

# RECONSTRUCCIÓN 3D DE OBJETOS SUMERGIDOS EN AGUA



**EDILSON QUIÑONEZ JIMENEZ  
HÉCTOR DANIEL VICTORIA PIZO**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL  
INGENIERÍA EN AUTOMÁTICA INDUSTRIAL  
POPAYÁN, FEBRERO DE 2013**

# RECONSTRUCCIÓN 3D DE OBJETOS SUMERGIDOS EN AGUA



**DOCUMENTO FINAL DE TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE  
INGENIERO EN AUTOMÁTICA INDUSTRIAL**

**EDILSON QUIÑONEZ JIMENEZ  
HÉCTOR DANIEL VICTORIA PIZO**

**DIRECTOR: ING. LEYDY VIVIANA MUÑOZ ROBLES**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL  
INGENIERÍA EN AUTOMÁTICA INDUSTRIAL  
POPAYÁN, FEBRERO DE 2013**

# Tabla de contenido

Pág.

<b>1. MARCO CONCEPTUAL.....</b>	<b>2</b>
<b>1.1. TÉCNICAS DE RECONSTRUCCIÓN 3D .....</b>	<b>2</b>
1.1.1. CLASIFICACIÓN DE TÉCNICAS DE RECONSTRUCCIÓN 3D .....	2
<b>1.2. FENÓMENO DE REFRACCIÓN DE LA LUZ .....</b>	<b>5</b>
<b>1.3. CALIBRACIÓN DE CÁMARAS.....</b>	<b>9</b>
1.3.1. MODELO <i>PIN-HOLE</i> .....	10
<b>1.4. PROCESAMIENTO DE IMÁGENES .....</b>	<b>11</b>
1.4.1. HISTOGRAMA.....	11
1.4.2. BINARIZACIÓN O UMBRALIZACIÓN.....	12
1.4.3. MORFOLOGÍA.....	13
<b>1.5. RENDERIZACIÓN .....</b>	<b>17</b>
<b>2. DISEÑO DE LA PLANTA Y CAPTURA DE IMÁGENES .....</b>	<b>19</b>
2.1. TÉCNICA SELECCIONADA.....	19
2.2. FENÓMENO QUE EXPERIMENTA UN OBJETO SUMERGIDO EN AGUA .....	21
2.3. CALIBRACIÓN DE LA CÁMARA: <i>HARDWARE - SOFTWARE</i> .....	23
2.4. PLANTA IMPLEMENTADA.....	31
2.5. CAPTURA DE IMÁGENES: <i>HARDWARE - SOFTWARE</i> .....	32
<b>3. PROCESAMIENTO DE IMÁGENES: ALGORITMOS IMPLEMENTADOS .....</b>	<b>39</b>
3.1. HISTOGRAMA.....	40
3.2. UMBRALIZACIÓN.....	42
3.3. SEGMENTACIÓN.....	44
3.4. ESQUELETIZACIÓN.....	50
<b>4. OBTENCIÓN DE LA NUBE DE PUNTOS Y RENDERIZADO .....</b>	<b>53</b>
4.1. CÁLCULO DE LAS COORDENADAS TRIDIMENSIONALES DEL OBJETO .....	53
4.2. RENDERIZADO.....	55
<b>5. ANÁLISIS DE RESULTADOS .....</b>	<b>61</b>
5.1. CALIBRACIÓN DE LA CÁMARA.....	61
5.2. PROCESAMIENTO DIGITAL DE IMÁGENES.....	62
5.2.1. CAPTURA DE IMÁGENES .....	62
5.2.2. UMBRALIZACIÓN DE IMÁGENES .....	63
5.2.3. SEGMENTACIÓN.....	63
5.2.4. ESQUELETIZACIÓN .....	63
5.3. CÁLCULO DE COORDENADAS ESPACIALES.....	64
5.4. RENDERIZADO.....	65
<b>6. TRABAJOS FUTUROS.....</b>	<b>66</b>
<b>7. CONCLUSIONES .....</b>	<b>67</b>
<b>8. REFERENCIAS .....</b>	<b>68</b>

## LISTA DE FIGURAS

	Pág.
Figura 1. Clasificación de técnicas de reconstrucción 3D .....	3
Figura 2. . Relación de triángulos semejantes en la técnica de triangulación .....	4
Figura 3 Refracción de un haz de luz.....	6
Figura 4. Un rayo de luz que atraviesa una lámina de vidrio .....	7
Figura 5. Imagen formada por efecto de la refracción de la luz .....	8
Figura 6. Observador visualizando un pez desde fuera del agua.....	9
Figura 7. Modelo Pin-hole.....	11
Figura 8. Histograma de una imagen .....	12
Figura 9. Proceso de binarización .....	13
Figura 10. a) Imagen de entrada A, b) imagen B o elemento estructurante, c) proceso de dilatación para dos pixeles ubicados en la frontera del objeto, d) proceso de dilatación para dos pixeles que no se encuentran en la orilla del objeto, e) imagen A dilatada con B donde los nuevos pixeles aparecen coloreados, f) resultado de A dilatada con B. ....	15
Figura 11. a) Imagen de entrada A, b) imagen B o elemento estructurante, c) dos pixeles que no pertenecen a A erosión B, d) dos pixeles que si pertenecen a A erosión B, e) A erosión B con pixeles eliminados coloreados de rojo, f) resultado de A erosión B.....	16
Figura 12. Proceso de esqueletización para una imagen.....	17
Figura 13. Técnica de triangulación para objetos sumergidos bajo el agua .....	22
Figura 14. Patrón de Calibración. Plano de ajedrez de 42 esquinas internas.....	26
Figura 15. Cámara a calibrar. LifeCam modelo VX-800.....	27
Figura 16. Muestra de 9 <i>frames</i> donde la cámara detectó las esquinas internas del patrón de calibración.....	29
Figura 17. Diagrama funcional del algoritmo de calibración de la cámara.....	29
Figura 18. . Imagen con distorsión radial y tangencial.....	30
Figura 19. Imagen rectificada.....	30
Figura 20. . a) Cubo de vidrio, b) mecanismo de escaneo, c) vista superior del interior de la planta, d) Parte delantera de la caja del componente electrónico, e) parte trasera del componente electrónico, f) vista externa de la planta sin tapas, g) tapa superior de la planta, h) tapa delantera de la planta, i) vista general de la planta. ....	32
Figura 21. Esquema del proceso de captura de imágenes .....	33
Figura 22. Diagrama del circuito para el control del sistema cámara láser .....	35
Figura 23. Objetos a ser reconstruido mediante la técnica planteada en este trabajo, a) objeto gato, b) objeto esfera, c) objeto rana, d) objeto tapa. ....	36
Figura 24. Perfiles del objeto gato capturados en el proceso de escaneado. ....	37
Figura 25. Perfiles del objeto rana capturados en el proceso de escaneado. ....	37
Figura 26. Perfiles del objeto tapa capturados en el proceso de escaneado. ....	38
Figura 27. Perfiles del objeto esfera capturados en el proceso de escaneado. ....	38
Figura 28. Etapas del proceso de abstracción de información de las imágenes.....	39

Figura 29. Canales de una imagen, <i>a</i> ) Canal rojo de la imagen, <i>b</i> ) Canal verde de la imagen, <i>c</i> ) Canal azul de la imagen.....	40
Figura 30. <i>a</i> ) Imagen de entrada, <i>b</i> ) histograma de la imagen.....	41
Figura 31. <i>a</i> ) Imagen de entrada, <i>b</i> ) imagen umbralizada.....	42
Figura 32. Imagen de entrada, <i>b</i> ) imagen umbralizada.....	43
Figura 33. <i>a</i> ) Imagen de entrada, <i>b</i> ) imagen umbralizada.....	43
Figura 34. Diagrama funcional del algoritmo de umbralización.....	44
Figura 35. Imagen con objetos ficticios.....	45
Figura 36. Zona definida para recorrer la imagen.....	46
Figura 37. Imagen con contorno generado.....	46
Figura 38. Imagen sin ruido.....	47
Figura 39. Diagrama funcional del algoritmo de segmentación.....	48
Figura 40. <i>a</i> ) Imagen umbralizada, <i>b</i> ) imagen segmentada.....	48
Figura 41. <i>a</i> ) Imagen umbralizada, <i>b</i> ) imagen segmentada.....	49
Figura 42. <i>a</i> ) Imagen umbralizada, <i>b</i> ) imagen segmentada.....	49
Figura 43. Diagrama funcional del algoritmo de esqueletización.....	51
Figura 44. <i>a</i> ) Imagen de entrada, <i>b</i> ) imagen segmentada, <i>c</i> ) imagen esqueletizada.....	52
Figura 45. <i>a</i> ) Imagen de entrada, <i>b</i> ) imagen segmentada, <i>c</i> ) imagen esqueletizada.....	52
Figura 46. <i>a</i> ) Imagen de entrada, <i>b</i> ) imagen segmentada, <i>c</i> ) imagen esqueletizada.....	52
Figura 47. Perfil al que se le va a calcular las coordenadas de sus puntos.....	54
Figura 48. Diagramas funcionales de los algoritmos desarrollados.....	54
Figura 49. Importación de la nube de puntos desde Meshlab.....	56
Figura 50. Vista de la nube de puntos desde Meshlab para el objeto gato y el objeto tapa, respectivamente.....	56
Figura 51. Algoritmo VCG desde Meshlab.....	57
Figura 52. Resultado de aplicar el algoritmo VCG para el objeto esfera, el objeto rana y el objeto gato.....	58
Figura 53. Resultado de aplicar el renderizado al objeto tapa.....	58
Figura 54. Resultado de aplicar el renderizado al objeto rana.....	59
Figura 55. Resultado de aplicar el renderizado al objeto gato.....	59
Figura 56. Resultado de aplicar el renderizado al objeto esfera.....	60

## **ANEXOS**

**ANEXO A.** Resultados obtenidos en la etapa de calibración de la cámara

**ANEXO B.** Algoritmos de renderizado.

**ANEXO C.** Análisis DAFO para escoger la técnica de reconstrucción 3D.

**ANEXO D.** Algoritmos desarrollados.

**ANEXO E.** Descripción del funcionamiento del Algoritmo de Zhang-Suen

**ANEXO F.** Validación del sistema hardware software implementado.

## INTRODUCCIÓN

Actualmente se encuentran muchos desarrollos de sistemas hardware/software que permiten reconstruir objetos en 3D en la superficie [1], [2], sin embargo no es común encontrar soluciones que permitan reconstruir objetos sumergidos en agua y aunque se conocen algunas soluciones comerciales [3] que permiten realizar el levantamiento de información, recolectar los datos, procesarlos y finalmente obtener un modelo en 3D de objetos en el agua, estas soluciones representan costos muy altos y sus aplicaciones están destinadas a inspecciones submarinas y estudios oceanográficos.

Es necesario buscar alternativas de sistemas de reconstrucción 3D que puedan ser implementadas en el medio acuático y que reduzcan los costes de su desarrollo.

En Colombia los únicos grupos de investigación que están trabajando en el tema de reconstrucción de objetos sumergidos en agua y que gestionan proyectos relacionados, son el grupo de Control, Comunicaciones y Diseño Naval, perteneciente a la Escuela Naval Almirante Padilla y el grupo Programa de Diseño e Ingeniería Naval "PRODIN", donde el enfoque especial está dirigido a la realización de inspecciones de navíos sobre el agua. Sin embargo no se han obtenido resultados definitivos.

De acuerdo a esto, se hace necesario diseñar e implementar un sistema hardware/software de bajo costo capaz de realizar la reconstrucción de objetos sumergidos en agua, donde el mayor problema a enfrentar consiste en que los sistemas de visión acuáticos tienen que hacer frente a una serie de factores que dificultan la recolección de información, tales como: las condiciones dinámicas de iluminación del ambiente, los fenómenos físicos que ocurren en el agua, la configuración y utilización de los instrumentos que conforman el sistema. Factores como la luz del ambiente, la refracción y la profundidad, hacen que las imágenes obtenidas en el agua sean más difíciles de procesar [4], [5].

Otro factor importante es la selección de la técnica que el sistema implementará para obtener la información de las coordenadas espaciales del objeto a reconstruir [6], [7], [8].

En el presente trabajo se hace un estudio de toda la información necesaria para entender el funcionamiento de los sistemas de reconstrucción 3D y se presenta una solución de bajo costo que permite realizar el proceso de reconstrucción 3D de objetos sumergidos en agua.

# 1. MARCO CONCEPTUAL

## 1.1. TÉCNICAS DE RECONSTRUCCIÓN 3D

Las técnicas de reconstrucción 3D son un conjunto de procedimientos sistemáticos cuyo objetivo es obtener una representación tridimensional de objetos físicos. Hoy en día existen sistemas y/o instrumentos que aplican estas técnicas y que permiten obtener diversa información y modelos 3D de objetos, algunos de estos son: los escáneres digitales, que pueden representar modelos CAD (Diseño Asistido por Computador), los sonares y ecosondas, que obtienen información completa del océano (forma, textura, tamaño, color, etc.), la Kinect de *Microsoft* que realiza la reconstrucción facial del ser humano, etc.

Estos sistemas de adquisición de la geometría y el color de objetos basados en la Visión por Computador se han desarrollado rápidamente, debido a la gran variedad de aplicaciones en las que estas informaciones son necesarias. Las nuevas técnicas de reconstrucción 3D permiten capturar los puntos de medida con mayor precisión y velocidad, pudiéndose emplear sobre distintos tipos de objetos, de distintas dimensiones, geometrías y texturas [9].

### 1.1.1. CLASIFICACIÓN DE TÉCNICAS DE RECONSTRUCCIÓN 3D

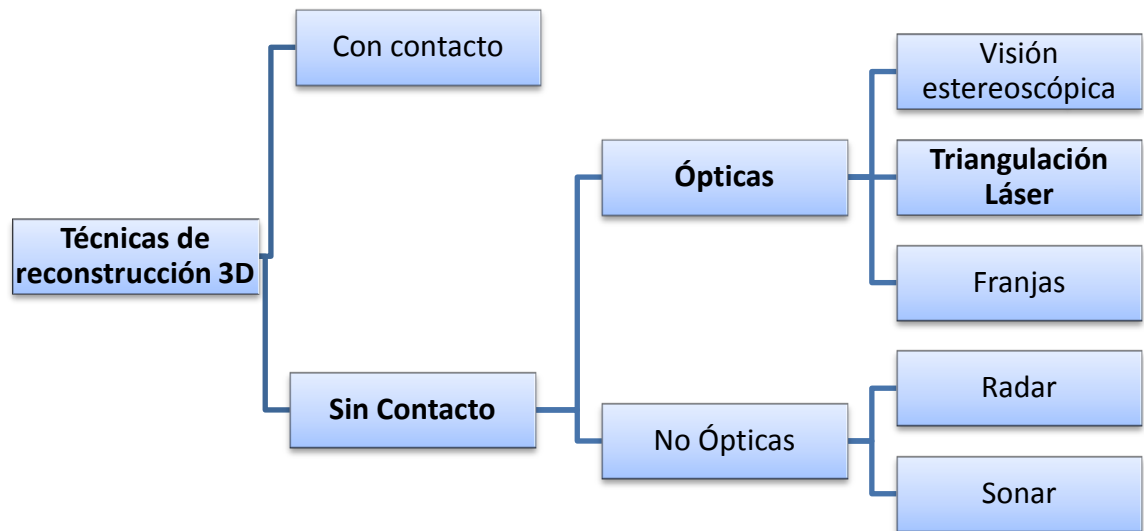
Existen distintas técnicas de reconstrucción 3D que se pueden dividir en dos grandes grupos: técnicas con contacto o sin contacto con el objeto a reconstruir. Las técnicas con contacto son las mas antiguas y principalmente se emplean para la verificación dimensional de piezas industriales (control de calidad), como ejemplo de uso de estas técnicas está el laminado y los brazos articulados. Las técnicas sin contacto se dividen en no ópticas y en ópticas. Las técnicas no ópticas utilizan el sonido como medio de transmisión y recepción de la información a reconstruir, en el caso de los sonares y/o ecosondas, ó utilizan las señales electromagnéticas si se trata de los radares. Las técnicas ópticas funcionan bajo escenarios donde la iluminación del ambiente es controlada. Además dependen en gran medida de la fuente de luz (láser) que se utiliza para iluminar la escena. El sistema mínimo utilizado para aplicar técnicas de reconstrucción 3D ópticas consta de una cámara y de un láser. Entre estas técnicas encontramos la visión estereoscópica, la triangulación láser y las franjas. En la figura 1 se detalla la clasificación de estas técnicas [10].

La visión estereoscópica emula el sistema visual humano el cual permite obtener información de profundidad mediante la fusión de dos escenas monoculares, que son las escenas que captan cada uno de los ojos. Este sistema permite al ser humano "ver en 3D" y es conocido como principio de visión estereoscópica. Así pues, dicho principio es aplicado en visión artificial utilizando dos cámaras las



cuales permiten tener dos puntos de vista de un mismo objeto para encontrar las coordenadas tridimensionales [7], [11].

Figura 1. Clasificación de técnicas de reconstrucción 3D



La triangulación láser utiliza una cámara y un láser. Esta técnica basa su funcionamiento en el cálculo de triángulos semejantes entre los elementos sensor óptico (cámara), emisor de luz (láser) y objeto de la escena a reconstruir [12]. El cálculo de las coordenadas tridimensionales del objeto se realiza de acuerdo a las características y posición de cada uno de los elementos del sistema. Generalmente se establece una relación de semejanza entre dos triángulos que comparten un vértice común que es el punto focal de la cámara. En la figura 2 se puede apreciar de manera visual el principio de funcionamiento de esta técnica. El punto de la escena  $P_w(X, Y, Z)$  se proyecta en la imagen como el punto  $P_u(x, y)$ .

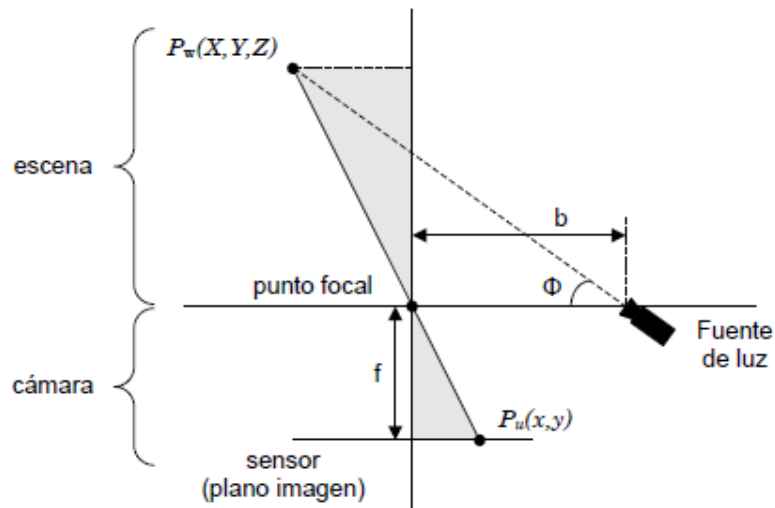
Gracias a la relación descrita en las ecuaciones (1), (2) entre los triángulos semejantes, se pueden establecer las ecuaciones (3), (4), (5) para calcular las coordenadas  $(X, Y, Z)$  del punto  $P_w$  que pertenece al objeto de la escena.

La formación de una imagen se hace a través de un proceso proyectivo del mundo real tridimensional al sub espacio bidimensional de la misma, en este proceso desaparece una dimensión, la de la profundidad. Para recuperar esta dimensión,

es necesario conocer la relación entre las coordenadas de la imagen y las del objeto a reconstruir [6], [13].

La técnica de reconstrucción 3D por franjas es una variante de la técnica triangulación láser debido a que no se necesita recorrer con el haz (línea) de luz al objeto a reconstruir, por el contrario emite un conjunto de líneas de luz sobre el objeto iluminándolo simultáneamente por completo. Este patrón se emite en primer lugar sobre un plano de referencia y posteriormente sobre el objeto, de tal forma que al hacer una diferencia entre ambas imágenes (la del plano patrón y la del objeto) se pueda extraer la información necesaria para el cálculo de la profundidad [14].

Figura 2. . Relación de triángulos semejantes en la técnica de triangulación



Fuente: [12]

$$x * Z = X * f \quad (1)$$

$$y * Z = Y * f \quad (2)$$

$$X = \frac{b}{f * \cot(\Phi - x)} * x \quad (3)$$

$$Y = \frac{b}{f * \cot(\Phi - x)} * y \quad (4)$$

$$Z = \frac{b}{f * \cot(\Phi - x)} * f \quad (5)$$

## 1.2. FENÓMENO DE REFRACCIÓN DE LA LUZ

Cuando un haz de luz se emite de un medio transparente a otro, parte de él se refleja y otra parte pasa al segundo medio, así como se muestra en la figura 3. La parte de luz que pasa al segundo medio se dobla en la frontera cambiando su dirección. A este fenómeno se le denomina refracción. Una particularidad de este hecho, es que tanto el rayo incidente, el rayo reflejado y el rayo refractado se encuentran en el mismo plano. La relación entre el ángulo de incidencia  $\theta_1$  (ángulo formado por el rayo incidente y la normal) y el ángulo de refracción  $\theta_2$  (ángulo formado por el rayo refractado y la normal) se establecen mediante la ley de Snell, la cual se describe en la ecuación (6) [15].

$$\frac{\text{sen}\theta_2}{\text{sen}\theta_1} = \frac{v_2}{v_1} = \frac{n_1}{n_2} \quad (6)$$

Dónde:

$\theta_1$  = ángulo de incidencia

$\theta_2$  = ángulo de refracción.

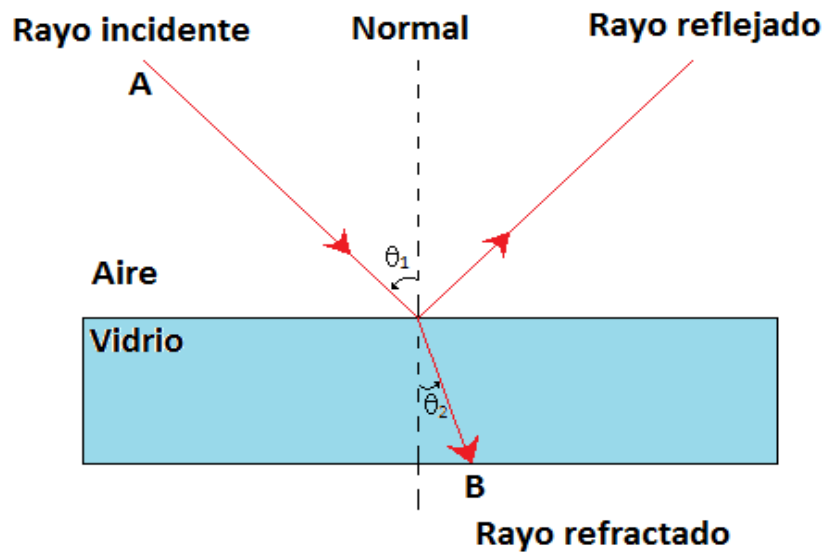
$n_1$  = constante de refracción del medio incidente

$n_2$  = constante de refracción del medio refractado

$v_1$  = velocidad de la luz en el medio incidente

$v_2$  = velocidad de la luz en el medio refractado

Figura 3 Refracción de un haz de luz

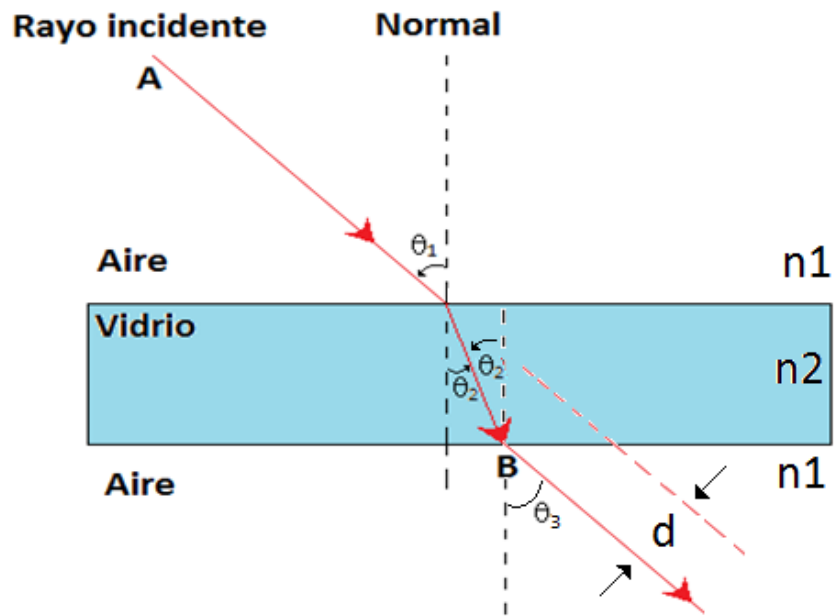


La constante de refracción de un medio está determinada por la velocidad de la luz en dicho medio.

Por otra parte, cuando un haz de luz pasa de un medio a otro y vuelve al medio inicial, este haz mantiene su dirección pero sufre un desplazamiento “ $d$ ” paralelo a la posición inicial. Este fenómeno se puede evidenciar en la figura 4 donde un rayo de luz atraviesa un plano de vidrio [16].

Teniendo en cuenta lo anterior, es necesario estudiar cómo se ven afectadas las imágenes en estas condiciones. Para ello considérese la figura 5 en donde el objeto está en un medio de índice de refracción “ $n_1$ ” mayor que el medio donde se lo observa “ $n_2$ ”. En la figura 5 el punto “O” representa la posición real del objeto, “I” la posición aparente, “ $p$ ” distancia entre la frontera refractante y la posición real, mientras que “ $q$ ” describe la distancia entre la frontera refractante y la imagen aparente.

Figura 4. Un rayo de luz que atraviesa una lámina de vidrio



La expresión matemática que relaciona las posiciones de la imagen real y la imagen aparente se describe en la ecuación (7) [16].

$$\frac{n_1}{p} = -\frac{n_2}{q} \quad (7)$$

$$q = -\frac{n_2}{n_1} p \quad (8)$$

Dónde:

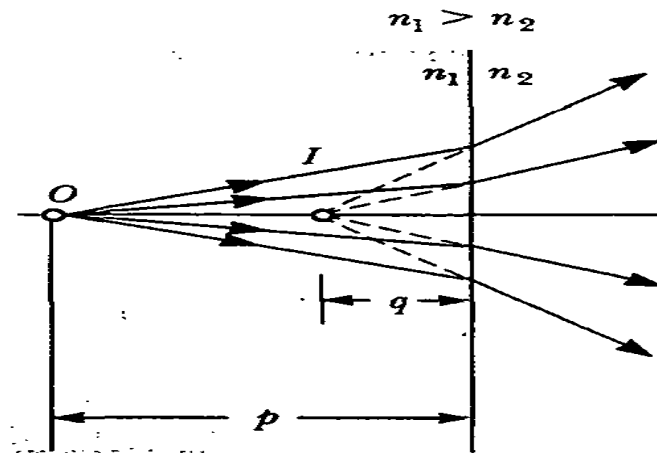
$n_1$  = índice de refracción del medio 1

$n_2$  = índice de refracción del medio 2

$p$  = distancia entre posición real del objetivo y la frontera refractante

$q$  = distancia entre posición aparente del objeto y la frontera refractante

Figura 5. Imagen formada por efecto de la refracción de la luz

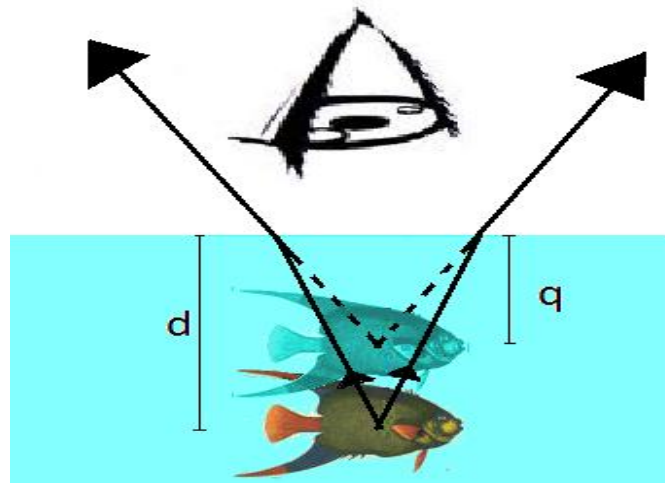


Fuente: [16]

De la ecuación (8) se puede observar que el signo de  $q$  es opuesto al de  $p$ , lo que quiere decir que la imagen virtual formada (imagen aparente) está en el mismo medio donde se encuentra el objeto.

Para aclarar mejor este fenómeno refiérase al caso de una persona que visualiza un pez que se encuentra en el agua. En este caso el observador visualizará el pez más cerca (posición aparente) de donde en realidad se encuentra el animal (posición real), véase la figura 6.

Figura 6. Observador visualizando un pez desde fuera del agua



### 1.3. CALIBRACIÓN DE CÁMARAS

La calibración de cámaras es un proceso que se utiliza en el área de Visión Artificial y que permite conocer cómo una cámara proyecta un objeto del mundo real en el plano imagen para así extraer información métrica a partir de esa imagen. Este proceso culmina con la obtención de los parámetros intrínsecos y extrínsecos de la cámara utilizando fotografías o *frames* de video tomados por la misma. Estos parámetros definen las condiciones de formación de la imagen, de tal forma que los parámetros intrínsecos arrojan información relacionada con las propiedades físicas y de construcción de la cámara tales como: geometría interna y óptica, mientras que los parámetros extrínsecos son los que relacionan los sistemas de referencia del mundo real y la cámara describiendo la posición y orientación de la cámara en el sistema de coordenadas del mundo real [17].

La distancia focal, la relación pixel/milímetro, el punto principal y los coeficientes de distorsión conforman el conjunto de parámetros intrínsecos, mientras que el vector de posición y la matriz de orientación representan los parámetros extrínsecos de la cámara. En la sección 1.1.3.1 se da una idea general de los parámetros [17].

### 1.3.1. MODELO PIN-HOLE

Es el modelo de cámara que tradicionalmente se utiliza para pasar de coordenadas reales 3D (en unidades de longitud) a coordenadas de imagen 2D (píxeles). El principio de funcionamiento radica en que los rayos de luz provenientes de un objeto iluminado atraviesan un agujero fino (foco de la cámara) para impactar en el sensor imagen, el cual está compuesto por pequeñas celdas que capturan la luz y la convierten en energía eléctrica. La transformación de coordenadas depende de la relación pixel/ milímetro (parámetro intrínseco) que es propio de la construcción del sensor imagen (CCD ó CMOS), de la distancia focal que es la distancia medida desde el foco de la cámara hasta su plano imagen, del punto principal que es el centro del plano imagen y que idealmente coincide con el eje óptico (recta que pasa por el foco de la cámara y que corta el plano imagen) de la cámara. En la figura 7 se puede apreciar el proceso de transformación de coordenadas 3D a 2D [17].

En esta figura, “ $C$ ” representa el foco de la cámara, éste es cortado por el eje óptico que a su vez corta el plano imagen en el punto  $(u_0, v_0)$  que se conoce como punto principal y que coincide con el centro del plano imagen, el plano imagen se encuentra a una distancia “ $f$ ” del foco llamada distancia focal.

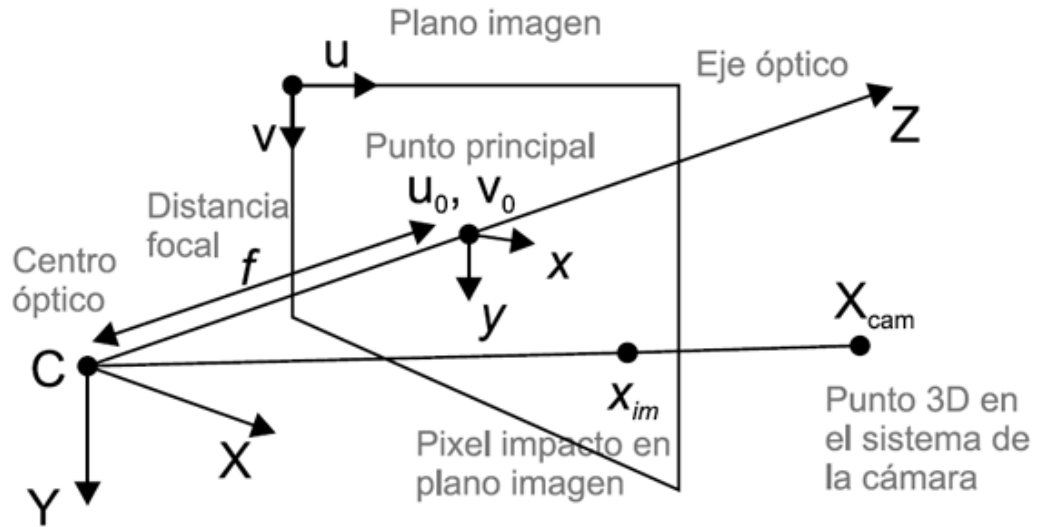
El punto en el espacio es representado en el plano de referencia de la cámara y posteriormente transformado a un punto en el plano imagen con coordenadas de píxeles utilizando todos los parámetros descritos anteriormente.

Sin embargo el modelo de *pin-hole* es ideal y no tiene en cuenta las distorsiones que se generan por causa de la configuración de los lentes de las cámaras a la hora de su construcción. Estas distorsiones en la práctica, alteran negativamente la transformación de coordenadas causando que se obtengan imágenes deformadas.

Para obtener los valores de los parámetros de la cámara es necesario implementar métodos de calibración desarrollados a lo largo de los años y que han sido probados y evaluados en diferentes trabajos de investigación. Los métodos de calibración de cámaras que existen hoy en días son: métodos de calibración clásicos y métodos de autocalibración [23].



Figura 7. Modelo Pin-hole



Fuente: [17]

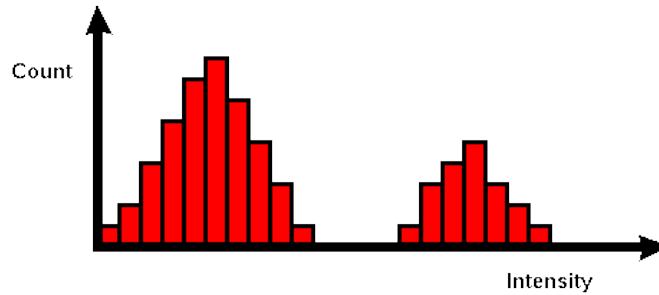
## 1.4. PROCESAMIENTO DE IMÁGENES

### 1.4.1. HISTOGRAMA

En procesamiento de imágenes, el histograma hace referencia a los valores de intensidad de pixel. El histograma es un gráfico que muestra el número de pixeles de la misma intensidad que se repiten en una imagen. Así pues, para un canal que se encuentra en escala de grises donde el valor de cada pixel está en el rango de  $[0 - 255]$ , el gráfico del histograma muestra la distribución de pixeles para las 256 posibles intensidades diferentes [18].

Los histogramas son utilizados para analizar el rango de intensidades que más se distribuye en la imagen y así poder abstraer la información relevante, y por consiguiente las zonas de interés. Esto se puede apreciar en la figura 8.

Figura 8. Histograma de una imagen



Fuente: [18]

### 1.4.2. BINARIZACIÓN O UMBRALIZACIÓN

La binarización de una imagen consiste en la reducción de información donde sólo persisten los valores verdadero y falso, los cuales en una imagen digital están representados por cero (0) y uno (1) o por los colores negro (valor de gris 0) o blanco (valor de gris 255).

Las imágenes binarizadas se usan generalmente para separar regiones de interés del resto de la imagen o bien para identificar individualmente objetos. Para llevar a cabo este proceso es necesario definir los umbrales (valores de grises entre 0 y 255), lo cual actuará como un filtro, así pues los pixeles que estén dentro de los umbrales asignados tomarán el valor de 255 (color blanco) y el resto tomarán el valor de cero (color negro). Ver figura 9.

Lo anterior se puede ver como se muestra en la ecuación (9):

$$P_{out}(x,y) = \begin{cases} 255, & umbral1 \leq P_{in}(x,y) \leq umbral2 \\ 0, & P_{in}(x,y) < umbral1 \text{ y } P_{in}(x,y) > umbral2 \end{cases} \quad (9)$$

Dónde

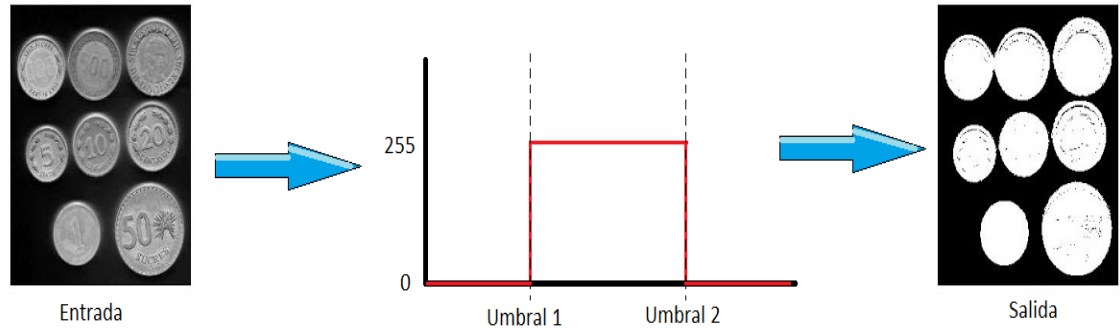
$P_{out}(x,y)$  = valor del pixel de salida (pixel binarizado)

$P_{in}(x,y)$  = valor del pixel de entrada

$umbral1$  = valor inferior de umbral

$umbral2$  = valor superior de umbral

Figura 9. Proceso de binarización



### 1.4.3. MORFOLOGÍA

En procesamiento de imágenes la morfología es el conjunto de operaciones que se realizan sobre éstas con el objetivo de extraer componentes de la misma, útiles para la caracterización, descripción y representación de formas (contornos, esqueletos etc.). A continuación se describen algunos de éstos, para los cuales se tomará como pixeles blancos a aquellos que hacen parte del objeto de la imagen y los pixeles negros como el fondo de la misma.

**A) Dilatación.** La dilatación es una de las operaciones básicas de morfología de imágenes que se usa generalmente cuando éstas se encuentran umbralizadas. La función principal de la dilatación es aumentar gradualmente los límites de la región de los objetos de la imagen mientras que los agujeros de esa región se hacen más pequeños [19], [20].

La dilatación de una imagen A con una imagen B (elemento estructurante EE), consiste en dibujar réplicas de B sobre cada uno de los pixeles que representan el objeto de A. Para ilustrar esta operación véase la figura 10 (a) que representa una imagen A la cual se va a dilatar con la imagen B, véase la figura 10 (b). Para un pixel de A que se encuentre en la frontera del objeto, al dibujarse la réplica de B sobre éste, aparecerán nuevos pixeles que harán parte de la nueva imagen, ver

figura 10 (c) (los nuevos pixeles están ligeramente coloreados de naranja). Sin embargo los lugares donde los pixeles no están cercanos a la frontera del objeto quedaran iguales sin sufrir ningún cambio, ya que al dibujarse la réplica de B sobre A no modifica a la imagen (ver figura 10 (d)). Una vez realizada esta operación sobre todos los pixeles de la imagen A se obtiene una imagen dilatada. En la figura 10 (e) se muestra la imagen A dilatada con B, donde los nuevos pixeles aparecen coloreados, mientras que en la figura 10 (f) se observa el resultado final de esta operación.

**B) Erosión.** La erosión es otra de las operaciones fundamentales de morfología de imágenes al igual que la dilatación, sin embargo a diferencia de ésta tiene un efecto contrario el cual es de adelgazar las imágenes de forma gradual reduciendo los límites de los objetos, mientras que los agujeros se hacen más grandes [19], [20] .

El resultado de erosionar una imagen A con una imagen B (elemento estructurante EE) es el conjunto de pixeles donde B está contenida en A, una vez se haya recorrido B a lo largo de A, por lo tanto los pixeles que no cumplen con esta condición son eliminados, obteniendo de este modo una imagen A con sus objetos reducidos. Para ilustrar esta operación véase la figura 11 (a) donde se muestra una imagen A que va a ser erosionada con una imagen B, figura 11 (b). Para un pixel ubicado en la frontera del objeto de la imagen A, al posicionarse la imagen B, se da la situación en que B no se encuentra contenida en A, ver figura 11 (c), por consiguiente este pixel no pertenece a la erosión de A con B, razón por la cual es eliminado. Caso contrario ocurre para un pixel de A localizado en el interior del objeto dibujado en A, donde al posicionarse la imagen B sobre este pixel, se cumple que B está contenido en A, así que éste pixel analizado pertenece a la erosión de A con B, de manera que no se elimina, ver figura 11 (d). Realizando este proceso para todos los pixeles de la imagen A se obtiene una nueva imagen donde el objeto en cuestión se encuentra reducido. En la figura 11 (e) se puede detallar que los pixeles que no pertenecen a la erosión de A con B, y que han sido eliminados se encuentran coloreados de rojo, mientras que la figura 11 (f) muestra el resultado final de ésta operación.

Figura 10. a) Imagen de entrada A, b) imagen B o elemento estructurante, c) proceso de dilatación para dos pixeles ubicados en la frontera del objeto, d) proceso de dilatación para dos pixeles que no se encuentran en la orilla del objeto, e) imagen A dilatada con B donde los nuevos pixeles aparecen coloreados, f) resultado de A dilatada con B.

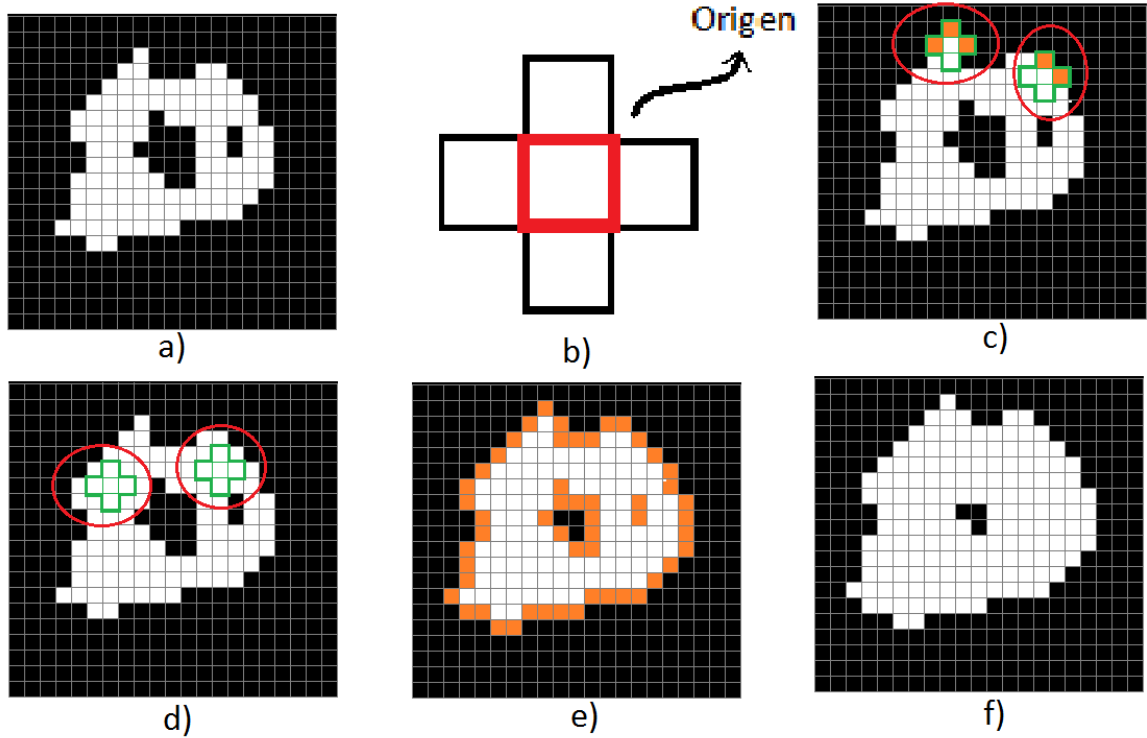
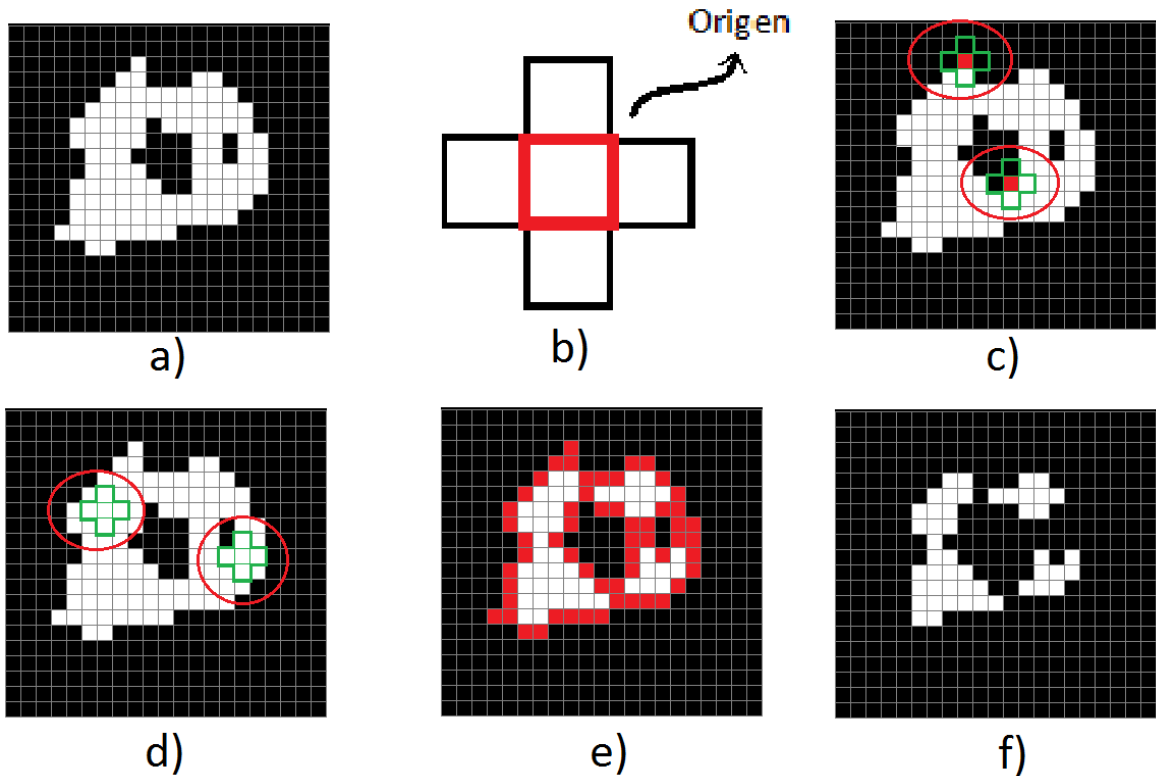


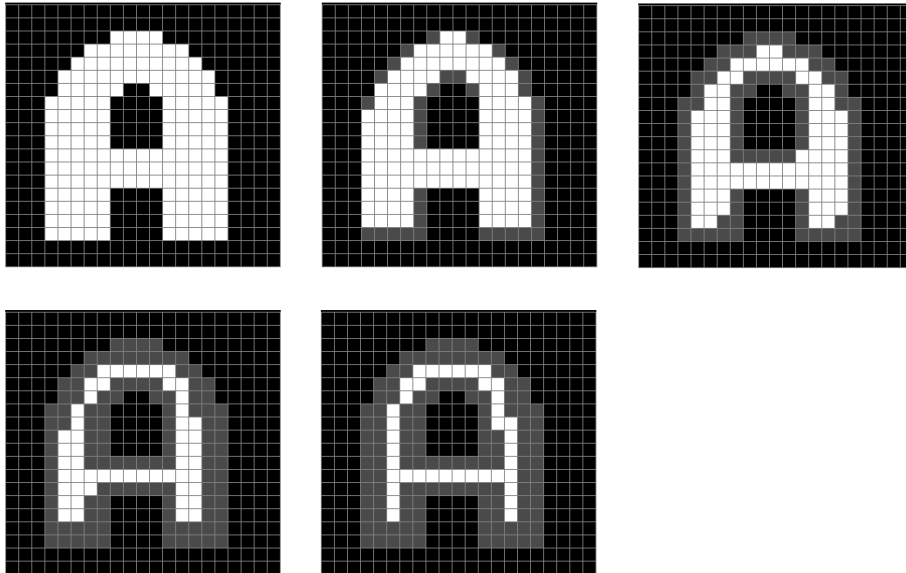
Figura 11. a) Imagen de entrada A, b) imagen B o elemento estructurante, c) dos pixeles que no pertenecen a A erosión B, d) dos pixeles que si pertenecen a A erosión B, e) A erosión B con pixeles eliminados coloreados de rojo, f) resultado de A erosión B.



**C) Esqueletización.** La Esqueletización es una operación morfológica para reducir regiones y objetos del primer plano. Es el proceso de quitar de un patrón la mayor cantidad de pixeles posibles sin que se afecte el patrón general, conservando la forma, de tal manera que después del proceso de esqueletizado el objeto debe ser reconocido. Para mayor información referirse a [20].

La esqueletización consta de procesos iterativos donde se van eliminando los pixeles del contorno del objeto de una imagen, hasta finalmente obtener un patrón que conserva la misma forma del objeto original representado con la menor cantidad de pixeles posibles. Para ejemplarizar esta operación véase la figura 12, donde se ilustra una imagen que ha sufrido el proceso de esqueletizado. Se puede observar que cada vez se va obteniendo una imagen con menor cantidad de pixeles, pero que aun así conserva la forma del objeto de la imagen inicial. Al final se muestra la imagen completamente esqueletizada.

Figura 12. Proceso de esqueletización para una imagen



## 1.5. RENDERIZACIÓN

La renderización es el proceso que se lleva a cabo para representar en imágenes 3D un conjunto de datos, utilizando métodos que permiten relacionar entre sí estos datos a través de estructuras poligonales regulares. El conjunto de datos generalmente es conocido como nube de puntos, donde cada punto está conformado por coordenadas  $(X,Y,Z)$ , que en un sistema de coordenadas tridimensional representan vértices, los cuales se unen por algún método de creación de polígonos y se obtiene la superficie externa del objeto del cual se obtuvieron dichos puntos [21].

La nube de puntos se obtiene utilizando alguna de las técnicas de reconstrucción 3D descritas en la sección 1.1 y puede ser almacenada en diferentes formatos dependiendo de la herramienta que se utilice para realizar el renderizado. Los formatos más utilizados son: Archivos VRML y Archivos PLY. Una vez cargada la nube de puntos a la herramienta software se prosigue a aplicar algoritmos de renderizado soportados por la herramienta.

Los algoritmos de renderizado se clasifican en cinco categorías los cuales pertenecen a una de ellas de acuerdo a: tipo de dato de entrada, división espacial,

tipo de representación de la superficie, aproximación o interpolación, diferentes asunciones que realizan los algoritmos. Estas categorías son listadas y explicadas en el Anexo B.

Para construir una superficie poligonal a partir de una nube de puntos se deben realizar tres pasos los cuales se detallan en el Anexo B.

1. Pre-Procesamiento: En esta fase se eliminan datos erróneos correspondientes a ruido, en un sistema cámara láser podría ser ocasionado por el reflejo del láser en un objeto diferente al objetivo, ocasionando puntos aislados de la imagen a reconstruir.
2. Generación de superficie poligonal: Mallas triangulares o cuadráticas son creadas satisfaciendo ciertos requerimientos de cualidades, como son, límites en el tamaño de los elementos de la malla, no intersección entre “breaklines”, etc.
3. Post-Procesado: Cuando el modelo es creado, operaciones de edición son comúnmente aplicadas para refinar y perfeccionar la superficie poligonal.

Todas las herramientas software de renderizado realizan estos pasos, los cuales ayudan a tratar la información importada y así obtener el mejor modelo posible del objeto escaneado. En el Anexo B se describen las herramientas de renderizado que existen hoy en día.



## 2. DISEÑO DE LA PLANTA Y CAPTURA DE IMÁGENES

### 2.1. TÉCNICA SELECCIONADA

En el capítulo 1 (sección 1.1) se expusieron las técnicas de reconstrucción 3D que actualmente se utilizan en proyectos de investigación y afines. Éstas se utilizan de acuerdo al objetivo que se desea satisfacer y la única limitación para implementarlas en proyectos académicos es el costo que demandarían los instrumentos y/o sistemas que las utilizan.

No obstante existen preferencias para aplicar determinada técnica de acuerdo al contexto de investigación, por ejemplo en el campo industrial generalmente se utilizan las técnicas de triangulación láser y franjas, en el campo de visión artificial es común encontrar desarrollos donde se ha aplicado la técnica de visión estereoscópica.

De acuerdo a lo anterior se puede pensar que cualquier técnica que no represente altos costos de implementación puede ser aplicada en el desarrollo de este trabajo. Sin embargo, dependiendo de la técnica es posible que en la adquisición de los datos varíe su complejidad y que se cuente con un mayor o menor grado de incertidumbre en la validez de los resultados.

Por lo tanto se han definido criterios que ayuden a determinar qué técnica de reconstrucción 3D es la más adecuada para aplicar en este trabajo, sin perder el horizonte trazado en la definición del objetivo que se desea cumplir. Los criterios son los siguientes:

- **Bajo costo:** Este criterio atiende al objetivo principal de este trabajo. Por tanto se debe garantizar que toda la instrumentación y materiales utilizados para aplicar la técnica de reconstrucción 3D no requieran valores elevados de adquisición.
- **Fácil implementación:** Este criterio se refiere a que no se necesite implementar una infraestructura compleja para que la técnica funcione.
- **Implementación bajo el agua:** Indica que la técnica sea implementable en trabajos que se relacionen con reconstrucción 3D de objetos sumergidos en agua y que el acoplamiento y/o adaptación de ésta técnica en dicho medio no afecte negativamente los demás criterios.

- Precisión: Indica la proximidad de la medida proporcionada por el sistema de visión que utiliza la técnica de reconstrucción 3D, al valor real de la variable medida.
- Poca iluminación del ambiente: Se refiere al buen funcionamiento de la técnica en un ambiente de trabajo que no cuente con un buen sistema de iluminación, es decir que en el ambiente no sea necesario colocar fuentes de luz para adecuar la iluminación de la escena trabajada.
- Fácil acceso a instrumentos y/o dispositivos que se necesitan para construir el sistema: Con este criterio se busca garantizar que toda la instrumentación que se requiere en la construcción del sistema que utiliza la técnica de reconstrucción 3D se pueda adquirir factiblemente.
- Implementación en números trabajos de investigación: Se refiere a que la técnica seleccionada haya sido aceptada y aplicada a lo largo de los años en numerosos trabajos de investigación y que dichos trabajos hayan arrojado resultados satisfactorios.

Con los criterios definidos, se desarrolló un análisis de las técnicas de reconstrucción 3D que se han expuesto en el capítulo 1 (sección 1.1) a través de una matriz FODA [22]. El análisis de estos criterios se encuentra en el Anexo C.

Con los resultados obtenidos en la evaluación de los criterios se pudo determinar que la técnica mas adecuada para aplicar en este trabajo de investigación es la “Triangulación Láser”. Ésta se escogió por las siguientes razones:

- Es la técnica de menor costo y fácil implementación. Sólo necesita como mínimo una cámara y un láser para elaborar el sistema.
- Aunque se tiene un solo perfil por posición del sistema cámara - láser, se puede realizar un barrido al objeto moviendo este sistema. Esto permite que se simule el funcionamiento de un proyector de luz.
- Según otros trabajos de investigación se obtienen resultados que permiten concluir que esta técnica es suficiente para satisfacer el objetivo planteado.

## 2.2. FENÓMENO QUE EXPERIMENTA UN OBJETO SUMERGIDO EN AGUA

En la sección 1.1 del capítulo 1 se habla sobre las técnicas ópticas como herramientas para llevar a cabo el proceso de reconstrucción 3D, donde se detalla y explica la triangulación láser, la cual es la que se implementó en este trabajo. Sin embargo esta técnica se enmarca en condiciones donde el sistema a implementar se encuentra en el mismo medio, por ejemplo aire (ver figura 3). Ahora se mostrará cómo funciona la técnica cuando la cámara – láser se encuentra en el aire y el objeto está sumergido en agua. En éste proyecto los experimentos se llevaron a cabo en un acuario (para más detalle sobre el hardware utilizado remítase a la sección 2.4), donde interviene el efecto tanto del agua como del vidrio.

Es claro afirmar que la luz al pasar de un medio transparente a otro cambia su dirección, véase la figura 3. En este trabajo se implementa un sistema en el cual intervienen tres medios, aire, vidrio y agua con distintos índices de refracción. El fenómeno que sufre el haz de luz al pasar sobre estos tres medios se explica a continuación.

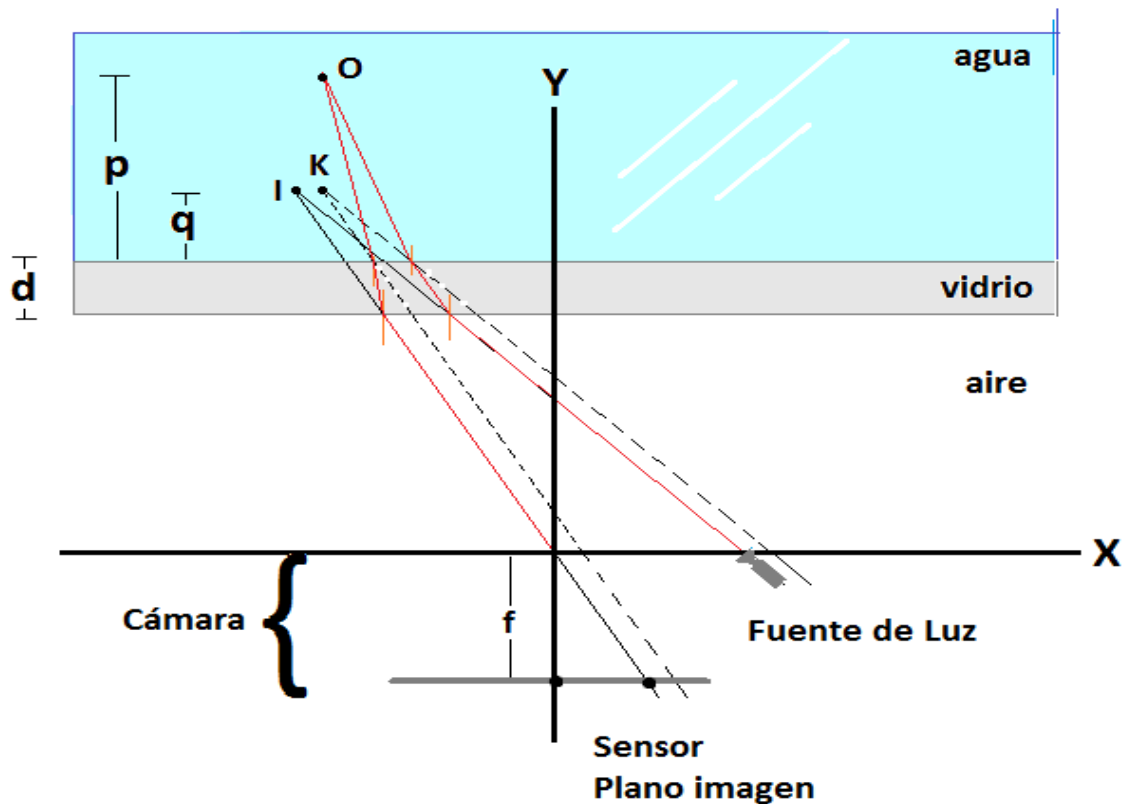
Para entender mejor la consecuencia de aplicar la técnica de triangulación en el contexto donde interviene el aire, vidrio y agua véase la figura 13, donde se observa como la imagen sufre un desplazamiento respecto a su posición original por efecto de la refracción. El punto “O” describe la coordenada de la posición real del objeto, “K” es el punto aparente del objeto por causa de la refracción del agua, e “I” representa la posición aparente de éste por efecto del vidrio.

Una imagen formada por efecto de la refracción se describe en el capítulo 1 (sección 1.1) donde se afirma que existe una posición real del objeto y una posición aparente del mismo y que se relacionan mediante la ecuación (6). Este efecto explica el fenómeno que sufren los puntos “O” (posición real) y “K” (posición aparente por causa del agua) donde se tiene la apariencia de que el objeto se encuentra más cerca (por motivo de los índices de refracción de los medios evaluados). A este hecho se une la acción del vidrio, el cual imprime la característica de que un haz de luz que lo atraviese sufre un desplazamiento pero conserva su dirección (mientras este vuelva al mismo medio, véase la figura 4) y que se demuestra por la ecuación (7). Esto se evidencia en la figura 13 donde se observa que la nueva posición aparente se desplaza del punto “K” al punto “I”. De lo anterior se puede concluir que un objeto sumergido en un acuario (vidrio) con agua, y que es observado desde afuera (aire) adopta la apariencia de que se encuentra más cerca (posición aparente) respecto la posición real, sin embargo

éste no sufre deformaciones en cuanto a su tamaño y forma, ya que las fronteras refractoras son planas.

Por consiguiente al aplicar la técnica triangulación láser se pueden calcular las coordenadas del objeto respecto a una posición aparente del mismo. En este trabajo se busca obtener un modelo 3D que represente al objeto real en cuanto a forma y que mantenga una proporción en cuanto a su tamaño, y no es crítico conocer el desplazamiento del objeto aparente (imagen aparente) respecto al real (imagen real) ya que este hecho no deforma la imagen aparente del objeto. De esta manera se puede recoger la información de la imagen aparente del objeto para llevar a cabo el cálculo de coordenadas tridimensionales y obtener un modelo 3D del objeto real.

Figura 13. Técnica de triangulación para objetos sumergidos bajo el agua



### 2.3. CALIBRACIÓN DE LA CÁMARA: *HARDWARE - SOFTWARE*

Para el desarrollo de esta etapa es necesario conocer cual método de calibración es el más adecuado. Teniendo como referente los métodos de calibración descritos en [23] se puede concluir lo siguiente de los métodos de calibración clásicos y de los de auto calibración:

- **Métodos de calibración clásicos**
  - ✓ En estos métodos los valores del centro óptico, la distancia focal y la relación pixel-milímetro varían según la posición del patrón de calibración. Los demás parámetros se ven afectados en menor medida.
  - ✓ La estimación de la distancia a la cual se encuentra el objeto es un factor crítico debido a que ésta es muy sensible a variaciones.
  - ✓ La variación de la distancia focal (zoom) trae como consecuencia la variación de los parámetros extrínsecos y el primer coeficiente de distorsión radial.
  - ✓ El aumento de la cantidad de puntos definidos en el patrón de calibración no es determinante en el momento de estimar los parámetros de la cámara, pero si ayuda a mejorar la precisión de los valores de los mismos.
  - ✓ Los factores como iluminación y tamaño de los puntos tienen una leve incidencia en la obtención de las coordenadas de la imagen (coordenadas 2D en pixeles).
  - ✓ De los métodos de calibración clásicos, los dos que resultan más fiables son los de Zhang y Batista [23], ya que estos presentan niveles de error más bajos en el momento de hacer la reconstrucción de los puntos.
- Los métodos de calibración clásicos presentan buenas estimaciones en el cálculo de los valores de los parámetros de la cámara para unas condiciones establecidas, si dichas condiciones se cambian la estimación de los parámetros presentan una fuerte dispersión.

- **Métodos de auto calibración**

- ✓ Unos métodos presentan mejores resultados en la estimación de los parámetros de la cámara si los parámetros intrínsecos se mantienen constantes a lo largo de la secuencia utilizada para la calibración, en contraparte demandan mucho coste computacional, este es el caso del método de McLauchlan [23].
- ✓ Otros presentan buena estimación del valor del centro óptico pero mala estimación en el valor de la relación pixel-milímetro. Adicionalmente no exige demasiado coste computacional.

De los métodos de calibración clásicos como los de autocalibración no se puede afirmar que exista un método que sea mejor o superior que otro. Cada método presenta sus ventajas y sus desventajas. Sin embargo se puede decir que los métodos de calibración de cámaras que mejores resultados arrojan en cuanto a la precisión de cálculo de los valores de los parámetros intrínsecos y extrínsecos y que no demandan un gran coste computacional son los clásicos.

Tomando lo anterior como criterio de selección del método a utilizar para desarrollar el proceso de calibración de la cámara, se optó por implementar este proceso con un método de calibración clásico. Una vez escogido el método se buscó una librería que brinde funciones desarrolladas para realizar este proceso de calibración y que dichas funciones estén basadas en los métodos de calibración clásicos, además de esto, que sea de código no privativo, que presente estabilidad funcional y que sea masivamente utilizada en el campo de la visión artificial. La búsqueda dio como respuesta a OpenCv como librería a implementar en el proceso de calibración de la cámara. Seguidamente se pensó en utilizar un lenguaje de programación que brinde confiabilidad y buen rendimiento en cuanto al tiempo de cálculo, procesamiento y que sea de código no privativo, de los más fuertes en el mercado se decidió escoger el lenguaje de programación multiplataforma, interpretado y de código abierto Python.

Estudiando la librería OpenCv se encontraron funciones que permiten realizar el proceso de calibración de cámaras utilizando el modelo de calibración *pin-hole* descrito en la sección 1.3.1.

La gran mayoría de los procedimientos de calibración se basan en este modelo (modelo de cámara de agujero o *pin-hole*), que es el modelo más simple que se puede obtener de una cámara, por lo tanto es el que necesita un menor número

de parámetros para ser representado. Como se explicó en la sección 1.3.1 este modelo se basa en que la proyección de un punto de la escena se obtiene de la intersección de una línea que pasa por este punto y el centro de proyección (foco) con el plano imagen.

La función utilizada para la calibración de la cámara es la llamada CalibrateCamera2. Esta función recibe como parámetros de entrada las coordenadas  $(X, Y, Z)$  de los puntos del objeto que se encuentra en el mundo real, los puntos de la imagen (que son los puntos del objeto proyectados en la vista de la cámara), el número de puntos en cada vista particular y el tamaño de la imagen. Recibe como parámetros de salida una matriz de  $3 \times 3$  inicializada donde se van a guardar los valores que corresponden a los parámetros intrínsecos de la cámara, un vector inicializado donde se guardan los valores que corresponden a los coeficientes de distorsión, una matriz inicializada donde se guardan los valores que corresponden a los parámetros extrínsecos de rotación y un vector inicializado donde se guardan los valores que corresponden a los parámetros extrínsecos de traslación [17].

Para el proceso de calibración se elaboró un tablero de ajedrez como patrón de calibración (necesario en los métodos clásicos de calibración) que contiene 7 esquinas internas de ancho por 6 esquinas internas de largo, para un total de 42 esquinas internas espaciadas entre si 3.4 centímetros (ver figura 14).

Para la elaboración del patrón de calibración se tuvieron en cuenta los siguientes criterios:

- Para una buena estimación en los valores de los parámetros de la cámara se debe contar como mínimo con 12 puntos, que equivale a 12 esquinas internas del tablero, por lo tanto el número mínimo de tableros es de  $4 \times 3$ . El tamaño adecuado del tablero es de  $7 \times 6$  o  $8 \times 7$  cuadrados.
- Las dimensiones de cada cuadrado perteneciente al patrón de calibración es un factor importante para la buena estimación de los parámetros, entre mayor sea la dimensión de cada cuadrado mejor serán los resultados obtenidos.
- Se debe garantizar la diferenciación entre los cuadrados claros y los oscuros. Esto con el fin de que estén bien definidas las esquinas internas del tablero y que la cámara no vaya a tener inconvenientes a la hora de detectar dichas esquinas que tenga como consecuencia obtener valores erróneos.

Una vez elaborado el tablero de ajedrez se instaló la cámara en el sistema operativo Ubuntu. La cámara utilizada es una Webcam LifeCam modelo VX-800 de Microsoft (ver figura 15). Cabe aclarar que este procedimiento de calibración se puede utilizar en cualquier tipo de cámaras digitales.

Existen diferentes tipos de cámaras que se pueden implementar en este proceso, sin embargo éstas difieren en los costos, resolución de la imagen, precisión en su construcción. Algunas de las más sofisticadas traen datos de fabricación que brindan información de algunos parámetros intrínsecos como la distancia focal, el factor de escala y el tamaño del sensor de imagen.

En este trabajo se utilizó una cámara de gama baja por lo tanto no se cuenta con datos del fabricante. Esto implica que todos los valores de los parámetros se deben estimar experimentalmente y que se debe definir un método de validación de dichos valores. En contra parte si se tuvieran los datos del fabricante la validación de estos valores se podría hacer de manera directa.

Figura 14. Patrón de Calibración. Plano de ajedrez de 42 esquinas internas

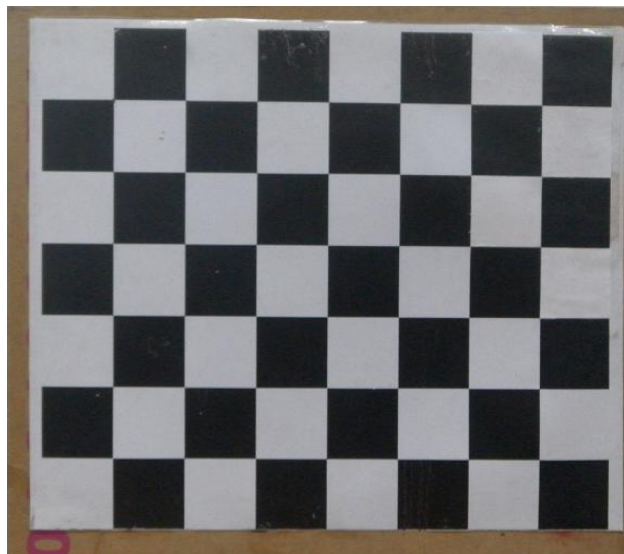




Figura 15. Cámara a calibrar. LifeCam modelo VX-800



**Fuente:** <http://www.microsoft.com/hardware/en-us/p/lifecam-vx-800/JSD-00014>

Posteriormente se desarrolló el código necesario para realizar el proceso de calibración de la cámara. Los algoritmos se desarrollaron en lenguaje Python utilizando el IDE de desarrollo Spyder. Para la interfaz de usuario se utilizó la herramienta PyQt4 y para las operaciones con matrices la librería de Python Numpy.

Para cada conjunto de puntos detectado por *frame* el sistema automáticamente calcula las coordenadas de la imagen en unidades de píxeles y las guarda. Las mismas que posteriormente se utilizan para estimar los parámetros de la cámara.

Además de las coordenadas de la imagen que el sistema calcula automáticamente, este mismo necesita que se definan las coordenadas espaciales del objeto. Para esto se tiene en cuenta el número de cuadrados a lo ancho y largo del tablero de ajedrez como también la cantidad de puntos que se tienen en dicho tablero, en este caso 42 puntos equivalentes a las esquinas internas. Las coordenadas se definen de acuerdo al criterio del interesado, donde este último debe definir las unidades con que se va a trabajar y conservar este criterio en todo el proceso de calibración, inclusive si se piensa desarrollar trabajos adicionales de visión artificial. En este trabajo se definieron las coordenadas de la siguiente manera: para el eje x cada coordenada se determinó con la división entre una variable incremental y el número de cuadrados a lo ancho del tablero de ajedrez. Dicha variable inicia en cero y se incrementa en uno hasta que obtiene el valor equivalente al total de puntos detectados menos uno ( $0/7, 1/7, 2/7, \dots, 41/7$ ). Del

mismo modo se definieron las coordenadas del eje y con la variación de que el cociente de la división ahora es el número de cuadrados a lo largo del tablero de ajedrez ( $0/6$ ,  $1/6$ ,  $2/6$ ,.....,  $41/6$ ). Las unidades de cada coordenada están en milímetros. Con estos datos y con las coordenadas en pixeles guardadas se obtiene la estimación de los dos tipos de parámetros, los intrínsecos, los extrínsecos y los coeficientes de distorsión. Estos son guardados en un archivo con formato *xml* una vez que la cámara ha terminado de tomar todos los *frames* que se especificaron en la interfaz de usuario.

Las cámaras de gama baja como la que se ha utilizado en este trabajo presentan problemas de distorsión radial y tangencial que hacen que las imágenes queden deformadas. Es por eso que se debe aplicar un proceso de rectificación a la cámara. Para hacer esto, es necesario contar con los parámetros intrínsecos bien estimados y con los coeficientes de distorsión. Con estos valores el algoritmo utiliza una función de OpenCv para rectificar la cámara.

Una vez estimados los parámetros y rectificado la cámara se puede representar cualquier punto del mundo real sobre el plano 2D de la cámara, lo que resulta necesario para las siguientes etapas de este trabajo.

En la figura 17 se muestra el diagrama funcional del algoritmo desarrollado para realizar la calibración de la cámara utilizada y en las figuras 18 y 19 se muestra una imagen sin la cámara rectificada y una imagen con la cámara rectificada, respectivamente.

Figura 16. Muestra de 9 *frames* donde la cámara detectó las esquinas internas del patrón de calibración

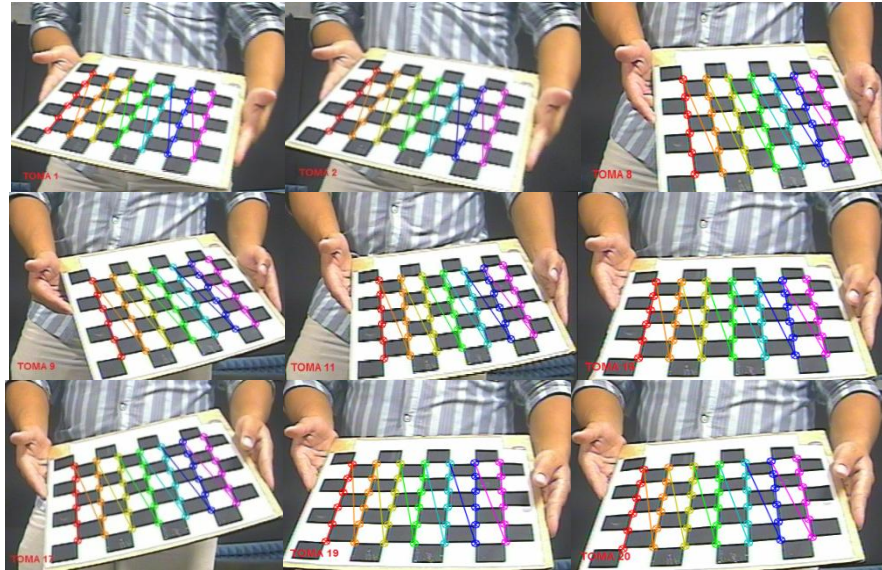


Figura 17. Diagrama funcional del algoritmo de calibración de la cámara

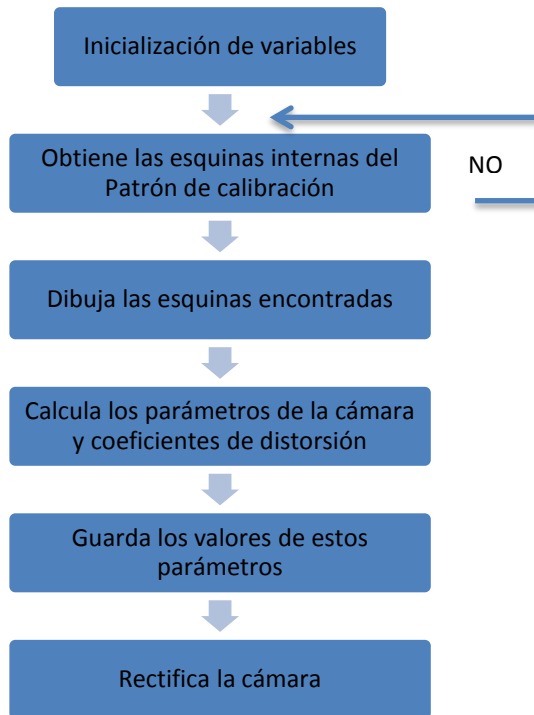


Figura 18. . Imagen con distorsión radial y tangencial

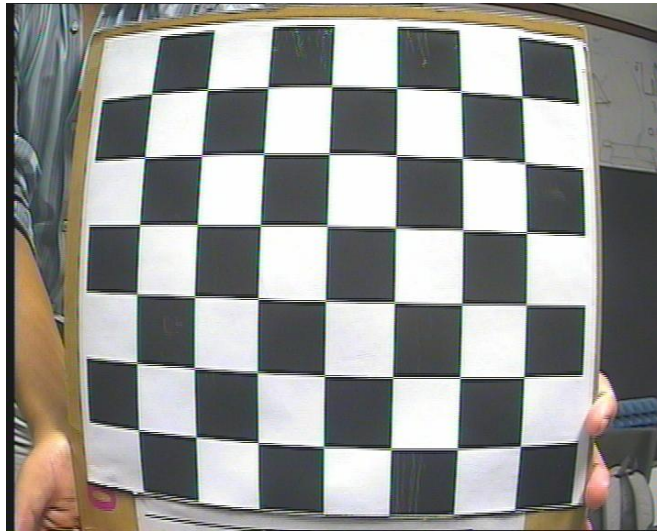


Figura 19. Imagen rectificada

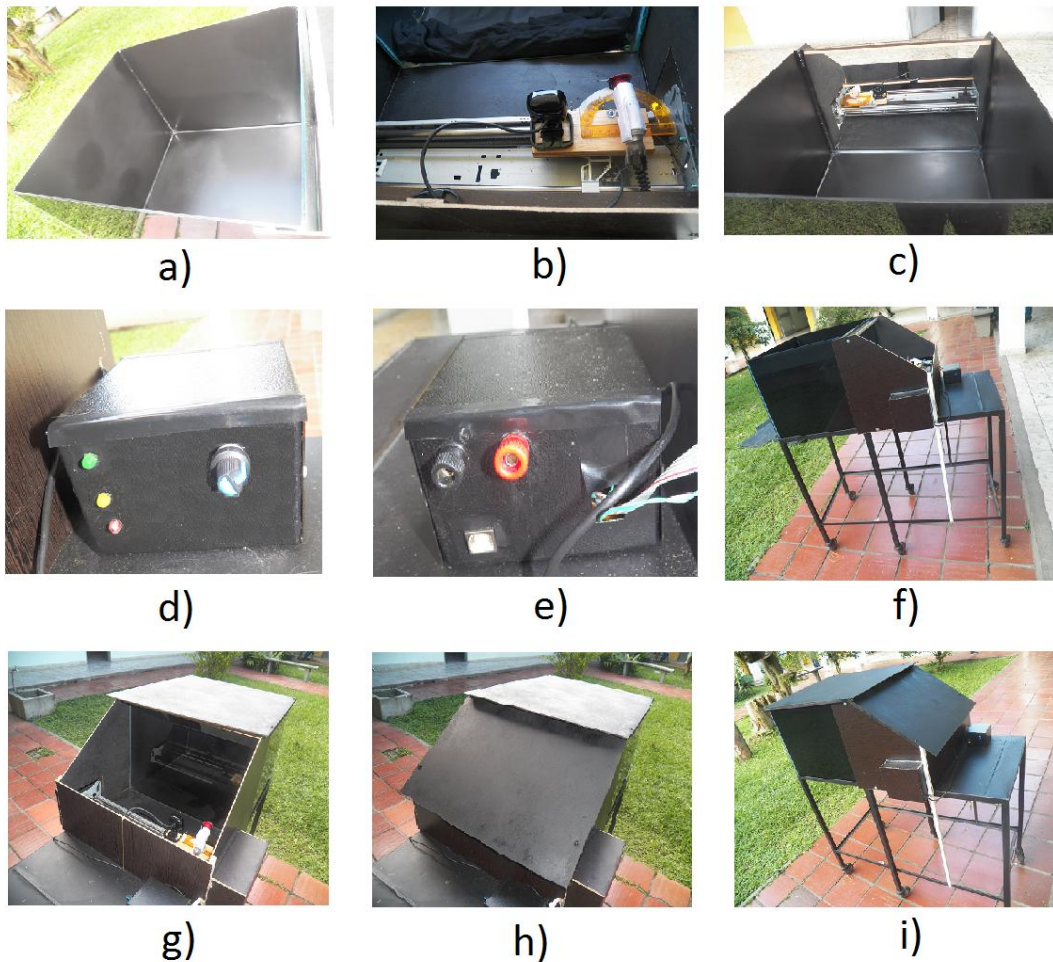


Los resultados obtenidos en esta etapa del proyecto están registrados en el Anexo A.

## **2.4. PLANTA IMPLEMENTADA**

Para llevar a cabo los experimentos de reconstrucción 3D en un medio acuático, fue necesario diseñar e implementar una planta que permite trabajar bajo esas condiciones. La planta utilizada está compuesta por un cubo de vidrio para el depósito de agua, medio donde va a estar sumergido el objeto a reconstruir. También cuenta con un sistema cámara - láser situado en la parte externa del cubo de vidrio, el cuál se desplaza de forma horizontal por medio de un motor paso a paso, para efectuar el proceso de escaneado. Inicialmente se pensó en localizar el sistema de escaneo dentro del cubo de vidrio con el objetivo de trabajar bajo el mismo medio, sin embargo surgió el inconveniente de que este hecho provoca movimiento del agua, produciendo así alteración de las imágenes tomadas y una errónea toma de datos que no permiten efectuar la reconstrucción. La planta está totalmente cubierta con el objetivo de controlar los efectos de luz externa que puedan alterar los resultados, teniendo en cuenta que la técnica de triangulación seleccionada está restringida a que la luz más intensa en la escena debe ser solamente la del láser. Por este motivo, se pintaron de negro las paredes del cubo de vidrio y se agregaron tapas movibles en la parte superior de la planta. La parte electrónica que controla el sistema de escaneo, se encuentra encapsulada en una caja con entradas para la fuente y el cable de comunicación. Además tiene un variador de velocidad del motor paso a paso, el cual es un potenciómetro no lineal. La planta cuenta con un soporte de hierro que lleva incrustada en sus patas rodachines que permiten su fácil traslado. Ésta se puede visualizar en la figura 20.

Figura 20. . a) Cubo de vidrio, b) mecanismo de escaneo, c) vista superior del interior de la planta, d) Parte delantera de la caja del componente electrónico, e) parte trasera del componente electrónico, f) vista externa de la planta sin tapas, g) tapa superior de la planta, h) tapa delantera de la planta, i) vista general de la planta.



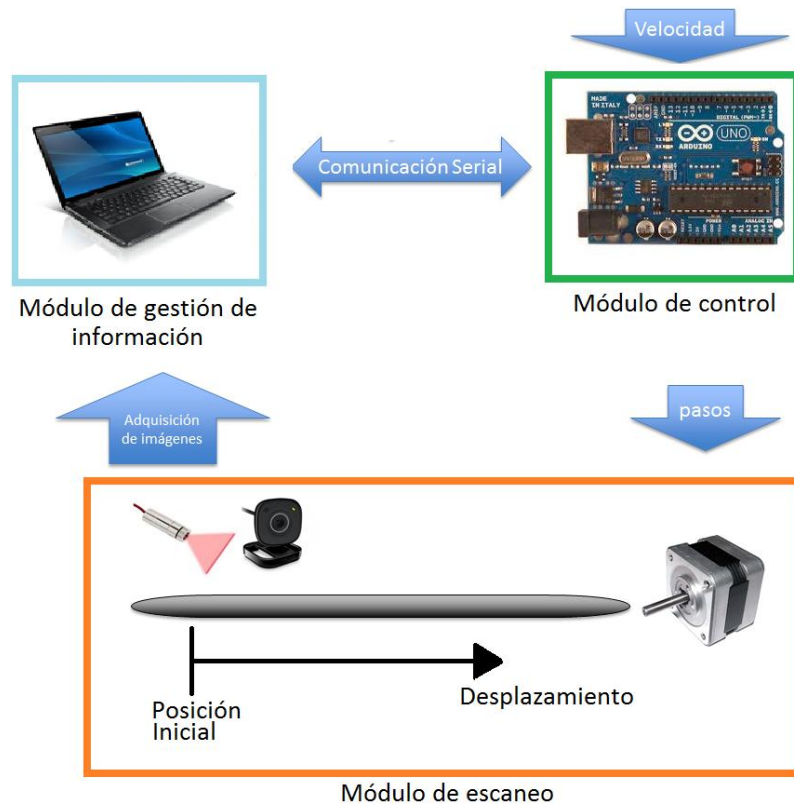
## 2.5. CAPTURA DE IMÁGENES: *HARDWARE - SOFTWARE*

La captura de imágenes es la primera etapa de implementación del sistema hardware/software para realizar la reconstrucción 3D de los objetos utilizados en este trabajo. En esta etapa se capturan los *perfiles* que el haz de luz láser refleja en el objeto mientras se está efectuando el escaneo del mismo. Para llevar a cabo esta etapa se cuenta con un esquema conformado por tres módulos los



cuales son: módulo de gestión de información, módulo de control y un módulo de escaneado, ver figura 21.

Figura 21. Esquema del proceso de captura de imágenes



**Módulo de gestión de información.** Este módulo está compuesto por un ordenador y su función principal consiste en enviar la orden de escaneo del sistema cámara-láser (módulo de escaneo), especificando la distancia en centímetros que se debe desplazar y además es el encargado de encender la cámara para realizar la captura de imágenes en el proceso de escaneo. Inicialmente envía la orden al módulo de control mediante comunicación serial, luego espera la respuesta de éste e inmediatamente activa la cámara web para que realice la captura de imágenes. Una vez realizado el proceso de escaneo, recibe del módulo de control un mensaje que indica el final del proceso, por lo tanto deja de capturar imágenes.

El desarrollo del software implementado se realizó bajo el lenguaje de programación Python, utilizando las librerías de PyQt4 para la interfaz gráfica. Para detallar el código del programa elaborado véase el Anexo D.

**Módulo de control.** Este módulo está compuesto por una tarjeta Arduino Uno y sus componentes electrónicos que conforman el circuito. La función principal del módulo de control es la de gestionar las señales de mando necesarias para el motor paso a paso que se encuentra en el módulo de escaneo. Inicialmente recibe por comunicación serial la orden de la distancia en centímetros que debe recorrer el sistema cámara-láser desde el módulo de gestión de información. Luego verifica que el sistema cámara láser se encuentre en la posición inicial de escaneo, en el caso de no ser así, activa el motor paso a paso y localiza este sistema en dicha posición. Luego traduce las órdenes en cantidad de pasos que el motor debe realizar para ocasionar el desplazamiento del sistema cámara láser y ejecuta esta acción. Una vez que se haya cumplido esta orden, envía un mensaje por comunicación al módulo de gestión de información con el fin de indicarle que el escaneo ha terminado e inmediatamente reinicia las condiciones iniciales posicionando la cámara y el láser en su posición de origen. Cabe anotar que este módulo permite graduar de manera manual la velocidad del motor.

**Módulo de escaneo.** Este módulo está conformado por un motor paso a paso y el sistema cámara láser. La función principal consiste en llevar a cabo el proceso de escaneado mediante el desplazamiento del sistema cámara láser por medio de la activación del motor. Inicialmente recibir la orden de mando del módulo de control que activa el motor paso a paso, el cual mueve el sistema cámara láser por medio de una correa dentada. Es preciso anotar que el láser se activa y se posiciona de manera manual mientras que la cámara se manipula a través del módulo de gestión de información. Además este sistema cuenta con un sensor de proximidad o elemento fin de carrera que permite localizar la cámara y el láser en su posición inicial.

El esquema del circuito implementado para los módulos de control junto con el módulo de escaneo se puede visualizar en la figura 22.





Figura 23. Objetos a ser reconstruido mediante la técnica planteada en este trabajo, a) objeto gato, b) objeto esfera, c) objeto rana, d) objeto tapa.



a)



b)



c)



d)

La primera etapa de la reconstrucción 3D utilizando la técnica planteada en este trabajo consiste en capturar las imágenes de todos los perfiles del objeto mediante el proceso de escaneo láser. Algunos de los perfiles tomados para los objetos que se muestran en la figura 23, se pueden visualizar en las figuras 24, 25, 26 y 27.

Figura 24. Perfiles del objeto gato capturados en el proceso de escaneado.

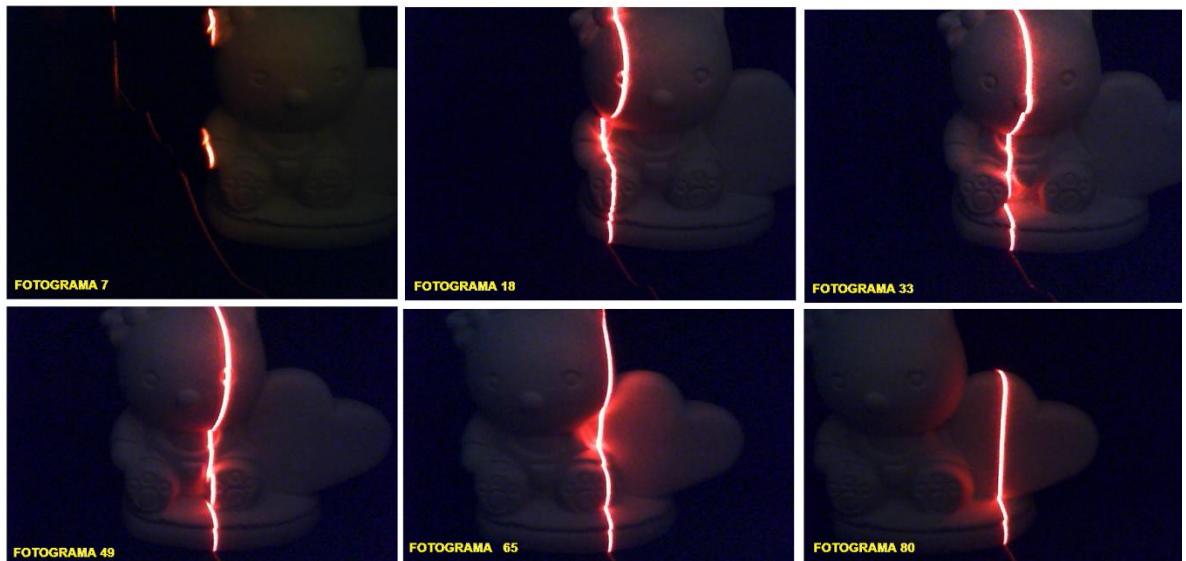


Figura 25. Perfiles del objeto rana capturados en el proceso de escaneado.

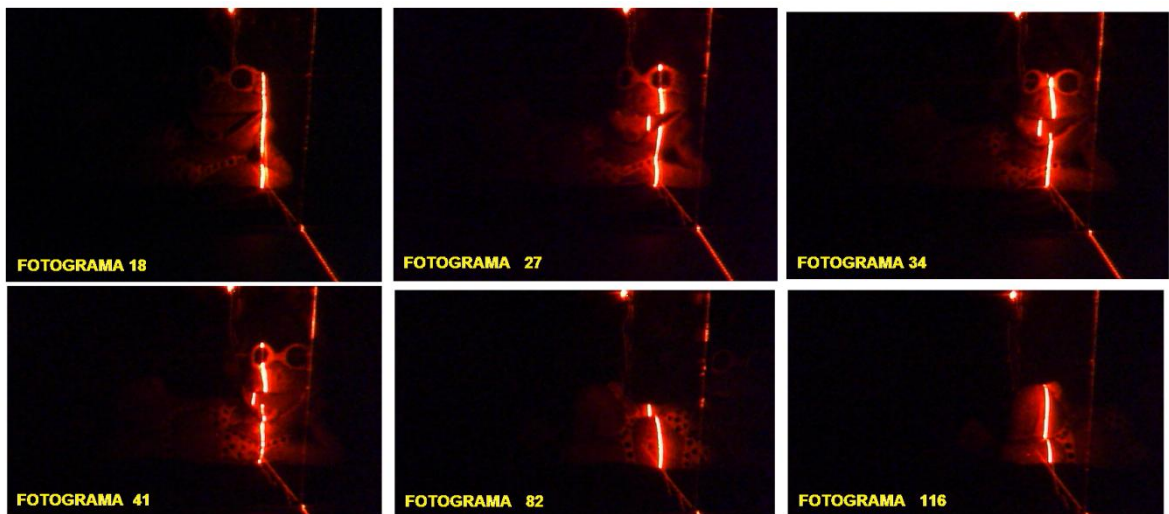


Figura 26. Perfiles del objeto tapa capturados en el proceso de escaneado.

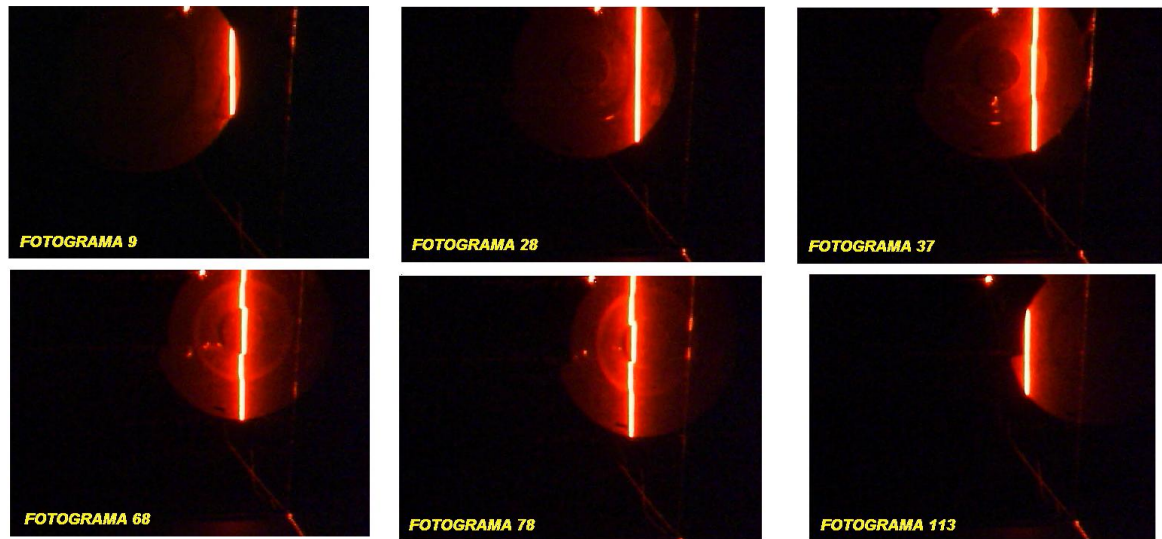
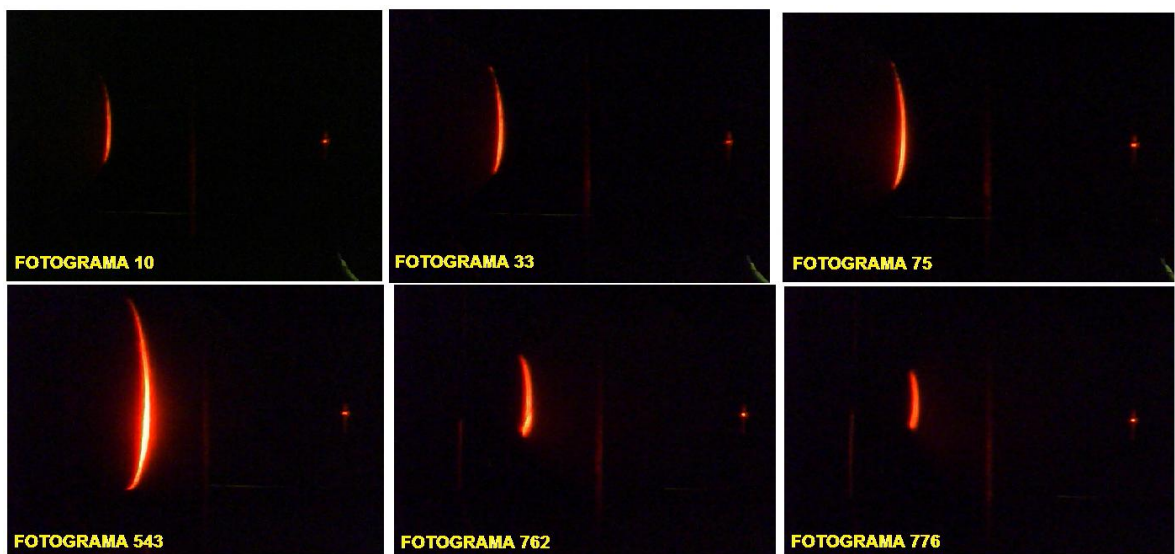


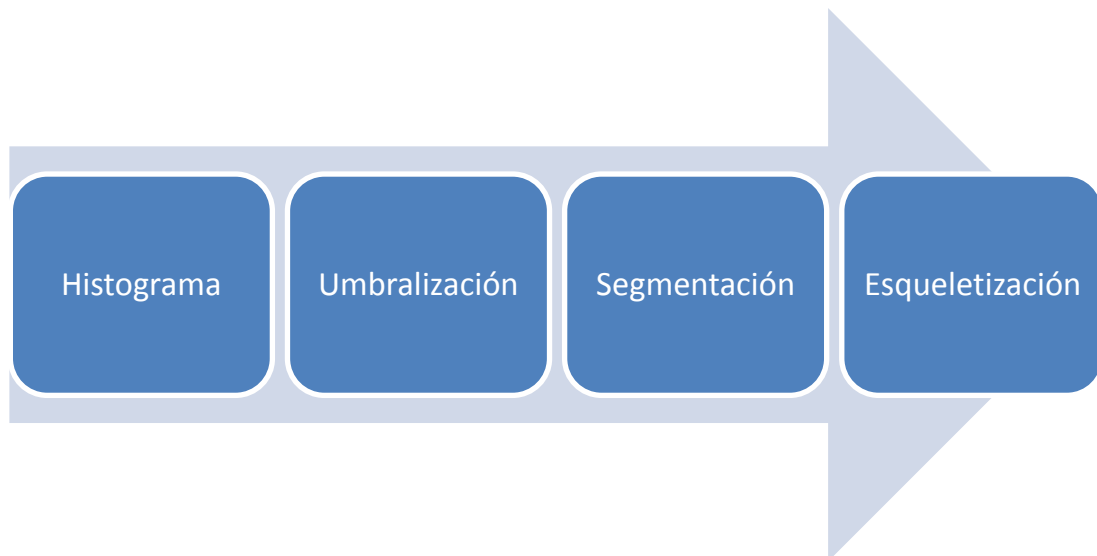
Figura 27. Perfiles del objeto esfera capturados en el proceso de escaneado.



### 3. PROCESAMIENTO DE IMÁGENES: ALGORITMOS IMPLEMENTADOS

En la sección 2.4 se explicó el proceso de captura de imágenes que culmina con el almacenamiento de las mismas en una carpeta específica del computador. En esta etapa inicia el procesamiento de éstas. Este proceso tiene que ver con el conjunto de operaciones realizadas sobre cada una de ellas, con el fin de abstraer las características relevantes, que permitan realizar los cálculos necesarios para obtener las coordenadas espaciales del objeto a reconstruir. La información útil de las imágenes hace referencia a las líneas de luz (nombradas de ahora en adelante *perfiles*) que proyecta el láser y que se deforman al reflejarse sobre el objeto escaneado. Por lo tanto las operaciones de procesamiento de imágenes aplicables a este proyecto tienen que ver con la selección y extracción de dichas líneas deformadas considerando el resto de información de la imagen como ruido. Las fases que hacen posible la selección y extracción de los *perfiles* de las imágenes se muestran en la figura 28.

Figura 28. Etapas del proceso de abstracción de información de las imágenes.



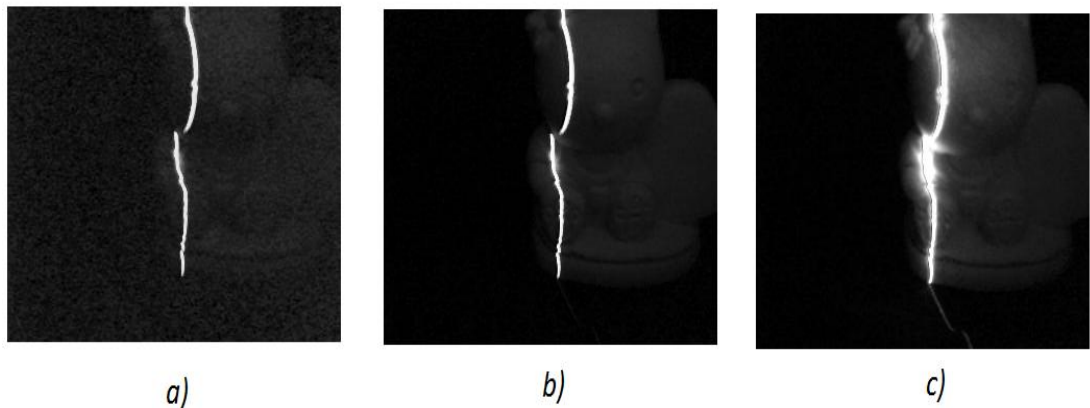
Estas fases se desarrollan a continuación.

### 3.1. HISTOGRAMA

En esta fase se realiza el respectivo análisis a cada imagen tomada en la etapa de captura de imágenes (ver sección 2.5, figuras 24, 25, 26 y 27) buscando la información de la misma que tenga mayor intensidad, debido a que esta información es la del *perfil* a extraer. Para esto se buscan los valores de los píxeles que más se repitan en la imagen, aplicando la teoría de histograma expuesta en la sección 1.4.

Antes de esto es necesario escoger el canal de la imagen que almacene la mayor información y que a su vez esta información tenga la menor cantidad de ruido. Es de aclarar que una imagen está compuesta por tres canales: rojo, verde y azul, conocidos por sus siglas como RGB. En la figura 29 se muestra un *perfil* tomado donde la imagen se dividió en los tres canales.

Figura 29. Canales de una imagen, a) Canal rojo de la imagen, b) Canal verde de la imagen, c) Canal azul de la imagen.



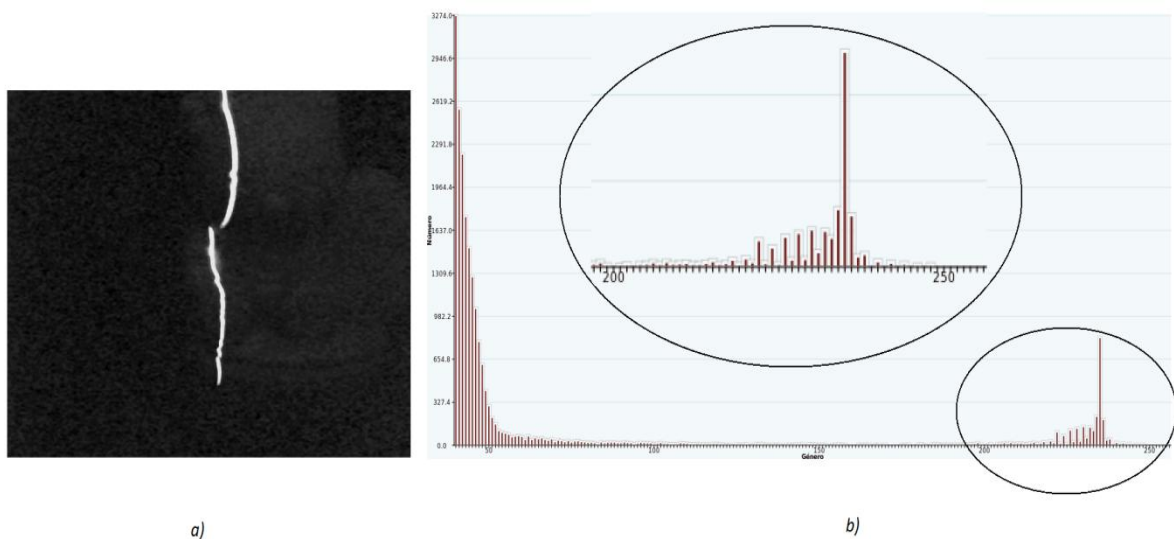
De la figura 29 se puede observar que el *perfil* del canal rojo está bien definido y no presenta grandes iluminaciones en sus bordes, el *perfil* del canal verde también está bien definido pero presenta iluminaciones en sus bordes, el *perfil* del canal azul está totalmente iluminado. Por lo tanto el mejor *perfil* se encuentra en el canal rojo ya que las iluminaciones de los demás *perfiles* representan ruido y sería



complicado eliminarlo en las fases posteriores. Este análisis es aplicable para todas las imágenes capturadas de los diferentes objetos escaneados debido a que en este trabajo se utilizó un solo láser.

Al aplicar el histograma a una imagen capturada se obtiene la información que se muestra en la figura 30. En la figura 30 (a) se encuentra la imagen de entrada y en la figura 30 (b) su respectivo histograma, donde se puede notar que los niveles de grises están concentrados de 0 a 50 con mayor tendencia en las intensidades cercanas a 0, confirmando así el predominio de los colores negros, que equivale al fondo de la imagen, sin embargo en este caso la información útil tiene que ver con los pixeles blancos que dibuja la línea láser, los cuales se encuentran en el intervalo 200 a 245 como se aprecia en esta figura.

Figura 30. a) Imagen de entrada, b) histograma de la imagen.



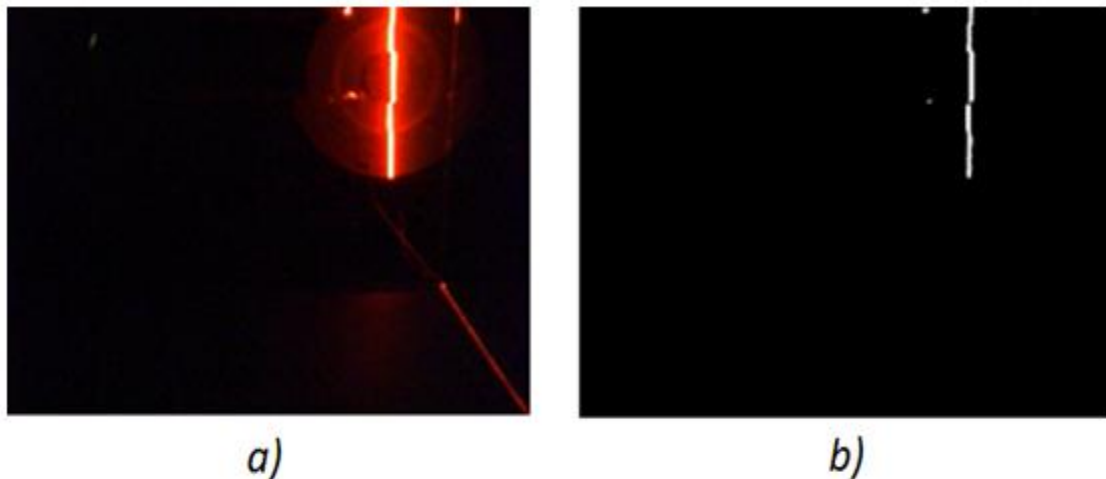
El cálculo del histograma fue necesario hacerlo una sola vez porque la intensidad y potencia del haz de luz con que se escanearon los objetos se mantuvieron constantes. Con este cálculo se obtuvo el rango en que se encuentran los *perfiles* en las imágenes. Este rango se definió entre 180 y 255.

### 3.2. UMBRALIZACIÓN

Esta fase permite reducir la información de la imagen a píxeles de color negro (0) y blanco (255), quedando así identificados los objetos dentro de la imagen con color blanco. Esto facilita el análisis de la información y aumenta el rendimiento de los algoritmos que procesen imágenes. Para realizar la umbralización o binarización se necesita el rango de píxeles calculado en la fase anterior (180 - 255). Al aplicar la umbralización los píxeles que se encuentren en este rango tomarán intensidades de 255 y aquellos que estén por fuera de éste tomarán valores de 0, como se ilustra en la sección 1.4. Con esta fase también se consigue que parte del ruido de la imagen sea eliminado (aquellos píxeles cuyos valores se encuentren fuera del rango calculado en la fase de histograma).

Como se puede apreciar en la figura 31 al aplicar la umbralización se han eliminado componentes de ruido causados por la refracción del haz de luz en los medios en los que se encuentra el objeto (tapa plástica). Este ruido corresponde a las líneas medianamente intensas de la imagen (figura 31 (a)), sin embargo no se eliminó por completo todo el ruido, existen pequeños puntos blancos que aparecen en la imagen umbralizada (ver figura 31 (b)).

Figura 31. *a)* Imagen de entrada, *b)* imagen umbralizada.



Del mismo modo en la figura 32 se muestra otra imagen que se ha umbralizado, la cual tiene ruido de alta y media intensidad. El ruido de media intensidad fue



eliminado, pero el de alta intensidad se mantuvo en la imagen umbralizada. Este ruido se muestra en la figura 32 (b).

Por último en la figura 33 se muestra otra imagen de entrada y la imagen umbralizada. En esta figura se puede apreciar que la imagen umbralizada (figura 33 (b)) está libre de ruido debido a que el valor de éste está por fuera del rango del umbral establecido.

Figura 32. Imagen de entrada, *b*) imagen umbralizada.

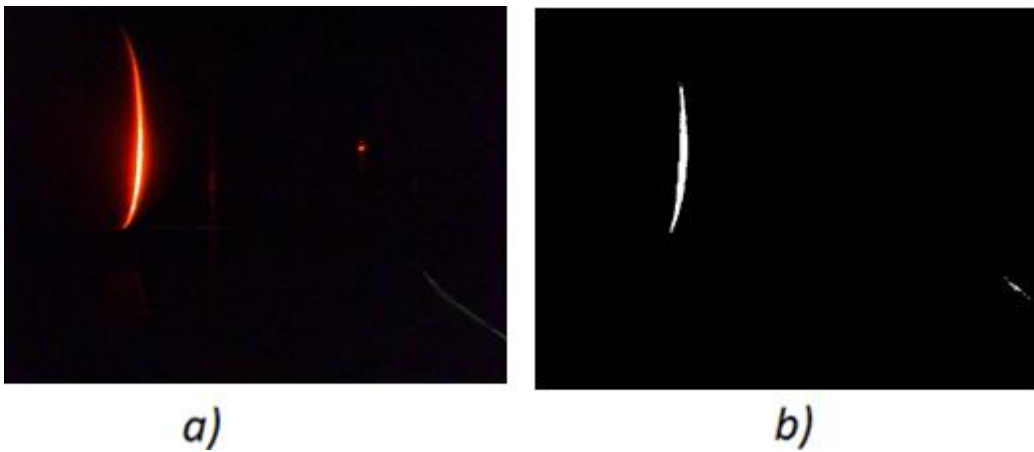
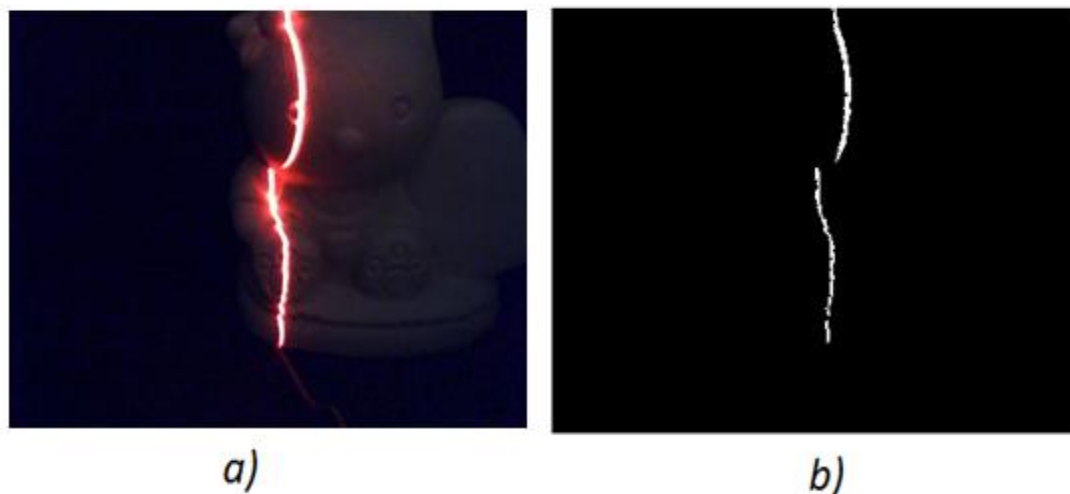
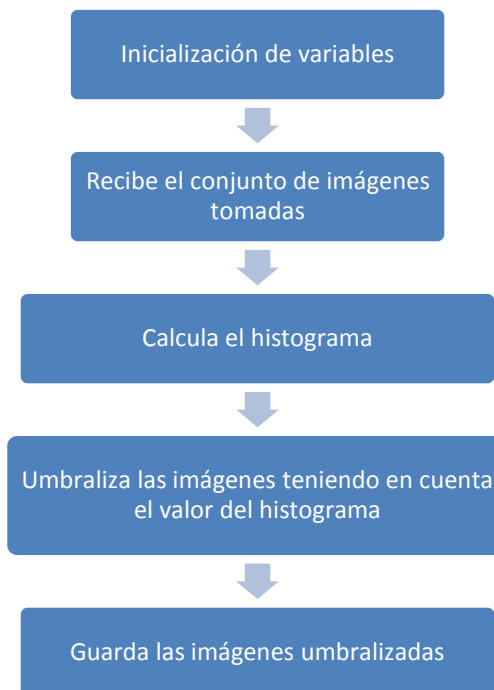


Figura 33. *a*) Imagen de entrada, *b*) imagen umbralizada.



Para procesar las imágenes y someterlas a las fases: histograma y umbralización se desarrollo un algoritmo el cual se describe en la figura 34 por medio de un diagrama funcional.

Figura 34. Diagrama funcional del algoritmo de umbralización.



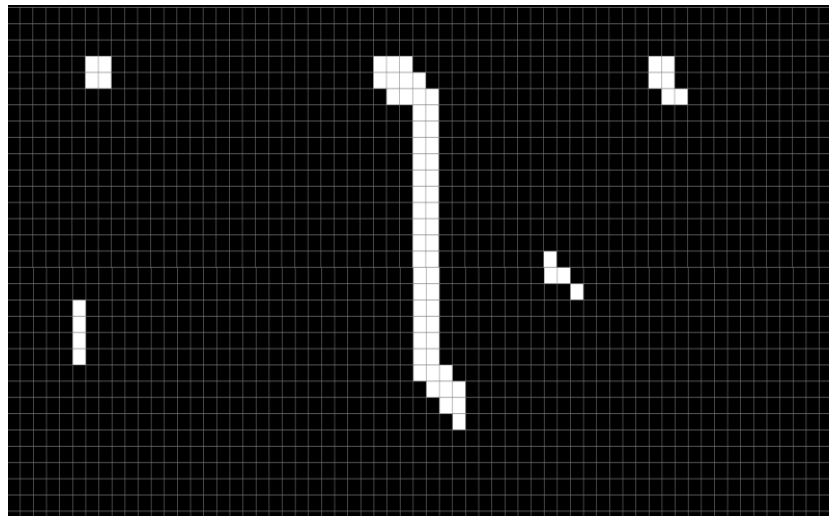
### 3.3. SEGMENTACIÓN

En esta fase se mejora la definición de los *perfiles* que sufrieron algún tipo de fragmentación leve o que se tornan borrosos por causa de la umbralización. También se elimina todo el ruido de alta intensidad que no se pudo eliminar en la fase previa. Es fundamental garantizar que se elimine todo el ruido de las imágenes porque de las mismas se extrae la información para calcular las coordenadas tridimensionales. Si esto no se garantiza, en las coordenadas tridimensionales calculadas existirán puntos que representan a dicho ruido y en el momento de realizar la renderización aparecerá éste en el modelo 3D obtenido. Si el ruido es muy grande causará que el modelo obtenido se deforme y no se pueda obtener una representación adecuada del objeto real.

Para arreglar los *perfiles* borrosos o levemente agrietados se utilizan las operaciones morfológicas Dilatación y Erosión explicadas en la sección 1.4.3. Para eliminar el ruido se desarrolló un algoritmo que calcula la región donde se encuentra el *perfil* (región de interés), establece un contorno de esta región y elimina toda la información (píxeles blancos) que esté por fuera de este contorno.

Para explicar la lógica de este algoritmo se utilizan varios objetos ficticios de ejemplo. En la figura 35 se muestra una imagen la cual contiene un objeto grande y cuatro objetos pequeños. El objeto grande representa la información que se quiere conservar y los objetos pequeños representan el ruido de la imagen. Cada cuadrado de la imagen representa un píxel, los cuadrados negros son píxeles con valor 0 y los cuadrados blancos son píxeles con valor 1.

Figura 35. Imagen con objetos ficticios.



En la figura 36 se muestra, además de los objetos anteriores, una zona de color rojo de tamaño 5x3. Esta zona se utiliza para recorrer la imagen buscando píxeles de color blanco. Como se aprecia en la figura 36 el algoritmo ubica la zona en el primer píxel blanco que encuentra, en ese momento evalúa que dentro de la zona existan “n” o más píxeles de color blanco (en la práctica  $n \geq 8$ ), si la condición se cumple se guarda la posición  $(u, v)$  del píxel donde se ubicó la zona. Con esta lógica se recorre toda la imagen.

Con las posiciones  $(u, v)$  guardadas de los pixeles, se hace el cálculo de las posiciones máximas y mínimas, para luego generar un contorno rectangular donde los vértices sean las coordenadas mínimas y máximas calculadas. En la figura 37 se muestra el contorno generado.

Por último, el algoritmo elimina todos los pixeles blancos que se encuentran por fuera del contorno generado. Esto se aprecia en la figura 38.

Figura 36. Zona definida para recorrer la imagen.

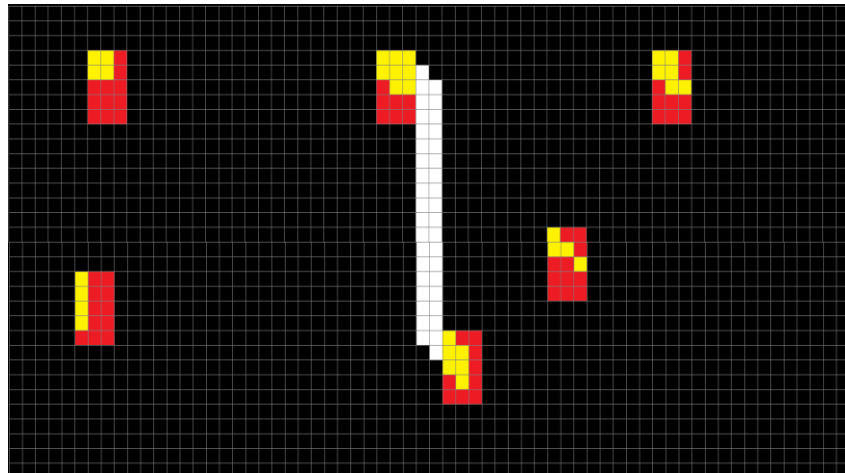
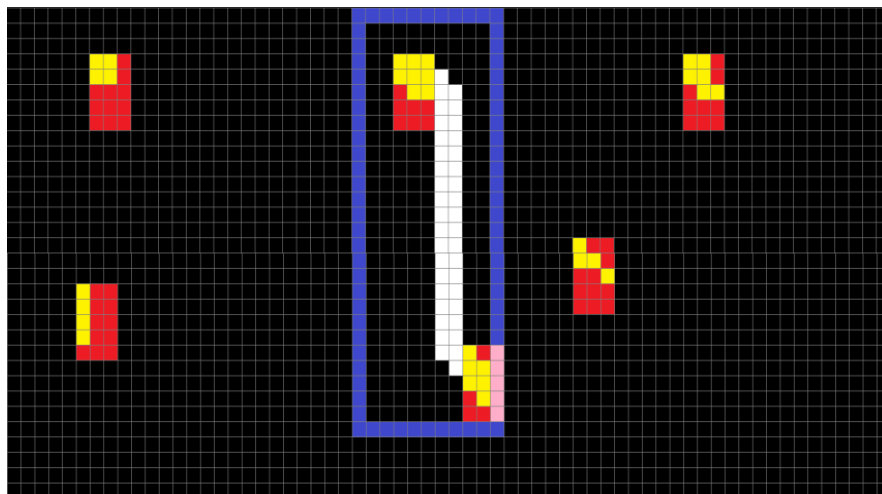
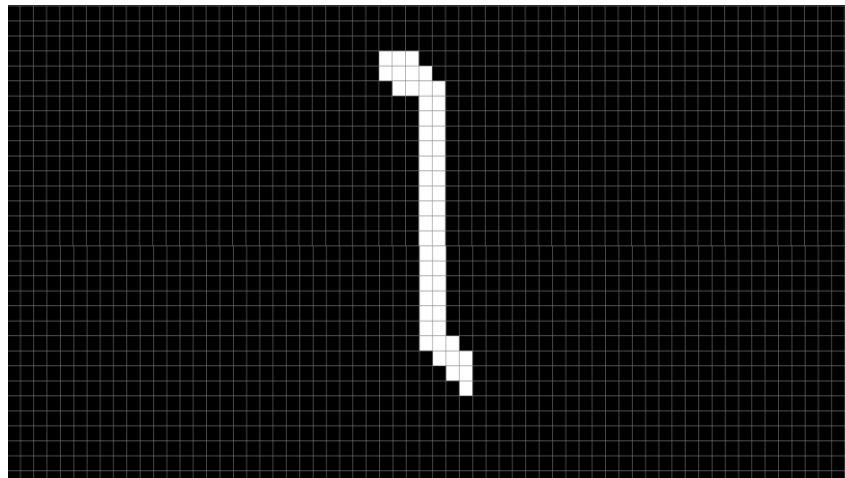


Figura 37. Imagen con contorno generado.



Los objetos con menor número de píxeles blancos fueron eliminados debido a que cuando la zona se posicionó en cada uno de sus píxeles no se cumplió la condición ( $n \geq 8$ ) definida anteriormente. El tamaño de la zona y el valor mínimo que toma "n" se definieron de acuerdo a las características del ruido inmerso en las imágenes capturadas. Si el ruido tuviese similar número de píxeles blancos que el número de píxeles blancos de los *perfiles* en las imágenes el algoritmo no podría eliminar dicho ruido. En la figura 39 se muestra el diagrama funcional del algoritmo implementado.

Figura 38. Imagen sin ruido.



En las figuras 40, 41 y 42 se muestran los resultados obtenidos en esta fase de implementación para los mismos *perfiles* umbralizados en la etapa previa. Nótese que en las figuras 40 (b), 41 (b) y 42 (b) se han obtenido imágenes con *perfiles* mejor definidos y en las figuras 41 (b) y 42 (b) se han corregido separaciones leves de los *perfiles* que se muestran en las figuras 41 (a) y 42 (a).

Además de lo anterior se obtienen nuevas imágenes sin ningún componente de ruido. Esto se puede apreciar en las figuras 40 (b) y 41 (b) donde se han eliminado los pequeños puntos que aparecen en las figuras 40 (a) y 41 (a).

Figura 39. Diagrama funcional del algoritmo de segmentación.

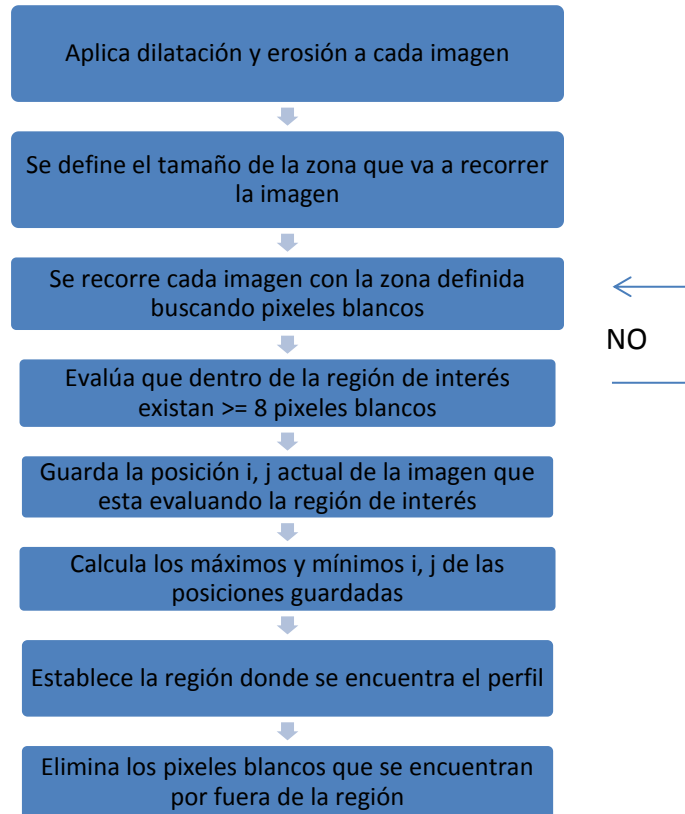


Figura 40. a) Imagen umbralizada, b) imagen segmentada.

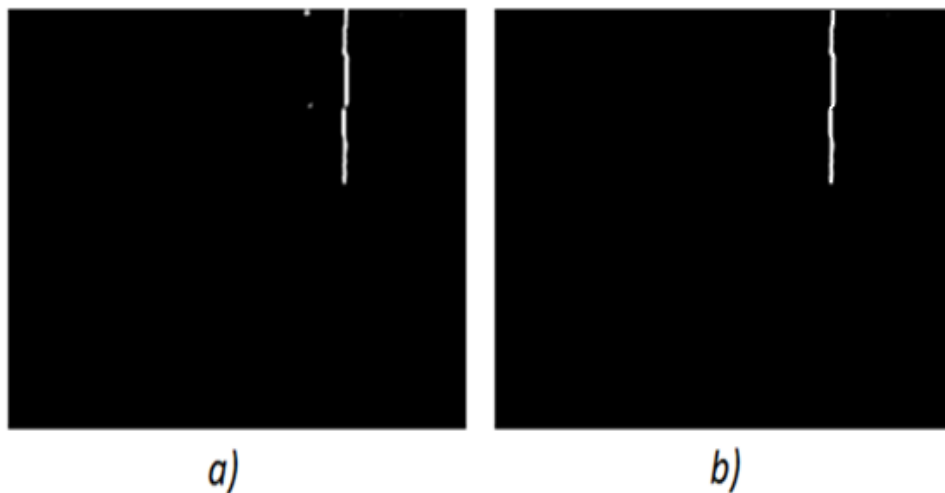


Figura 41. a) Imagen umbralizada, b) imagen segmentada



Figura 42. a) Imagen umbralizada, b) imagen segmentada.



### 3.4. ESQUELETIZACIÓN

Como se explicó en la sección 1.4.3, el proceso de esqueletización pretende obtener un *perfil adelgazado* del *perfil* original garantizando que el primero conserve la forma del segundo. En este ámbito existen algoritmos que permiten resolver este requerimiento. Estos están basados en reglas para la eliminación de píxeles blancos de manera iterativa hasta que ya no haya más cambios por realizar. En esta fase el resultado debe tener tres propiedades: que el nuevo *perfil* sea tan delgado como sea posible, que esté conectado y que coincida con el centro del *perfil* original.

Los algoritmos de esqueletización existentes varían en cuanto al tiempo de procesamiento de las imágenes y en cuanto a resultados satisfactorios. Teniendo en cuenta esta premisa, se eligió el algoritmo de Zhang – Suen [24] el cual ha sido utilizado en muchas aplicaciones que tienen que ver con el procesamiento digital de imágenes, ya que éste proporciona sencillez en su implementación, rendimiento y sobretodo se obtienen buenos resultados.

El algoritmo de Zhang Suen [24] usa una máscara (zona pequeña de la imagen) que recorre toda la imagen analizando en cada recorrido los píxeles blancos, bajo ciertas reglas y condiciones básicas, para determinar si dichos píxeles son borrados (volverlos negros) de la imagen o por el contrario si se mantienen (de dejan blancos) en la misma. Este algoritmo es reiterativo, y por cada iteración realiza dos recorridos a la imagen, reduciendo de manera secuencial píxeles de los bordes de los *perfiles* [24]. En el Anexo E se explica el funcionamiento de este algoritmo.

Se desarrolló un algoritmo basado en el algoritmo de Zhang Suen [24] para realizar el proceso de esqueletización a las imágenes previamente segmentadas. En la figura 43 se describe el diagrama funcional de este algoritmo.

Con la culminación de esta fase se obtienen los *perfiles* de las imágenes completamente esqueletizados. El algoritmo que se desarrolló permite obtener *perfiles* de un solo píxel de ancho. Esto es importante ya que se tiene información limpia y no redundante por cada punto de cada *perfil*.

En las figuras 44, 45 y 46 se evidencia el resultado obtenido al aplicar el algoritmo de esqueletización. En las figuras 44 (a), 45 (a) y 46 (a) se muestran las imágenes capturadas sin haberles hecho ningún procesamiento. En las figuras 44 (b), 45 (b) y 46 (b) se muestran las imágenes que han pasado por las fases de



histograma, umbralización y segmentación. En las figuras 44 (c), 45 (c) y 46 (c) se muestran las imágenes que ha pasado por la fase de esqueletización.

Cabe aclarar que todas las imágenes obtenidas en la etapa de captura de imágenes, descrita en la sección 2.4, son procesadas por medio de las cuatro fases explicadas en este capítulo.

Figura 43. Diagrama funcional del algoritmo de esqueletización.

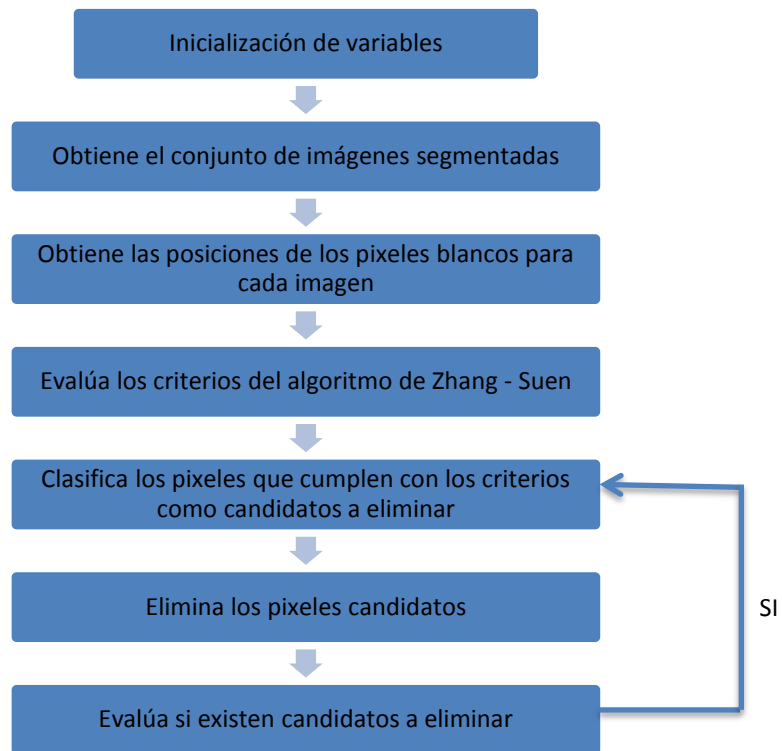


Figura 44. a) Imagen de entrada, b) imagen segmentada, c) imagen esqueletizada.

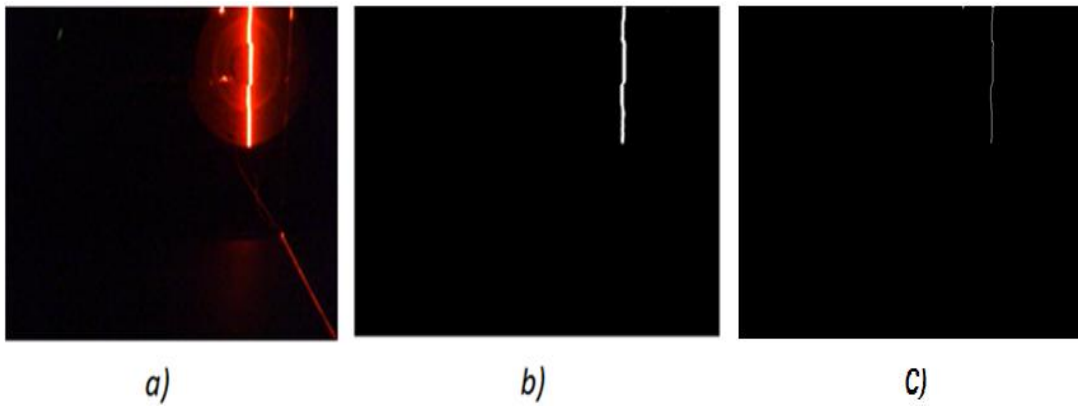


Figura 45. a) Imagen de entrada, b) imagen segmentada, c) imagen esqueletizada.

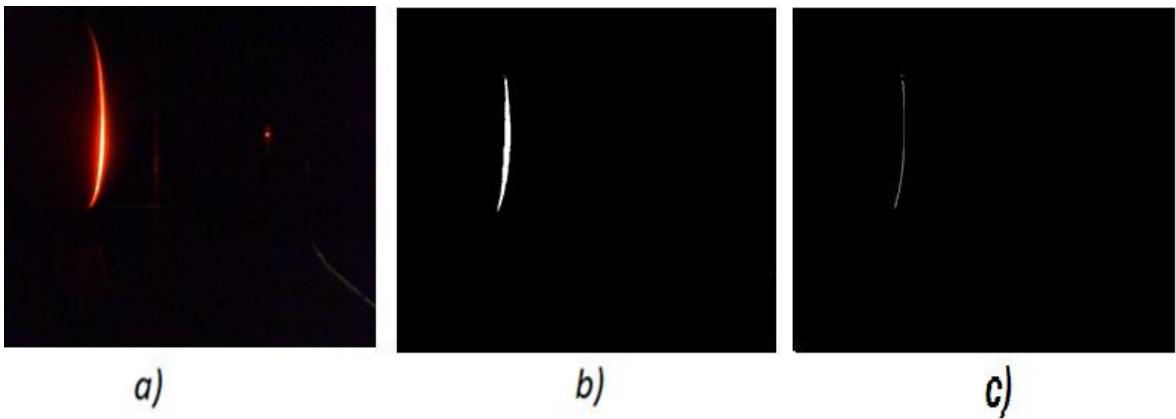
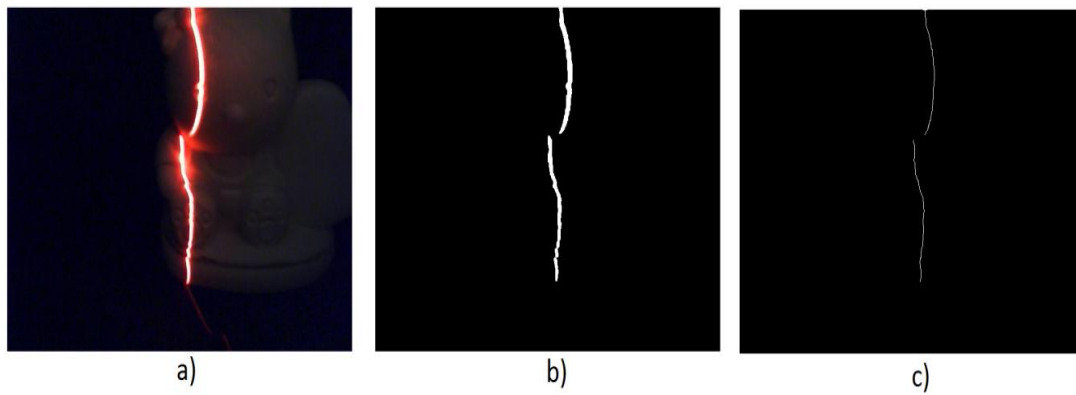


Figura 46. a) Imagen de entrada, b) imagen segmentada, c) imagen esqueletizada



## 4. OBTENCIÓN DE LA NUBE DE PUNTOS Y RENDERIZADO

En esta etapa de desarrollo del proyecto se calculan las coordenadas tridimensionales de los objetos escaneados y se generan los modelos 3D que representan a estos objetos. Se debe recordar que hasta el momento se tienen los *perfiles* tratados (esqueletizados) que poseen puntos con coordenadas  $(u, v)$  en píxeles. Para contar con las coordenadas 3D del objeto es necesario recuperar la coordenada que se perdió en las etapas anteriores, esta coordenada es la profundidad. Con la técnica de reconstrucción 3D (triangulación láser) implementada se consigue lo anterior y se obtienen las coordenadas  $(X, Y, Z)$  de los puntos que se encuentran en cada *perfil*.

### 4.1. CÁLCULO DE LAS COORDENADAS TRIDIMENSIONALES DEL OBJETO

Con las imágenes esqueletizadas (*perfiles* del objeto adelgazados y sin ruido) se utiliza el algoritmo desarrollado para obtener las coordenadas  $(u, v)$  de cada *perfil*. En la figura 47 se observa un *perfil* al cual se le va a calcular las coordenadas de los puntos pertenecientes a dicho *perfil*. Para conseguir esto, el algoritmo encuentra la posición de cada píxel blanco y calcula la distancia a lo ancho (coordenada  $u$ ) y largo (coordenada  $v$ ) de ese píxel al centro de la imagen ( $u_0 = 320, v_0 = 240$ ), haciendo esto para todos los píxeles blancos de la imagen. Al final de este proceso se cuenta con una lista de puntos con sus coordenadas  $(u, v)$ . Seguido de esto, para cada punto de la lista se aplican las ecuaciones de la técnica triangulación láser (para mayor información referirse a la sección 1.1) y se realiza la transformación de coordenadas (2D a 3D) de ese punto. Al terminar este proceso se obtiene el conjunto de coordenadas tridimensionales de los puntos pertenecientes a cada *perfil*.

La sumatoria de todos estos puntos conforman la nube de puntos del objeto real necesaria para la fase de renderizado que se explica en la siguiente sección.

En la figura 48 se describen los diagramas funcionales de los algoritmos desarrollados que permiten obtener la nube de puntos de los objetos utilizados en este proyecto, para generar una representación 3D de los mismos.

Figura 47. Perfil al que se le va a calcular las coordenadas de sus puntos.

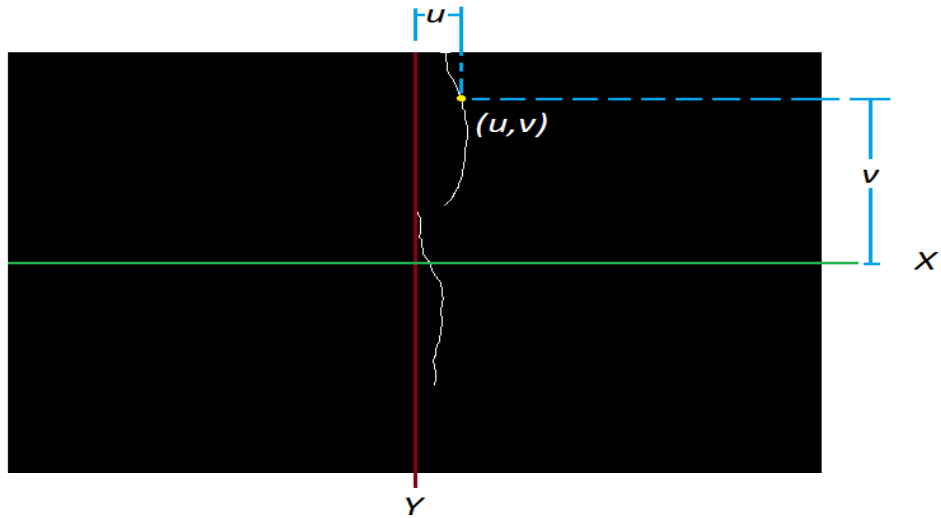
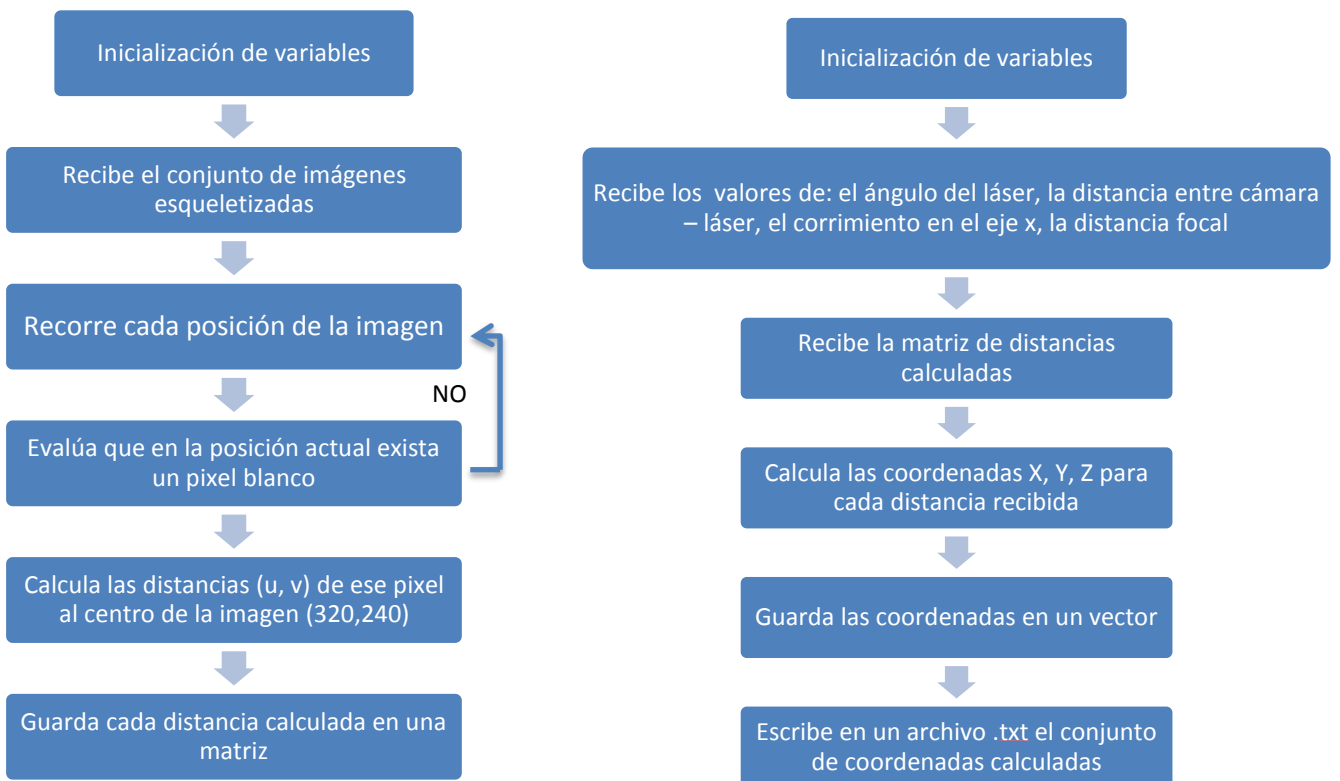


Figura 48. Diagramas funcionales de los algoritmos desarrollados.



## 4.2. RENDERIZADO

La renderización es el proceso a través del cual se genera una imagen en 3D a partir de un modelo, utilizando una herramienta software. El modelo representado es una descripción en tres dimensiones de un objeto del mundo real en una estructura de datos definida en un formato. El modelo puede contener información de la forma del objeto (geometría), textura e iluminación, sin embargo dentro de este proyecto solo se tendrá en cuenta la geometría.

El proceso de renderizado implementado al modelo obtenido consiste en dibujar polígonos entre los puntos o coordenadas espaciales que describen al objeto original. Para llevar a cabo este proceso existen muchas técnicas, sin embargo para este proyecto se trabajó con Triangulación de Delaunay [25], [26], y el VCG [27] el primero consiste en dibujar una red de triángulos con los puntos mientras que el segundo realiza relleno al modelo mediante la incursión de nuevos puntos dibujados alrededor de los existentes, que al unirlos con líneas forman las mallas que le dan textura al mismo.

En este trabajo se utilizó la herramienta Meshlab [28] para el renderizado, que es un software de fuente abierta y portable, que permite procesar la nube de puntos o coordenadas espaciales, mediante algoritmos de renderizado que trae la propia herramienta en los que se incluye la Triangulación de Delaunay y el VCG para obtener un modelo en 3D del objeto analizado. Los objetos que se van a reconstruir se muestran en la figura 23.

Inicialmente se importa desde Meshlab la nube de puntos del objeto. Éstos han sido generados en formato .PLY para permitir compatibilidad con la herramienta. La importación de estos puntos se hace seleccionando la opción *File* → *Import Mesh* como se ilustra en la figura 49.

Una vez importada la nube de puntos de los objetos desde el programa de Meshlab, se pueden visualizar en un esquema tridimensional, permitiendo detallar sus formas (ver figura 50).

Figura 49. Importación de la nube de puntos desde Meshlab.

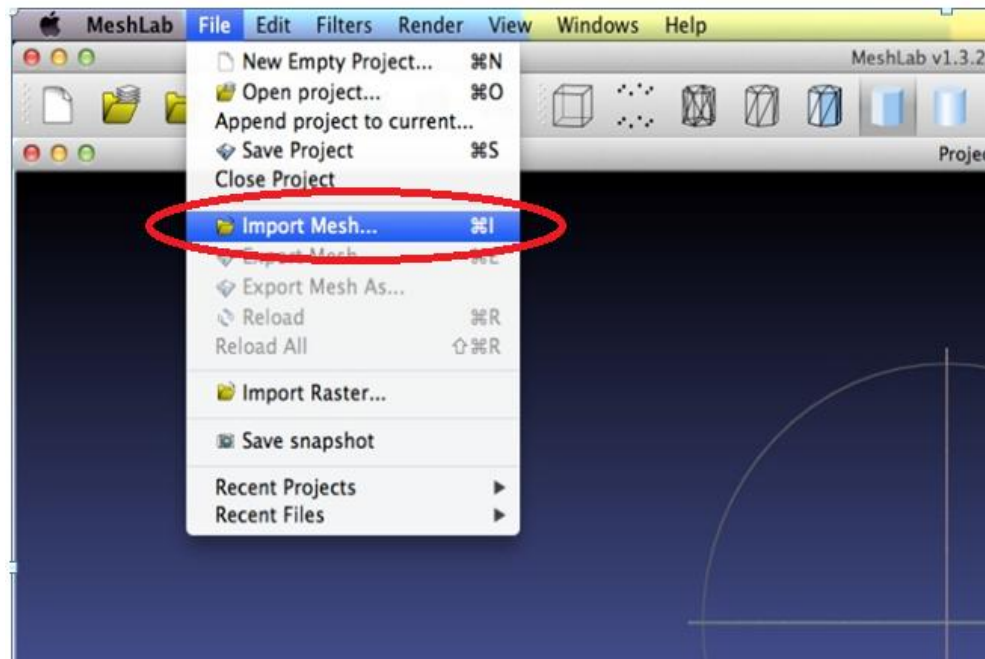
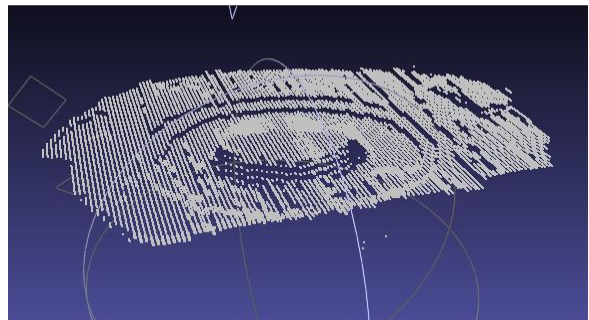
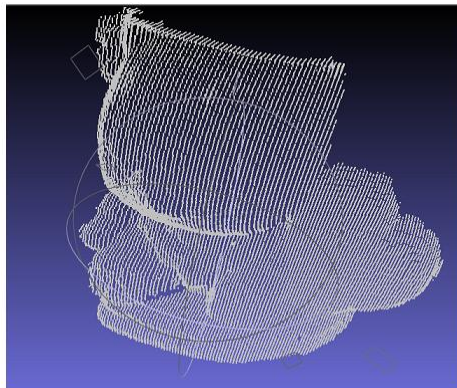


Figura 50. Vista de la nube de puntos desde Meshlab para el objeto gato y el objeto tapa, respectivamente.



Los pasos siguientes consisten en llevar a cabo los procesos de renderizado para obtener figuras sólidas. En vista de este hecho se ejecuta el algoritmo VCG con el fin de rellenar con nuevos puntos aquellos espacios que existen entre los perfiles importados, para lo cual se selecciona la opción *Filters* → *Remeshing*,

*simplification and Reconstruction* → *Surface Reconstruction: VCG* del mismo modo como se muestra en la figura 51. Una vez seleccionada esta opción aparece una ventana desde donde se habilita la opción *Vertex splatting* y finalmente se da click en *Apply* para ejecutar el algoritmo.

En la figura 52 se puede observar el resultado de aplicar el algoritmo VCG donde se detalla la nube de puntos más densa debido a los nuevos puntos que se han generado a partir de los existentes.

Una vez realizado lo anterior, el modelo se encuentra listo para ser renderizado, proceso que consiste en dibujar líneas entre los puntos del modelo para generar el relleno de los objetos. En la figura 53 se observa el resultado del modelo 3D obtenido para el objeto tapa, donde se puede detallar que conserva la forma del objeto original (ver figura 23). Las partes del modelo 3D donde existen agujeros y algunos de los bordes malformados se debe a información que se perdió en el procesamiento de las imágenes y a los sitios donde la luz láser no se pudo reflejar en el momento de escaneado.

Figura 51. Algoritmo VCG desde Meshlab.

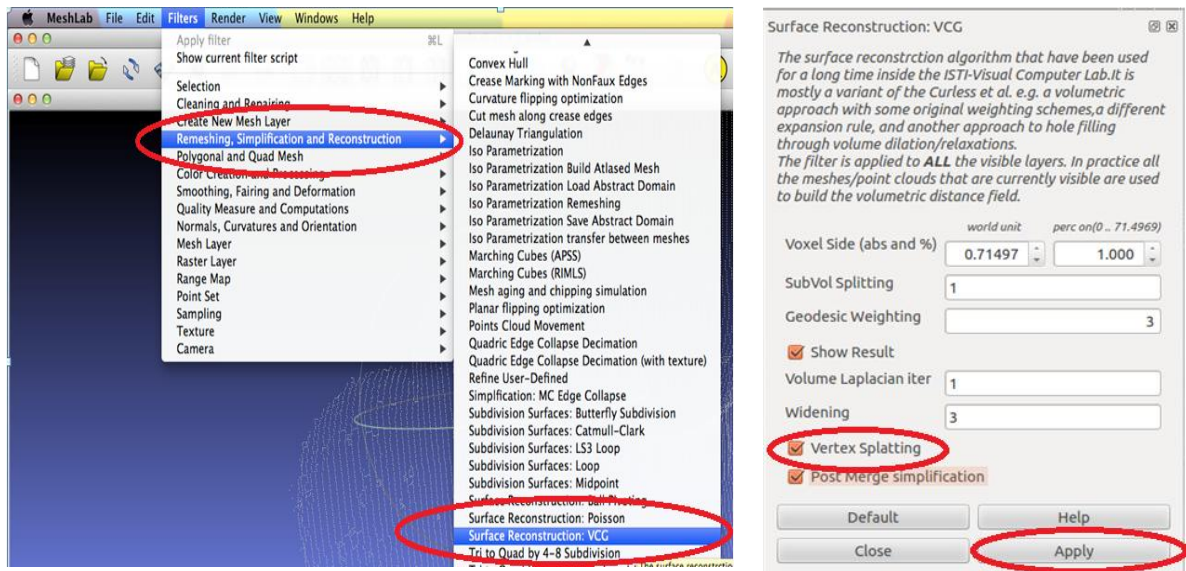


Figura 52. Resultado de aplicar el algoritmo VCG para el objeto esfera, el objeto rana y el objeto gato.

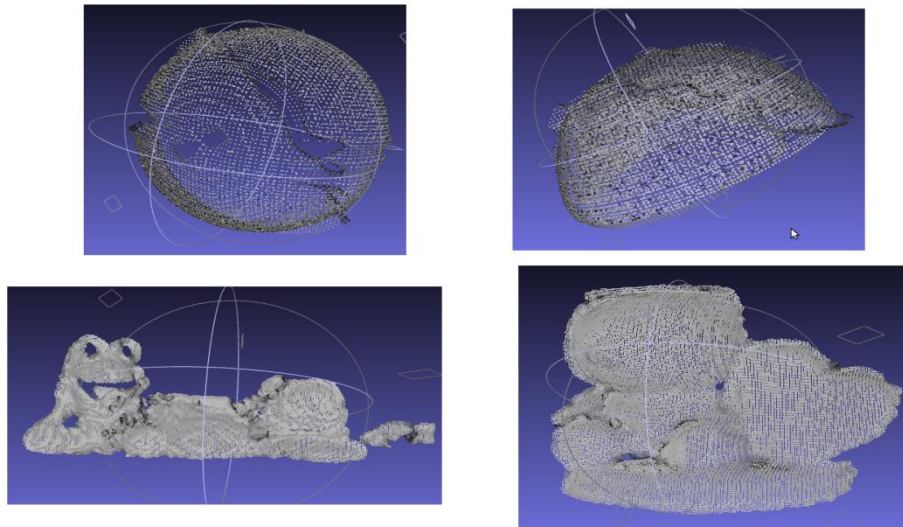
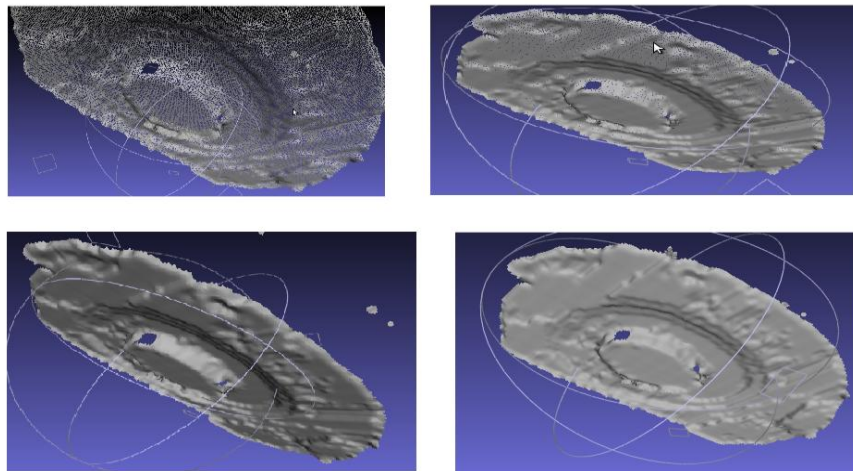


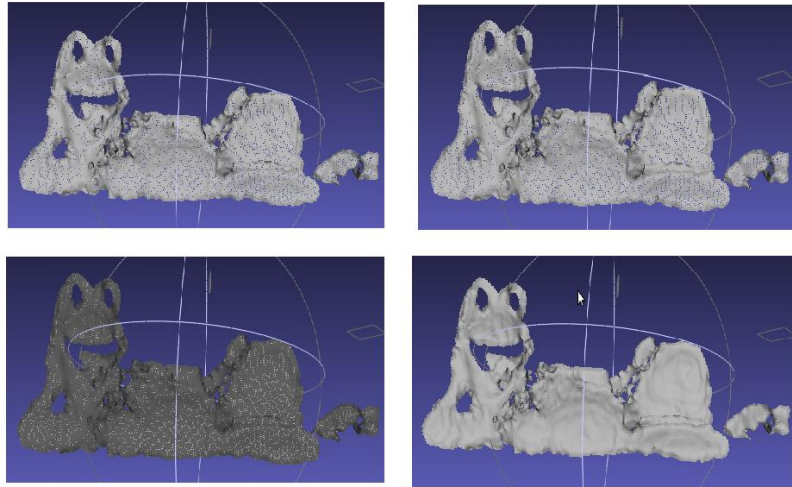
Figura 53. Resultado de aplicar el renderizado al objeto tapa.



En la figura 54 se muestra el resultado del renderizado para la nube de puntos del objeto rana (ver figura 23). Aquí se observa que existen regiones del objeto donde se han perdido los detalles, que corresponde a información difusa de la nube de puntos para estos lugares del objeto.

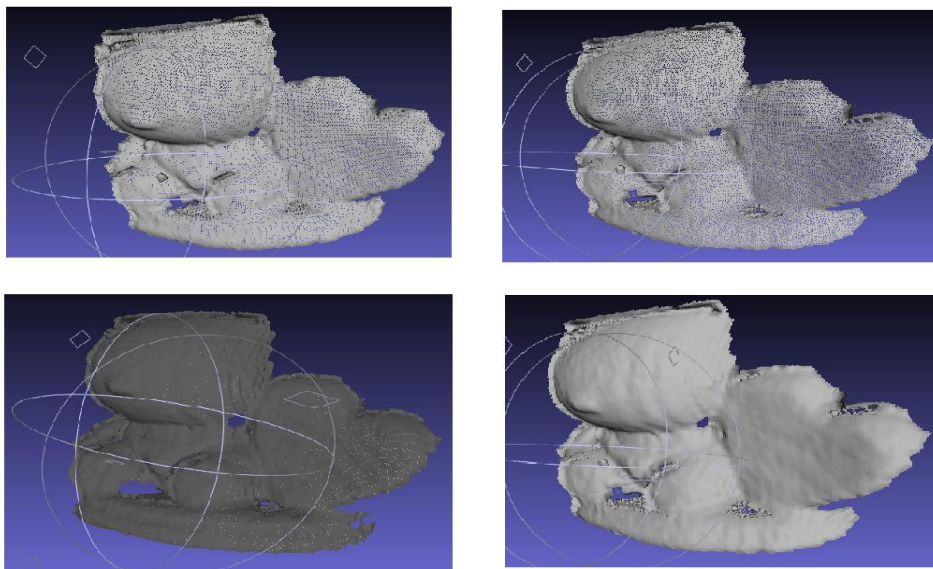


Figura 54. Resultado de aplicar el renderizado al objeto rana.



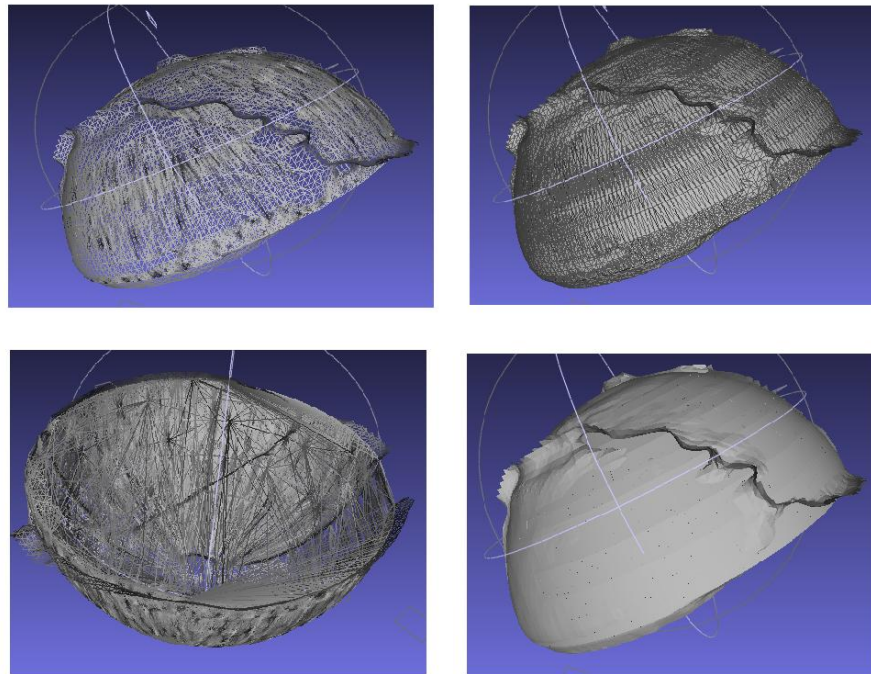
.Los resultados de renderizado para el objeto gato se puede visualizar en la figura 55, donde se observa que el modelo 3D obtenido conserva la forma del objeto original (ver figura 23), no obstante se han perdido detalles finos como los ojos y las huellas de los pies.

Figura 55. Resultado de aplicar el renderizado al objeto gato.



Para el renderizado del objeto esfera se realizó una variante ya que además de aplicar el algoritmo VCG para la adición de nuevos puntos, también se aplicó *Triangulación de Delaunay* desde la opción *Filters* → *Remeshing, simplification and Reconstruction* → *Delaunay Triangulation*. Lo anterior se hizo porque el objeto esfera es simétrico y al aplicar este último algoritmo formará un objeto sólido sin dañar la forma del mismo, lo cual no ocurriría con los demás objetos analizados puesto que al unir los puntos extremos para la formación de polígonos triangulares deforma la figura original de los objetos. Los resultados se muestran en la figura 56.

Figura 56. Resultado de aplicar el renderizado al objeto esfera.



## 5. ANÁLISIS DE RESULTADOS

En este capítulo se analizan todos los resultados que se obtuvieron en cada una de las etapas de desarrollo propuestas en este trabajo.

### 5.1. CALIBRACIÓN DE LA CÁMARA

Se pudo comprobar que los valores de los parámetros estimados son muy sensibles a variaciones cuando se modifican las características del patrón de calibración (tablero de ajedrez), o cuando se cambia la posición y orientación relativa del mismo respecto al sistema de referencia de la cámara.

En este trabajo no se tuvieron en cuenta los parámetros extrínsecos de la cámara debido a que la orientación del objeto respecto a la cámara, o la orientación de la cámara respecto al objeto se mantuvieron fijas. La posición relativa del objeto respecto a la cámara por cada perfil se calculó utilizando la técnica de triangulación láser descrita en la sección 1.1 del capítulo 1.

Del análisis de los resultados obtenidos se concluyó:

- Se obtiene una buena estimación de la distancia focal  $f$  y de los factores de relación horizontal y vertical pixel/milímetro  $k_u$  y  $k_v$ , cuando la diferencia entre los valores de los factores  $f * k_u$  y  $f * k_v$  tiende a ser cero.
- Si el patrón de calibración (tablero de ajedrez) se ubica a pocos centímetros de la cámara se obtiene una buena estimación de los valores de la distancia focal y de la relación de aspecto, en contra parte se obtiene una mala estimación del valor del centro óptico. Del mismo modo, si el patrón de calibración se encuentra a una distancia mayor a un metro la estimación de los valores de la distancia focal y relación de aspecto tiende a ser errónea mientras que la estimación el valor del centro óptico es muy buena. La distancia óptima a la que debe estar el patrón de calibración de la cámara, para obtener la mejor estimación de los parámetros intrínsecos, es a un metro.
- Para la cámara utilizada en este proyecto se determinó que con un número de capturas menor o igual a 14, los valores de los parámetros intrínsecos tienden a ser muy variantes y se alejan de los valores ideales. Con un número de capturas mayor o igual a 15 se obtienen valores adecuados.

Después de 15 capturas, el número de éstas no influye en el mejoramiento en la estimación de los parámetros intrínsecos. Se debe aclarar que estos resultados no se generalizan para todas las cámaras digitales.

- Se determinó que las posiciones del patrón de calibración que mejores resultados arrojan son las esquinas de la vista de la cámara.

Para consulta de los resultados obtenidos en esta etapa referirse al Anexo A.

## **5.2. PROCESAMIENTO DIGITAL DE IMÁGENES**

### **5.2.1. CAPTURA DE IMÁGENES**

Se pudo observar que esta etapa del proceso de reconstrucción 3D es crucial en la obtención de las coordenadas  $(X, Y, Z)$  de los *perfiles* tomados del objeto a reconstruir, debido a que la técnica de reconstrucción 3D utilizada es muy sensible y dependiente de esta etapa.

Debido a que esta técnica es óptica, son determinantes las condiciones de iluminación de la escena donde se encuentra el objeto a reconstruir. Se estimó que una buena iluminación de la escena no es adecuada para la obtención de las imágenes (*frames*) que contienen los *perfiles* del objeto. Si el sistema de captura de imágenes (cámara - láser) se encuentra con buena iluminación, ejemplo la luz del sol, la cámara captura los *perfiles* formados por el haz de luz láser, ya que tienen buena intensidad, y todas las partes de la escena que tienen similar intensidad. Estas partes tienen valores de píxeles muy similares a los valores de los píxeles de los *perfiles*, causando que en el momento de umbralizar la imagen (establecer en 255 (color blanco) los valores de los píxeles que estén por encima del valor de umbral y en 0 (color negro) los valores de los píxeles que estén por debajo del valor de umbral), se obtengan varios objetos de color blanco (los *perfiles* y otros objetos de la escena). Con esto, el algoritmo de cálculo de coordenadas espaciales arroja las coordenadas de los *perfiles* del objeto y las coordenadas del ruido en la imagen. En el momento de realizar la renderización (la unión de los puntos por medio de mallas, etc.) el sistema intenta unir los puntos de los *perfiles* con los del ruido de la imagen y se obtendrán deformaciones del objeto, y en el peor de los casos no se puede hacer la reconstrucción.

Por lo tanto lo ideal es que la escena donde se encuentra el sistema de captura de imágenes esté estrictamente controlada, en este caso se buscó que la única fuente de luz que ilumine la escena sea el haz de luz emitido por el láser. Con esta condición se consiguió obtener imágenes con muy poco ruido.

Una alternativa que ayudó en cierta medida fue generar el haz de luz del láser con una mayor potencia para elevar la intensidad de los *perfiles* formados del objeto, de tal forma que esta intensidad sobrepase la intensidad del resto de la escena capturada por la cámara. En el momento de umbralizar se pudo trabajar con un rango de umbral bajo (cercano al máximo valor que es 255) y así se evitó en gran medida introducir ruido de mediana y baja intensidad en la imagen.

### **5.2.2. UMBRALIZACIÓN DE IMÁGENES**

De acuerdo al análisis de las imágenes tomadas se determinó que el valor de umbral óptimo para eliminar el ruido de las imágenes y a la vez garantizar que no se perdiera información de los perfiles del objeto fuese de 180. Con este valor se consigue que los valores de pixeles que estén en el rango de 180 a 255 se conserven en la imagen y los valores de pixeles que estén por debajo de este rango (entre 0 y 179) sean eliminados. Aunque con estas condiciones se consiguen buenos resultados, hubo componentes de ruido que lograron mantenerse después de aplicar el umbral.

### **5.2.3. SEGMENTACIÓN**

Las operaciones que se aplicaron en esta fase arrojaron buenos resultados. En primer lugar se evidenció que con la dilatación y erosión se mejoraron las definiciones de los *perfiles*. Con la generación del contorno del *perfil* se logró eliminar el ruido de las imágenes. Este hecho radica en que el ruido de las imágenes que llega a esta etapa es muy poco y el valor en pixeles es mucho menor que los valores en pixeles de las imágenes.

### **5.2.4. ESQUELETIZACIÓN**

El algoritmo de esqueletización que se desarrolló ofrece muy buenos resultados, es decir, recibe cada imagen capturada, procesa el perfil que tiene ésta y crea una imagen que contiene una línea (equivalente al perfil analizado) con valor de un

pixel en cada punto de la misma. Este resultado es el óptimo ya que se cuenta con un solo punto del mismo valor (en unidades de pixeles) para los N puntos que tienen los *perfiles*, lo cual se traduce en que no se repite el cálculo de coordenadas espaciales para un mismo punto.

### **5.3. CÁLCULO DE COORDENADAS ESPACIALES**

En el Anexo F se registran los resultados obtenidos del proceso de validación que se llevó a cabo para determinar si el sistema implementado en este trabajo, permite generar un modelo que represente en forma al objeto escaneado. Este proceso se dividió en dos etapas. En la primera se pudo determinar que al aplicar la técnica triangulación láser, el sistema calcula las coordenadas tridimensionales del objeto con un error cercado a los 3 milímetros. Aunque los valores calculados no son muy precisos, son suficientes para obtener un modelo 3D que representa en forma al objeto real. Este resultado se obtiene de la validación del sistema plasmada en el Anexo F donde se detalla el proceso que se llevó a cabo para obtener el error del sistema.

Con este resultado se puede decir que el análisis realizado en la sección 2.2 es correcto y que las imágenes aparentes del objeto que captura la cámara conservan las características de forma y tamaño, aunque no se tomen en la posición real del objeto escaneado.

Sin embargo para los valores cercanos a 90 grados del ángulo del haz de luz emitido por el láser, se observa que el error es cercano a un centímetro. Esto se debe a que en las ecuaciones de la técnica triangulación láser se cuenta con una función tangente, la cual no está definida para un ángulo de 90 grados. Se supone que para valores más cercanos a 90 grados el error es proporcional, lo que implica que este aumente su valor.

De la segunda etapa de validación se obtuvo un error alrededor de los dos milímetros. Este error permite ratificar que el fenómeno de refracción de la luz que ocurre entre diferentes medios (aire – vidrio - agua) trae como consecuencia que la cámara vea los objetos en una posición aparente, la cual es diferente a la posición real a la que se encuentra el objeto sumergido en el agua y que la forma de este objeto es invariante.

Por lo tanto al realizar el proceso de reconstrucción 3D de la vista aparente de este objeto, se obtiene los mismos resultados que si se hiciera la reconstrucción 3D de la vista real del mismo.

Una de las causas de los errores obtenidos se deben a que en la toma de datos se introdujeron errores sistemáticos que tienen que ver con muchos factores, tales como: la medición del ángulo, la precisión de medir manualmente la coordenada de un punto iluminado por la luz láser, así como de errores de la propia técnica de reconstrucción que tienen que ver con el cálculo de los parámetros intrínsecos de la cámara.

#### **5.4. RENDERIZADO**

Los resultados obtenidos en el proceso de renderizado para la obtención de los modelos 3D de los objetos analizados en el presente trabajo (ver Figura 44) muestran que a nivel general se consiguen buenos resultados (ver figuras 49,50, 51, 52). Los modelos construidos conservan la forma de los objetos originales, aunque con pequeñas variaciones que se deben a información que no pudo ser recuperada por la técnica implementada debido a varias causas. Los mejores modelos 3D encontrados son los del objeto esfera y el objeto gato. El primero se debe a que éste objeto no cuenta con detalles y es muy simétrico, mientras que el segundo a pesar de tener algunos detalles no es un objeto brillante por lo que se obtienen buenas imágenes de sus perfiles en el proceso de escaneado. Caso contrario ocurre con el objeto rana y tapa, ya que son objetos muy brillantes debido a sus materiales de fabricación, por tal motivo cuando entran en contacto con la luz en la captura de los perfiles que se obtienen en el proceso de escaneado, causan brillos y dispersiones de luz, razón por la cual la información de las coordenadas espaciales que se obtienen cuenta con información errónea que se ve reflejada en la textura de los modelos 3D obtenidos, no obstante su geometría y forma en gran parte se conservan.

## 6. TRABAJOS FUTUROS

Para la obtención de un modelo completo en la reconstrucción 3D se hace necesario implementar mecanismos que permitan realizar escaneos de 360° para abstraer todas las vistas laterales del objeto y así obtener un prototipo completo del mismo.

Es importante efectuar reconstrucción 3D en ambientes poco controlados, es decir en escenas donde la luz no sea un factor crítico o relevante para llevar a cabo este trabajo, para lo cual es necesario implementar otras técnicas en las que se destacan la visión estereoscópica, un sistema híbrido como visión estereoscópica con triangulación láser entre otras.

Un reto importante es el de realizar reconstrucción 3D de objetos sumergidos en agua en condiciones donde este fluido cuente con niveles de turbidez, con el fin de simular un medio real como un río o el mar.



## 7. CONCLUSIONES

La calibración de la cámara es importante como etapa inicial de un proyecto de reconstrucción 3D, ya que caracteriza la geometría interna de la misma, obteniendo así los parámetros necesarios para obtener una representación digital del objeto analizado mediante la aplicación de la técnica de Triangulación Láser.

El buen funcionamiento de la técnica de reconstrucción 3D utilizada en este trabajo depende demasiado de la iluminación de la escena donde se encuentra el objeto. Es necesario tener escenas con poca iluminación, razón por la cual se hace necesario controlar estas condiciones antes de llevar a cabo la reconstrucción cuando se utiliza técnicas de Triangulación Láser.

En el procesamiento de imágenes es necesario garantizar una buena esqueletización de los perfiles de las imágenes debido a que se necesita contar con un solo valor por cada punto de estos perfiles. De este modo se evita obtener datos repetidos que no aportan información relevante para la reconstrucción y que hacen más lento el cálculo de las coordenadas espaciales.

El ángulo de inclinación del láser respecto a la cámara no debe acercarse a valores de  $90^\circ$  ya que la relación de tangentes descritas en las ecuaciones de la técnica utilizada hace que los valores obtenidos tiendan a infinito alterando así su buen funcionamiento. Por ello es necesario ubicar el objeto analizado en una posición donde al calibrar la dirección del láser, éste no tenga valores cercanos a  $90^\circ$ .

Al momento de realizar el escaneo láser para la captura de los perfiles del objeto a reconstruir, es necesario que el agua se encuentre en reposo, ya que agua en movimiento causará desviaciones del haz de luz láser y de la imagen del objeto observado, causando así errores de medición que alteran los resultados obtenidos.

Los objetos a reconstruir no deben ser totalmente blancos o brillantes, porque causan que el haz de luz láser reflejado en el objeto deslumbre la vista del mismo y se pierda el perfil formado por el haz de luz emitido. Además el objeto debe ser pequeño debido a que se trabaja a distancias cortas y el rango de visión de la cámara es limitado.

## 8. REFERENCIAS

- [1] J. Suarez, L. Córdoba, H. Hurtado. “Un sistema sencillo y de bajo costo para la reconstrucción tridimensional de estructuras embrionarias a partir de cortes seriados”. Tesis Doctoral. Facultad de Ciencias Básicas. Universidad Militar Nueva Granada. Bogotá, Colombia 2007.
- [2] E. López, F. Ramírez, I. Romero, J. de la Garza & J.L. Castillo, “Diseño de un instrumento para la captura de geometrías 3D con aplicación en la manufactura de productos,” *Revista de la Sociedad Mexicana de Ingeniería Mecánica*, vol. 2, No. 3, pp. 101-110, Septiembre 2006.
- [3] Hypack Inc. “Hydrographic Survey and Processing Software”, [Online]. Disponible en: <http://www.hypack.com/new>. Consultado: Agosto de 2012.
- [4] L. Torres, J. Méndez, G. Dudek. “Color Correction of Underwater Images for Aquatic Robot Inspection”, en *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, pp. 60–73. 2005.
- [5] G. Telem, S. Filin. “Photogrammetric modeling of underwater environments”, *International Journal of Photogrammetry and Remote Sensing*, vol.65, pp. 433-444, Israel, 2010.
- [6] F. Collado. “New Methods for Triangulation-Based Shape Acquisition using Láser Scanners”. Tesis Doctoral. Departamento de Electrónica, Informática y Automática. Universidad de Girona, España, mayo 2004.
- [7] A. Gómez, D. Zamorano. “Visión Estereoscópica y Estimación de Pose para el Posicionamiento de un Brazo Robótico”. Tesis de Maestría. Departamento de Ingeniería Mecatrónica, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca Morelos, México, Abril 2008.
- [8] P. Alcantarilla. “6DOF localización y mapeado simultáneo (SLAM) en tiempo real mediante cámara estéreo movida con la mano”. Tesis de Maestría. Departamento de Electrónica, Escuela Politécnica Superior, Universidad de Alcalá, España, 2008.
- [9] E. López, F. Ramírez, I. Romero, J. de la Garza and J.L. Castillo, “Diseño de un instrumento para la captura de geometrías 3D con aplicación en la manufactura de productos”. *Revista de la Sociedad Mexicana de Ingeniería Mecánica*, Vol. 2, No. 3, pp. 101-110, septiembre, 2006.

- [10] J. Molleda, “Técnicas de Visión por Computador para la Reconstrucción en Tiempo Real de la Forma 3D de Productos Laminados”. Tesis Doctoral. Departamento de Informática, Universidad de Oviedo, Gijón España 2008.
- [11] J. Rouses, “Sistema de visión Estereoscópica Autocalibrado”. Trabajo fin de carrera. Ingeniería Técnica de Telecomunicación: especialidad Sonido e Imagen, Universidad Carlos III De Madrid, España, 2009.
- [12] Grupo de Control de Informática Industrial GCII, “Visión artificial: Luz estructurada” [Online]. Departamento de Electrónica, Automática e Informática Industrial, Universidad Politécnica de Madrid. Disponible en: <http://www.elai.upm.es/webantigua/spain/Investiga/GCII/gcii.htm>. Consultado: Julio 2012.
- [13] V. Soancatl, “Recuperación de Información tridimensional usando luz estructurada”. Tesis de Maestría en Ciencias Computacionales. Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla México, 2003.
- [14] E. López, F. Ramírez, I. Romero, J. de la Garza & J.L. Castillo, “Diseño de un instrumento para la captura de geometrías 3D con aplicación en la manufactura de productos”. *Revista de la Sociedad Mexicana de Ingeniería Mecánica*, Vol. 2, No. 3, pp. 101-110, septiembre, 2006.
- [15] R. Serway, R. Beichner, “La naturaleza de la luz y las leyes de la óptica geométrica”, en *Física para ciencias e ingeniería*, 3 ed., Vol. 2, V. Campos, México D. F., México: McGraw-Hill, 2002, pp. 1106 – 1138.
- [16] R. Serway, R. Beichner, “Óptica geométrica”, en *Física para ciencias e ingeniería*, 3 ed., Vol. 2, V. Campos, México D. F., México: McGraw-Hill, 2002, pp. 1138 – 1184.
- [17] G. Bradsky, A. Kaehler, “Learning OpenCV,” Primera Edición [Online]. Sebastopol: O’Reilly, Septiembre 2008. Disponible en: <http://www.cse.iitk.ac.in/users/vision/dipakmj/papers/OReilly%20Learning%20OpenCV.pdf>. Consultado: Agosto 2012.
- [18] R. Fisher, S. Perkins, A. Walker & E. Wolfart, “Image Processing Learning Resources: Intensity Histogram,” [Online]. Disponible en: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/histogram.htm> Consultado: Julio 2012.
- [19] IMAGine: Cursos Interactivos de Tratamiento Digital de Imagen. “Operaciones Morfológicas” [Online]. Disponible en:

[http://www.tsc.uc3m.es/imaginer/Curso\\_ProcesadoMorfologico/Contenido/Operaciones/OperacionesMorfologicas.html](http://www.tsc.uc3m.es/imaginer/Curso_ProcesadoMorfologico/Contenido/Operaciones/OperacionesMorfologicas.html). Consultado: Julio 2012.

[20] R. Fisher, S. Perkins, A. Walker & E. Wolfart, "Image Processing Learning Resources: Morphology" [Online]. Disponible en: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm>. Consultado: Julio 2012.

[21] H. López, A. Gómez, "Superficies a partir de nubes de puntos" [Online]. Computación Gráfica II, Universidad Simón Bolívar, Noviembre 2011. Disponible en: <http://ldc.usb.ve/~alacruz/cursos/ci5321/SurfacePointClouds/Presentaci%C3%B3n%20Te%C3%B3rica/Informe.pdf>. Consultado: Agosto 2012.

[22] Colaboradores de Wikipedia. Análisis DAFO [Online]. Wikipedia, La enciclopedia libre. Disponible en: [http://es.wikipedia.org/w/index.php?title=An%C3%A1lisis\\_DAFO&oldid=63884510](http://es.wikipedia.org/w/index.php?title=An%C3%A1lisis_DAFO&oldid=63884510). Consultado: Diciembre 2012.

[23] J. Isern, "Estudio experimental de métodos de Calibración y auto calibración de cámaras". Tesis Doctoral. Departamento de Informática y Sistemas, Universidad de Las Palmas de Gran Canaria, España, 2003.

[24] F. Carranza, "Esqueletización" [Online]. Tópicos Especiales en Procesamiento Gráfico, Escuela Académico Profesional de Informática, Universidad Nacional de Trujillo, Perú, 2006. Disponible en: <http://es.scribd.com/doc/49462631/esqueletizacion>. Consultado: Octubre 2012.

[25] I. Gómez, "Algoritmo para la construcción de grandes mallas mediante la triangulación de Delaunay". Trabajo fin de carrera. Facultad de Informática, Universidad Politécnica de Madrid, España, Marzo 2011.

[26] P. Goswami, F. Erol, R. Mukhi, R. Pajarola & E. Gobbetti, "An efficient multi-resolution framework for high quality interactive rendering of massive point clouds using multi-way kd-trees", *The Visual Computer*, vol. 29, No. 1, pp. 69-83, Springer-Verlag, 2012.

[27] M. Fernández, "Modelado, texturizado y ajuste de malla" [Online]. Madrid: Archivos Universidad Carlos III de Madrid, 2011. Disponible en: [http://e-archivo.uc3m.es/bitstream/10016/12936/1/modelado\\_fernandez\\_2011\\_pp.pdf](http://e-archivo.uc3m.es/bitstream/10016/12936/1/modelado_fernandez_2011_pp.pdf). Consultado: Septiembre 2012.

[28] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli and G. Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool". Eurographics Italian Chapter Conference, Pisa Italy, 2008.