

Reconstrucción 3D de objetos sumergidos en aguas limpias

3D reconstruction of objects submerged in clean water

Resumen

Este artículo presenta el desarrollo de un sistema hardware/software que permite reconstruir objetos sumergidos en agua y brinda la posibilidad de que los resultados puedan ser vistos desde diferentes perspectivas en un ambiente virtual. Se describe cómo a partir de la proyección de una línea láser sobre el objeto, es posible extraer las coordenadas espaciales (x, y, z) de cada punto proyectado gracias a la utilización de la técnica de triangulación. Dentro del documento se describen las etapas que llevaron a obtener el modelo tridimensional de una vista isométrica del objeto, las cuales inician con el escaneo del objeto, pasando por la captura, procesamiento y tratamiento de las imágenes (segmentación y esqueletización), y finalizando con el cálculo de las coordenadas espaciales, obtención de la nube de puntos y renderizado. Se presentan experimentos con varios objetos cotidianos, donde el sistema planteado obtiene resultados cualitativos positivos.

Palabras clave: dilatación, distancia focal, erosión, esqueletización, *frames*, nube de puntos, *perfil*, renderizado, umbralización, vista isométrica.

Abstract

This paper presents the development of a hardware / software system that allows reconstructing objects immersed in water and provides the possibility that the results can be seen from different perspectives in a virtual environment. Describes how from projecting a laser line on the object, it is possible to extract the spatial coordinates (x, y, z) of each projected point by using the triangulation technique. Within the document describes the steps taken to obtain the three-dimensional model of an isometric view of the object, which begin with scanning of the object, passing through the capture, processing and image processing, and ending with the calculation of the coordinates space, obtaining and rendering point cloud. Preliminary results with a variety of common objects are described.

Key words: dilatation, focal length, erosion, skeletonization, *frames*, point cloud, *profile*, rendering, thresholding, isometric view.

Leydy Muñoz

Universidad del Cauca
leydyviviana@hotmail.com

Edilson Quiñónez

Universidad del Cauca
equinonez@unicauca.edu.co

Héctor Victoria

Universidad del Cauca
hectordaniel@unicauca.edu.co



Fecha recibido: abr. 17/2013
Fecha modificado: nov. 8/2013
Fecha aceptado: nov. 29/2013



1. 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.

La principal motivación de este proyecto está orientada a que actualmente no se conoce soluciones de bajo costo que puedan ser utilizadas en grupos de investigación para inspecciones submarinas y estudios oceanográficos, lo que se convierte en una oportunidad para abrir este campo de investigación, mediante la búsqueda de alternativas de sistemas de reconstrucción 3D que puedan ser implementadas en el medio acuático y que reduzcan los costes de 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 [4].

De acuerdo con esto, se considera pertinente 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 [5], [6].

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 [7], [8], [9].

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.

2. Técnicas de reconstrucción 3D y triangulación láser

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.) y la Kinect de Microsoft.

Existen distintas técnicas de reconstrucción 3D que se pueden dividir en dos grandes grupos: técnicas con contacto con el objeto a reconstruir y técnicas sin contacto. Las primeras son las más 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, o 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.

En este proyecto se utilizó una técnica óptica, llamada triangulación láser que consta de una cámara y de un láser. Entre otras técnicas encontramos la visión estereoscópica, y las franjas [10].

Triangulación láser: 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 [11]. 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 1 se puede apreciar de manera visual el principio de funcionamiento de esta técnica. Donde el punto de la escena $P_w(X, Y, Z)$ se proyecta en la imagen como el punto $P_u(x, y)$.

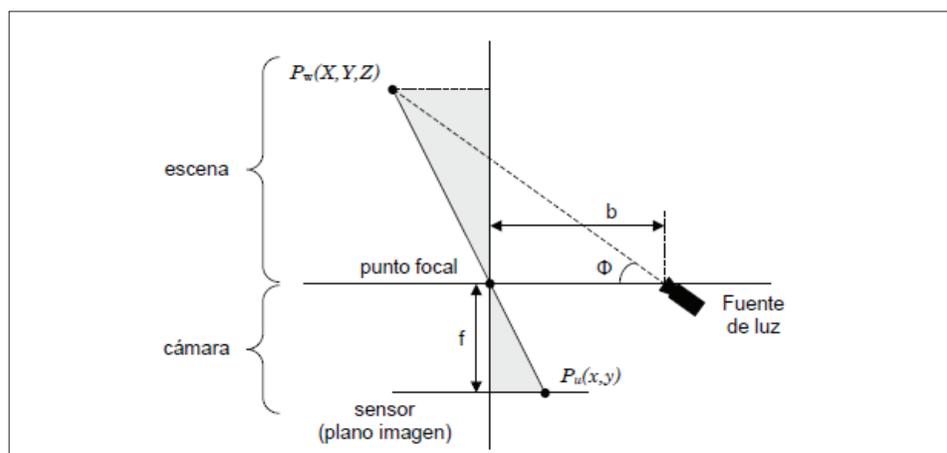


Figura 1. Relación de triángulos semejantes en la técnica de triangulación. Fuente: [10]



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 [7], [12].

De la imagen observamos que se cumple que:

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

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

Ahora bien, gracias a la relación descrita en las ecuaciones (1), (2) entre los triángulos semejantes, se pueden establecer las ecuaciones (3), (4), y (5) para calcular las coordenadas (X,Y,Z) del punto PW que pertenece al objeto de la escena. Los valores de X, Y, Z serían.

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

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

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

Calibración de la cámara: es un proceso que permite conocer cómo una cámara proyecta un objeto del mundo real su 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. 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. 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 [13].

3. Captura de imágenes

Para llevar a cabo los experimentos de reconstrucción 3D en un medio acuático, fue necesario realizar una planta e implementar tres módulos que permiten, en conjunto, obtener las imágenes de los perfiles del objeto proyectado ante desplazamientos contro-

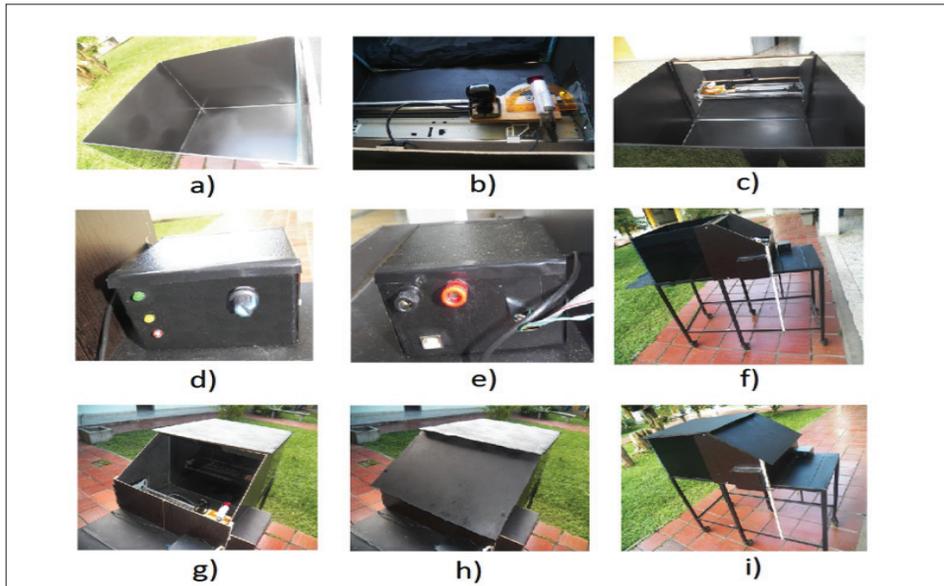


Figura 2. a) Cubo de vidrio, b) módulo de escaneo, c) vista superior del interior de la planta, d) Parte delantera de la caja del componente electrónico, e) Módulo de control, f) vista externa de la planta.

lados del láser. A continuación se describen los tres módulos, consistentes en el módulo de gestión de información, módulo de control y un módulo de escaneo.

Planta implementada: 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 corresponder a la del láser.

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 este hecho provoca movimiento del agua, produciendo así alteración de las imágenes tomadas y una errónea toma de datos. La Figura 2 muestra la planta construida.

Captura de imágenes (*hardware/software*): la captura de imágenes es la primera etapa de implementación del sistema hardware/software para realizar la reconstrucción. 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 escaneo (ver figura 3).

Antes de realizar la captura de imágenes es necesario realizar la calibración de la cámara para corregir las distorsiones que ésta presenta. Para esto se utilizó la librería de Visión Artificial OpenCV [13].



- a) **Módulo de gestión de información.** este módulo está compuesto por un ordenador y su función principal consiste en enviar la orden de escaneo al sistema cámara-láser (módulo de escaneo), especificando la distancia en centímetros que se debe desplazar, además de encender la cámara para realizar la captura de imágenes en el proceso de escaneo. Inicialmente el ordenador envía la instrucción al módulo de control mediante comunicación serial, y 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, y termina la captura de imágenes. El desarrollo del software implementado se realizó bajo el lenguaje de programación Python, utilizando las librerías de PyQt4 para la interfaz gráfica.
- b) **Módulo de control.** este módulo está compuesto por una tarjeta Arduino Uno consistente en una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. La función principal del módulo construido es la de gestionar las señales de mando necesarias para controlar el motor paso a paso que se encuentra en el módulo de escaneo. Inicialmente recibe por comunicación serial desde el módulo de gestión de información, la orden de la distancia en centímetros que debe recorrer el sistema cámara-láser y verifica que efectivamente se encuentre en la posición inicial de escaneo, ver Figura 3, en caso de no ser así, activa el motor paso a paso

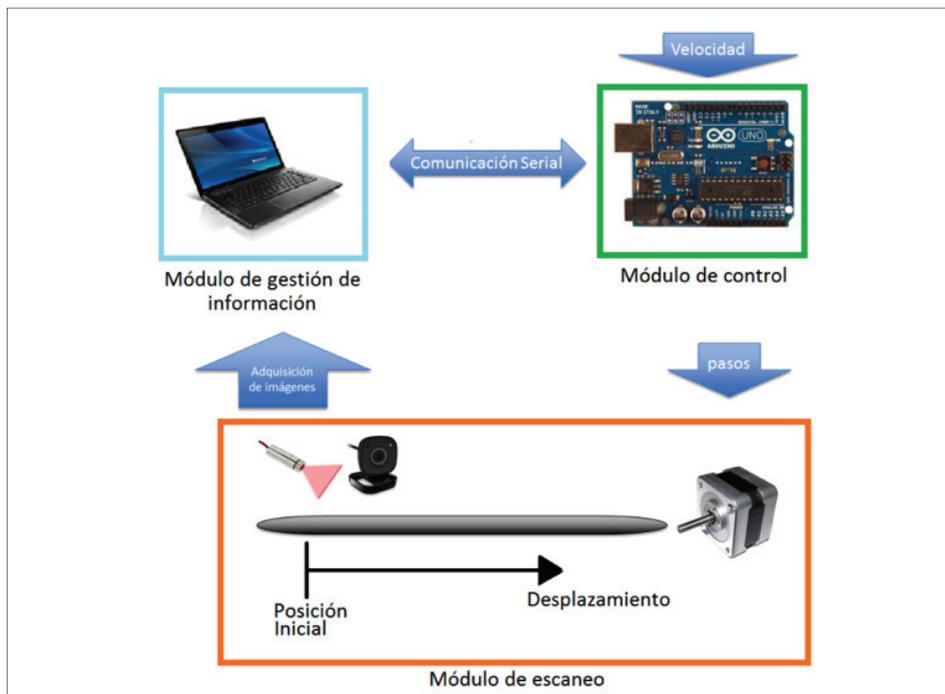


Figura 3. Proceso de captura de imágenes, conformado por un PC, un módulo de control y un sistema cámara láser que se desplaza en forma horizontal para hacer el barrido de la imagen

y lo localiza en la posición inicial. Una vez en la posición inicial, traduce las órdenes de distancia en cantidad de pasos que el motor debe realizar para ocasionar el desplazamiento del sistema cámara láser y ejecuta esta acción. Cumplida esta orden, envía un mensaje por comunicación serial al módulo de gestión de información con el fin de indicarle que el escaneo ha terminado e inmediatamente posiciona el sistema cámara y el láser en su posición de origen. En este módulo, la velocidad del motor puede ser graduada de forma manual por el usuario, mediante un potenciómetro no lineal.

- c) **Módulo de escaneo.** este módulo está conformado por el motor paso a paso y el sistema cámara láser ver Figura 1 (b). Su función principal es llevar a cabo el proceso de escaneado mediante el desplazamiento del sistema cámara láser por medio de la activación del motor. Inicialmente recibe la orden de mando del módulo de control para activar el motor paso a paso, el cual mueve el sistema cámara láser por medio de una correa dentada, al tiempo que el módulo de gestión de información va guardando en el ordenador las imágenes obtenidas. La activación y el ángulo del láser se realiza de forma 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.

Este proceso de captura de imágenes termina con el almacenamiento de las imágenes en el computador, una vez realizado, se pasa a una segunda etapa consistente en el procesamiento de las imágenes para obtener únicamente el perfil proyectado por el láser.

4. Procesamiento de imágenes

En este proceso se abstraen las líneas de luz que proyecta el láser sobre el objeto (de ahora en adelante *perfiles*) y que han sido capturadas en cada instante de tiempo definido por el usuario, 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 4 y se detallan a continuación:

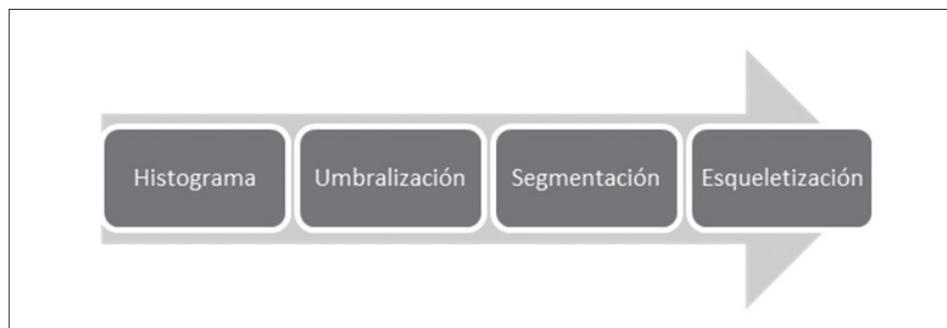


Figura 4. Fases del proceso de abstracción de información de las imágenes.

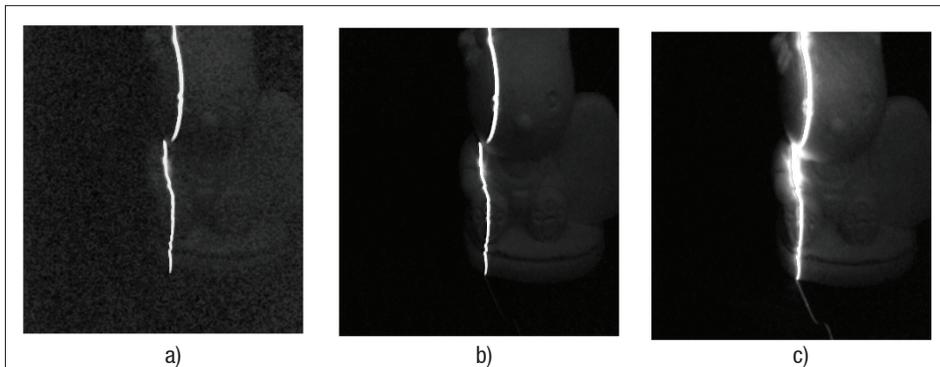


Figura 5. Canales de la imagen de un perfil, a) Canal rojo, b) Canal verde, c) Canal azul

Histograma: el histograma es una representación global de los tonos de los píxeles de una imagen. Al aplicarle el algoritmo de histograma a una imagen lo que se obtiene es en realidad una gráfica de barras de tonos contra número de ocurrencia de estos. Para este proyecto es importante realizar este análisis debido a que es necesario saber entre cuáles tonos se encuentra la mayor cantidad de información, es decir cuales píxeles se repiten más, es evidente que por tratarse de imágenes tomadas en un ambiente sin iluminación, las luces del láser claras pueden corresponder a reflejos o ruidos y los píxeles más intensos corresponden a la línea del láser proyectada sobre el objeto, que es de nuestro interés.

Las imágenes están compuesta por tres canales: rojo, verde y azul, conocidos por sus siglas como RGB, es importante escoger el canal de la imagen que almacene la mayor información relevante y que a su vez esta información tenga la menor cantidad de ruido. Para seleccionar el canal se tomó un *perfil* y se dividió la imagen en los tres canales (ver figura 4). En esta figura se puede observar que el *perfil* del canal rojo figura 5(a) está bien definido y no presenta grandes iluminaciones en sus bordes, el *perfil* del canal verde figura 5(b) también está bien definido pero presenta iluminaciones en sus bordes, el *perfil* del canal azul figura 5(c) 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.

Al aplicar la función de OpenCV del histograma nos entrega el rango de tonalidades de píxeles y la ocurrencia de los mismos, es de nuestro interés los píxeles con mayor nivel de intensidad en el rojo porque son las zonas donde la línea del láser proyecta la imagen, otras zonas con niveles menores de rojo podrían corresponder al reflejo de la línea láser sobre el objeto, obtener esta información nos permitirá en la etapa posterior filtrar la imagen por encima y por debajo de estos valores.

Umbralización: una vez se obtuvo en el histograma que la mayor cantidad de píxeles se encuentran en el rango de 180 a 255 se determina esta zona como la información de interés y se umbralizaron todas las imágenes con el fin de volver blancos los píxeles en el rango [180 -255] y negros los píxeles entre (180 – 0]. En la figura 6 se muestra el resultado de aplicar el umbral a imágenes tomadas de un gato de porcelana sumergido en agua, aplicando la función de umbralización de OpenCV.

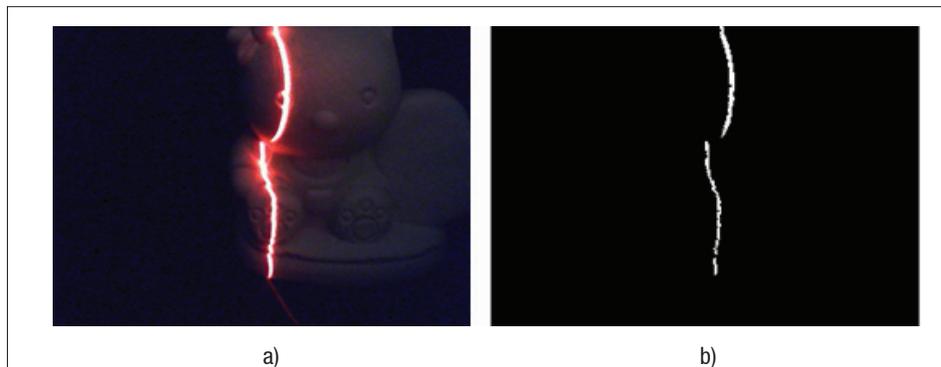


Figura 6. a) Imagen de entrada, b) imagen umbralizada

Segmentación: el objetivo de la segmentación es separar los objetos del fondo (binarizar), así, la segmentación de una imagen permite destacar zonas con características específicas de color o de forma. Para realizar este proceso se construyó una función que delimita la zona donde se encuentran píxeles consecutivos convirtiéndola en la región de interés, dejando fuera de esta región aquellos píxeles o conjunto de píxeles que se encuentran aislados del perfil de la imagen. En la figura 7(a) observe que un punto a la derecha de la línea (perfil de la imagen), este conjunto de píxeles debe ser eliminado, ya que de no hacerlo ocasionaría que estos píxeles formen parte de la imagen, y al aplicar la función de renderización al final de proceso, que lo que hace es construir líneas entre los píxeles para formar superficies, los resultados no serían los esperados, causando que el modelo obtenido se deforme y no se pueda obtener una representación adecuada del objeto real. [14].

Esqueletización: con este proceso se 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 [14] [15]. Estos están basados en reglas para la eliminación de píxeles blancos de manera iterativa hasta que ya no haya más cambios por realizar. Estos algoritmos de esqueletización varían en cuanto al tiempo de procesamiento de las imágenes y en cuanto a resultados satisfactorios, sin embargo tardan tiempos inaceptablemente altos con grandes cantidades de imágenes.

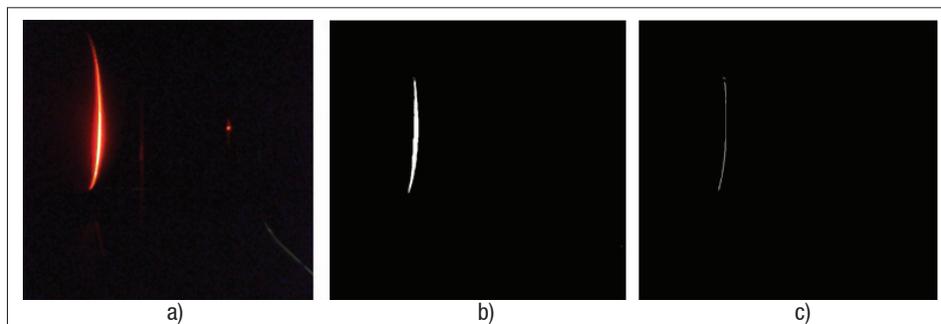


Figura 7. a) Perfil de una imagen sin segmentar. b) Imagen segmentada

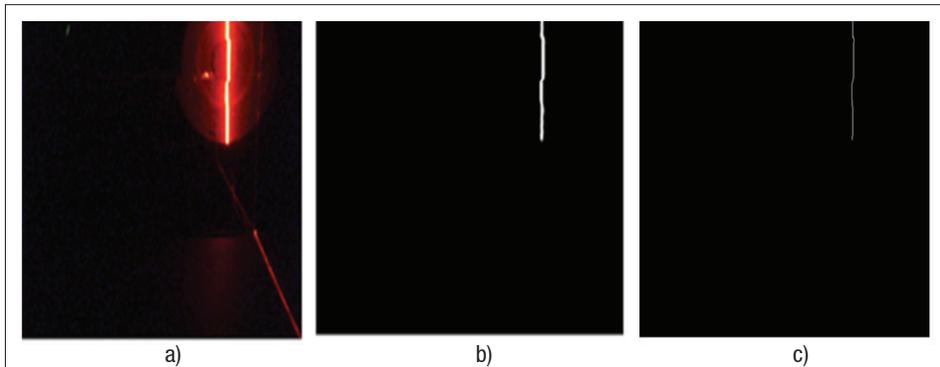


Figura 8. a) Imagen de entrada, b) imagen segmentada, c) imagen esqueletizada

Teniendo en cuenta esta premisa, se eligió el algoritmo matemático de Zhang – Suen [16] y se implementó en un software, con un mejor rendimiento que los otros algoritmos [14] [15]. Este algoritmo representa uno de los mayores aportes en este proyecto, debido a que fue optimizado para reducir el tiempo en que esqueletizan las imágenes, mediante la utilización de procesos paralelos, debido a que generalmente se tienen cientos de imágenes tomadas de un mismo objeto.

En la figura 8 se evidencia el resultado obtenido al aplicar el algoritmo de esqueletización construido sobre un perfil de la imagen. La figura 8(a) muestra la imagen capturada sin ningún procesamiento, la figura 8(b) una imagen que ha pasado por la fase de histograma, umbralización y segmentación y la figura 8(c) se ilustra el resultado de aplicar el algoritmo de esqueletización desarrollado, obsérvese obteniéndose una imagen de un pixel de ancho que conserva la forma del perfil.

5. Obtención de la nube de puntos y renderizado

En esta etapa de desarrollo del proyecto se calculan las coordenadas tridimensionales de cada uno de los píxeles que conforman el perfil esqueletizado, por esto es de suma importancia garantizar que se tenga el mínimo posible de píxeles, ya que de no ser así deberían calcularse las coordenadas de los miles de píxeles que conforman la imagen original y los tiempos de procesamiento serían inaceptables.

Sin embargo surge la duda de cómo obtener tres coordenadas (x, y, z) de cada uno de los píxeles de la imagen esqueletizada, a partir de una imagen de dos dimensiones, es decir solo cuenta con las 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*.

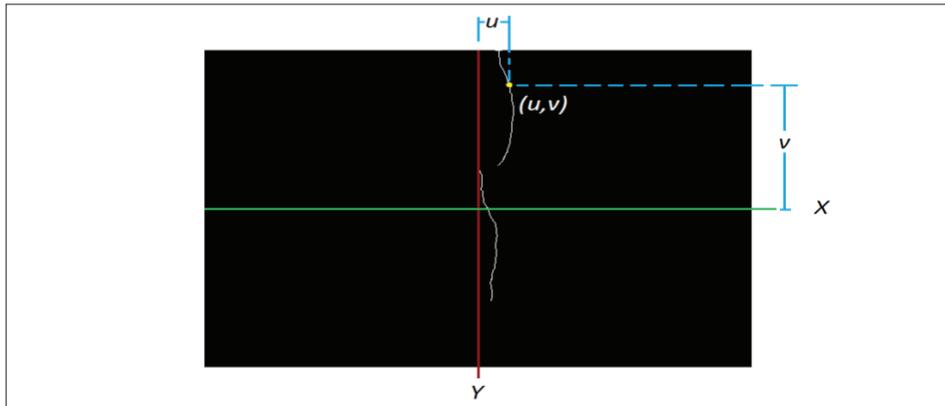


Figura 9. Ilustración del cálculo de las coordenadas de los puntos pertenecientes al *perfil*

Cálculo de coordenadas espaciales (u, v) : para el cálculo de las coordenadas espaciales se construyó un algoritmo que permite obtener las coordenadas de cada pixel de la imagen esquelizada a partir de un perfil. El funcionamiento del algoritmo es el siguiente, el algoritmo ubica el primer pixel blanco contenido en la imagen partiendo de la parte superior de la imagen y desplazándose de izquierda a derecha, una vez encontrado, calcula la posición de ese pixel tanto en el eje horizontal (coordenada u) como en el eje vertical (coordenada v) respecto al centro de la imagen (cruce entre la línea verde y la línea roja de la figura 9). El algoritmo es capaz de detectar el tamaño de la imagen y calcular su centro, por ejemplo en el caso de una imagen de tamaño 640 x 480, el centro de la imagen tendría el valor de 320 por 240 ($u_0 = 320, v_0 = 240$). El proceso se repite tantas veces como pixeles blancos contenga la imagen, por esto la importancia del algoritmo construido de esquelizado, que busca perfil tenga el menor número de pixeles sin que se pierda su forma y así se evita realizar una mayor cantidad de operaciones.

Una vez finalizado el proceso se forma un vector con las coordenadas (u, v) y se inicia el proceso de calcular la coordenada z para cada punto, para lograrlo se aplican las ecuaciones de la técnica triangulación láser explicada en el apartado II y se realiza la transformación de coordenadas (2D a 3D) de cada 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.

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, dentro de este proyecto solo se tendrá en cuenta la geometría.



El proceso de renderizado implementado al modelo obtenido consiste en ubicar todos los puntos obtenidos en la fase anterior en el espacio y dibujar polígonos entre los puntos o coordenadas espaciales que describen al objeto original. Para llevar a cabo este proceso existen muchas técnicas, para este proyecto se trabajó con Triangulación de Delaunay [17], [18], y el VCG [19]. 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 [20] 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.

6. Resultados experimentales

En la figura 10 se pueden observar los objetos a ser reconstruidos utilizando la técnica planteada en este trabajo, los cuales son: un gato de porcelana, una esfera de *poliestireno expandido* conocido comúnmente como icopor en Colombia, una rana fabricada en loza y una tapa de plástico.

Inicialmente se importa desde Meshlab la nube de puntos del objeto. Éstos fueron obtenidos anteriormente como se explicó en el ítem cálculo de coordenadas espaciales y fueron guardados por el algoritmo en un archivo con formato .PLY para permitir compatibilidad con la herramienta Meshlab.

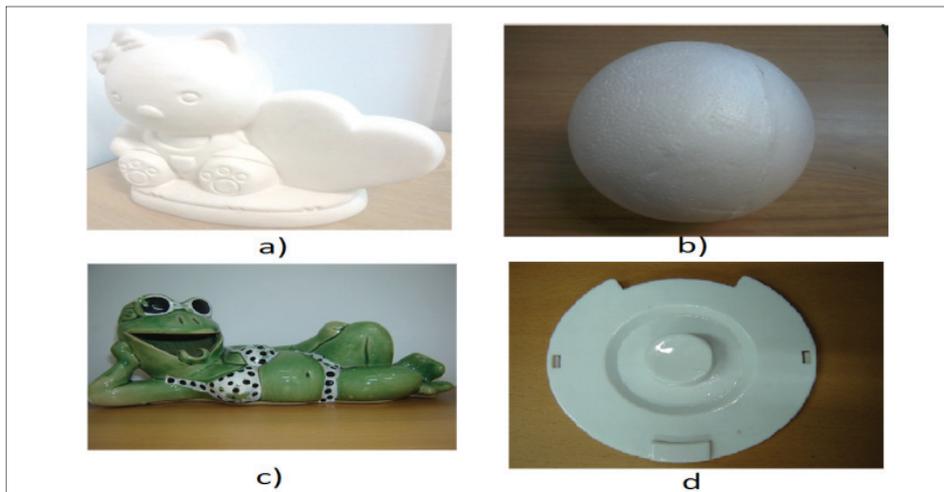


Figura 10. Objetos a ser reconstruidos mediante la técnica planteada en este trabajo, a) gato, b) esfera, c) rana, d) tapa

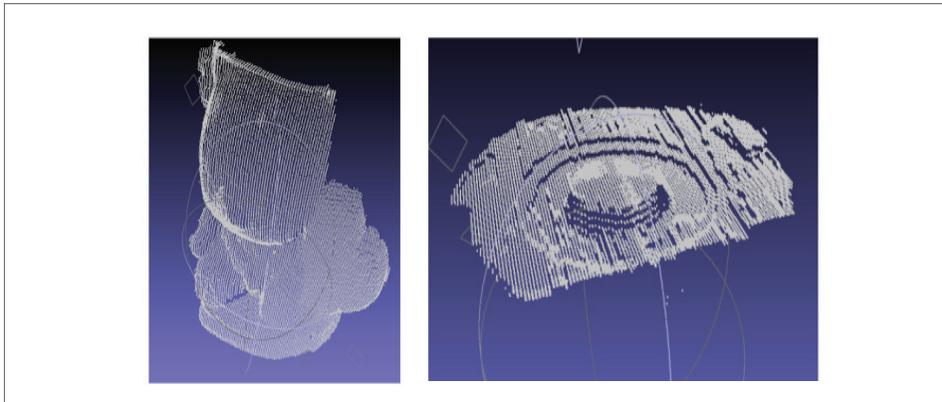


Figura 11. Vista de las nubes de puntos desde Meshlab para el objeto gato

En la figura 11 se muestra la importación de las nubes de puntos del objeto gato, en este caso no se ha realizado ningún proceso sobre los datos, simplemente han sido importados, como se observa en la imagen los puntos tienen diferente profundidad debido a la coordenada z que se recuperó en cada pixel.

Una vez importados los puntos, se hace necesario obtener figuras sólidas que sean una representación en tres dimensiones del objeto original, para ello se utiliza el algoritmo VCG de Meshlab con el fin de rellenar con nuevos puntos aquellos espacios que existen entre los perfiles importados. En la figura 12 se puede observar el resultado de aplicar el algoritmo VCG. Obsérvese el aumento en la densidad de los puntos que conforman la imagen.

Una vez realizado lo anterior, el modelo se encuentra listo para ser renderizado, proceso que consiste en dibujar líneas entre los puntos del modelo (malla) para generar una superficie. En la figura 13 se observa el resultado del modelo 3D obtenido para el objeto

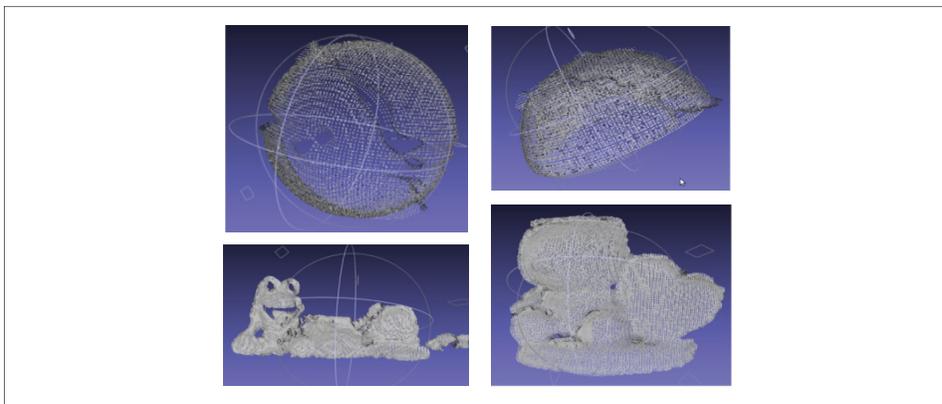


Figura 12. Resultado de aplicar el algoritmo VCG para el objeto gato



tapa, donde se puede detallar que conserva la forma del objeto original (ver figura 10). 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.

En la figura 14 se muestra el resultado del renderizado para la nube de puntos del objeto rana (ver figura 10 para observar el objeto original). Aquí se observa que existen regiones del objeto donde se han perdido los detalles, que corresponde a información donde el brillo u otros factores evitaron la proyección del láser para ese perfil.

Los resultados de renderizado para el objeto gato se puede visualizar en la figura 15, donde se observa que el modelo 3D obtenido conserva la forma del objeto original (ver figura 10), no obstante se han perdido detalles finos como los ojos y las huellas de los pies.

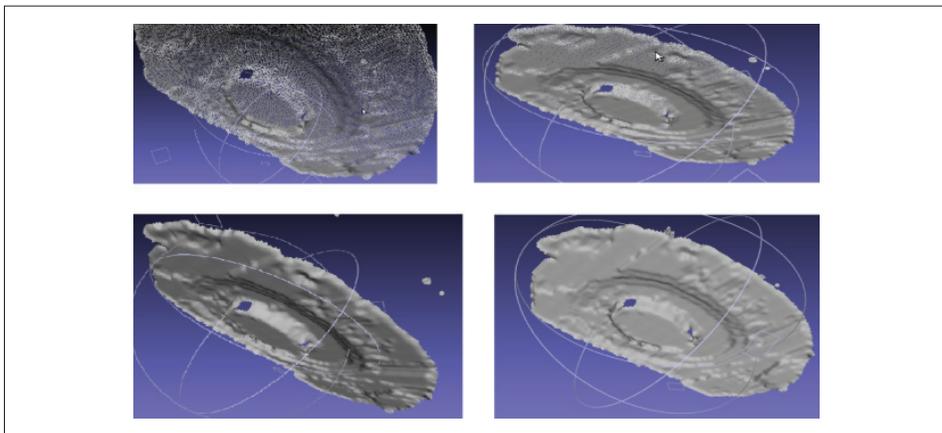


Figura 13. Resultado de aplicar el renderizado al objeto tapa

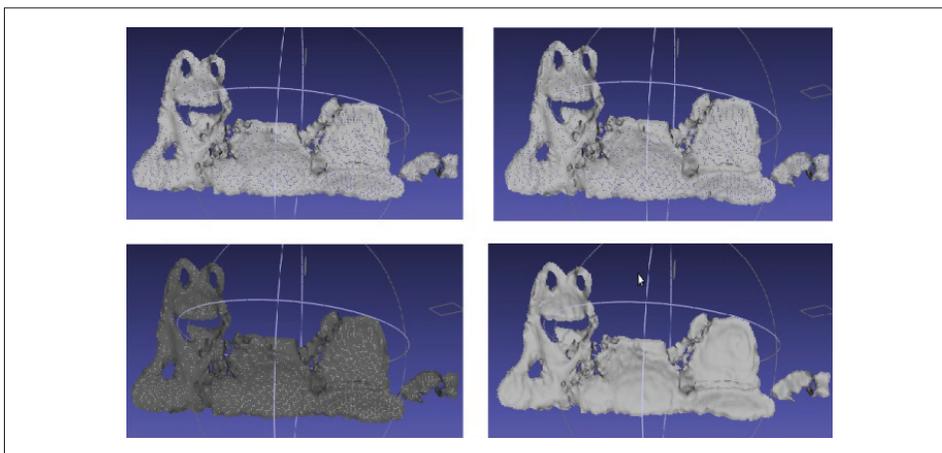


Figura 14. Resultado de aplicar el renderizado al objeto rana

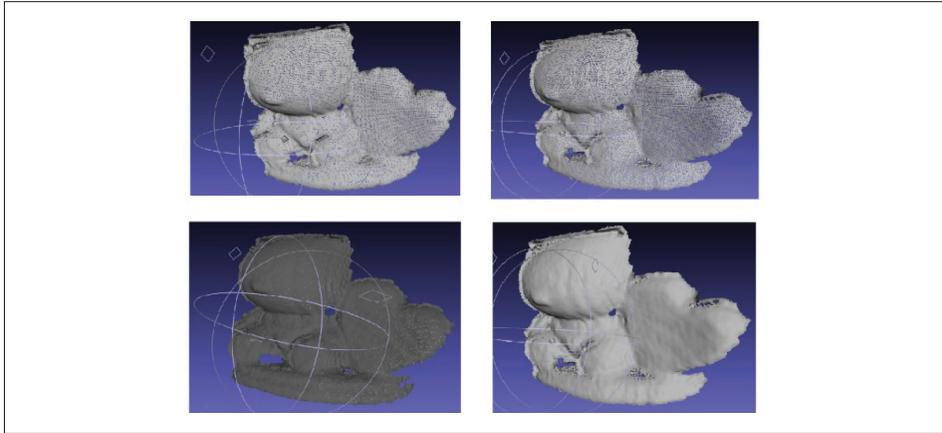


Figura 15. Resultado de aplicar el renderizado al objeto gato

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*. 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 16.

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 10), muestran que a nivel general se consiguen buenos resultados (ver figuras 13, 14, 15, 16). 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

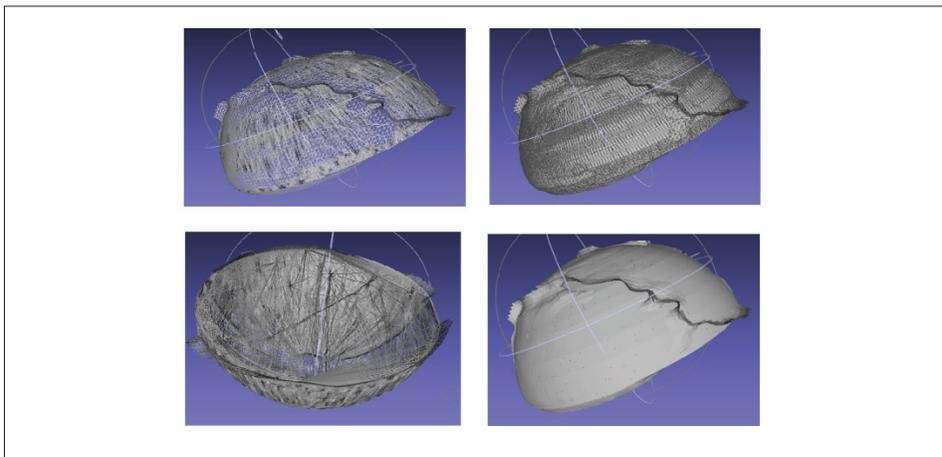


Figura 16. Resultado de aplicar el renderizado al objeto esfera



debido a varias causas. Los mejores modelos 3D encontrados son los del objeto esfera y el objeto gato. El primero se debe a que este 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 se necesita capturar los perfiles durante el proceso de escaneado causan brillos y dispersiones de luz, haciendo que la información de las coordenadas espaciales que se obtienen, contenga 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.

7. Conclusiones

Numerosos trabajos [21] [22] [23] realizan la reconstrucción a partir del principio de la triangulación utilizando diferentes instrumentos como láser, espejos octogonales y cámaras. La técnica implementada en este proyecto utiliza un láser y una cámara para realizar la reconstrucción, el uso de espejos no se consideró debido a que los objetos se encuentran sumergidos en agua y las imágenes aparecerían desplazadas al ser proyectadas en un espejo. No se encontró en la literatura reconstrucciones 3D de objetos sumergidos en agua, pero se observó que la única variante es que el objeto aparece desplazado al ser visto desde afuera del agua por la cámara, sin embargo no modifica su forma o tamaño. Así, cualquier técnica en la que el objeto no realice algún movimiento para la captura de los perfiles podría ser implementada, 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.

El buen funcionamiento de la técnica de reconstrucción 3D utilizada en este trabajo depende en gran medida de la iluminación de la escena donde se encuentra el objeto, es necesario tener escenas con poca iluminación, los objetos a reconstruir no deben ser totalmente blancos o brillantes, porque causan que el haz de luz láser reflejado en el objeto deslumbré la vista del mismo y se pierda el perfil formado por el haz de luz emitido, el objeto debe ser pequeño debido a que se trabaja a distancias cortas y el rango de visión de la cámara es limitado, la cámara sea calibrada previamente, y el ángulo de inclinación del láser respecto a la cámara no debe tener valores cercanos a los 90° 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.

En este proyecto se utilizó la librería Open CV para el procesamiento de imágenes, debido a que tiempo de procesamiento de los perfiles al realizar las operaciones de histograma, umbralización, segmentación y esqueletización es menor que el obtenido con Matlab; si bien existen implementaciones utilizando Matlab que resultan más sencillas, los tiempos de procesamiento aumentan [21] [22]. También existen otros trabajos utilizando la librería OpenCV (ver por ejemplo [23]).

Un aporte importante de este proyecto se encuentra en el algoritmo de esqueletización construido, el cual es capaz de analizar los perfiles y extraer todos aquellos píxeles blancos no relevantes para la forma del perfil, evitando obtener datos repetidos que no aportan información relevante para la reconstrucción y que hacen más lento el cálculo de las coordenadas espaciales. Esto permitió además obtener buenos resultados en la reconstrucción de los objetos, conservando la forma.

Referencias Bibliográficas

- [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] J. Contreras, Plataforma sensorica a bordo de ROV para inspección portuaria, bases militares y oceanografía, Nombre comercial: Plataforma sensorica a bordo de ROV para inspección portuaria, bases militares y oceanografía, contrato/registro: RC 604-.2008, En: Colombia, 2009
- [5] 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.
- [6] G. Telem, S. Filin. “Photogrammetric modeling of underwater environments”, International Journal of Photogrammetry and Remote Sensing, vol.65, pp. 433-444, Israel, 2010.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] J. Molleda, “Técnicas de Visión por Computador para la Reconstrucción en Tiempo Real de la Forma 3D de Productos Lamina-dos”. Tesis Doctoral. Departamento de Informática, Universidad de Oviedo, Gijón España 2008.
- [11] 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: octubre 2012.
- [12] 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.
- [13] 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.



- [14] Gonzalez, R. C., & Wintz, P. (1996). Procesamiento digital de imágenes. Addison-Wesley, Tema, 2, 23-56.
- [15] IMAGine: Cursos Interactivos de Tratamiento Digital de Imagen. “Operaciones Morfológicas” [Online]. Disponible en: http://www.tsc.uc3m.es/imagine/Curso_ProcesadoMorfologico/Contenido/Operaciones/OperacionesMorfologicas.html. Consultado: julio 2012.
- [16] F. Carranza, “Esqueletización” [Online]. Tópicos Especiales en Procesamiento Gráfico, Escuela Académico Profesional de Infor-mática, Universidad Nacional de Trujillo, Perú, 2006. Disponible en: <http://es.scribd.com/doc/49462631/esqueletizacion>. Consultado: octubre 2012.
- [17] 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.
- [18] 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.
- [19] 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. Consultado: septiem-bre 2012.
- [20] 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.
- [21] Pérez, E., Salamanca, S., Gallego, D., Merchán, P., Adán, A., & Cerrada. SHADES-3D: Una Herramienta para la Adquisición de Da-tos 3D Bajo el Entorno Matlab©. *RIAI*, 3(1), 94-103, 2010.
- [22] E. Hernández, S. Mino, D. Martinez, P. Garcia, Q. Oliver, & C. Somolinos, SHADES-3D: Una Herramienta Para La Adquisicion De Datos 3d Bajo El Entorno Matlab. *Revista iberoamericana de automática e informática industrial (RIAI)*, 3(1), 94-103, 2006
- [23] L. Revelo, F. Usama, & J. Flórez. 3D Reconstruction of Scenes by Means of a Stereoscopic Vision System, Based on Feature Ex-traction and Developed in OpenCV. *Ingeniería y Universidad*, 16(2), 485-500, 2012.

Leydy Viviana Muñoz Robles

Obtuvo el título de Ingeniera en Automática Industrial de la Universidad del Cauca, Popayán, Colombia. En la actualidad es estudiante de la Maestría en Automática de la misma universidad. e-mail: leydyviviana@hotmail.com

Edilson Quiñónez Jiménez

Es Ingeniero en Automática Industrial de la Universidad del Cauca, Popayán, Colombia. e-mail: equinonez@unicauca.edu.co

Héctor Daniel Victoria Pizo

Es Ingeniero en Automática Industrial de la Universidad del Cauca, Popayán, Colombia. Actualmente se desempeña como ingeniero de desarrollo de LEGIS. e-mail: hectorandaniel@unicauca.edu.co