

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA



MÉTODO DE DISMINUCIÓN DE LOS EFECTOS DE
COMPORTAMIENTO NO LINEAL DE UN SISTEMA DE VISIÓN
TÉCNICA PARA LA OBTENCIÓN DE MODELOS
TRIDIMENSIONALES

TESIS

PARA OBTENER EL TÍTULO DE

DOCTOR EN CIENCIAS

PRESENTA

IVAN YENISEYSK ALBA CORPUS

DIRECTORA DE TESIS: DRA. WENDY FLORES FUENTES

CODIRECTOR DE TESIS: DR. OLEG SERGIYENKO

MEXICALI, B.C.

AGOSTO 2025

Resumen

El escaneo tridimensional ha revolucionado la captura y digitalización de objetos físicos en diversas áreas, desde la ingeniería inversa hasta la manufactura aditiva. Sin embargo, los sistemas de visión técnica (TVS) de Universidad Autónoma de Baja California (UABC) presentan limitaciones en la precisión y captura de datos de los modelos tridimensionales que se generan a través de ellos debido a múltiples factores derivados del diseño y características del mismo. En este trabajo, se investiga el impacto de estos efectos en la calidad de las reconstrucciones obtenidas mediante el prototipo 2 del TVS, analizando cómo la posición del objeto dentro del campo de visión influye en la precisión de la nube de puntos generada. Se realizaron experimentos en distintas ubicaciones del campo visual, identificando discrepancias en la fidelidad de los modelos.

Para mitigar estas variaciones, se implementaron técnicas de filtrado de ruido, suavizado de nubes de puntos y análisis de entropía. Además, se aplicaron algoritmos de reducción de puntos atípicos y ajuste de coordenadas para mejorar la coherencia de las reconstrucciones. Los resultados experimentales demostraron una mejora significativa en la calidad de los modelos tridimensionales tras la implementación del método propuesto. Este estudio contribuye al desarrollo de estrategias de corrección en sistemas de escaneo 3D, optimizando su uso en aplicaciones industriales y científicas. Además, se investigaron otros posibles usos del sistema en reconocimiento de colores mediante reconocimiento del color de las coordenadas a través de modelos de clasificación de inteligencia artificial.

Palabras clave: visión técnica, escaneo 3D, efectos no lineales, nubes de puntos, filtrado de ruido, reconstrucción tridimensional.

Abstract

Three-dimensional scanning has revolutionized the capture and digitization of physical objects in various fields, from reverse engineering to additive manufacturing. However, the technical vision systems (TVS) at Universidad Autónoma de Baja California (UABC) exhibit limitations in the accuracy and data capture of the 3D models generated due to multiple factors arising from their design and inherent characteristics. This study investigates the impact of these effects on the quality of reconstructions obtained through Prototype 2 of the TVS, analyzing how the object's position within the field of view influences the accuracy of the generated point cloud. Experiments were conducted at different locations within the visual field, identifying discrepancies in model fidelity.

To mitigate these variations, noise filtering techniques, point cloud smoothing, and entropy analysis were implemented. Additionally, outlier reduction algorithms and coordinate adjustment methods were applied to improve the coherence of the reconstructions. Experimental results demonstrated a significant improvement in the quality of the 3D models after implementing the proposed method. This study contributes to the development of correction strategies for 3D scanning systems, optimizing their use in industrial and scientific applications. Furthermore, other potential uses of the system were investigated, such as color recognition of color of the coordinates through artificial intelligence classification models.

Keywords: technical vision, 3D scanning, nonlinear effects, point clouds, noise filtering, 3D reconstruction.

Dedicatoria

A mi familia y a mis amigos.

Agradecimientos

A Dra. Wendy Flores, a Dr. Oleg Sergiyenko y Dr. Lars Lindner.
Gracias por sus años de paciencia y apoyo.

Índice general

1. Introducción	11
1.1. Planteamiento del problema	16
1.2. Justificación	19
1.3. Objetivos	20
1.3.1. Objetivo general	20
1.3.2. Objetivos específicos	20
1.4. Hipótesis	21
1.5. Antecedentes	21
2. Marco teórico	24
2.1. Investigación del estado del arte	24
2.2. Sistemas de visión mediante cámaras	25
2.3. Sistemas de visión mediante láser	26
2.3.1. Interferometría y escaneo láser de luz estructurada	26
2.3.1.1. Triangulación láser	26
2.3.1.2. Detección y Alcance de la Luz	27
2.3.2. Visión por Computadora Fotogramétrica	28
2.3.2.1. Procesamiento de Imágenes con Monovisión	30
2.3.3. Capacidades de sistemas 2D y 3D	32
2.3.4. Triangulación dinámica	33
2.3.4.1. Time of flight	35
2.3.4.2. Medición de fase	36
2.4. Algoritmos de visualización 3D	37

2.5.	TVS1	37
2.6.	TVS2	38
2.6.1.	Construcción mecánica del TVS2	39
2.6.2.	Control y funcionamiento del TVS2	41
2.7.	TVS3	42
2.7.1.	Ruido y puntos atípicos	43
2.7.2.	Procesamiento de datos y eliminación de puntos atípicos	44
2.7.2.1.	Método de intercuartiles	44
2.7.2.2.	Método Tau-Thompson modificado	45
2.7.2.3.	Cuantiles de distribución Chi^2	45
2.7.2.4.	Suavizado de la nube de puntos	45
2.7.3.	Registro de nube de puntos	46
2.7.3.1.	Registro de Punto de Correspondencia Iterativa (ICP)	46
2.7.3.2.	Registro de Consenso de Muestra Aleatoria (RANSAC)	47
2.7.4.	Verdad Terreno	47
2.7.5.	Reconstrucción	47
2.8.	Inteligencia Artificial y Redes Neuronales	48
2.8.1.	Reconocimiento de nubes 3D	48
2.8.2.	Segmentación	49
2.8.3.	Clasificadores	49
3.	Metodología	51
3.1.	Diseño del experimento para la obtención de modelos tridimensionales	51
3.2.	Diseño de experimento para reconocimiento de colores	56
3.2.1.	Extracción de características	56
3.2.2.	Clasificación	65
4.	Experimentación y resultados	68
4.1.	Resultados del experimento para la obtención de modelos tridimensionales	69
4.1.1.	Evaluación de similitudes	71
4.2.	Resultados del experimento para el reconocimiento de colores	78

5. Conclusiones	88
5.1. Experimento para la obtención de modelos tridimensionales	88
5.1.1. Comparación con datos sin filtrar	88
5.1.2. Limitaciones e incertidumbres	89
5.1.3. Experimento para el reconocimiento de colores	89
5.2. Discusión y conclusiones	90
Bibliografía	94
Apéndice A: Publicaciones	102
Apéndice B: Códigos Computacionales	114

Índice de figuras

1.1. Escaneo de objetos	17
1.2. Creación de nubes de puntos	17
2.1. Configuración de triangulación de láser	27
2.2. Configuración de LIDAR	28
2.3. Triangulación dinámica	34
2.4. Medición de fase	36
2.5. Medición de fase	37
2.6. Campo de visión discreto de TVS2	39
2.9. Ejemplo de movimiento de motor a pasos	42
2.10. Campo de visión continuo de TVS3	43
3.2. Configuración de medición.	57
3.3. Generación de la señal optoelectrónica durante el escaneo del punto láser.	58
3.4. Configuración de medición.	58
3.5. Muestreo de las señales optoelectrónicas y de referencia.	60
3.6. Delimitación de la señal Gaussiana (360°) y ventana de la señal Gaussiana (180°).	60
3.7. Diagrama de árbol de configuración del experimento.	66
4.2. Distribución de frecuencia de la cantidad de mediciones exitosas con el ángulo del LP a diferentes profundidades; a) Todas las medidas; b) Medidas a 70 cm; c) Medidas a 80 cm; d) Medidas a 90 cm.	80

4.3.	Distribución de frecuencia de la cantidad de mediciones exitosas con el ángulo del LP en diferentes tipos de papel a) Rojo brillante; b) Rojo mate; c) Azul brillante; d) Azul mate (Sin resultados); e) Amarillo brillante; f) Amarillo mate.	81
4.4.	Comparación de las señales optoelectrónicas en la misma coordenada 3D espacial.	82
4.5.	Error de medición de ángulo de posicionamiento láser. a) Por distancia de profundidad; b) Por tipo de papel.	84
4.6.	Matriz de confusión de validación del modelo de Ensemble de Subespacio KNN	87
1.	3D Data Processing and Entropy Reduction for Reconstruction from Low-Resolution Spatial Coordinate Clouds in a Technical Vision System	103
2.	Surface color estimation in 3D spatial coordinate remote sensing by a technical vision system	104
3.	3D spatial measurement for model reconstruction: A review	105
4.	A structural health monitoring method proposal based on optical scanning and computational models	106
5.	Detection of Atypical Data in Point Cloud of Technical Vision System using Digital Filtering.	107
6.	Obstacle Coordinates Transformation from TVS Body-Frame to AGV Navigation-Frame	108
7.	Evaluation of Photosensors for Laser Scanning Range Extension	109
8.	Point Cloud Optimization Employing Multisensory Vision	110
9.	Conjunto de datos de señales optoelectronicas.	111
10.	Extracción de características en el dominio del tiempo de señales optoelectronicas de un sistema de visión técnica.	112
11.	Herramienta de escaneo y visualización de datos 3D	113

Índice de tablas

3.1. Factores de experimentación	52
3.2. Configuración del experimento para los objetos de poliestireno escaneados	53
3.3. standard deviation	54
3.4. Entropía aproximada calculada para cada modelo tridimensional.	55
3.5. Entropía de Shannon calculada para cada modelo tridimensional.	55
3.6. Parámetros Extraídos en el Dominio del Tiempo	61
4.1. Entropía aproximada calculada para cada modelo tridimensional.	73
4.2. Entropía de Shannon calculada para cada modelo tridimensional.	73
4.3. Comparación de medidas de objetos escaneados en ambas posiciones desde la referencia de coordenadas del TVS.	74
4.4. Puntuación de ajuste para las caras de la letra “A” y el número “6” en la posición POS1.	74
4.5. Puntuaciones de las caras, normalizadas desde un máximo de 50.6% y mínimo de 0%.	75
4.6.	76
4.7.	77
4.8. Puntuaciones de ajuste para volúmenes. Hacer clic en cada imagen para visualizarla mejor en otra ventana.	77
4.9. Relación de mediciones de coordenadas obtenidas.	79
4.10. Evaluación de Exactitud y Precisión de Mediciones	83
4.11. Porcentaje de exactitud de validación de los clasificadores de modelos	86

Introducción

El progreso de la tecnología de escaneo 3D ha cambiado por siempre la forma en que capturamos y replicamos el mundo físico en forma digital. Desde la documentación de artefactos históricos [1] [2], hasta la precisión requerida en el diseño industrial, las aplicaciones de esta tecnología son vastas y variadas. De estos, los sistemas de escaneo láser son utilizados en una amplia variedad de campos, entre ellos específicamente como tecnología de detección remota. Estos sistemas son fundamentales en varios sectores, incluidos la atención médica, el levantamiento geoespacial, la fabricación aditiva, la industria minera y la ingeniería inversa, por mencionar solo algunos.

De lo anterior, la aplicación del escaneo láser en el campo médico, por ejemplo, permite un mapeo y modelado precisos de formas físicas, mientras que en el campo geoespacial, facilita la evaluación detallada de topografías terrestres y estructuras arquitectónicas. En el campo de la fabricación aditiva, el escaneo láser contribuye a la precisión y fidelidad de los objetos impresos en 3D, y en la minería, ayuda en el análisis volumétrico y la planificación de los sitios de extracción. Los procesos de ingeniería inversa se benefician significativamente del escaneo láser al proporcionar un medio para recrear modelos de objetos físicos existentes con alta precisión.

Múltiples estudios ilustran la influencia de las tecnologías de escaneo láser tanto en la investigación como en las aplicaciones prácticas. Por ejemplo, este artículo [3] explora la medición de estructuras de dosel vegetal utilizando escaneo láser, demostrando la capacidad del método para capturar datos detallados del dosel in situ sin contacto directo. Este estudio [4] introduce índices de vegetación innovadores derivados del escaneo láser terrestre, que cuantifican de manera efectiva la organización espacial 3D de las

comunidades vegetales. Otro ejemplo es este artículo [5], que presenta un análisis de un sistema de escaneo láser 3-D de rastreo, centrándose en su inmunidad a los efectos de la luz solar ambiental y su resolución geométrica. Para una exploración exhaustiva del desarrollo histórico y las aplicaciones diversas de los escáneres láser 3D, este trabajo [6] ofrece una visión general extensa, trazando la evolución de la tecnología y su impacto en diferentes industrias.

La evolución de la tecnología de escaneo 3D ha llevado al desarrollo de dispositivos cada vez más sofisticados que son capaces de capturar digitalmente objetos en su entorno inmediato, registrando con precisión incluso los detalles más pequeños y las superficies complejas sin la necesidad de puntos de referencia o marcadores predefinidos. Además, muchos escáneres láser 3D contemporáneos mejoran sus capacidades de recopilación de datos al integrar GPS, lo que proporciona una mayor comprensión de las superficies que se están escaneando [7] [8]. Estas tecnologías utilizan luz para reunir grandes volúmenes de datos, que comprenden millones de puntos discretos, que luego se compilan en archivos de nubes de puntos grandes. Este enfoque no solo mejora significativamente la precisión de las representaciones digitales, sino que también amplía el alcance para el examen detallado y la aplicación de la información recopilada. Estos dispositivos han mejorado sus principios subyacentes para no solo volverse más eficientes, sino también para capturar nubes de puntos altamente precisas.

La versatilidad de estas nubes de puntos 3D ha llevado a su aplicación generalizada en varios dominios, incluida la inspección y mapeo de estructuras y paisajes, el reconocimiento de objetos y la navegación autónoma, entre otros. A pesar de su utilidad, las nubes de puntos a menudo exhiben pequeñas variaciones en las coordenadas de cada punto escaneado, lo que indica que los detalles más finos que la resolución del escáner puede no capturarse con precisión. Esta limitación introduce un cierto grado de imprecisión en el escaneo digital, similar a las limitaciones de resolución que se encuentran en los escáneres fotográficos, estableciendo así un límite a la proximidad con la que cada traza láser puede reflejar con precisión los detalles del sujeto.

En el ámbito de las aplicaciones industriales, han surgido dos tipos principales de escáneres como líderes: escáneres de contacto y sin contacto. Los escáneres de contacto

son conocidos por su precisión, mientras que los escáneres sin contacto cuentan con una operación más rápida. La mayoría de los sistemas de medición de coordenadas espaciales 3D sin contacto dependen de sensores ópticos combinados con técnicas sofisticadas de procesamiento de señales. Estos sistemas capturan las superficies de los objetos en segmentos discretos para determinar sus coordenadas, lo que facilita la creación de una nube de puntos. Esta nube luego se puede analizar para reconstruir con precisión la forma y las dimensiones de un objeto [9]. Estas tecnologías sin contacto se clasifican principalmente en dos grupos según su funcionamiento: basadas en cámaras y basadas en láser. Además, se pueden distinguir por su forma, siendo dispositivos portátiles o de escritorio.

Los escáneres portátiles suelen emplear tecnología de luz estructurada, que proyecta un patrón de luz específico sobre el objeto. Las cámaras duales capturan y analizan los patrones de luz proyectados, discerniendo diferencias en los diversos campos de visión. Mientras tanto, otros escáneres basados en cámaras aprovechan una matriz de sensores dispuesta para recopilar datos 3D desde múltiples ángulos alrededor del objeto [10], mejorando la profundidad y la precisión de los datos recopilados. Algunos dispositivos mejoran la precisión mediante el uso de patrones de calibración, lo que permite mediciones precisas desde múltiples perspectivas alrededor del objeto. En contraste, los escáneres de triangulación láser utilizan uno o más haces de láser dirigidos a puntos específicos en la superficie del objeto [11]. Este método mide la distancia entre el escáner y estos puntos, capturando iterativamente datos espaciales para cubrir toda el área dentro del campo de visión deseado. Algunos escáneres láser están equipados con mecanismos de compensación de nivel, diseñados para contrarrestar cualquier movimiento durante el proceso de escaneo, asegurando así la integridad de los datos recopilados. Además, los escáneres especializados pueden incorporar plataformas giratorias para ajustar la posición del sujeto y la dirección de la recepción de luz, lo que permite al escáner 3D lograr mediciones con mayor precisión [12]. La prevalencia de tecnologías basadas en sensores ópticos se deriva de su naturaleza no invasiva, eliminando la necesidad de contacto directo con los objetos en estudio. Entre sus atributos más valorados se encuentra la rápida tasa a la que se pueden adquirir mediciones. No obstante, las limitaciones

notables de estos sistemas son su susceptibilidad a inexactitudes de medición causadas por el ruido óptico ambiental y las complejidades inherentes a su diseño.

A pesar de los avances en precisión y exactitud, los datos obtenidos de los procesos de escaneo 3D a menudo exhiben características que pueden inhibir significativamente la calidad de los modelos resultantes. Estas imperfecciones se manifiestan de diversas formas, incluyendo baja resolución, ruido y la presencia de puntos atípicos, cada uno de los cuales contribuye a la reducción de la claridad y la utilidad de los datos. Además, mientras que las tecnologías de escaneo basadas en cámaras son conocidas por producir datos que pueden incluir una mayor proporción de puntos atípicos, particularmente en condiciones de iluminación adversas o en entornos con visibilidad reducida debido a factores como la lluvia o la niebla, los escáneres basados en láser, aunque más precisos, pueden ser más lentos y pueden tener dificultades para detectar con precisión los reflejos de ciertas superficies, lo que puede llevar a la generación de datos adicionales de puntos atípicos. Para abordar estos problemas, se han desarrollado y aplicado diversos métodos para detectar y filtrar datos atípicos en nubes de puntos en una variedad de campos, como la bioingeniería, la ingeniería inversa, la ingeniería civil y arqueológica, así como la electrónica y la automatización industrial. La implementación de técnicas de filtrado de puntos atípicos en nubes de puntos facilita la creación y reconstrucción de mallas y modelos geométricos que se asemejan de cerca a los objetos físicos en apariencia y dimensiones. Por ejemplo, [13] discute varios métodos de filtrado de nubes de puntos, en [14] han desarrollado un algoritmo que aprovecha la segmentación del factor de variación de superficie, aplicando diferentes técnicas de filtrado a varias secciones de la nube de puntos para mejorar los resultados en diferentes regiones. Además, [15] explora el uso de la inteligencia artificial para diseñar filtros adaptativos que consideran el tamaño y la forma de la región específica de la nube de puntos que se está procesando, lo que refina aún más la precisión y la confiabilidad de los modelos digitales resultantes.

Se realizó una amplia investigación sobre las tecnologías y técnicas de escaneo 3D antes de los experimentos detallados en este trabajo de tesis, lo que llevó a la publicación de una revisión exhaustiva [16]. Esta revisión proporciona un análisis profundo de varios métodos de escaneo 3D, sus aplicaciones y avances en el tema. Adicionalmente, se llevó a cabo un estudio complementario enfocado en la detección del color de los objetos escaneados mediante técnicas de redes neuronales. A pesar de las limitaciones del TVS, entre ellas la utilización exclusiva de un láser rojo y un fotorreceptor, se desarrolló un modelo de clasificación basado en algoritmos de inteligencia artificial que permitió inferir con precisión los colores de los objetos analizados. Este resultado representa un avance en la funcionalidad del TVS, al ampliar sus aplicaciones potenciales en tareas de reconocimiento y clasificación de materiales en diversos entornos industriales y científicos.

Esta tesis está estructurada de la siguiente manera: Primero, se describen los materiales y métodos, donde se explica la función básica del TVS y el procedimiento para escanear objetos usando este sistema. También se incluye la explicación de los métodos utilizados para reducir el ruido en las nubes de puntos obtenidas del TVS, así como las técnicas de suavizado y los algoritmos utilizados para comparar los resultados, así como explicaciones de diferentes algoritmos de clasificación usados en el reconocimiento de colores. En segundo lugar, se presentan la metodología y los resultados experimentales, donde se detallan los experimentos realizados, incluyendo el escaneo de los volúmenes originales, la rotación de coordenadas, la reducción de puntos atípicos y la entropía, así como los resultados de estos algoritmos en forma de tablas. En esta sección, se incluye un enlace directo para visualizar los resultados de las nubes de puntos 3D, así como imágenes planas de las caras escaneadas. La última sección es la discusión y conclusiones, donde se resumen los resultados de los experimentos y se discuten posibles trabajos futuros.

1.1. Planteamiento del problema

En la Universidad Autónoma de Baja California, se han desarrollado múltiples sistemas de medición láser que utilizan un método denominado triangulación dinámica para medir coordenadas espaciales. Este método implica ajustar los ángulos del emisor y receptor de sistema de escaneo láser mediante controles motorizados. El proceso permite que el láser apunte a una área específica dentro del campo de visión, reflejándose en la superficie de un objeto y siendo detectado por una apertura de escaneo, donde se encuentra un fotodetector. Esta apertura gira a una velocidad constante, con el objetivo de capturar la luz láser reflejada. Basado en este principio, el laboratorio de Optoelectrónica y mediciones automatizadas del Instituto de Ingeniería de la UABC ha desarrollado tres prototipos principales de un “Sistema de Visión Técnica” (TVS), diseñado para medir coordenadas 3D dentro de un campo de visión determinado. Estos prototipos utilizan un solo sensor de luz para recopilar información de la superficie de los objetos escaneados, lo que respalda una variedad de aplicaciones. Algunos usos del sistema van desde la navegación y mapeo en entornos industriales, incluidas tuberías, hasta el monitoreo estructural y la asistencia en el campo médico para la detección de problemas en la columna vertebral. Con esta información, se puede triangular la ubicación en el espacio del objeto usando trigonometría básica. Algunos de los sistemas basados en este principio utilizan también esta técnica en combinación con cámaras para ajustar en tiempo real la posición del objeto. A través de este método, también se puede obtener información sobre la distribución de energía del objeto escaneado, así como información sobre el color del material, junto con los ángulos de los actuadores. El prototipo número 2 del TVS (Figura 1.2), en particular, se enfoca en capturar las caras planas de los objetos para crear nubes de puntos. Este sistema consta de tres componentes principales: el Posicionador Láser (LP), la Apertura de Escaneo (SA) y el Brazo de Posicionamiento (PA). Estos elementos trabajan juntos utilizando el principio de triangulación dinámica. Este método mide las distancias y ángulos de puntos 3D en los objetos observados y los convierte en coordenadas rectangulares para ensamblar las nubes de puntos, mientras el sistema ajusta los ángulos entre la apertura de escaneo

del sistema y el posicionador láser para medir con precisión los objetos. Se utiliza una base giratoria para obtener datos de múltiples caras de un objeto para su posterior procesamiento en un objeto completo, siguiendo el método descrito en la Figura 1.1.

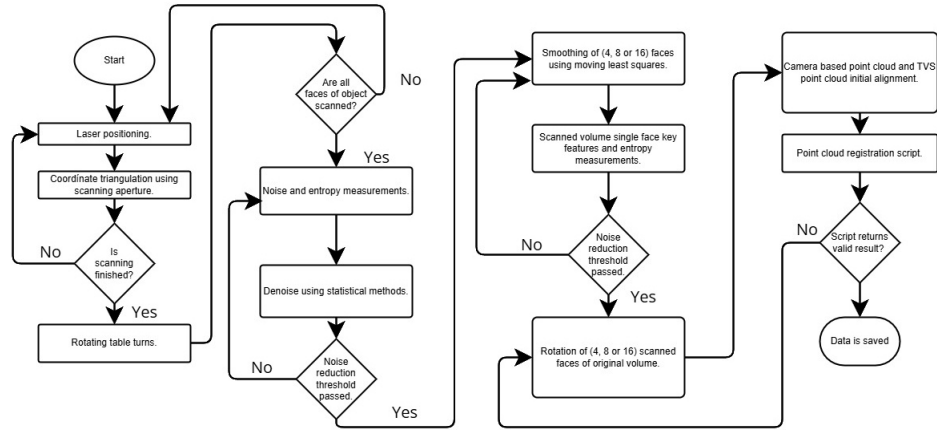


Figura 1.1: Diagrama de flujo para procesamiento de datos 3D y reducción de entropía para la reconstrucción. [17]

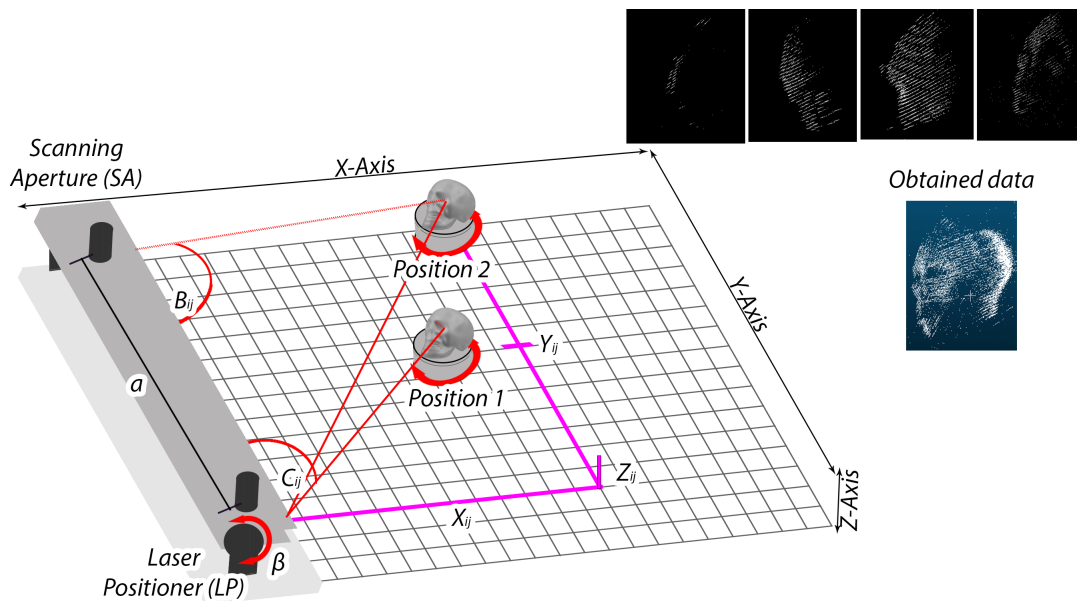


Figura 1.2: Creación de nube de puntos en TVS2

El mejor rendimiento del TVS 2 se encuentra en la posición 1 en su campo de visión para escanear puntos 3D, y ha sido previamente evaluado. Sin embargo, en esa investigación, la evaluación del campo de visión se centró en una vista frontal del objeto, mientras que en esta investigación con el propósito de obtener múltiples caras para integrar todo el volumen del objeto a escanear, se identificó que en algunos ángulos de

rotación de la mesa de escaneo el cuerpo del objeto a escanear presentaba oclusiones debido a protuberancias del objeto (como se representa en la Figura 1.2 con el cráneo en la posición 1, con este ángulo de rotación de la mesa el láser es ocluido por la nariz del cráneo y no puede ser alcanzado por la apertura de escaneo. Es importante hacer notar que solo algunas de las coordenadas 3D sufren por estos efectos y no todas las coordenadas 3D de la vista frontal son correctamente detectadas), lo que motivó la evaluación de las mediciones en la posición 2.

La apertura de escaneo actúa como un receptor para el sistema. Para el experimento, se programa una ventana de escaneo específica para que solo se obtenga información 3D del objeto bajo medición, luego se utiliza una base giratoria para escanear un número determinado de caras planas del objeto. La apertura de escaneo luego recopila datos de cada punto, como la curva de distribución de energía de cada punto medido, reflejando el láser del objeto hacia un fotosensor. El posicionador láser, por otro lado, es responsable de dirigir el haz láser a diferentes puntos programados sobre el objeto en la ventana de escaneo definida, lo que permite que el sistema mida los ángulos de reflexión (B_{ij}) a través de la apertura de escaneo.

Después de capturar los datos, se almacenan en varios archivos para su posterior procesamiento. El siguiente paso es identificar el pico de la señal para cada punto, determinar cuándo ocurre este pico de señal y relacionarlo con la posición de la apertura de escaneo en ese momento. Este paso se aplica a cada punto dentro de la ventana de escaneo establecida, preparando la base para el trabajo detallado de creación de una nube de puntos. La siguiente parte del proceso utiliza los ángulos del posicionador láser (C_{ij}) y la inclinación (ángulo β) del propio TVS. Aplicando reglas básicas de trigonometría, como se muestra en (2.1), (2.2), y (2.3) se calculan las coordenadas (x, y, z) para cada punto. Donde a es la distancia entre el PL y el SA. Este proceso convierte los puntos de datos recopilados en una nube de puntos estructurada.

$$x_{ij} = a \frac{\sin B_{ij} \sin C_{ij}}{\sin (B_{ij} + C_{ij})} \quad (1.1)$$

$$y_{ij} = a \left(\frac{1}{2} - \frac{\cos B_{ij} \sin C_{ij}}{\sin (B_{ij} + C_{ij})} \right) \quad (1.2)$$

$$z_{ij} = a \frac{\sin B_{ij} \sin C_{ij} \tanh \beta}{\sin (B_{ij} + C_{ij})} \quad (1.3)$$

1.2. Justificación

La adquisición precisa de nubes de puntos 3D es esencial en áreas como la ingeniería inversa, la metrología y el diseño asistido por computadora. Los sistemas de triangulación dinámica, específicamente el TVS2 y el TVS3, han demostrado ser herramientas que se pueden utilizar para obtener detalles como el color de objetos, daños estructurales e incluso en la industria médica. Un campo aún no explorado en el área de estos sistemas es la de escaneos de 360 grados, en los cuales se podría aplicar este tipo de procesamientos. Sin embargo, al aplicarlos en escaneos de superficies, surgen desafíos significativos relacionados con errores y efectos no lineales que afectan la calidad y precisión de los modelos tridimensionales resultantes. Estos errores pueden deberse a múltiples factores, como la calibración inadecuada de los sistemas, las limitaciones en la resolución de los sensores o las imperfecciones en los algoritmos de reconstrucción. Además, los métodos existentes para corregir estos errores a menudo no están optimizados para los sistemas de triangulación dinámica, lo que limita su efectividad. Mientras que los sistemas basados en visión con cámaras únicas o estereoscópicas presentan sus propias ventajas y limitaciones y pueden ofrecer soluciones alternativas para la captura de objetos en 360 grados, es necesario entender cómo se comparan con los sistemas de triangulación dinámica en términos de precisión, eficiencia y complejidad. Al abordar esta problemática, se espera mejorar significativamente la calidad de los modelos 3D obtenidos, lo que tendrá un impacto positivo en aplicaciones industriales y científicas que requieren alta precisión. De la misma manera, se pretende llegar a un balance entre el tiempo de escaneo de los sistemas TVS contra la calidad obtenida de los barridos. Además, al comparar estos sistemas con los basados en visión con cámaras, se propor-

cionará una comprensión más profunda de las fortalezas y debilidades de cada enfoque. Esto permitirá a profesionales y académicos seleccionar la tecnología más adecuada para sus necesidades específicas, optimizando recursos y mejorando resultados.

1.3. Objetivos

La presente investigación se enfoca en mejorar la precisión y calidad de las nubes de puntos 3D obtenidas mediante los sistemas de triangulación dinámica TVS2 y TVS3, especialmente al escanear objetos en 360 grados. Con este contexto, se establecen los siguientes objetivos para guiar el desarrollo y alcance de la investigación.

1.3.1. Objetivo general

Desarrollar un método y algoritmos que disminuyan los errores en las nubes de puntos 3D generadas por los sistemas de triangulación dinámica TVS2 y TVS3 al escanear objetos en 360 grados, creando, si es necesario, un método específico para este tipo de sistema de escaneo. Además, comparar las diferencias de uso entre estos sistemas de triangulación dinámica y los sistemas basados en visión que utilizan cámaras únicas o estereoscópicas.

1.3.2. Objetivos específicos

- **Investigar el estado del arte** de distintos sistemas de escaneo de puntos, así como métodos y algoritmos de corrección de errores y construcción de modelos basados en nubes de puntos.
- **Completar el desarrollo del sistema TVS3**, iniciado durante el proyecto de maestría, para mejorar sus capacidades y funcionalidades.
- **Construir una base rotatoria** para el escaneo en 360 grados de objetos e integrarla al sistema de control de posicionamiento existente.

- **Comparar diferentes algoritmos de corrección de errores** para familiarizarse con su funcionamiento y evaluar su efectividad en la mejora de las nubes de puntos.
- **Desarrollar métodos y algoritmos propios** específicos para su uso con sistemas de triangulación dinámica, para la reducción de errores y mejora de la precisión.
- **Diseñar y realizar experimentos** para comparar los métodos y algoritmos desarrollados con los existentes, evaluando su desempeño en condiciones controladas.
- **Redactar documentos científicos** para su publicación y fomentar la colaboración en investigación con instituciones extranjeras, contribuyendo al intercambio de conocimiento en el campo.

1.4. Hipótesis

Es posible utilizar un sistema de visión técnica para el escaneo completo de todas las caras de un objeto. Además, se puede desarrollar un algoritmo que disminuya los efectos no lineales en las nubes de puntos adquiridas utilizando un sistema de visión técnica. De igual manera, es factible aumentar la precisión de los modelos tridimensionales obtenidos a partir de estas nubes de puntos mediante el uso de un sistema de triangulación dinámica.

1.5. Antecedentes

El desarrollo de modelos tridimensionales de objetos y escenas ha cobrado gran importancia en diversos campos del conocimiento, especialmente en áreas que requieren la aplicación de ingeniería inversa para obtener diseños asistidos por computadora de elementos de los cuales no se dispone de información específica. Los sistemas de escaneo láser para mediciones de coordenadas en 3D se han convertido en herramientas

indispensables para la digitalización de características físicas y geométricas de objetos y entornos.

Los sistemas de visión artificial utilizados para obtener mediciones 3D dependen en gran medida de la precisión mecánica y electrónica de su construcción, así como de la correcta alineación del camino óptico. En particular, los sistemas de barrido láser pueden verse afectados por factores fuera del control del operador, como interferencias causadas por electricidad estática, vibraciones en los motores controlados mediante bucles cerrados, variaciones en los ángulos reportados por el controlador de posicionamiento y ruido ambiental lumínico que interactúa con el sensor del sistema. Además, la reflexión natural de la luz láser en diferentes tipos de superficies y formas de objetos puede introducir errores en las mediciones.

En la Universidad Autónoma de Baja California (UABC), se ha desarrollado la técnica de triangulación dinámica para la medición de coordenadas espaciales, dando lugar a tres prototipos de un Sistema de Visión Técnica (TVS). Esta técnica implica cambiar dinámicamente los ángulos del emisor y receptor láser del sistema mediante control motorizado. El láser apunta a una área específica dentro del campo de visión, se refleja en la superficie del objeto y es detectado por una apertura de escaneo que gira a velocidad constante para capturar la luz reflejada. El prototipo 2 del TVS (TVS2) consta de dos componentes mecánicos principales: el Posicionador Láser (PL) y la Apertura de Escaneo (AE). Estos elementos trabajan en conjunto utilizando el principio de triangulación dinámica. Sin embargo, debido al uso de motores paso a paso, el TVS2 presenta un campo de visión discreto. Esto limita la resolución y hace imposible el escaneo de objetos de menor tamaño que la resolución del sistema, ya que la distancia entre los puntos aumenta con la distancia de escaneo. Para superar estas limitaciones, se desarrolló el prototipo 3 del TVS (TVS3), que reemplaza los motores paso a paso por motores de corriente directa utilizados como servomotores mediante la adición de codificadores rotativos. Esto permitió obtener un campo de visión continuo, aumentando considerablemente la resolución del sistema y posibilitando el escaneo de objetos con mayor precisión. En investigaciones previas, se demostró que el uso de un amplificador de transimpedancia en el TVS3 mejora la adquisición de datos adicionales del objeto es-

caneado, proporcionando información sobre formas, colores, texturas y materiales. Este amplificador también extendió el rango máximo de escaneo hasta 3 metros, superando a otros sistemas similares que cuentan con rangos de medio hasta un metro.

A pesar de estos avances, los sistemas de escaneo 3D enfrentan desafíos significativos al realizar escaneos completos de objetos en 360 grados. La rotación del objeto y la transformación de sus coordenadas a través de matrices y coordenadas homogéneas para crear nubes de puntos integrales requieren abordar problemas como:

- Datos atípicos y redundantes: Las nubes de puntos pueden contener puntos fuera de orden o redundantes, lo que dificulta el procesamiento y puede afectar la calidad del modelo 3D resultante.
- Falta de relación topológica: La ausencia de información sobre proporciones y relaciones entre puntos dificulta el ordenamiento y unión de los puntos 3D en la nube.
- Errores de medición: Factores como oclusiones láser causadas por protuberancias del objeto, variaciones en los ángulos de los actuadores y ruido ambiental pueden introducir errores no lineales en las mediciones.

Diversos autores han documentado estas dificultades en el procesamiento de nubes de puntos. Por ejemplo, en [18], se aborda el desafío del ordenamiento de puntos en ausencia de relaciones topológicas claras. Además, investigaciones como [19] comparan el uso de escaneo mediante cámaras y escaneo láser, evidenciando las limitaciones y ventajas de cada método.

En campos donde la precisión es crítica, como la ingeniería inversa de piezas con tolerancias mecánicas estrictas o en aplicaciones de visión para la evasión de objetos, estos errores pueden reducir significativamente la utilidad del proceso. Por lo tanto, es esencial desarrollar métodos y algoritmos que reduzcan los efectos de comportamiento no lineal en los sistemas de visión técnica basados en barrido láser y triangulación dinámica.

Marco teórico

2.1. Investigación del estado del arte

Los sistemas de escaneo han cambiado el cómo se aplican múltiples disciplinas, como las tecnologías de detección, inspección, topografía y navegación, que por muchos años fueron hechas de manera visual y en general han encontrado un extenso campo de aplicaciones. Estos sistemas se pueden encontrar en múltiples campos haciendo múltiples tareas, como en la medicina, en la ingeniería civil al usarse para el mapeo de terrenos y edificios junto a la inspección de fallas estructurales así como en minería e ingeniería inversa, por nombrar algunos. En algunas aplicaciones de estos, era común el uso de plataformas estacionarias haciendo uso de escáneres comerciales; sin embargo, las tendencias en los últimos años de desarrollo de estas tecnologías tienden al uso de estaciones móviles de escaneo en múltiples plataformas diferentes [20], desde automóviles hasta botes, mochilas, helicópteros y drones. Similarmente a esto, los métodos de procesamiento de las nubes de puntos resultantes han cambiado de herramientas sencillas que permitieran la visualización de las nubes de puntos a procesos automáticos o semiautomáticos mediante machine learning, el cual es una de las herramientas más representadas en el estado del arte. Una vista de las múltiples técnicas y aplicaciones para el escaneo de estructuras puede verse en el artículo [21] y en [6] se puede encontrar una descripción general sobre la historia y las aplicaciones de los escáneres láser 3D. A continuación se nombran algunas de las características más importantes y trabajos relacionados de diferentes tipos de sistemas de visión.

Las coordenadas espaciales 3D pueden obtenerse mediante métodos de contacto y no contacto. El método de contacto más común se ejecuta mediante los brazos robóticos de las máquinas de medición por coordenadas (CMM), las cuales miden la geometría de los objetos detectando puntos discretos sobre su superficie con una sonda (mecánica, óptica, láser o luz blanca). Este método requiere una trayectoria de sonda predefinida y conocimiento del cuerpo a medir [22].

Los métodos sin contacto para la obtención de coordenadas espaciales en 3D se basan principalmente en principios ópticos, acústicos y magnéticos [23]. Entre estos, los métodos ópticos son los más utilizados debido a su naturaleza no destructiva y su capacidad para adquirir datos con rapidez sin necesidad de contacto físico con los objetos. No obstante, las mediciones obtenidas mediante estos métodos pueden presentar valores atípicos e imprecisiones como consecuencia del ruido ambiental que afecta a los sensores, por ejemplo, la interferencia de la luz solar. Sin embargo, estas limitaciones pueden mitigarse mediante técnicas de preprocesamiento de datos, lo que permite alcanzar altos niveles de precisión y flexibilidad en la adquisición de información.

Los métodos sin contacto pueden clasificarse en dos categorías principales: activos y pasivos [24]. Los métodos activos emplean la emisión de una señal óptica para interactuar con el objeto, generalmente mediante la iluminación controlada de la superficie a medir. En contraste, los métodos pasivos no requieren la emisión de luz adicional, basándose en la detección y análisis de la radiación reflejada o emitida por los objetos en su entorno natural.

2.2. Sistemas de visión mediante cámaras

Los métodos de escaneo usando sensores de visión se utilizan ampliamente en el área de la navegación de robots móviles y muestran un buen rendimiento en las tareas de detección de objetos 3D; sin embargo su procesamiento suele ser lento [25] y propensos son propensos a datos atípicos en la densidad de las imágenes y en superficies con colores uniformes, además de tener problemas para hacer corresponder diferentes

imágenes tomadas. Algunos de los usos de estos sistemas de visión son, por ejemplo, el trabajo [26], donde se utilizan marcas visuales estacionarias para la reconstrucción en tres dimensiones de diferentes áreas. En [27] se utiliza una cámara de un teléfono inteligente para la recreación en modelos 3D mediante software. Otro ejemplo de un sistema basado en visión está en [28] donde se usan dos cámaras para la ubicación de un robot flexible para endoscopias en el sector médico.

2.3. Sistemas de visión mediante láser

2.3.1. Interferometría y escaneo láser de luz estructurada

La interferometría y el escaneo láser de luz estructurada se clasifican como métodos activos, ya que requieren la emisión de luz para detectar y caracterizar objetos a partir de la interacción entre la luz emitida y la superficie del objeto. La interferometría óptica es ampliamente utilizada para la medición de distancias, empleando señales ópticas transducidas por un fotodetector para obtener información sobre una escena, proporcionando tanto coordenadas espaciales como imágenes [29].

Por otro lado, el escaneo láser se basa en la reflexión especular, donde un haz láser incide sobre una superficie con un ángulo específico y es reflejado con un ángulo similar para ser capturado por un fotosensor. Esta tecnología permite la adquisición de nubes de puntos con alta densidad y a una tasa de captura rápida. Sin embargo, las mediciones obtenidas pueden presentar imprecisiones en regiones cercanas a bordes afilados y en estructuras con orificios, ya que estas características pueden no ser identificadas con precisión [30].

2.3.1.1. Triangulación láser

La triangulación láser es un método de escaneo opto-electro-mecánico que se fundamenta en la disposición física y el ángulo relativo entre la fuente de luz láser y el fotosensor (opto-detector). La profundidad de los objetos se estima a partir de dos puntos cuya posición relativa es conocida. Este tipo de escaneo es particularmente sensible

a las características superficiales del objeto bajo análisis, incluyendo sus propiedades reflectantes y la iluminación ambiental. No obstante, la intensidad del láser y la sensibilidad del sensor opto-electrónico pueden ajustarse para optimizar su desempeño en entornos específicos.

En este sistema, la luz láser reflejada por la superficie del objeto es guiada mediante espejos direccionales y concentrada a través de una lente, mientras que las fuentes de luz externas son filtradas mediante procesamiento de señales. La alineación precisa de la trayectoria óptica es esencial para garantizar mediciones confiables. Diversas configuraciones de triangulación láser han sido reportadas en la literatura [31].

La Fig. 2.1 muestra dos configuraciones clásicas. En la configuración mostrada en la Fig. 2.1(a), la luz emitida incide sobre la superficie del objeto y es reflejada hacia un espejo inclinado, el cual redirige la luz reflejada hacia un sistema de lentes para colimación, que posteriormente la dirige al opto-detector [32–39]. En la configuración mostrada en la Figura 2.1 (b), la luz emitida es redirigida inicialmente por un espejo inclinado hacia el objeto. Posteriormente, la luz reflejada regresa al mismo espejo, que la redirige hacia un sistema de lentes para colimación antes de ser capturada por el opto-detector.

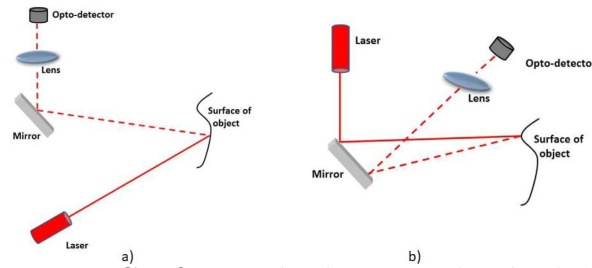


Figura 2.1: Configuración de triangulación de láser

2.3.1.2. Detección y Alcance de la Luz

El método *Light Detection and Ranging* (LiDAR), también conocido como medición láser, se basa en la emisión de pulsos de luz modulados en el dominio temporal y su posterior detección tras reflejarse en un objeto. Este enfoque, identificado también como método de Tiempo de Vuelo (*Time of Flight*, ToF), emplea pulsos ultracortos de luz que viajan hasta la superficie del objeto, se dispersan y regresan al sensor óptico. La

profundidad se calcula a partir del tiempo transcurrido entre la emisión del pulso y su recepción.

A diferencia de la triangulación láser, LiDAR no requiere el uso de configuraciones geométricas complejas para la estimación de la profundidad, lo que lo hace particularmente adecuado para aplicaciones de largo alcance que demandan alta precisión en la medición. Diversas implementaciones de este sistema pueden encontrarse tanto en la literatura como en el mercado, como en el caso del Kinect One [40].

La Figura 2.2 presenta dos configuraciones clásicas [41]. En la configuración mostrada en la Figura 2.2(a), se utiliza un único emisor y un único receptor. En contraste, la configuración de la Figura 2.2(b) en la cual se hace uso de una matriz compuesta por múltiples emisores y receptores, permitiendo una mayor cobertura espacial y mejorando la resolución del sistema.

Para obtener mediciones precisas, estos sistemas requieren una configuración estable de la trayectoria óptica y el uso de componentes mecánicos de alta precisión. Además, la calidad de los resultados se ve favorecida si la superficie del objeto presenta propiedades reflectantes adecuadas.

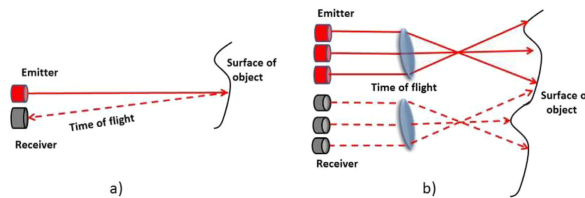


Figura 2.2: Configuración de LIDAR

2.3.2. Visión por Computadora Fotogramétrica

La fotogrametría permite obtener imágenes bidimensionales en las que falta una dimensión de la escena para representar el mundo tridimensional. A través de la visión por computadora, es posible estimar esta tercera dimensión, la “profundidad”, con el objetivo de obtener información sobre el tamaño, la distancia, la posición y la orientación de los objetos, así como estructurar la escena de manera más detallada. Estos métodos se clasifican principalmente como pasivos, ya que emplean fotografías o secuencias de video capturadas por cámaras para extraer coordenadas tridimensionales a partir de

imágenes bidimensionales, sin necesidad de emitir señales luminosas.

En [42] se analiza la distinción entre fotogrametría y visión por computadora. Mientras que la fotogrametría se enfoca en la precisión métrica de las coordenadas tridimensionales y en la calibración de los parámetros intrínsecos de las cámaras (como la distancia focal y la distorsión radial), la visión por computadora tiene como objetivo la automatización de la extracción de información a partir de imágenes, incluyendo segmentación, reconocimiento de patrones y redes semánticas. A diferencia de la fotogrametría, la visión por computadora no requiere necesariamente conocer los parámetros intrínsecos de la cámara; sin embargo, en aplicaciones de reconstrucción tridimensional, ambas disciplinas comparten el propósito de obtener modelos 3D y su apariencia final desde distintas perspectivas. En última instancia, ambas técnicas buscan identificar características comunes en las imágenes para reconstruir la geometría de la escena observada.

La precisión y la velocidad de estos métodos dependen tanto de las capacidades de la cámara para adquirir las imágenes como del método de correlación empleado para establecer correspondencias entre imágenes. Una diferencia fundamental entre la interferometría y los escáneres láser de luz estructurada respecto a la visión por computadora fotogramétrica es la forma en que se representa la información. Mientras que en los primeros la salida es una nube de puntos obtenida directamente, en la visión fotogramétrica la salida inicial consiste en imágenes, de las cuales es necesario extraer características para estimar la profundidad y generar posteriormente la nube de puntos [43, 44].

Las técnicas basadas en cámaras pueden clasificarse en dos enfoques principales para la extracción de características: (i) métodos basados en puntos y (ii) métodos basados en líneas, siendo el primero el más utilizado. Una vez extraídas las características relevantes, se procede al emparejamiento de puntos clave, seguido de la estimación del movimiento de la cámara, la reconstrucción 3D dispersa mediante triangulación, la corrección de los parámetros del modelo (incluyendo los parámetros intrínsecos y extrínsecos de la cámara, como la rotación y la traslación), la recuperación de la escala absoluta y, finalmente, la generación de una nube de puntos 3D densa [45].

Adicionalmente, la precisión de estos métodos varía en función de las característi-

cas superficiales del objeto analizado. Si la superficie presenta una textura detallada, la precisión puede aumentar, ya que la textura proporciona referencias útiles para la extracción de características. En cambio, en superficies homogéneas o con poca textura, la precisión disminuye debido a la falta de elementos distintivos que faciliten la correspondencia de puntos entre imágenes.

Existen diversas configuraciones de cámaras utilizadas en visión fotogramétrica, siendo la más común la visión binocular o estereovisión, que se considera un método pasivo. Sin embargo, los métodos basados en cámaras pueden combinarse con la emisión de señales luminosas, donde la proyección de patrones de luz sirve como referencia para la reconstrucción tridimensional. Este enfoque es conocido como visión por computadora fotogramétrica basada en patrones de luz estructurada [46].

2.3.2.1. Procesamiento de Imágenes con Monovisión

La percepción de la profundidad y la extracción de características tridimensionales desde un único punto de observación es posible mediante una cámara monocular. Esta percepción puede lograrse tanto si la cámara como el objeto se encuentran en movimiento, permitiendo la correlación de características del objeto con sus coordenadas espaciales.

Los enfoques de visión artificial para la medición espacial 3D utilizando monovisión se clasifican en fotogrametría clásica y estructura a partir del movimiento (SfM) [47]. Recientemente, ha surgido un nuevo enfoque basado en aprendizaje profundo para estimar la profundidad y las coordenadas espaciales 3D a partir de imágenes 2D capturadas por cámaras monoculares [48].

La fotogrametría clásica para la estimación de la profundidad invierte el proceso fotográfico para reconstruir el espacio del objeto a partir de imágenes, estableciendo una relación matemática entre la imagen y el objeto para derivar las coordenadas terrestres de los puntos a partir de imágenes superpuestas. Este proceso, denominado ajuste por bloques, se lleva a cabo mediante el método de ajuste por bloques de haces, en el cual las posiciones y rotaciones de todos los haces en el bloque se ajustan simultáneamente en el espacio para lograr la mejor intersección de los rayos conjugados de cada punto del

conjunto. Este método depende del control de la geometría de captura de la cámara.

El proceso general de la fotogrametría clásica incluye:

- Comparación de imágenes para la extracción de puntos de enlace.
- Uso de aprendizaje automático para refinar la ubicación de los puntos de enlace.
- Aplicación del ajuste de bloques de haces para optimizar las entradas del algoritmo de mínimos cuadrados no lineales, minimizando errores en la posición de los puntos de enlace [49].

El procedimiento comienza con la identificación de la perspectiva de la cámara mediante el punto de fuga y la proyección en perspectiva de los rayos sobre el plano de la imagen. Basándose en el principio epipolar, se pueden emplear diferentes enfoques:

- **Estimación de la geometría:** selección de las coincidencias de características más confiables y eliminación de valores atípicos.
- **Enfoque incremental:** inicialización con dos imágenes de alta calidad, alta densidad de coincidencias y una pose de cámara correctamente definida.
- **Enfoque adaptativo:** ajuste dinámico conforme se incorporan nuevas imágenes.
- **Enfoque jerárquico:** agrupación de coincidencias de características en imágenes para optimizar el procesamiento.

La estructura a partir del movimiento (SfM) se basa en la estimación de la homografía en múltiples vistas de una escena. Si la cámara se mueve alrededor de un objeto fijo o si el objeto se desplaza mientras la cámara permanece estática, es posible extraer características del objeto para reconstruirlo en 3D a partir de una serie de imágenes 2D capturadas desde distintos puntos de vista [50].

Esta técnica emplea principios matemáticos para superar algunas de las limitaciones geométricas de la fotogrametría clásica, particularmente aquellas relacionadas con errores en el posicionamiento. SfM permite la reconstrucción tridimensional sin requerir información previa sobre la ubicación, orientación o calibración de la cámara, ni la presencia de puntos de referencia en la escena.

El proceso general de SfM incluye:

- **Detección de puntos clave:** identificación de características en imágenes individuales mediante algoritmos de detección de bordes, detección de esquinas, detección de manchas y detección de crestas. Para describir estas características y generar vectores distintivos, se emplean métodos como SIFT, SURF, ASIFT, FAST, FREAK, MSD, HOG y GLOH [45].
- **Emparejamiento de puntos clave:** búsqueda de correspondencias entre pares de imágenes utilizando algoritmos como RANSAC, ORSA, ANN y FLANN, con el objetivo de eliminar coincidencias erróneas.
- **Reconstrucción 3D:** estimación de la pose de la cámara y ubicación de los puntos tridimensionales mediante:
 - Estimación del movimiento de la cámara utilizando la Transformada Lineal Directa (DLT), determinando los parámetros intrínsecos y extrínsecos.
 - Reconstrucción 3D dispersa mediante triangulación, calculando la posición tridimensional de los puntos a partir de sus correspondencias en múltiples imágenes.
 - Corrección de los parámetros del modelo mediante Ajuste de Bloques (BA), refinando la ubicación de los puntos.
 - Recuperación de la escala absoluta, determinando la escala de la nube de puntos mediante registro georreferenciado, medidas manuales o calibración previa.
 - Generación de una reconstrucción 3D densa para obtener detalles completos de la escena [51].

2.3.3. Capacidades de sistemas 2D y 3D

Los sistemas láser 2D, como su nombre lo indica, utilizan sensores para capturar las dimensiones X e Y . Esto puede lograrse de diversas maneras, por ejemplo, mediante

la proyección de un anillo de luz o el giro de una fuente láser. Tanto los sistemas de proyección de luz como los sensores LiDAR pueden obtener el mismo tipo de dimensiones X e Y . El movimiento sucesivo de los sensores 2D en el eje Z permite recolectar múltiples secciones de datos 2D, que luego pueden presentarse como datos 3D. Si bien es posible adquirir datos 3D con este método, estos sistemas no se consideran sensores 3D reales. La detección con sensores 2D se utiliza principalmente para operaciones de detección y medición de distancias.

Los sistemas 3D operan de forma similar a los 2D; es decir, además de la detección de las coordenadas X e Y , recopilan también los datos del eje Z . Por lo general, esta información adicional se obtiene mediante el uso de múltiples láseres dirigidos en ángulos distintos y líneas de proyección específicas.

2.3.4. Triangulación dinámica

En la UABC se ha desarrollado la técnica de triangulación dinámica para la medición de coordenadas espaciales y, junto con ella, tres prototipos de un sistema de visión técnica que realiza estas mediciones. La triangulación dinámica consiste en cambiar de manera dinámica el ángulo del emisor y del receptor del sistema, utilizando motores. El apuntador láser se dirige hacia una sección específica del campo de visión; cuando incide en la superficie de un objeto, la luz se refleja y es capturada por la apertura de escaneo, que gira a velocidad constante en busca de esta reflexión láser. Esto hace que la triangulación dinámica, al emplear ángulos no fijos para el emisor y el receptor, presente un intervalo de medición independiente del tamaño del sensor utilizado para la detección. Al estar en movimiento los elementos del sistema, se pueden adaptar automáticamente a la superficie del objeto que se escanea, a diferencia de la triangulación estática, donde el tamaño del sensor está vinculado a la luz que retorna.

Mediante estos elementos (ver Figura 2.3), de los cuales se obtiene una curva gaussiana con la información de la distribución de energía del objeto escaneado, además de los ángulos en los que se encuentran los actuadores, es posible emplear trigonometría simple para triangular la posición del objeto [52]. Como un ejemplo de su aplicación, en [53] se utiliza el sistema TVS para la navegación de un robot móvil.

Para el propósito de experimentación de esta tesis, se programa una ventana de escaneo específica para que solo se obtenga información 3D del objeto bajo medición, luego se utiliza una base giratoria para escanear un número determinado de caras planas del objeto. La apertura de escaneo recopila datos de cada punto, con la curva de distribución de energía de cada punto medido, reflejando el láser del objeto hacia un fotosensor. El posicionador láser, por otro lado, es responsable de dirigir el haz láser a diferentes puntos programados sobre el objeto en la ventana de escaneo definida, lo que permite que el sistema mida los ángulos de reflexión (B_{ij}) a través de la apertura de escaneo.

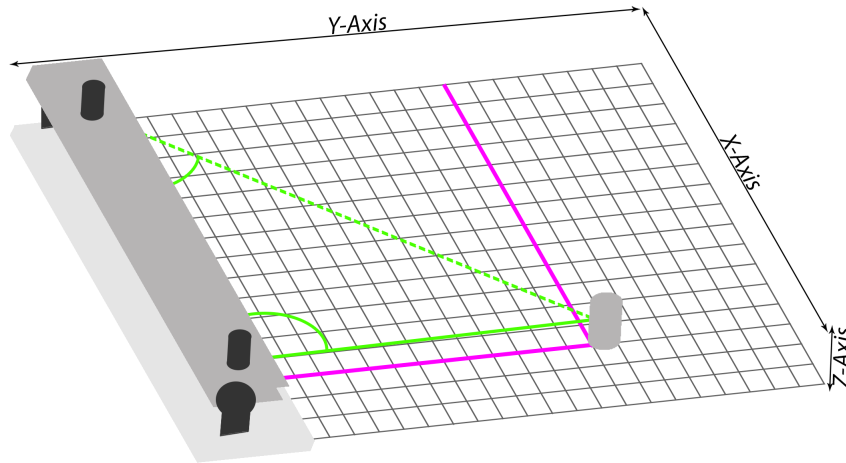


Figura 2.3: Triangulación dinámica

Después de capturar los datos, se almacenan en varios archivos para su posterior procesamiento. El siguiente paso es identificar el pico de la señal para cada punto, determinar cuándo ocurre este pico de señal y relacionarlo con la posición de la apertura de escaneo en ese momento. Este paso se aplica a cada punto dentro de la ventana de escaneo establecida, preparando la base para el trabajo detallado de creación de una nube de puntos. La siguiente parte del proceso utiliza los ángulos del posicionador láser (C_{ij}) y la inclinación (ángulo β) del propio TVS. Aplicando reglas básicas de trigonometría, como se muestra en (2.1), (2.2), y (2.3) se calculan las coordenadas (x, y, z) para cada punto. Donde a es la distancia entre el PL y el SA. Este proceso convierte los puntos de datos recopilados en una nube de puntos estructurada.

$$x_{ij} = a \frac{\sin B_{ij} \sin C_{ij}}{\sin (B_{ij} + C_{ij})} \quad (2.1)$$

$$y_{ij} = a \left(\frac{1}{2} - \frac{\cos B_{ij} \sin C_{ij}}{\sin (B_{ij} + C_{ij})} \right) \quad (2.2)$$

$$z_{ij} = a \frac{\sin B_{ij} \sin C_{ij} \tanh \beta}{\sin (B_{ij} + C_{ij})} \quad (2.3)$$

2.3.4.1. Time of flight

El tiempo de vuelo (*Time of flight*) consta de una fuente emisora de láser, dos fotoreceptores y una unidad de procesamiento de señales. Este método emplea el tiempo, la amplitud o la frecuencia como variables para calcular la distancia entre la fuente láser y su reflejo en la superficie. El escáner láser mide el tiempo de vuelo absoluto Δt de un pulso láser emitido y recibido; por lo tanto, el pulso debe llegar y reflejarse de forma perpendicular sobre la superficie medida, de modo que el transmisor y el receptor se combinen en la misma unidad de medición. Como ventaja, estos métodos ofrecen alta velocidad de reacción; su principal desventaja radica en el costo elevado, dado que requieren alta resolución.

Un ejemplo de este método de medición puede verse en [54], donde se utiliza el tiempo de vuelo directo en un sistema LiDAR con 32 sensores diferentes, y en [55] se aplica una técnica de corrección de nubes de puntos para mejorar la calidad de los datos obtenidos mediante LiDAR. En [56], se emplea un escáner láser para la detección de objetos en un sistema de evasión para vehículos aéreos no tripulados y como apoyo al piloto en vuelos manuales. En [57], se describe un escáner láser con 6-GDL para un método de mapeo de terreno en tiempo real y con baja desviación durante el movimiento. Por su parte, en [3] se mide la estructura de la vegetación con un escaneo láser, obteniendo información sobre la copa de la planta sin tocarla. En [4], se introducen nuevos índices de vegetación para cuantificar la configuración espacial tridimensional de comunidades de plantas mediante escaneo láser terrestre. Finalmente, en [5] se presenta un análisis de un sistema de rastreo láser 3D enfocado en la inmunidad a la luz solar y en la resolución

geométrica; y en [25], se utiliza un escáner láser 3D de triangulación para dotar a un robot miniatura de navegación y evasión de obstáculos.

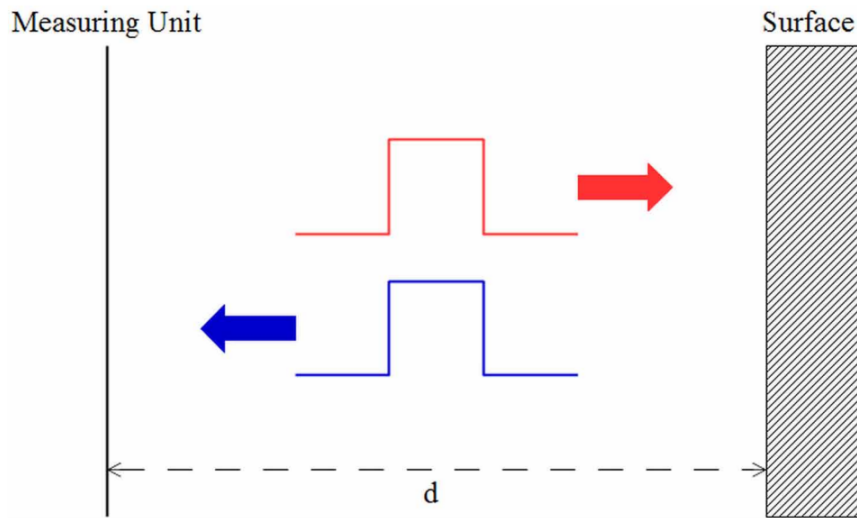


Figura 2.4: Tiempo de vuelo [58].

2.3.4.2. Medición de fase

Para el método de medición de fase, se envía una señal láser armónica que se superpone con su señal reflejada. La principal ventaja de este método radica en su alta resolución, mientras que su mayor desventaja es la distancia limitada para el escaneo de objetos. La distancia entre la unidad de medición y la superficie medida se determina a partir del desplazamiento de fase entre el tren de pulsos láser emitido (rojo) y el reflejado (azul), como se muestra en la Figura 2.5.

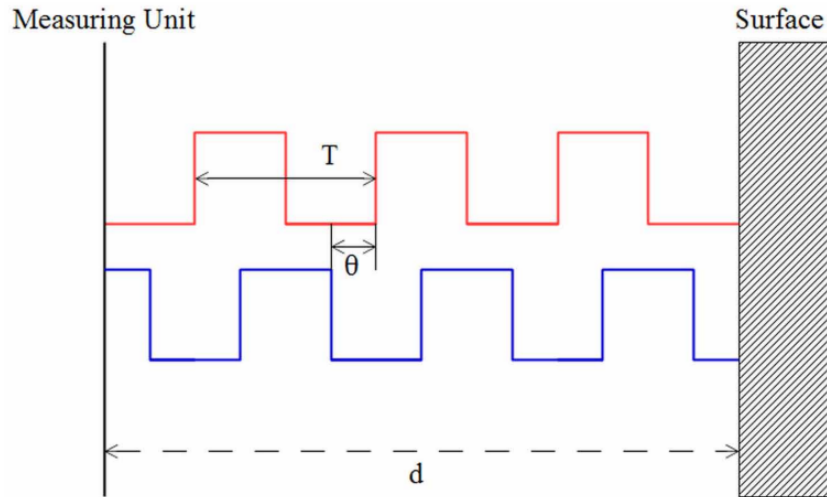


Figura 2.5: Medición de fase [58].

2.4. Algoritmos de visualización 3D

Los algoritmos de visualización y procesamiento de nubes de puntos son diversos. En el ámbito de la corrección y reconstrucción 3D de nubes de puntos, el trabajo mostrado en [59] reconstruye secciones seleccionadas automáticamente de una nube 3D basándose en la corrección de finalización de profundidad. En [60], se analizan diversos factores que influyen en la precisión de los puntos escaneados por un sistema láser 3D y se formulan dos modelos de corrección de errores. Por su parte, [61] propone una modificación al método de comparación M3C2 para el análisis de cambios geomórficos en nubes de puntos, con el objetivo de obtener estimaciones menos sesgadas en la detección de niveles.

2.5. TVS1

Desarrollado en el laboratorio de optoelectrónica y mediciones automatizadas de la UABC, el TVS está constituido principalmente por un sistema de triangulación dinámica que emplea motores a pasos para posicionar tanto la apertura de escaneo como un apuntador láser. El apuntador láser se dirige a una sección específica del campo de visión; al incidir sobre la superficie de un objeto, la luz se refleja y es capturada por

la apertura de escaneo, que gira sobre un motor de corriente continua en busca de esa reflexión [62]. Esta triangulación dinámica, basada en ángulos variables para el transmisor y el receptor, posibilita un rango de medición independiente del tamaño del sensor [52]. Aunque este prototipo fue sustituido por el TVS2 y el TVS3 y ya no se utiliza, el principio de funcionamiento es el mismo para los dos prototipos posteriores.

2.6. TVS2

El predecesor del sistema prototipo actual, denominado TVS2, consta de dos partes mecánicas principales: el posicionador láser y la apertura de escaneo. Aplicando el principio de triangulación dinámica, este sistema presenta ventajas respecto de métodos tradicionales basados en cámaras, pues ambas partes mecánicas se desplazan mediante motores a pasos, generando un campo de visión discreto. Sin embargo, dicha discretización impide el escaneo de objetos más pequeños que la resolución del sistema, constituyendo su principal desventaja. Tal como se menciona en [52], este campo de visión puede observarse en la Figura 2.6, donde la distancia entre los puntos p se incrementa a medida que crece la distancia de escaneo d .

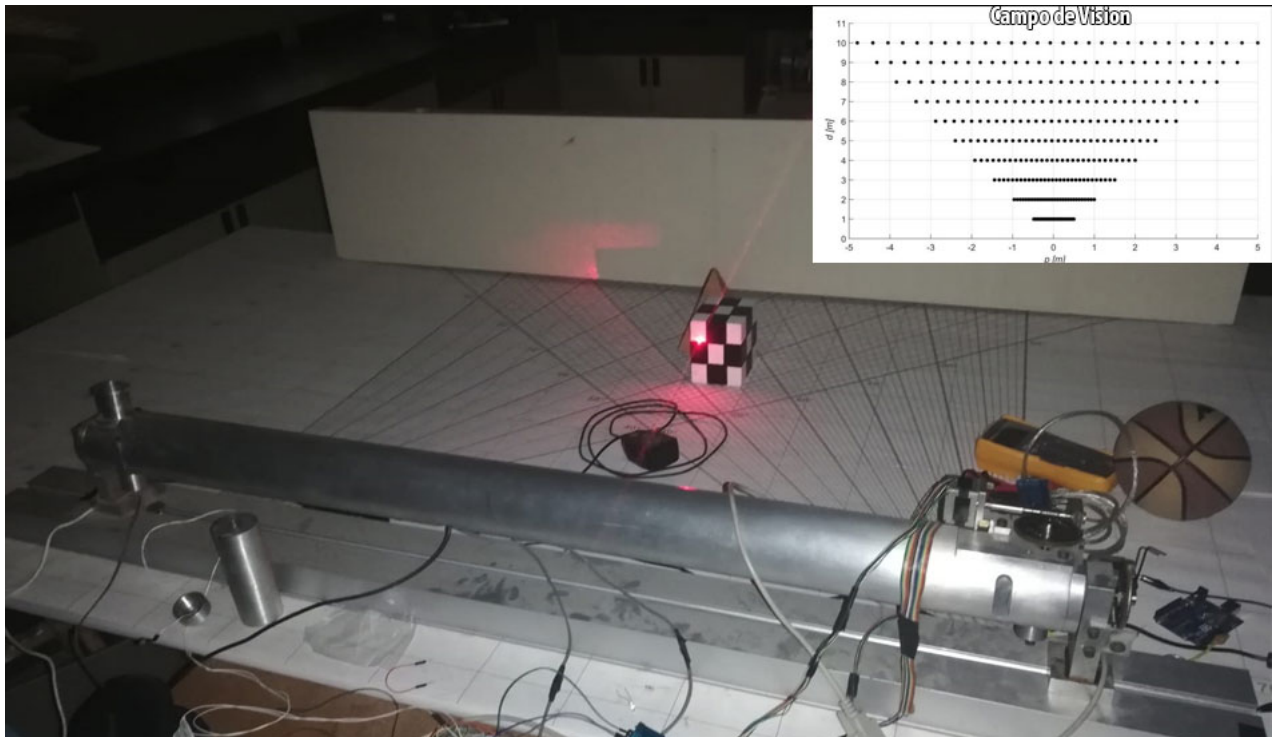


Figura 2.6: Campo de visión discreto de TVS2 [52].

2.6.1. Construcción mecánica del TVS2

El TVS2 utiliza un sistema con engranes y tornillos sinfín para rotar sobre su propio eje así como el posicionador láser. En operación normal, el escaneo horizontal de la apertura de escaneo se mueve mediante un motor DC mediante un voltaje constante o mediante un sistema de control PID según sea requerido, mientras que el movimiento del posicionador láser se realiza con un motor a pasos mediante el cual se mueve un espejo de 45° para proyectar un haz de luz láser. Las Figuras 2.7 y 2.8 ilustran los componentes clave del prototipo 2 del TVS, detallando la construcción y funcionalidad del sistema. La Figura 2.7 se centra en los elementos del posicionador láser, destacando: (1) un motor paso a paso que permite la rotación alrededor de su eje; (2) otro motor paso a paso acoplado a un espejo de 45° para dirigir el haz láser; (3) una combinación de tornillo sin fin y engranaje que facilita la rotación del brazo de posicionamiento; (4) un mecanismo similar para ajustar la posición del espejo de 45° ; y (5) el propio espejo de 45° , que refleja el láser hacia el objeto objetivo.

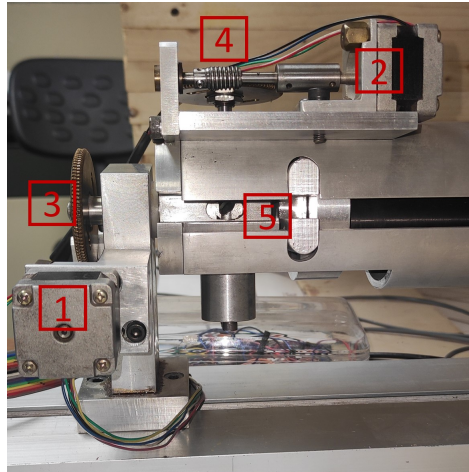


Figura 2.7: Posicionador Láser del TVS2

Por otro lado, la Figura 2.8 muestra la apertura de escaneo, compuesta por: (A) un espejo de 45° para la reflexión del láser; (B) lentes que enfocan la luz láser; (C) un motor DC para la rotación del espejo; (D) un fotosensor para detectar la luz láser; y (E) cojinetes de teflón para un movimiento suave. El motor DC de la apertura de escaneo gira libremente, alimentado por una fuente de voltaje regulada para alcanzar la velocidad deseada. Sin embargo, debido a esta configuración, la velocidad del motor puede variar, lo que requiere mediciones de velocidad en cada revolución con un optoacoplador para mantener la precisión. El posicionador láser emplea un láser rojo de 10mW, dirigido mediante motores paso a paso capaces de 19200 pasos por revolución. El fotosensor del sistema, un fototransistor modelo BPW77N de VISHAY Semiconductor, captura la luz láser reflejada para su posterior procesamiento.

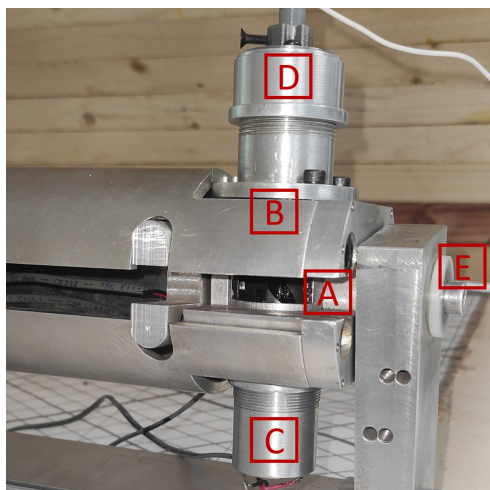


Figura 2.8: Apertura de Escaneo del TVS2

2.6.2. Control y funcionamiento del TVS2

El modo de funcionamiento del TVS2 consiste en la proyección de luz coherente mediante el posicionador láser del sistema, mientras que en la apertura de escaneo el motor DC gira a una velocidad constante el espejo de 45° el cual recibe y redirige la luz de láser reflejada del objeto hacia los lentes de enfoque y fotosensor. El movimiento de los motores a pasos integrados en el posicionador láser y en el cuerpo del sistema se realiza mediante el envío de pulsos para crear pasos desde el microcontrolador mediante un circuito de transistores para aumentar la corriente que llega a los motores. Cada motor tiene un total de 200 pasos que son incrementados en un factor de 96 por el conjunto de engrane-tornillo sinfín, haciendo que cada revolución completa de los motores sea de un total de 19200 pasos. Figura 2.9 muestra un ejemplo de la configuración de Full-Step o paso completo que se utiliza para mover los motores.

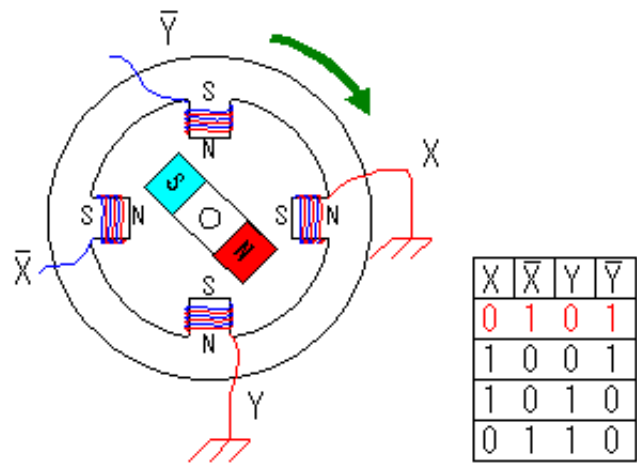


Figura 2.9: Ejemplo de movimiento de motor a pasos.

El voltaje en el fotosensor es entonces leído por un microcontrolador para determinar el pico de la señal de cada rotación. El motor DC de la apertura de escaneo cuenta con un sensor de presencia que marca el termino de cada revolución, el cual es leído por el microcontrolador. Una vez marcado el inicio de una nueva revolución del motor de la apertura de escaneo el microcontrolador inicia un contador de tiempo el cual marca la duración de cada vuelta, además del tiempo tardado desde el inicio de la revolución hasta la detección del pico de la señal de la apertura de escaneo. Al final de cada vuelta el microcontrolador envía ambos datos, tiempo de revolución y tiempo de pico de señal a un equipo de computo. El equipo de computo entonces triangula la posición de cada punto escaneado.

2.7. TVS3

El prototipo de TVS actual, o TVS3, por otro lado, utiliza un campo de visión continuo al recurrir al reemplazo de los motores a pasos usados en el prototipo 2 del TVS por motores de corriente directa, siendo usados como servomotores mediante la adición de codificadores rotativos. Al hacer esto se ha aumentado considerablemente la resolución del sistema, obteniendo un campo de visión continuo, como se muestra en Figura 2.10. Además, en una investigación anterior se ha desarrollado un sistema

de control para estos servomotores para el posicionamiento exacto de estos motores, haciendo posible que, entre otras cosas, se abrieran posibilidades para un escaneo de objetos más preciso [63].

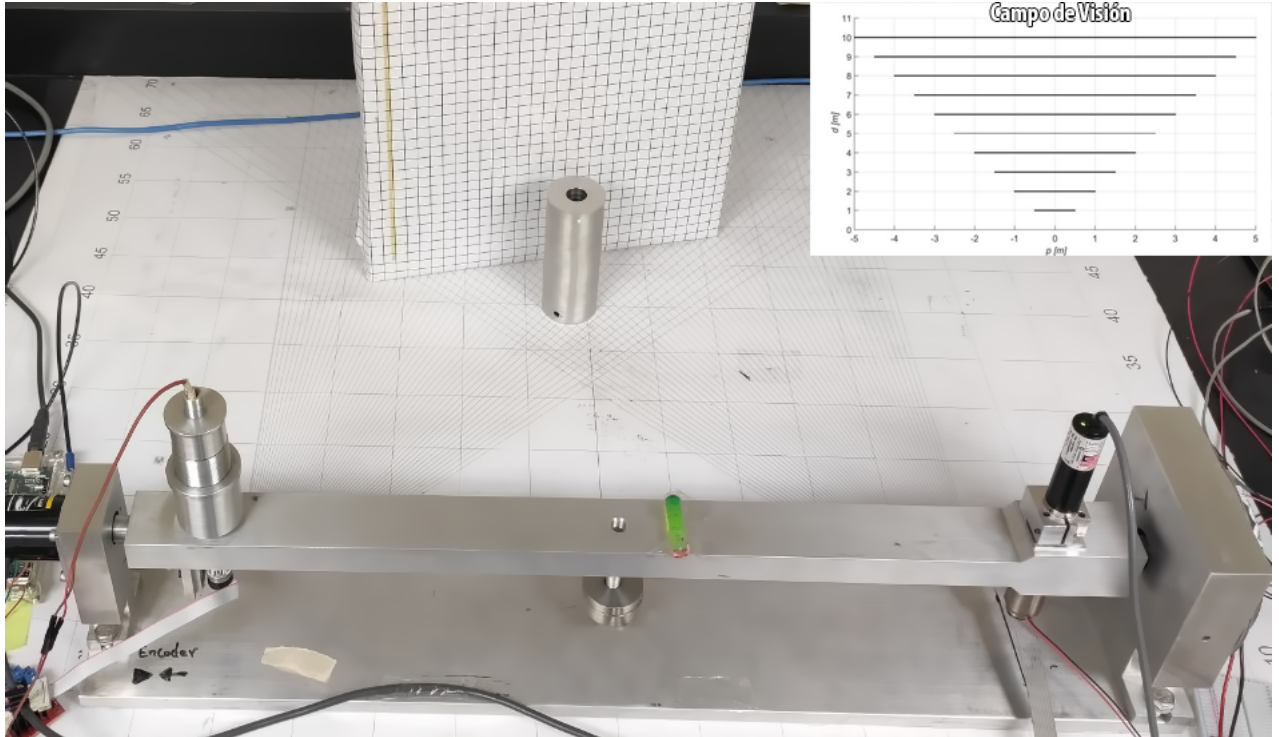


Figura 2.10: Campo de visión continuo de TVS3 [52].

2.7.1. Ruido y puntos atípicos

En el proceso de escaneo, diversos factores pueden introducir ruido y puntos atípicos en los datos capturados por este sistema. Estas irregularidades a menudo son causadas por problemas como la velocidad no constante del espejo de la apertura de escaneo, dificultades para detectar con precisión el centro de energía de la señal durante el procesamiento de datos, y el ángulo de inclinación del sistema. Estos elementos pueden complicar la precisión de los datos, dificultando mantener la integridad de los resultados. El ruido y los puntos atípicos también pueden oscurecer detalles importantes en las nubes de puntos, lo que requiere filtrar para mejorar los resultados. Aunque los errores de triangulación son más comunes a lo largo del eje X, los puntos atípicos también pueden aparecer en otros ejes, lo que dificulta mantener la claridad en las nubes de

puntos. Abordar estos problemas es crucial para refinar los datos y garantizar que las nubes de puntos producidas sean lo más precisas y representativas del objeto escaneado.

Los puntos atípicos se refieren a puntos de datos que difieren significativamente en valor del resto del conjunto de datos, siendo mucho más altos o más bajos que los datos circundantes. En el contexto de las nubes de puntos, estos puntos atípicos pueden impactar negativamente tanto en la visualización de las nubes de puntos como en cualquier aplicación posterior de los datos. Su presencia puede distorsionar la representación general del objeto, lo que lleva a inexactitudes en el análisis o desafíos en la utilización de los datos para modelado u otros propósitos.

2.7.2. Procesamiento de datos y eliminación de puntos atípicos

Para la mayoría de los datos atípicos obtenidos, se aplican filtros estadísticos en diferentes ejes de las nubes de puntos obtenidas, dependiendo de varios factores; en particular, se mencionan 3 métodos para este propósito:

2.7.2.1. Método de intercuartiles

Este método [64] se aplica a datos univariantes, que pueden segmentarse en cuartiles. El primer cuartil significa que el 25 % de los puntos de datos caen por debajo de este valor, el segundo cuartil representa la mediana del conjunto de datos, y el tercer cuartil muestra que el 25 % de los puntos de datos exceden ese valor. El método de intercuartiles se emplea para determinar el rango de datos que debe mantenerse, dado que:

$$[Q_1 - K(Q_3 - Q_1), Q_3 + K(Q_3 - Q_1)] \quad (2.4)$$

Donde K es una constante dada y tiene un valor no negativo, que puede usarse para controlar la cantidad de datos que se detectan como atípicos, mientras que Q_1 y Q_3 corresponden a los cuartiles utilizados como umbrales superior e inferior para los datos. Los valores fuera del umbral definido por la Ec. (2.4) se consideran puntos atípicos.

2.7.2.2. Método Tau-Thompson modificado

El método Tau-Thompson modificado [65] es una estrategia alternativa para identificar puntos atípicos dentro de un conjunto de datos. Aquí el enfoque requiere tratar con un conjunto de datos univariable, evaluando posibles puntos atípicos uno a la vez y eliminando aquellos que se encuentren más allá de dos desviaciones estándar de la media del conjunto de datos.

Para aplicar esta técnica, primero se debe determinar la desviación media absoluta δ_i , seguida por el cálculo del umbral Tau τ :

$$\tau = \frac{t_{\alpha/2}(n-1)}{\sqrt{n}\sqrt{n-2+(t_{\alpha/2})^2}} \quad (2.5)$$

Donde $t_{\alpha/2}$ es la función de distribución acumulativa inversa y n es el tamaño del conjunto de datos.

De esta manera, se determina que un valor es un punto atípico si se encuentra que $\delta_i > \tau * S$, donde S es la desviación estándar.

2.7.2.3. Cuantiles de distribución Chi²

En este método [66], para cada muestra de dimensión d se determina la distancia de Mahalanobis dado que:

$$Di = (X_i - \bar{X}) * \sum^{-1} * (X_i - \bar{X})' \quad (2.6)$$

Donde todos los valores de Di que exceden el valor crítico calculado son detectados.

2.7.2.4. Suavizado de la nube de puntos

Después del procesamiento de puntos atípicos, se utiliza el algoritmo de Mínimos Cuadrados Móviles (MLS) [67]. Esta técnica se basa en la regresión lineal ponderada. El método MLS es particularmente eficaz para reconstruir superficies a partir de conjuntos de puntos, a menudo utilizado para crear superficies 3D a partir de nubes de puntos mediante el muestreo descendente o ascendente. Su utilidad es notable en el manejo

de datos que están espaciados irregularmente o contienen ruido, ya que proporciona un enfoque de ajuste flexible que se adapta a la estructura local de los datos.

$$S = \sum_{i=1}^n w_i \cdot (y_i - f(x_i))^2 \quad (2.7)$$

Donde: S es la suma que debe minimizarse, n es el número de puntos de datos, w_i representa el peso para el i -ésimo punto de datos, a menudo una función de la distancia para enfatizar puntos más cercanos al punto de interés. y_i y x_i son las coordenadas del i -ésimo punto de datos y $f(x_i)$ es el valor de la función de aproximación en x_i .

2.7.3. Registro de nube de puntos

Utilizado en visión por computadora, robótica y reconstrucción 3D, el registro de nube de puntos es el proceso de alinear múltiples nubes de puntos, cada una capturada desde diferentes posiciones dentro del mismo entorno, en un marco de coordenadas unificado. Este proceso resulta en un conjunto de datos consolidado que representa con precisión la escena que se está estudiando [68]. Para este experimento, se utilizaron 2 algoritmos de registro de nubes de puntos similares, el Algoritmo de Punto de Correspondencia Iterativa (ICP) para el registro local y el Algoritmo de Consenso de Muestra Aleatoria (RANSAC) para el registro global a través de la biblioteca Open3D [69].

2.7.3.1. Registro de Punto de Correspondencia Iterativa (ICP)

Este proceso implica alinear dos o más nubes de puntos, identificando una transformación espacial, que podría implicar escalado, rotación o traslación, y reduce las disparidades entre ellas. El objetivo final es combinar estos múltiples conjuntos de datos en un modelo o marco de coordenadas cohesivo y globalmente consistente. El proceso comienza con dos nubes de puntos y una transformación inicial que posiciona aproximadamente la nube de puntos fuente en alineación con la nube de puntos objetivo. El resultado es una transformación que alinea estrecha y precisamente las dos nubes de puntos.

$$E(R, t) = \sum_{i=1}^N \|(R \cdot p_i + t) - q_i\|^2 \quad (2.8)$$

Donde: R es la matriz de rotación, t es el vector de traslación, p_i son los puntos del conjunto de datos fuente, q_i son los puntos correspondientes más cercanos en el conjunto de datos objetivo, N es el número de pares de puntos.

2.7.3.2. Registro de Consenso de Muestra Aleatoria (RANSAC)

Otra categoría de técnicas de registro, conocida como registro global, incluye algoritmos que no requieren una alineación inicial para comenzar. Por lo general, este método utiliza RANSAC [70] para la alineación inicial de las dos nubes de puntos, y luego se utiliza ICP para la comparación.

2.7.4. Verdad Terreno

Para la comparación con las dimensiones originales de los objetos en el experimento, se utilizó una cámara Intel RealSense para medir las proporciones de los objetos. Esta cámara integra un sistema avanzado de visión estereoscópica, capturando información de profundidad de los objetos en su campo de visión además de las imágenes 2D tradicionales.

2.7.5. Reconstrucción

La reconstrucción utiliza el algoritmo Poisson tamizado [71], que se usa para crear una superficie continua suave que representa el objeto o escena escaneada a partir de una nube de puntos. Se basa en la ecuación de Poisson, donde en un sistema de coordenadas cartesianas tridimensional, toma la forma:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \varphi(x, y, z) = f(x, y, z) \quad (2.9)$$

Donde, φ y f son funciones reales o complejas.

2.8. Inteligencia Artificial y Redes Neuronales

La **Inteligencia Artificial (IA)** es un campo de la computación que busca dotar a las máquinas de la capacidad de realizar tareas que tradicionalmente requieren inteligencia humana. Esto incluye el razonamiento, la toma de decisiones y el aprendizaje a partir de datos o de la experiencia. En el contexto de los sistemas de medición y visión, la IA puede contribuir al procesado de información, la identificación de patrones y la adaptación de algoritmos para mejorar la precisión y robustez de las mediciones.

Dentro de la IA, las **redes neuronales** son modelos inspirados en la estructura del cerebro biológico, compuestas por nodos o “neuronas” organizadas en capas. Cada neurona recibe entradas ponderadas, las procesa y genera una salida que puede servir de entrada a otras neuronas. Mediante técnicas de entrenamiento, las redes neuronales ajustan los pesos asociados a cada conexión para minimizar un error definido sobre un conjunto de datos. Así, las redes neuronales son capaces de aprender representaciones complejas y de realizar tareas como clasificación, regresión o predicción de manera eficiente.

2.8.1. Reconocimiento de nubes 3D

Las nubes de puntos en tres dimensiones ($3D$) son un conjunto de coordenadas espaciales que describen la geometría superficial de uno o varios objetos. La IA puede desempeñar un papel importante en el procesamiento de las nubes de puntos, al permitir:

- **Análisis de la forma:** comprensión de la estructura y morfología de los objetos escaneados.
- **Identificación de patrones:** detección de características o regiones de interés en la nube de puntos.
- **Aprendizaje a partir de datos:** ajustar modelos para describir con mayor fidelidad la realidad observada.

Las redes neuronales enfocadas en el reconocimiento de nubes 3D suelen emplear métodos para tratar la información tridimensional de manera directa o indirecta, facilitando así la clasificación de objetos, la estimación de pose o el seguimiento de elementos dentro de una escena.

2.8.2. Segmentación

La **segmentación** consiste en dividir una nube de puntos (o una imagen 2D, en otros contextos) en regiones coherentes que comparten ciertas características, como el color, la forma o la ubicación espacial. En el caso de nubes 3D, la segmentación puede distinguir objetos, partes de un objeto o zonas de interés dentro de un entorno mayor.

- **Segmentación basada en umbrales:** se fundamenta en separar datos dependiendo de una o varias propiedades medibles (distancia, intensidad, etc.).
- **Segmentación basada en agrupamiento (clustering):** agrupa puntos que presentan similitudes según algún criterio de similitud.
- **Segmentación apoyada en redes neuronales:** aprovecha modelos de aprendizaje profundo para clasificar cada punto según su pertenencia a distintas categorías o regiones.

La elección del enfoque depende de la complejidad de la escena, la densidad de la nube de puntos y los recursos de cómputo disponibles.

2.8.3. Clasificadores

Los **clasificadores** son algoritmos que asignan una categoría o etiqueta a cada dato de una nube de puntos u otros datos que contengan patrones reconocibles, basándose en sus características o descriptores. En inteligencia artificial, los clasificadores pueden implementarse mediante modelos supervisados o no supervisados:

- **Modelos supervisados:** requieren ejemplos etiquetados para aprender a distinguir entre distintas clases. Un ejemplo clásico son las redes neuronales entrenadas con conjuntos de datos anotados.

- **Modelos no supervisados:** agrupan los datos sin etiquetas previas, guiándose por la similitud intrínseca de las características. Este enfoque se conoce como *clustering* y se utiliza cuando no se dispone de información previa acerca de las clases.

En el reconocimiento de nubes 3D e imágenes 2D los clasificadores permiten, por ejemplo, distinguir un objeto de fondo o separar varios objetos entre sí. Asimismo, pueden facilitar la detección de anomalías o la identificación de zonas de interés en aplicaciones de inspección y control de calidad. El desempeño de estos clasificadores depende de la calidad de los datos de entrada, la representatividad de los ejemplos utilizados en el entrenamiento y la robustez de los métodos de aprendizaje empleados.

Metodología

Esta sección describe el diseño de experimentos para la obtención de modelos tridimensionales con un sistema TVS2, realizando escaneos 3D de las diferentes caras de objetos. Se compara con los resultados obtenidos al utilizar una cámara Intel RealSense, utilizando un algoritmo de registro de nubes de puntos. Además, se realiza un experimento para el reconocimiento del color de las coordenadas de los objetos escaneados.

3.1. Diseño del experimento para la obtención de modelos tridimensionales

El experimento principal consistió en llevar a cabo escaneos tridimensionales para evaluar el desempeño del Sistema de Visión Técnica (TVS) bajo diversas resoluciones de escaneo y configuraciones de ventana de escaneo. El propósito fue comparar la eficiencia de los escaneos generados por el TVS 2 en múltiples caras de un objeto, en diversas configuraciones de escaneo como referencia para evaluar las mediciones del TVS utilizando una cámara Intel RealSense. Esta comparación se enfocó en determinar la capacidad del TVS 2 para conservar el detalle y la precisión en comparación con los escaneos obtenidos mediante la cámara Intel RealSense, así como en hallar una forma de reducir el tiempo y las ventanas de escaneo al emplear el sistema TVS, para posteriormente utilizar un algoritmo de registro de nubes de puntos en el análisis.

Los factores considerados durante los experimentos incluyeron las condiciones de iluminación del laboratorio, la regulación de la velocidad del espejo de la apertura de

escaneo (SA) para verificar la consistencia y mitigar el ruido, y la disposición espacial del TVS 2. La Tabla 3.1 presenta un resumen de los factores de experimentación y sus niveles:

Tabla 3.1: Factores de experimentación

Factor	Nivel / Rango	Δ
Luz de Laboratorio	APAGADA (< 1 Lux)	-
Voltaje del Fotorresistor	3.3V	-
Resistencia del Amplificador	830K Ω	-
Velocidad de Rotación de la SA	300rpm	-
Distancia de Escaneo de Profundidad x	55cm - 85cm	35cm

El experimento se llevó a cabo mediante una base giratoria (para orientar el objeto frente al TVS) controlada por un motor paso a paso y un mecanismo reductor que permite un manejo preciso. Los objetos seleccionados incluyeron figuras con formas de letras y números, además de una calavera de espuma de poliestireno. Para cada objeto se definieron el número de caras a escanear, la resolución del escaneo (determinada por la cantidad de puntos a obtener) y la cantidad de lecturas del sensor de luz por cada punto medido. El objeto, colocado sobre la base giratoria, se ubicó en dos posiciones diferentes frente al TVS. Estas configuraciones pueden observarse en la Tabla 3.2.

Tabla 3.2: Configuración del experimento para los objetos de poliestireno escaneados

Objeto	Ángulo del paso de la mesa giratoria	Caras obtenidas	Posición de la mesa giratoria en TVS (x, y, z) en cm	Ventana de escaneo yxz
Letra A	90.0°	4	POS1(33,30,10)	13°x15°
Número 6	90.0°	4	POS1(33,30,10)	13°x15°
Calavera	90.0°	4	POS1(33,30,10) y, POS2(21,23,10)	13°x15°
Calavera	45.0°	8	POS1(33,30,10) y, POS2(21,23,10)	13°x15°
Calavera	22.5°	16	POS1(33,30,10) y, POS2(21,23,10)	13°x15°

El sistema TVS escanea cada cara predefinida del objeto y posiciona el láser en puntos específicos definidos como parte de las configuraciones de ventana y resolución de escaneo. Registra los datos de posición de cada componente del TVS para su compilación posterior, junto con las lecturas de voltaje del fotosensor del sistema, que emplea una disposición básica de colector común con una resistencia. Una vez que se completa la recolección de datos de la ventana de escaneo, la base giratoria se mueve un número predeterminado de grados, generando un archivo para cada punto definido dentro de la ventana de escaneo.

Después del proceso de escaneo, los datos se procesan en una computadora personal. Utilizando las Ecuaciones. (2.1), (2.2), y (2.3), se generan nubes de puntos para cada cara de los objetos seleccionados. Debido al diseño del TVS, estas nubes de puntos tienden a incluir ruido y puntos atípicos. Por lo tanto, se emplean varios filtros estadísticos para minimizar estas anomalías de datos. Un ejemplo de esta reducción se ilustra en la Tabla 3.3, que muestra la reducción de la desviación estándar en el eje x para una cara de un objeto de poliestireno en forma de la letra “A”. Los niveles de ruido y las características de los puntos atípicos observados para la letra “A” fueron representativos de todos los demás objetos de espuma examinados en este estudio. Los datos se suavizan mediante el método de Mínimos Cuadrados Móviles (Moving Least Squares - MLS), lo que resulta en una representación más precisa y suave de la superficie del objeto.

Tabla 3.3: Desviación estándar de la cara de la letra A de espuma

Método/Conjunto de Datos	Cantidad de puntos 3D	Desviación estándar
Original	18462	1.1100
Intercuartil	9232	0.2350
Tau-Thompson $\alpha=0.01$	17239	0.5475
Tau-Thompson $\alpha=0.1$	14026	0.3751
Chi ² 97.5 %	18116	0.6900
Chi ² 90 %	17916	0.64200

Luego de la adquisición de datos para cada cara, las nubes de puntos de las caras capturadas se rotan, utilizando el centro de la mesa giratoria como el nuevo centro de la nube de puntos. Esto implica definir dos puntos, P y O , que representan coordenadas espaciales en un espacio tridimensional de la siguiente manera: $P = [P_1P_2P_3]$ y $O = [O_1O_2O_3]$. Donde P representa cada punto en el espacio del conjunto de datos de la nube de puntos 1 y O representa cada punto en el espacio del conjunto de datos de la nube de puntos 2.

El vector de distancia entre estos dos puntos se puede calcular de la siguiente manera: $\vec{PO} = O - P$. Se calcula la magnitud de ambos vectores de distancia, y se determina el vector a traducir. Posteriormente, se utiliza una matriz de transformación para realizar la rotación de la nube de puntos:

$$P' = \begin{bmatrix} P'_1 \\ P'_2 \\ P'_2 \\ 1 \end{bmatrix} = T_\alpha P = \begin{bmatrix} 1 & 0 & 0 & V_\alpha \\ 0 & 1 & 0 & V_\alpha \\ 0 & 0 & 1 & V_\alpha \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P'_1 \\ P'_2 \\ P'_2 \\ 1 \end{bmatrix} = P + V_\alpha \quad (3.1)$$

Los datos luego se rotan utilizando matrices de rotación simples alrededor del eje Z.

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} X_f \\ Y_f \\ Z_f \\ 1 \end{bmatrix} \quad (3.2)$$

Las mediciones de entropía se utilizan para proporcionar una evaluación cuantitativa del desorden o la aleatoriedad dentro de los datos de la nube de puntos generados por el Sistema de Visión Técnica (TVS). La entropía, un concepto proveniente de la teoría de la información, ofrece una métrica sencilla para evaluar la variabilidad en los datos, lo que puede ser indicativo de niveles de ruido o inconsistencias en el escaneo. Las medidas de entropía aproximada y entropía de Shannon pueden verse en la Tabla 3.4 y la Tabla 3.5.

Tabla 3.4: Entropía aproximada calculada para cada modelo tridimensional.

Objeto	Posición	Paso angular de la mesa giratoria	Caras obtenidas	Entropía aproximada antes del filtrado
Letra A	Pos1	90.0°	4	0.384
Número 6	Pos1	90.0°	4	0.449
Calavera	Pos1	45.0°	8	0.434
Calavera	Pos1	90.0°	4	0.357
Calavera	Pos1	22.5°	16	1.155
Calavera	Pos2	45.0°	8	0.591
Calavera	Pos2	90.0°	4	0.688
Calavera	Pos2	22.5°	16	1.391

Tabla 3.5: Entropía de Shannon calculada para cada modelo tridimensional.

Objeto	Posición	Paso angular de la mesa giratoria	Caras obtenidas	Entropía de Shannon antes del filtrado
Letra A	Pos1	90.0°	4	3.515
Número 6	Pos1	90.0°	4	3.007
Calavera	Pos1	45.0°	8	4.365
Calavera	Pos1	90.0°	4	3.850
Calavera	Pos1	22.5°	16	4.835
Calavera	Pos2	45.0°	8	4.871
Calavera	Pos2	90.0°	4	3.576
Calavera	Pos2	22.5°	16	3.777

Después de la rotación de los puntos, se genera una nube “completa” y se crea un

sólido 3D mediante el algoritmo Poisson tamizado. Esta reproducción está lista para ser editada o impresa en 3D (Figura 3.1).

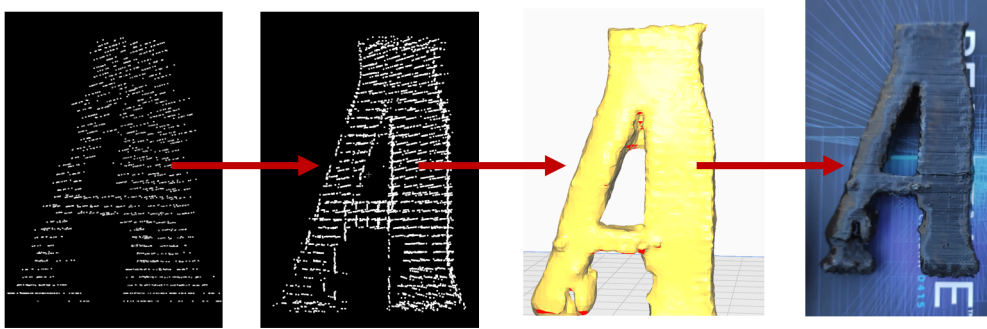


Figura 3.1: De la nube de puntos a la impresión 3D

3.2. Diseño de experimento para reconocimiento de colores

Esta subsección describe la metodología experimental aplicada, en la cual se han empleado los materiales y métodos previamente detallados. Se expone el proceso seguido para implementar una red de clasificación de color utilizando la información obtenida del sistema TVS2 como un problema de investigación secundario.

3.2.1. Extracción de características

En una primera etapa, las mediciones se realizaron escaneando un objeto cubierto con papeles de distintos colores a diversas profundidades en el eje x a lo largo del Campo de visión, en tres planos z (β_j). La configuración de medición se muestra en la Figura 3.2. Se eligió una barra como objeto de escaneo, la cual se recubrió con diversos tipos de papel (rojo brillante, rojo mate, azul brillante, azul mate, amarillo brillante y amarillo mate). El objeto se posicionó a 70 cm, 80 cm y 90 cm del TVS (profundidad en el eje x). Los tres planos z se escanearon ajustando el ángulo β_j a -2.5° , 0° y 2.5° . El posicionador láser (LP) recorrió el Campo de visión, iniciando en $C_{ij} = 45^\circ$ e incrementando 0.99375° en cada posicionamiento hasta alcanzar $C_{ij} = 80^\circ$. En cada posición del láser, la apertura de escaneo (AE) rotó 13 veces, generando 13 señales

optoelectrónicas que permitieron calcular 13 veces la coordenada 3D correspondiente al punto láser observado por la AE en ese posicionamiento, como se ilustra en la Figura 3.7.

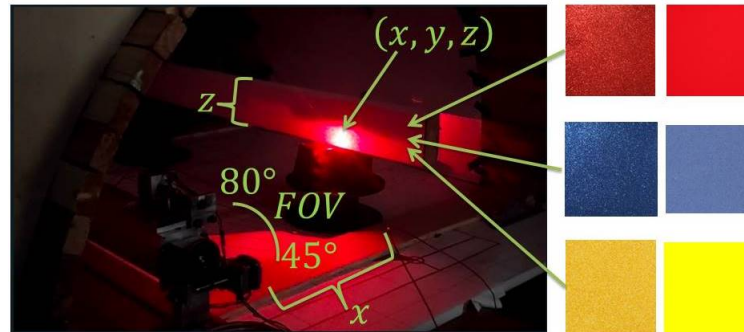


Figura 3.2: Configuración de medición. [72]

Las señales optoelectrónicas generadas por el TVS2 corresponden a la adquisición de un voltaje en función del tiempo, derivado de la conversión de señales ópticas a señales eléctricas mediante la recepción del láser reflejado por el objeto a través del sensor optoelectrónico de la AE. La magnitud del voltaje varía con el tiempo: durante la rotación del espejo de 45° en el escaneo del campo de visión se distinguen tres etapas: *i*) una superposición parcial y progresiva entre el plano reflejante del espejo y el láser reflejado, en la que la magnitud del voltaje aumenta; *ii*) el instante en el que el plano reflejante y el láser reflejado se superponen completamente, alcanzando el nivel máximo de voltaje; y *iii*) una disminución parcial y progresiva de la superposición, en la que la magnitud del voltaje desciende hasta que ambos dejan de coincidir, como se ilustra en la Figura 3.3 [73].

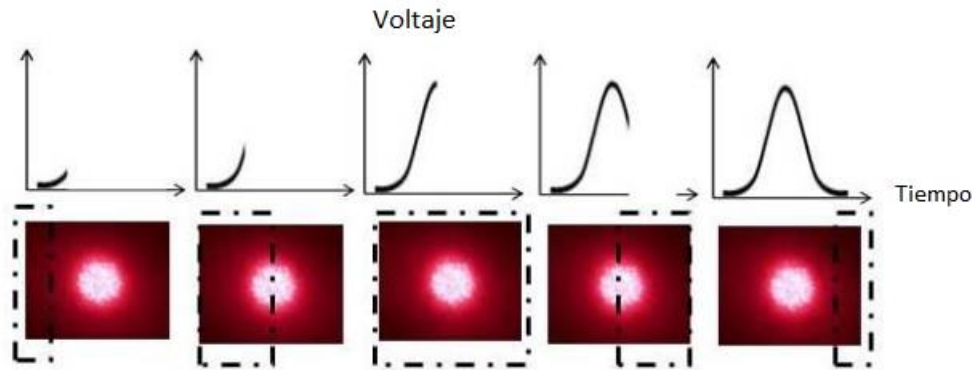


Figura 3.3: Generación de la señal optoelectrónica durante el escaneo del punto láser. [72]

De estas señales optoelectrónicas se debería de obtener una forma gaussiana, pero debido a las características del sistema, las señales obtenidas presentan solo una aproximación a estas características, exhibiendo formas adicionales como picos y desplazamientos. Estas variaciones se deben a los efectos inherentes a los sistemas ópticos de escaneo TVS y a la manera en la que el centro energético de la señal se detecta, así como a las propiedades de la superficie del objeto, incluyendo su forma y color [74].

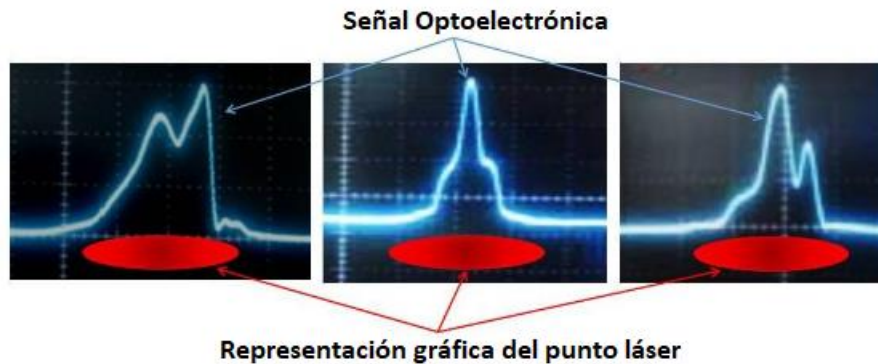


Figura 3.4: Señal optoelectrónica del escaneo de un punto láser mediante TVS. [72]

Las señales obtenidas incorporan implícitamente la estructura de la luz láser reflejada por las superficies del objeto escaneado. Las variaciones en la forma gaussiana de dicha señal se deben a las interacciones entre la iluminación láser y la orientación local de la superficie, así como a los fenómenos de refracción y reflexión. La refracción y la reflexión son procesos ópticos que ocurren cuando los rayos de luz láser inciden sobre

una superficie y cambian de dirección. En particular, la refracción se produce cuando el rayo láser incidente transita de un medio a otro, modificando su ángulo de propagación. La reflexión puede ser especular, cuando los rayos láser inciden de manera paralela y se reflejan con ángulos iguales, o difusa, en el caso de superficies rugosas, donde los rayos se reflejan en ángulos variados.

La transmisión de la superficie y las propiedades de absorción del objeto inciden significativamente en las señales optoelectrónicas, especialmente en lo que se refiere a la absorción selectiva, en la cual la superficie absorbe rangos específicos del espectro electromagnético y refleja el resto de las longitudes de onda. La reflectancia en función de la longitud de onda de los materiales cromáticos proporciona información valiosa, ya que cada longitud de onda posee un albedo asociado (la proporción de la luz láser incidente que se refleja en la superficie). De esta manera, mediante el análisis de la longitud de onda emitida, es posible estimar las características de superficies con colores desconocidos [75].

Los parámetros característicos extraídos en el dominio del tiempo de las señales eléctricas pueden representar el fenómeno físico que origina dichas señales. La interpretación de las señales en este dominio se ha aplicado a diversos tipos de señales, tales como las biométricas [76], las de comunicación [77] y las optoelectrónicas [78]. Aunque las señales también pueden analizarse en el dominio de la frecuencia, este tipo de procesamiento se centra en la extracción de las funciones energéticas mediante la Densidad Espectral de Potencia (PSD, por sus siglas en inglés) [79], sin incorporar la distribución espacial.

La Figura 3.5 muestra 14 señales optoelectrónicas en el dominio del tiempo. Estas mediciones se realizaron posicionando el LP en la misma coordenada espacial 3D, lo que equivale a 14 rotaciones de la AE para escanear los 360° en busca del láser reflejado. A diferencia de las señales presentadas en la Figura 3.4, en la Figura 3.5 se incluye un impulso en cada inicio de revolución para indicar la referencia a 0° . Asimismo, se observa que las señales aparecen rotadas en 180° en comparación con las de la Figura 3.4, debido al voltaje de referencia empleado en la adquisición de la señal a procesar.

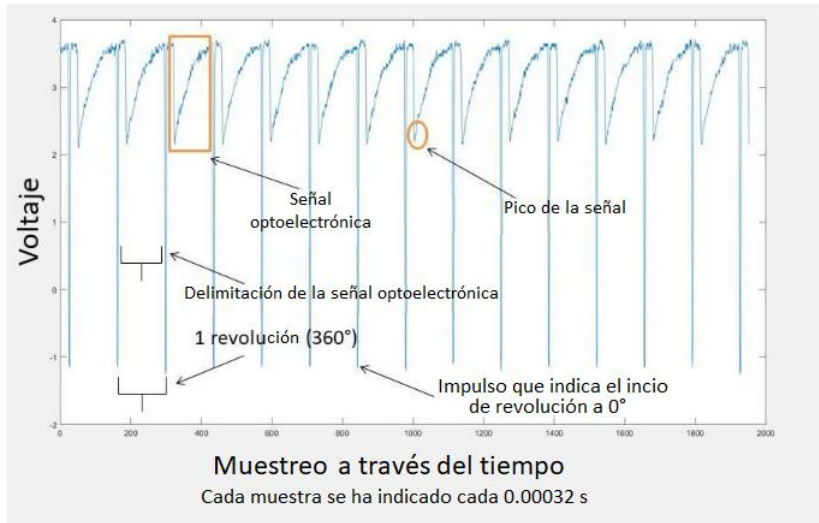


Figura 3.5: Muestreo de las señales optoelectrónicas y de referencia. [72]

La Figura 3.6 ilustra los límites de una señal gaussiana abarcando los 360° de escaneo, junto con una ventana que comprende 180°, proporcionando una vista más representativa de la AE.

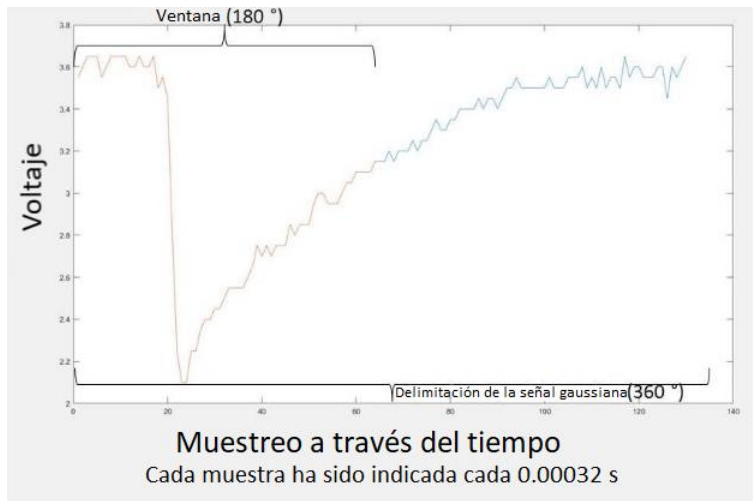


Figura 3.6: Delimitación de la señal Gaussiana (360°) y ventana de la señal Gaussiana (180°). [72]

Parámetros específicos, estrechamente relacionados con el principio de funcionamiento del TVS, así como parámetros estadísticos reconocidos para estimar las distribuciones de los parámetros físicos subyacentes en las mediciones observadas [80], fueron extraídos según se detalla en la Tabla 3.6.

Tabla 3.6: Parámetros Extraídos en el Dominio del Tiempo

No.	Parámetro	Descripción	Ecuación
1	C_{ij}	Ángulo del posicionamiento del láser.	No aplica
2	B_{ij}	Ángulo del posicionamiento de la apertura.	Ec. 3.16
3	x_{ij}	Dimensión x de la coordenada espacial.	Ec. 3.17
4	y_{ij}	Dimensión y de la coordenada espacial.	Ec. 3.18
5	z_{ij}	Dimensión z de la coordenada espacial.	Ec. 3.19
6	RS	Tamaño de la revolución (Cantidad de muestras en 360°).	No aplica
7	$OSBS$	Cantidad de muestras en la delimitación de la señal optoelectrónica.	Ec. 3.3
8	RS_{avg}	Promedio de la cantidad de mediciones (RS) en una coordenada espacial.	Ec. 3.4
9	WS	Cantidad de muestras en la ventana de 180° .	Ec. 3.3
10	V_{peak}	Voltaje pico de la Gaussiana.	Ec. 3.5
11	$Pikewide$	Cantidad de muestras con el valor del voltaje pico de la Gaussiana.	No aplica
12	$Piketopike$	Voltaje entre el máximo y mínimo valor de la Gaussiana en los 360° .	Ec. 3.6
13	$Piketopike_w$	Voltaje entre el máximo y mínimo valor de la Gaussiana en la ventana de 180° .	Ec. 3.6
14	$Mean$	Promedio de las mediciones en los 360° .	Ec. 3.4
15	$Mean_{pike}$	Promedio de las mediciones en la ventana de 180° .	Ec. 3.4
16	$Median$	Mediana de las mediciones en los 360° .	Ec. 3.7
17	$Median_w$	Mediana de las mediciones en la ventana de 180° .	Ec. 3.7

Continúa en la siguiente página...

No.	Parámetro	Descripción	Ecuación
18	<i>Rms</i>	Media cuadrática de las mediciones en los 360°.	Ec. 3.8
19	<i>Rms_w</i>	Media cuadrática de las mediciones en la ventana de 180°.	Ec. 3.8
20	<i>Variance</i>	Varianza cuadrática de las mediciones en los 360°.	Ec. 3.9
21	<i>Variance_w</i>	Varianza de las mediciones en la ventana de 180°.	Ec. 3.9
22	<i>Sdeviation</i>	Desviación estándar cuadrática de las mediciones en los 360°.	Ec. 3.10
23	<i>Sdeviation_w</i>	Desviación estándar de las mediciones en la ventana de 180°.	Ec. 3.10
24	<i>MAdeviation</i>	Desviación media absoluta de las mediciones en los 360°.	Ec. 3.11
25	<i>MAdeviation_w</i>	Desviación media absoluta de las mediciones en la ventana de 180°.	Ec. 3.11
26	<i>Skewness</i>	Oblicuidad de las mediciones en los 360°.	Ec. 3.12
27	<i>Skewness_w</i>	Oblicuidad de las mediciones en la ventana de 180°.	Ec. 3.12
28	<i>Kurtosis</i>	Curtosis de las mediciones en los 360°.	Ec. 3.13
29	<i>Kurtosis_w</i>	Curtosis de las mediciones en la ventana de 180°.	Ec. 3.13
30	<i>RSSquare</i>	Raíz de la suma de los cuadrados de las mediciones en los 360°.	Ec. 3.14
31	<i>RSSquare_w</i>	Raíz de la suma de los cuadrados de las mediciones en la ventana de 180°.	Ec. 3.14

Los parámetros en la Tabla 3.6 del 1 al 6 han sido descritos en la sección de triangulación dinámica. Los parámetros 7 y 9 se definen mediante la Ec. 3.3, donde $V(s)$ representa el voltaje adquirido en cada muestra; los límites para el parámetro 7 corresponden a los impulsos entre dos revoluciones, mientras que los límites para el parámetro 9 se corresponden con los límites de la ventana. Para los parámetros 8, 14 y 15 se utiliza la ecuación de la media (Ec. 3.4), en la que T es la cantidad de datos a promediar y x_t es la dimensión considerada en el promedio. El parámetro 10 corresponde al valor máximo de $V(s)$, y el parámetro 11 a la cantidad de muestras que cumplen la condición del parámetro 10 (Ec. 3.5). Los parámetros 12 y 13 se calculan según la Ec. 3.6, aplicándose los límites definidos para cada uno. Asimismo, los parámetros 16 y 17 se obtienen mediante la Ec. 3.7, los parámetros 18 y 19 mediante la Ec. 3.8, y los parámetros 20 y 21 a través de la Ec. 3.9, donde \bar{x} representa la media de las muestras T . Los parámetros 22 y 23 se calculan con la Ec. 3.10, mientras que los parámetros 24 y 25 se determinan con la Ec. 3.11. Por último, los parámetros 26 y 27 se calculan con la Ec. 3.12, los parámetros 28 y 29 con la Ec. 3.13, y los parámetros 30 y 31 con la Ec. 3.14.

Los parámetros estadísticos estiman las distribuciones de los parámetros físicos subyacentes en el conjunto de datos observados. En particular, la media (Ec. 3.4) calcula el promedio de los valores de amplitud de la señal en las muestras; la mediana (Ec. 3.7) indica el valor de amplitud en el punto medio de la distribución de frecuencia de las muestras observadas; la media cuadrática (Ec. 3.8) representa, en términos eléctricos, el valor de una constante que produciría la misma disipación de potencia en una carga resistiva; la varianza (Ec. 3.9) y la desviación estándar (Ec. 3.10) estiman la dispersión de los valores de amplitud de las muestras; la desviación media absoluta (Ec. 3.11) es el promedio de las desviaciones de los valores de amplitud respecto a la media; la asimetría (Ec. 3.12) cuantifica la inclinación de la distribución de la señal; y la curtosis (Ec. 3.13) evalúa el pico de la distribución de probabilidad [79].

$$\textit{l\acute{i}mite izquierdo} < V(s) < \textit{l\acute{i}mite derecho} \quad (3.3)$$

$$\text{Mean} = \frac{1}{T} \sum_{t=1}^T x_t \quad (3.4)$$

$$\text{Peak} = \max V(s) \quad (3.5)$$

$$\text{Peaktpeak} = \max V(s) - \min V(s) \quad (3.6)$$

$$\text{Median} = \begin{cases} x \left[\frac{t+1}{2} \right] & \text{si } t \text{ es impar} \\ \frac{x \left[\frac{n}{2} \right] + x \left[\frac{n}{2} + 1 \right]}{2} & \text{si } t \text{ es par} \end{cases} \quad (3.7)$$

$$\text{Root mean square} = \sqrt{\frac{1}{T} \sum_{t=1}^T |x_t^2|} \quad (3.8)$$

$$\text{Variance} = \frac{1}{T-1} \sum_{t=1}^T |x_t - \bar{x}|^2 \quad (3.9)$$

$$\text{Standard deviation} = \sqrt{\frac{1}{T-1} \sum_{t=1}^T |x_t - \bar{x}|^2} \quad (3.10)$$

$$\text{Mean absolute deviation} = \frac{1}{T} \sum_{t=1}^T |x_t - \bar{x}| \quad (3.11)$$

$$\text{Skewness} = \frac{\frac{1}{T} \sum_{t=1}^T (x_t - \bar{x})^3}{\left(\sqrt{\frac{1}{T} \sum_{t=1}^T (x_t - \bar{x})^2} \right)^3} \quad (3.12)$$

$$\text{Kurtosis} = \frac{\frac{1}{T} \sum_{t=1}^T (x_t - \bar{x})^4}{\left(\frac{1}{T} \sum_{t=1}^T (x_t - \bar{x})^2 \right)^2} \quad (3.13)$$

$$\text{Root sum of square} = \sqrt{\sum_{t=1}^T x_t^2} \quad (3.14)$$

3.2.2. Clasificación

El aprendizaje supervisado es un enfoque para crear inteligencia artificial (IA), donde un algoritmo informático se entrena en datos de entrada que se han etiquetado por una salida en particular, aportando un modelo que puede utilizarse con nuevos datos para predecir su etiqueta correspondiente [81]. Los clasificadores considerados en este trabajo son los árboles de decisión, análisis discriminante, clasificador Naive Bayes, máquinas de vector soporte, clasificador del vecino más cercano, ensembles y redes neuronales. Los árboles de decisión pueden clasificarse en finos, medianos y gruesos, depende de la cantidad de ramas añadidas para predecir una respuesta, las decisiones en el árbol se siguen desde el nodo de la raíz hasta el nodo de la hoja. El análisis discriminante puede clasificarse en lineal y cuadrático, clasifica al crear límites entre clases (o etiquetas), una función de ajuste estima los parámetros de una distribución gaussiana para cada clase. El clasificador Naive Bayes comúnmente se utiliza para proyectos multiclase, puede clasificarse en Gaussiano y Kernel, y se basa en el Teorema de Bayes, hace la suposición que los predictores son condicionalmente independientes para cada clase. Las máquinas de vector soporte, también usadas para dos proyectos multiclase, pueden clasificarse en lineal, cuadrática, cúbica, gaussiano fino, gaussiano medio y gaussiano grueso para hacer separaciones (hiperplanos) y crear vectores de soporte con los puntos de datos más cercanos al hiperplano de separación para distinguir entre clases. Los vecinos más cercanos pueden clasificarse en fino, medio, grueso, coseno, cúbico y ponderado. Cada uno utiliza tipos distintos de distancias entre vecinos, y una cantidad distinta de vecinos se utiliza para el cálculo de la definición de clases. El algoritmo de ensembles fusiona los resultados de muchos modelos aprendices en un solo modelo de ensemble de alta calidad. Puede clasificarse en árboles potenciados (basados en AdaBoost con aprendices de árbol de decisión), árboles empaquetados (basados en un empaquetado de bosque aleatorio con aprendices de árbol de decisión), subespacio discriminante (basado en el subespacio con aprendices discriminantes), subespacio KNN (basado en el subespacio con aprendices del vecino más cercano) y árboles RUSBoost (basados en RUSBoost con aprendices de árbol de decisión). Las redes neuronales, un

algoritmo basado en capas de neuronas que crean redes, pueden clasificarse en estrechas, medianas, amplias, de dos capas, de tres capas, con feedforward y de capas completamente conectadas. La primera capa completamente conectada de la red neuronal tiene un conexión de la entrada de red, y cada capa subsecuente tiene una conexión de la capa anterior. Cada capa completamente conectada multiplica la entrada mediante una matriz ponderada y añade un vector de sesgo. Una función de activación se añade después de cada capa completamente conectada. La última capa completamente conectada y la función subsecuente de activación softmax produce la salida de la red, específicamente las puntuaciones de clasificación y las etiquetas pronosticadas.

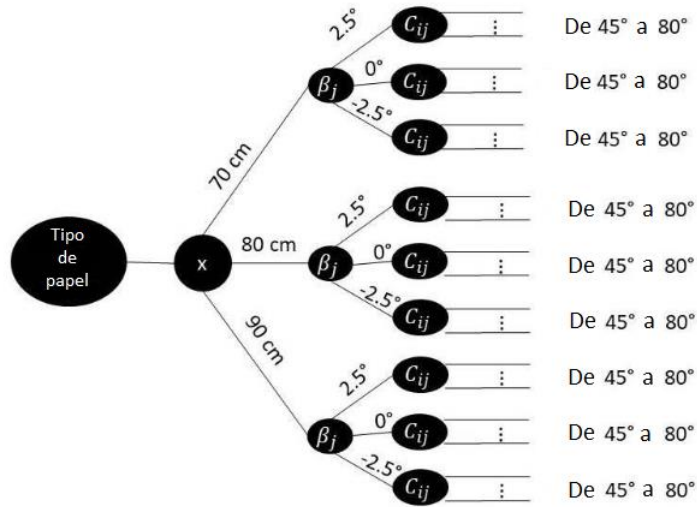


Figura 3.7: Diagrama de árbol de configuración del experimento. [72]

$$T_{2\pi} = \frac{2\pi}{\omega} = \frac{2\pi}{2\pi f} = \frac{1}{f} = 2\pi \quad (3.15)$$

$$B_{ij} = t_B 2\pi f \quad (3.16)$$

$$x_{ij} = a\left(\frac{\text{sen}(B_{ij})\text{sen}(C_{ij})}{\text{sen}(B_{ij} + C_{ij})}\right) \quad (3.17)$$

$$y_{ij} = a\left(\frac{1}{2} - \frac{\text{cos}(B_{ij})\text{sen}(C_{ij})}{\text{sen}(B_{ij} + C_{ij})}\right) \quad (3.18)$$

$$z_{ij} = a \frac{\text{sen}(B_{ij})\text{sen}(C_{ij})\tan(\beta)}{\text{sen}(B_{ij} + C_{ij})} \quad (3.19)$$

En una segunda etapa, se procesaron las señales adquiridas, primero para calcular la coordenada 3D (x, y, z) de acuerdo con las Ecuaciones 3.17 a 3.19, y posteriormente para extraer los parámetros en el dominio del tiempo, tal como se describe en la Tabla 3.6. Debido al comportamiento no lineal del TVS a lo largo del Campo de visión, como se ilustra en la figura de segmentación del Campo de visión en [82], no todas las señales optoelectrónicas resultaron útiles para el cálculo de la coordenada 3D. Tras descartar aquellas señales que no permitieron obtener la coordenada 3D, así como las correspondientes a valores atípicos, se obtuvieron finalmente 4701 mediciones de coordenadas. Los resultados se muestran en el capítulo experimentación y resultados.

Experimentación y resultados

Durante el proceso experimental realizado en este trabajo, diversos factores pudieron afectar el desempeño del sistema TVS, introduciendo variabilidad en los resultados obtenidos. En este capítulo se describen detalladamente los procedimientos seguidos en la experimentación, así como los resultados alcanzados tras analizar la influencia de la posición espacial del objeto en el campo de visión del sistema, el tratamiento de los datos obtenidos desde el sistema y su mejora tras la aplicación de diferentes filtros estadísticos y la capacidad del mismo para reconocer colores bajo condiciones controladas.

La primera fase experimental se centró en evaluar cómo la ubicación del objeto dentro del campo de visión impacta en la calidad y precisión de los datos generados por el TVS. Debido a factores como la perspectiva, el ángulo de escaneo y limitaciones técnicas inherentes al sistema, se identificaron diferencias notables en los resultados según la posición específica del objeto escaneado. Estas discrepancias pueden llevar a dificultades para mantener la consistencia y precisión esperada en las mediciones.

Tras esta fase inicial, se realizó un análisis de los datos obtenidos, aplicando diversos métodos estadísticos y filtros con el objetivo de reducir la dispersión y mejorar la calidad general de la información adquirida. Entre las técnicas aplicadas están filtros explicados en la sección de Metodología, que permitieron reducir tanto la dispersión como la entropía de los conjuntos de datos que permitieron la obtención de resultados más claros y representativos del objeto estudiado. Asimismo, para validar estos resultados, se utilizaron herramientas adicionales como técnicas de registro de nubes de puntos, con el propósito de comparar los datos obtenidos por el TVS con escaneos de

referencia realizados utilizando una cámara Intel RealSense. Esta comparación permitió identificar de manera más objetiva las diferencias y similitudes entre ambos.

En una segunda fase, se implementaron y evaluaron algoritmos orientados al reconocimiento automatizado de colores, buscando determinar la precisión y robustez del sistema frente a diferentes condiciones ambientales, variaciones en la iluminación y posiciones espaciales del objeto. La presencia de variabilidad en estos factores puede introducir errores o incertidumbre en los resultados, afectando la fiabilidad del reconocimiento de colores.

A continuación, se detallan los experimentos realizados, explicando claramente la metodología utilizada, así como un análisis exhaustivo de los resultados obtenidos. Este análisis permitirá identificar aspectos clave relacionados con el rendimiento del sistema y proporcionará bases sólidas para posibles mejoras en aplicaciones futuras del mismo.

4.1. Resultados del experimento para la obtención de modelos tridimensionales

Después de la rotación y alineación de los puntos adquiridos en diferentes posiciones del escaneo, se obtiene una nube de puntos denominada “completa”. Esta nube integra las distintas vistas del objeto, proporcionando una representación tridimensional continua. Sin embargo, en esta etapa, los datos aún carecen de una superficie sólida bien definida, lo que limita su utilidad directa en aplicaciones prácticas como la edición CAD o la fabricación aditiva.

Para solventar esto, se procedió a aplicar el algoritmo de reconstrucción superficial conocido como Poisson (*Poisson Surface Reconstruction*). Este método es utilizado debido a su eficacia para generar superficies cerradas y suaves a partir de nubes de puntos dispersas o incompletas. El algoritmo Poisson construye implícitamente un campo escalar tridimensional basado en las normales de los puntos disponibles, definiendo una superficie mediante el cálculo de una función indicadora cuyos valores determinan el interior y el exterior del objeto escaneado. Posteriormente, una malla triangular se extrae de este campo escalar, resultando en un modelo sólido detallado y listo para procesos

subsecuentes.

La aplicación del algoritmo de Poisson permitió obtener una malla 3D representativa del objeto original, con una reducción significativa de irregularidades y puntos atípicos que pudieran afectar su integridad geométrica. Este sólido resultante es apto para tareas posteriores tales como edición en software CAD, simulaciones numéricas, análisis dimensional y especialmente impresión 3D, ya que posee continuidad superficial requeridas para estas aplicaciones.

Después de obtener una nube de puntos completa una malla que representa todas las caras del objeto estudiado, se utilizó el algoritmo ICP (*Iterative Closest Point*) para evaluar las similitudes y diferencias existentes entre la nube de puntos generada mediante la cámara Intel RealSense y aquella obtenida por el sistema de visión técnica TVS 2.

El procedimiento seguido consistió, primero, en establecer un modelo tridimensional de referencia usando la cámara Intel RealSense. Este escaneo se seleccionó como referencia debido a la precisión y resolución ampliamente documentadas de dicha cámara en aplicaciones de digitalización 3D. Posteriormente, la nube de puntos adquirida por el TVS 2 fue sometida al proceso de registro mediante ICP. Este algoritmo opera ajustando iterativamente la nube de puntos objetivo (en este caso, la obtenida por el TVS 2) a la nube de puntos de referencia (la generada con la RealSense), minimizando la distancia entre ambas hasta obtener la mejor correspondencia posible.

El algoritmo ICP permitió cuantificar la desviación entre ambos modelos tridimensionales y determinar numéricamente la exactitud de los datos obtenidos por el TVS 2 respecto al escaneo de referencia. Los resultados de este análisis mostraron una buena correspondencia global, aunque se detectaron algunas desviaciones locales asociadas principalmente a las regiones periféricas y bordes del objeto. Estas discrepancias pueden atribuirse a factores tales como la resolución propia del sistema TVS 2, su ángulo de incidencia respecto al objeto, y la presencia de ruido o puntos atípicos que no pudieron ser completamente eliminados durante las etapas anteriores de filtrado.

En términos cuantitativos, los resultados del registro indicaron una distancia promedio entre ambas nubes de puntos dentro de márgenes aceptables para las aplicaciones

previstas del sistema. Sin embargo, el análisis reveló la necesidad de ajustar ciertos parámetros operativos, tales como el ángulo óptimo de escaneo y la posición del objeto en el campo visual, para minimizar aún más las discrepancias entre las mediciones obtenidas por los dos sistemas.

4.1.1. Evaluación de similitudes

Nuestro análisis se enfocó en mediciones específicas dentro de características reconocibles en objetos fabricados con poliestireno, tales como el cráneo, la letra “A” y el número “6”. Estos objetos, elaborados con el mismo material, presentaron distintos grados de complejidad geométrica que sirvieron para evaluar el desempeño del TVS en condiciones variadas. Se observó que las dimensiones críticas para la identificación precisa de cada objeto—por ejemplo, el espacio entre los ojos y la nariz en el cráneo, o el área interior de la letra “A”—cayeron consistentemente dentro de las tolerancias establecidas por la ventana de escaneo, según se resume en la Tabla 4.3.

Este resultado indica que la replicación perfecta de mediciones en esquinas o puntos discretos estaría condicionada por la naturaleza misma del campo visual discontinuo del sistema TVS. En otras palabras, para alcanzar una fidelidad absoluta en la reproducción de estas mediciones puntuales sería necesario contar con una ventana de escaneo de campo visual continuo, situación que actualmente representa una limitación técnica del sistema utilizado.

Para complementar el análisis dimensional y evaluar con mayor precisión la calidad geométrica de los escaneos, se aplicó un algoritmo de registro de nubes de puntos. Este proceso consistió, primero, en simplificar la nube de puntos obtenida por la cámara Intel RealSense, con el objetivo de facilitar una comparación efectiva con la nube generada por el TVS. Posteriormente, se realizó una alineación manual preliminar seguida por una alineación precisa y automática mediante el algoritmo de registro, facilitando así la comparación global de las características clave presentes en ambas nubes.

Durante la comparación, se observó claramente que la calidad del ajuste (score) obtenido para las nubes de puntos generadas por el TVS estuvo fuertemente condicionada por la complejidad geométrica del objeto analizado. En casos como la letra “A”, que

presenta un volumen relativamente sencillo y superficies planas, se logró alcanzar una alta correspondencia entre ambas nubes de puntos, con puntuaciones de ajuste de hasta 0.96, tal como se observa en la Tabla 4.8.

En contraste, los objetos con características geométricas más complejas, como el cráneo de poliestireno, plantearon mayores desafíos. Las irregularidades y curvaturas significativas en la superficie, particularmente en áreas como los huesos cigomáticos, provocaron una pérdida considerable en la calidad de los escaneos. En estos casos, las puntuaciones de ajuste volumétrico alcanzaron un máximo aproximado de 0.88, disminuyendo notablemente hasta 0.50 en algunas de las caras individuales escaneadas, como se detalla en la Tabla 4.5 y Tabla 4.4. Este fenómeno ocurrió a pesar de que ciertas caras cumplían satisfactoriamente con los criterios establecidos en las mediciones clave.

La reducción de la precisión observada en superficies complejas se atribuyó principalmente a deformaciones derivadas de la reflexión del haz láser del TVS al incidir sobre superficies con curvatura considerable. Bajo estas condiciones, una parte significativa de la luz reflejada no logró alcanzar la apertura del sistema, provocando la aparición de ruido excesivo, puntos atípicos y regiones incompletas en los datos obtenidos. Este efecto impidió que el algoritmo de registro identificara claramente ciertas características, limitando la precisión general del ajuste.

Un ejemplo representativo de esta limitación puede observarse en la Figura 4.1, donde una sección de la nube de puntos generada por el TVS fue alineada con la nube obtenida mediante la cámara RealSense. En esta figura, las líneas negras, correspondientes al escaneo realizado con el TVS, siguen adecuadamente la estructura global del cráneo pero presentan interrupciones y desviaciones notorias en la zona que representa los huesos cigomáticos. Esto evidencia claramente cómo las características geométricas complejas afectan negativamente la calidad del escaneo con el TVS, especialmente en áreas críticas donde se requiere precisión elevada.

Adicionalmente, con la intención de mitigar estos problemas y mejorar la calidad global de los datos obtenidos, se aplicaron técnicas de filtrado estadístico enfocadas en reducir tanto la entropía como la entropía de Shannon en las nubes de puntos generadas por el sistema. Como puede observarse en las Tablas 4.1 y 4.2, estas técnicas permitieron

una reducción significativa de ambas medidas de entropía, indicando una disminución efectiva del ruido y la dispersión presentes en las nubes originales. Estos resultados reflejan claramente la utilidad de los métodos de filtrado estadístico aplicados, logrando mejorar la calidad y la precisión de los modelos tridimensionales obtenidos con el sistema TVS.

Tabla 4.1: Entropía aproximada calculada para cada modelo tridimensional.

Objeto	Posición	Paso angular de la mesa giratoria	Caras obtenidas	Entropía aproximada antes del filtrado	Entropía aproximada después del filtrado
Letra A	Pos1	90.0°	4	0.384	0.244
Número 6	Pos1	90.0°	4	0.449	0.252
Calavera	Pos1	45.0°	8	0.434	0.295
Calavera	Pos1	90.0°	4	0.357	0.267
Calavera	Pos1	22.5°	16	1.155	0.699
Calavera	Pos2	45.0°	8	0.591	0.362
Calavera	Pos2	90.0°	4	0.688	0.340
Calavera	Pos2	22.5°	16	1.391	0.591

Tabla 4.2: Entropía de Shannon calculada para cada modelo tridimensional.

Objeto	Posición	Paso angular de la mesa giratoria	Caras obtenidas	Entropía de Shannon antes del filtrado	Entropía de Shannon después del filtrado
Letra A	Pos1	90.0°	4	3.515	2.984
Número 6	Pos1	90.0°	4	3.007	2.878
Calavera	Pos1	45.0°	8	4.365	3.899
Calavera	Pos1	90.0°	4	3.850	3.162
Calavera	Pos1	22.5°	16	4.835	4.023
Calavera	Pos2	45.0°	8	4.871	3.846
Calavera	Pos2	90.0°	4	3.576	3.104
Calavera	Pos2	22.5°	16	3.777	3.255

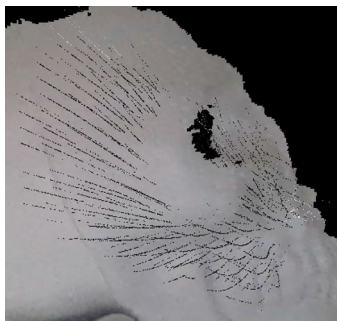


Figura 4.1: Face skull face scanned by TVS

Tabla 4.3: Comparación de medidas de objetos escaneados en ambas posiciones desde la referencia de coordenadas del TVS.

Objeto	Posición	Sección medida	Objetivo (cm)	Escaneado (cm)
Letra A	POS1	De base a punta superior	15.8	16.0
Letra A	POS1	Abertura central (vertical)	4.2	4.5
Número 6	POS1	De base a punta superior	9.6	9.7
Número 6	POS1	Abertura central (vertical)	2.2	2.2
Cráneo	POS1	Centro del ojo a nariz	3.5	3.6
Cráneo	POS1	De mandíbula a parte superior	13.8	14.0
Cráneo	POS2	Centro del ojo a nariz	3.5	3.7
Cráneo	POS2	De mandíbula a parte superior	13.8	14.3

Tabla 4.4: Puntuación de ajuste para las caras de la letra “A” y el número “6” en la posición POS1.

Objeto			
Letra A en POS1		Número 6 en POS1	
Rotación	Máxima puntuación de ajuste	Rotación	Máxima puntuación de ajuste
Rot1de4	0.913	Rot1de4	0.750
Rot2de4	0.343	Rot2de4	0.263
Rot3de4	0.887	Rot3de4	0.765
Rot4de4	0.440	Rot4de4	0.360

Tabla 4.5: Puntuaciones de las caras, normalizadas desde un máximo de 50.6 % y mínimo de 0 %.

Cráneo de poliestireno			
Posición 1		Posición 2	
Rotación	Máxima puntuación de ajuste	Rotación	Máxima puntuación de ajuste
Rot1de16	0.543	Rot1de16	0.260
Rot2de16	0.943	Rot2de16	0.202
Rot3de16	1.000	Rot3de16	0.159
Rot4de16	0.624	Rot4de16	0.000
Rot5de16	0.604	Rot5de16	0.481
Rot6de16	0.339	Rot6de16	0.159
Rot7de16	0.000	Rot7de16	0.000
Rot8de16	0.000	Rot8de16	0.000
Rot9de16	0.000	Rot9de16	0.000
Rot10de16	0.513	Rot10de16	0.000
Rot11de16	0.549	Rot11de16	0.000
Rot12de16	0.381	Rot12de16	0.149
Rot13de16	0.876	Rot13de16	0.206
Rot14de16	0.567	Rot14de16	0.240
Rot15de16	0.000	Rot15de16	0.503
Rot16de16	0.583	Rot16de16	0.381
Rot1de8	0.523	Rot1de8	0.130
Rot2de8	0.765	Rot2de8	0.210
Rot3de8	0.824	Rot3de8	0.015
Rot4de8	0.098	Rot4de8	0.220
Rot5de8	0.448	Rot5de8	0.080
Rot6de8	0.757	Rot6de8	0.402
Rot7de8	0.947	Rot7de8	0.341
Rot8de8	0.662	Rot8de8	0.543
Rot1de4	0.543	Rot1de4	0.341
Rot2de4	0.684	Rot2de4	0.143
Rot3de4	NA	Rot3de4	0.108
Rot4de4	0.583	Rot4de4	0.106

La siguiente tabla es una representación visual de las caras de los objetos utilizadas para calcular las puntuaciones de ajuste después del procesamiento de la nube de coordenadas espaciales. Los puntos rojos corresponden al objetivo, los puntos verdes corresponden a las mediciones del TVS. Haga clic sobre cada imagen para una mejor visualización en otra ventana.

Tabla 4.6

Object Foam Letter A					
				<i>Intentionally left blank</i>	<i>Intentionally left blank</i>
LetterARot1of4	LetterARot2of4	LetterARot3of4	LetterARot4of4	Intentionally left blank.	Intentionally left blank.
Object Foam Number 6					
				<i>Intentionally left blank</i>	<i>Intentionally left blank</i>
Number6Rot1of4	Number6Rot2of4	Number6Rot3of4	Number6Rot4of4	Intentionally left blank.	Intentionally left blank.
Object Foam skull					
					
SkullPOS1Rot1of16	SkullPOS1Rot2of16	SkullPOS1Rot3of16	SkullPOS1Rot4of16	SkullPOS1Rot5of16	SkullPOS1Rot6of16
					
SkullPOS1Rot7of16	SkullPOS1Rot8of16	SkullPOS1Rot9of16	SkullPOS1Rot10of16	SkullPOS1Rot11of16	SkullPOS1Rot12of16
					
SkullPOS1Rot13of16	SkullPOS1Rot14of16	SkullPOS1Rot15of16	SkullPOS1Rot16of16	SkullPOS1Rot1of8	SkullPOS1Rot2of8
					
SkullPOS1Rot3of8	SkullPOS1Rot4of8	SkullPOS1Rot5of8	SkullPOS1Rot6of8	SkullPOS1Rot7of8	SkullPOS1Rot8of8
					
SkullPOS1Rot1of4	SkullPOS1Rot2of4	SkullPOS1Rot3of4	SkullPOS1Rot4of4	SkullPOS2Rot1of16	SkullPOS2Rot2of16
					
SkullPOS2Rot3of16	SkullPOS2Rot4of16	SkullPOS2Rot5of16	SkullPOS2Rot6of16	SkullPOS2Rot7of16	SkullPOS2Rot8of16
					
SkullPOS2Rot9of16	SkullPOS2Rot10of16	SkullPOS2Rot11of16	SkullPOS2Rot12of16	SkullPOS2Rot13of16	SkullPOS2Rot14of16
					
SkullPOS2Rot15of16	SkullPOS2Rot16of16	SkullPOS2Rot1of8	SkullPOS2Rot2of8	SkullPOS2Rot3of8	SkullPOS2Rot4of8

Continúa en la siguiente página

Tabla 4.7

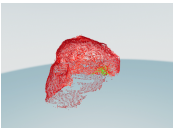
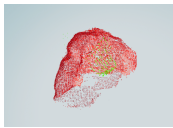
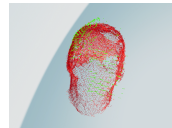
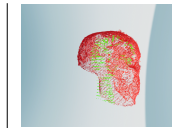
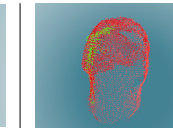
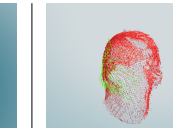
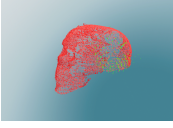
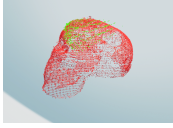
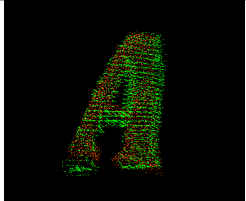
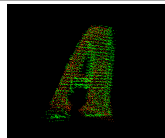
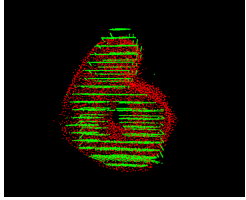
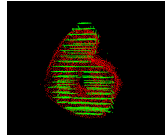
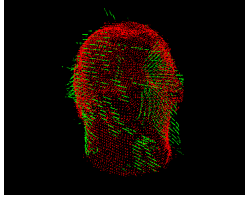
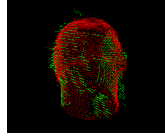
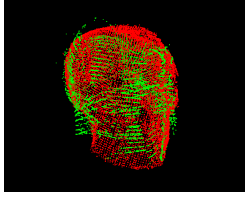
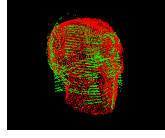
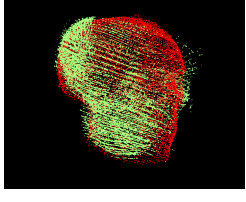
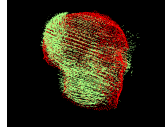
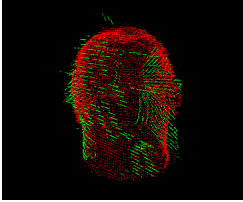
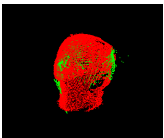
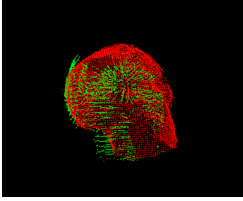
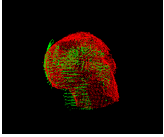
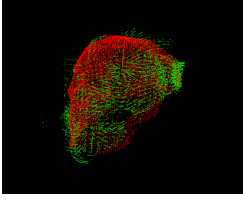
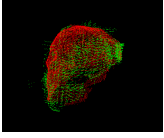
					
SkullPOS2Rot5of8	SkullPOS2Rot6of8	SkullPOS2Rot7of8	SkullPOS2Rot8of8	SkullPOS2Rot1of4	SkullPOS2Rot2of4
		<i>Intentionally left blank</i>	<i>Intentionally left blank</i>	<i>Intentionally left blank</i>	<i>Intentionally left blank</i>
SkullPOS2Rot3of4	SkullPOS2Rot4of4	Intentionally left blank.	Intentionally left blank.	Intentionally left blank.	Intentionally left blank.

Tabla 4.8: Puntuaciones de ajuste para volúmenes. Hacer clic en cada imagen para visualizarla mejor en otra ventana.

Objeto	Posición	Rotación	Puntuación	Imagen plana en 3D	Vista externa 3D
Letra A	Pos1	90.0°	0.961		 Clic para vista 3D
Número 6	Pos1	90.0°	0.919		 Clic para vista 3D
Cráneo	Pos1	90.0°	0.219		 Clic para vista 3D
Cráneo	Pos1	45.0°	0.853		 Clic para vista 3D
Cráneo	Pos1	22.5°	0.883		 Clic para vista 3D

Continúa en la siguiente página

Tabla 4.8 (continuación)

Objeto	Posición	Rotación	Puntuación	Imagen plana en 3D	Vista externa 3D
Cráneo	Pos2	90.0°	0.118		 Clic para vista 3D
Cráneo	Pos2	45.0°	0.319		 Clic para vista 3D
Cráneo	Pos2	22.5°	0.716		 Clic para vista 3D

4.2. Resultados del experimento para el reconocimiento de colores

Como mencionado anteriormente, en esta fase se procedió al procesamiento de las señales captadas. Inicialmente se determinó la posición en el espacio 3D, representada por (x, y, z) , utilizando las Ecuaciones 3.17 a 3.19, y a continuación se extrajeron los parámetros temporales detallados en la Tabla 3.6.

Debido a la naturaleza no lineal del comportamiento del TVS a lo largo del FOV, evidenciado en la figura de segmentación del FOV de [82], no todas las señales optoelectrónicas resultaron aptas para calcular la posición 3D. Luego de eliminar tanto las señales inadecuadas como los valores atípicos, se obtuvieron finalmente 4701 mediciones de coordenadas, tal y como se resume en la Tabla 4.9.

Tabla 4.9: Relación de mediciones de coordenadas obtenidas.

Tipo de papel	Profundidad de las mediciones (x)			Cantidad de mediciones
	70 cm	80 cm	90 cm	
Rojo brillante	320	393	336	1049
Rojo mate	350	481	359	1190
Azul brillante	62	11	54	127
Azul mate	0	0	0	0
Amarillo brillante	238	230	212	680
Amarillo mate	445	768	442	1655
Total	1415	1883	1403	
Total General				4701

De acuerdo a la distribución de frecuencia de las mediciones obtenidas (clasificadas por el ángulo de posicionamiento láser) y analizadas por la profundidad de medición, se puede ver en la Figura 4.2 a) que, en general, el mejor FOV varía con el ángulo del LP desde 46° hasta 60° , las distancias de profundidad afectan la calidad de las señales para poder realizar las mediciones, la Figura 4.2 c) para las mediciones a 80 cm presenta el FOV más amplio y la cantidad más alta de mediciones obtenidas con éxito de acuerdo a la Tabla 4.9.

Replicando el análisis de distribución de frecuencia del ángulo del LP, ahora de acuerdo al tipo de papel, se ha encontrado que los papeles mates aportan la mayor cantidad de mediciones exitosas, con excepción del color azul, que no presentó ninguna reflexión del láser emitido sobre él, como se muestra en la Figura 4.3 y se describe en la Tabla 4.9.

Las señales optoelectrónicas de una misma coordenada 3D ($x = 80 \text{ cm}$, $\beta = 0^\circ$, $C_{ij} = 54^\circ$) con tipos diferentes de papeles se seleccionaron para comparar con el fin de hallar diferencias relacionadas con el color de los papeles, como se muestra en la Figura 4.5, confirmando la necesidad de un método automático para diferenciarlas entre sí. Posteriormente, se extrajeron los parámetros del dominio del tiempo, obteniendo un conjunto de datos con una matriz de 4701×32 , 4701 instancias, 4701 etiquetas y 31 atributos (parámetros) de acuerdo a la Tabla 3.6.

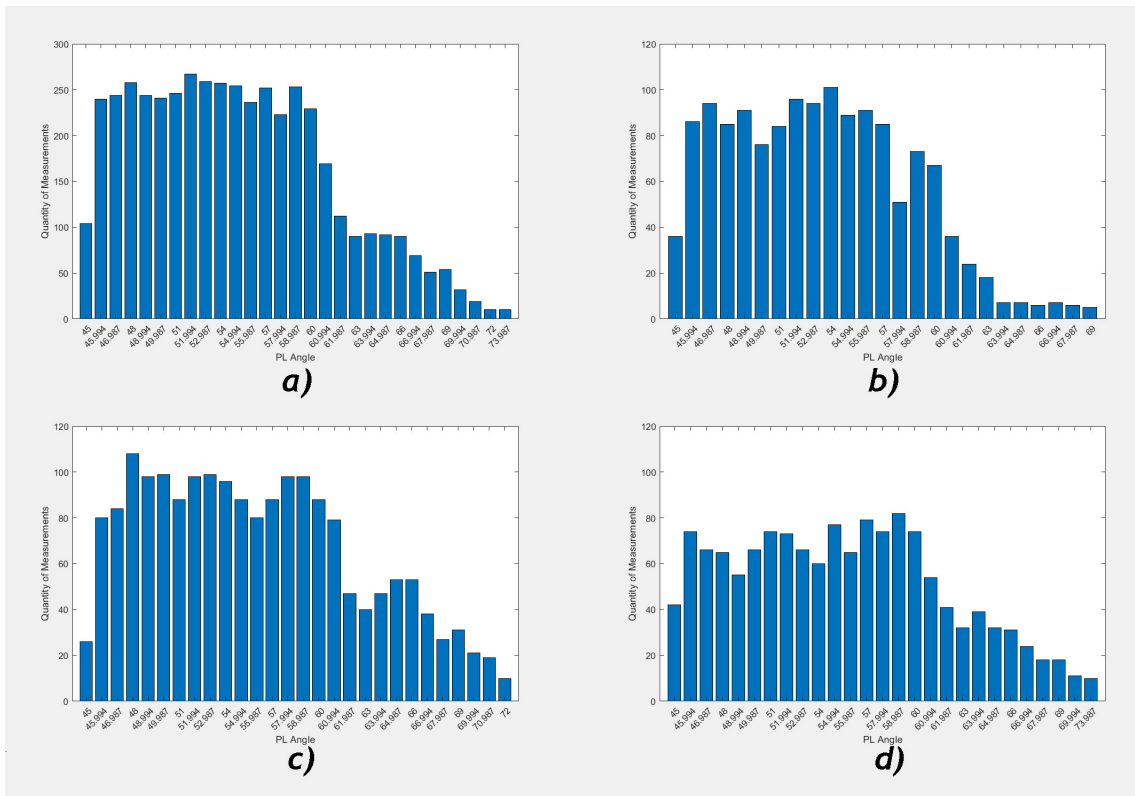
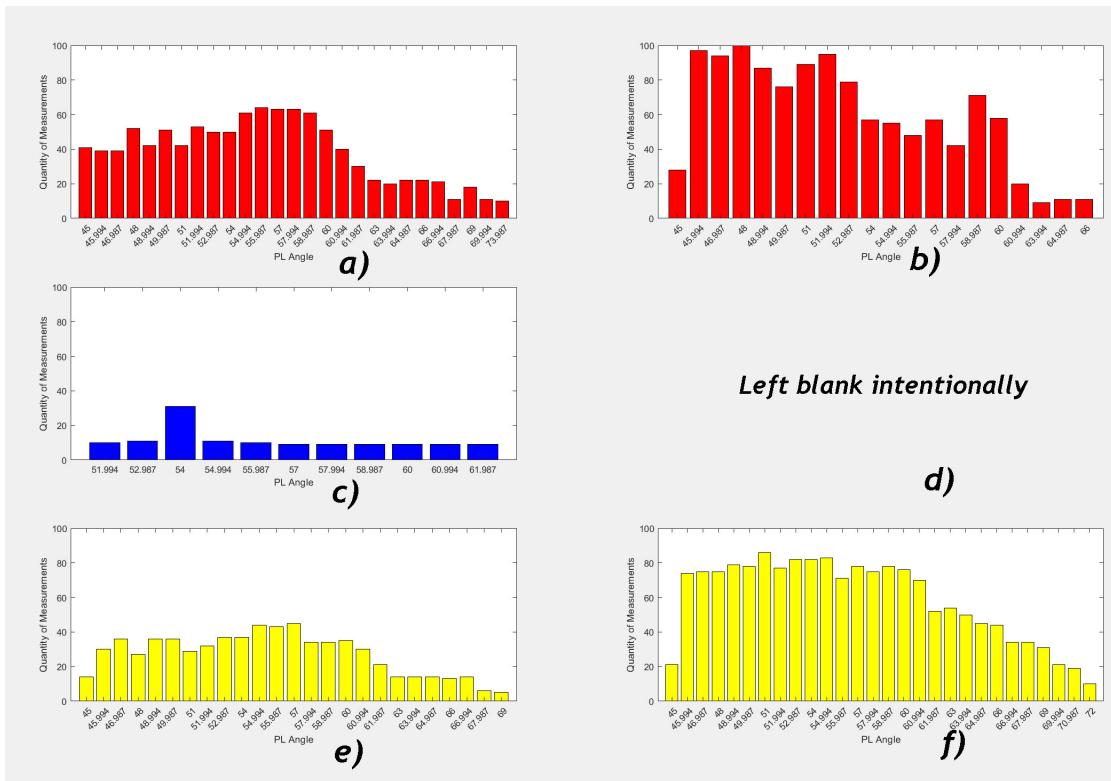


Figura 4.2: Distribución de frecuencia de la cantidad de mediciones exitosas con el ángulo del LP a diferentes profundidades; a) Todas las medidas; b) Medidas a 70 cm; c) Medidas a 80 cm; d) Medidas a 90 cm.



Left blank intentionally

d)

Figura 4.3: Distribución de frecuencia de la cantidad de mediciones exitosas con el ángulo del LP en diferentes tipos de papel a) Rojo brillante; b) Rojo mate; c) Azul brillante; d) Azul mate (Sin resultados); e) Amarillo brillante; f) Amarillo mate.

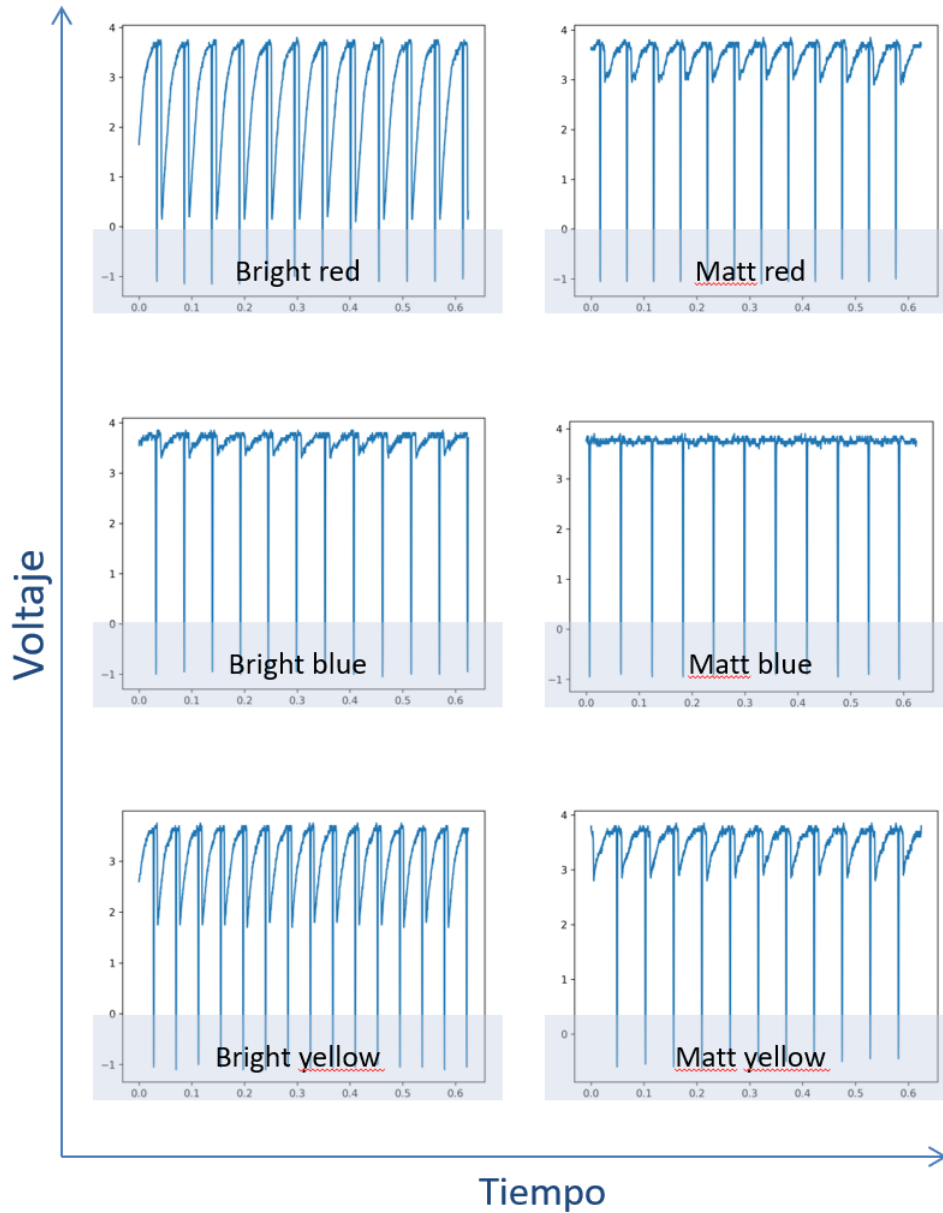


Figura 4.4: Comparación de las señales optoelectrónicas en la misma coordenada 3D espacial.

En la tercera etapa la calidad de las mediciones del TVS se evaluó mediante la medición del ángulo del posicionador láser C_{ij} contra la profundidad; ver Figura 16 a), y el ángulo del posicionador láser C_{ij} contra tipo de color, ver Figura 16 b), el error absoluto fue calculado con la Ec. 4.1, y escalado de 0 a 1 por medio de la Ec. 4.2, donde

ϵ es el error absoluto de medición, $\epsilon_{S_{0-1}}$ es el error absoluto de medición escalado de 0 a 1, ϵ_{min} es el valor mínimo de error y ϵ_{max} es el valor máximo de error.

Todos los conjuntos de datos de subgrupos se evaluaron para determinar la exactitud de las mediciones por la media de $\epsilon_{S_{0-1}}$, y la precisión de las mediciones se determinó por su desviación estándar, como se define en la Tabla 4.10 (exactitud y precisión en cm). Donde, el conjunto de datos con mayor exactitud en las mediciones se realizaron con el papel rojo mate, seguido por el papel amarillo, mate, a una profundidad de 80 cm. Mientras que el conjunto de datos con mayor precisión es el que se realizó a una distancia de 70 cm, mientras que los resultados de la precisión en función de los colores no fue determinante.

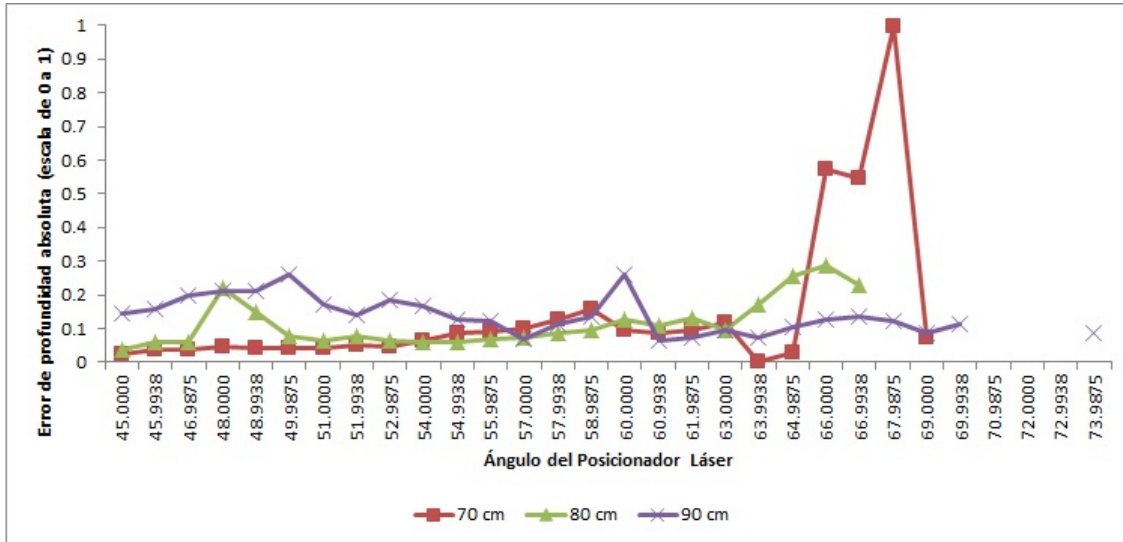
$$\epsilon = |Profundidad\ real - Profundidad\ medida| \quad (4.1)$$

$$\epsilon_{S_{0-1}} = \frac{\epsilon - \epsilon_{min}}{\epsilon_{max} - \epsilon_{min}} \quad (4.2)$$

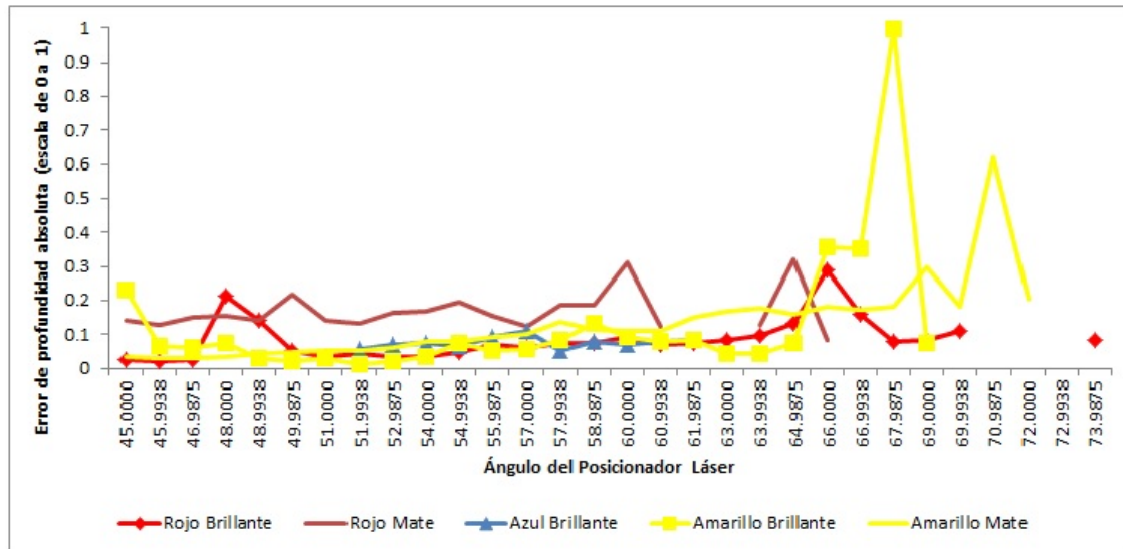
Tabla 4.10: Evaluación de Exactitud y Precisión de Mediciones

Conjunto de datos	Exactitud	Precisión
Mediciones a una profundidad de 70 cm	0.4113	0.1410
Mediciones a una profundidad de 80 cm	0.3595	0.2081
Mediciones a una profundidad de 90 cm	0.3854	0.2243
Papel Rojo Brillante	0.4731	0.1929
Papel Rojo Mate	0.3382	0.2373
Papel Azul Brillante	0.4169	0.2150
Papel Amarillo Brillante	0.4246	0.2117
Papel Amarillo Mate	0.3552	0.2267

En la cuarta etapa, el conjunto de datos se evaluó para identificar a qué tipo de papel corresponde cada instancia. El conjunto de datos originales con 4701 instancias se preprocesó para eliminar valores atípicos, cada una de las 31 columnas de atributos se evaluó de manera separada; cada instancia con al menos un valor atípico en cualquiera de sus columnas de atributos se eliminó del conjunto de datos. Cualquier valor en las columnas de atributos mayor a 3 desviaciones medianas absolutas (MAD por sus siglas en inglés) registradas lejos de la mediana se consideró un valor atípico. Se eliminaron



a)



b)

Figura 4.5: Error de medición de ángulo de posicionamiento láser. a) Por distancia de profundidad; b) Por tipo de papel.

1424 instancias, resultando en un segundo conjunto de datos sin valores atípicos con 3277 instancias. Un tercer conjunto de datos se generó reemplazando los valores atípicos con una interpolación lineal de entradas no atípicas, conservando el mismo tamaño del conjunto de datos originales. Cada uno de los tres conjuntos de datos se normalizó con tres opciones *i*) mediante el método de puntuación Z (Z Score) para tener una media de 0 y una desviación de 1; *ii*) escalando los valores de 0 a 1; *iii*) escalando los valores de -1 a 1.

Se implementó validación cruzada para seleccionar la mejor partición de datos, 5 particiones se seleccionaron estimando la precisión en cada partición para evitar un sobre ajuste durante el entrenamiento de los modelos supervisados; 29 entrenamientos de modelos se exploraron para seleccionar un tipo de clasificador debido a que el mejor tipo de clasificador depende de las características del conjunto de datos, se obtuvieron los resultados descritos en la Tabla 4.11, donde A es el conjunto de datos originales, B es el conjunto de datos sin valores atípicos, C es el conjunto de datos con los valores atípicos reemplazados por los valores interpolados, A_N , B_N , y C_N son los conjuntos de datos A, B y C normalizados por el método de puntuación Z (Z Score), $A_{.01}$, $B_{.01}$, $C_{.01}$ son los conjuntos de datos A, B y C escalados de 0 a 1. $A_{.11}$, $B_{.11}$, $C_{.11}$ son los conjuntos de datos A, B y C escalados de -1 a 1. Obteniendo con el modelo de Ensemble de Subespacio KNN con los conjuntos de datos $A_{.01}$ y $A_{.11}$ los resultados con más alta exactitud (accuracy), con un valor de 99.3%.

Para el modelo de Ensemble de Subespacio KNN con el conjunto de datos $A_{.01}$, la Figura 4.6 muestra la matriz de confusión de validación, donde TVP son las “Tasas de Verdaderos Positivos”, TFN son las “Tasas de Falsos Negativos”, VPP son los “Valores Predictivos Positivos” y TDF son las “Tasas de Descubrimiento Falso”.

El Análisis de Componentes Principales (ACP) se aplicó para reducir la dimensionalidad del conjunto de datos, pero conforme disminuyó el número de componentes, la precisión de la validación también disminuyó, por lo que fue descartado.

Tabla 4.11: Porcentaje de exactitud de validación de los clasificadores de modelos

Conjunto de Datos													
No.	Modelo Clasificador	A	A_N	A_01	A_11	B	B_N	B_01	B_11	C	C_N	C_01	C_11
1	Fine Tree	90.6	90.2	90.9	90.0	90.0	90.4	88.6	88.7	88.3	89.1	88.0	89.5
2	Medium Tree	79.7	77.9	78.9	79.9	75.2	75.3	74.3	76.4	78.4	78.3	78.8	77.9
3	Coarse Tree	69.7	68.1	68.8	69.1	69.7	69.7	69.6	70.2	69.5	69.6	69.6	69.4
4	Linear Discriminant	80.6	80.8	80.7	80.8	85.4	85.3	85.0	85.1	81.6	81.6	81.4	81.3
5	Quadratic Discriminant	Fallo	Fallo	Fallo	Fallo	Fallo	Fallo	Fallo	Fallo	Fallo	Fallo	Fallo	Fallo
6	Gaussian Naive Bayes	69.3	69.5	69.6	69.4	59.7	59.9	59.5	59.7	69.9	70.0	70.2	69.9
7	Kernel Naive Bayes	73.1	73.0	73.2	73.2	63.2	63.7	63.0	63.6	71.3	71.3	71.4	71.4
8	Linear SVM	83.6	83.6	83.8	83.8	83.8	83.6	83.8	84.1	84.4	84.6	84.7	84.5
9	Quadratic SVM	94.5	94.3	94.3	94.2	94.5	95.0	94.8	95.0	94.8	94.9	94.8	95.1
10	Cubic SVM	96.3	96.4	96.2	96.2	96.8	96.7	96.4	96.2	96.4	96.7	96.6	96.4
11	Fine Gaussain SVM	95.6	95.8	95.6	95.5	88.6	88.6	88.6	88.4	93.8	94.1	94.0	94.3
12	Medium Gaussain SVM	90.8	90.7	90.7	90.7	93.0	92.5	92.6	92.6	93.4	93.4	93.5	93.4
13	Coarse Gaussain SVM	80.2	80.3	80.2	80.2	79.5	79.7	79.4	79.7	81.2	81.3	81.3	81.2
14	Fine KNN	96.0	96.4	96.3	96.5	95.6	95.5	95.6	95.6	96.2	96.3	96.5	96.8
15	Medium KNN	94.0	94.2	94.3	94.4	93.4	93.4	93.4	93.3	94.9	94.8	94.7	95.1
16	Coarse KNN	83.5	83.5	83.5	83.6	83.1	83.2	83.1	82.9	85.1	85.4	85.3	85.4
17	Cosine KNN	93.6	93.0	93.2	93.6	92.6	93.2	92.7	92.9	93.5	93.8	93.8	93.9
18	Cubic KNN	92.3	96.3	92.1	92.3	92.5	92.2	92.2	92.0	93.7	93.9	93.5	93.9
19	Weighted KNN	95.7	92.0	96.0	96.3	95.3	95.5	95.3	95.4	96.4	96.3	96.3	96.6
20	Ensemble Boosted Trees	86.5	86.4	85.9	86.5	85.1	84.5	85.3	84.9	85.9	86.5	86.1	86.2
21	Ensemble Bagged Trees	97.2	97.6	97.5	97.7	96.7	96.8	96.8	96.7	97.4	97.7	97.5	97.6
22	Ensemble Subspace Discriminant	76.1	76.3	76.4	76.3	83.5	83.5	83.4	83.5	79.5	79.7	79.4	79.5
23	Ensemble Subspace KNN	90.2	97.8	99.3	99.3	88.3	96.7	97.2	96.9	90.8	97.5	97.8	97.8
24	Ensemble RUSBoosted Trees	82.9	82.9	82.5	83.3	73.6	74.4	72.6	74.5	80.0	80.9	81.6	79.9
25	Narrow Neural Network	95.9	95.0	95.7	95.6	95.2	94.3	94.9	94.4	95.3	95.1	95.1	95.4
26	Medium Neural Network	97.1	96.9	97.1	97.3	96.3	96.0	96.5	96.0	97.0	97.0	97.2	96.7
27	Wide Neural Network	97.3	97.4	97.5	97.4	96.7	96.2	96.8	96.4	97.2	97.1	97.3	96.9
28	Blayered Neural Network	96.4	96.3	96.7	96.2	95.2	95.9	95.1	95.3	96.1	96.4	95.6	96.5
29	Trilayered Neural Network	96.7	96.3	96.7	96.6	95.4	95.3	95.5	95.5	96.2	95.9	95.8	96.0

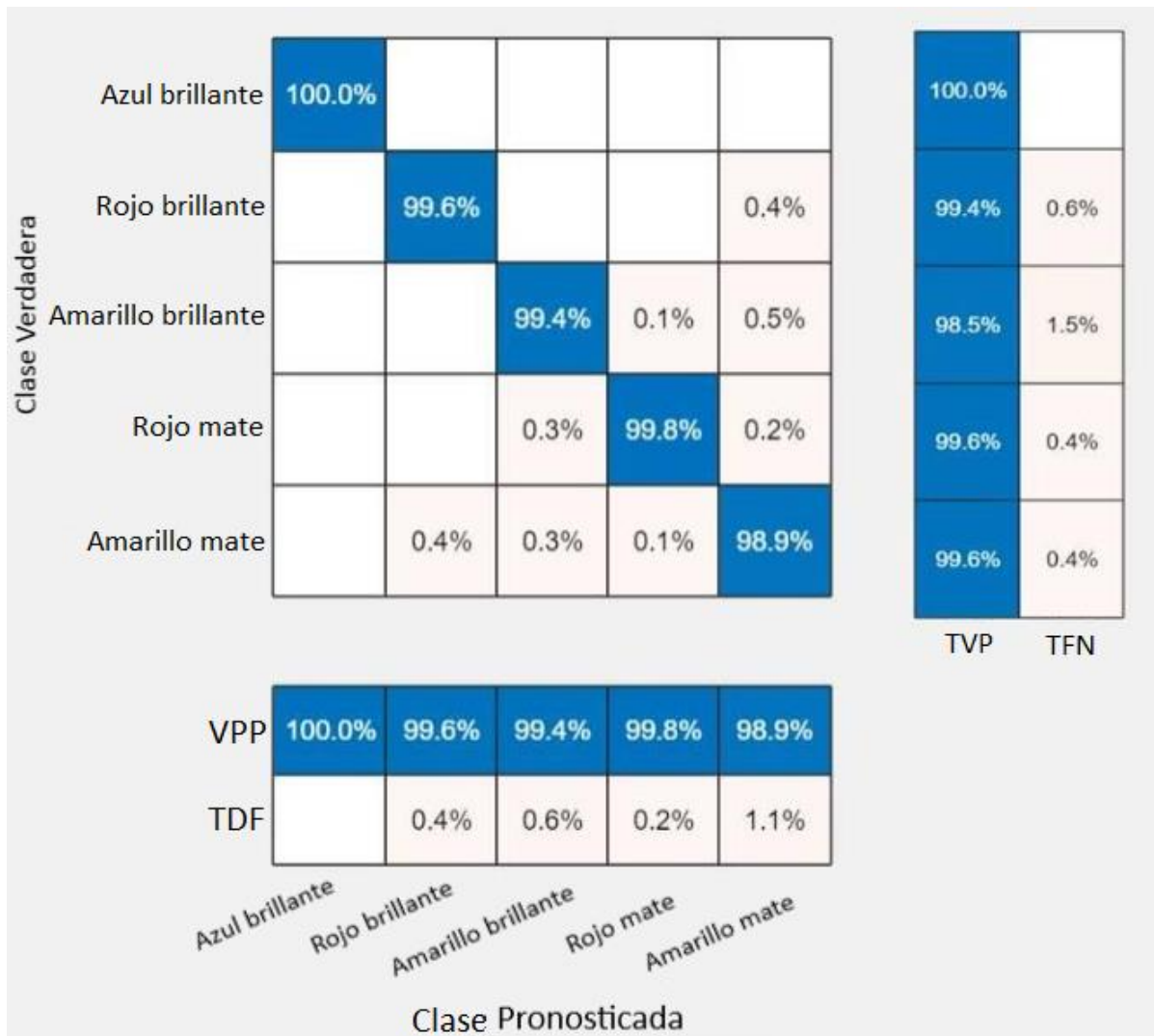


Figura 4.6: Matriz de confusión de validación del modelo de Ensemble de Subespacio KNN

Conclusiones

5.1. Experimento para la obtención de modelos tridimensionales

5.1.1. Comparación con datos sin filtrar

Los beneficios del posprocesamiento al utilizar este sistema se pueden ver al comparar los datos filtrados con su contraparte sin filtrar. En particular, para algunas de las caras no filtradas del cráneo de poliestireno, la presencia de demasiado ruido en los datos hizo imposible realizar mediciones clave precisas, mientras que los algoritmos de registro global no fueron efectivos, fallando en producir una puntuación mayor a 0. Esto resalta la importancia de los métodos de filtrado estadístico utilizados y el procesamiento de datos para mitigar las inexactitudes en el TVS, mejorando así la claridad y la usabilidad de las nubes de puntos generadas. Los resultados de los datos filtrados con respecto a la reducción de entropía se pueden ver en la Tabla 3.4 para el “Método de Entropía Aproximada” y en la Tabla 3.5 para el “Método de Entropía de Shannon”, las mediciones clave de algunos de los volúmenes se pueden ver en la Tabla 4.3, las puntuaciones de ajuste para los volúmenes y las caras se pueden ver en la Tabla 4.4 para las caras de la Letra A y el Número 6, en la posición 1. Nótese que la evaluación en la posición 2 no fue requerida debido a que los objetos son principalmente planos y no se produce oclusión del láser debido a la planitud de la cara del objeto. La Tabla 4.5 es para las caras del cráneo en la posición 1 y la posición 2, debido a que el cuerpo del

cráneo no es plano y presenta oclusión del láser en algunos puntos protuberantes (por ejemplo, en la nariz, lo que impide ver la mejilla) cuando está posicionado en la posición 1, y se deseaba evaluar las mediciones en la posición 2 para evitar la oclusión del láser para la apertura de escaneo. Finalmente, se muestra el resultado en la Tabla 4.8 para los volúmenes de la Letra A, Número 6, y el cráneo. También se ha proporcionado un enlace a una vista 3D en vivo [83].

5.1.2. Limitaciones e incertidumbres

La precisión de las mediciones y la efectividad del algoritmo de registro están limitadas por la resolución y fidelidad del proceso de escaneo, el cual está influenciado por las propiedades del material de los objetos y el entorno de escaneo, así como la antigüedad del sistema y el comportamiento de los sistemas de escaneo láser. Además, la naturaleza discreta de la ventana de escaneo introduce un elemento inevitable de aproximación en la captura de los detalles finos de geometrías complejas. Tales factores contribuyen a la variabilidad en las puntuaciones de ajuste y resaltan los desafíos lograr alta fidelidad en el escaneo 3D y la comparación de modelos.

5.1.3. Experimento para el reconocimiento de colores

Se ha resaltado la relevancia de la teledetección y de los sistemas de visión artificial de última generación en el contexto de tecnologías esenciales y emergentes, junto con sus aplicaciones más destacadas. En este sentido, se presentó un sistema de visión técnica, describiéndose sus ventajas frente a otras tecnologías y exponiéndose la intención de potenciarlo para que pueda realizar mediciones de color en coordenadas tridimensionales. Se analizó asimismo el papel de las mediciones 3D como una aproximación a la percepción visual humana, enfatizando el atractivo y la posibilidad de incorporar información cromática.

A partir de ello, el estudio se centro en extraer datos de color asociados a las coordenadas espaciales de un objeto mediante un sistema basado en principios de escaneo óptico y triangulación dinámica. La tecnología TVS utilizada permite recolectar una

gran cantidad de puntos de datos que, en conjunto, capturan de manera integral la estructura geométrica de los objetos. Dichas mediciones en 3D aportan detalles precisos sobre la topografía de la superficie, facilitan la obtención de texturas y permiten incluso reconstruir modelos completos, lo que resulta especialmente útil para objetos con geometrías complejas.

Una vez diseñado el experimento, se realizaron mediciones en el campo visual considerando diversas coordenadas de color 3D. Las señales optoelectrónicas obtenidas fueron caracterizadas, extrayéndose 31 atributos por cada muestra, lo que generó una matriz de 4701 x 32 (4701 instancias, 4701 etiquetas y 31 variables). Este conjunto de datos fue posteriormente preprocesado para su aplicación en modelos supervisados de inteligencia artificial, evaluándose un total de 29 modelos. Como resultado, se logró identificar con éxito un modelo basado en un Ensemble de Subespacio KNN que alcanzó una precisión de validación del 99.3%, confirmando así la eficacia propuesta del sistema de visión técnica.

Para investigaciones futuras se sugiere ampliar el diseño experimental incorporando una mayor diversidad de colores en las coordenadas 3D, con el objetivo de enriquecer el conjunto de datos para el entrenamiento, evaluar algoritmos no supervisados y agregar el modelo predictivo obtenido al proceso de medición de coordenadas 3D mediante TVS.

5.2. Discusión y conclusiones

En este estudio, se exploró el rendimiento del Sistema de Visión Técnica (TVS) en diferentes resoluciones de escaneo y configuraciones de ventana con el objetivo de obtener un sólido 3D fiel al objeto escaneado original, basándose en las caras escaneadas por el TVS utilizando una mesa giratoria, centrado en la precisión de los escaneos 3D para capturar las características clave de diferentes objetos de poliestireno. El objetivo original del estudio era investigar las inconsistencias observadas en la calidad de los escaneos 3D producidos por el sistema al escanear objetos ubicados en diferentes posiciones dentro del campo de visión del sistema. Los desafíos significativos fueron planteados por el ruido y otras particularidades del TVS, como la velocidad de la apertura de escaneo,

que tuvieron que ser abordados mediante diferentes técnicas para mitigar estos problemas. Entre las técnicas utilizadas, los filtros estadísticos usados para el procesamiento de datos y el algoritmo MLS aumentaron significativamente la fidelidad de los modelos 3D, así como la tasa de ajuste de la comparación entre las nubes de puntos. El estudio ha sido exitoso en identificar áreas del campo de visión que parecen producir mejores resultados en la obtención de sólidos 3D comparables con el original en sus características clave y mantienen una puntuación de ajuste similar al objeto original mientras reducen la entropía aproximada de los datos.

Los experimentos se realizaron en un entorno controlado debido a que el propósito original y principal de la investigación era escanear y reconstruir digitalmente objetos usando una mesa giratoria. Esta configuración permitió obtener múltiples escaneos desde diferentes caras de un objeto para la ingeniería inversa. Sin embargo, después de reconocer el potencial del sistema TVS para el escaneo 3D a resoluciones reducidas y ventanas de escaneo óptimas, exploramos el uso de técnicas de filtrado de datos estadísticos y modelos de regresión. Estos métodos permiten generar un escaneo 3D que sigue siendo reconocible con la menor cantidad de puntos 3D. Este enfoque equilibra la necesidad de una representación precisa del objeto con la eficiencia requerida para aplicaciones prácticas, como la navegación robótica. Al reducir el número de puntos necesarios para un escaneo confiable, un sistema de escaneo como el TVS podría procesar y comprender más rápidamente la escena en el campo de visión del robot, apoyado por algoritmos de clasificación de aprendizaje profundo. La aplicación potencial en la navegación robótica implica equilibrar el número de puntos escaneados con el tiempo necesario para comprender la escena, optimizando tanto la precisión como la eficiencia del procesamiento.

En este estudio, fue posible obtener figuras sólidas reconocibles a partir de nubes de puntos de baja calidad y alto ruido creadas usando un Sistema de Visión Técnica, así como reducir la entropía en los datos hasta un 57%. Los datos obtenidos del sistema representaron múltiples caras de un objeto rotado, cada cara fue filtrada utilizando métodos estadísticos como Tau-Thompson modificado y cuantiles de la distribución Chi^2 . Luego,

los datos se suavizaron utilizando el algoritmo de mínimos cuadrados móviles (MLS) y se rotaron para ser ensamblados en una única nube de puntos representativa del volumen original. Para validar que los resultados de los algoritmos eran precisos, los datos se compararon con nubes de puntos de una cámara Intel RealSense utilizando registro de nubes de puntos. Las nubes de puntos fueron comparadas con cada cara escaneada por el TVS, así como con las representaciones rotadas. Estos resultados se mostraron en la Tabla 4.5 para las caras de los objetos y en la Tabla 4.8 para los volúmenes. Los resultados mostraron que para cada cara, la puntuación de ajuste máxima obtenida fue del 50.6 %, y una vez que las caras fueron rotadas y ensambladas en una única representación en nube de puntos del volumen original, la puntuación de ajuste máxima obtenida fue del 96 %. También se demostró que la primera posición seleccionada para el campo de visión del sistema fue el lugar óptimo para escanear objetos con este sistema en particular.

Sin embargo, las limitaciones de nuestro entorno experimental, incluida la elección de los materiales del objeto y el entorno de laboratorio controlado, sugieren la necesidad de más estudios para evaluar la aplicabilidad de estas técnicas en contextos de escaneo variados. Las investigaciones futuras también deberían considerar la integración de algoritmos alternativos y la exploración de configuraciones de escaneo adaptativas para abordar los desafíos matizados presentados por las complejidades geométricas de diferentes objetos. Los hallazgos experimentales también confirmaron que la calidad de los escaneos varió significativamente según la posición, con la Posición 2 ofreciendo consistentemente resultados inferiores en comparación con la Posición 1. Esta variación puede atribuirse a factores como el ángulo de incidencia del láser, la distancia al escáner o posibles obstrucciones que afectan la línea de visión del escáner en diferentes posiciones.

En última instancia, este estudio contribuye a la evolución del TVS añadiendo la capacidad de obtener reconstrucciones 3D completas de objetos, al tiempo que ofrece ideas sobre la optimización de los procesos de escaneo y el potencial para mejorar la reconstrucción de modelos. Al continuar refinando estos métodos, las complejidades del modelado digital pueden integrarse a las capacidades del sistema, allanando el camino para aplicaciones más amplias en el diseño industrial, la conservación del patrimonio,

y más allá.

Los hallazgos de este trabajo indican claramente que la Posición 1 ofrece consistentemente escaneos de mayor calidad en comparación con la Posición 2, lo que destaca la necesidad de una colocación óptima del escáner para garantizar la precisión de los datos. Para abordar los desafíos planteados por posiciones de escaneo menos favorables, el desarrollo futuro del TVS debería enfocarse en explorar estrategias de escaneo adaptativas que puedan mitigar los efectos de posicionamiento subóptimo y aumentar la flexibilidad del sistema. El uso de técnicas de filtrado de datos más avanzadas y de mejora también será útil para mejorar la fidelidad de los escaneos 3D en diversos entornos operativos, así como para obtener la capacidad de compensar las desventajas posicionales mediante técnicas de procesamiento de datos más avanzadas. Además, las mejoras continuas en el diseño, la operación y el hardware del sistema podrían estabilizar aún más los resultados de los escaneos, asegurando una calidad más consistente en diferentes entornos operativos.

Bibliografía

- [1] Muhammed Enes Atik, Zaide Duran, Mustafa Yanalak, Dursun Zafer Seker, and Abdulaziz Ak. 3d modeling of historical measurement instruments using photogrammetric and laser scanning techniques. *Digital Applications in Archaeology and Cultural Heritage*, 30, 2023. Cited by: 2.
- [2] Ramazan Alper Kuçak and Adalet Dervişoğlu. The heritage building information modelling of the church of st. sophia in ohrid, north macedonia. *Mediterranean Archaeology and Archaeometry*, 23(3):31 – 43, 2023. Cited by: 0.
- [3] Xudong Li, Huijie Zhao, Yang Liu, Hongzhi Jiang, and Yun Bian. Laser scanning based three dimensional measurement of vegetation canopy structure. *Optics and Lasers in Engineering*, 54:152–158, March 2014.
- [4] Jeffrey Richardson, Monika Moskal, and Jonathan Bakker. Terrestrial Laser Scanning for Vegetation Sampling. *Sensors*, 14:20304–20319, October 2014. Publisher: MDPI.
- [5] François Blais, Jean-Angelo Beraldin, and Sabry El-Hakim. Range Error Analysis of an Integrated Time-of-Flight, Triangulation, and Photogrammetric 3D Laser Scanning System. In *SPIE Proceedings*, Orlando, Florida, April 2000. SPIE.
- [6] Milan Edl, Marek Mizerák, and Jozef Trojan. 3D laser scanners, history and applications. *Acta Simulatio*, 4(4):1–5, December 2018.
- [7] Frederic Hake, Paula Lippmann, Hamza Alkhatib, Vincent Oettel, and Ingo Neumann. Automated damage detection for port structures using machine learning algorithms in heightfields. *Applied Geomatics*, 15(2):349 – 357, 2023. Cited by: 2; All Open Access, Hybrid Gold Open Access.
- [8] Seok Hyun Moon, Ock woo Nam, and Yun Soo Choi. Accuracy comparison analysis of mobile handheld and backpack laser scanners for indoor 3d mapping; year = 2023, journal = Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, volume = 41, number = 6, pages = 529 – 536, doi = 10.7848/ksgpc.2023.41.6.529, url = <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85184778652doi=10.7848>

- [9] Aditya M Intwala and Atul Magikar. A review on process of 3d model reconstruction. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 2851–2855. IEEE, 2016.
- [10] Adam L. Kaczmarek and Bernhard Blaschitz. Equal baseline camera array—calibration, testbed and applications. *Applied Sciences (Switzerland)*, 11(18), 2021. Cited by: 2; All Open Access, Gold Open Access.
- [11] Piotr Siekański, Krzysztof Magda, Krzysztof Malowany, Jan Rutkiewicz, Adam Styk, Jakub Krzesłowski, Tomasz Kowaluk, and Andrzej Zagórski. On-line laser triangulation scanner for wood logs surface geometry measurement. *Sensors*, 19(5), 2019.
- [12] Tu Dawei, Lan Hao, and Zhang Xi. Digital three-dimensional reconstruction technology of cultural relics. *Laser and Optoelectronics Progress*, 56(19), 2019. Cited by: 8.
- [13] Xian-Feng Han, Jesse S. Jin, Ming-Jie Wang, Wei Jiang, Lei Gao, and Liping Xiao. A review of algorithms for filtering the 3d point cloud. *Signal Processing: Image Communication*, 57:103–112, 2017.
- [14] Chao Chuan Jia, Chuan Jiang Wang, Ting Yang, Bing Hui Fan, and Fu Gui He. A 3d point cloud filtering algorithm based on surface variation factor classification. *Procedia Computer Science*, 154:54–61, 2019. Proceedings of the 9th International Conference of Information and Communication Technology [ICICT-2019] Nanning, Guangxi, China January 11-13, 2019.
- [15] Seon-Ho Lee and Chang-Su Kim. Saf-nets: Shape-adaptive filter networks for 3d point cloud processing. *Journal of Visual Communication and Image Representation*, 79:103246, 2021.
- [16] Wendy Flores-Fuentes, Gabriel Trujillo-Hernández, Iván Y. Alba-Corpus, Julio C. Rodríguez-Quiñonez, Jesús E. Miranda-Vega, Daniel Hernández-Balbuena, Fabian N. Murrieta-Rico, and Oleg Sergiyenko. 3d spatial measurement for model reconstruction: A review. *Measurement*, 207:112321, 2023.
- [17] Ivan Y. Alba Corpus, Wendy Flores-Fuentes, Oleg Sergiyenko, Julio C. Rodríguez-Quiñonez, Jesús E. Miranda-Vega, Wendy Garcia-González, and José A. Núñez-López. 3d data processing and entropy reduction for reconstruction from low-resolution spatial coordinate clouds in a technical vision system. *Entropy*, 26(8), 2024. Cited by: 0; All Open Access, Gold Open Access, Green Open Access.
- [18] Lihui Wang and Baozong Yuan. Curvature and density based feature point detection for point cloud data. In *IET 3rd International Conference on Wireless, Mobile and Multimedia Networks (ICWMNN 2010)*, pages 377–380, 2010.

- [19] Yawar Rehman, Hafiz M. Ameen Uddin, Taha Hasan Masood Siddique, Haris, Syed Riaz Un Nabi Jafri, and Anas Ahmed. Comparison of camera and laser scanner based 3d point cloud. In *2019 4th International Conference on Emerging Trends in Engineering, Sciences and Technology (ICEEST)*, pages 1–5, 2019.
- [20] B. Riveiro and R. Lindenbergh. *Laser Scanning: An Emerging Technology in Structural Engineering*. ISPRS Book Series. CRC Press, 2019.
- [21] A.Q. Bhatti, A. Wahab, and W. Sindi. An overview of 3d laser scanning techniques and application on digitization of historical structures. *Innovative Infrastructure Solutions*, 6(4), 2021. cited By 0.
- [22] H. Hu, Q. Chen, S. Feng, and C. Zuo. Microscopic fringe projection profilometry: A review. *Optics and Lasers in Engineering*, 135:106192, 2020.
- [23] G. De Angelis, A. De Angelis, A. Moschitta, and P. Carbone. Comparison of measurement models for 3d magnetic localization and tracking. *Sensors*, 17(11):2527, 2017.
- [24] S. Zhang. High-speed 3d shape measurement with structured light methods: A review. *Optics and Lasers in Engineering*, 106:119–131, 2018.
- [25] Guoqiang Fu, Arianna Menciassi, and Paolo Dario. Development of a low-cost active 3D triangulation laser scanner for indoor navigation of miniature mobile robots. *Robotics and Autonomous Systems*, 60:1317–1326, October 2012. Publisher: Elsevier.
- [26] S. Se, D.G. Lowe, and J.J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.
- [27] X. Xiang, H. Jiang, G. Zhang, Y. Yu, C. Li, X. Yang, D. Chen, and H. Bao. Mobile3dscanner: An online 3d scanner for high-quality object reconstruction with a mobile device. *IEEE Transactions on Visualization and Computer Graphics*, 27(11):4245–4255, 2021. cited By 0.
- [28] Z. Zhang, B. Rosa, O. Caravaca-Mora, P. Zanne, M.J. Gora, and F. Nageotte. Image-guided control of an endoscopic robot for oct path scanning. *IEEE Robotics and Automation Letters*, 6(3):5881–5888, 2021. cited By 0.
- [29] B. Sun, G. Zheng, and X. Zhang. Application of contact laser interferometry in precise displacement measurement. *Measurement*, 174:108959, 2021.
- [30] R. Koch, S. May, P. Murmann, and A. Nüchter. Identification of transparent and specular reflective material in laser scans to discriminate affected measurements for faultless robotic SLAM. *Robotics and Autonomous Systems*, 87:296–312, 2017.
- [31] J. Schlarp, E. Csencsics, and G. Schitter. Design and evaluation of an integrated scanning laser triangulation sensor. *Mechatronics*, 72:102453, 2020.

- [32] M. J. Castro-Toscano, J. C. Rodríguez-Quiñonez, O. Sergiyenko, W. Flores-Fuentes, L. R. Ramírez-Hernández, D. Hernández-Balbuena, L. Lindner, and R. Rascón. Novel sensing approaches for structural deformation monitoring and 3d measurements. *IEEE Sensors Journal*, 21(10):11318–11328, 2020.
- [33] M. Ivanov, O. Sergiyenko, V. Tyrsa, L. Lindner, W. Flores-Fuentes, J. C. Rodríguez-Quiñonez, W. Hernandez, and P. Mercorelli. Influence of data clouds fusion from 3d real-time vision system on robotic group dead reckoning in unknown terrain. *IEEE/CAA Journal of Automatica Sinica*, 7(2):368–385, 2020.
- [34] O. R. Real-Moreno, M. J. Castro-Toscano, J. C. Rodríguez-Quiñonez, O. Sergiyenko, D. Hernández-Balbuena, M. Rivas-López, W. Flores-Fuentes, and L. Lindner. Surface measurement techniques in machine vision: Operation, applications, and trends. In *Optoelectronics in Machine Vision-Based Theories and Applications*, pages 79–104. IGI Global, 2019.
- [35] L. Lindner, O. Sergiyenko, M. Rivas-López, D. Hernández-Balbuena, W. Flores-Fuentes, J. C. Rodríguez-Quiñonez, F. N. Murrieta-Rico, M. Ivanov, V. Tyrsa, and L. C. Básaca-Preciado. Exact laser beam positioning for measurement of vegetation vitality. *Industrial Robot: An International Journal*, 2017.
- [36] O. Real-Moreno, M. J. Castro-Toscano, J. C. Rodríguez-Quiñonez, D. Hernández-Balbuena, W. Flores-Fuentes, and M. Rivas-López. Implementing k-nearest neighbor algorithm on scanning aperture for accuracy improvement. In *2018 IEEE 44th Annual Conference of the Industrial Electronics Society (IECON)*, pages 3182–3186, 2018.
- [37] W. Flores-Fuentes, M. Rivas-López, O. Sergiyenko, F. F. González-Navarro, J. Rivera-Castillo, D. Hernández-Balbuena, and J. C. Rodríguez-Quiñonez. Combined application of power spectrum centroid and support vector machines for measurement improvement in optical scanning systems. *Signal Processing*, 98:37–51, 2014.
- [38] J. C. Rodríguez-Quiñonez, O. Sergiyenko, D. Hernández-Balbuena, M. Rivas-López, W. Flores-Fuentes, and L. Basaca-Preciado. Improve 3d laser scanner measurements accuracy using a ffbp neural network with widrow-hoff weight/bias learning function. *Opto-Electronics Review*, 22(4):224–235, 2014.
- [39] O. Y. Sergiyenko, M. V. Ivanov, V. Tyrsa, V. M. Kartashov, M. Rivas-López, D. Hernández-Balbuena, W. Flores-Fuentes, J. C. Rodríguez-Quiñonez, J. I. Nieto-Hipólito, W. Hernandez, et al. Data transferring model determination in robotic group. *Robotics and Autonomous Systems*, 83:251–260, 2016.
- [40] E. Aldao, H. González-Jorge, and J. Pérez. Metrological comparison of lidar and photogrammetric systems for deformation monitoring of aerospace parts. *Measurement*, 174:109037, 2021.

- [41] P. Saura-Gómez, Y. Spairani-Berrio, J. A. Huesca-Tortosa, S. Spairani-Berrio, and C. Rizo-Maestre. Advances in the restoration of buildings with lidar technology and 3d reconstruction: Forged and vaults of the refectory of santo domingo de orihuela (16th century). *Applied Sciences*, 11(18):8541, 2021.
- [42] I. Aicardi, F. Chiabrando, A. M. Lingua, and F. Noardo. Recent trends in cultural heritage 3d survey: The photogrammetric computer vision approach. *Journal of Cultural Heritage*, 32:257–266, 2018.
- [43] M. Azarafza, A. Ghazifard, H. Akgün, and E. Asghari-Kaljahi. Development of a 2d and 3d computational algorithm for discontinuity structural geometry identification by artificial intelligence based on image processing techniques. *Bulletin of Engineering Geology and the Environment*, 78(5):3371–3383, 2019.
- [44] M. Azarafza, Y. Nanekaran, H. Akgün, and Y. Mao. Application of an image processing-based algorithm for river-side granular sediment gradation distribution analysis. *Advances in Materials Research*, 10(3):229–244, 2021.
- [45] Z. Ma and S. Liu. A review of 3d reconstruction techniques in civil engineering and their applications. *Advanced Engineering Informatics*, 37:163–174, 2018.
- [46] J. Salvi, S. Fernández, T. Pribanić, and X. Llado. A state of the art in structured light patterns for surface profilometry. *Pattern Recognition*, 43(8):2666–2680, 2010.
- [47] O. Duran. *A Comparative Accuracy Analysis of 3D Model Programs: A Case Study of SfM and Photogrammetry Based Programs*. PhD thesis, California State University, Fresno, 2022.
- [48] Z. Bauer. *Monocular Depth Estimation: Datasets, Methods, and Applications*. PhD thesis, Universitat d’Alacant-Universidad de Alicante, 2021.
- [49] K. L. El-Ashmawy. Photogrammetric block adjustment without control points. *Geodesy and Cartography*, 44(1):6–13, 2018.
- [50] J. Iglhaut, C. Cabo, S. Puliti, L. Piermattei, J. O’Connor, and J. R. Rosette. Structure from motion photogrammetry in forestry: A review. *Current Forestry Reports*, 5(3):155–168, 2019.
- [51] Y.-M. Wei, L. Kang, B. Yang, and L.-D. Wu. Applications of structure from motion: A survey. *Journal of Zhejiang University Science C*, 14(7):486–494, 2013.
- [52] Lars Lindner and Oleg Sergiyenko. *Theoretical method to increase the speed of continuous mapping in a three-dimensional laser scanning system using servomotors control. [recurso electrónico]*. 2017., 2017.

- [53] Miguel Reyes-Garcia, Cesar Sepulveda-Valdez, Oleg Sergiyenko, Moisés Rivas-López, Julio Cesar Rodríguez-Quiñonez, Wendy Flores-Fuentes, Daniel Hernandez-Balbuena, Juan-Ivan Nieto-Hipolito, Fabian N. Murrieta-Rico, Lars Lindner, and Mykhailo Ivanov. Digital Control Theory Application and Signal Processing in a Laser Scanning System Applied for Mobile Robotics. In *Control and Signal Processing Applications for Mobile and Aerial Robotic Systems*. 2020. ISBN: 9781522599241 Pages: 215-265 Publisher: IGI Global.
- [54] Alan A. Author, Bill B. Author, and Cathy Author. Title of article. *Title of Journal*, 10(2):49–53, 2005.
- [55] X. Song, S. Dai, C. Lin, S. Zhan, and Y. Zhao. Lidar data enhancement via pseudo-lidar point cloud correction. *Journal of Image and Graphics*, 26(5):1157–1168, 2021. cited By 0.
- [56] A. Stulgis, L. Ambroziak, M. Kondratiuk, and A. Nikonowicz. Simple sense and avoid system for unmanned aerial vehicles. volume 2029, 2018. cited By 0.
- [57] Ji Zhang and Sanjiv Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41(2):401–416, February 2017.
- [58] L. Lindner. *Laser Scanners*. 2016.
- [59] F. Tian, Y. Gao, Z. Fang, J. Gu, and S. Yang. 3d reconstruction with auto-selected keyframes based on depth completion correction and pose fusion. *Journal of Visual Communication and Image Representation*, 79, 2021. cited By 0.
- [60] J. Fang, P. Jiang, Q. Wang, and G. Zhang. Experimental study on the precision evaluation and error correction of 3d laser scanners. *Modern Tunnelling Technology*, 58(3):169–175, 2021. cited By 0.
- [61] L. Winiwarter, K. Anders, and B. Höfle. M3c2-ep: Pushing the limits of 3d topographic point cloud change detection by error propagation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 178:240–258, 2021. cited By 0.
- [62] Lars Lindner, Oleg Sergiyenko, Julio C. Rodriguez-Quiñonez, Wendy Flores-Fuentes, and Ivan Giannoccaro, editors. *Examining Optoelectronics in Machine Vision and Applications in Industry 4.0*. Advances in Computational Intelligence and Robotics. IGI Global, 2021.
- [63] Miguel Reyes García, Lars Lindner, and Oleg Sergiyenko. *Método teórico para disminuir el error de posicionamiento en un sistema de barrido láser usando un controlador digital embebido y motores de corriente directa. [recurso electrónico]*. 2019.
- [64] H. P. Vinutha, B. Poornima, and B. M. Sagar. Detection of outliers using interquartile range technique from intrusion dataset. In Suresh Chandra Satapathy, Joao Manuel R.S. Tavares,

Vikrant Bhateja, and J. R. Mohanty, editors, *Information and Decision Sciences*, pages 511–518, Singapore, 2018. Springer Singapore.

- [65] R. Thompson. A note on restricted maximum likelihood estimation with an alternative outlier model. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):53–55, 1985.
- [66] Peter Filzmoser, C Reimann, and RG Garrett. *A multivariate outlier detection method*. Citeseer, 2004.
- [67] David Levin. The approximation power of moving least-squares. *Math. Comput.*, 67(224):1517–1532, October 1998.
- [68] N. Brightman and L. Fan. A brief overview of the current state, challenging issues and future directions of point cloud registration. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-3/W1-2022:17–23, 2022.
- [69] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.
- [70] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981.
- [71] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.
- [72] Wendy Flores-Fuentes, Eduardo Arellano-Vega, Oleg Sergiyenko, Iván Y. Alba-Corpus, Julio C. Rodríguez-Quiñonez, Moises J. Castro-Toscano, Félix F. González-Navarro, S. Vasavi, Jesús E. Miranda-Vega, Daniel Hernández-Balbuena, Fabián N. Murrieta-Rico, and Moisés Rivas-López. Surface color estimation in 3d spatial coordinate remote sensing by a technical vision system. *Optical and Quantum Electronics*, 56(3), 2024. Cited by: 1.
- [73] Wendy Flores-Fuentes, Moises Rivas-Lopez, Oleg Sergiyenko, Julio C Rodriguez-Quinonez, Daniel Hernández-Balbuena, and Javier Rivera-Castillo. Energy center detection in light scanning sensors for structural health monitoring accuracy enhancement. *IEEE Sensors Journal*, 14(7):2355–2361, 2014.
- [74] Moisés Rivas, Wendy Flores, Javier Rivera, Oleg Sergiyenko, Daniel Hernández-Balbuena, and Alejandro Sánchez-Bueno. A method and electronic device to detect the optoelectronic scanning signal energy centre. *Optoelectronics-Advanced Materials and Devices, Mexico*, 2013.
- [75] Phillip J Marlow, Karl R Gegenfurtner, and Barton L Anderson. The role of color in the perception of three-dimensional shape. *Current Biology*, 32(6):1387–1394, 2022.

- [76] Pei-Yong Zhang and Hui-Yan Wang. A framework for automatic time-domain characteristic parameters extraction of human pulse signals. *Eurasip Journal on advances in signal Processing*, 2008:1–9, 2007.
- [77] Da Ke, Xiang Wang, Kaizhu Huang, Haoyuan Wang, and Zhitao Huang. Minimum power adversarial attacks in communication signal modulation classification with deep learning. *Cognitive Computation*, pages 1–10, 2022.
- [78] Mupeng Li, Tianhua Xu, Shuang Wang, Wenxiu Hu, Junfeng Jiang, and Tiegeng Liu. Probe pulse design in brillouin optical time-domain reflectometry. *IET Optoelectronics*, 16(6):238–252, 2022.
- [79] Cemil Altın and Orhan Er. Comparison of different time and frequency domain feature extraction methods on elbow gesture’s emg. *European journal of interdisciplinary studies*, 2(3):35–44, 2016.
- [80] Kensuke Murakami, Mahfuzul Islam, and Hidetoshi Onodera. Cdf distance based statistical parameter extraction using nonlinear delay variation models. In *2021 IEEE 27th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 1–6. IEEE, 2021.
- [81] Alaa Alslaity and Rita Orji. Machine learning techniques for emotion detection and sentiment analysis: current state, challenges, and future directions. *Behaviour amp; Information Technology*, 43(1):139–164, December 2022.
- [82] Mikhail Ivanov, Oleg Sergiyenko, Vera Tyrsa, Lars Lindner, Miguel Reyes-García, Julio Cesar Rodríguez-Quiñonez, Wendy Flores-Fuentes, Jesús Elías Miranda-Vega, Moisés Rivas-López, and Daniel Hernández-Balbuena. Data exchange and task of navigation for robotic group. *Machine Vision and Navigation*, pages 389–430, 2020.
- [83] Miquel Massot and Ivan Corpus. ivancorpus/pointcloud_web_viewer: Basic release, April 2024.

Apéndice A: Publicaciones

Article

3D Data Processing and Entropy Reduction for Reconstruction from Low-Resolution Spatial Coordinate Clouds in a Technical Vision System

Ivan Y. Alba Corpus ^{1,2}, Wendy Flores-Fuentes ^{1,*}, Oleg Sergiyenko ³, Julio C. Rodríguez-Quíñonez ¹, Jesús E. Miranda-Vega ², Wendy García-González ¹ and José A. Núñez-López ³

¹ Facultad de Ingeniería, Universidad Autónoma de Baja California, Mexicali 21280, Mexico; ivan.alba@uabc.edu.mx (I.Y.A.C.); julio.rodriguez81@uabc.edu.mx (J.C.R.-Q.); wendy.garcia26@uabc.edu.mx (W.G.-G.)

² IT de Mexicali, Tecnológico Nacional de México, Mexicali 21376, Mexico; elias.miranda@itmexicali.edu.mx

³ Instituto de Ingeniería, Universidad Autónoma de Baja California, Mexicali 21100, Mexico; srnkn@uabc.edu.mx (O.S.); jose.nunez10@uabc.edu.mx (J.A.N.-L.)

* Correspondence: flores.wendy@uabc.edu.mx

Abstract: This paper proposes an advancement in the application of a Technical Vision System (TVS), which integrates a laser scanning mechanism with a single light sensor to measure 3D spatial coordinates. In this application, the system is used to scan and digitalize objects using a rotating table to explore the potential of the system for 3D scanning at reduced resolutions. The experiments undertaken searched for optimal scanning windows and used statistical data filtering techniques and regression models to find a method to generate a 3D scan that was still recognizable with the least amount of 3D points, balancing the number of points scanned and time, while at the same time reducing effects caused by the particularities of the TVS, such as noise and entropy in the form of natural distortion in the resulting scans. The evaluation of the experimentation results uses 3D point registration methods, joining multiple faces from the original volume scanned by the TVS and aligning it to the ground truth model point clouds, which are based on a commercial 3D camera to verify that the reconstructed 3D model retains substantial detail from the original object. This research finds it is possible to reconstruct sufficiently detailed 3D models obtained from the TVS, which contain coarsely scanned data or scans that initially lack high definition or are too noisy.

Keywords: 3D reconstruction; spatial coordinate cloud; cloud registration



Citation: Alba Corpus, I.Y.; Flores-Fuentes, W.; Sergiyenko, O.; Rodríguez-Quíñonez, J.C.; Miranda-Vega, J.E.; García-González, W.; Núñez-López, J.A. 3D Data Processing and Entropy Reduction for Reconstruction from Low-Resolution Spatial Coordinate Clouds in a Technical Vision System. *Entropy* **2024**, *26*, 646. <https://doi.org/10.3390/e26080646>

Academic Editor: Wei Li

Received: 20 May 2024

Revised: 17 July 2024

Accepted: 18 July 2024

Published: 30 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The progress of 3D scanning technology has revolutionized the way we capture and replicate the physical world in digital form. From documentation of historical artifacts [1,2] to the precision required in industrial design, the applications of this technology are vast and varied. Of these systems, the versatility of laser scanning systems as remote sensing technology is the reason they are used across a wide array of fields. These systems are integral to various sectors, including healthcare, geospatial surveying, additive manufacturing, the mining industry, and reverse engineering, to highlight just a few. The application of laser scanning in the medical field enables precise mapping and modeling of physical forms, while in geospatial surveying, it facilitates the detailed assessment of land topographies and architectural structures. In the field of additive manufacturing, laser scanning contributes to the accuracy and fidelity of 3D-printed objects, and in mining, it aids in the volumetric analysis and planning of extraction sites. Reverse engineering processes benefit significantly from laser scanning by providing a means to recreate models of existing physical objects with high accuracy.

Multiple studies illustrate the influence of laser scanning technologies in both research and practical applications. For instance, ref. [3] explores the measurement of vegetation



Surface color estimation in 3D spatial coordinate remote sensing by a technical vision system

Wendy Flores-Fuentes¹ · Eduardo Arellano-Vega¹ · Oleg Sergiyenko² · Iván Y. Alba-Corpus¹ · Julio C. Rodríguez-Quiñonez¹ · Moises J. Castro-Toscano¹ · Félix F. González-Navarro² · S. Vasavi³ · Jesús E. Miranda-Vega⁴ · Daniel Hernández-Balbuena¹ · Fabián N. Murrieta-Rico⁵ · Moisés Rivas-López⁵

Received: 17 July 2023 / Accepted: 25 October 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

This paper proposes to enhance the capabilities of a Technical Vision System (TVS) based on optical scanning and dynamic triangulation for 3D spatial coordinates measurements to determine and add to the record of its measurements the color of the objects coordinates scanned. Optoelectronics signals output from the TVS scanning aperture contain a huge amount of data that cannot be friendly or obvious to human understanding and interpretation. However, time domain characteristic parameters can be extracted from the optoelectronics signals, which can describe aspects of the optoelectronic signal and the 3D spatial coordinate to whom it belongs; that is, it is specific and punctual information from the area of the object under scanning. In this theoretical and experimental evaluation, the parameters obtained from optoelectronics signals have been used as feature inputs for supervised learning and as an approach to implement artificial intelligence to the TVS signal processing by a model able to predict in real time the color of the 3D spatial coordinates measured.

Keywords Remote sensing · Color information extraction · 3D spatial coordinates · Automatic measurements · Technical vision systems · Artificial intelligence

1 Introduction

Remote sensing is the acquisition of physical data from an object without touch or contact; it is closely related to photogrammetry, 3D spatial information systems, and computer vision. Most of all, 3D coordinates measurement by photogrammetry and by 3D spatial information systems have been used for many decades in a wide field of applications at different scales. One example is the Earth 3D reconstruction based on satellite imagery for hydrology and weather conditions evaluation (Sabitha et al. 2023),

Wendy Flores-Fuentes, Eduardo Arellano-Vega, Oleg Sergiyenko, Iván Y. Alba-Corpus, Julio C. Rodríguez-Quiñonez, Moises J. Castro-Toscano, Félix F. González-Navarro, S. Vasavi, Jesús E. Miranda-Vega, Daniel Hernández-Balbuena, Fabián N. Murrieta-Rico and Moisés Rivas-López are contributed equally to this work.

Extended author information available on the last page of the article

Published online: 25 January 2024



Apéndice A 2: Surface color estimation in 3D spatial coordinate remote sensing by a technical vision system



Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

3D spatial measurement for model reconstruction: A review

Wendy Flores-Fuentes^{a,*}, Gabriel Trujillo-Hernández^a, Iván Y. Alba-Corpus^a,
 Julio C. Rodríguez-Quirón^a, Jesús E. Mirada-Vega^d, Daniel Hernández-Balbuena^a,
 Fabian N. Murrieta-Rico^c, Oleg Sergiyenko^{b,*}

^a Facultad de Ingeniería, Universidad Autónoma de Baja California, Mexicali, Baja California, México.

^b Instituto de Ingeniería, Universidad Autónoma de Baja California, Mexicali, Baja California, México.

^c Universidad Politécnica de Baja California, Mexicali, Baja California, México.

^d Tecnológico Nacional de México/IT de Mexicali, Mexicali, Baja California, México.

ARTICLE INFO

Keywords:

3D spatial coordinate measurements
 3D model reconstruction
 3D point cloud data
 Cyber systems
 Machine vision

ABSTRACT

The measurement of 3D spatial coordinates for model reconstruction through artificial machine vision systems based on optical sensors and the corresponding signal processing associated with algorithms is a powerful module for cyber systems. It provides an efficient, functional, and intelligent vision and data information of the objects and scenes under observation for decisions, as well as for remote environment interactivity and autonomous robot systems actuation. Over the past 20 years, the artificial machine vision has benefited from emerging technology and a promising huge potential is peeking out, but also technical difficulties achieving customized and true commercial applications. This paper reviews the research progress, trends, and future research directions; the state-of-the-art of topics related to the 3D spatial measurement for model reconstruction. It classifies the technology by its fundamental principles and applications, to construct an outlook about its advantages, disadvantages, and challenges.

1. Introduction

Over the past 20 years, the artificial machine vision has benefited from emerging technology [1]. Sensors and actuators sophisticated manufacturing improvement, electronic and computational advances for the application of artificial intelligence in systems-on-chip and embedded systems [2], combined with the communication connectivity, cloud processing, and storage solutions provided by the Internet of Things (IoT) and the spatial web, which have brought challenges for the services of Industry 4.0 and Society 5.0 [3]. Machine vision is based on the acquisition of spatial coordinate's measurement for a wide range of technologies for multiples fields of applications [4], such as terrestrial robot navigation [5], aerial vehicle navigation [6], medical scanning [7], structural health monitoring [8], and industrial measurements applications [9], moreover, the growing adoption of machine vision technologies plays an important role in emerging technologies such as smart manufacturing [10], and even in those technologies directly applied to a personal device such as computers, and mobiles. Machine vision methods can be applied in any applications that perform tasks which require seen; it is the eyes of industrial process handle by robotic, so machine vision can be considered the eyes of cyber systems; it endows them with a human visual perception to

improve their intelligence behavior, by this way it enables robotics and automated systems to identify, locate, measure, classify, forecast, and even reproduce the 3D model of objects. That is how it provides enhanced attributes than the ones that humans possess, to accomplish tasks previously done by humans in different fields, especially in the industry, but also in service fields such as medicine, and the commercial sector. It is a visualizing tool to monitoring various phenomenon that occur during industrial process, rescue [11], vigilance [12], mapping [13], dangerous objects/subjects detection [14], and other areas where machine control based on vision plays an important role. The electronics sensory and computer vision have guided the machine vision to the tendency of an enhanced duplication of the abilities of human vision by electronically perceiving and understanding an image for high dimensional data, optimizing the data storage requirement, and the time processing due to the complexity of algorithms to extract important patterns and trends to understand what data says. The model of objects can be reconstructed by artificial machine vision systems as well as our brain reconstructs the images of objects and scenes observed by our eyes. Machine vision systems can perform measurements to image the 3D shape. Even so, with enhanced properties than humans to estimate its coordinates, dimensions, location,

* Corresponding authors.

E-mail addresses: flores.wendy@uabc.edu.mx (W. Flores-Fuentes), srgnk@uabc.edu.mx (O. Sergiyenko).

<https://doi.org/10.1016/j.measurement.2022.112321>

Received 18 June 2022; Received in revised form 18 November 2022; Accepted 4 December 2022

Available online 30 December 2022

0263-2241/© 2022 Elsevier Ltd. All rights reserved.



A structural health monitoring method proposal based on optical scanning and computational models

Wendy Flores-Fuentes¹ , Ivan Yeniseysk Alba-Corpus¹,
Oleg Sergiyenko² and Julio C. Rodríguez-Quiñonez¹

Abstract

This article proposes a method for continuous bridge displacement monitoring combining the dynamic triangulation scanning and load estimation by vehicle image recognition. The vehicle–bridge interaction is a non-stationary dynamic process parameter of high relevance to understanding the static instability behavior of bridges. The knowledge of the load on a bridge in the specific time when its structural spatial coordinates are measured allows correlating the bridge displacement with the effect of vehicle–bridge interaction. The evaluation of such correlation is mandatory in order to verify if the observed bridge displacement is due to the nature of its operation or due to it is presenting structural damage. The proposed method is continuous structural health monitoring method, based on the combination of three approaches evaluated at laboratory environment: (1) a three-dimensional optical scanning system for displacement measurement, (2) a load measurement system for vehicle–bridge interaction assessment, and (3) a two-measurement systems data correlation; to be implemented in bridges at a real environment to collect their historical behavior. Overall, for each approach, the measurement systems' principles, the laboratory experimental methodology followed, and results obtained are presented.

Keywords

Bridge, displacement, structural health monitoring, vehicle–bridge interaction, optical scanning, image processing, data correlation

Date received: 28 August 2021; accepted: 31 May 2022

Handling Editor: Francesc Pozo

Introduction

Continuous structural health monitoring (SHM) of bridges is indispensable to seeking information about their health. This is essential to understand their behaviors, to figure out if they are operating under healthy conditions, are overstressed, and/or are under a low-/medium-/or high-degradation process.

As the cities grow and modernize, the vehicle traffic increases, and the design of bridges is every time more sophisticated, coupled with this, the smart cities, sustainability concepts and intelligent technology emergence, guide to the substitution of conventional SHM methods to non-contact, continuous SHM methods.

By other side, SHM has always been a challenging task. A wide range of technologies have emerged, given place to different SHM systems, according to the structure characteristics under monitoring; the different

¹Facultad de Ingeniería, Universidad Autónoma de Baja California, Mexicali, Mexico

²Instituto de Ingeniería, Universidad Autónoma de Baja California, Mexicali, Mexico

Corresponding author:

Julio C Rodríguez-Quiñonez, Facultad de Ingeniería, Universidad Autónoma de Baja California, Blvd. Benito Juárez S/N, 21280 Mexicali, Baja California, Mexico.
Email: julio.rodriguez81@uabc.edu.mx



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (<https://creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

Apéndice A 4: A structural health monitoring method proposal based on optical scanning and computational models

Detection of Atypical Data in Point Cloud of Technical Vision System using Digital Filtering

Ivan Y. Alba-Corpus
 Wendy Flores-Fuentes
 Julio C. Rodríguez-Quiñonez
 Daniel Hernández-Balbuena
Facultad de Ingeniería
Universidad Autónoma de Baja California
 Mexicali, B.C., México
 ivan.alba@uabc.edu.mx

Félix F. González-Navarro
 Oleg Sergiyenko
 Ruben Alaniz-Plata
Instituto de Ingeniería
Universidad Autónoma de Baja California
 Mexicali, B.C., México
 fernando.gonzalez@uabc.edu.mx

Abstract—Most non-contact methods and systems for 3D spatial coordinate measurement are based on optical sensors and signal processing. Object surfaces are discretely detected to obtain their coordinates and generate a point cloud that can be processed to reconstruct the object's shape and dimensions. Optical sensor-based technologies are the most widely used because they are non-destructive and do not need to touch objects. One of their best properties is their fast measurement acquisition rate. However, one of their disadvantages is the presence of atypical values in their measurements due to their sensitivity to optical ambient noise. Filtering atypical values from point clouds allow the generation and reconstruction of mesh and geometrical models of objects with appearance and dimensions similar to reality. This work is focused on the point cloud post-processing of 3D spatial coordinates measurements obtained from a technical vision system to eliminate inaccuracies in the reconstruction models.

Index Terms—Technical vision system, 3D point cloud, Atypical Values.

I. INTRODUCCIÓN

La mayoría de los métodos y sistemas sin contacto para la medición de coordenadas espaciales en 3D se basan en sensores ópticos y procesamiento de señales. Las superficies de los objetos se detectan discretamente para obtener sus coordenadas y generar una nube de puntos que se puede procesar para reconstruir la forma y las dimensiones del objeto [1].

Las tecnologías basadas en sensores ópticos son las más utilizadas porque no son destructivas y no necesitan tocar objetos. Una de sus mejores propiedades es su rápida tasa de adquisición de medidas. Sin embargo, una de sus desventajas es la presencia de valores atípicos en sus medidas debido a su sensibilidad al ruido ambiental óptico, y a la naturaleza de su composición mecatrónica.

Este trabajo se centra en el posprocesamiento de nubes de puntos de medidas de coordenadas espaciales 3D obtenidas a partir de un sistema de visión técnica para eliminar imprecisiones en los modelos de reconstrucción.

La detección de datos atípicos en nubes de puntos de sistemas de visión técnica usando filtros digitales es de gran importancia para muchas aplicaciones de distintos campos del conocimiento como lo es la bioingeniería, la ingeniería inversa, la ingeniería civil y arqueológica, así como para la electrónica y la automatización industrial. Debido a su flexibilidad, las nubes de puntos 3D se utilizan en múltiples aplicaciones de los distintos campos, como lo son la inspección y mapeo de estructuras y terrenos, reconocimiento de objetos, y navegación autónoma, por mencionar algunas.

El filtrado de valores atípicos de nubes de puntos permite la generación y reconstrucción de mallas y modelos geométricos de objetos con apariencia y dimensiones similares a la realidad. Por ejemplo en [2] se examinan diferentes métodos de filtrado en nubes de puntos. En [3] se desarrollo un algoritmo basado en segmentación del factor de variación de la superficie, utilizando diferentes métodos de filtrado en diferentes secciones de la nube de puntos para obtener mejores resultados en diferentes regiones de la nube de puntos. Otro ejemplo de filtrado es [4] donde se utiliza IA para generar filtros adaptivos tomando en cuenta el tamaño y forma de la región de la nube de puntos que se esta procesando.

Contrario a otras representaciones como los sólidos o mallas poligonales, las nubes de puntos no necesitan almacenar ningún tipo de dato adicional para la consistencia de los datos, los cuales son obtenidos mediante escáneres 3D. Estos escáneres se pueden dividir principalmente en escáneres basados en cámaras y basados en láser. De estos, los escáneres basados en cámaras son una buena fuente de datos para obtener nubes de puntos, los resultados obtenidos de estos suelen ser afectados por más datos atípicos [5] y también pueden tener problemas operando en la oscuridad o en condiciones de baja visibilidad como la lluvia o la niebla [6], mientras que los escáneres basados en láser pueden ser comparativamente lentos y tener problemas para detectar la reflexión de ciertos tipos de superficies.

Obstacle Coordinates Transformation from TVS Body-Frame to AGV Navigation-Frame

Xavier de J. Garcia-Gutierrez
Ruben Alaniz-Plata
Gabriel Trujillo-Hernández
Ivan Y.Alba-Corpus
Facultad de Ingeniería, UABC
Méxicali, B. C. México
{garciax, ruben.alaniz, gabriel.trujillo,
ivan.alba}@uabc.edu.mx

Paolo Mercorelli
Institute of Product and Process Innovation
Leuphana University of Lüneburg
Lüneburg, Germany
paolo.mercorelli@leuphana.de

Wendy Flores-Fuentes
Julio C. Rodríguez-Quiñonez
Daniel Hernández-Balbuena
Facultad de Ingeniería, UABC
Méxicali, B. C. México
{flores.wendy, julio.rodriguez81,
dhernan}@uabc.edu.mx

Jesús E. Miranda-Vega
Department of Computer Systems,
Tecnológico Nacional de México/IT de
Méxicali
Méxicali, Baja California, México
elias.miranda@itmexicali.edu.mx

Oleg Sergiyenko
Félix F. González-Navarro
Instituto de Ingeniería, UABC
Méxicali, B. C. México
{srgnk, fernando.gonzalez}@uabc.edu.mx

Fabian N. Murrieta-Rico
Universidad Politécnica de Baja
California
Méxicali, Baja California, México
fabian.murrieta@uabc.edu.mx

Abstract— This paper describes a mathematical method for obstacle coordinates transformation from a Technical Vision System (TVS) Body-Frame to an Automated Guided Vehicle (AGV) Navigation-Frame. The method is for AGVs equipped with a TVS to scan the scene in front of its travel direction in order to detect the presence of objects and measure their coordinates in the Navigation-Frame to avoid them. The proposed coordinate's transformation method is based on a AGVs kinematic model that includes the calculation of the AGVs load variation influence by the change of the center of gravity of the AGV traveling in an ideal floor conditions.

Keywords—automated guided vehicle, navigation, 3D scanning, and technical vision system

I. Introduction

The AGVs must navigate conflict-free and using optimized paths; should execute their task starting from a predefined initial pose, moving to the desired position. Efficient algorithms are able to avoid obstacles, solve traffic jams and collisions. In the last years, several algorithms for AGV routing have been classified into two categories, static routing, and dynamic routing. In the category of static routing, the route from position i to position j is previously determined and then used if the AGV has to be transported from position i to position j . Thus, a simple assumption is to choose the route with the shortest distance from position i to position j . However, these static algorithms cannot be adapted to changes in the AGV system and traffic conditions. By other side, in the dynamic routing, the information necessary to determine efficient routes are analyzed by intelligent algorithms to make decisions from the various possible routes that can be chosen between position i and j , taking into account important features in the routing, such as path maneuvers, time curves, load and unload operations, speed, dispatching rules, and dispatching quantity of AGVs [1], [2]. This paper proposes the use of a technical vision system for obstacles detection on AGVs navigation, and describes a method for obstacle coordinates transformation from TVS body-frame to AGV navigation-frame as information for dynamic routing. The application of the proposed method can be bound in recent works about obstacle avoidance using vision systems such as [3], [4], [5].

II. AGVs CONFIGURATION, SENSING AND MEASURING

The steering system of an AGV as the function to control the direction of movement; it allows following the desired path. The steering system is generally integrated by linkages, rods, pivots, and gears, some of the systems types are the Ackermann steering, Bell-crank steering, and the Rack and pinion steering, this last one is the most common and composed of a steering wheel, a pinion, a rack & pinion case, and a rack. Some of the most common steering wheel system configurations are skid steer 4 wheels, differential, tricycle wheeled, quad, omnidirectional, and hybrid [6]. Regarding the control architecture which define the behavior control of robots; it is characterized by the way it uses the information collected to analyze time and spatial distribution patterns for decision making, some examples are the deliberative, reactive, behavior based, as well as hybrid systems [7]. By other side the sensors make possible to observe the working environment or guiding the AGV to follow a specified path. Measurements by sensors allow knowing the deviation between the AGV and the path defined in the working environment for further decisions. These sensors can be used on fixed and free path guidance, also in a fusion of them, the most used sensors for the AGVs position and navigation are cameras, laser scanners (reflector-type laser-guided to detect landmarks, triangulation based on reflectors positioned in the environment, dynamic triangulation based on a laser emitter and optoelectronic scanning aperture, Lidar, and laser rangefinder), inertial measurement units, encoders, magnetic sensors, optical sensors systems for measurement base on infrared, signal strength indicators, electro magnetic inductive sensors, and global position systems. With the listed sensors for AGVs have been mentioned position references related with some of those sensors to follow predetermined paths marked with physical and virtual guide-paths. Other sensors are also used for sensing techniques for localization and mapping, without physical guide-paths like a dead reckoning, beacon systems, inertial navigation systems, and computer vision. Some navigation methods provide simultaneous localization and mapping (SLAM), which provide information about the absolute coordinates of the AGV with high accuracy. Others only provide the deviation with respect to a predefined trajectory,

Evaluation of Photosensors for Laser Scanning Range Extension

Ivan Yeniseysk Alba Corpus, Lars Lindner, Oleg Sergiyenko*
Julio C. Rodríguez-Quiñonez, Wendy Flores-Fuentes**
Fabian N. Murrieta-Rico***

* *Instituto de Ingeniería, UABC, Mexicali, México (e-mail: lindner.lars@uabc.edu.mx).*

** *Facultad de Ingeniería, UABC, Mexicali, México*

*** *Departamento de Ingeniería Mecatrónica, UPBC, Mexicali, México*

Abstract: Laser scanners are widely used in industry and science, where measurement of 3D coordinates of any object under observation is required. A novel laser scanner called Technical Vision System (TVS), which uses the Dynamic Triangulation measurement method, has proven to present a fast and reliable laser scanning system. The measurement range of the current TVS prototype No.3 shall be extended. For this purpose and in the present paper, two types of photosensors and two types of amplifier circuits for the TVS Scanning Aperture are evaluated and compared to each other. It has been found that using a phototransistor together with a transimpedance amplifier represents a better solution than using a photodiode or a non-inverting amplifier. This configuration extends the field of application of the TVS to areas that were previously inaccessible.

Keywords: Laser scanning, Technical vision system, Dynamic triangulation, Transimpedance amplifier, Photosensor characterization

1. INTRODUCTION

Laser scanners as sensors in image processing play an important role when it comes to measuring three-dimensional coordinates of scanning objects. These coordinates can then be used to map terrain, obstacle avoidance routines and surveying enclosed spaces such as pipes or ducts. The measurement of 3D coordinates is mainly realized by optical methods in a large amount of applications. While optical methods such as camera-based vision systems are a good source of datasets for point clouds, they are prone to outliers (Rehman et al., 2019) and can struggle when operating in darkness or in weather conditions such as rain or fog (Zywanowski et al., 2020). 3D mapping (Zhang and Singh, 2017), autonomous robots (Fu et al., 2012), automatic inspections (Mavrinac et al., 2015), coat thickness measurement (Shao et al., 2019), object existence and location detection (Aloni and Yitzhaky, 2014), show clear examples of accurate positioning technologies. A novel Technical Vision System (TVS, prototype No.3) (Lindner et al., 2016d) was developed at the Engineering Institute of UABC, which represents a laser scanning system capable of measuring 3D coordinates in the TVS field-of-view (FOV), to determine the information of scanned surfaces for scientific and industrial application. The TVS uses the Dynamic Triangulation measurement principle and trigonometric

functions, to measure distances and angles of any object under observation (Lindner et al., 2016a,c,b, 2017; Lindner, 2017). The applications of the TVS range from mobile robot navigation (Reyes-García et al., 2020), through static and dynamic structural health monitoring (Flores-Fuentes et al., 2014; Rivera-Castillo et al., 2016; Castro-Toscano et al., 2020), to medical applications (Rodríguez-Quiñonez et al., 2013; Rodríguez-Quiñonez et al., 2014).

The Scanning Aperture (SA) is one essential technical component of this TVS, which represents the receiver unit and which processes and detects the reflected laser beam angle. The TVS currently uses a low power green laser (518 nm), which is currently difficult to detect, using the available sensor installed in the system which has a wavelength peak sensitivity of 850nm, making it not optimal for usage in this system. Coupled with the low availability of sensors with high sensitivity to the laser beam, as well as the interference in the sensors caused by natural and artificial light going from 400nm to 700nm, which is in the spectrum of sensibility for the new sensors, makes it necessary to develop a filter, amplifier and algorithms to condition the laser beam signal received by the TVS SA (Corpus et al., 2020). The measurements taken with the current available sensors also show that the output current is too low to be measured and that these sensors only output a specific amount of voltage and

Point Cloud Optimization Employing Multisensory Vision



Cesar Sepulveda-Valdez, Ruben Alaniz-Plata, José A. Núñez-López, Ivan Yeniseysk Alba-Corpus, Humberto Andrade-Collazo, Wendy Flores-Fuentes, Julio C. Rodríguez-Quiñonez, Paolo Mercorelli, Vera Tyrsa, Santiago Camacho-López, and Oleg Sergiyenko

1 Introduction

Nowadays, computer vision has become an important and popular part of the research field presenting challenging tasks for authors to address. One of the primary focuses of researchers is the tools utilized to perform different types of information acquisition in a wide range of applications related to 3D reconstruction. Typically the selected tool is a laser scanner, a camera setup with distinct configurations or an arrangement of sensorial elements that allow systems to acquire environment information. Laser scanners are used in multiple practical applications. Whether it is for path planning in mobile robots, single-point distance measurements, surface recognition, object detection, medicine, construction, semiconductor fabrication, or products for household applications, the type of laser scanner can vary depending on the application objective and precision requirements. The type of laser scanner can vary depending on the application objective and precision requirements. In addition,

C. Sepulveda-Valdez (✉) · R. Alaniz-Plata · J. A. Núñez-López · I. Y. Alba-Corpus · H. Andrade-Collazo · W. Flores-Fuentes · J. C. Rodríguez-Quiñonez · V. Tyrsa · O. Sergiyenko
Universidad Autonoma de Baja California, Mexicali, BC, Mexico
e-mail: cesar.sepulveda@uabc.edu.mx; ruben.alaniz@uabc.edu.mx; jose.nunez97@uabc.edu.mx;
ivan.alba@uabc.edu.mx; humberto.andrade@uabc.edu.mx; flores.wendy@uabc.edu.mx;
julio.rodriguez81@uabc.edu.mx; vtyrsa@uabc.edu.mx; srgnk@uabc.edu.mx

P. Mercorelli
Institute for Production Technology and Systems, Leuphana University of Lueneburg, Lueneburg, Germany
e-mail: paolo.mercorelli@leuphana.de

S. Camacho-López
Departamento de Óptica, Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, BC, Mexico
e-mail: camachol@cicese.mx

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
J. C. Rodríguez-Quiñonez et al. (eds.), *Scanning Technologies for Autonomous Systems*, https://doi.org/10.1007/978-3-031-59531-8_10

275

Derechos de autor

CERTIFICADO

Registro Público del Derecho de Autor

Para los efectos de los artículos 13, 162, 163 fracción I, 164 fracción I, y demás relativos de la Ley Federal del Derecho de Autor, se hace constar que la **OBRA** cuyas especificaciones aparecen a continuación, ha quedado inscrita en el Registro Público del Derecho de Autor, con los siguientes datos:

AUTORES: ALBA CORPUS IVAN YENISEYSK
ARELLANO VEGA EDUARDO
FLORES FUENTES WENDY
GONZALEZ NAVARRO FELIX FERNANDO
HERNANDEZ BALBUENA DANIEL
MIRANDA VEGA JESUS ELIAS
MURRIETA RICO FABIAN NATANAEL
RODRIGUEZ QUIÑONEZ JULIO CESAR
SERGIYENKO OLEG

TÍTULO: CONJUNTO DE DATOS DE SEÑALES OPTOELECTRONICAS

RAMA: COMPILACION DE DATOS (BASE DE DATOS)

TITULAR: UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA (CON FUNDAMENTO EN EL ARTICULO 83 DE LA L.F.D.A.)

Con fundamento en lo establecido por el artículo 107 de la Ley Federal del Derecho de Autor, las bases de datos o de otros materiales legibles por medio de máquinas o en otra forma, que por razones de selección y disposición de su contenido constituyan creaciones intelectuales, quedarán protegidas como compilaciones. Dicha protección no se extenderá a los datos y materiales en sí mismos.

Con fundamento en lo establecido por el artículo 168 de la Ley Federal del Derecho de Autor, las inscripciones en el registro establecen la presunción de ser ciertos los hechos y actos que en ellas consten, salvo prueba en contrario. Toda inscripción deja a salvo los derechos de terceros. Si surge controversia, los efectos de la inscripción quedarán suspendidos en tanto se pronuncie resolución firme por autoridad competente.

Con fundamento en los artículos 2, 208, 209 fracción III y 211 de la Ley Federal del Derecho de Autor; artículos 64, 103 fracción IV y 104 del Reglamento de la Ley Federal del Derecho de Autor; y artículos 1, 3 fracción I, 4, 8 fracción I y 9 del Reglamento Interior del Instituto Nacional del Derecho de Autor, se expide el presente certificado.

Número de Registro: 03-2023-020111081800-01

Ciudad de México, a 07 de febrero de 2023

EL DIRECTOR DEL REGISTRO PÚBLICO DEL DERECHO DE AUTOR

JESÚS PARETS GÓMEZ
SECRETARÍA DE CULTURA
INSTITUTO NACIONAL DEL
DERECHO DE AUTOR
DIRECCIÓN DEL REGISTRO
PÚBLICO
DEL DERECHO DE AUTOR



CULTURA
SECRETARÍA DE CULTURA



INDAUTOR
INSTITUTO NACIONAL DEL DERECHO DE AUTOR

Apéndice A 9: Conjunto de datos de señales optoelectronicas.

CERTIFICADO

Registro Público del Derecho de Autor

Para los efectos de los artículos 13, 162, 163 fracción I, 164 fracción I, y demás relativos de la Ley Federal del Derecho de Autor, se hace constar que la **OBRA** cuyas especificaciones aparecen a continuación, ha quedado inscrita en el Registro Público del Derecho de Autor, con los siguientes datos:

AUTORES: ALBA CORPUS IVAN YENISEYSK
ARELLANO VEGA EDUARDO
FLORES FUENTES WENDY
GONZALEZ NAVARRO FELIX FERNANDO
HERNANDEZ BALBUENA DANIEL
MIRANDA VEGA JESUS ELIAS
MURRIETA RICO FABIAN NATANAEL
RODRIGUEZ QUIÑONEZ JULIO CESAR
SERGIYENKO OLEG

TÍTULO: EXTRACCION DE CARACTERISTICAS EN EL DOMINIO DEL TIEMPO DE SEÑALES OPTOELECTRONICAS DE UN SISTEMA DE VISION TECNICA

RAMA: PROGRAMAS DE COMPUTACION

TITULAR: UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA (CON FUNDAMENTO EN EL ARTICULO 83 DE LA L.F.D.A.)

Con fundamento en lo establecido por el artículo 168 de la Ley Federal del Derecho de Autor, las inscripciones en el registro establecen la presunción de ser ciertos los hechos y actos que en ellas consten, salvo prueba en contrario. Toda inscripción deja a salvo los derechos de terceros. Si surge controversia, los efectos de la inscripción quedarán suspendidos en tanto se pronuncie resolución firme por autoridad competente.

Con fundamento en los artículos 2, 208, 209 fracción III y 211 de la Ley Federal del Derecho de Autor; artículos 64, 103 fracción IV y 104 del Reglamento de la Ley Federal del Derecho de Autor; y artículos 1, 3 fracción I, 4, 8 fracción I y 9 del Reglamento Interior de Instituto Nacional del Derecho de Autor, se expide el presente certificado.

Número de Registro: 03-2023-020111075200-01

Ciudad de México, a 07 de febrero de 2023

EL DIRECTOR DEL REGISTRO PÚBLICO DEL DERECHO DE AUTOR

JESÚS PARETS GÓMEZ



CULTURA
SECRETARÍA DE CULTURA



INDAUTOR
INSTITUTO NACIONAL DEL DERECHO DE AUTOR

Apéndice A 10: Extracción de características en el dominio del tiempo de señales opto-electronicas de un sistema de visión técnica.

CERTIFICADO

Registro Público del Derecho de Autor

Para los efectos de los artículos 13, 162, 163 fracción I, 164 fracción I, y demás relativos de la Ley Federal del Derecho de Autor, se hace constar que la **OBRA** cuyas especificaciones aparecen a continuación, ha quedado inscrita en el Registro Público del Derecho de Autor, con los siguientes datos:

AUTORES: ALBA CORPUS IVAN YENISEYSK
CASTRO TOSCANO MOISES JESUS
FLORES FUENTES WENDY
LINDNER LARS
MURRIETA RICO FABIAN NATANAEL
RODRIGUEZ QUIÑONEZ JULIO CESAR
SERGIYENKO OLEG

TÍTULO: HERRAMIENTA DE ESCANEAMIENTO Y VISUALIZACIÓN DE PUNTOS 3D

RAMA: PROGRAMAS DE COMPUTACION

TITULAR: UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA (CON FUNDAMENTO EN EL ARTÍCULO 83 DE LA L.F.D.A.)

Con fundamento en lo establecido por el artículo 168 de la Ley Federal del Derecho de Autor, las inscripciones en el registro establecen la presunción de ser ciertos los hechos y actos que en ellas consten, salvo prueba en contrario. Toda inscripción deja a salvo los derechos de terceros. Si surge controversia, los efectos de la inscripción quedarán suspendidos en tanto se pronuncie resolución firme por autoridad competente.

Con fundamento en los artículos 2, 208, 209 fracción III y 211 de la Ley Federal del Derecho de Autor; artículos 64, 103 fracción IV y 104 del Reglamento de la Ley Federal del Derecho de Autor; y artículos 1, 3 fracción I, 4, 8 fracción I y 9 del Reglamento Interior de Instituto Nacional del Derecho de Autor, se expide el presente certificado.

Número de Registro: 03-2021-051912354100-01

Ciudad de México, a 18 de junio de 2021

EL DIRECTOR DEL REGISTRO PÚBLICO DEL DERECHO DE AUTOR

JESUS PARETS GOMEZ



CULTURA
SECRETARÍA DE CULTURA



INDAUTOR
INSTITUTO NACIONAL DEL DERECHO DE AUTOR

Apéndice A 11: Herramienta de escaneo y visualización de datos 3D

Apéndice B: Códigos Computacionales

Programa de escaneo en Teensy

```
1  #define izquierda 2
2  #define derecha 3
3  #define arriba 4
4  #define abajo 5
5  #define pin_cH 8
6  #define pin_cV 9
7
8  #define m1_1 33
9  #define m1_2 34
10 #define m1_3 35
11 #define m1_4 36
12 #define m2_1 37
13 #define m2_2 38
14 #define m2_3 39
15 #define m2_4 40
16
17 #define pwmMotor 2
18
19 //Variables para el control de motores
20 int dir = 0, steps = 0, bin = 1;
```

```

21 int lecturas = 0;
22 bool enc = false;
23 //int cH = 0, cV = 0, mov = 0, vh = 0, posant = 0, pin_c = 0;
24
25 //Variables para la adquisicion
26 //Voltajes
27 float vmax0 = 0, vmin0 = 999, vref0 = 0;
28
29 //Banderas
30 int capture = 0, c = 0, pos = 0, pos_ant = 0, change = 0;
31
32 //Datos individuales
33 float opto = 0, foto = 0;
34
35 //Posiciones de picos y valores
36 int pico0[2] = {0}, picoF = 0, maxF = -9999, cpicos0 = 0, diff0 = 0, diffF = 0;
37
38 void setup() {
39     analogReadResolution(13);
40     // Sets ADC range to 11 bits (0 to 2043). Teensy 3.2 ranges from 10 to 13 bits
41     analogReadAveraging(2);
42     //Average 4, 8, 16 and 32 readings, smooths stick input and jitter.
43     pinMode(A0, INPUT);
44     pinMode(A1, INPUT);
45     pinMode(izquierda, OUTPUT);
46     pinMode(derecha, OUTPUT);
47     pinMode(arriba, OUTPUT);
48     pinMode(abajo, OUTPUT);
49     pinMode(pin_cH, INPUT);
50     pinMode(pin_cV, INPUT);
51     pinMode(13, OUTPUT);
52     digitalWrite(13, HIGH);

```

```

53
54  pinMode(m1_1,OUTPUT);
55  pinMode(m1_2,OUTPUT);
56  pinMode(m1_3,OUTPUT);
57  pinMode(m1_4,OUTPUT);
58  pinMode(m2_1,OUTPUT);
59  pinMode(m2_2,OUTPUT);
60  pinMode(m2_3,OUTPUT);
61  pinMode(m2_4,OUTPUT);
62
63  digitalWrite(m1_1,0);
64  digitalWrite(m1_2,0);
65  digitalWrite(m1_3,0);
66  digitalWrite(m1_4,0);
67  digitalWrite(m2_1,0);
68  digitalWrite(m2_2,0);
69  digitalWrite(m2_3,0);
70  digitalWrite(m2_4,0);
71
72  pinMode(pwmMotor,OUTPUT);
73
74  analogWrite(pwmMotor,40);
75
76  Serial.begin(192000);
77 }
78
79 void loop() {
80
81  // Posicionador
82  if (Serial.available() > 0){
83      String dataSt = Serial.readString();
84      int str_len = dataSt.length() + 1;

```

```

85     char data[str_len];
86     dataSt.toCharArray(data, str_len);
87     sscanf(data, "D%d S%d L%d", &dir, &steps, &lecturas);
88
89     int i = 0;
90
91     for (i = 1; i <= steps; i++){
92         delay(10);
93
94         if(dir == 2 || dir == 3){
95             if (dir == 2){digitalWrite(m1_1+bin-1,0);}
96             if (dir == 3){digitalWrite(m2_1+bin-1,0);}
97         }
98
99         if(dir == 1 || dir == 4){
100             if (dir == 1){digitalWrite(m1_4-bin+1,0);}
101             if (dir == 4){digitalWrite(m2_4-bin+1,0);}
102         }
103
104         if (bin == 4){bin = 0;}
105         bin++;
106
107         if (dir == 1 || dir == 2){
108             if (dir == 2){digitalWrite(m1_1+bin-1,1);}
109             if (dir == 1){digitalWrite(m1_4-bin+1,1);}
110
111         }
112
113         if (dir == 3 || dir == 4){
114             if (dir == 3){digitalWrite(m2_1+bin-1,1);}
115             if (dir == 4){digitalWrite(m2_4-bin+1,1);}
116         }

```

```

117     }
118     Serial.println((String) "T");
119     delay(3);
120     if(lecturas>0){
121         leersensores(lecturas);
122     }
123     Serial.println((String) "W");
124
125 }
126
127 //}
128 }
129
130 void leersensores(int cantidad)
131 {
132     // Serial.println((String) i+("F" +(String)i));
133     //for(int i=0; i<cantidad; i++)
134     int refe = 0;
135     while(1)
136     {
137         // Apertura
138         //if (enc == false){
139         if (c>50000){
140             vmax0 = vmin0 * 2;
141         }
142
143         opto = analogRead(A0);
144         foto = analogRead(A1);
145
146         foto = - foto;
147
148         //Obtiene el valor de vref del opto

```

```

149     if (opto < vmin0){
150         vmin0 = opto;
151     }
152     if (opto > vmax0 || vref0 > vmax0){
153         vmax0 = opto;
154     }
155
156     vref0 = (vmin0 + vmax0) / 2;
157
158     //Determina cuando se comienzan a analizar datos
159     if (opto<=vref0){
160         capture=1;
161     }
162
163     //Analiza datos
164     if (capture==1){
165         //Obtiene el maximo del fotosensor
166         if (foto > maxF){
167             maxF = foto;
168             picoF = c;
169         }
170
171         c++;
172
173         if (opto > vref0 * 1.1){
174             pos = 1;
175             if (pos_ant == 0){
176                 change = 1;
177                 if (cpicos0 == 1){
178                     pico0[0] = c;
179                 }
180                 else{

```

```

181         pico0[1] = c;
182     }
183 }
184     pos_ant = 1;
185 }
186 else if (opto < vref0 * .9){
187     pos = 0;
188     if (pos_ant == 1){
189         change = 2;
190         if (cpicos0 == 0){
191             cpicos0++;
192         }
193         else{
194             cpicos0=2;
195         }
196     }
197     pos_ant = 0;
198 }
199
200 if (cpicos0 == 2 && change == 1){
201     diff0=pico0[1]-pico0[0];
202     diffF=picoF-pico0[0];
203     //Serial.println((String) "0" + diff0 + " F" + diffF);
204     if(diffF>310){
205         if(refe>1)
206         {
207             Serial.println((String)diff0 + "F" + diffF);
208         }
209         refe=refe+1;
210     }
211     maxF = -9999;
212     c = 0;

```

```

213     cpicos0 = 1;
214     pico0[0] = 0;
215     pico0[1] = 0;
216     }
217     }
218 if(refe==(cantidad+2)){break;}
219     }
220
221     }

```

Programa de mesa rotativa

```

1
2 //definicion de pins
3 const int motorPin1 = 8;    // 28BYJ48 In1
4 const int motorPin2 = 9;    // 28BYJ48 In2
5 const int motorPin3 = 10;   // 28BYJ48 In3
6 const int motorPin4 = 11;   // 28BYJ48 In4
7
8 const int m1_1 = 8;        // 28BYJ48 In1
9 const int m1_2 = 9;        // 28BYJ48 In2
10 const int m1_3 = 10;       // 28BYJ48 In3
11 const int m1_4 = 11;       // 28BYJ48 In4 D1 S28663 L0
12
13 long dir = 0, steps = 0, bin = 1;
14 int lecturas = 0;
15 bool enc = false;
16
17
18 //definicion variables
19 int motorSpeed = 1200;
20 //1200 //variable para fijar la velocidad

```

```

21 int stepCounter = 0;
22 // contador para los pasos
23 int stepsPerRev = 4076000;
24 //de hecho son 4096 x 84 dientes del engrane pasos para una vuelta completa
25
26 //tablas con la secuencia de encendido (descomentar la que necesiteis)
27 //secuencia 1-fase
28 //const int numSteps = 4;
29 //const int stepsLookup[4] = { B1000, B0100, B0010, B0001 };
30
31 //secuencia 2-fases
32 const int numSteps = 4;
33 const int stepsLookup[4] = { B1100, B0110, B0011, B1001 };
34
35 //secuencia media fase
36 //const int numSteps = 8;
37 //const int stepsLookup[8] = { B1000, B1100, B0100, B0110, B0010, B0011, B0001, B1001 };
38
39
40 void setup()
41 {
42     //declarar pines como salida
43     pinMode(motorPin1, OUTPUT);
44     pinMode(motorPin2, OUTPUT);
45     pinMode(motorPin3, OUTPUT);
46     pinMode(motorPin4, OUTPUT);
47     Serial.begin(19200);
48
49 }
50
51 void loop()
52 {

```

```

53
54  if (Serial.available() > 0){
55
56      String dataSt = Serial.readString();
57      int str_len = dataSt.length() + 1;
58      char data[str_len];
59      dataSt.toCharArray(data, str_len);
60      sscanf(data, "D%d S%u L%d", &dir, &steps, &lecturas);
61      Serial.print((String) "ACT ");
62      Serial.println(steps);
63
64      int i = 0;
65
66      for (i = 1; i <= steps; i++){
67
68
69          if(dir == 2){
70              if (dir == 2){clockwise();}
71              delayMicroseconds(motorSpeed);
72          }
73
74          if(dir == 1){
75              if (dir == 1){anticlockwise();}
76              delayMicroseconds(motorSpeed);
77          }
78
79          if (bin == 4){bin = 0;}
80          bin++;
81
82
83      }
84      Serial.println((String) "T");

```

```

85     delay(3);
86
87
88
89     }
90
91
92
93
94
95
96 }
97
98 void clockwise()
99 {
100     stepCounter++;
101     if (stepCounter >= numSteps) stepCounter = 0;
102     setOutput(stepCounter);
103 }
104
105 void anticlockwise()
106 {
107     stepCounter--;
108     if (stepCounter < 0) stepCounter = numSteps - 1;
109     setOutput(stepCounter);
110 }
111
112 void setOutput(int step)
113 {
114     digitalWrite(motorPin1, bitRead(stepsLookup[step], 0));
115     digitalWrite(motorPin2, bitRead(stepsLookup[step], 1));
116     digitalWrite(motorPin3, bitRead(stepsLookup[step], 2));

```

```
117     digitalWrite(motorPin4, bitRead(stepsLookup[step], 3));
118 }
119
120
```

Programa de escaneo en PC

```
1  #318.577777777778 pulsos por 8 grados
2  #39.8222222222222 pulsos por 1 grado
3  #1792 por 45 grados
4  import getopt, sys, os
5  import serial
6  import time
7  import warnings # std visa libraries
8  import numpy as np
9  from struct import unpack
10 import pylab
11 #import pyvisa as visa # http://github.com/hgrecco/pyvisa
12 import csv
13 np.set_printoptions(threshold=sys.maxsize)
14 #vc = visa.constants
15 fileSaveLocation = ''
16 bufferSize = 1024
17
18 puertocom = 'COM8'
19 velocidaddeconexion = 19200
20 gradoinicio = 0.0
21 gradofinal = 0.0
22 gradofinalz = 0.0
23 nombrebase = ""
24 step=0.1
25 ztep = 1.0
26 lecturas = 10
```

```

27  timeout = 10
28  rotacionact = 0
29
30  # Remove 1st argument from the
31  # list of command line arguments
32  argumentList = sys.argv[1:]
33
34  # Options
35  options = "p:mo:i:f:z:n:s:c:l:r"
36
37  # Long options
38  long_options = ["Puerto", "My_file", "Output =", "Inicio =", "Final =", "EjeZ =",
39  "Step =", "Zstep =", "Numerolecturas =", "RotacionActual ="]
40
41  try:
42      # Parsing argument
43      arguments, values = getopt.getopt(argumentList, options, long_options)
44
45      # checking each argument
46      for currentArgument, currentValue in arguments:
47
48          if currentArgument in ("-p", "--Puerto"):
49              print ("Se selecciono el puerto (% s)" % (currentValue))
50              puertocom = currentValue
51
52          elif currentArgument in ("-m", "--My_file"):
53              print ("Displaying file_name:", sys.argv[0])
54
55          elif currentArgument in ("-o", "--Output"):
56              print ("Enabling special output mode (% s)" % (currentValue))
57
58          elif currentArgument in ("-i", "--Inicio"):

```

```

59     print (("Grados iniciales = (% s)" % (currentValue))
60           gradoinicio = float(currentValue)
61
62     elif currentArgument in ("-f", "--Final"):
63         print (("Grados finales = (% s)" % (currentValue))
64               gradofinal = float(currentValue)
65
66     elif currentArgument in ("-z", "--EjeZ"):
67         print (("Eje Z final = (% s)" % (currentValue))
68               gradofinalz = float(currentValue)
69
70     elif currentArgument in ("-n", "--NombreBase"):
71         print (("Nombre base = (% s)" % (currentValue))
72               nombrebase = currentValue
73
74     elif currentArgument in ("-s", "--Step"):
75         print (("Step base = (% s)" % (currentValue))
76               step = float(currentValue)
77
78     elif currentArgument in ("-c", "--Zstep"):
79         print (("Step base en Z = (% s)" % (currentValue))
80               zstep = float(currentValue)
81
82     elif currentArgument in ("-l", "--Numerolecturas"):
83         print (("Lecturas por punto = (% s)" % (currentValue))
84               lecturas = int(currentValue)
85
86     elif currentArgument in ("-r", "--Rotacionactual"):
87         print (("Rotacion actual set = (% s)" % (currentValue))
88               rotacionact = currentValue
89
90 except getopt.error as err:

```

```

91     # output error, and return with an error code
92     print (str(err))
93
94     # Connect to arduino
95     ser = serial.Serial(puertocom, velocidaddeconexion, timeout=10)
96     print('Conectado a ' + puertocom)
97
98
99     #variables
100    pasos = 19200.0
101    grados = 360.0
102    pasosporgrado = (pasos/grados)
103    pasosporstep = pasosporgrado*step
104
105    gradoactual = 90.0
106    gradosiguiente = 90.0
107    stepactual = 4800
108    stepsiguiente = 4800
109
110    gradosiguiente = (gradoactual-gradoinicio)
111    avanzar = gradosiguiente * pasosporgrado
112    print(avanzar)
113    stepsiguiente=stepsiguiente-int(avanzar)
114    hacia = 0
115    Direccion = 1
116
117     #Choose where the motor is going
118
119    if(gradosiguiente < gradoactual):
120        hacia = -1
121        Direccion = 1
122    elif(gradosiguiente==gradoactual):

```

```

123     hacia = 1
124     Direccion = 2
125 else:
126     hacia = 1
127     Direccion = 2
128
129 path=(os.path.dirname(os.path.realpath(sys.argv[0])))+"\\")+nombrebase
130 fileSaveLocation=((path)+"\\SCOPE")
131 print(path)
132 try:
133     os.mkdir(path)
134     os.mkdir((path)+"\\SCOPE")
135 except:
136     print("Directorio existente, sobrescribiendo")
137 mensaje = "D" + str(Direccion) + " S" + str(int(avanzar)) + " L" + str(lecturas)
138 print("Enviando " + mensaje + "...")
139 ser.write((mensaje).encode())
140 print("Esperando respeusta 1...")
141
142 while 1:
143     xdata = ser.readline().decode('utf-8').strip()
144     print(xdata)
145     if (xdata=="T"):
146         print("Posicionamiento inicial completo\n\nIniciando escaneo...\n\n")
147         gradoactual = gradoinicio
148         stepactual = stepsiguiente
149         print(gradoactual)
150         print(stepactual)
151         break
152
153 gradozactual = 0.0
154 stepzactual = 0

```

```

155
156
157
158
159
160 seguro = False
161
162 for c in np.arange((0), (gradofinalz+ztep), ztep):
163     ser.reset_input_buffer()
164     ser.reset_output_buffer()
165     print("Z se ubica en " + str(gradozactual))
166     gradozsiguiente = c
167     zavanzar = gradozsiguiente - gradozactual
168     zavanzar2 = zavanzar*pasosporgrado
169     print("Se requiere mover Z a " + str(gradozsiguiente))
170     print("Se avanzara " +str(int(zavanzar2)) +" pasos")
171     mensajez = "D3" + " S" + str(int(zavanzar2)) + " L0"
172     print("Enviando " + str(mensajez))
173     print("\n\n")
174     stepzactual = stepzactual+int(zavanzar2)
175     gradozactual = (stepzactual/pasosporgrado)
176     ser.write((mensajez).encode())
177     print("Esperando respuesta 2...")
178     while 1:
179         xdata = ser.readline().decode('utf-8').strip()
180         print(xdata)
181         if (xdata=="W" or (mensajez == "D3 S0 L0" or mensajez == "D4 S0 L0")):
182             print("Posicionamiento Z completo...\n\n")
183             seguro = False
184             break
185     print("\n\n")
186

```

```

187
188     for i in np.arange((gradoinicio), (gradofinal+step), step):
189         listaF = []
190         listaO = []
191         listaX = []
192         listaZ = []
193         listarot= []
194         print("Se deberia estar en " + str(i))
195         gradosiguiente=i
196         gradoavanzar = gradosiguiente-gradoactual
197         avanzar = gradoavanzar * pasosporgrado
198         print("ubicado en "+ str(gradoactual))
199         print("se requiere ir a "+str(gradosiguiente))
200         print("Avanzara " + str(int(avanzar)) + " pasos el motor")
201         mensaje = "D2" + " S" + str(int(avanzar)) + " L" + str(lecturas)
202         print("Enviando " + mensaje + "...")
203         ser.write((mensaje).encode())
204         print("Esperando respuesta 3...")
205
206     while 1:
207         xdata = ser.readline().decode('utf-8').strip()
208         print(xdata)
209         if (xdata=="T"):
210             print("Posicionamiento PASO completo\n\nIniciando escaneo...\n\n")
211             break
212     stepactual = stepactual+int(avanzar)
213     gradoactual = (stepactual/pasosporgrado)
214
215     while 1:
216         xdata = ser.readline().decode('utf-8').strip()
217         print(xdata)
218         if (xdata=="W"):

```

```

219         print("Posicionamiento PASO completo\n\nIniciando escaneo...\n\n")
220         break
221     R = xdata.split("F")
222     try:
223         lista0.append(int(R[0]))
224         listaF.append(int(R[1]))
225         listaX.append(float(gradoactual))
226         listaZ.append(float(gradozactual))
227         listarot.append(rotacionact)
228         seguro = True
229     except:
230         print("woops")
231
232     print("\n")
233
234
235
236
237     Voltlist = []
238     Timelist = []
239
240     #savefig = path + "\\SCOPE\\" + nombrebase + "_SCOPE " + str(stepactual) + "_Z_"
241     + str(gradozactual) + "_X_" + str(gradoactual) + "_DOS.png"
242     #[_Volts, _Time] = savescope(savefig)
243     #Voltlist = _Volts.tolist()
244     #Timelist = _Time.tolist()
245     #varfile = path + "\\SCOPE\\" + nombrebase + "SCOPE " + str(stepactual) + "_Z_"
246     + str(gradozactual) + "_X_" + str(gradoactual) + "_DOS.csv"
247
248     #with open(varfile, 'w') as f:
249     #    writer = csv.writer(f, delimiter='\t')
250     #    writer.writerows(zip(Timelist, Voltlist))

```

```

251
252
253     #Volltlist = []
254     #Timelist = []
255
256     varfile = path + "\\\"+str(rotacionact) + "_Z_" + str(gradozactual)
257     + "_X_"+str(gradoactual)+".csv"
258     if seguro == True:
259         with open(varfile, 'w', newline='') as f:
260             writer = csv.writer(f, delimiter=',')#\t')
261             writer.writerows(zip("OP","FOT", "X Calculada", "Z"))
262             writer.writerows(zip(lista0,listaF, listaX, listaZ, listarot))
263
264     mensaje = "D2" + " S2 L0"
265     print("Enviando " + mensaje + "...")
266     ser.write((mensaje).encode())
267     print("Esperando respuesta 4...")
268
269     while 1:
270         xdata = ser.readline().decode('utf-8').strip()
271         print(xdata)
272         if (xdata=="T"):
273             print("Posicionamiento compenzacion completo\n\nContinuando...\n\n")
274             break
275     gradosiguiente = (gradoactual-gradoinicio)
276     avanzar = gradosiguiente * pasosporgrado
277     print(avanzar)
278     stepsiguiente=stepsiguiente-int(avanzar)
279     mensaje = "D1 S" + str(int(avanzar)) + " L0"
280     print("Enviando " + mensaje + "...")
281     ser.write((mensaje).encode())
282     print("Esperando respeusta 5...")

```

```

283
284     while 1:
285         xdata = ser.readline().decode('utf-8').strip()
286         print(xdata)
287         if (xdata=="T"):
288             print("Posicionamiento siguiente Z completo\n\nIniciando escaneo...\n\n")
289             stepactual = stepactual-int(avanzar)
290             gradoactual = (stepactual/pasosporgrado)
291             print(gradoactual)
292             print(stepactual)
293             break
294
295
296 ser.close()
297 print("Es script se ha completado con exito\n")
298 val = input("Presione ENTER para salir....")
299
300 #for i in np.arange(Rangelow, Rangehigh, Delta):
301     #print(i)
302
303 #x=ser.readlines()
304 #x=[l.decode('utf-8').strip() for l in x]
305 #print(x)
306
307

```

Programa de registro de nubes de puntos

```

1
2
3 import open3d as o3d
4
5 import argparse

```

```

6
7
8 def preprocess_point_cloud(pcd, voxel_size):
9     pcd_down = pcd.voxel_down_sample(voxel_size)
10    pcd_down.estimate_normals(
11        o3d.geometry.KDTreeSearchParamHybrid(radius=voxel_size * 2.0,
12                                               max_nn=30))
13    pcd_fpfh = o3d.pipelines.registration.compute_fpfh_feature(
14        pcd_down,
15        o3d.geometry.KDTreeSearchParamHybrid(radius=voxel_size * 5.0,
16                                               max_nn=100))
17    return (pcd_down, pcd_fpfh)
18
19
20 if __name__ == '__main__':
21    pcd_data = o3d.data.DemoICPPointClouds()
22    parser = argparse.ArgumentParser(
23        'Global point cloud registration example with RANSAC')
24    parser.add_argument('src',
25                        type=str,
26                        default=pcd_data.paths[0],
27                        nargs='?',
28                        help='path to src point cloud')
29    parser.add_argument('dst',
30                        type=str,
31                        default=pcd_data.paths[1],
32                        nargs='?',
33                        help='path to dst point cloud')
34    parser.add_argument('--voxel_size',
35                        type=float,
36                        default=0.90, #0.05
37                        help='voxel size in meter used to downsample inputs')

```

```

38 parser.add_argument(
39     '--distance_multiplier',
40     type=float,
41     default=1.5,
42     help='multiplier used to compute distance threshold'
43     'between correspondences.'
44     'Threshold is computed by voxel_size * distance_multiplier.')
45 parser.add_argument('--max_iterations',
46                     type=int,
47                     default=1000000,
48                     help='number of max RANSAC iterations')
49 parser.add_argument('--confidence',
50                     type=float,
51                     default=0.999,
52                     help='RANSAC confidence')
53 parser.add_argument(
54     '--mutual_filter',
55     action='store_true',
56     help='whether to use mutual filter for putative correspondences')
57
58 args = parser.parse_args()
59
60 voxel_size = args.voxel_size
61 distance_threshold = args.distance_multiplier * voxel_size
62
63 o3d.utility.set_verbosity_level(o3d.utility.VerbosityLevel.Debug)
64 print('Reading inputs')
65 #src = o3d.io.read_point_cloud("1.ply")#o3d.io.read_point_cloud(args.src)
66 src = o3d.io.read_point_cloud("\Mediciones craneo y objetos
67 \\6\\Pos1 (21, 23, 10)\\solo 6 camara lite.ply")#o3d.io.read_point_cloud(args.src)
68 Esta es la base
69 #dst = o3d.io.read_point_cloud("1-respaldo.ply")#o3d.io.read_point_cloud(args.dst)

```

```

70     #dst = o3d.io.read_point_cloud("\Mediciones craneo y objetos - Copy
71     \\Skull\\Pos2 (33, 30, 10)\\22.5\\8of16 - Cloud.ply")
72     #o3d.io.read_point_cloud(args.dst)Esta se mueve
73     dst = o3d.io.read_point_cloud("\Mediciones craneo y objetos
74     \\6\\Pos1 (21, 23, 10)\\rot4.ply")#o3d.io.read_point_cloud(args.dst)Esta se mueve
75
76     print('Downsampling inputs')
77     src_down, src_fpfh = preprocess_point_cloud(src, voxel_size)
78     dst_down, dst_fpfh = preprocess_point_cloud(dst, voxel_size)
79
80     print('Running RANSAC')
81     result = o3d.pipelines.registration.registration_ransac_based_on_feature_matching(
82         src_down,
83         dst_down,
84         src_fpfh,
85         dst_fpfh,
86         mutual_filter=args.mutual_filter,
87         max_correspondence_distance=distance_threshold,
88         estimation_method=o3d.pipelines.registration.
89         TransformationEstimationPointToPoint(False),
90         ransac_n=3,
91         checkers=[
92             o3d.pipelines.registration.CorrespondenceCheckerBasedOnEdgeLength(
93                 0.9),
94             o3d.pipelines.registration.CorrespondenceCheckerBasedOnDistance(
95                 distance_threshold)
96         ],
97         criteria=o3d.pipelines.registration.RANSACConvergenceCriteria(
98             args.max_iterations, args.confidence))
99
100     src.paint_uniform_color([1, 0, 0])
101     dst.paint_uniform_color([0, 1, 0])

```

```
102     o3d.visualization.draw([src.transform(result.transformation), dst])
103
104
105
```

Programa de compilación de datos

```
1
2
3 import getopt, sys, os
4 import numpy as np
5 from struct import unpack
6 import csv
7 import math
8
9 [x,y,z]=0.0,0.0,0.0
10 # Remove 1st argument from the
11 # list of command line arguments
12 argumentList = sys.argv[1:]
13
14 # Options
15 options = "p:mo:i:f:z:n:s:c:l:"
16
17 # Long options
18 long_options = ["Puerto", "My_file", "Output =", "Inicio =", "Final =",
19 "EjeZ =", "Step =", "Zstep =", "Numerolecturas ="]
20
21 try:
22     # Parsing argument
23     arguments, values = getopt.getopt(argumentList, options, long_options)
24
25     # checking each argument
26     for currentArgument, currentValue in arguments:
```

```

27
28     if currentArgument in ("-p", "--Puerto"):
29         print (("Se selecciono el puerto (% s)") % (currentValue))
30         #puertocom = currentValue
31
32     elif currentArgument in ("-m", "--My_file"):
33         print ("Displaying file_name:", sys.argv[0])
34
35     elif currentArgument in ("-o", "--Output"):
36         print (("Enabling special output mode (% s)") % (currentValue))
37
38     elif currentArgument in ("-i", "--Inicio"):
39         print (("Grados iniciales = (% s)") % (currentValue))
40         #gradoinicio = float(currentValue)
41
42     elif currentArgument in ("-f", "--Final"):
43         print (("Grados finales = (% s)") % (currentValue))
44         #gradofinal = float(currentValue)
45
46     elif currentArgument in ("-z", "--EjeZ"):
47         print (("Eje Z final = (% s)") % (currentValue))
48         #gradofinalz = float(currentValue)
49
50     elif currentArgument in ("-n", "--NombreBase"):
51         print (("Nombre base = (% s)") % (currentValue))
52         #nombrebase = currentValue
53
54     elif currentArgument in ("-s", "--Step"):
55         print (("Step base = (% s)") % (currentValue))
56         #step = float(currentValue)
57
58     elif currentArgument in ("-c", "--Zstep"):

```

```

59     print ("Step base en Z = (% s)" % (currentValue))
60     #zstep = float(currentValue)
61
62     elif currentArgument in ("-l", "--Numerolecturas"):
63         print ("Lecturas por punto = (% s)" % (currentValue))
64         #lecturas = int(currentValue)
65
66
67 except getopt.error as err:
68     # output error, and return with an error code
69     print (str(err))
70
71
72
73
74
75 path=(os.path.dirname(os.path.realpath(sys.argv[0])))+"\\\\"#+nombrebase
76 fileSaveLocation=((path)+"\\SCOPE")
77 #print(path)
78
79
80 def getfiles(mypath):
81     f = []
82     for (dirpath, dirnames, filenames) in os.walk(mypath):
83         f.extend(filenames)
84         break
85     return(dirpath, dirnames, filenames)
86
87 [A,B,C] = getfiles(path)
88 print(A)
89 print(B)
90 cm = B[0].split(" ")

```

```

91 cm=float(cm[0])
92 print(cm)
93 print(C)
94 varfile = path + "result2.csv"
95 with open(varfile, 'w', newline='') as f:
96     writer = csv.writer(f, delimiter=',')#\t')
97     writer.writerow(("Cij", "Bij", "Xij", "Yij", "Zij", "Xcv",
98         "Ycv", "Zcv", "DifX", "DifY", "DifZ"))
99
100 for dirs in B:
101     #print(dirs)
102     #####print(path)
103     [D,E,F] = getfiles((path+dirs))
104     color = D.split("\\")
105     print(color,"asd")
106     [G,H,I] = getfiles((path+str(dirs)+str("\\")+str(E[0])))
107     for fil in I:
108         csvfile = ((path+str(dirs)+str("\\")+str(E[0])) + "\\ "+ fil)
109         #print(csvfile)
110         x = fil.split("_")
111         #print(x)
112         L=x[4].split(".csv")
113         #print(L)
114         with open(csvfile, "r") as my_file:
115             B_ij = float(0.0)
116             B_360 = float(360.0)
117             t_alpha = float(0.0)
118             T2pi = float(0.0)
119             C_ij = float(0.0)
120             X_ij = float(0.0)
121             Y_ij = float(0.0)
122             Z_ij = float(0.0)

```

```

123     file_reader = csv.reader(my_file)
124     x_pr = 0.0
125     y_pr = 0.0
126     z_pr = 0.0
127     i2 = 0
128     x=1000
129     y=1000
130     z=1000
131     for i in file_reader:
132         if(i[0] != "0"):
133             #           i2=i2+1
134             #print(i2)
135             T2pi = float(i[0])
136             t_alpha = float(i[1])
137             C_ij = float(i[2])
138             B_ij=((t_alpha*B_360)/T2pi)
139             betha_j = float(i[3])
140             #print(B_ij)
141             X_ij=float(100)*float(((math.sin(B_ij/57.2957)*
142             (math.sin(C_ij/57.2957))/(math.sin((B_ij+C_ij)/57.2957))))))
143             Y_ij = float(100)*float(((1/2)-((math.cos(B_ij/57.2957)*
144             (math.sin(C_ij/57.2957))/(math.sin((B_ij+C_ij)/57.2957))))))
145             Z_ij = (math.tan(betha_j/57.2957))*
146             X_ij
147             X_V = cm
148             c_V = (cm/(math.sin(math.radians(C_ij))))
149             y_V = math.sqrt(((c_V**2)-(cm**2)))
150             y_V = y_V -50
151             z_V = (math.tan(math.radians(betha_j)))*X_V
152             #print(i2)
153             if(X_ij<120 and X_ij>30 and Y_ij<120 and Y_ij>-120):
154                 #i2=i2+1

```

```

155         #x_pr = x_pr+X_ij
156         #y_pr = y_pr + Y_ij
157         #z_pr = z_pr + Z_ij
158
159
160         #print(str(Y_ij), str(y_V), str(C_ij),str(cm))
161         #print(X_ij)
162         if(X_ij>59 and X_ij<65 and Y_ij<100 and (Y_ij>-100)):
163         writer.writerow((str(X_ij), str(Y_ij), str(Z_ij)))
164         #valores restringidos
165             i2=i2+1
166             x_pr = x_pr+X_ij
167             y_pr = y_pr + Y_ij
168             z_pr = z_pr + Z_ij
169             #ver mas alto
170             if(X_ij<x):
171                 x=X_ij
172                 y=Y_ij
173                 z=Z_ij
174
175         #writer.writerow((str(X_ij), str(Y_ij),
176         str(Z_ij)))
177         #valores calculados
178         #writer.writerow((str(X_V), str(y_V), str(z_V)))
179         #Valores "verdaderos"
180         difx = X_V-X_ij
181         dify = y_V-Y_ij
182         difz = z_V-Z_ij
183         #writer.writerow((str(C_ij), str(B_ij), str(X_ij), str(Y_ij),
184         #str(Z_ij), str(X_V), str(y_V), str(z_V), str(difx), str(dify), str(difz)))
185         #Valores de tabla
186         #print("termino archivo")

```

```
187
188     try:
189         x_pr = x_pr/i2
190         y_pr = y_pr/i2
191         z_pr = z_pr/i2
192         #writer.writerow((str(x_pr), str(y_pr), str(z_pr)))
193         #valores promediados
194         #writer.writerow((str(x), str(y), str(z))) #mas altos
195
196     except:
197         print("no se escribira nada")
198
199
200
201
202
203
204
205
206
207
208
```
