disipar el calor generado por el diodo láser de alta potencia.



Figura IV.3. Módulo Láser de 10 Watts fabricado por Endurance.

El cabezal contiene un diodo láser fabricado por la compañía japonesa *Nichia*, ha sido probado por los fabricantes durante 72 horas continuas de uso, se estima una vida útil de 10 000 horas. La tabla 4.2 muestra las especificaciones técnicas del módulo láser.

Tabla IV.2 Especificaciones técnicas del módulo láser de 10 Watts de *Endurance*.

Característica	Especificación
Tipo de laser	De diodo, marca Nichia
Dimensiones	60 x 95 x 60 mm
Corriente	5-8 Amperes
Voltaje	9- 24Volts CD
Driver	Caja Externa de control
longitud de onda	445nm
Tamaño mínimo de punto láser	0,15 mm
Impulso (Pico) de potencia de salida óptica láser	10000mW \pm 20% (8 y 12 W)
Salida promedio de potencia óptica láser (CW)	8000mW \pm 15% (6.8-9.2 W)

La figura 4.4 muestra las partes del módulo láser, la carcasa del cabezal y la cubierta de la caja de control son de aluminio y cuenta con ventiladores en la parte superior. La caja de control y el cabezal se conectan entre ellas mediante cables que no se muestran. Los dos cables que se aprecian en la figura se conectan a la tablilla de control general de la máquina CNC que interpreta el Código G.

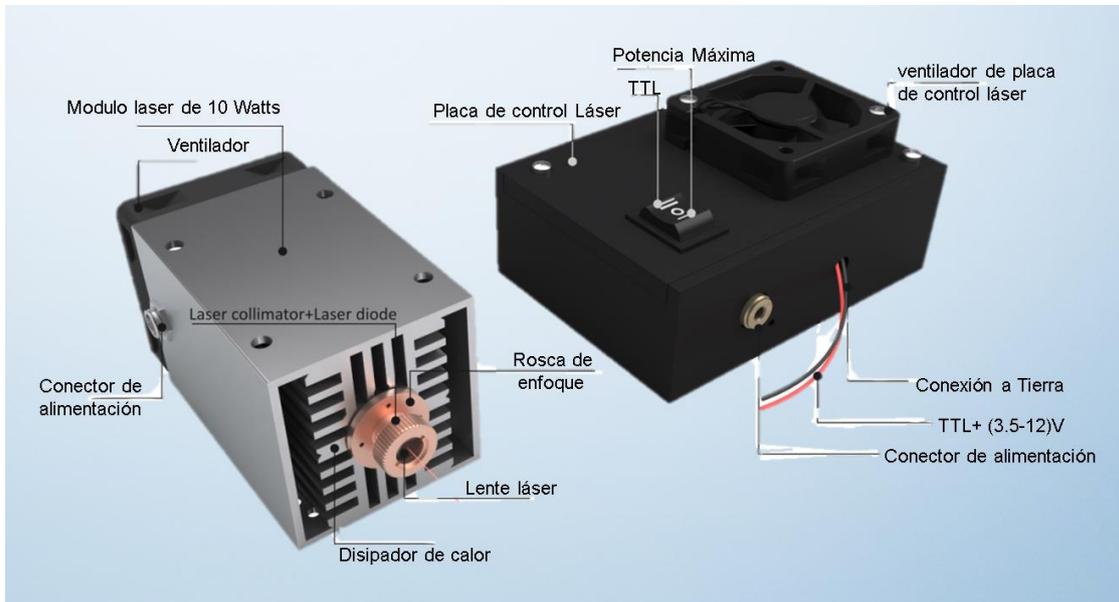


Figura IV.4. Cabezal del láser y tablilla o placa de control del láser.

1.16 Adaptación del Módulo Láser en la Impresora 3D

Los cables que llevan las señales eléctricas a la caja de control del módulo láser fueron conectados a la tablilla principal de impresora Ender 3 Pro mostrada en la figura 4.5. Se utiliza la señal del puerto de control del abanico del extrusor señalado por la flecha azul para lo que se tuvo que hacer una conexión en forma de Y para duplicar la señal.

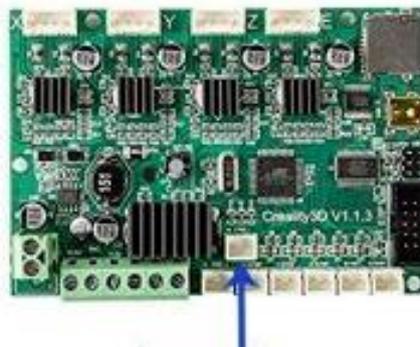


Figura IV.5. Tablilla principal de la impresora Ender 3 Pro.

Se fabricó un sistema de montaje intercambiable entre el cabezal del extrusor y el cabezal del módulo láser. Se utilizaron piezas impresas en 3D que permiten hacer un cambio sencillo entre ambos cabezales. Se imprimieron las piezas diseñadas por Jòn Schone obtenidas de la plataforma Thingiverse con el número de pieza 3369444 [32]. La figura 4.6 muestra piezas instaladas en la impresora 3D y los cabezales del extrusor y láser.

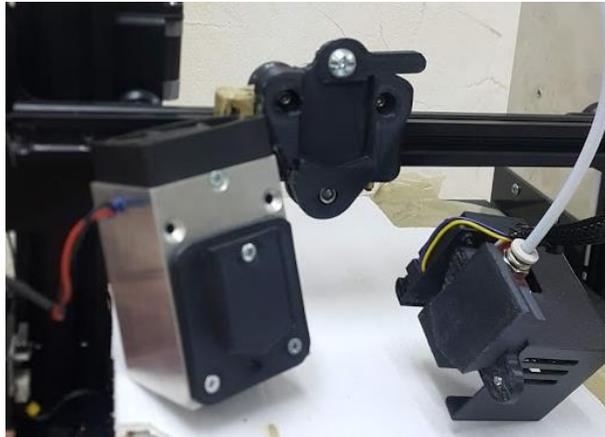


Figura IV.6. Cabezales del extrusor y láser con las piezas de montaje intercambiable.

Una vez montado el cabezal láser fue necesario calibrarlo. Para establecer la distancia óptima de trabajo se debe determinar la distancia a la cual el punto generado por el haz de luz láser tiene el diámetro mínimo alcanzado sobre la superficie de trabajo. Debido a la lente, el haz de luz láser toma una forma de X alargada, por lo que hay que encontrar la parte central de la X que es donde se concentra toda la energía. Se llevó a cabo un procedimiento de calibración que consistió en avanzar el láser sobre un bloque de madera inclinado y medir la distancia en el eje Z en el lugar donde el haz de luz láser tiene el mínimo diámetro al incidir sobre la superficie de la madera, como se muestra en la figura 4.7.

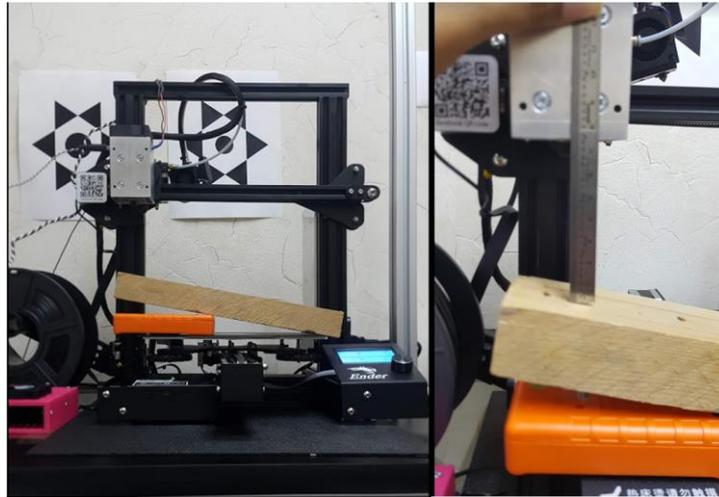


Figura IV.7. Procedimiento para determinar la distancia óptima de trabajo del cabezal láser.

Para finalizar la adaptación del módulo láser a la impresora 3D se instaló una placa de aluminio sobre la cama de trabajo de la impresora, lo que evita que el rayo láser dañe la superficie plástica que la recubre. El sistema de tallado de madera con tecnología láser completo se muestra en la figura 4.8.

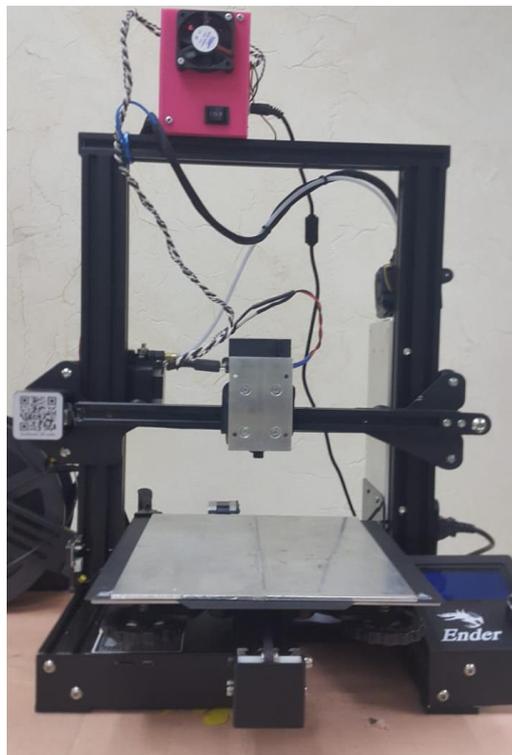


Figura IV.8. Sistema de tallado de madera con tecnología láser.

Capítulo V. EXPERIMENTOS DE TALLADO EN MADERA

La primera prueba realizada con el sistema de tallado en madera con tecnología láser desarrollado fue utilizando un archivo de Código G obtenido en línea que ya había sido utilizado para probar otros sistemas de grabado láser en madera. La figura 5.1 muestra el dibujo grabado en un trozo de madera del tipo MDF con el Código G de prueba, se puede observar que hay líneas que unen las letras y objetos que no deberían aparecer, esto se debe a que no se apaga el láser antes de mover el cabezal para iniciar un nuevo trazo.



Figura V.1. Dibujo grabado en madera tipo MDF con el Código G de prueba.

Primero se analizó el código buscando los comandos de encendido y apagado requeridos por el sistema desarrollado, específicamente el comando M106 que fue agregado al Código G de prueba ya que no tenía. Sin embargo, no se consiguió el resultado deseado. La guía de uso del módulo láser de *Endurance* indica que el encendido/apagado solo funciona a la mitad de su potencia, tomando 0 como mínimo y 255 como máximo, lo cual significa que la máxima potencia de trabajo es de 5 W y se traduce en un menor avance en el eje Z, pero aun así no se logró encender y apagar el láser durante el movimiento del cabezal, por lo que se siguieron haciendo pruebas hasta lograrlo. Esto fue a un 35.29% de su potencia que su equivalente 3.52 W utilizando el comando S90.

La figura 5.2 muestra el dibujo grabado con el Código G de prueba ajustado, logrando encender y apagar el láser entre los diferentes trazos, aunque aún se observan pequeñas líneas al terminar el trazo, esto se debe a que se mueve sin estar completamente apagado. Esto se solucionó agregando un tiempo de espera

después de la instrucción de apagado y bajando la velocidad de trabajo a 1000mm/min.



Figura V.2 Código con modificaciones. Se muestran fallas señaladas en rojo.

Asegurado el correcto funcionamiento del sistema de tallado en madera con tecnología láser se procedió a hacer las primeras pruebas con el Código G ajustado de las rebanadas de la nube de puntos de la pieza obtenida con el escáner tridimensional, los resultados se muestran en la figura 5.3.



Figura V.3. Primeras pruebas de tallado en madera.

El carbón es producto de la interacción del láser con la madera, es un efecto típico del proceso. Las zonas más oscuras muestran el resultado justo después de haber sido procesado. Las zonas que se ven más claras fueron limpiadas con una lija fina de para madera de grano 120.

La nube de puntos de la pieza obtenida con el escáner tridimensional es una figura irregular. Se realizó otra prueba utilizando la nube de puntos de la pirámide de

4 lados generada con Matlab con el objetivo de experimentar el tallado de superficies planas utilizando cuatro diferentes tipos de madera, los resultados se muestran en la figura 5.4.



Figura V.4. Pirámide de 4 lados tallada en a) madera de pino, b) madera de nogal, c) MDF y d) triplay.

La Figura V.5 muestra los parámetros utilizados en la configuración del programa GRBL Laser que fue utilizado para generar el Código G de las rebanadas de la nube de puntos de la pirámide y que después fue ajustado utilizando el programa desarrollado en Python. Los parámetros más importantes se encuentran resaltados en amarillo.

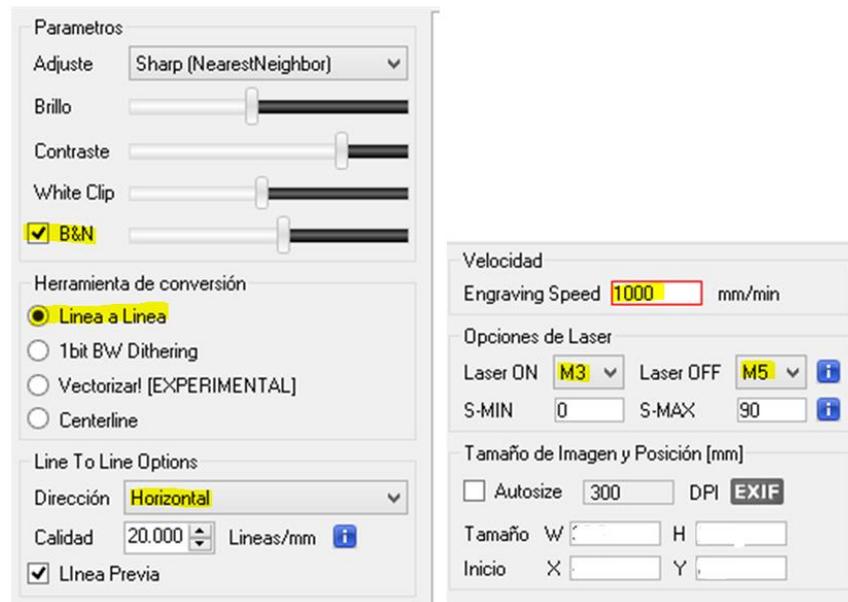


Figura V.5. Parámetros de configuración del programa *GRBL Laser*.

El código fue hecho para trabajar con la dirección “Horizontal”, aunque se realizó una prueba usando la dirección diagonal con los datos de la pirámide, los resultados se muestran en la Figura V.6, se puede observar que en una parte de la pieza los resultados son similares a los obtenidos en horizontal, pero la otra parte no se talló correctamente. Esta prueba muestra que los ajustes al Código G no funcionan de la misma manera.

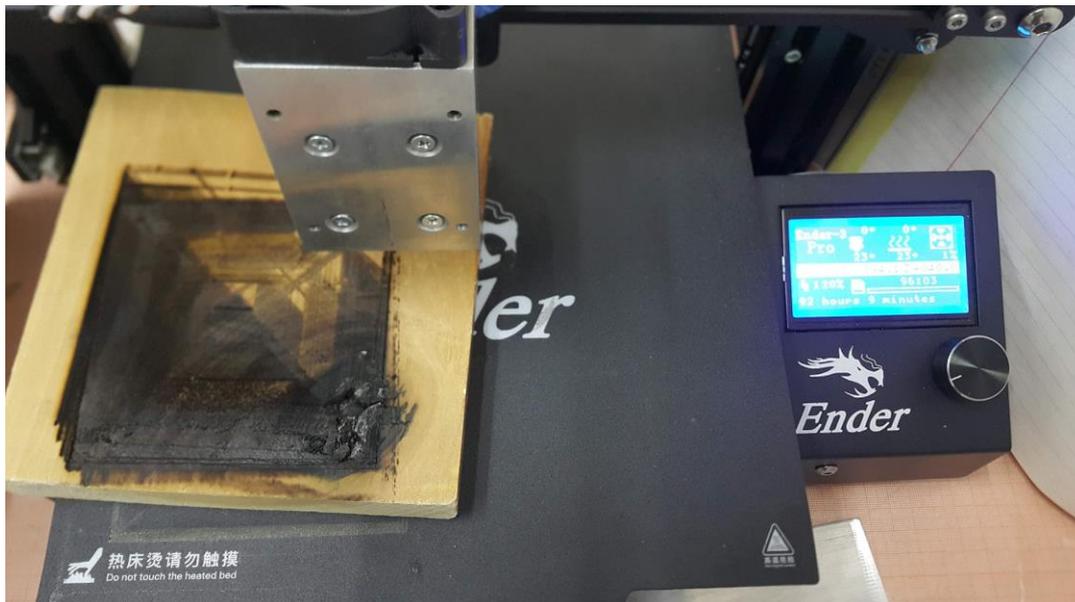


Figura V.6. Prueba de tallado con dirección “*Diagonal*”.

Las pruebas realizadas permitieron determinar que, a mayor velocidad de avance, se realiza un menor desgaste en la madera y si la velocidad es muy lenta el

desgaste se empieza a extender lateralmente (efecto no deseado), además de que el láser no alcanza a apagarse al final de un trazo. Otra situación que se puede advertir en la Figura V.6 es que la madera con la que se hizo la prueba no quedó bien sujeta a la base, lo que creó un desplazamiento y un desfase. Aunque el movimiento de la máquina parece ser mínimo, esto ocasiona una falla.

Se hizo otra prueba con los datos de la pirámide para tallar un bloque de madera de 20mm de altura, se crearon veinte rebanadas y se hicieron 10 repeticiones por rebanada. Los resultados se muestran en la Figura V.7, a la izquierda se muestra el objeto tallado con algunas partes que aún se encuentran adheridas al bloque de madera y a la derecha se muestra el objeto tallado después de haber sido separada del bloque de madera. Se perdió la parte inferior de la pirámide al separarla del bloque ya que estaba completamente carbonizada y se desprendió al tacto.

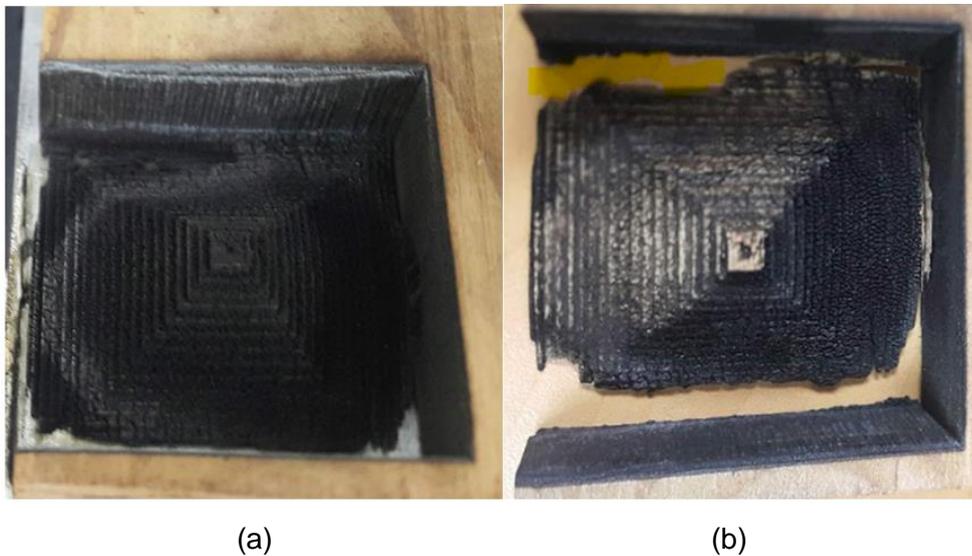


Figura V.7. Prueba de tallado de la pirámide en (a) la pieza terminada y en (b) la pieza terminada separada del bloque de madera.

La Figura V.8 muestra la pirámide separada del bloque de madera en donde se observan unos puntos que nos ayudan a entender el giro de la pieza y como cambia el espesor dependiendo de la dirección en que fue tallada por el láser.

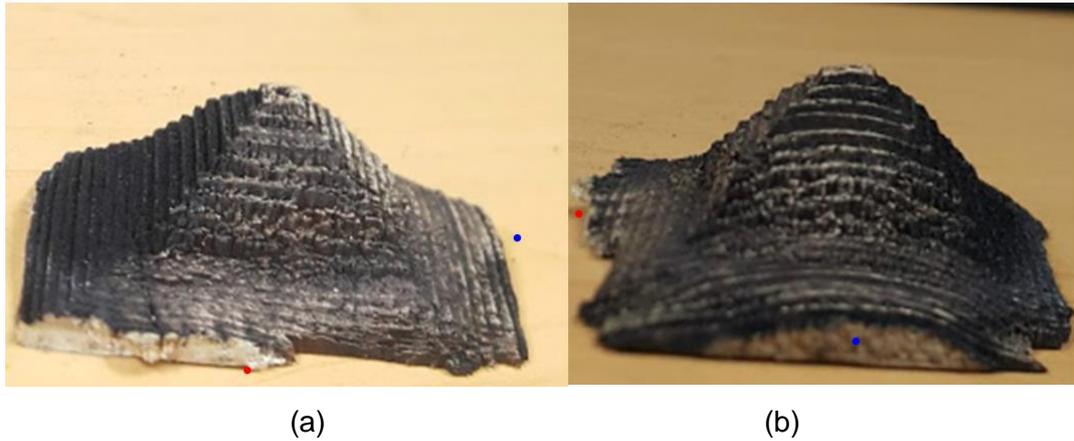


Figura V.8. Pirámide tallada limpiada con lija. (a) Vista frontal y (b) vista lateral.

Una prueba con resultados similares, pero menos notorios, fueron obtenidos tallando la pirámide en un bloque de madera de pino con veinte rebanadas y 4 repeticiones por rebanada. La Figura V.9 muestra la vista frontal en (a) con dos ligeras pendientes y la vista lateral en (b) con un arco hacia abajo.

El tiempo para tallar el objeto completo depende del número de rebanadas y el número de repeticiones que se debe realizar a cada rebanada para lograr el avance deseado. Para tallar la pirámide con 20 rebanadas y 10 repeticiones tomó 103 horas en realizarse. El tiempo para tallar la pirámide en un bloque de madera de pino con 20 rebanadas y 4 repeticiones por rebanada fue de 41 horas.



Figura V.9 Pirámide tallada en madera de pino. (a) Vista frontal y (b) vista lateral.

Capítulo VI. CONCLUSIONES

Se desarrolló el prototipo de un sistema de tallado en madera con tecnología láser de modelos tridimensionales de objetos de dimensiones reducidas. En la primera etapa se analizaron diferentes métodos para generar modelos tridimensionales de objetos, uno tiene que ver con el uso de un escáner tridimensional para capturar datos de objetos reales, en otro método se generan datos tridimensionales por medio de código para construir el modelo del objeto, también se pueden diseñar los modelos tridimensionales de objetos utilizando programas de CAD. Se desarrolló un procedimiento que permite procesar los datos del modelo tridimensional del objeto para ser tallado en madera. Este procedimiento consiste en procesar nubes de puntos tridimensionales, seccionándolas para obtener rebanadas de puntos y proyectarlas sobre uno de los planos obteniendo una imagen binaria por rebanada que sirve para determinar que partes de la madera se deben tallar en cada sección o rebanada. Las imágenes binarias se convierten en Código G que es procesado por un sistema CNC. En la segunda etapa se desarrolló el sistema de tallado de madera con tecnología láser, para esto se determinó utilizar una impresora 3D como sistema CNC y sustituir el extrusor con un módulo láser de 10 watts. Fue necesario adecuar el Código G obtenido de las rebanadas de las nubes de puntos para que pudiera ser procesado por el sistema de tallado en madera desarrollado.

Se realizaron experimentos de tallado de objetos en madera, se generó el modelo tridimensional de un objeto con superficies irregulares utilizando un escáner tridimensional, se obtuvieron las rebanadas de la nube de puntos y se realizaron pruebas con el sistema de tallado. Se realizaron otras pruebas con el modelo tridimensional de una pirámide de base rectangular generado por código, se obtuvieron las rebanadas de la nube de puntos y se generó el Código G para cada rebanada. Con estos datos se realizaron varias pruebas que permitieron determinar el ajuste de parámetros de funcionamiento del sistema de tallado utilizando diferentes configuraciones y tipos de madera. Los resultados obtenidos tuvieron variaciones debido a las propiedades isotrópicas de la madera que varían no solo de acuerdo al tipo, sino también por épocas del crecimiento y condiciones climatológicas. El ancho de la rebanada define el avance del cabezal sobre el eje Z, para lograr ese avance el láser debe realizar varias pasadas para devastar el

material necesario. El proceso repetitivo produce variaciones en el avance que dependen de la dirección en que se mueve el módulo láser. En trabajos futuros puede agregarse un sistema de visión que trabaje en conjunto con el módulo láser midiendo el avance logrado y comparándolo con el avance esperado.

También se debe considerar que el encendido y el apagado de este tipo de láser no es inmediato, tiene una curva de encendido en donde al principio se comporta como simple luz y posteriormente ya como láser. Esto se puede determinar ya que, en la vista frontal de la pirámide tallada, las líneas son más rectas donde el encendido y apagado se hacía con menos frecuencia, es decir, permanecía encendido a lo largo de todo el movimiento. Se observa una línea curva en la vista lateral, donde hay encendidos y apagados constantes. Se considera que una mejora en la generación de las rutas del Código G permitiría aumentar la potencia del láser para evitar estas fallas, además de lograr un mayor avance y evitar los tiempos de espera para apagarlo, esto disminuye el tiempo de trabajo requerido para tallar el objeto.

Capítulo VII. BIBLIOGRAFÍA

- [1] Y. T. Jun, V. H. Raja, "*Extracting geometric attributes directly from scanned data sets for feature recognition*", vol. 15, International Journal of Computer Integrated Manufacturing, 2002.
- [2] E. P. Hernández, "*Técnicas de relleno de huecos en superficies adquiridas mediante Escaners 3D*", España: Tesis Doctoral, UNED. Universidad Nacional de Educación a Distancia (España), 2011.
- [3] Y. Alshwabkeh, "*Integration of laser scanning and photogrammetry for Heritage documentation*", Tesis doctoral, Stuttgart: Institute for Photogrammetry at the University of Stuttgart, 2006.
- [4] F. García, "*Escaneo 3D en un campo de visión de 360° utilizando la técnica de proyección de franjas y un arreglo de espejos*", Tesis Maestría, León, Guanajuato: Centro de Investigaciones en Óptica, A.C., 2013.
- [5] L. F. A. Pérez, "*Adquisición de información de profundidad mediante la técnica "structured light, three phase-shift"*", Tesis de Maestría, Madrid: Universidad Complutense de Madrid, 2011.
- [6] M. Hansard, S. Lee, O. Choi y R. Horaud, *Time of Flight Cameras: Principles, Methods, and Applications*, Springer, Briefs in Computer Science, 2012.
- [7] A. M. Wolfgang BOEHLER, Artist, *3D SCANNING INSTRUMENTS*. [Art]. i3mainz, Institute for Spatial Information and Surveying Technology, FH Mainz, University of Applied Sciences.
- [8] R. H. Chandelier, *3D Modeling of Buildings: Outstanding Sites*, Great Britain, USA: Wiley & Sons, Inc., 2014.

- [9] M. O. Cantero, J. B. Rodríguez y J. R. Hernández, *Informática industrial*, vol. Colección Ciencia y Técnica nº 13, La Mancha: Servicio de Publicaciones de la universidad de Castilla- Lamanca, Colección Ciencia y Técnica nº 13, 1997.
- [10] Hilton A., Stoddart A.J., Illingworth J., Windeatt T., "Reliable surface reconstruction from multiple range images", *Lecture Notes in Computer Science*, vol. 1064, pp. 117-126, Springer, 1996.
- [11] Alias System Corp., *NURBS Modeling version 6*, Toronto, Canada: Alias, 2004.
- [12] W. Shengli and Z. Chongming, "NURBS surface generation by control points", *2011 IEEE 3rd International Conference on Communication Software and Networks*, Xi'an, 2011, pp. 544-547.
- [13] P. Theologou, I. Pratikakis y T. Theoharis, "A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation", *Computer Vision and Image Understanding*, vol. 135, Elsevier, 2015, pp. 49-82.
- [14] Y. Wang, D.-M. Yan, X. Liu, C. Tang, J. Guo, X. Zhang y P. Wonka, "Isotropic Surface Remeshing without Large and Small Angles", *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 7, pp. 2430-2442, 1 July 2019.
- [15] J. C. Ion, *Laser Processing of Engineering Materials, principles, procedure and industrial application*, Elsevier, 2005.
- [16] F. S. y. J. Manzanares, "El láser de semiconductor", *Revista Mexicana de Física*, vol. 37, no. 2, pp. 382-390, 1991.
- [17] MuzafferZeren, "A comparative study of natural stone modification by laser", *Journal of Materials Processing Technology J MATER PROCESS TECHNOL*, vol. 127, Elsevier, 2002, pp. 336-342.

- [18] C. Leone, V. Lopresto y I. Iorio, "Wood engraving by Q-switched diode-pumped frequency-doubled Nd:YAG green laser", *Optics and Lasers in Engineering*, Elsevier, 2008, pp. Volume 47, Issue 1, January 2009, Pages 161-168.
- [19] C. Wang y X. Zeng, "Study of laser carving three-dimensional structures on ceramics: Quality controlling and mechanisms", *Optics & Laser Technology*, vol. 39, Pages, Elsevier, 2007, pp. 1400-1405.
- [20] R. Paschotta, "Gain Media", *Encyclopedia of Laser Physics and Technology*, 1. edition October Wiley-VCH, 2008.
- [21] C. Pelosi, L. Calienno, D. Fodaroc, E. Borrelli, A. R. Rubinoc, L. Sforzini y A. L. Monaco, *An integrated approach to the conservation of a wooden sculpture representing Saint Joseph by the workshop of Ignaz Günther (1727–1775) Analysis, laser cleaning and 3D documentation*, Elsevier, 2016.
- [22] E. Meier, "The wood database 2008-2016", [En línea]. Disponible: <http://www.wood-database.com/european-lime/>. Último acceso: 10/10/2017.
- [23] R. B. Miller, "Structure of wood", *Wood handbook : wood as an engineering material.*, Madison, WI, USDA Forest Service, Forest Products Laboratory, General technical report FPL, 1999, p. 463.
- [24] K. Ogata, *Ingeniería de control moderna*, Edición en México: Pearson, 1998.
- [25] C. W. d. Silva, *Mechatronics an integrated approach*, Boca Raton, Florida: CRC PRESS, 2005.
- [26] A. G. Gonzalez, "Panamahitek.com", 2 Diciembre 2006. [En línea]. Disponible: <http://panamahitek.com/que-es-y-como-funciona-un-servomotor/> Último acceso: 17/01/2019.

- [27] S. Mkhemer, "3D printing", *3D printing operation and principle of working, as machine. Conference paper*, 2014.
- [28] Jones, R., Haufe, P., Sells, E., Iravani, P., Olliver, V., Palmer, C., and Bowyer, A, "RepRap - The Replicating Rapid Prototyper", *Robotica*, vol. 29, p. 177–191, 2011.
- [29] Shenzhen Creality 3D Technology Co., Ltd., "Creality", Shenzhen Creality 3D Technology Co., Ltd., 2014. [En línea]. Disponible: <https://creality.com/>. Último acceso: 15 /10/ 2019.
- [30] F. Reyes Cotés y J. Cid Monjaraz, ARDUINO - Aplicaciones en Robótica, Mecatrónica e Ingenierías, Primera ed., Alfaomega, 2016.
- [31] L. Giaccari, Surface Reconstruction From Scattered Points Cloud, [En línea] Disponible:<https://www.mathworks.com/matlabcentral/fileexchange/63730-surface-reconstruction-from-scattered-points-cloud>, MATLAB Central File Exchange. Último acceso: 16/12/2019.
- [32] J. Schone, "thingiverse.com" January 2019. [En línea]. Disponible: <https://www.thingiverse.com/thing:3369444>. Último acceso: 16/12/2019.

Anexo 1

LecturaDePieza_girado_recorte.m

%Este archivo contiene el procedimiento para hacer el giro de la pieza y recorte. Los valores de sel indican el área que deseamos mantener: sel*m es el valor menor y los de sel*M es el valor mayor

```
function giradoyRecorte(pieza,giroX,giroY,giroZ,selXm,selXM,selYm,
selYM,selZm,selZM)
%% lectura del archivo %%-----
ptCloud = pcread(pieza);
figure();

%% giro en el eje Y %%-----
matGiroY=[ cos(giroY)      0      sin(giroY)  0;
           0               1       0         0;
          -sin(giroY)      0       cos(giroY)  0;
           0               0       0         1];

tform = affine3d(matGiroY); %affine3d describe la transformación 3D
rígida.
ptCloudGiradaY = pctransform(ptCloud,tform);

%% giro en el eje Z %%-----
matGiroZ=[ cos(giroZ)  -sin(giroZ)  0      0;
           sin(giroZ)  cos(giroZ)  0      0;
           0           0           1      0;
           0           0           0      1];

tform = affine3d(matGiroZ);
ptCloudGiradaZ = pctransform(ptCloudGiradaY,tform);

%% giro en el eje X-----
matGiroX=[ 1      0      0      0;
           0      cos(giroX) -sin(giroX)  0;
           0      sin(giroX) cos(giroX)  0;
           0      0           0          1];

tform = affine3d(matGiroX);
ptCloudGiradaX = pctransform(ptCloudGiradaZ,tform);

%% seleccionar la parte que queremos modificar -----
xyz = ptCloudGiradaX.Location(:, :, 1);

j=1;
for i=1:size(xyz,1)
    if xyz(i,1)>selXm && xyz(i,1)< selXM&& xyz(i,2)> selYm && xyz(i,2)<
selYM && xyz(i,3)> selZm&& xyz(i,3)< selZM
        ptCloudN(j,:) = xyz(i,:);
        j=j+1;
    end
end
ptCloudN1= pointCloud(ptCloudN);
```

```
%% filtrado-----  
ptCloudOut = pcdenoise(ptCloudN1, 'NumNeighbors', 10, 'Threshold', 10);  
ptCloudOut1 = pcdenoise(ptCloudOut, 'NumNeighbors', 50, 'Threshold', 10);  
ptCloudOut5 = pcdenoise(ptCloudOut1, 'NumNeighbors', 50, 'Threshold', 10);  
  
xyz1 = ptCloudOut5.Location(:, :, 1);  
p=double(xyz1);  
save('mycopy_saquito_sinMuestre2o22-05-19.mat', 'p', '-v7.3');  
%With the -v7.3 flag you can store variables > 2GB with compression  
%Without the -v7.3 flag (e.g. if the default version is set to -v7 or  
lower) you have no (or less) compression, but cannot store large arrays
```

Anexo 2

RebanadoPTC.m

```

%Este archivo contiene la función que realiza el rebanado de la nube de
%puntos
%El avance se realiza sobre el eje Y, si se desea realizar las rebanadas
%sobre el eje X o el eje Z se deberá rotar la nube de puntos antes de
%procesarla con este programa
function rebanadoptc(pieza,factor,avance)

%% Carga la nube de puntos y la separa en x,y,z
load pieza; %pieza contiene los datos de la nube de puntos p

xdata = p(:,1);
ydata = p(:,2);
zdata = p(:,3);

i = max(xdata);
j = max(ydata);
k = max(zdata);

ii = min(xdata);
jj = min(ydata);
kk = min(zdata);
%figure();plot3(p(:,1),p(:,2),p(:,3),'.') %...para mostrar la nube de
puntos

%%
factor = 0.90; %%%***valor de 0 a 1, su programación por default es de 0.5
contador = 1;

%si se desea determinar el número de piezas, use esta sección____ %|
Npiezas= 100; %|
%avance=(abs(jj-j)/Npiezas); %|
% _____|

%si se desea determinar el grado de avance, habilite esta sección
Npiezas = floor(abs((jj-j)/avance)); %|
% _____|

%% la parte mas ancha de la pieza

imin = ( i );
imax = imin -( avance + 0.1 );
Base_MatrizMedia = p(p(xdata>=imin)& p(xdata<imax),:);%considerar todos los
valores(de las 3 dimensiones)entre imin & imax y guardarlos en una matriz
x = Base_MatrizMedia(:,1);
z = Base_MatrizMedia(:,3);
bordeBase = boundary(x,z,factor);
XbordeBase = x(bordeBase);
ZbordeBase = z(bordeBase);
matrizBordeBase = [bordeBase XbordeBase ZbordeBase];

%% Ciclo para analizar los puntos de la parte mas ancha de la pieza
% servirá para comparar las partes en izquierda o derecha para el eje de

```

```

% las x y poder compararlas.... el Valor_mitadX se considerara para todas
las partes
[tam,~] = size(matrizBordeBase(:,2));
Valor_mitadX = (ii + (i))/2;
l = 1;
o = 1;

for c = 1 : tam
    if matrizBordeBase(c,2) >= Valor_mitadX
        dermatrix(o,:) = matrizBordeBase(c,:);
        o = o + 1;
    elseif matrizBordeBase(c,2) < Valor_mitadX
        izqmatrix(l,:) = matrizBordeBase(c,:);
        l = l + 1;
    else
        warning('error en la recopilación de datos, favor de revisar');
    end
end

%% la parte a trabajar de la pieza

for contador = 1 : (double(Npiezas)/2)% &

    % A partir de la segunda iteración se agregan los valores anteriores y
los actuales en una sola matriz
    % para luego hacer todo el procedimiento para 'boundary'

    if contador==1

        jmax = j;
        jmin = jmax - (avance - 0.01); % cuando el count sale en cero,
hay que revisar estas condiciones (el signo antes del avance)
        MatrizNueva = (p(ydata>=jmin & ydata<jmax,:)); %toma todos los
valores
        x = MatrizNueva(:,1);
        z = MatrizNueva(:,3);
        bordeNow = boundary(x,z,factor);
        XbordeNow = x(bordeNow);
        ZbordeNow = z(bordeNow);
        MatrizAnterior = MatrizNueva;

    else

        j = j - avance;
        jmax = j;
        jmin = jmax - (avance-0.01);
        MatrizNuevaL = [(p(ydata >= jmin & ydata <
jmax,:));MatrizAnterior]; %toma todos los valores e incluye los anteriores
        x = MatrizNuevaL(:,1);
        z = MatrizNuevaL(:,3);
        bordeNow = boundary(x,z,factor);
        XbordeNow = x(bordeNow);
        ZbordeNow = z(bordeNow);
        MatrizAnterior = MatrizNuevaL;
    end
end

```

```

% ciclo corrección. Para evitar que puntos de "ruido" generen
imperfecciones-----
-----

matrizBordeNow = [bordeNow XbordeNow ZbordeNow];

[tam,~] = size(matrizBordeNow(:,2));% el dos indica el valor Horizontal
Valor_mitadX = (ii+ (i))/2;
l = 1;
o = 1;

for c = 1 : 1 : tam
    if matrizBordeNow(c,2) >= Valor_mitadX
        dermatrixNow(o,:) = matrizBordeNow(c,:);
        M = find(abs(dermatrix(:,3)-(dermatrixNow(o,3))) <= 10);
%dermatrix es donde se busca, dermatrixNow es el valor que buscamos y 0.15
es la tolerancia que le damos

        %aquí inicia el ciclo de corrección en X -----
-para derecha
        [a,~] = size(M);
% |
        for b = 1 : a
% |
            if dermatrix(M(b),2) < dermatrixNow(o,2)
% |
                dermatrixNow(o,2) = dermatrix(M(b),2);
% |
            end
% |
            if dermatrix(M(b),2) >= dermatrixNow(o,2)
% |
                continue
% |
            end
% |
        end
% |
        %aqui termina-----
-----
        o=o+1;

    elseif matrizBordeNow(c,2)<Valor_mitadX
        izqmatrixNow(l,:)=matrizBordeNow(c,:);
        N= find(abs(izqmatrix(:,3)-(izqmatrixNow(l,3))) <= 20);

        %aquí inicia el ciclo de corrección en X -----
-para izquierda
        [a,~] = size(N);
% |
        for b = 1 : a
% |
            if izqmatrix(N(b),2) < izqmatrixNow(l,2)
% |
                izqmatrixNow(l,2) = izqmatrix(N(b),2);
% |
            end
% |
            if izqmatrix(N(b),2) >= izqmatrixNow(l,2)
% |

```

```

                                continue
%   |
                                end
%   |
                                end
%   |
                                %aquí termina-----
-----
                                l = l + 1;
                                else
                                warning('error en la recopilacion de datos, favor de revisar');
                                end
                                end
                                end

%%

                                figure()
                                set(gca,'DataAspectRatio', [1 1 1],'Units','pixels','XColor',
'none','YColor','none');
                                daspect([1 1 1]);
                                patch(XbordeBase, ZbordeBase, 'k');
                                hold on
                                patch(XbordeNow, ZbordeNow, 'w','EdgeColor','none');%%fill(XbordeNow,
YbordeNow, 'w') es un equivalente que no mostró mejora
                                axis tight;

%% para guardar el resultado de las imágenes

                                saveas(gca,sprintf('saquito_%dY_%dF006D-
%davance%s.png',contador, (factor), (avance),datestr(now,'dd-mmm-yyyy') ));

                                end

                                %gcf es la figura justo anterior creada (graphic current figure)handle **
                                podemos cambiar el formato**

                                %para ponerle nombre a la figura, la identificare de la siguiente manera
                                %las primeras tres letras de la pieza
                                *_saq*_
                                %dos cifras para definir el número de arriba hacia abajo que tiene la
                                %pieza, se pondrá una Y de la sig. forma
                                *_00Y*_
                                %el factor de reducción usado en decimal ej. 0.5 o 0.1 (siempre indicando
                                el
                                %cero antes del punto) después de una F y sustituyendo el punto por un
                                guion *_0-1F*_
                                %el número de prueba realizado, con tres cifras
                                *_001*_
                                %caracteres especiales: D para uso de puntos denso
                                %
                                %           F para uso después del filtrado
                                %
                                %           S para sub-muestreo

```

Anexo 3

procesar_pc.m

```

% Este archivo contiene el código simplificado que realiza rebanado de la
%nube de puntos.
%El avance se realiza sobre el eje Z, si se desea realizar las rebanadas
%sobre el eje X o el eje Y se deberá rotar la nube de puntos antes de
%procesarla con este programa.
function procesar_pc(pieza, avance)
load pieza;
bx=[ptCloudA.XLimits(1) ptCloudA.XLimits(2) ptCloudA.XLimits(2)
ptCloudA.XLimits(1)];
by=[ptCloudA.YLimits(1) ptCloudA.YLimits(1) ptCloudA.YLimits(2)
ptCloudA.YLimits(2)];
z=ptCloudA.ZLimits(2);

%Realiza el rebanado de la pieza
i=1;
while(i)
    %Define la region de interes
    roi= [ptCloudA.XLimits;ptCloudA.YLimits; z-avance z];
    indices = findPointsInROI(ptCloudA, roi);
    %Extrae los puntos
    ptCloudB = select(ptCloudA,indices);
    %Proyecta los puntos sobre el plano Z
    x=ptCloudB.Location(:,1);
    y=ptCloudB.Location(:,2);
    %Define el contorno y genera la imagen binaria
    k=boundary(x,y,0.1);
    figure;
    patch(bx,by,'k');
    hold on
        1. patch(x(k),y(k),'w');
    axis off
    daspect([1 1 1]);
    set(gca,'DataAspectRatio',[1 1 1],'Units','normalized')
    %Guarda la imagen en format eps (Rasterizada)
    FileName=['imagen_',num2str(i)];
    saveas(gca,sprintf(FileName),'eps');
    %Convierte la imagen eps a png utilizando GhostScript
    command = ['"c:\Program Files\gs\gs9.27\bin\gswin64.exe" -dSAFER -
dBATC -dNOPAUSE -dEPCrop -r300 -sDEVICE=pngalpha -
sOutputFile=',FileName,'.png ',FileName,'.eps'];
    system(command);
    %Avanza la coordenada en Z al inicio de la siguiente rebanada
    z=z-avance;
    %Si llegó al limite en Z sale del ciclo
    if z<=ptCloudA.ZLimits(1)
        break
    end
    i=i+1;
end
end

```

Anexo 4

mo-gcodeModif6.py

```
# Adecuación del Archivo para la impresora Ender3pro y el láser
# Endurance 10W.
# Este programa adapta el código proveniente de "GRBL Laser" para una
# impresora que no acepta los comandos M3 y M5.
#-----LIBRERIAS-----
    from pygcode import Line
    import re
    import os
    import decimal
    import contextvars
    from decimal import Decimal, getcontext
#-----

    getcontext().prec = 6
    listado = []
    nombre = []
    guia_nombres = []
    print("-----\n \n \n")
    print("Asegúrese de colocar el archivo que va a correr en la carpeta
\ndonde se encuentran los archivos a trabajar, para evitar fallas")
    print("\n \n \n-----")
    path=(r"C:\Users\msaenz\Documents\python\helloWorld\Gcode-
python\gcodes\Piramide_unitaria\completo_piramide")
    mat_with = float(input('\n\tAncho del material en milímetros:___'))
    dist_Foc = float(input('\tLa distancia de trabajo del laser:___'))
    inicZ = Decimal(mat_with) + Decimal(dist_Foc)- Decimal(17.2)      #17.2
es la distancia que tenemos desde la maquina Z=0 de la placa a la orilla
del laser
    for file in os.listdir(path):
        if file.endswith( ".nc"):
            listado.append(os.path.join(file))
            listado.sort(key = len)
```

```

for i in (listado):
    nombre.append( i.replace('.nc',''))
tam=len(listado)
print("\n\n\t\t\t El número de elementos en la lista es de: ", tam)
print('\n\t\t\t Tu distancia inicial de trabajo será de:_' +str(inicZ))
print('\n\n\n')
rest = float(input('¿Cuál es tu avance en Z por cada capa? (en mm):
'))
AvanceZ=0
Rep = int(input('Veces que quieres que se repita tu instruccion  '))

for i in range(tam):
    file = open("prueba"+nombre[i]+"M106"+"gcode", "w")
    guia_nombres.append(nombre[i]+"M106"+"gcode")
    precompile_pattern = re.compile(r"(S0)")
    precompile_pattern2 = re.compile(r"S([0-9])([0-9])")
    precompile_pattern3 = re.compile(r"S([0-9])([0-9])([0-9])")
    if listado[i] == listado[0]: #para iniciar el avance en Z
        AvanceZ=inicZ

    completo=[]
    with open(listado[i] , 'r') as fh:
        for line_text in fh.readlines():
            line = Line(line_text)

            if re.search(r"(M3)",line.block.text):
                new_line = "G21\nM106 S0\nG0 Z"+str(AvanceZ)+'\n'
            elif re.search(precompile_pattern,line.block.text) is not
None:
                new_line = re.sub(precompile_pattern, r"\nM106" + "
S90", line.block.text)
            else:
                new_line = re.sub(precompile_pattern2 or
precompile_pattern3, r"\nM106" + " S0\nG4 P200", line.block.text)

```

```

file.write(new_line + '\n' )

if line.block.text == 'M5':
    file.write("M106 S0\n")
completo.append(new_line + '\n')

for i in range(1,Rep):
    for renglon in completo:
        file.write(renglon)

AvanceZ=Decimal(AvanceZ)-Decimal(rest)
file.close()

# ---pedir que el archivo se repita un determinado numero de veces----
# --- juntar los archivos anteriores en uno solo -----
one = str(input('¿Desea juntarlos todos en un solo archivo?
Si(1)/No(2) '))
if one == '1':
    cont=0
    contt=0

    file = open("PruebaUnSoloArchivo_de_"+ str(guia_nombres[0]),'w')
    for i in range(tam)
        precompile_pattern = re.compile(r"G0 X0 Y([0-9])([0-9])")
#\nG0 X0 Y3\nG21&"

        precompile_pattern2 = re.compile(r"G0 X0 Y0")

    with open(guia_nombres[i] , 'r') as fh:
        lineas = []
        for line_text in fh.readlines():
            line = Line(line_text)

            if re.search(precompile_pattern,line.block.text) is
not None:
                new_line =
re.sub(precompile_pattern,'',line.block.text)
                cont = cont + 1

```

```
        file.write(new_line + '\n' )
elif re.search(precompile_pattern2,line.block.text) is
not None:
        new_line  =  re.sub(precompile_pattern2,  ' ',
line.block.text)
        contt = contt + 1

        file.write(new_line + '\n' )
else:
        file.write(line.block.text +'\n')

file.close()

else:
    print('opcion no incluida')
```