



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
RECONSTRUCCIÓN VOLUMÉTRICA VIRTUAL DE OBJETOS
EN UN ENTORNO INDOOR MEDIANTE CÁMARAS
ESTEREOSCÓPICAS Y PROCESAMIENTO DIGITAL DE
IMÁGENES”**

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES**

AUTOR: GABRIEL DAVID CASTRO DUTAN

DIRECTOR: Ing. PAUL DAVID MORENO AVILÉS PhD.

Riobamba – Ecuador

2023



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
RECONSTRUCCIÓN VOLUMÉTRICA VIRTUAL DE OBJETOS
EN UN ENTORNO INDOOR MEDIANTE CÁMARAS
ESTEREOSCÓPICAS Y PROCESAMIENTO DIGITAL DE
IMÁGENES”**

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES**

AUTOR:

GABRIEL DAVID CASTRO DUTAN

Riobamba – Ecuador

2023

© 2023, Castro Dutan Gabriel David

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Gabriel David Castro Dutan, declaro que el presente Trabajo de Integración Curricular es de mi autoría y los resultados de este son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 30 de marzo del 2023



Gabriel David Castro Dutan
2100812086

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; tipo: Proyecto Técnico, “**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RECONSTRUCCIÓN VOLUMÉTRICA VIRTUAL DE OBJETOS EN UN ENTORNO INDOOR MEDIANTE CÁMARAS ESTEREOSCÓPICAS Y PROCESAMIENTO DIGITAL DE IMÁGENES**”, realizado por el señor **GABRIEL DAVID CASTRO DUTAN**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

FIRMA

FECHA

Ing. Jefferson Alexander Ribadeneira Ramírez
PRESIDENTE DEL TRIBUNAL



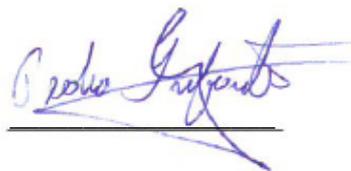
2023-03-30

Ing. Paul David Moreno Avilés
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR



2023-03-30

Ing. Pedro Severo Infante Moreira
ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR



2023-03-30

DEDICATORIA

El presente trabajo se lo dedico a mi Dios, a mis padres Rafael y Sara, hermanos Kevin y Angela, amigos y profesores, quienes creyeron en mí, con lo cual estoy eternamente agradecido por su paciencia, su ayuda moral y amor para conmigo. A mis padres que con tanto esfuerzo y trabajo lograron esto, a mis hermanos cuya paciencia y soporte me ayudaron a culminar esta meta. A mis amigos cuya ayuda para compartir los conocimientos lograron que avanzara. Y, primeramente, a mi Dios, en quien reside toda gota de esperanza para ser una persona mejor.

Gabriel

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS.....	xi
ÍNDICE DE ILUSTRACIONES.....	xii
ÍNDICE DE ANEXOS.....	xv
RESUMEN.....	xvi
SUMMARY	xvii
INTRODUCCION	1

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA	2
1.1 Antecedentes.....	2
1.2 Formulación del problema.....	2
1.2.1 <i>Sistematización del problema</i>	3
1.3 Justificación.....	3
1.3.1 <i>Justificación teórica</i>	3
1.3.2 <i>Justificación aplicada</i>	5
1.4 Objetivos.....	7
1.4.1 <i>Objetivo general</i>	7
1.4.2 <i>Objetivos específicos</i>	8

CAPÍTULO II

2. ESTADO DEL ARTE	9
2.1 Digitalización 3D.....	9
2.2 Técnicas de reconstrucción 3D	9
2.2.1 <i>Técnicas de reconstrucción por contacto</i>	10

2.2.2	<i>Técnicas de reconstrucción sin contacto</i>	11
2.2.2.1	<i>Técnicas de reconstrucción activas</i>	11
2.2.2.1.1	<i>Tiempo de vuelo</i>	11
2.2.2.1.2	<i>Escáner laser por triangulación</i>	12
2.2.2.1.3	<i>Escáner de luz estructurada</i>	13
2.2.2.1.4	<i>Escáner por diferencia de fase</i>	14
2.2.2.1.5	<i>Tomografía computarizada</i>	14
2.2.2.1.6	<i>No ópticas</i>	15
2.2.2.2	<i>Técnicas de reconstrucción pasivas</i>	15
2.2.2.2.1	<i>Fotogrametría</i>	15
2.2.2.2.2	<i>Profundidad desde estéreo</i>	16
2.3	Trabajos relacionados	17
2.4	Visión Estereoscópica	18
2.5	Perspectiva y sensación de profundidad	18
2.5.1	<i>Claves monoculares: percibidas por cada ojo individualmente</i>	19
2.5.2	<i>Claves estereoscópicas: percibidas combinando los dos ojos, visión binocular</i>	19
2.6	Concepto de cámaras 3D	20
2.7	Cámara Estéreo	20
2.8	Mynt Eye S1030-IR-120/Mono	21
2.9	Calibración de la cámara estéreo	23
2.10	Modelo de cámara extrínseco e intrínseco	23
2.11	Modelo de proyección estenopeica	24
2.12	Modelos de distorsión de lente	25
2.12.1	<i>Modelo de distorsión radial-tangencial</i>	26
2.12.2	<i>Modelo de distorsión de Kannala-Brad</i>	27
2.13	Mejores prácticas de calibración	28
2.14	Geometría Epipolar	29

2.15	Triangulación estéreo.	30
2.16	Correspondencia estéreo.	32
2.16.1	<i>Métodos locales</i>	32
2.16.2	<i>Métodos globales</i>	33
2.17	Disparidad estéreo mediante coincidencia de bloques semi-globales (SGBM)	34
2.18	SDK de Mynt Eye	35
2.19	OpenCV	36
2.20	Microsoft Visual Studio.	37
2.21	Matlab.	38
2.22	MeshLab	38
2.23	Colmap.	39

CAPÍTULO III

3.	MARCO METODOLÓGICO	40
3.1	Análisis: análisis y especificación de los requisitos del sistema.	40
3.1.1	<i>Características y funciones del sistema de reconstrucción virtual de objetos.</i>	41
3.1.2	<i>Requisitos del sistema de reconstrucción virtual de objetos.</i>	42
3.1.3	<i>Análisis de la definición de los requisitos.</i>	42
3.2	Diseño: diseño y especificación del sistema.	43
3.2.1	<i>Etapas del sistema.</i>	43
3.2.1.1	<i>Obtención de imágenes mediante SDK MYNT EYE y Visual Studio.</i>	45
3.2.1.2	<i>Lectura y procesamiento de imágenes en Matlab.</i>	46
3.2.1.3	<i>Calibración de imágenes estéreo.</i>	47
3.2.1.4	<i>Rectificación, disparidad, segmentación y triangulación de imágenes estéreo.</i>	53
3.2.1.5	<i>Unión y costura de nubes de puntos.</i>	58
3.2.2	<i>Diseño del sistema de reconstrucción volumétrica.</i>	61
3.3	Implementación del sistema de reconstrucción.	63

3.4	Verificación: pruebas unitarias y método de evaluación del sistema.....	63
3.4.1	<i>Configuración de los parámetros de imagen.....</i>	64
3.4.2	<i>Parámetros de calibración.....</i>	74
3.4.3	<i>Rectificación de imágenes.....</i>	76
3.4.4	<i>Disparidad.....</i>	77
3.4.5	<i>Reconstrucción.....</i>	80
3.4.6	<i>Costura de nube de puntos.....</i>	81
3.4.7	<i>Métodos de evaluación del sistema de reconstrucción.....</i>	82
3.5	Mantenimiento: Entrega del sistema de reconstrucción.....	83

CAPÍTULO IV

4.	MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS	84
4.1	Evaluación del sistema de reconstrucción.....	84
4.1.1	<i>Velocidad del sistema.....</i>	84
4.1.1.1	<i>Tiempo de Rectificación de imágenes.....</i>	84
4.1.1.2	<i>Tiempo de Procesamiento de imágenes.....</i>	86
4.1.1.3	<i>Tiempo de Unión y costura de nube de puntos.....</i>	87
4.1.1.4	<i>Tiempo total medio del sistema de reconstrucción.....</i>	88
4.1.2	<i>Precisión del sistema.....</i>	88
4.1.2.1	<i>Prueba T-Student para medias de dos muestras emparejadas entre medidas obtenidas por este proyecto y el programa COLMAP.....</i>	89
4.1.3	<i>Mediciones de exactitud.....</i>	93
4.1.3.1	<i>Error absoluto entre las medidas obtenidas mediante el sistema de reconstrucción y el dato real.....</i>	93
4.2	Discusión de los parámetros de comparación.....	95
4.3	Objetos digitalizados.....	95
4.4	Limitación del sistema.....	100

4.5	Costos del sistema de reconstrucción.....	100
	CONCLUSIONES.....	102
	RECOMENDACIONES.....	103
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-2:	MYNTEYE S1030-IR datasheet.	22
Tabla 1-3:	Tipos de datos para formato PNG.	46
Tabla 2-3:	Variación de la ganancia de la cámara.	65
Tabla 3-3:	Variación del brillo de la cámara.....	67
Tabla 4-3:	Variación del contraste de la cámara.....	71
Tabla 5-3:	Parámetros de calibración.	74
Tabla 6-3:	Disparidad vs Distancia.....	78
Tabla 7-3:	Matrices A y B para cada vista del objeto.....	81
Tabla 8-3:	Indicadores de desempeño de escáneres 3D.....	82
Tabla 1-4:	Tiempo de rectificación de imágenes.	85
Tabla 2-4:	Tiempo de rectificación de imágenes.	86
Tabla 3-4:	Tiempo de unión de nubes de puntos.	87
Tabla 4-4:	Prueba de normalidad de Shapiro-Wilk para los tiempos de ejecución del algoritmo.	88
Tabla 5-4:	Medidas en centímetros del objeto mediante el sistema propuesto y COLMAP.	89
Tabla 6-4:	Prueba de distribución normal de las dos muestras pareadas.....	90
Tabla 7-4:	Prueba F para varianza de dos muestras.....	91
Tabla 8-4:	Prueba T-Student para dos muestras pareadas.	91
Tabla 9-4:	Prueba Z de una muestra.	93
Tabla 10-4:	Error absoluto entre el sistema propuesto y medida real.....	94
Tabla 11-4:	Costos de implantación del sistema de reconstrucción.	101

ÍNDICE DE ILUSTRACIONES

Ilustración 1-1:	Topología de arreglo multi-vista.	4
Ilustración 2-1:	Coordenada Z a partir de un par de cámaras.	5
Ilustración 3-1:	Imagen izquierda y derecha de cámara estéreo.....	5
Ilustración 4-1:	Diagrama de flujo de algoritmo de digitalización estéreo.	7
Ilustración 1-2:	Técnicas de reconstrucción 3D.....	9
Ilustración 2-2:	Brazo de medición articulado.	10
Ilustración 3-2:	Máquina de medición por coordenadas.	11
Ilustración 4-2:	Funcionamiento de sensor basado en tiempo de vuelo.....	12
Ilustración 5-2:	Esquema de triangulación por láser.	13
Ilustración 6-2:	Geometría de luz estructurada.	13
Ilustración 7-2:	Esquema de escáner por diferencia de fase.	14
Ilustración 8-2:	Reconstrucción 3D del cráneo mediante TC.	15
Ilustración 9-2:	Esquema técnico sfm.	16
Ilustración 10-2:	Esquema estéreo cámara.....	16
Ilustración 11-2:	Descripción de la visión binocular.	18
Ilustración 12-2:	Descripción de la perspectiva de una imagen real.....	19
Ilustración 13-2:	Descripción de la perspectiva de una imagen real.....	20
Ilustración 14-2:	MYNTEYE S1030 camera.	22
Ilustración 15-2:	Modelo de proyección de cámara.	23
Ilustración 16-2:	Modelo de proyección estenopeica.....	24
Ilustración 17-2:	Ejemplos de las distintas distorsiones ópticas. De izquierda a derecha, imagen sin distorsión óptica, imagen con distorsión radial positiva, imagen con distorsión radial negativa e imagen con distorsión tangencial.....	26
Ilustración 18-2:	Imagen con distorsión tangencial.	27
Ilustración 19-2:	Imagen con distorsión tangencial.	28

Ilustración 20-2:	Plano epipolar.	30
Ilustración 21-2:	Geometría en el plano epipolar para cámaras paralelas.	31
Ilustración 22-2:	Ejemplos de configuración de la cámara Mynt s1030.	36
Ilustración 23-2:	Software de desarrollo Visual Studio.	37
Ilustración 1-3:	Metodología <i>waterfall</i>	41
Ilustración 2-3:	Diagrama de flujo del sistema de reconstrucción.	44
Ilustración 3-3:	Imágenes izquierda y derecha de la cámara Mynt Eye s1030.	46
Ilustración 4-3:	Comandos Matlab para lectura y procesado de imágenes.	47
Ilustración 5-3:	Patrón de calibración tablero de ajedrez.	47
Ilustración 6-3:	Patrón de calibración tablero de ajedrez.	48
Ilustración 7-3:	Obtención de imágenes con respecto al plano de cámara.	49
Ilustración 8-3:	Stereo Camera Calibrator App.	50
Ilustración 9-3:	Stereo Camera Calibrator App: Agregación de imágenes.	51
Ilustración 10-3:	Stereo Camera Calibrator App: Re-proyección de imágenes.	51
Ilustración 11-3:	Stereo Camera Calibrator App: Media de error en pixeles.	52
Ilustración 12-3:	Stereo Camera Calibrator App: Diagrama de calibración.	52
Ilustración 13-3:	Stereo Camera Calibrator App: Resultados de rectificación.	53
Ilustración 14-3:	Rectificación de imágenes.	54
Ilustración 15-3:	Mapa de disparidad mediante el método SGM y BM de una caja rectangular.	55
Ilustración 16-3:	Nube de puntos de una sola vista de una caja rectangular.	57
Ilustración 17-3:	Nube de puntos de una sola vista de una caja rectangular segmentada.	58
Ilustración 18-3:	Vista izquierda y derecha del objeto.	59
Ilustración 19-3:	Vista izquierda y derecha del objeto paralelas al eje X.	60
Ilustración 20-3:	Vista izquierda y derecha del objeto en un solo plano.	60
Ilustración 21-3:	Objeto reconstruido con dos vistas.	61
Ilustración 22-3:	Geometría del sistema de reconstrucción.	62

Ilustración 23-3:	Diseño del sistema de reconstrucción.....	62
Ilustración 24-3:	Implementación del sistema de reconstrucción.	63
Ilustración 25-3:	Luxómetros lux Light Meter Pro y Free.	65
Ilustración 26-3:	Patrón de calibración, vistas izquierda y derecha.	74
Ilustración 27-3:	Rectificación de imágenes.	77
Ilustración 28-3:	Rango de disparidad.	80
Ilustración 29-3:	Reconstrucción 3D de las caras del objeto.	81
Ilustración 30-3:	Sistema de reconstrucción: arreglo de cámaras.	83
Ilustración 1-4:	Imagen de tres cajas apiladas y su nube de puntos.	84
Ilustración 2-4:	Dispersión de datos del sistema propuesto y COLMAP.....	92
Ilustración 3-4:	Nube de puntos de vista lateral de un elefante.....	95
Ilustración 4-4:	Archivo PLY.....	96
Ilustración 5-4:	Ventana de edición de Meshlab.....	96
Ilustración 6-4:	Nube de puntos de elefante filtrado el ruido.....	97
Ilustración 7-4:	Malla del objeto elefante.	97
Ilustración 8-4:	Reconstrucción de violín, imagen, nube de puntos y malla del objeto respectivamente.	98
Ilustración 9-4:	Reconstrucción de rana: imagen, nube de puntos y malla del objeto respectivamente.	98
Ilustración 10-4:	Reconstrucción de cajas apiladas: imagen, nube de puntos y malla del objeto respectivamente.	99
Ilustración 11-4:	Reconstrucción de mochila: imagen, nube de puntos y malla vista izquierda y derecha respectivamente.....	99
Ilustración 12-4:	Textura de objeto digital.....	100

ÍNDICE DE ANEXOS

Anexo A: Código de adquisición de imágenes estéreo mediante SDK.

Anexo B: Código del sistema de reconstrucción en Matlab.

RESUMEN

El objetivo de este trabajo fue el de diseñar e implementar un sistema de reconstrucción volumétrico virtual de objetos en un entorno indoor mediante cámaras estereoscópicas y procesamiento digital de imágenes, para lo cual se obtuvieron imágenes de un arreglo cuadrado de cuatro cámaras estéreo localizadas en las esquinas del arreglo. Las imágenes fueron obtenidas mediante cámaras estéreo de la marca *MYNT EYE S1030*, mediante la metodología en cascada se desarrolló un algoritmo de reconstrucción que hizo uso de imágenes rectificadas y calibradas, la búsqueda de la profundidad fue realizada mediante el método de coincidencia de bloques semi-global (SGBM), por lo tanto, la reconstrucción utilizó el método de triangulación a partir del mapa de profundidad encontrado; para formar un objeto completo a partir de nubes de puntos individuales obtenidas de cada vista del arreglo, el algoritmo ajustó cada nube de puntos respecto a la vista vecina anterior por medio de matrices de rotación y traslación, de manera que todas las vistas formaron una sola nube de puntos, el sistema fue evaluado mediante el análisis estadístico de tres parámetros característicos de rendimiento para sistemas de escaneo volumétrico: velocidad de respuesta, precisión y confiabilidad. El sistema de reconstrucción presentó un tiempo medio de reconstrucción de 12,84 segundos, para la precisión se comparó con COLMAP que por medio de una prueba *t-student* de 30 muestras pareadas se demostró que no existe diferencia significativa entre las medias de estas, en cuanto a la confiabilidad el sistema demostró que no es preciso, resultando en un error de 2 mm de media. Se concluye que el sistema no es preciso como en otros métodos de reconstrucción debido a la técnica estéreo, pero puede ser utilizado por aplicaciones que no requieran un alto nivel de precisión. Se recomienda la utilización de otras técnicas activas como triangulación laser.

Palabras clave: <RECONSTRUCCIÓN ESTÉREO>, <COINCIDENCIA DE BLOQUES SEMI-GLOBAL (SGBM)>, <CALIBRACIÓN ESTÉREO>, <NUBE DE PUNTOS>, <RECONSTRUCCIÓN MULTI-VISTA>, <CÁMARA MYNT>, < TRIANGULACIÓN >, < GEOMETRÍA EPIPOLAR >.



Abin By.

SUMMARY

The objective of this work was to design and implement a virtual volumetric reconstruction system of objects in an indoor environment using stereoscopic cameras and digital image processing, for which images were obtained from a square arrangement of four stereo cameras located in the corners of the arrangement. The images were obtained using MYNT EYE S1030 brand stereo cameras, through the cascade methodology a reconstruction algorithm was developed that made use of rectified and calibrated images, the depth search was performed using the semi-block matching method. global (SGBM), therefore, the reconstruction used the triangulation method from the depth map found; To form a complete object from individual point clouds obtained from each view in the array, the algorithm adjusted each point cloud relative to its previous neighboring view using rotation and translation matrices, such that all views formed a single view. point cloud, the system was evaluated through the statistical analysis of three characteristic performance parameters for volumetric scanning systems: response speed, precision and reliability. The reconstruction system presented a mean reconstruction time of 12.84 seconds, for precision it was compared with the COLMAP software, which by means of a t-student test of 30 paired samples showed that there is no significant difference between the means of these samples. In terms of reliability, the system showed that it is not accurate, resulting in an error of 2 mm on average. It is concluded that the system is not precise as in other reconstruction methods due to the stereo technique, but it can be used for applications that do not require a high level of precision. The use of other active techniques such as laser triangulation is recommended.

Keywords: <STEREO RECONSTRUCTION>, <SEMI-GLOBAL BLOCK MATCHING (SGBM)>, <STEREO CALIBRATION>, <POINT CLOUD>, <MULTI-VIEW RECONSTRUCTION>, <MYNT CAMERA (HARDWARE)>, <TRIANGULATION>, <EPIPOLAR GEOMETRY >.



MSc. Wilson G. Rojas

C.I 0602361842

INTRODUCCION

El siguiente trabajo de investigación “Diseño e implementación de un sistema de reconstrucción volumétrica virtual de objetos en un entorno indoor mediante cámaras estereoscópicas y procesamiento digital de imágenes” es el desarrollo de un prototipo funcional de escaneo de objetos que mediante procesamiento de imágenes obtenidas por medio de cámaras estereoscópicas y técnicas de reconstrucción volumétrica, realiza la digitalización y construcción en tres dimensiones del objeto, la característica principal del sistema es la obtención de nube de puntos capaces de recrear objetos colocados dentro del sistema, el sistema desarrollado de múltiples vistas posee la capacidad de obtener datos de cada una de las caras y luego unirlos para formar un solo objeto tridimensional. El interés académico de realizar este proyecto es el de profundizar en el procesamiento de imágenes, así también como el de aportar ideas sobre algún futuro trabajo relacionado a la obtención de características por medio de la estereoscopia. El método utilizado para la realización del sistema de reconstrucción fue obtenido mediante la revisión bibliográfica sobre técnicas de triangulación, métodos de calibración de cámaras estéreo, herramientas y funciones de procesamiento de imágenes orientado a la visión estereoscópica, siguiendo la metodología de cascada. Este trabajo está estructurado partiendo del planteamiento del problema en el capítulo uno. La revisión sobre el estado del arte sobre las diferentes técnicas, avances tecnológicos y herramientas en el desarrollo de la reconstrucción volumétrica, en el capítulo dos. En el capítulo tres, proponemos y diseñamos el sistema, así como su estructura y sus componentes. En el capítulo cuatro, validamos el sistema completo verificando el funcionamiento por medio de los resultados.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

1.1 Antecedentes

Apenas 7 años después de la invención de la fotografía en 1833 el físico británico Wheatstone creó el primer aparato estereoscópico (Castro, 1840). Capaz de proporcionarle a nuestro cerebro dos imágenes similares, pero con pequeñas diferencias, que se conoce como disparidad, tal y como funcionan nuestros ojos en nuestro cerebro, al reflejar dos imágenes en 2D en dos espejos que reflejaran cada uno a un ojo, y así conseguir emular la visión binocular humana. Años más tarde en 1849 David Brewster diseñó y construyó la primera cámara estereoscópica, capaz de obtener fotografías en relieve (Epelbaum, 2010, p. 64). Considerando la importancia e impacto que tiene el conocimiento de las imágenes estereoscópicas, ahora se han desarrollado una serie de aplicaciones basadas en los diferentes métodos involucrados, como la reconstrucción de la tierra en tres dimensiones ha logrado avances significativos en los últimos años debido a la importancia que tiene. La recopilación de modelos 3D de escenas, a partir de imágenes capturadas por cámaras. Los avances en visión artificial han hecho que este procedimiento sea cada vez más flexible y preciso, a pesar de que es un proceso complejo (Cárdenas Quiroga, Morales Martín y Ussa Caycedo, 2015, p. 203). Los excelentes avances e investigaciones realizadas en el campo de la fotometría han propiciado su uso en otros campos científicos, donde se ha reconocido que puede realizar importantes aportes a problemas específicos del organismo. Esto ya está sucediendo en el campo médico, más específicamente en los campos de la telemedicina y la imagen médica. Estos campos requieren análisis ambientales y de imágenes automatizados seguros y de alta calidad, requisitos que la medición óptica puede cumplir (Cárdenas Quiroga, Morales Martín y Ussa Caycedo 2015, p. 204). El resultado de la combinación de estas tecnologías junto con algoritmos de corrección y calibración es la generación de objetos 3D los cuales pueden ser utilizados para elaboración de mapas tridimensionales, medición del volumen de objetos, replicas tridimensionales de piezas para la industria, hay muchas aplicaciones para esta tecnología de las cuales las más usadas son la medicina, la industria y el cine (Portella Coaguila, 2015, p. 12).

1.2 Formulación del problema

¿Cómo Diseñar e implementar un sistema de reconstrucción volumétrica virtual de objetos en un entorno indoor mediante cámaras estereoscópicas y procesamiento digital de imágenes?

1.2.1 Sistematización del problema

¿Cómo revisar y elaborar un estado del arte sobre sistemas de reconstrucción 3D de objetos?

¿Cómo diseñar un arreglo de cámaras estereoscópicas de al menos 4 cámaras, para la captura de imágenes de modo que se visualice por completo el objeto en un entorno indoor?

¿Cómo analizar las imágenes estereoscópicas obtenidas para realizar una configuración adecuada de los parámetros de operación de las cámaras estereoscópicas?

¿Cómo realizar el procesamiento digital de las imágenes obtenidas por el arreglo de cámaras para la reconstrucción volumétrica del objeto?

¿Cómo implementar el sistema de reconstrucción volumétrica virtual de objetos?

¿Cómo evaluar el correcto funcionamiento del sistema de reconstrucción volumétrica de objetos?

1.3 Justificación

1.3.1 Justificación teórica

Actualmente, la reconstrucción tridimensional es un campo muy utilizado en ciencia e ingeniería, que es muy útil en el modelado y visualización de geometría para aplicaciones científicas. Debido a que los dispositivos comerciales para la reconstrucción tienen un plano láser o una fuente de haz, lo que indica su alto costo, estos dispositivos se utilizan solo en organizaciones de gran presupuesto como museos, estudios de arqueología o arquitectura (Portella Coaguila, 2015, p. 13). Por esta razón, la reproducción de volumen basada en sistemas estéreo con dos o más cámaras según sea necesario, se desarrollará en una configuración convergente para que las cámaras se enfoquen en el objeto de interés. Los perfiles de convergencia proporcionan un espacio de trabajo adecuado para que cuando el objeto esté cerca, la convergencia de los ejes se produzca sobre dicho objeto (Portella Coaguila, 2015, p. 8). En el caso de (Lee, Huang y Lin, 2012, p. 59) realizan reconstrucción de modelos 3D poco densos utilizando múltiples vistas, utilizando cámaras simples en lugar cámaras estéreo, y lo complementan con una técnica llamada estructura a partir de movimiento (SfM por sus siglas en inglés), la cual presenta problemas muy similares a los encontrados en el uso de cámaras estéreo, ya que ocupa imágenes consecutivas para la detección de la profundidad. Una

ventaja es que no necesitan cámaras estéreo ya que SfM funciona con cámaras simples, por otro lado, la desventaja es que necesitan algunas docenas de imágenes para realizar la reconstrucción, además utilizar cámaras estéreo debería de mejorar la precisión de la reconstrucción, ya que se genera un aumento en la información disponible. Por otro lado, (Hernández y Schmitt, 2003, p. 3-5) emplean el algoritmo de los modelos deformables (snakes) y adicionalmente hacen uso de cámaras estéreo, junto con una secuencia de imágenes a color calibradas. Los snakes ofrecen la ventaja de que son fáciles de implementar y de ajustar sus parámetros. Finalmente, (Eisert, Steinbach y Girod, 2000, p. 264) plantearon un sistema que utiliza una imagen estéreo y múltiples vistas para generar los modelos 3D, se puede afirmar que, de manera general, hace uso de cámaras sin calibrar con un objeto de referencia, genera una imagen estéreo y las vistas (imágenes en distintas posiciones) que son agregadas mediante SfM para ajustar la forma y la textura. La reconstrucción mediante múltiples vistas es una variación de la reconstrucción "simple", en este caso se intentan recuperar tanto la forma como la apariencia de un objeto a partir de un conjunto de imágenes del objeto, en la ilustración 1-1 se observa la topología de un arreglo de múltiples vistas.

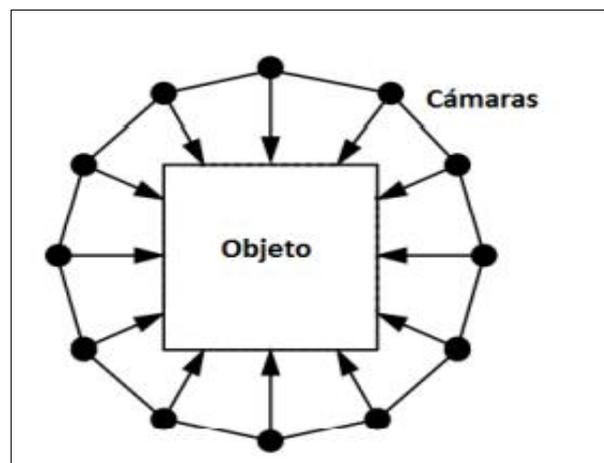


Ilustración 1-1: Topología de arreglo multi-vista.

Fuente: (Wong Quijano, 2013, p. 12)

En esta técnica se pueden utilizar 2, 3, 4 o tantas lentes como se considere necesario. En el caso de utilizar sólo dos lentes el sistema se conoce con el nombre de sistema estereoscópico y surge de la idea de simular el sistema visual humano compuesto por dos ojos (Torres Civit, 2007, p. 3). La reconstrucción Volumétrica consiste en obtener las coordenadas respecto a la cámara de un punto a partir de su par conjugado. Las coordenadas X y Y se obtienen al seleccionar una de las imágenes para el caso de una sola imagen (en el caso de estéreo la convención es utilizar la imagen izquierda), mientras que la coordenada Z , Ilustración 2-1, es calculada mediante el mapa de disparidad obtenido de la búsqueda de correspondencia. Una vez que se une la información de las 3 coordenadas se obtiene un modelo 3D de la escena (Wong Quijano, 2013, p. 16-17). Aunado a lo

anterior se sabe que la disparidad es inversamente proporcional a la distancia, lo que nos permite medir con mayor precisión la distancia que existe entre la cámara y los objetos cercanos a los objetos lejanos.

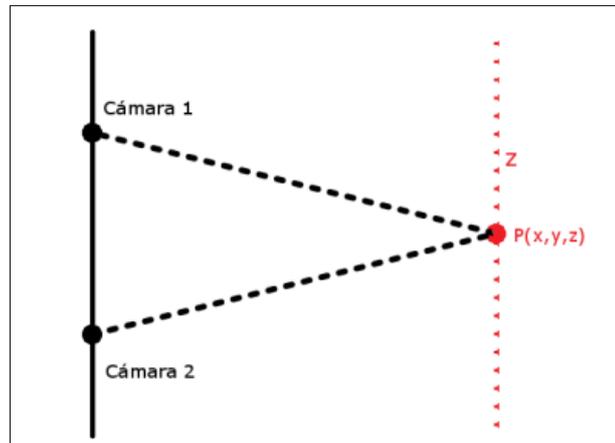


Ilustración 2-1: Coordenada Z a partir de un par de cámaras.

Fuente: (Wong Quijano, 2013, p. 12)

Dicha disparidad es la que nuestro cerebro utiliza a fin de estimar la distancia relativa de los objetos observados, este proceso consta de tres pasos: Primero: Una ubicación específica de una superficie en particular debe ser seleccionada en la primera imagen. Segundo: La misma ubicación de la misma superficie debe ser identificada en la otra imagen. Tercero: La disparidad entre los puntos correspondientes en ambas imágenes debe ser medida (Wong Quijano, 2013, p. 26). En la ilustración 3-1 se observan dos imágenes correspondientes a la vista izquierda y derecha de destornillador.

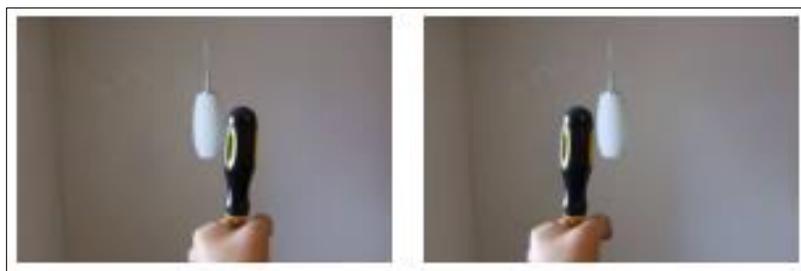


Ilustración 3-1: Imagen izquierda y derecha de cámara estéreo.

Fuente: (Wong Quijano, 2013, p. 12)

1.3.2 Justificación aplicativa

En contraste, la presente propuesta utiliza un método de calibración de las cámaras y la información de todas las vistas provendrá de imágenes estéreo, partiendo del supuesto de que el

uso de múltiples imágenes estéreo debería proporcionar un aumento en la información obtenida. La presente tesis está enfocada en la creación de modelos 3D de objetos, mediante la unión de múltiples mapas de disparidad a fin de poder reconstruir un objeto rígido y opaco utilizando la información proveniente de varios puntos de vista, haciendo uso del color y/o la textura de este siempre que sea posible. El utilizar múltiples vistas permite una reconstrucción más completa del objeto. Primero el diseñar un sistema de reconstrucción volumétrica por medio de la utilización de al menos cuatro cámaras Mynt eye s1030 que son capaces de capturar imágenes al mismo tiempo por medio de sus dos lentes, con una resolución de 752x480 a 60 FPS se pueden obtener imágenes claras incluso en ambientes de poca luz. El Angulo de visión de estas cámaras en cada eje es de 146, 122 y 76 grados horizontal y vertical lo que permite tener un aplico enfoque en objetos pequeños y medianos, además de su gran alcance desde el medio metro hasta las 18 metros. Tres de estas cámaras estarán dispuestas geométricamente de forma vertical con el objeto de modo de que se observe del objeto por completo, para lograr observar con claridad el objeto se encontrarán parámetros como la distancia entre las cámaras, distancia entre las cámaras y el objeto, y entre otros parámetros geométricos. La cuarta cámara será colocada en la parte superior del objeto, esta disposición del arreglo de cámaras se ha pensado con el objetivo de capturas todas las superficies del objeto, aunque queda de manifiesto la utilización de más cámaras según sea necesario, el diseño del arreglo dependerá de los objetos a reconstruir. Segundo: analizar las imágenes obtenidas mediante el arreglo para configurar los parámetros de operación de las cámaras como resolución, claridad, calibración, triangulación y otros parámetros que permitan la captura de imágenes de excelente calidad. Tercero: realizar el procesamiento digital de las imágenes por medio de un algoritmo desarrollado en Matlab o en la herramienta propia del fabricante Mynt de las cámaras con el propósito de calcular el mapa de disparidad entre pares estéreo, con esto obtendremos la profundidad de los objetos escaneados, también el de generar nubes de puntos a partir de mapas de disparidad, en este proceso se unirán todos los mapas de disparidad provenientes de todas las cámaras, se segmentará los mapas tomando lo que se necesita en base al objeto. Y se unirán las nubes de puntos para crear un modelo en 3D del objeto, de acuerdo con objetos fijos, aunque en base a los resultados se podría realizar la reconstrucción de objetos en tiempo real y en movimiento. Cuarto: Implementar el sistema de reconstrucción volumétrica de objetos mediante la unión de los procesos anteriores: la implementación del arreglo de cámaras, para mantener las cámaras fijas se elaborará alguna estructura sólida para asegurar la integridad el arreglo de cámaras, el análisis de las imágenes en conjunto con el procesamiento digital dará como resultado un sistema de reconstrucción volumétrica virtual de objetos listo para ser probado. Y, por último, evaluar el correcto funcionamiento del sistema de reconstrucción virtual de objetos, de esta manera se podrá comprobar que el diseño y la

implementación de nuestro sistema es funcional, para realizar la evaluación se tomaran varios objetos y se digitalizaran, se observara su estructura virtual y se harán cambios en base a los resultados. En la ilustración 4-1, se presenta el diagrama de flujo de los procesos a realizar para para la reconstrucción volumétrica de los objetos comenzado desde la obtención de los datos (imágenes) hasta la evaluación de los objetos reconstruidos.

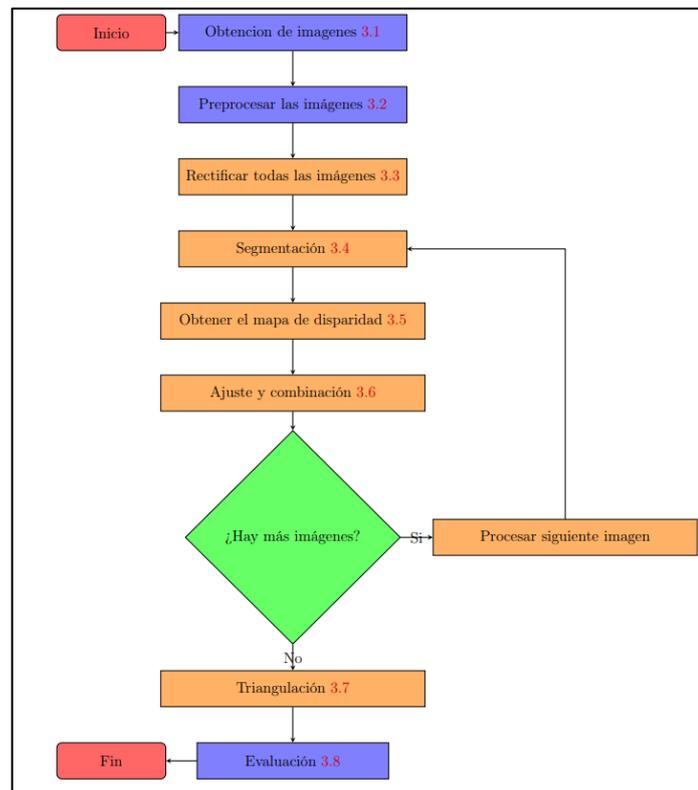


Ilustración 4-1: Diagrama de flujo de algoritmo de digitalización estéreo.

Fuente: (Quijano, 2013, p. 17)

La estereoscopia tiene muchas aplicaciones y la reconstrucción volumétrica es una de ellas, con esta podremos determinar la forma de objetos difíciles de digitalizar de una manera más sencilla, ahorrando tiempo y recursos humanos.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar e implementar un sistema de reconstrucción volumétrica virtual de objetos en un entorno indoor mediante cámaras estereoscópicas y procesamiento digital de imágenes.

1.4.2 Objetivos específicos

- ❖ Revisar y elaborar un estado del arte sobre los métodos y tecnologías de reconstrucción volumétrica.
- ❖ Diseñar un arreglo de cámaras estereoscópicas de al menos 4 cámaras, para la captura de imágenes de modo que se visualice por completo el objeto en un entorno indoor.
- ❖ Analizar las imágenes estereoscópicas obtenidas para realizar una configuración adecuada de los parámetros de operación de las cámaras estereoscópicas.
- ❖ Realizar el procesamiento digital de las imágenes obtenidas por el arreglo de cámaras para la reconstrucción volumétrica del objeto.
- ❖ Implementar el sistema de reconstrucción volumétrica virtual de objetos.
- ❖ Evaluar el correcto funcionamiento del sistema de reconstrucción volumétrica de objetos.

CAPÍTULO II

2. ESTADO DEL ARTE

2.1 Digitalización 3D

La digitalización 3D según (Juárez, 2022) es el proceso de escaneo de un objeto real para convertirlo en un objeto digital, por medio de la tecnología de escáner 3D, permitiendo obtener información del color, la textura y la geometría del objeto. (León, 2003) plantea la digitalización de manera general como una técnica de adquisición de medidas tridimensionales que permiten reconstruir modelos CAD a partir de objetos físicos. Hoy en día son muchas las aplicaciones de la digitalización 3D en campos como la medicina, la educación, la industria y el entretenimiento. (Morón A, 2018, p. 13) recalca que las cámaras son distintas a los escáneres 3D, las primeras obtienen una representación 2D de la escena, mientras que los segundos obtienen información de la forma geométrica de la escena.

2.2 Técnicas de reconstrucción 3D

En la actualidad existen algunos métodos de reconstrucción 3D de objetos. Estos pueden clasificarse en dos tipos, los basados por técnicas de contacto y los de sin contacto. A su vez, los métodos basados sin contacto pueden clasificarse como técnicas activas y pasivas. En la ilustración 1-2 se observa esta clasificación.

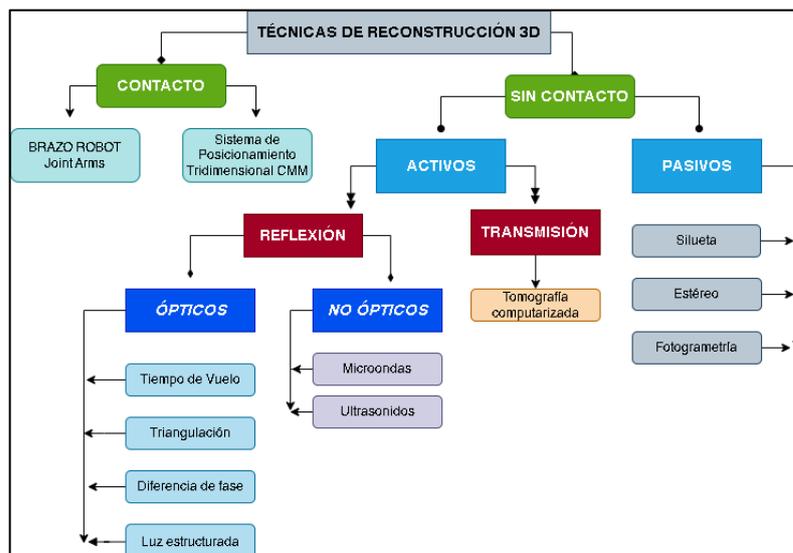


Ilustración 1-2: Técnicas de reconstrucción 3D.

Fuente:(Rocchini et al. 2001, p. 3)

Realizado por: Castro, Gabriel; 2023

2.2.1 Técnicas de reconstrucción por contacto

Estas son de las técnicas más antiguas, por el avance y desarrollo de otras técnicas más rápidas, los métodos por contacto ya no se utilizan mucho. Su funcionamiento radica en la obtención de datos 3D por medio de un palpador que recorre por la superficie del objeto, el palpador por lo general es una punta de acero. Un grupo de sensores permiten establecer la posición del palpador y obtener sus coordenadas tridimensionales. Una máquina de medición por coordenadas (CMM) y un brazo robótico son ejemplos de esta técnica. En la ilustración 2-2 se muestra un brazo articulado para medición de coordenadas.

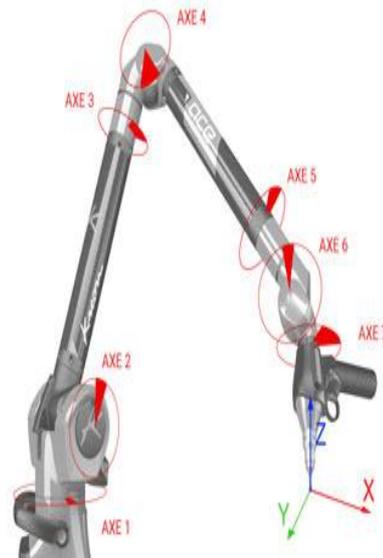


Ilustración 2-2: Brazo de medición articulado.

Fuente: (Kreon, 2022)

La estructura de un brazo robótico articulado se basa en la morfología de un brazo humano con sus articulaciones en muñecas, hombro, codo y segmentos como el antebrazo y el brazo. Las posiciones angulares y las longitudes geométricas del brazo devueltas por los decodificadores presentes en cada eje permiten determinar la posición exacta del palpador en forma de coordenadas XYZ. Estos brazos también son conocidos como poli articulados y CMM portátiles (Kreon, 2022).

Una máquina de medición por coordenadas al igual que un brazo articulado realiza la medición por medio de contacto con el objeto. El sistema de palpado se mueve a lo largo de tres ejes coordenados, estos ejes son ortogonales entre sí. Cada eje tiene un sensor que revisa la posición de la punta mientras esta se mueve por un objeto (Creaform, 2021). En la ilustración 3-2 se muestra una máquina CMM.



Ilustración 3-2: Máquina de medición por coordenadas.

Fuente: (Vázquez, 2015)

Una desventaja de las CMM es el daño que le ocasiona al objeto por fricción al contacto. Otra desventaja es la velocidad de escaneo baja, al tener que posicionar la sonda por todo el objeto (Arriaga Hernandez, 2013).

2.2.2 Técnicas de reconstrucción sin contacto

Las técnicas de reconstrucción sin contacto se basan en la relación entre el objeto y algún tipo de energía (luz, laser, infrarrojo, rayos x, ondas de radio, sonido). Este método puede subdividirse en activas y pasivas. Las primeras emiten alguna clase de energía o radiación, la obtención de la geometría del objeto se logra analizando la reflexión o transmisión de dicha energía por la superficie del objeto. Las segundas no emiten ningún tipo de energía, pero se enfocan en analizar la luz o energía reflejada del ambiente sobre el objeto. Las técnicas basadas en activas suelen ser más precisas, pero de mayor costo (Morón A, 2018, p. 16).

2.2.2.1 Técnicas de reconstrucción activas

Una de las ventajas más notables de las técnicas activas es que al no ser invasivas no dañan ni alteran la forma del objeto real, estas técnicas se caracterizan por transmitir alguna radiación o energía y posterior el análisis del comportamiento de esta en el objeto.

2.2.2.1.1 Tiempo de vuelo

Método de reconstrucción activa que está conformado de dos elementos principales, un generador de pulsos de luz (laser), y un medidor de rango (Morón A, 2018, p. 17). De manera general, funcionan enviando un haz de luz que generalmente suele ser un infrarrojo para evitar distorsión con la luz natural y entonces medir para cada píxel el tiempo de transmisión y de recepción (Hornberg, 2017, p. 1190). La ventaja de esta técnica es que ofrecen mapas de profundidad completos del objeto con cada toma a una velocidad de fotogramas altos. El sistema realiza la obtención de puntos 3D mediante pulso y no por barrido, lo que implica que el sistema medirá la distancia punto a punto, y el barrido se realiza moviendo el emisor (BIMND, 2020). Este tipo de escáneres pueden medir la longitud de hasta 100000 puntos por segundo, esta técnica es empleada para la reconstrucción de grandes edificios y entornos. En la ilustración 4-2 se muestra la geometría del sistema. Un ejemplo de esta técnica es el escáner LIDAR.

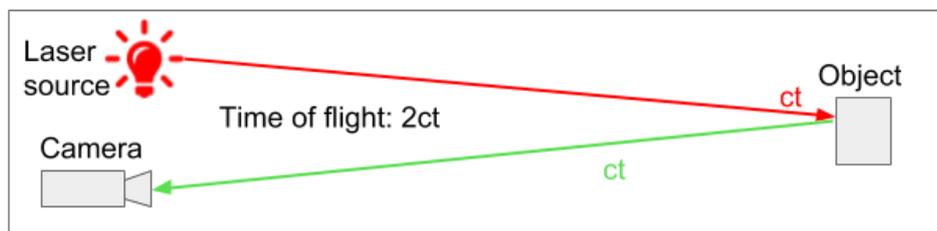


Ilustración 4-2: Funcionamiento de sensor basado en tiempo de vuelo.

Fuente: (Bitfab, 2020)

2.2.2.1.2 Escáner laser por triangulación.

Este método de reconstrucción se hace uso de una fuente de luz (laser) y un sensor (cámara) cuyos ejes no están alineados, entre ellos forman un ángulo llamado ángulo de triangulación. Su funcionamiento consiste en la proyección de luz sobre el objeto y el sensor recoge la luz reflejada sobre el objeto. La dirección de propagación del rayo es conocida, para obtener la posición en los tres ejes coordenados del punto, es necesario la búsqueda de la distancia a la que le rayo incide sobre la superficie del objeto. Esta distancia se determina con la lectura del fotodetector (sensor) y el ángulo de triangulación (Cuesta et al., 2014, p. 2). La posición del emisor, el objeto, y el receptor forman un ángulo, de allí el nombre de triangulación. La velocidad de este método es mejor que los métodos anteriores, el láser genera una línea vertical u horizontal en vez de un solo punto que barre la superficie del objeto. La precisión de esta técnica está en orden de los micrómetros por lo que es muy útil para aplicaciones que requieren gran precisión. En la ilustración 5-2 se muestra la geometría del sistema.

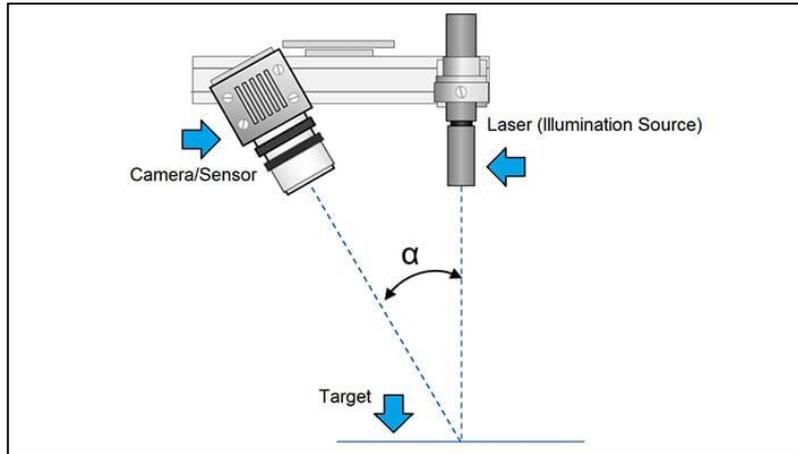


Ilustración 5-2: Esquema de triangulación por láser.

Fuente: (Contreras, 2019)

2.2.2.1.3 Escáner de luz estructurada.

Este método al igual que la triangulación laser también hace uso de la técnica de triangulación con la diferencia es que en vez de una línea vertical u horizontal o un punto funciona proyectando un patrón de luz sobre el objeto a reconstruir. Esto mejora sustancialmente el tiempo de reconstrucción ya que, al no ser solamente una línea o un punto, la reconstrucción se puede realizar mediante una única toma del objeto. Según (Hornberg, 2017, p. 1193) los objetos deben presentar ciertas características como reflectancia, no completamente negros pero si homogénea y difusa, las superficies no deben ser pronunciadas debido a que provocan distorsión del patrón proyectado. En la ilustración 6-2 se observa la geometría del escáner por luz estructurada.

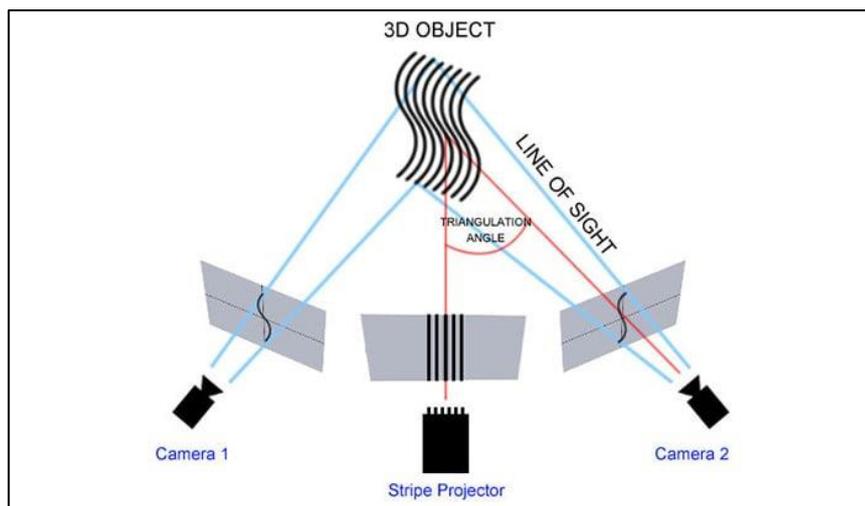


Ilustración 6-2: Geometría de luz estructurada.

Fuente: (Contreras, 2019)

El patrón proyectado es captado por los sensores, estos sensores se encuentran desplazados con respecto al proyector, las cámaras captan la deformación de la luz provocada por la superficie del objeto, analizando la deformación con algunas imágenes del objeto se puede obtener el modelo tridimensional. Una de las limitaciones de esta técnica es la sensibilidad a la luz por lo que se debe realizar en un entorno controlado.

2.2.2.1.4 Escáner por diferencia de fase

Según (Golubeva, 2022) los escáneres basados en desplazamiento de fase envían un haz de luz laser a frecuencias altas, la búsqueda de la distancia al objeto se determina midiendo la diferencia de fase entre la señales transmitidas y recibidas. A diferencia de los escáneres convencionales, estos funcionan a distancias más reducidas de 80 a 120 metros como máximo, normalmente de 1 a 50 metros. En la ilustración 7-2 se muestra el esquema de este método.

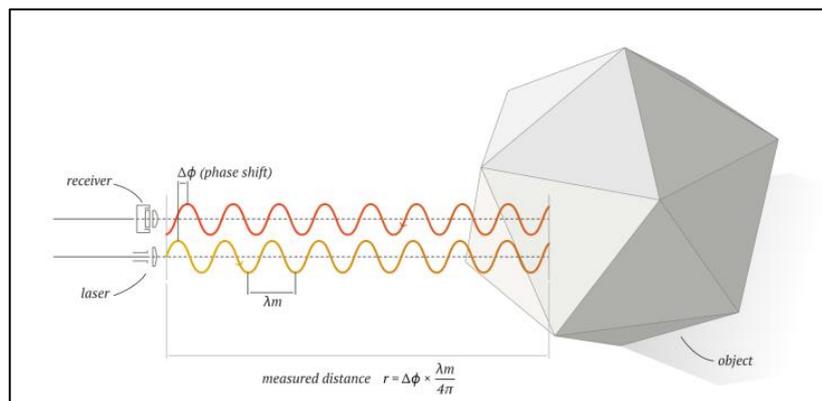


Ilustración 7-2: Esquema de escáner por diferencia de fase.

Fuente: (Golubeva, 2022)

Este sistema suele ser el más rápido, de hasta un millón de puntos por segundo. También ofrecen mayor precisión que los basados en tiempo de vuelo.

2.2.2.1.5 Tomografía computarizada

Es un método muy utilizado en medicina, es un método de diagnóstico muy efectivo que revolucionó la medicina, permite al médico examinar las partes internas del cuerpo de forma no invasiva y con precisión (Restrepo, calle y Gutiérrez, 2019). Esta técnica permite la obtención de imágenes 2D mediante rayos X del cuerpo y luego se superponen como capas para formar un objeto 3D.

En la ilustración 8-2 se muestra una imagen 3D obtenida mediante tomografía computarizada (TC) multi corte de 16 canales.

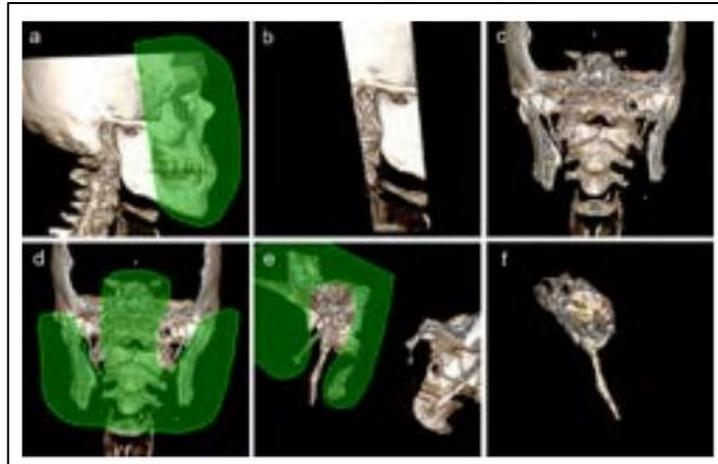


Ilustración 8-2: Reconstrucción 3D del cráneo mediante TC.

Fuente: (Hernández-Díaz, Hernandez y Ortega 2021, p. 3)

2.2.2.1.6 *No ópticas*

Los métodos basados en ultrasonidos para uso general proporcionan imágenes bidimensionales de los tejidos blandos sin someter a los pacientes a radiación ionizante en medicina. Los escáneres ultrasónicos se refieren a dispositivos que emiten ondas sonoras a frecuencias por encima del rango de la audición humana, frecuentemente entre 2 y 15 MHz.

2.2.2.2 *Técnicas de reconstrucción pasivas*

Los métodos de reconstrucción pasiva no utilizan alguna fuente de luz dentro de su sistema, estos hacen uso de varias características diferentes de la escena sobre el objeto a reconstruir (Hornberg 2017, p. 1188). Una de las ventajas es que son muy baratos con respecto a las técnicas activas, debido a que no necesitan hardware muy específico y costoso (Manteca Fernández, 2018, p. 11).

2.2.2.2.1 *Fotogrametría*

También conocida como estructura desde el movimiento (SFM), es una técnica muy buena para obtener datos tridimensionales del objeto, sea que el objeto este o no en movimiento. Su funcionamiento se basa en analizar varias imágenes consecutivas del objeto, y analizar las

diferencias entre estas. Este método busca puntos comunes en todas las imágenes del objeto, para posteriormente realizar su reconstrucción, pero presenta una desventaja considerable, y es que requieren de una alta capacidad de cómputo y procesamiento de imágenes para encontrar las referencias de cada imagen y que estas se combinen para formar un solo objeto 3D (Manteca Fernández, 2018, p. 12,13). En la ilustración 9-2 se muestra el esquema de SFM.

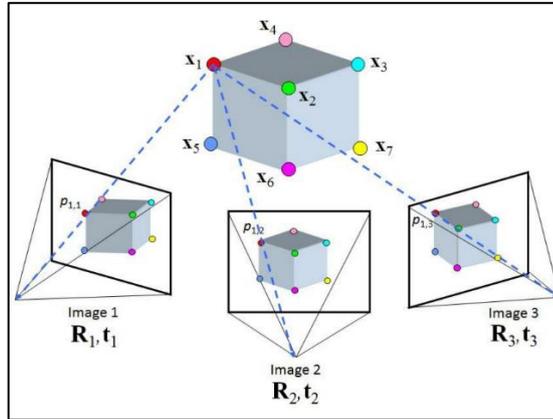


Ilustración 9-2: Esquema técnico sfm.

Fuente: (Yilmaz y Karakus 2013, p. 22)

2.2.2.2.2 Profundidad desde estéreo

Es una técnica que se basa en la toma de imágenes mediante dos cámaras, conocida la distancia entre estas, el funcionamiento radica en buscar los mismo característicos del objeto en ambas cámaras y triangular para encontrar la coordenada faltante z, se basa en la búsqueda de la disparidad que es la resta de las proyecciones de dichos puntos sobre un plano común. A partir de esta disparidad y la distancia entre las cámaras y un parámetro llamado distancia focal es posible el cálculo de la profundidad para cada pixel (Manteca Fernández, 2018, p. 11) y (Hornberg, 2017, p. 1189). En la ilustración 10-2 se muestra el esquema estero.

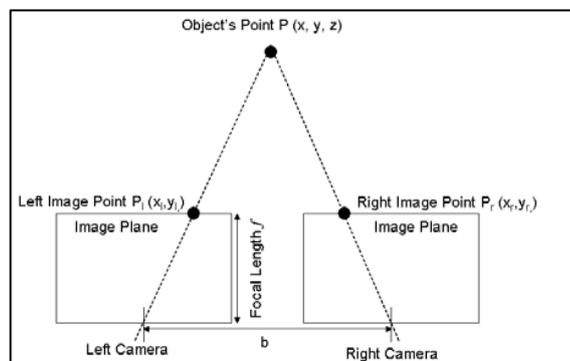


Ilustración 10-2: Esquema estéreo cámara.

Fuente:(Saidon et al. 2009, p. 4)

2.3 Trabajos relacionados.

La reconstrucción tridimensional a partir de dos imágenes ha sido uno de los campos de investigación desde los 80s. Los avances en el ámbito de la visión estéreo han permitido el establecimiento de restricciones con el fin de mejorar los resultados de reconstrucción, pero sin embargo por la complejidad de la técnica no existe método exacto para la obtención de mapas de mapas de profundidad precisos. En los inicios de la investigación sobre la obtención de mapas tridimensionales, estos eran obtenidos mediante métodos basados en áreas y en características como en el caso de (Tardón García, 1999, p. 6). Posterior surgen nuevos métodos de correspondencia de puntos basados en métodos globales que presentan mejores resultados ayudándose de las texturas de los objetos para mejorar la correspondencia como es el caso de (Jiménez Medina, 2008, p. 46), en (Martínez, 2010, p. 79-80) se realizó la búsqueda de un método de coincidencia que mejor resultado presente para la obtención de la disparidad y concluye que los métodos de energía global y los basados en segmentación y medidas de similitud son los que mejor mapa de disparidad presentaron.

A partir de aquí, trabajos como el de (Wong Quijano, 2013, p. 42) que propusieron la reconstrucción de objetos mediante el uso de múltiples vistas, haciendo el uso de múltiples cámaras estéreo con el supuesto de que cada cámara estéreo aportaría información tridimensional importante, como una secuencia de nube de puntos que aportan datos al objeto final de reconstrucción. También están trabajos como el de (Jáuregui Elso, 2018, p. 38,49) que se planteó la evaluación de precisión de un sistema de reconstrucción basado en estéreo, pero con un método de correspondencia semi-manual, los resultados presentaron un error máximo de 1.606 mm que comparados con un sistema de CMM son altos, pero bastante considerables para aplicaciones que no requieran de un nivel de precisión de decima de milímetro. En (Kapeller, 2018, p. 79) se propuso la evaluación de un sistema de reconstrucción formado por tres cámaras estero, los algoritmos de filtrado de volumen de costo (CVF) y *PatchMatch* presentaron los mejores mapas de profundidad arrojando resultados comparables en términos de precisión del modelo de malla texturizada, el sistema fue capaz de reconstruir objetos esféricos con una precisión del 90 %, algunas imperfecciones se mostraron en áreas pequeñas de los bordes de los objetos. (Saquin, 2019, p. 107) y (Basulto Lantsova 2020, p. 59) desarrollaron un sistema de reconstrucción para objetos debajo del agua y láminas de metal, ambos concluyeron que SGBM presentaba el mejor resultado en cuanto a mapa de disparidad se refiere, los dos hicieron uso de cámaras de la marca ZED, utilizando librerías de OpenCV en el software Matlab.

2.4 Visión Estereoscópica

La visión estereoscópica es la capacidad de combinar las imágenes percibidas por cada uno de nuestros ojos en una sola imagen tridimensional con profundidad, luminosidad y relieve, un proceso complejo en el cerebro que toma dos imágenes diferentes y las analiza. Los ojos de una persona ocupan diferentes posiciones en la cabeza. Por eso, no se observa lo mismo en la vista de cada uno de nuestros ojos. La diferencia que se produce entre dos imágenes se denomina desajuste binocular. El cerebro crea a partir de estas dos imágenes una imagen tridimensional de profundidad y relieve. (Oftalvist, 2019, p. 1). En la ilustración 11-2 muestra la descripción cómo cada ojo interpreta el mismo escenario 3D y finalmente produce la imagen final.

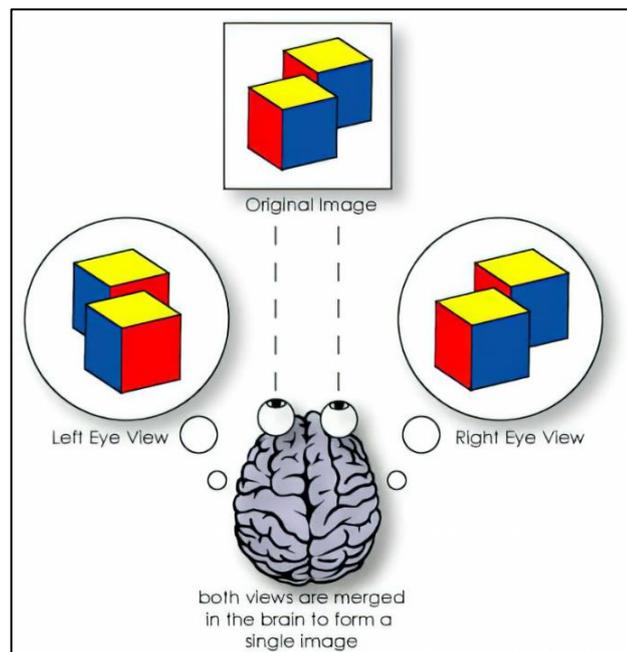


Ilustración 11-2: Descripción de la visión binocular.

Fuente: (Revopoint, 2019)

2.5 Perspectiva y sensación de profundidad.

Cuando el cerebro necesita más información de la que pueden proporcionar los ojos individuales, este hace uso de un recurso para interpretar correctamente la imagen tridimensional: es decir la perspectiva. El recurso de perspectiva es el uso de la memoria visual y la diferencia de objetos y texturas de imágenes para intentar reconstruirlos y colocarlos en un espacio 3D. El espacio es tridimensional. Esto significa que necesita conocer las tres direcciones principales (x, y, z) para capturar el espacio correctamente. La sensación de profundidad se logra cuando el cerebro es capaz de interpretar las imágenes entrantes y distribuir sus partes en estas tres dimensiones (Català

Sarmiento, 2014, p. 14). La ilustración 12-2 muestra una representación de la perspectiva de una imagen.

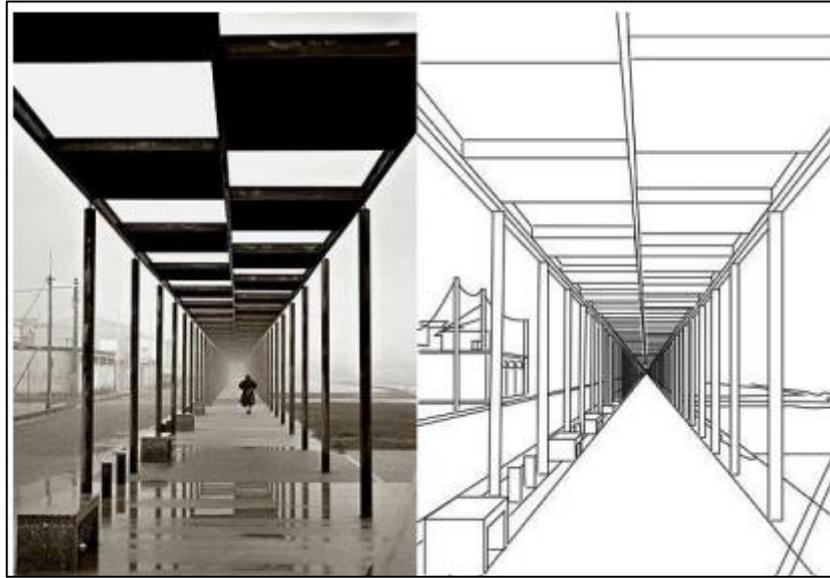


Ilustración 12-2: Descripción de la perspectiva de una imagen real.

Fuente: (Català Sarmiento 2014, p. 14)

Las diferentes claves para la percepción de la profundidad son conocidas y se pueden dividir en dos subgrupos:

2.5.1 Claves monoculares: percibidas por cada ojo individualmente.

- **Perspectiva:** consiste en calcular la distancia tomando como referencia la dirección de las líneas de objetos comunes y como estas se van juntando con la distancia.
- **Tamaño relativo:** se basa en objetos de la misma talla, los más lejanos aparecen más pequeños, así como los más cercanos se aprecian más grandes.
- **Interposición:** un objeto que tapa parcialmente a otro se da por supuesto que está por delante.
- **Sombras y tonos:** la relación con la fuente de luz y las sombras dan muchas pistas sobre la colocación de un objeto en el espacio.

2.5.2 Claves estereoscópicas: percibidas combinando los dos ojos, visión binocular.

- **Disparidad retinal:** consiste en analizar la diferencia de ángulo mediante el cual una imagen entra por cada ojo.
- **Convergencia:** clave ofrecida por el mecanismo de enfoque del ojo. Según se enfoca un objeto, de más cerca o más lejos, se consigue una pista de su situación.

Estas claves manifiestan el armado de los objetos en el espacio 3D (Català Sarmiento, 2014, p. 15). La ilustración 13-2 presenta la percepción de cada ojo y como la unión de las dos imágenes crea una tridimensionalidad.

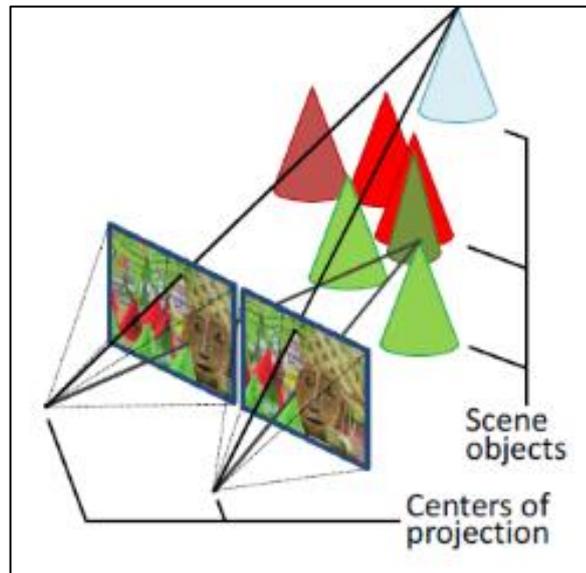


Ilustración 13-2: Descripción de la perspectiva de una imagen real.

Fuente: (IDS 2014, p. 1)

2.6 Concepto de cámaras 3D

Una cámara 3D es un dispositivo de imagen que permite la percepción de profundidad de una imagen para reproducir las tres dimensiones percibidas por la visión binocular humana. Algunas cámaras 3D usan dos o más lentes para capturar múltiples puntos de vista, mientras que otras usan una sola lente que puede cambiar su posición. La combinación de dos puntos de vista, como el punto de vista ligeramente diferente de dos ojos humanos, proporciona una percepción de profundidad. La fotografía en 3D le permite capturar un momento congelado en el tiempo para crear una imagen fija o un video lo suficientemente realista como para tocarlo (Krypton, 2021).

2.7 Cámara Estéreo

Una cámara estereo es una combinación de dos cámaras a una distancia conocida con un sensor de imagen separado para cada lente. Se utilizan debido a su capacidad para reproducir escenas en 3D mediante triangulación estereo (Cappellaro, 2019, p. 2).

2.8 Mynt Eye S1030-IR-120/Mono

La MYNTEYE S1030-IR-120/Mono es una cámara estéreo monocromática visible e infrarroja (IR), con una unidad de medición inercial (IMU) integrada de 6 ejes y un ángulo visual alto. Tiene un obturador global y las cámaras están sincronizadas por hardware. El producto proporciona un kit de desarrollo de software (SDK) compatibles con los sistemas operativos Ubuntu 16.04 y Windows, permiten interactuar con la cámara y registrar datos usando los paquetes ROS proporcionados. El Robot Operating System (ROS) es "un meta-sistema operativo de código abierto" para robots. Proporciona los servicios que normalmente brinda un sistema operativo y un conjunto de herramientas para ejecutar código en múltiples máquinas (Cappellaro, 2019, p. 2). La cámara puede ser utilizada para la detección de profundidad precisa con un rango flexible entre 0,5 y 18 metros. Rendimiento optimizado en condiciones normales de luz o condiciones de poca luz. Precisión con un amplio campo de visión a 146 grados. La IMU de seis ejes combinada con la sincronización de fotogramas proporciona una precisión de menos de un milisegundo. Consta de un paquete completo con SDK simple de integrar. Fácil desarrollo y rápida integración con los datos de profundidad creados a través del sensor EYE S. La combinación de la amplia línea de base y el rango flexible de hasta veinte metros proporciona una solución versátil para aplicaciones logísticas, donde las esquinas son estrechas, las variables en movimiento son muchas y el requisito de precisión de los datos de profundidad es alto. MYNT EYE proporciona una tasa de continuidad de respuesta rápida mientras la plataforma permanece en movimiento. Para navegación o escaneo volumétrico, en áreas con poca iluminación, los componentes IR pueden ofrecer información de profundidad secundaria consistente, creando flexibilidad. También el evitar obstáculos requiere continuidad y una alta tasa de llenado. Los drones y plataformas similares con alta movilidad y peso ligero que pueden cambiar la dirección del movimiento rápidamente se benefician del mapa de profundidad de 60 FPS de MYNT EYE y un amplio campo de visión, que cubre más área y variables con información de profundidad confiable. En comparación con una solución de módulo multisensorial que cubre la misma área, el ligero MYNT EYE produce un mapa de profundidad de mayor definición con una distorsión mínima a través de un obturador global. Los proyectores IR mejoran la precisión al reducir el ruido espacial y el ruido temporal. Los datos adicionales añaden precisión y mejoran el resultado de alimentación para la medición volumétrica. Las aplicaciones de reconocimiento facial, identificación de objetos, seguimiento de movimiento, mapeo, AR y VR se pueden integrar a la perfección. La IMU de menos de un milisegundo y la velocidad de sincronización de cuadros agregan ventajas adicionales para el uso de MYNT EYE al reducir el ruido temporal (MYNTAI, 2019a, p. 5). En la ilustración 14-2 se muestra la cámara y en la tabla 1-2 un resumen de sus características.



Ilustración 14-2: MYNTEYE S1030 camera.

Fuente: (MYNTAI 2019a, p. 5)

Tabla 1-2: MYNTEYE S1030-IR datasheet.

Model	S1030-IR-120/Mono
Size	PCB dimension 149mm × 24 mm total dimension 165×31.5×29.6 mm
Frame Rate	10/15/20/25/30/35/40/45/50/55/60 FPS
Resolving Power	752x480
Depth Resolving Power	Base on CPU/GPU Up to 752×480@60FPS
IR detectable range	Up to 3m
Visual Angle	D:140°H:120°V:75°
Color Pattern	Monochrome
Pixel Size	6.0 × 6.0 μm
Shutter Speed	17 milliseconds
Baseline	120.0 mm
Camera Lens	M6 Camera Attachment
Focal Length	2.1mm
Motion Perception	6 Axis IMU
Control Function	Exposure/Shutter/Brightness/IR
IMU Frequency	100/200/250/333/500 Hz
Cameras Synchronization Precision	<1ms (up to 0.05ms)
Working Distance	0.8-5m+
Scanning Mode	Global shutter
Power	3.5W @ 5V DC from USB

Output data format	Raw data
Data transfer Interface	USB 3.0
Weight	196g
UVC MODE	YES

Fuente: (MYNTAI 2019b, p. 5)

Realizado por: Castro, Gabriel; 2023

2.9 Calibración de la cámara estéreo.

La calibración de cámara geométrica es una composición del modelado de cámara, que trata de una aproximación matemática de la física óptica usando un conjunto de parámetros, y la calibración, que trata del uso de métodos directos o iterativos para estimar los valores de estos parámetros (Cappellaro, 2019, p. 4). En la ilustración 15-2 se observa el modelo de cámara con coordenadas mundiales.

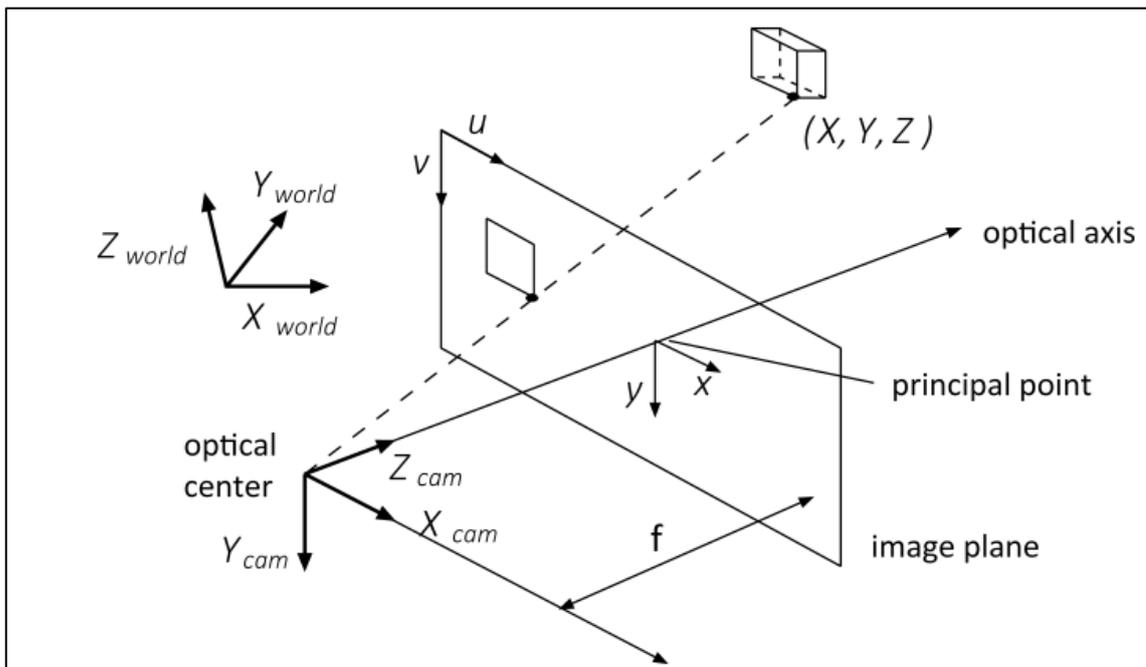


Ilustración 15-2: Modelo de proyección de cámara.

Fuente: (Cappellaro 2019, p. 4)

2.10 Modelo de cámara extrínseco e intrínseco

El modelo de cámara extrínseco proporciona una transformación de la posición y orientación de la cámara con respecto a un sistema de coordenadas mundial. Esto se realiza mediante una matriz de transformación homogénea que representa una roto-traslación definida en la ecuación 2.1.

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} R_W^C & & & t_W^C \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2.1)$$

donde t_W^C es un vector que representa el origen del marco de referencia mundial con respecto al marco de referencia de la cámara, y R_W^C es una matriz de rotación que representa los ejes del marco de referencia mundial en el de la cámara.

El modelo de cámara intrínseco describe la transformación de un punto 3D en el marco de la cámara a un punto 2D en el marco de la imagen y, por lo tanto, modela la geometría interna y las características ópticas del sensor de imagen. Suele estar compuesto por un modelo de proyección y un modelo de distorsión de lente.

2.11 Modelo de proyección estenoica

El modelo estenoico asume que todos los rayos de luz pasan a través de un solo punto, el centro óptico o centro de la cámara, y que existe una correlación lineal entre la posición del punto de la imagen y la dirección del rayo. Esto transforma un punto 3D en el marco de la cámara en un punto 2D en el marco de la imagen, utilizando proyección prospectiva. La intersección entre el eje óptico de la cámara y el plano de la imagen se denomina punto principal. En el modelo, el plano de la imagen se mueve frente al plano principal o plano de la lente, para evitar la inversión de la imagen (y se convierte en un plano de imagen virtual), como se observa en la ilustración 16-2.

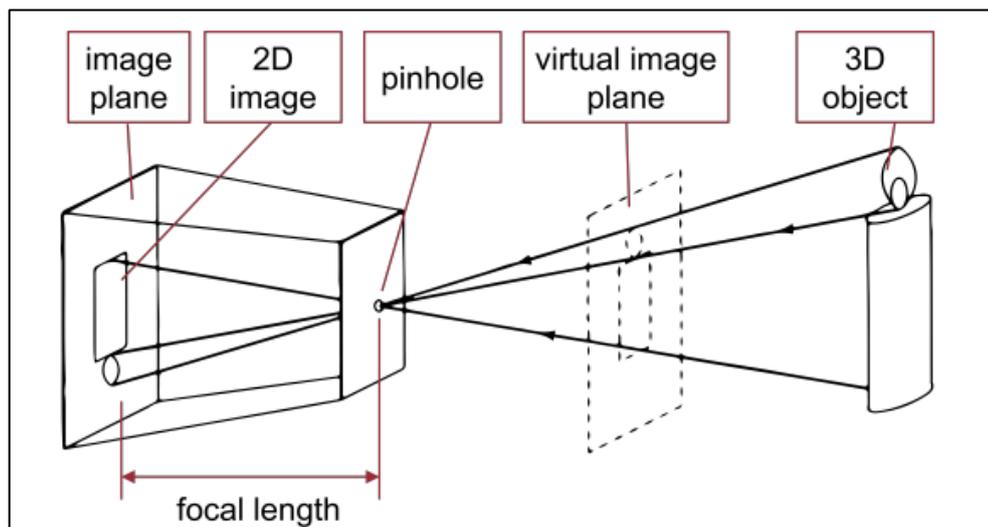


Ilustración 16-2: Modelo de proyección estenoica.

Fuente: (Cappellaro 2019)

El mapeo geométrico de 3D a 2D se puede expresar, usando coordenadas homogéneas, en la ecuación 2.2.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \quad (2.2)$$

donde λ es el factor de escala homogéneo, f es la distancia focal (en píxeles) y luego la matriz de 3×4 se denomina matriz de proyección (P). La mayoría de las cámaras actuales definen el origen del sistema de coordenadas de píxeles en la esquina superior izquierda de la imagen. Por lo tanto, las coordenadas del punto principal son $[u_0 \ v_0]^T$. Esto cambia la ecuación de proyección prospectiva en la ecuación 2.3 como:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \quad (2.3)$$

Los sensores reales tienen más imperfecciones que deben tenerse en cuenta. De hecho, es posible que los píxeles no estén cuadrados, lo que da lugar a dos distancias focales diferentes en cada dirección, y pueden estar potencialmente sesgados, lo que lleva a un factor de sesgos. El píxel puede aparecer torcido en caso de que un capturador de fotogramas adquiera la imagen, debido a una sincronización imprecisa del proceso de muestreo de píxeles. La definición más general del mapeo de proyección es entonces en la ecuación 2.4 como:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = [K \ | \ 0] \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \quad (2.4)$$

La matriz K, ecuación 2.5 se llama matriz de cámara intrínseca y

$$K = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

En la práctica, usando una cámara digital reciente, se puede suponer que $f_x = f_y$ y $s = 0$.

2.12 Modelos de distorsión de lente

En las cámaras reales existe una distorsión no lineal producida por los objetivos, que se debe tener en cuenta. Se considera la proyección del agujero de alfiler en el foco unitario $x = \frac{X_{cam}}{Z_{cam}}$, $y = \frac{Y_{cam}}{Z_{cam}}$ y $r = \sqrt{x^2 + y^2}$. Por supuesto, considerando las coordenadas de píxeles distorsionadas, $[X_{dist}, Y_{dist}]^T$, se observa esta relación en la ecuación 2.6.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = [K] \begin{bmatrix} X_{dist} \\ Y_{dist} \\ 1 \end{bmatrix} \quad (2.6)$$

2.12.1 Modelo de distorsión radial-tangencial

Para las cámaras con bajo campo de visión, un buen modelo es el considerado en OpenCV (Fernández, 2018), el modelo de distorsión radial-tangencial. OpenCV es una biblioteca de visión de código abierto muy utilizada para el procesamiento de imágenes en tiempo real. El modelo mencionado necesita 5 parámetros, (k_1 , k_2 , p_1 , p_2 , k_3), ecuación 2.7.1 y 2.7.2, que son coeficientes de distorsión. "la distorsión radial se debe principalmente a una curvatura radial defectuosa de los elementos de la lente". Depende de la distancia radial desde el punto principal del plano de la imagen, por lo que es más notable en la periferia de la imagen. Una distorsión de barril consiste en un desplazamiento radial negativo de los puntos de la imagen, y hace que los puntos exteriores se acerquen cada vez más entre sí y la escala disminuya, ilustración 17-2. Una distorsión de cojín es, en cambio, un desplazamiento radial positivo que hace que los puntos exteriores se extiendan y la escala aumente. La función utilizada en el modelo de OpenCV es:

$$x_{distortion}(x) = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.7.1)$$

$$Y_{distortion}(Y) = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.7.2)$$

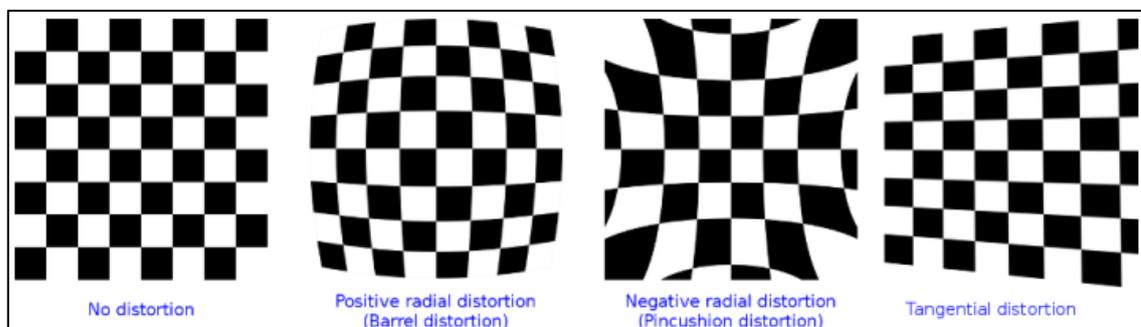


Ilustración 17-2: Ejemplos de las distintas distorsiones ópticas. De izquierda a derecha, imagen sin distorsión óptica, imagen con distorsión radial positiva, imagen con distorsión radial negativa e imagen con distorsión tangencial.

Fuente: (Jáuregui Elso 2018, p. 8)

La distorsión tangencial depende principalmente del no paralelismo entre el plano de la imagen y el plano de la lente. Es un efecto asimétrico y es mucho más pequeño en comparación con las distorsiones radiales de las cámaras modernas. Este efecto se observa en la Ilustración 18-2. Y en las ecuaciones 2.8.1 y 2.8.2 se establece el modelo de distorsión tangencial que dependen de dos valores p_1 y p_2 , valores de corrección.

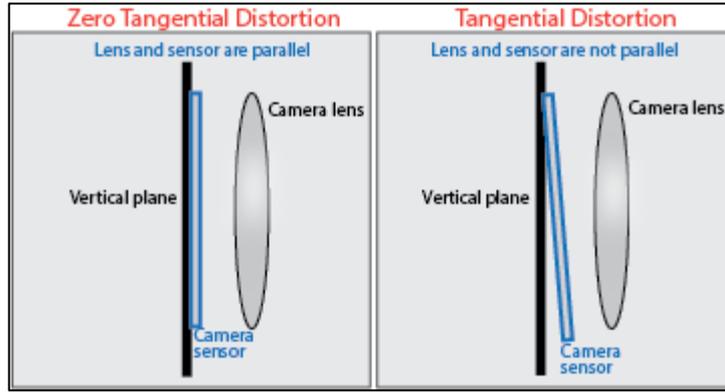


Ilustración 18-2: Imagen con distorsión tangencial.

Fuente: (Matlab 2023, p. 1)

$$x_{distortion}(x,y) = x + 2p_1xy + p_2(r^2 + 2x^2) \quad (2.8.1)$$

$$y_{distortion}(x,y) = y + 2p_2xy + p_1(r^2 + yx^2) \quad (2.8.2)$$

Por supuesto, la distorsión total es la suma de las dos, la radial y tangencial.

2.12.2 Modelo de distorsión de Kannala-Bradt

Tanto OpenCV, Matlab como Kalibr tienen el mismo modelo de cámara, denominada ojo de pez (estenopeica) en las dos primeras tanto en OpenCV como en Matlab y equidistante en Kalibr. Esta es una combinación de un modelo de cámara estenopeica con un modelo de distorsión Kannala-Bradt. Incluso si este no es uno de los modelos completos de cámara para lentes de ojo de pez, es una simplificación bastante buena. El nombre equidistante proviene del modelo de cámara equidistante en el que se basa. El modelo de cámara equidistante tiene un modelo de proyección parametrizado como $r = f\theta$, donde θ es el ángulo entre el eje principal y el rayo entrante, r es la distancia entre el punto de la imagen y el punto principal y f es la distancia focal, véase la ilustración 19-2. La mayoría de las lentes de ojo de pez están hechas para seguir este modelo de proyección, por lo que es la más utilizada en las cajas de herramientas de calibración. Usando el modelo de proyección como modelo de distorsión, consideramos la distorsión dada por una esfera que se proyecta sobre un plano. Considerando $\theta = \arctan\left(\frac{r}{z_{cam}}\right)$, el modelo de distorsión de Kannala-Bradt se define en la ecuación 2.9 como:

$$\theta_{distortion} = \theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9 \quad (2.9)$$

y la coordenada del píxel de distorsión se convierten en 2.10.1 y 2.10.2:

$$x_{distortion}(x, y, Z_{cam}) = \theta_{dist} \left(\frac{x}{r} \right) \quad (2.10.1)$$

$$y_{distortion}(x, y, Z_{cam}) = \theta_{dist} \left(\frac{y}{r} \right) \quad (2.10.2)$$

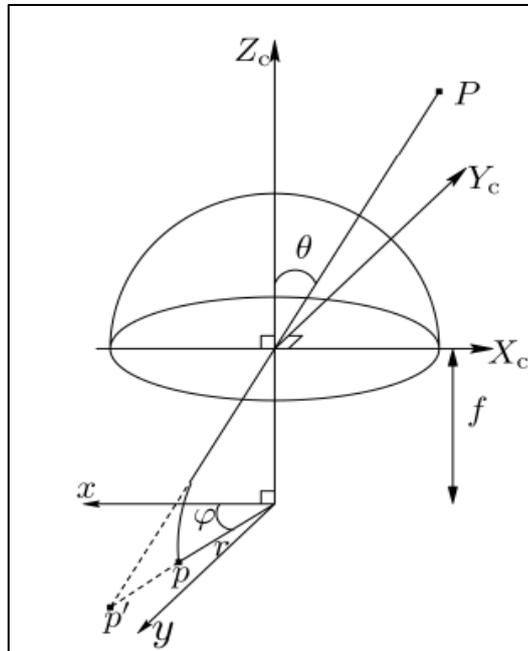


Ilustración 19-2: Imagen con distorsión tangencial.

Fuente: (Cappellaro 2019, p. 8)

En el modelo de cámara ojo de pez, la imagen del punto P es p, mientras que sería p' usando un modelo de cámara estenopeica.

2.13 Mejores prácticas de calibración

La calibración precisa es de importancia clave para el rendimiento en la mayoría de las tareas de visión artificial y de máquinas. A continuación, se enumeran algunas recomendaciones que ayudarían a realizar una buena calibración.

- Se selecciona el objetivo de calibración del tamaño correcto. Lo suficientemente grande para enlazar los patrones de calibración correctamente. Preferiblemente, cubrir aproximadamente la mitad del área total de la imagen cuando se ve desde el frente y en paralelo.
- Se calibra a la distancia de trabajo aproximado de la aplicación final o proyecto a realizar. La cámara debe estar enfocada a esta distancia y el punto focal de la lente debe permanecer sin cambios después de la corrección.

- Se recoge imágenes de diferentes áreas e inclinaciones. Se mueve el objetivo para cubrir completamente el área de la imagen. Incluya tanto imágenes fronto-parallelas como imágenes tomadas con el patrón de inclinación hasta +/- 45 grados en ambas direcciones, horizontal y vertical. Por lo general, inclinar más no es una buena idea, ya que la precisión de la localización de características se ve afectada y puede sesgarse.
- Asegurar una buena iluminación. Esto a menudo se pasa por alto, pero es muy importante. El objetivo de calibración debe estar preferiblemente iluminado de forma difusa mediante iluminación fotográfica controlada. Las fuentes lumínicas puntuales fuertes dan lugar a una iluminación desigual, lo que posiblemente haga que la detección falle y no utilice muy bien el rango dinámico de la cámara.
- Tener suficientes observaciones. Por lo general, la calibración debe realizarse en al menos 6 observaciones (imágenes) de un objetivo de calibración. Si se utiliza una cámara de orden superior o un modelo de cámara con distorsión, son beneficiosas más imágenes.

Seguir estas prácticas debería garantizar la calibración más exacta y precisa posible (Jakob, 2018).

2.14 Geometría Epipolar.

La geometría epipolar es la geometría que se forma de un arreglo de dos cámaras monoculares, un punto en el espacio de coordenadas mundiales puede ser visto por las cámaras en dos posiciones distintas, por lo tanto, hay algunas relaciones geométricas entre los puntos 3D y sus proyecciones en las imágenes 2D. En la ilustración 20-2, se considera al plano epipolar como aquel que está formado por los dos centros ópticos C_I y C_D con cualquier punto M en el espacio de coordenadas mundiales. El plano epipolar $C_I M C_D$ corta a los dos planos de imagen I y D en dos líneas rectas ep_I y ep_D , denominadas líneas epipolares. La proyección del centro óptico de cada cámara sobre su par óptico define el epipolo e . Los epipolos de cada cámara serán los puntos por los que van a pasar todas las líneas epipolares. Esto implica que en cualquier plano epipolar como base, todos los puntos de coordenadas mundiales de la escena que pertenecen a este plano tendrán su imagen en cada una de las líneas epipolares de las dos imágenes. Por lo tanto, cada uno de los píxeles de una línea epipolar izquierda o derecha, debe tener su par correspondiente dentro de la línea epipolar en la otra imagen (López Valles et al., 2005, p. 37). Si se considera que los planos de imagen de cada cámara son coincidentes entre sí, y paralelos a la línea base (C_I , C_D) que une los dos centros de imagen, se observa que los epipolos de las dos cámaras se sitúan en el infinito, por lo tanto, las líneas epipolares serán todas paralelas entre sí, y a su vez paralelas con la línea base.

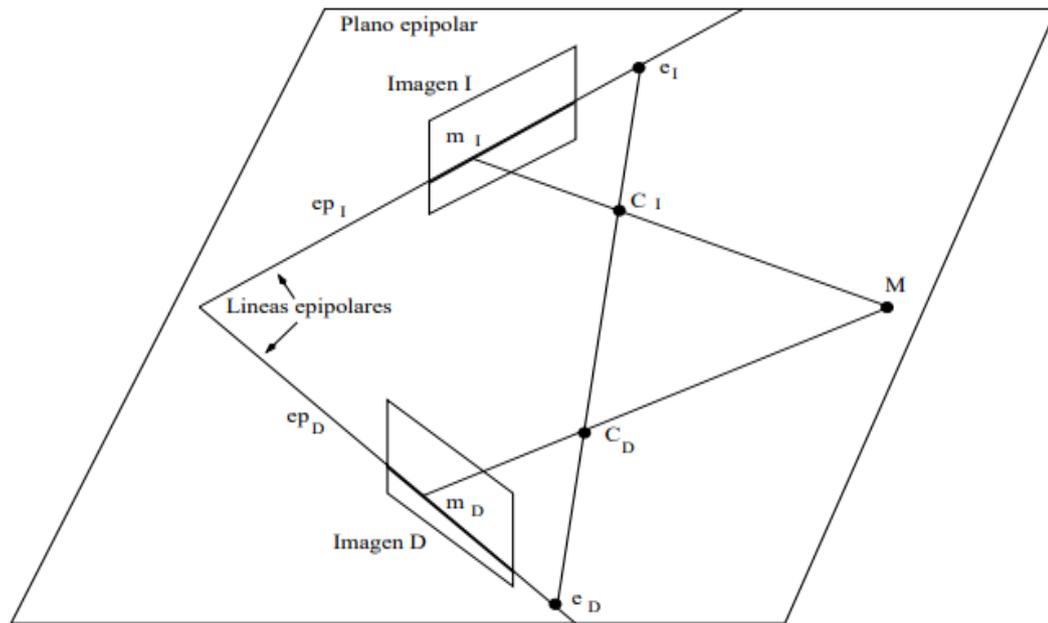


Ilustración 20-2: Plano epipolar.

Fuente: (López Valles, Fernández Caballero y Fernández 2005)

Con esta geometría, se puede conseguir que las líneas epipolares coincidan con las filas de ambas imágenes digitales obtenidas para la búsqueda un mapa de profundidad, disminuyendo la búsqueda a una sola línea en lugar de un espacio 2D.

2.15 Triangulación estéreo.

La triangulación estéreo consiste en la búsqueda de un punto en coordenadas mundiales mediante sus proyecciones en dos imágenes. En la ilustración 21-2 se considera que el sistema de referencia de la escena tiene su origen en el centro óptico de la cámara izquierda, el eje X es paralelo a la línea base y el eje Z es perpendicular al eje de las cámaras. Se considera un plano exclusivamente epipolar. Para encontrar las coordenadas de un punto $M(x,y,z)$ partiendo de las proyecciones del punto sobre los planes de imagen (u_I, v_I) y (u_D, v_D) , para esto se necesita la distancia focal y la longitud de la línea base. En las ecuaciones 2.11.1 y 2.11.2 se presentan las coordenadas x e y respectivamente de la imagen izquierda.

$$\frac{u_I}{f} = \frac{x}{z} ; x = \frac{u_I}{f} \cdot z \quad (2.11.1)$$

$$\frac{v_I}{f} = \frac{y}{z} ; y = \frac{v_I}{f} \cdot z \quad (2.11.2)$$

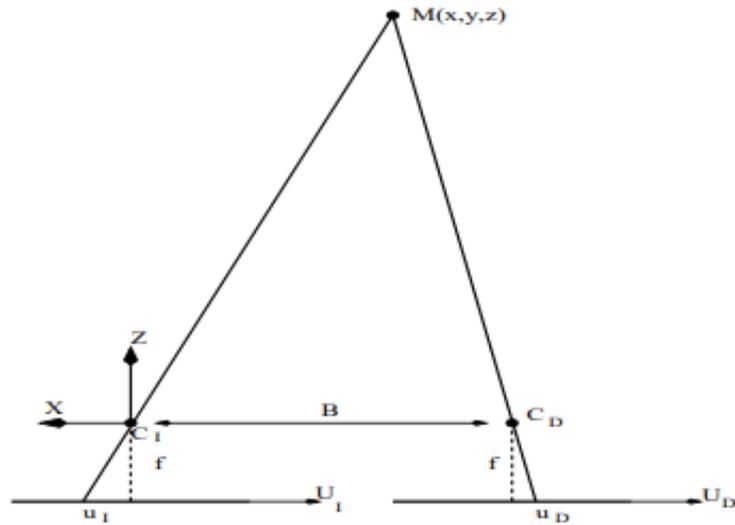


Ilustración 21-2: Geometría en el plano epipolar para cámaras paralelas.

Fuente: (López Valles, Fernández Caballero y Fernández 2005)

De igual manera para la imagen derecha, las ecuaciones 2.12.1 y 2.12.2 presentan las coordenadas x e y respectivamente.

$$x = \frac{u_D}{f} * z - B \quad (2.12.1)$$

$$y = \frac{v_D}{f} * z \quad (2.12.2)$$

Simplificando estas ecuaciones, se llegan a las siguientes expresiones.

$$x = \frac{u_I * B}{d} \quad (2.13.1)$$

$$y = \frac{v_I * B}{d} \quad (2.13.2)$$

$$z = \frac{f * B}{d} \quad (2.13.3)$$

La d de las ecuaciones 2.13.1, 2.13.2 y 2.13.3 es denominada como disparidad, que se conoce como la diferencia entre las coordenadas u_I y u_D respecto del centro de las imágenes. Al conjunto de todas las disparidades entre dos imágenes de un par estéreo se denomina mapa de disparidad. Las disparidades solo se pueden calcular de los puntos característicos visibles en ambas imágenes, las características que solo se ven en una sola imagen se denominan oclusiones (López Valles et al, 2005,p. 38).

2.16 Correspondencia estéreo.

La correspondencia estero es considera la etapa más crucial de la reconstrucción estereoscópica, debido que dadas un par o más imágenes se busca establecer correspondencias entre estas, como características o puntos idénticos en las imágenes, que son la proyección de un único punto en el espacio coordenadas mundiales (Tardón García, 1999, p. 47). Todas las técnicas de correspondencia buscan de crear un mapa de disparidad, que trata la colocación de un valor de disparidad a cada píxel de la imagen (Jiménez Medina, 2008, p. 22). La coincidencia estéreo se puede clasificar en dos tipos: globales y locales. La búsqueda de correspondencias por lo tanto es una tarea difícil, lo que ha retrasado los avances en esta área. El problema de correspondencia se da cuando un punto de una imagen no tiene o tiene más de dos correspondientes en la otra imagen, como ejemplo, si se tiene un par de imágenes de una escena uniforme de color y textura, se dificulta saber que punto de la primera imagen corresponde con la segunda imagen. Por ende, se han planteado algunas restricciones que ayuden a solucionar el problema de la correspondencia (Wong Quijano, 2013, p. 11). Se presentan las siguientes restricciones:

- **Compatibilidad:** Esta restricción plantea que la correspondencia de puntos oscuros solo se pueda emparejar con puntos oscuros, los de tonos de grises con tonos de grises y los de color con los puntos que tengan el mismo color.
- **Unicidad:** Esta restricción plantea que un punto característico en la primera imagen solo pueda ser emparejada con un único punto en la segunda imagen.
- **Continuidad:** Esta restricción plantea que la disparidad del par estéreo varia de manera suave en casi toda la imagen.

2.16.1 Métodos locales

Los métodos de correspondencia locales tratan de encontrar un punto característico de una imagen en la otra imagen, aplicando algunas restricciones, y utilizando la información que proveen un pequeño grupo de pixeles que rodea al píxel de interés. Pueden ser muy buenos para la búsqueda del mapa de disparidad, pero, sensitivos a oclusiones, texturas uniformes y regiones sin textura (Jiménez Medina, 2008, p. 23).

Los métodos locales pueden clasificarse en dos categorías: correspondencia por bloques y correspondencia por características. Los métodos de correspondencia por bloques estiman la disparidad realizando una comparación de uno a uno, de una región pequeña cercana al punto de

interés con una serie de regiones de interés obtenidas en la segunda imagen. La geometría epipolar es de mucha ayuda permitiendo reducir el espacio de búsqueda a una sola fila. El problema principal de este método es el de cómo evaluar la similitud entre dos bloques. Existen algunas métricas para encontrar la similitud entre dos bloques, la métrica de similitud más utilizada es la del error mínimo cuadrado, que se define en la ecuación 2.14.

$$MSE_{x,y,d} = \frac{1}{M*N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (E_L(x+i, y+j) - E_R(x+d, y+j))^2 \quad (2.14)$$

M y N son las dimensiones horizontal y vertical del bloque, d es desplazamiento no negativo y múltiplo de M. La redundancia de las operaciones en la realización de los mismos cálculos más de una vez es una de las desventajas de este método, por lo que provoca que se consuma mucho tiempo de cómputo y la minimización del número de cálculos resulta en un mapa de disparidad menos exacto (Jiménez Medina, 2008, p. 24).

Los métodos de correspondencia por características a diferencia de los métodos por bloques no permiten obtener mapas de disparidad densos, lo que ha ocasionado que el interés en el desarrollo de estas técnicas vaya decayendo en los últimos años. En lugar de realizar la búsqueda de correspondencia de píxel a píxel, este método se basa en la búsqueda de la forma del objeto, en concreto bordes, segmentos, esquinas y regiones.

2.16.2 Métodos globales

Los métodos globales minimizan una función de energía explícita sobre todos los píxeles de la imagen. Una función de energía típica tiene la forma de la ecuación 2.15.

$$E_p = E_{data}(p) + \delta E_{smooth}(p) \quad (2.15)$$

Donde:

- $E(p)$ es el valor de energía total de un píxel p de la imagen.
- El término de datos E_{data} explica la similitud del color, miden la semejanza entre los puntos correspondidos en ambas imágenes.
- E_{smooth} , el término de suavidad captura la suavidad local del objeto, tratando de asignar una disparidad con un valor igual a los puntos que se encuentran en un mismo vecindario.
- El parámetro $\delta \in R$ es la influencia relativa de los términos E_{data} y E_{smooth} .

Se puede obtener mayor información en el trabajo de (Kapeller, 2018, p. 19) de donde fueron obtenidos estos datos.

2.17 Disparidad estéreo mediante coincidencia de bloques semi-globales (SGBM)

El método de coincidencia de bloques semi-globales en ingles *semi-global block matching* SGBM es una variante del método *semi-global matching* desarrollado por (Hirschmuller 2008, p. 1). Este es un método global que permita la creación de un mapa de disparidad denso, SGBM a diferencia de SGM trabaja con bloques y no mediante una comparación de píxel a píxel. La manera de como SGBM realiza la búsqueda de la disparidad, es mediante la realización de un emparejamiento de bloque a bloque entre las imágenes estéreo con orientación externa e interna conocidas (Wong Quijano, 2013, p. 23), lo que implica que las imágenes están debidamente rectificadas, y la búsqueda de correspondencias se realizan solo en las filas de pixeles de las imágenes, minimizando la función de energía general de los métodos globales ecuación 2.15, la ecuación de energía de SGBM se observa en la ecuación 2.16.

$$E(D) = \sum_p (C(p, D_p) + \sum_{q \in N_p} P_1 I[|D_p - D_q| = 1] + \sum_{q \in N_p} [|D_p - D_q| > 1]) \quad (2.16)$$

En donde:

- D es el mapa de disparidad resultante. $E(D)$ es la energía para el mapa de disparidad.
- p, q representan índices de pixeles en la imagen. N_p es la vecindad de pixeles del punto p . $C(p, D_p)$ es el costo del emparejamiento con la disparidad en D_p . P_1 es la penalización para un cambio en el valor de la disparidad de 1 entre pixeles vecinos.
- P_2 es la penalización para un cambio en el valor de la disparidad mayores a 1.
- $I[.]$ es la función que regresa si los argumentos son verdaderos 1 y 0 si no lo son.

El valor de $C(p, D_p)$ es calculado como se observa en la ecuación 2.17.

$$C(p, D_p) = \min(d(p, p-d), d(p-d, p, I_R, I_L)) \quad (2.17)$$

Donde I_R, I_L son las imágenes rectificadas, en la ecuación 2.18 se observa la disparidad.

$$d(p-d, p, I_R, I_L) = \min_{p-d-0.5 \leq p-d+0.5} |I_L(p) - I_L(q)| \quad (2.18)$$

La función de minimización produce un mapa de disparidad cuya suavidad está relacionada directamente de P_1 y P_2 . En la ecuación 2.19 se observa la ecuación de costo agregado por píxel p y disparidad d .

$$S(p, d) = \sum_r L_r(p, d) \quad (2.19)$$

Se tiene entonces que r es la dirección usada para converger con el píxel p . $L_r(p, d)$ es el costo mínimo de la ruta tomada en dirección r del píxel para la disparidad d . En la ecuación 2.20 se observa el valor del costo $L_r(p, d)$.

$$L_r(p, d) = C(p, d) + \min(L_r(p - r, d), L_r(p - r, d - 1) + P_1, L_r(p, d + 1) + P_1, \min_i L_r(p - r, i) + P_2) - \min_k L_r(p - r, k) \quad (2.20)$$

La función de costo resulta de la suma de sub-costos para encontrar la disparidad, sumando el costo actual $C(p, d)$ al píxel anterior en dirección r . El costo mínimo en el píxel anterior con disparidad d , el costo en el píxel anterior con disparidad $d - 1$ y $d + 1$ con penalización P_1 , y el costo en el píxel anterior con disparidades menores a $d - 1$ y mayores a $d + 1$ con penalización P_2 . SGBM selecciona la disparidad cuyo costo sea el menor para ese bloque, por lo tanto, se suman todos los costos de los caminos que convergen en el píxel de búsqueda, el costo es calculado para el rango definido entre la máxima disparidad y el número de disparidades permitidas. Todos estos datos se encuentran disponibles en el trabajo de (Wong Quijano, 2013, p. 24).

2.18 SDK de Mynt Eye

SDK de MYNT está basado en CMake lo que le da la ventaja de ser múltiple plataforma. (MYNTAI 2019a, p. 9) proporciona dos instalaciones: descargar e instalar, y compilar e instalar desde el código fuente. Estas son las plataformas en el que se pueden utilizar el SDK:

- Windows 10
- Ubuntu 18.04.1 / 16.04.6 / 14.04.5
- Jetson TX1/TX2 / Xavier
- firefly RK3399

La instalación del SDK es sencilla, este puede ser instalado desde fuente para Ubuntu y mediante un archivo ejecutable .exe para Windows. Las versiones estables para el correcto funcionamiento del SDK en Ubuntu son 14.04, 16.04 y 18.04 LTS, en Windows funciona con Windows 10 y 11. La documentación para la instalación en cualquier plataforma, incluida ROS está disponible en la página web del fabricante. Para desarrollar aplicaciones y utilizar algunos de los ejemplos disponibles en el kit de desarrollo de software se debe de instalar primeramente CMake y posterior Visual Studio (MYNTAI 2019, p. 9) . El SDK hace uso de dependencias de OpenCV lo que permite interactuar con código de estas librerías. Para la obtención de imágenes mediante Visual Studio,

el SDK ofrece ejemplos de código ya desarrollados en C++ para la configuración de los parámetros de las cámaras como la auto exposición, el control del *framerate*, el control del infrarrojo, control del rango del sensor inercial, también se puede calibrar las cámaras mediante la escritura de un archivo *.yaml* en el que se pueden editar los parámetros de calibración y grabarlos directamente al hardware, hay ejemplos más aplicativos que permiten la obtención de la disparidad, la profundidad, la grabación de videos y la obtención de imágenes, este último es el que se utilizara para la obtención de un único par de imágenes. En la ilustración 22-2 se observan los ejemplos del kit de desarrollo de software de las cámaras *MYNT*.

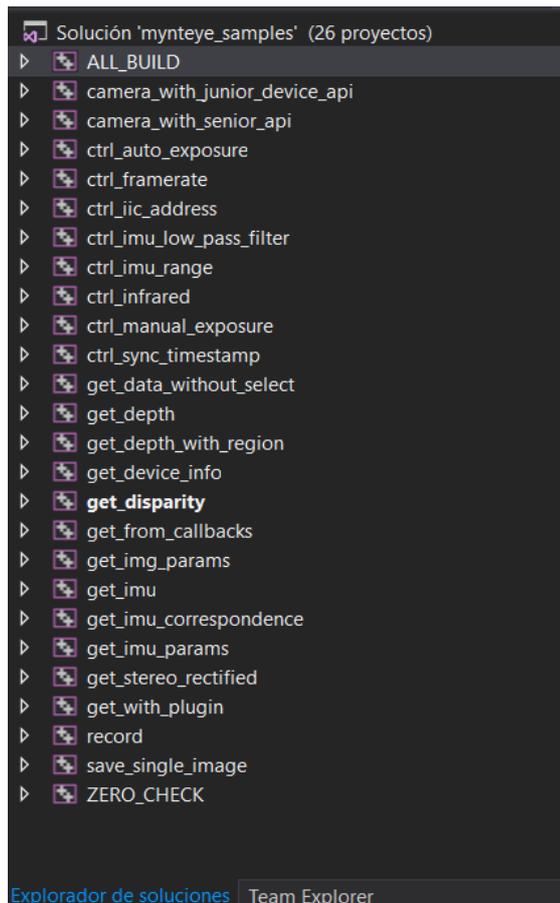


Ilustración 22-2: Ejemplos de configuración de la cámara Mynt s1030.

Fuente: MYNTAI;2019

Realizado por: Castro, Gabriel;2023

2.19 OpenCV

OpenCV es una biblioteca de código abierto que contiene implementaciones que abarcan más de 2500 algoritmos. Además, está especializada en el sistema de visión artificial y machine learning. Su utilización principal abarca la detección de objetos y rostros, especialmente en aspectos como

la seguridad, el marketing o incluso la fotografía (Rodríguez, 2021). Entre las aplicaciones más utilizadas de OpenCV están:

- Identificar objetos o caras (reconocimiento facial).
- Encontrar imágenes similares.
- Eliminar los ojos rojos de las fotografías.
- Reconocer escenarios.
- Seguir los movimientos de los ojos.
- Clasificar acciones humanas que estén en videos.
- Extraer modelos 3D.
- Útil en campos como la robótica y la realidad aumentada.

2.20 Microsoft Visual Studio.

Microsoft Visual Estudio es un entorno de desarrollo integrado (IDE), útil para desarrollo de aplicaciones informáticas. El IDE de visual estudio se caracteriza por ser un panel de inicio creativo para editar, depurar, compilar código. Es una herramienta bastante completa que incluye compiladores, herramientas de finalización de código, diseñadores gráficos y muchas herramientas más que permiten la comodidad al crear aplicaciones (TerryGLee, 2023), en la ilustración 23-2 se observa la ventana de programación del Visual Studio.

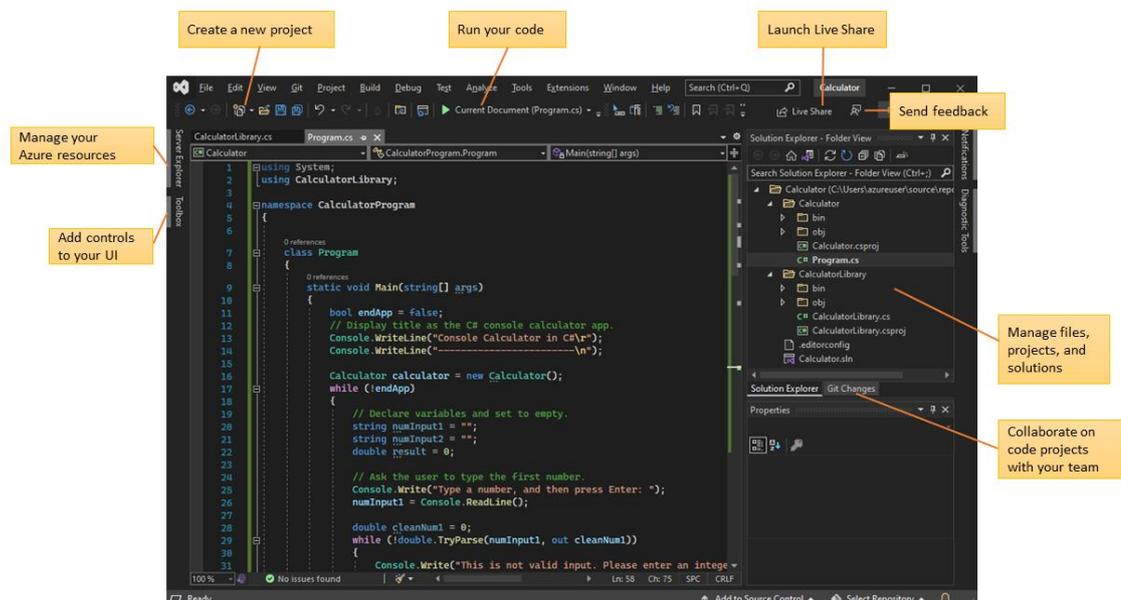


Ilustración 23-2: Software de desarrollo Visual Studio.

Fuente: (TerryGLee 2023, p. 1).

2.21 Matlab

MATLAB (abreviatura de Matrix laboratory) que en español significa laboratorio de matrices es un software de ingeniería muy completo, ofrece un entorno de desarrollo integrado, con su propio lenguaje de programación. Es de múltiple plataforma lo que le posibilita ser instalado en Windows, Linux y Apple, se basa en la manipulación y tratamiento de matrices, creación de funciones, creación de algoritmos y de hasta interfaces de usuario didácticas. Ofrece la ventaja de mejorar su capacidad con la ayuda de toolboxes que son cajas de herramientas que traen consigo funciones para el procesamiento de aplicaciones dedicadas, en el caso de procesamiento de imágenes cuenta con el toolbox “ Image Processing Toolbox”. Entre las aplicaciones que ofrece Matlab para el procesamiento de imágenes destacan:

- Visión artificial
- Calibración de cámaras monoculares y estéreo.
- Etiquetado de imágenes y videos.
- Registro y Segmentación de imágenes.
- Visualización de volumen.
- Navegador DICOM.
- Procesamiento de imágenes 3D
- Detección, seguimiento y reconocimiento de objetos.
- Procesamiento de nubes de puntos.
- visión estereoscópica.
- Integración con código abierto.

2.22 MeshLab

Meshlab es un software de procesamiento de mallas 3D de código abierto y gratuito. Con un set de herramientas que van desde la edición hasta la conversión de mallas 3D. Para (Cignoni et al. 2008, p. 2) meshlab fue creado teniendo en mente objetivos principales como facilidad de uso, es decir que no necesite de habilidades desarrolladas en el usuario para el procesamiento de mallas. También con el fin de que usuarios experimentados puedan utilizar, agregar y modificar funcionalidades para mejorar el rendimiento. Meshlab en su comienzo estuvo orientado solamente al procesamiento de mallas. Eficiencia, con la capacidad de manejar millones de parámetros primitivos en el procesamiento de mallas. Y por último sustentabilidad, con la capacidad de crecer y mantenerse por sí solo por unos años.

2.23 Colmap

Colmap es un software de reconstrucción que está basado en las técnicas de estructura a partir del movimiento (SFM) y estéreo multi visión (MVS), cuenta con una interfaz gráfica y línea de comandos. Ofrece un amplio set de características para la reconstrucción de escenas u objetos desde colecciones de imágenes desordenadas u ordenadas. Esta bajo la licencia de BSD. Es considerada una biblioteca de algoritmo licenciados que deben citarse al ser utilizados en investigaciones debido a que están bajo una licencia. En (Schonberger y Frahm, 2016, p. 4104) y (Schönberger et al., 2016, p. 2) se analizan las técnicas de SFM y MVS respectivamente que utiliza Colmap.

CAPÍTULO III

3. MARCO METODOLÓGICO

En este capítulo se detalló la ejecución del proyecto, la metodología utilizada, el diseño del sistema, definición de las etapas del proyecto, desarrollo del algoritmo e implementación del sistema de reconstrucción.

La metodología utilizada y que mejor se apega a nuestro proyecto en el desarrollo del sistema de reconstrucción virtual de objetos debido a que la parte más extensa se centra en la elaboración de un algoritmo, es la metodología en cascada en inglés *waterfall*. Esta es una metodología en la que las etapas se organizan de manera estructurada y ordenada de arriba hacia abajo, cada etapa se desarrolla de manera diferenciada por lo que se debe revisar si en cada fase se ha cumplido los requisitos y se pueda pasar a la siguiente fase, entre las ventajas de usar esta metodología es que tiene una estructura sencilla debido a que cada una de las fases esta diferenciada, también permite una buena documentación del proceso a través de etapas bien definidas, entre la desventaja más común radica en el hecho de que no se puedan visualizar los resultados hasta que el proyecto este lo suficientemente avanzado, lo que implica que no se puedan detectar fallos hasta haber culminado el proyecto.(Santander, 2020)

En la práctica, esta metodología se divide en 5 etapas o fases, en la ilustración 1-3 se observa esta metodología, organizada por secciones comienza por la etapa de análisis que constituye la planificación y especificación de los requisitos que debe cumplir el sistema, la etapa de diseño se caracteriza por realizar el diseño del sistema y las especificaciones, en la etapa de implementación se realiza el algoritmo y pruebas parciales de este, en verificación se integran todos los sistemas y se realizan pruebas de funcionamiento, terminando en la etapa de mantenimiento, esta recoge todos los datos y realiza mejoras en cuanto a fallas.

3.1 Análisis: análisis y especificación de los requisitos del sistema.

En la primera fase del método se definen cuáles son los requisitos para cumplir con el afán de tener un sistema funcional. Primeramente, se abordó la problemática del proyecto definida en el capítulo 1, cada una de las preguntas propuestas establece la necesidad de plantear requisitos que satisfagan el cumplimiento de cada uno de los objetivos. Cada una de las preguntas establece el desarrollo del proyecto durante todas las etapas de la metodología, responder a cada una

corresponde al desarrollo del proyecto. Por lo tanto, esta fase de análisis se centra en definir que funciones y características debe tener el sistema en base a la problemática para ser operativo, además de que requisitos debe cumplir el proyecto de reconstrucción.

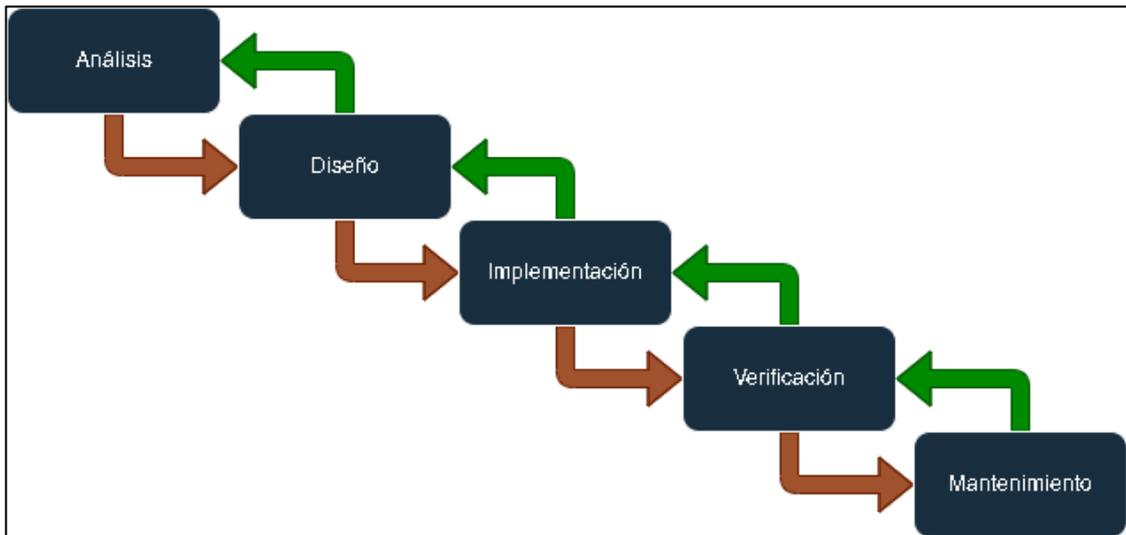


Ilustración 1-3: Metodología *waterfall*.

Fuente: Santander; 2020

Realizado por: Castro, Gabriel; 2023

3.1.1 Características y funciones del sistema de reconstrucción virtual de objetos.

- Reconstruye objetos dentro de su Angulo de visión, consiste en la capacidad del sistema de reconstruir objetos dentro del Angulo de visión de diseño.
- Con la opción de tomar múltiples vistas y unir las en una sola nube de puntos, consiste en unión de múltiples vistas en donde cada vista aporta positivamente en la mejora del objeto reconstruido mediante la toma de imágenes que las otras vistas no pueden ver.
- Utilizable para ambientes de poca iluminación, sistema con la capacidad de reconstruir objetos en ambientes de baja iluminación.
- De rápida implementación y mantenimiento, sistema sencillo de administrar, utilizar y ajustar de acuerdo con los requerimientos del objeto a reconstruir.
- Basado en visión estereoscópica, sistema únicamente basado en la técnica de reconstrucción de mapas de profundidad mediante la utilización de pares de cámaras debidamente calibradas
- Resolución de escaneado de hasta 752X480 píxeles, resolución de cada una de las cámaras que forman el sistema estéreo.
- No automático, el sistema no permite la adquisición de imágenes directamente desde el mismo algoritmo, se hace uso de varias herramientas.

3.1.2 *Requisitos del sistema de reconstrucción virtual de objetos.*

- El sistema debe estar basado en visión estereoscópica, debido al uso de cámaras estéreo el método de reconstrucción se realiza mediante la correspondencia de puntos en dos imágenes del mismo objeto.
- Cada cámara estéreo debe estar correctamente calibrada, con el objetivo de eliminar distorsiones e irregularidades que presenten las cámaras por defectos de fabricación.
- Diseño simple, de fácil implementación y mantenimiento, el diseño del sistema debe constar de varias vistas estéreo colocadas de manera en que todas obtengan una porción de visión del objeto, el sistema por lo tanto permite el posicionamiento de cada cámara respecto a la otra y al objeto, además de permitir realizar un fácil mantenimiento y limpieza de las cámaras.
- De múltiples vistas, el sistema debe tomar varias vistas obtenidas por las cámaras y procesar estas imágenes con el fin de obtener un solo plano tridimensional.
- Móvil y trasladable, con la capacidad de movilizar y transportar el sistema de manera que no vulnere la integridad de cada componente.
- Sistema compatible en cualquier sistema operativo con Matlab y Opencv 3.4.1, debido a la necesidad de herramientas de adquisición y procesamiento de imágenes los dispositivos controladores del sistema deben ser compatibles con Matlab es sus últimas versiones y con el SDK del fabricante el cual maneja OpenCV 3.4.1.

3.1.3 *Análisis de la definición de los requisitos.*

El sistema de reconstrucción virtual de objetos mediante procesamiento digital de imágenes, está basado únicamente en la técnica de visión estereoscópica que simula la visión humana, para esto el par estéreo debe estar completamente calibrado, con el objetivo de encontrar parámetros de corrección de errores en cuanto a forma de píxeles, distorsiones radiales y tangenciales vistas en el capítulo 2, el sistema debe ser sencillo lo que implica que cualquier persona pueda utilizarlo sin la necesidad de ser un experto programador. El algoritmo debe ser capaz de obtener nubes de puntos a partir de un par estéreo y posterior la unión de las diferentes nubes de puntos obtenidas de las diferentes vistas para formar una sola nube de puntos del objeto digitalizado. La estructura del sistema debe ser móvil, es decir podrá ser trasladado a diferentes lugares en el que se lo necesite. Y por último el sistema debe ser compatible en múltiples plataformas conocidas, evitando la necesidad de tener un solo computador que lo administre.

3.2 Diseño: diseño y especificación del sistema.

En esta etapa de la metodología se aborda el método que se utilizó, la definición de las fases de operación del sistema de reconstrucción y el diseño del sistema de reconstrucción virtual de objetos, partiendo de la problemática descrita en el capítulo 1 de ¿cómo diseñar e implementar un sistema de reconstrucción virtual de objetos?

En el capítulo 2 se han puesto de manifiesto algunos de las formas y métodos empleados para convertir objetos físicos a virtuales. En este proyecto se utilizó una técnica pasiva, es decir, la técnica de visión estereoscópica. Este método consiste en la triangulación de los puntos extraídos del estéreo par, que tiene como ventaja que es una técnica enteramente tridimensional partiendo de la captura de un par de fotos. (Cárdenas Quiroga, Morales Martín y Ussa Caycedo, 2015)

La técnica de reconstrucción estereoscópica realiza la virtualización de los objetos mediante una matriz de proyección que relaciona el plano tridimensional de coordenadas universales con el plano de imagen de la cámara que se encuentra en un plano bidimensional. La matriz de proyección es encontrada a través de la calibración de las cámaras estéreo, esta matriz esta formada por parámetros intrínsecos, extrínsecos, matriz de traslación y rotación.

Los intrínsecos son los parámetros de corrección de cada cámara como sesgo de los pixeles, posición de referencia de cada imagen, distancia focal, en cambio los parámetros extrínsecos son parámetros de distorsión como distorsión radial y tangencial que las cámaras puedan presentar por defectos de fabricación. Las matrices de traslación y rotación son matrices que relacionan la posición de una cámara respecto a la otra.

3.2.1 Etapas del sistema.

El sistema de reconstrucción virtual de objetos mediante procesamiento digital de imágenes está estructurado de manera ordenada y dividida en procesos, partiendo desde la obtención de imágenes estéreo hasta la evaluación del sistema de reconstrucción. En la ilustración 2-3 se observa la estructura del sistema mediante un diagrama de flujo, partiendo desde la obtención de imágenes estero hasta la evaluación, nótese que el sistema requiere que el par de imágenes provenga de una cámara estero calibrada.

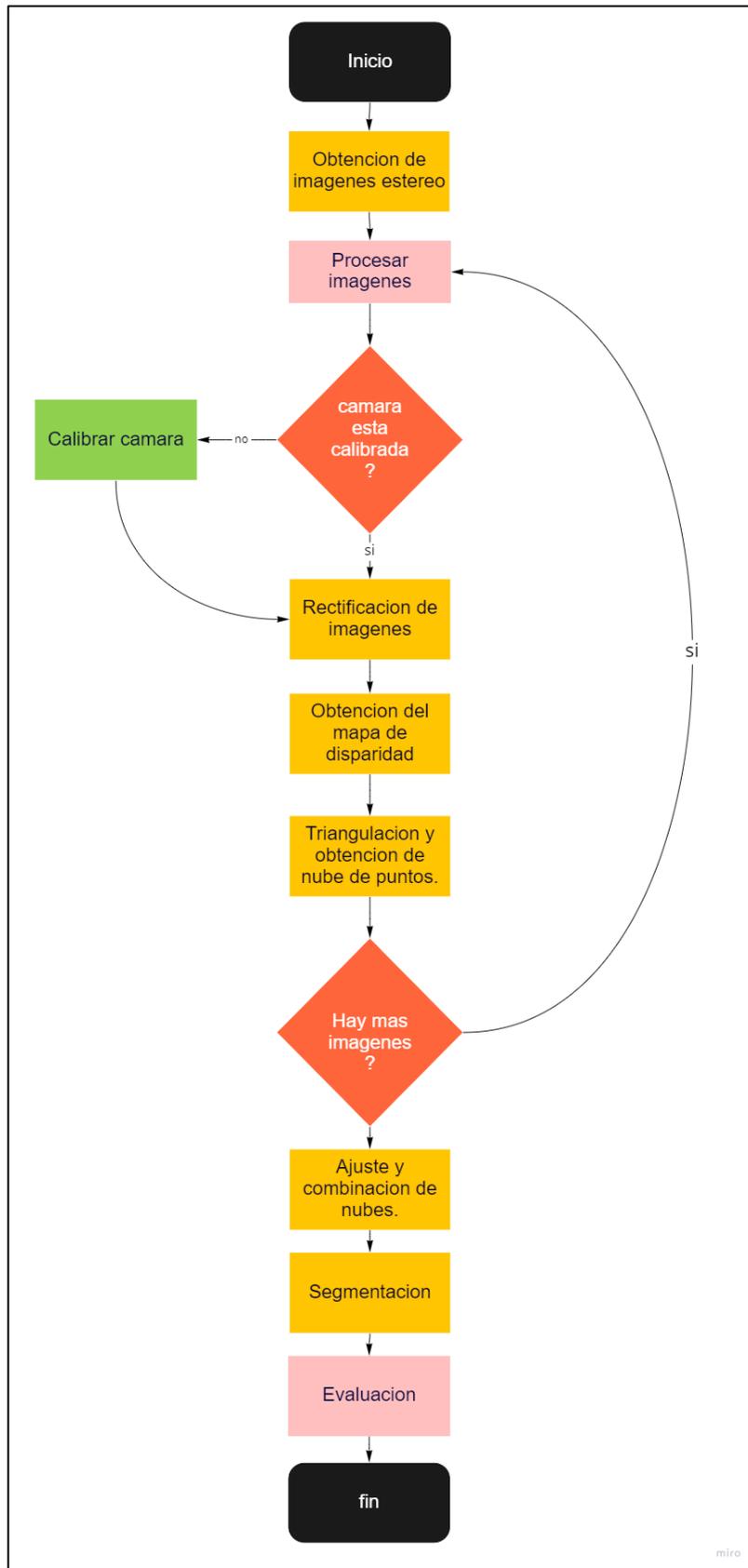


Ilustración 2-3: Diagrama de flujo del sistema de reconstrucción.

Realizado por: Castro, Gabriel; 2023

3.2.1.1 Obtención de imágenes mediante SDK MYNT EYE y Visual Studio.

La obtención de imágenes es la primera fase y una de las más importantes del sistema, para obtener imágenes se hace uso de cámaras estéreo S1030-IR-120/Mono de la marca MYNT, las características de esta cámara se presentan en la tabla 1-2, el control del hardware de la cámara es mediante dos aplicaciones. La una es el SDK (kit de desarrollo de software) provista por el fabricante de las cámaras estéreo y la otra es Visual Studio 2017 que es un compilador de código C++.

El fabricante recomienda no utilizar máquinas virtuales debido a los problemas de la virtualización con los controladores USB 3.0, por lo que es requerido un canal de alta tasa de transmisión de datos para la comunicación y envío de información desde la cámara hasta el computador.

El SDK proporciona una interfaz de tres niveles con dependencias de *OpenCV*:

- api, interfaz superior, con dependencias de *OpenCV*.
- device, interfaz entre capas, sin dependencias de *OpenCV*.
- uvc, interfaz inferior, sin dependencias de *OpenCV*.

Para el desarrollo de aplicaciones, se puede optar por deshabilitar las librerías de *OpenCV* desde el archivo *Option.cmake*, la opción *with API* a *OFF*. En este proyecto no se deshabilitó *OpenCV* por lo que se hizo uso de la interfaz superior API.

Para la obtención de un único par estéreo de imágenes, se hace uso del proyecto *save_single_image* desarrollado por (MYNTAI, 2019b, p. 31). Este código es desarrollado con *OpenCV*, es un código sencillo que captura imágenes manualmente con presionar las teclas ‘s’, ‘S’ o espacio y cierra el programa presionando la tecla ‘ESC’ o ‘Q’. En la documentación de la versión del SDK 2.5 se encuentra el detalle de cada una de las funciones que utiliza la interfaz superior API. El código llama a las funciones *Start()* y *Stop()* para empezar y detener la captura de imágenes y con *Source::video_streaming* definimos que solo queremos capturar imágenes. Cuando comienza la captura de datos, se llama a la función *WaitForStreams()*. Una vez que comience la captura de datos, se usa *GetStreamData()* para obtener los datos de cada cámara la izquierda y la derecha. Esto se hace en un ciclo *while* que es interrumpido por un *break* hasta que es presionado un *ESC* para finalizar el programa, mientras el *while* continua con la tecla de espacio podemos tomar fotos estéreo con una extensión de imagen *PNG*, estas imágenes son guardadas en la carpeta contenedora del proyecto.

El resultado del algoritmo es la salida de dos imágenes con una resolución de 752 x 480 píxeles sin corrección alguna, ejemplo en la ilustración 3-3.



Ilustración 3-3: Imágenes izquierda y derecha de la cámara Mynt Eye s1030.

Realizado por: Castro, Gabriel; 2023

3.2.1.2 Lectura y procesamiento de imágenes en Matlab.

Para el procesamiento de la imagen se hace uso de la herramienta matemática Matlab, esta no es compatible directamente con la cámara Mynt por lo que la obtención de imágenes se hace con el SDK y Visual Studio.

Las imágenes son cargadas en la carpeta contenedora del archivo *scanner.m*, archivo principal contenedor del algoritmo de reconstrucción 3D. Las imágenes son leídas mediante la función “*imread*” que tiene como argumento de entrada el nombre de la imagen junto con su formato, la función permite leer imágenes con formatos como: *PNG* que es el formato de salida de las cámaras estéreo. La salida de la función son tres matrices bidimensionales para imágenes *RGB* y solo una matriz bidimensional para imágenes en escala de grises. Los tipos de datos que se encuentran de acuerdo con la disposición de los píxeles se muestran en la tabla 1-3.

Tabla 1-3: Tipos de datos para formato PNG.

Bits de profundidad soportados	Clase de salida	Notas
8 bit	uint8	Grayscale or Indexed
24 bit	uint8	RGB
		Three 8-bit samples/pixel.

		Three	16-bit
		samples/píxel.	

Fuente: MATLAB; 2023

Realizado por: Castro, Gabriel; 2023

El tipo de dato uint8 define 8 bits para cada píxel representados por enteros de 0 a 255, un byte por píxel. Las líneas de código Matlab para la lectura de imágenes sin rectificar se muestra en la Ilustración 4-3.

```
%Read in the stereo pair of images.
InL = imread('pruebaleft.jpg');
InR = imread('pruebarigth.jpg');
```

Ilustración 4-3: Comandos Matlab para lectura y procesado de imágenes.

Relizado por: Castro, Gabriel; 2023

3.2.1.3 Calibración de imágenes estéreo.

Para la calibración de las cámaras, Matlab posee una herramienta didáctica de calibración de cámaras monoculares y estéreo, para este proyecto se utilizó la aplicación “Stereo camera calibrator” provista por Matlab, también existen otros métodos con los cuales se puede calibrar las cámaras Mynt, por ejemplo, con la ayuda de OpenCV o ROS. La aplicación es sencilla de utilizar, tiene como entrada imágenes estéreo de un patrón de calibración como el que se utiliza en el ajedrez “checkerboard”, el cual es utilizado para generar los patrones de calibración en este proyecto, aunque se puede utilizar patrones como cuadrículas circulares simétricas y asimétricas. Matlab define un proceso sencillo de pasos para calibrar cámaras estéreo como se observa en la ilustración 5-3.

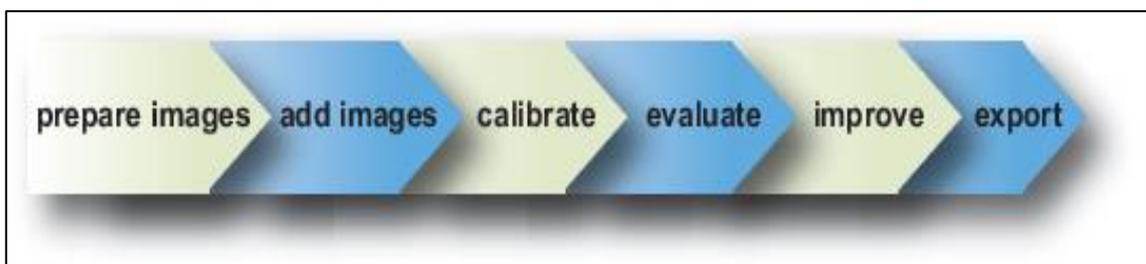


Ilustración 5-3: Patrón de calibración tablero de ajedrez.

Fuente: (MATHWORKS, 2022a)

Este proceso o secuencia de pasos presenta consejos y requerimientos que deben cumplirse para utilizar la App y obtener parámetros de calibración lo suficientemente buenos. La aplicación

requiere identificar un punto de origen único, por lo que es necesario un patrón de calibración sin una ambigüedad de 180 grados, es decir que el patrón no debe verse igual que cuando se gira a 180 grados.

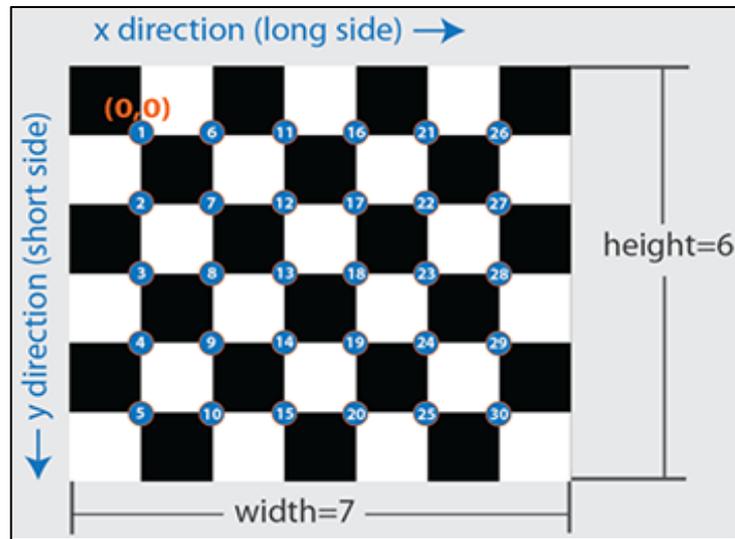


Ilustración 6-3: Patrón de calibración tablero de ajedrez.

Fuente: (MATHWORKS 2022a)

Para satisfacer esta condición el patrón debe tener un número par de cuadrados por un lado y un número impar por el otro como se observa en la Ilustración 6-3. Otro requerimiento importante es que el patrón de calibración tenga dos cuadrados negros en las esquinas de un lado y dos cuadrados blancos en las esquinas del otro lado, esto le permite a la aplicación determinar la orientación del patrón y el origen. La aplicación asigna al lado más largo al eje X y el lado más corto al eje Y.

Para la calibración de las cámaras Mynt Eye S1030 se utilizó un patrón de calibración de tablero de ajedrez de 11x10 vértices, cada cuadro con una medida de 3 cm de lado, el patrón de calibración debe ser colocado sobre una superficie plana con el objetivo de evitar deformaciones sobre la superficie del patrón que afecte la calibración.

Para la preparación de la cámara y captura de imágenes del patrón de calibración (MATHWORKS, 2022b) presenta algunas reglas y consejos que ayudan en la obtención de imágenes de patrones de calibración.

- Para capturar imágenes en cámaras con enfoque automático, se recomienda desactivar este y configurar el enfoque según la aplicación.
- Colocar el patrón de calibración a una distancia tal de la cámara de modo que este ocupe al menos el 20 % de la imagen.
- Capturar las imágenes de tal manera que el patrón sea visible para ambas cámaras.

- No cambiar la configuración interna de la cámara como *zoom*, contraste u otras entre imágenes, esto cambiara la distancia focal al momento de calibrar.
- Para una mayor precisión en la calibración es recomendable utilizar un mínimo de 10 a 20 pares de imágenes.
- Utilizar imágenes sin comprimir o imágenes en formato de compresión sin perdidas como *PNG*.
- La distancia que hay entre la cámara y el patrón de calibración es la distancia en la que se miden y se reconstruyen los objetos.
- Colocar la cámara en un ángulo inferior a 45° con respecto al plano de la cámara como se observa en la ilustración 7-3.
- Las oposiciones de las imágenes deben tener distintas orientaciones relativas a la cámara.
- Para cámaras con distorsión radial es necesario que al capturar las imágenes con el patrón de calibración éstas se tomen en las periferias horizontales y verticales debido a que la distorsión de la lente aumenta radialmente desde el centro de la imagen y no siempre es uniforme en todo el marco de la imagen.

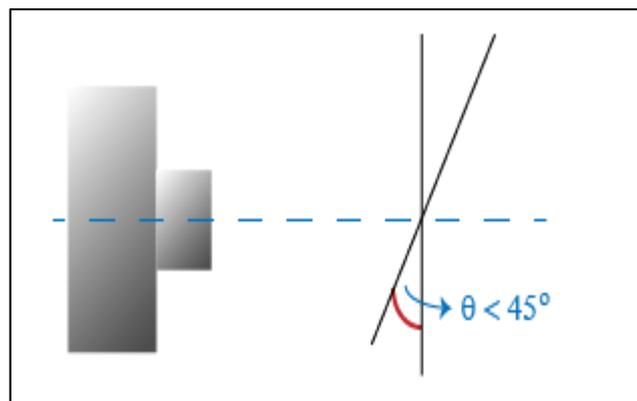


Ilustración 7-3: Obtención de imágenes con respecto al plano de cámara.

Fuente: (MATHWORKS 2022a)

La aplicación *Stereo Camera Calibrator* está ubicada en la pestaña aplicaciones, en la sección de procesamiento de imágenes y visión artificial o también se puede abrir desde el *command window* de Matlab escribiendo el comando *stereoCameraCalibrator*, el programa despliega una ventana en la que se puede agregar imágenes tanto de la cámara izquierda como de la cámara derecha, las cámaras Mynt entregan imágenes con nombre *left_x* y *right_x* respectivamente, siendo *x* el número de imagen que corresponda, esto facilita la identificación de cada imagen. La aplicación requiere la medida exacta del lado de uno de los cuadrados, esta medida es de 2,5 cm o 25 milímetros en este proyecto, también la de seleccionar el estado de la imagen en cuanto a

distorsión se refiere, esta puede ser alta o baja, como se ha mencionado para calibrar se puede utilizar patrones cuadrados o de círculos, por lo que la aplicación pide escoger con qué tipo de patrón se realiza la calibración. En la ilustración 8-3 se observa la ventana de imagen y propiedades del patrón de calibración.

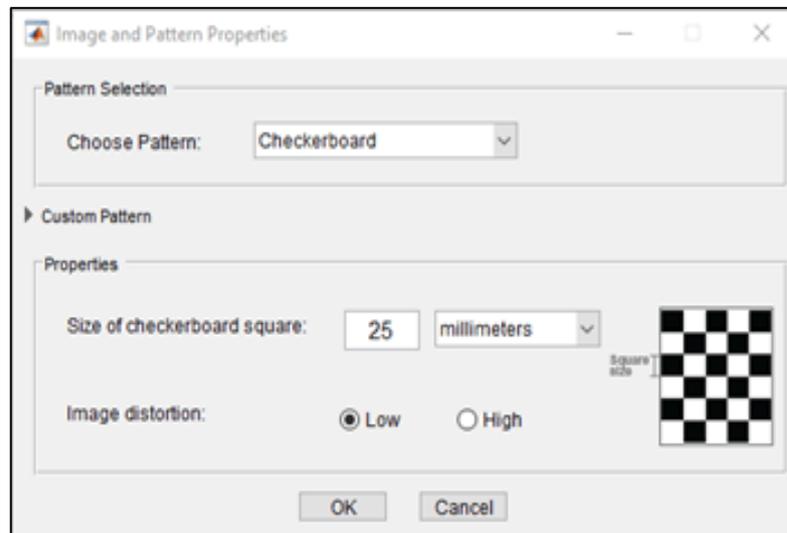


Ilustración 8-3: Stereo Camera Calibrator App.

Fuente: (MATHWORKS 2022a)

Posterior al agregar imágenes el programa analiza estas y detecta los patrones en cada imagen agregada, se muestran los resultados de cuantas imágenes fueron procesadas y cuantas de ellas son aceptadas y rechazadas. Las imágenes son rechazadas debido a que están duplicadas o están borrosas o no han podido detectarse los patrones completamente, la detección lleva más tiempo en patrones grandes y que contienen una gran cantidad de cuadros. En la ilustración 9-3 se observa la ventana de la aplicación con todas las imágenes que fueron aceptadas en la detección, cada par de imágenes presentan puntos verdes que representan la intercepción de los cuadros y un cuadro amarillo que representa el punto de origen de la imagen, además también se observa los ejes X y Y asignados. En la parte superior de la ventana se encuentran las opciones de calibración, estas permiten ajustar y seleccionar cuantos parámetros se requiere obtener, se ha decidido obtener los tres coeficientes de distorsión radial debido a la curvatura que las imágenes presentan en las periferías de estas, también se ha marcado la obtención de los coeficientes de distorsión tangencial debido a que el fabricante no presenta datos con respecto a la alineación de los sensores de imagen y se ha marcado la casilla de sesgo con el objetivo de corregir alguna deformación en los píxeles, también se presenta un casillero de optimización de opciones en donde se puede ingresar parámetros intrínsecos iniciales y parámetros de distorsión radial, estos no se colocaron debido a que el fabricante no ofrece estos datos.

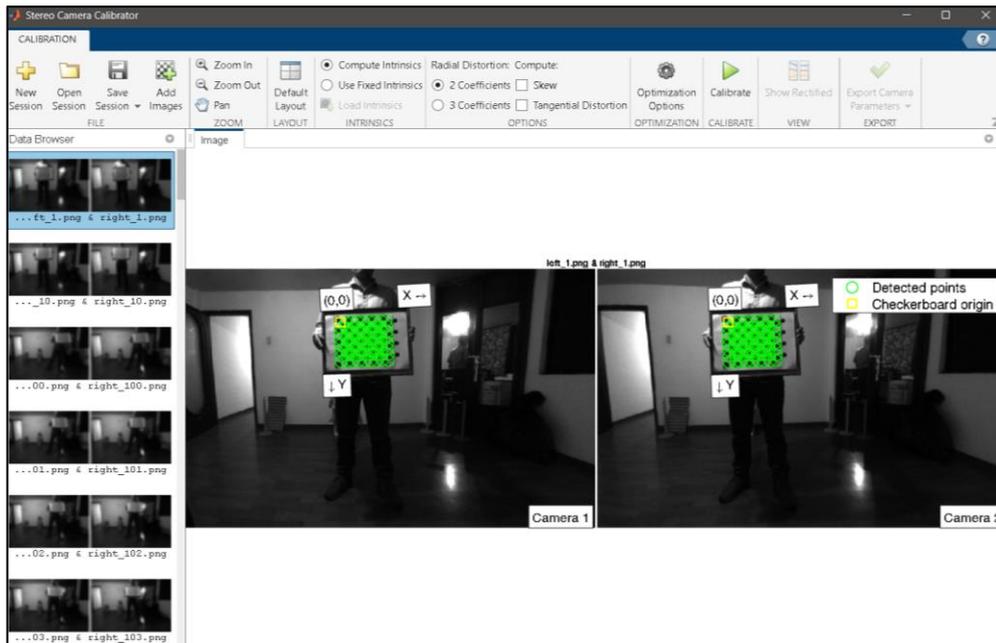


Ilustración 9-3: Stereo Camera Calibrator App: Agregación de imágenes.

Realizado por: Castro, Gabriel; 2023

Se puede analizar las imágenes antes de calibrar y borrar estas si se considera necesario, al calibrar con las imágenes seleccionadas la aplicación muestra los resultados de la calibración, se observan las imágenes con los puntos re proyectados y una estadística de errores, además de un diagrama físico de la obtención de patrones como se observa en la ilustración 10-3.



Ilustración 10-3: Stereo Camera Calibrator App: Re-proyección de imágenes.

Realizado por: Castro, Gabriel; 2023

Se puede evaluar la precisión de la calibración analizando los resultados obtenidos, en la ilustración 11-3 se muestran el error de proyección en pixeles de cada par de imágenes añadido,

si existe un par o varios pares de imágenes que presenten un error de proyección mayor a la media y que por lo tanto aporta negativamente a la calibración se puede recalibrar eliminando estos pares de imágenes de la sección de *data browser*. También se puede establecer un valor de error máximo, esto permite identificar los pares de imágenes que tengan un error de proyección mayor a ese valor establecido.

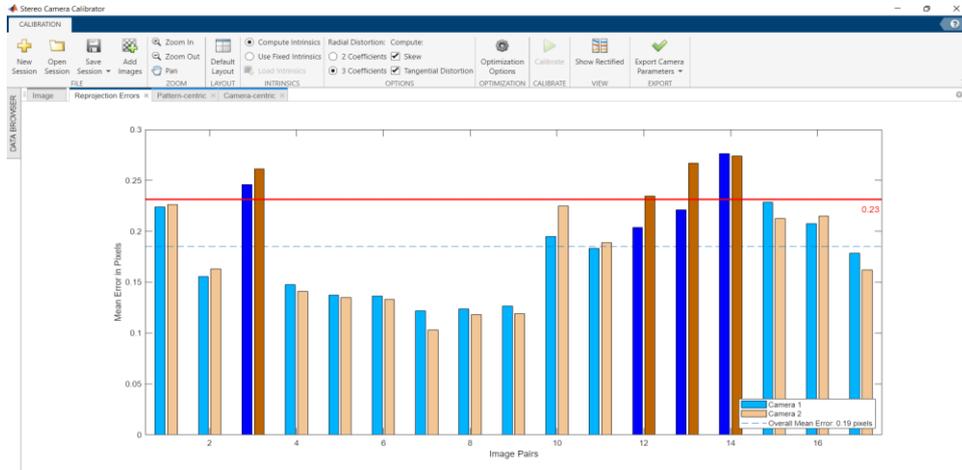


Ilustración 11-3: Stereo Camera Calibrator App: Media de error en pixeles.

Realizado por: Castro, Gabriel; 2023

Otra forma de evaluar es la observación del diagrama de cámara céntrica y patrón estático, mediante la observación se analiza si existen patrones con ángulos de inclinación mayor a 45 grados, o si a su vez hay cámaras que están detrás de patrón, esto puede ocurrir por algún error de calibración. En la ilustración 12-3 se muestran estos diagramas.

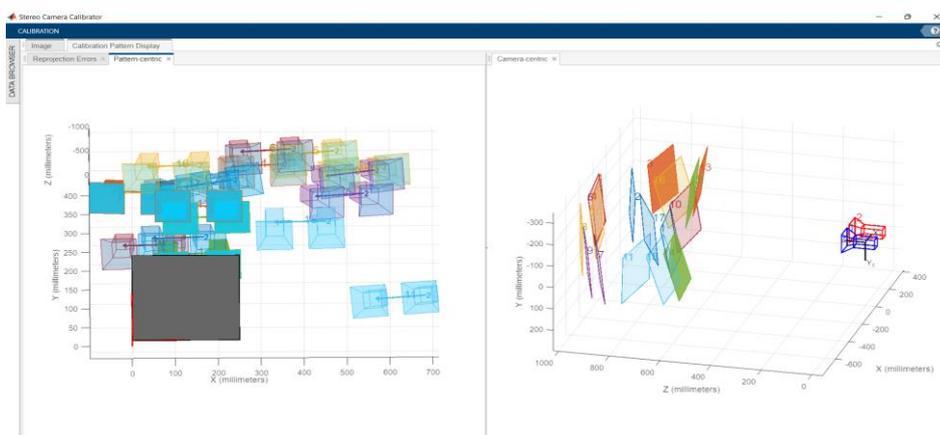


Ilustración 12-3: Stereo Camera Calibrator App: Diagrama de calibración.

Realizado por: Castro, Gabriel; 2023

Una última manera de evaluar la precisión de la calibración es verificando las imágenes rectificadas, se analiza las imágenes de entrada con las rectificadas, y si la rectificación es precisa las imágenes no se distorsionan y estas se alinean en fila como se muestra en la ilustración 13-3.

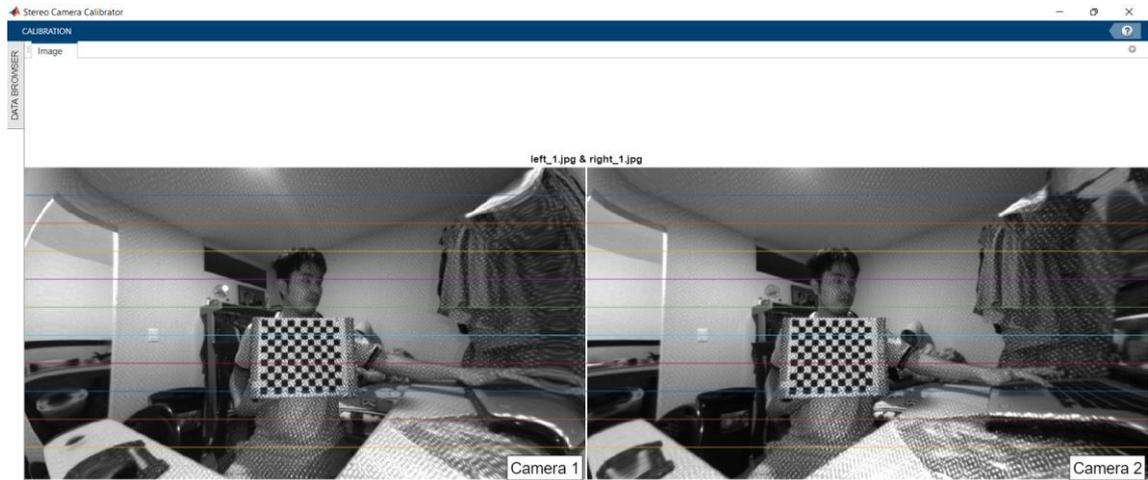


Ilustración 13-3: Stereo Camera Calibrator App: Resultados de rectificación.

Realizado por: Castro, Gabriel; 2023

Por último, para mejorar se pueden variar las opciones de calibración, eliminar imágenes con un error de proyección alto, eliminar imágenes borrosas en las que no se detecta correctamente el patrón de calibración, agregar más imágenes, variar la orientación del patrón con respecto a la cámara. Una vez se tenga los parámetros adecuados, estos se exportan al *Works pace* de Matlab.

3.2.1.4 Rectificación, disparidad, segmentación y triangulación de imágenes estéreo.

A partir de la calibración, Matlab ofrece funciones para la rectificación de imágenes, búsqueda de correspondencia de puntos entre pares estéreo y reconstrucción de escenas. La rectificación de las imágenes se realizó mediante la función *rectifyStereoImages*, esta función tiene como entrada las imágenes izquierda y derecha del objeto a reconstruir, y los parámetros de calibración obtenidos en el proceso anterior, la salida de esta función son dos imágenes rectificadas con una dimensión de 538 x 1178x3 pixeles con formato de tipo *uint8*. El proceso de rectificación conlleva a los ajustes de distorsión que existan en la imagen, corrección de la forma de los pixeles. También ayuda a búsqueda de la correspondencia en ambas imágenes, es decir que la búsqueda de puntos iguales en cada marco de imagen sea más precisa, la sintaxis de la rectificación se muestra a continuación, donde *OutL* y *OutR* son las imágenes rectificadas, *InL* y *InR* son las imágenes del objeto sin rectificar y *stereoParamsmil* son los parámetros de calibración. Este proceso con la calibración son los más importantes del proyecto, la rectificación realizara el recorte, el ajuste de posición de los pixeles con el objetivo de que cada píxel de coordenadas $(x1, y1)$ de la imagen 1 sea el mismo píxel con coordenadas $(x2, y2)$ de la imagen 2, en la ilustración 13-3 se observa este proceso en donde se ajustan en el eje 'y' los pixeles que corresponden al mismo punto.

[OutL, OutR] = rectifyStereoImages(InL, InR, stereoParamsml);

Las imágenes del antes y después de la función de rectificación se observan en la ilustración 14-3, en donde las imágenes rectificadas presentan una alineación en las periferias de su marco, a diferencia de las imágenes de entrada en donde las periferias forman una curva debido a la distorsión radial.

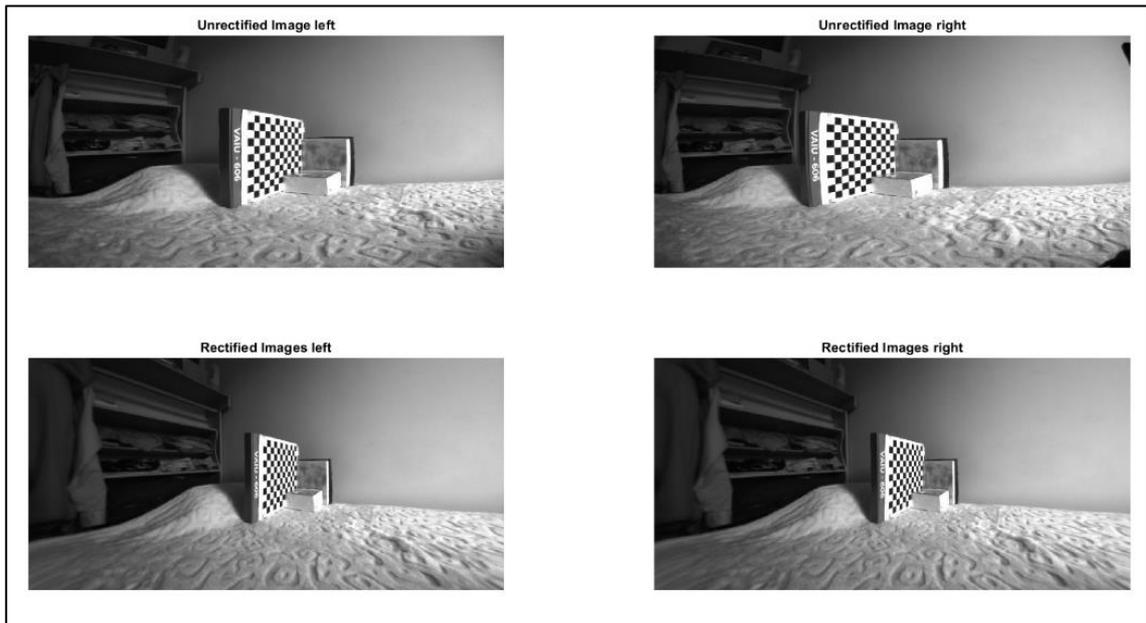


Ilustración 14-3: Rectificación de imágenes.

Realizado por: Castro, Gabriel; 2023

Para encontrar la disparidad vista en el capítulo 2, y que se define como la diferencia relativa de la posición de dos imágenes, se utilizó la función *disparityMap*, esta función utiliza el método *semi global Block matching*, que es una derivación del método definido por (Hirschmuller, 2008, p. 2), este método es un algoritmo de visión por computadora para la búsqueda de mapas de disparidad densos a partir de un par de imágenes estéreo rectificadas, midiendo la similitud de cada pixel en una imagen estéreo con cada pixel dentro de un subconjunto en la otra imagen estéreo. Este método es muy utilizado debido a su relación favorable entre la calidad de los resultados y el tiempo de cómputo. No obstante, se realizó la obtención de la disparidad mediante un segundo método, conocido como *block matching*, este es el más utilizado para la obtención de mapas de disparidad densos por ser simple y efectivo de implementar, el algoritmo consiste en segmentar la imagen en bloques de tamaño fijo y buscar para cada bloque el bloque correspondiente que proporcione la mejor coincidencia de la imagen de referencia. De manera general, el bloque que menor error de estimación proporcione es el bloque que se selecciona como

el bloque de coincidencia (Furht, 2008). Para el segundo método se utilizó la función *disparityBM* que tiene como entrada dos imágenes rectificadas y como salida una matriz del tamaño de las imágenes rectificadas. En la ilustración 15-3 se observan los mapas de disparidad para cada uno de los métodos.

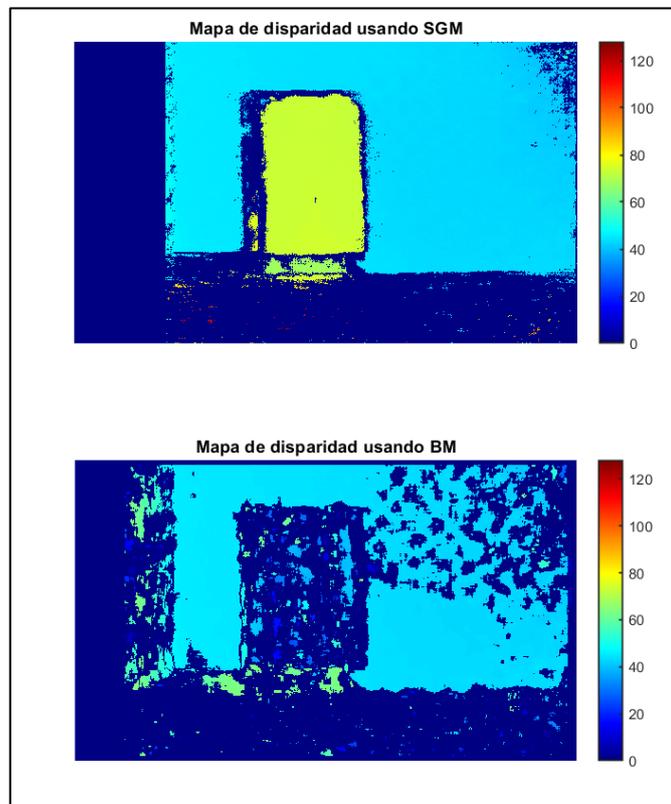


Ilustración 15-3: Mapa de disparidad mediante el método SGM y BM de una caja rectangular.

Realizado por: Castro, Gabriel; 2023

Los mapas de disparidad obtenidos por el método de correspondencia *semi-global block matching* son más precisos que los obtenidos por el método de correspondencia por bloques, por lo que el trabajo se desarrolló utilizando *SGBM* para reconstruir los objetos. La sintaxis para el cálculo de la disparidad se muestra a continuación.

```
disparityRange = [-16*1 256];  
disparityMap = disparity(OutLg,OutRg,'BlockSize',5,  
'DisparityRange',disparityRange,'UniquenessThreshold',11*1,'ContrastThreshold',0.05,'method',  
SemiGlobal');
```

La función acepta algunos parámetros de cuales se tiene:

- **OutLg**: imagen de cámara izquierda rectificada y en escala de grises.
- **OutRg**: imagen de cámara derecha rectificada y en escala de grises.
- **BlockSize**: Tamaño de bloque cuadrado. Este valor establece el ancho para el tamaño del bloque cuadrado. La función utiliza el bloque cuadrado de píxeles para realizar comparaciones entre I_1 e I_2 . El valor de este argumento es un número entero impar en el rango [5,255]
- **DisparityRange**: Rango de disparidad, especificado como el par separado por comas que consta de 'DisparityRange' y un vector de dos elementos. El vector de dos elementos debe tener el formato [MinDisparity, MaxDisparity]. Ambos elementos deben ser enteros y pueden ser negativos. MinDisparity y MaxDisparity deben estar en el rango [- ancho de imagen, ancho de imagen]. La diferencia entre MaxDisparity y MinDisparity debe ser divisible por 16.
- **UniquenessThreshold**: Valor mínimo de unicidad. Valor mínimo de exclusividad, especificado como el par separado por comas que consta de 'UniquenessThreshold' y un entero no negativo. El aumento de este parámetro hace que la función que marca más píxeles no sea confiable. Cuando el valor de singularidad de un píxel es bajo, la disparidad calculada para él es menos fiable.
- **ContrastThreshold**. Rango de umbral de contraste. Rango de umbral de contraste, especificado como el par separado por comas que consta de "ContrastThreshold" y un valor escalar en el rango (0,1). El umbral de contraste define un rango aceptable de valores de contraste. Al aumentar este parámetro, se marcan menos píxeles como no confiables.
- **Method**. Algoritmo de estimación de disparidad. algoritmo de estimación de disparidad, especificado como el par separado por comas que consta de 'Method' y cualquiera 'BlockMatching' o 'SemiGlobal'. La función de disparidad implementa los algoritmos básicos de Coincidencia de bloques y *Semi-Global Block Matching*. En el método 'BlockMatching', la función calcula la disparidad comparando la suma de las diferencias absolutas (SAD) de cada bloque de píxeles de la imagen. En el método 'SemiGlobal' de coincidencia, la función también fuerza una disparidad similar en los bloques vecinos. Esta restricción adicional da como resultado una estimación de disparidad más completa que en el método 'BlockMatching'.

Para la reconstrucción y triangulación de los objetos vista en el capítulo 2, y que consiste en encontrar la profundidad que es inversamente proporcional a la disparidad, se utilizó la función

reconstructScene, esta es una función propia de Matlab que utiliza un mapa de disparidad y los parámetros de calibración de la cámara estéreo, la función tiene como salida una matriz con las dimensiones de las imágenes rectificadas, la matriz de salida representa las coordenadas mundiales en tres dimensiones de los puntos de la escena a reconstruir. En la ilustración 16-3 se observa la salida de esta función.

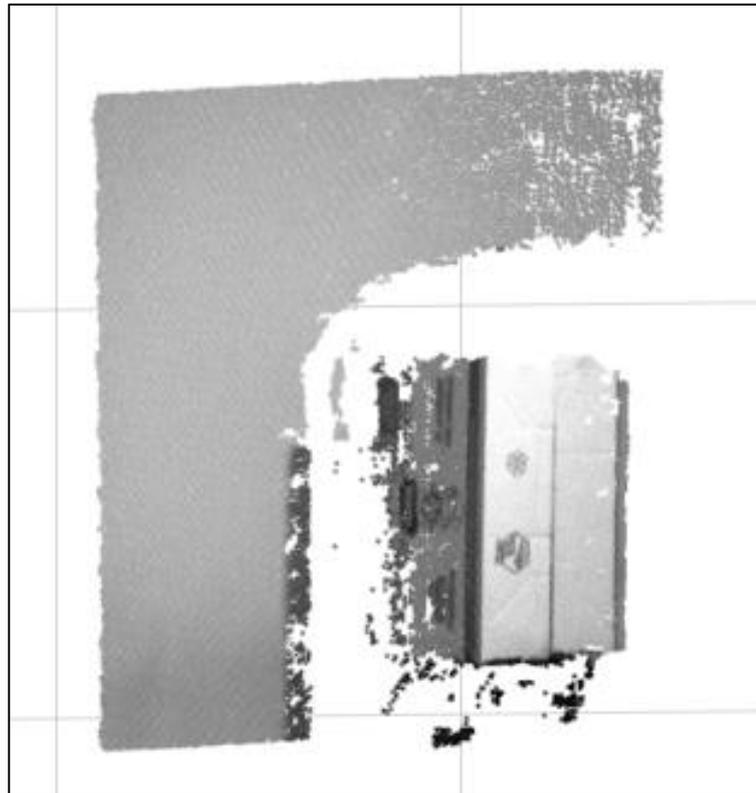


Ilustración 16-3: Nube de puntos de una sola vista de una caja rectangular.

Realizado por : Castro, Gabriel; 2023

Algo importante para tener en cuenta y en lo que se debe tomar el tiempo necesario es en el proceso de rectificación, este proceso se da mediante la obtención de buenos resultados de calibración, por lo tanto, una buena calibración permitirá rectificar las imágenes de la mejor manera, y ajustar los mismos puntos en las dos imágenes. El fin de todo esto es la búsqueda adecuada de la resta relativa en las dos imágenes que se conoce como disparidad, y con una buena disparidad los resultados de la profundidad serán mejores, es decir los valores de las coordenadas 'z' serán más aproximados a los valores de profundidad reales.

```
ptCloud = reconstructScene(disparityMap, stereoParamsml)/ 1000;
```

Para la segmentación se decidió la utilización de una matriz de coordenadas con los límites de las nubes de puntos en los tres ejes coordenados, lo que permite segmentar dentro de esos límites, la matriz de límites es la entrada de la función *thresholdPC*, esta función compara cada punto de la nube de puntos con los límites de la matriz de límites, los valores que no estén dentro de este rango se le asigna un valor NaN. La ilustración 17-3 muestra la salida de esta función.

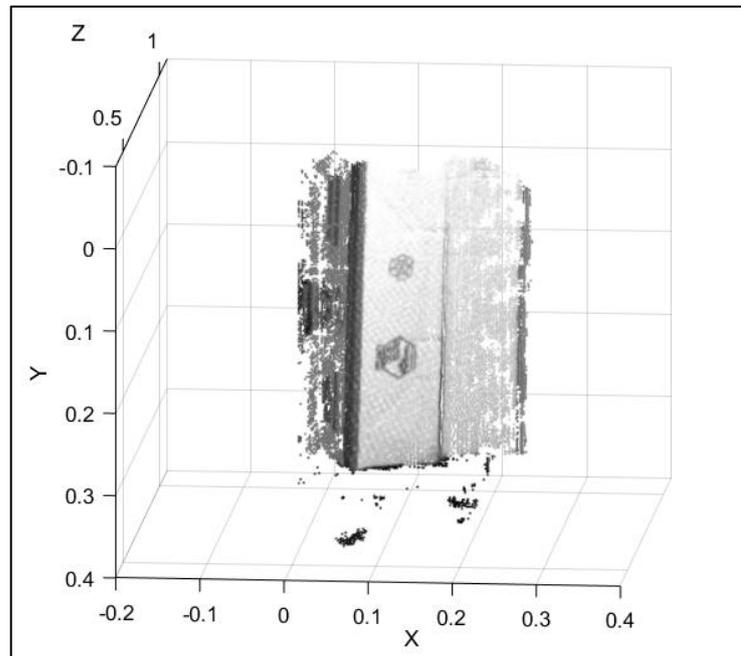


Ilustración 17-3:Nube de puntos de una sola vista de una caja rectangular segmentada.

Realizado por : Castro, Gabriel; 2023

3.2.1.5 Unión y costura de nubes de puntos.

Para la unión de nubes de puntos, se consideró la toma de pares de imágenes, en donde cada par de imágenes corresponde a una porción o vista del objeto. En el apartado siguiente se describe la geometría y diseño del sistema. La unión consiste en rotar y trasladar cada una de las nubes ajustando la orientación y la posición de cada nube respecto a la anterior, la función *pctransform* es la función que permite crear un objeto tridimensional modificado en su orientación y posición, esta función tiene como entrada la nube de puntos a ser modificada y un objeto *tform* que almacena información mediante la función *affin3D* sobre la transformación geométrica 3D, esta última función tiene como argumento una matriz A de 4×4 que describe la rotación y traslación del objeto, la sintaxis de la transformación se describe a continuación.

```

angle = pi/6;
A = [cos(angle),0, sin(angle),0;0, 1, 0, 0; -sin(angle), 0, cos(angle), 0; 0 0 0 1];
tform = affine3d(A);
ptCloudRef1 = pctransform(ptCloudRef, tform);

```

Las características de la función de salida *ptcloudRef1* es la misma que la función de entrada *ptcloudRef*, con una única diferencia en los límites de la nube de puntos de salida tanto en el eje *X* como en el *Z*, lo que significa que las coordenadas de la nube de salida han variado con respecto al eje *Y*, que es el eje estático en la transformación. La matriz *A*, es la matriz que define la rotación y traslación de cada nube de puntos. La función *affine3d* convierte la matriz *A* en un objeto de rotación y traslación que sirve como parámetro de entrada de la función *pctransform* que crea un nuevo objeto de nube de puntos. El proceso es similar en las nubes de puntos siguientes, básicamente consiste en cambiar la matriz de transformación para ajustar cada nube de puntos respecto a la anterior. En la ilustración 18-3 se muestran dos nubes de puntos que corresponden a la vista izquierda y derecha del objeto.

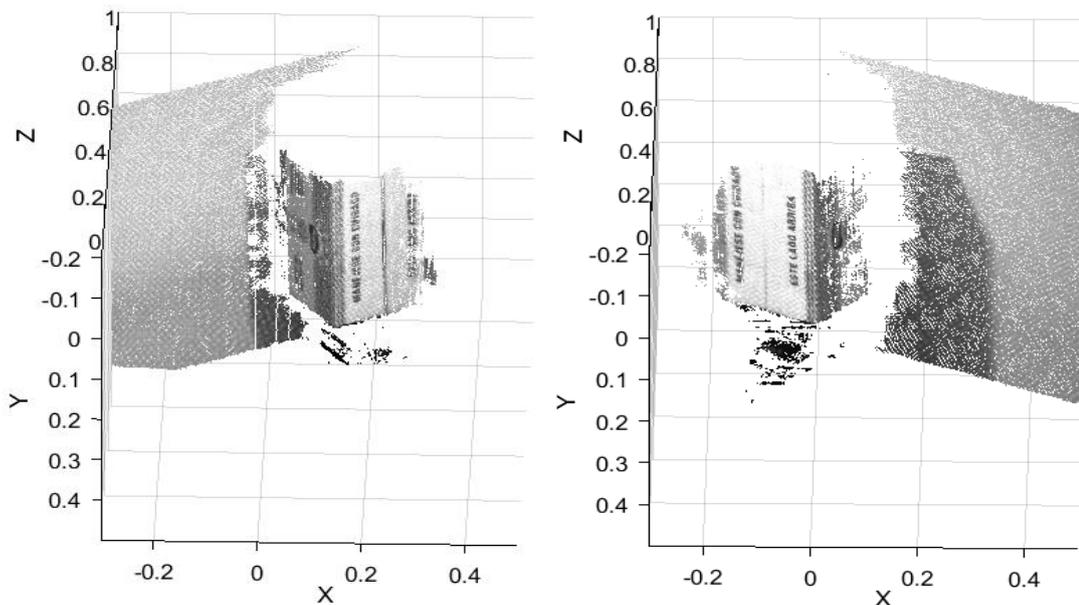


Ilustración 18-3: Vista izquierda y derecha del objeto.

Realizado por : Castro, Gabriel; 2023

En estas dos nubes se observa que mediante la vista frontal del objeto las nubes no son paralelas al eje *X*, por lo que requiere una rotación con respecto al eje *Y* de aproximadamente 30 grados en sentido horario y en sentido antihorario respectivamente para la vista izquierda y derecha. En la ilustración 19-3 se observa la rotación de cada vista.

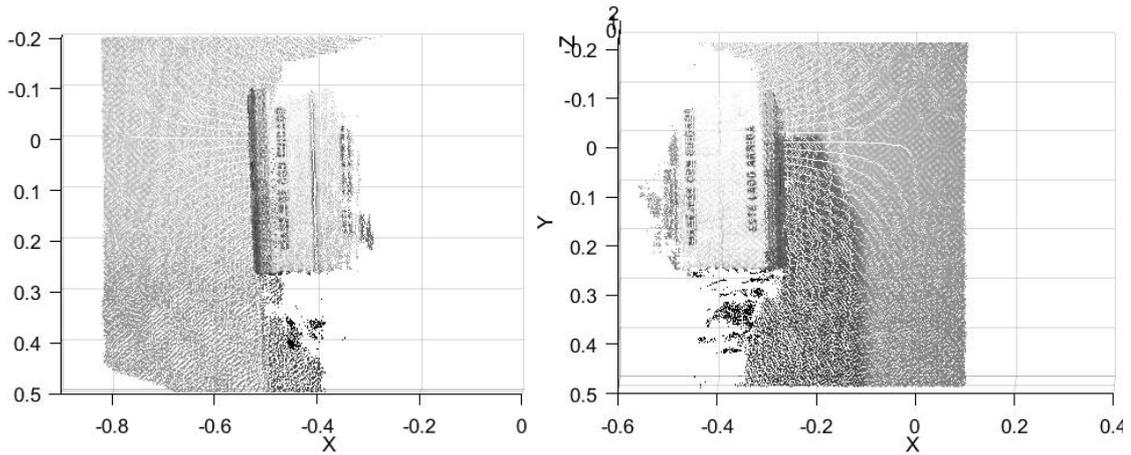


Ilustración 19-3: Vista izquierda y derecha del objeto paralelas al eje X.

Realizado por: Castro, Gabriel; 2023

Las vistas han sido rotadas para que sean paralelas al eje x, este procedimiento es realizado para cada una de las 4 vistas del objeto, el procedimiento siguiente es la traslación y ajuste de las nubes, esto se debe a que existen variaciones en la posición de cada nube al aplicar la rotación de estas. Para la traslación se aplica la misma matriz A cuya última fila describe la traslación de la nube en cada uno de los ejes coordenados. En la ilustración 20-3 se muestra la variación de posición de cada nube respecto a la anterior.

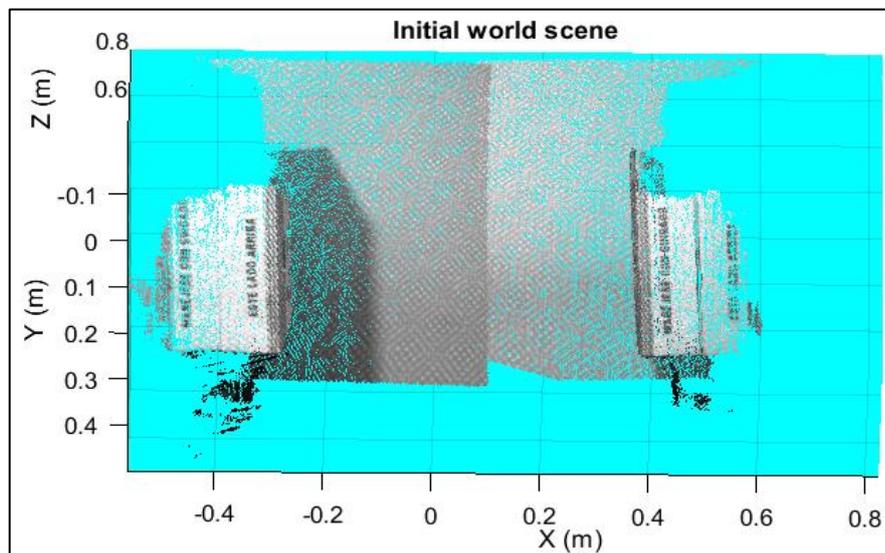


Ilustración 20-3: Vista izquierda y derecha del objeto en un solo plano.

Realizado por: Castro, Gabriel; 2023

Para ajustar se traslada la primera o la segunda nube en el eje x para hacer coincidir las caras, en este proyecto se tomó como referencia la primera nube de puntos correspondiente a la vista

izquierda, y todas las demás vistas se ajustan a esta para formar el objeto completo. En la ilustración 21-3 se muestra en resultado de trasladar y ajusta con respecto a la vista izquierda.

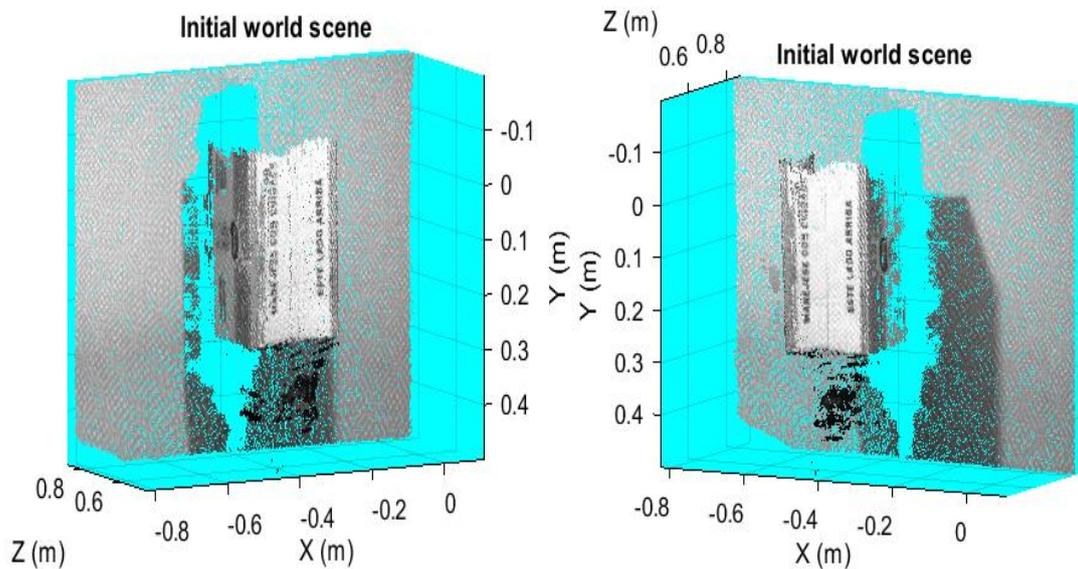


Ilustración 21-3: Objeto reconstruido con dos vistas.

Realizado por: Castro, Gabriel; 2023

Para unir las vistas izquierdas y derechas posteriores se realiza en mismo procedimiento, en la sección de resultados se observa el objeto reconstruido en su totalidad. El procedimiento de unión y costura de nubes es manual, lo que implica el ajuste de cada nube respecto a la otra variando ángulos y dimensiones con el afán de reconstruir el objeto de la manera más fiel.

3.2.2 Diseño del sistema de reconstrucción volumétrica.

Para la implementación del sistema de reconstrucción virtual de objetos, es necesario el diseño de un sistema que satisfaga las características de este, en el apartado de análisis y especificación de los requisitos del sistema se expusieron las características y requisitos que el sistema debe satisfacer, uno de los cuales es el diseño de un sistema que hace uso de múltiples vistas para reconstruir el objeto, las vistas son provistas por cada una de las cámaras, lo que requiere que el sistema sea un arreglo de cámaras estéreo de al menos cuatro cámaras colocados en los extremos de un arreglo cuadrado o circular que espacie las cámaras con la misma distancia en todos los lados.

Para el cumplimiento de este requerimiento en este proyecto se tomó un arreglo cuadrado como se observa en la ilustración 22-3.

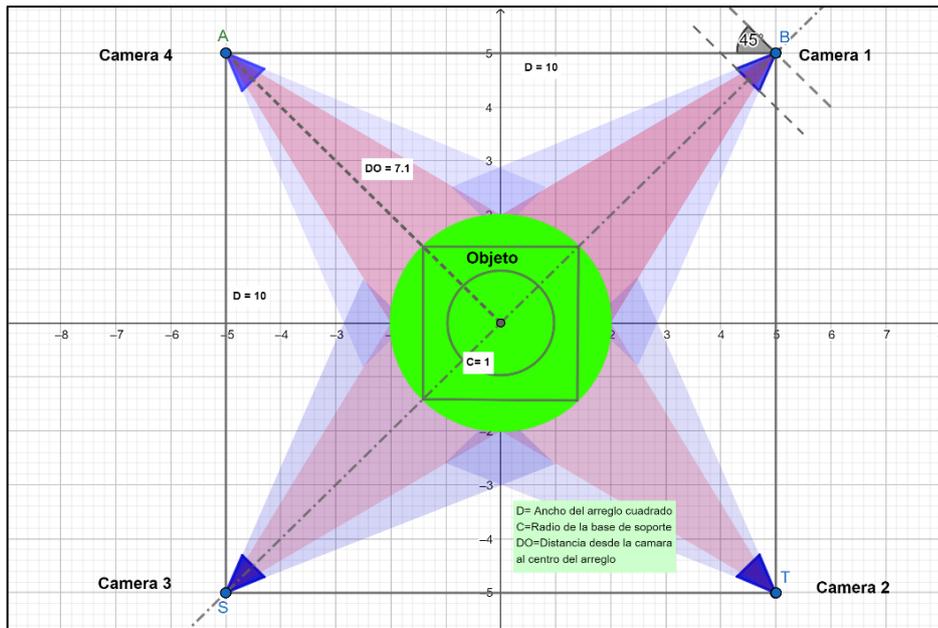


Ilustración 22-3: Geometría del sistema de reconstrucción.

Realizado por: Castro, Gabriel; 2023

Este diseño requiere la disposición de las cámaras en cada uno de los extremos del sistema, con un espaciado de 71 cm desde el centro del arreglo a cada una de las cámaras, el área de trabajo o de operación del sistema es de 1 metro cuadrado, la altura de cada una de las cámara es la misma y ha sido colocadas una altura de 80cm, cada cámara debe estar nivelada para obtener el mismo ángulo de visión en diferentes vistas, en el apartado de verificación, y pruebas unitarias se muestra el porqué de estas medidas. La dirección de cada cámara es paralela a las líneas que intersecan los vértices de la estructura cuadrada. En la ilustración 23-3 se muestra el diseño en 3D del sistema de reconstrucción.

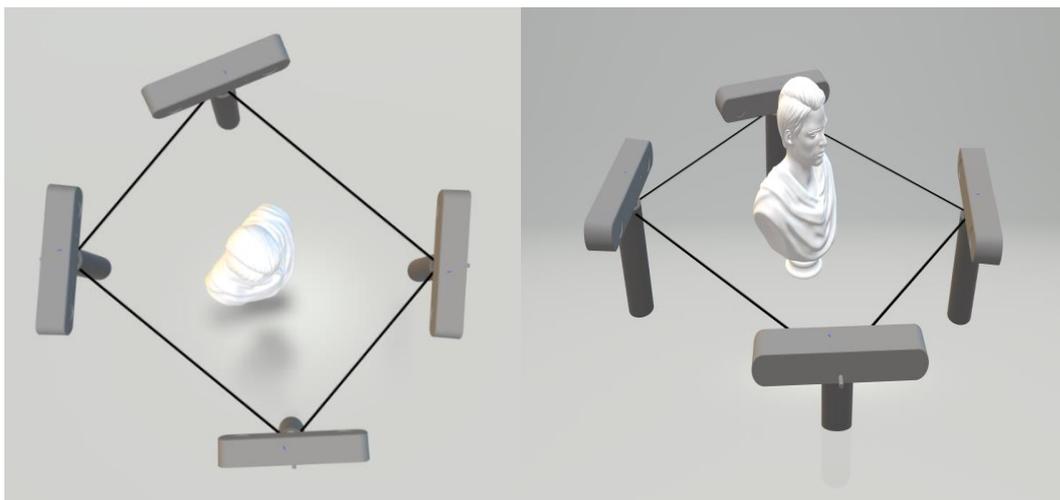


Ilustración 23-3: Diseño del sistema de reconstrucción.

Realizado por: Castro, Gabriel; 2023

3.3 Implementación del sistema de reconstrucción.

Para la implementación de la estructura se ha escogido el uso de madera para formar bases estacionarias que dispongan las cámaras de acuerdo con el diseño geométrico establecido en la sección anterior, cada base de madera dispone de un torillo ajustable que le da la capacidad de apretar las cámaras a disposición del usuario, la estructura es armable y desarmable cumpliendo con el requerimiento propuesta en la sección de requisitos del sistema con un sistema móvil y trasladable. Cada base debe está bien nivelada, por lo que la dirección de la cámara es perpendicular al eje de la base. La estructura cumple con el diseño establecido en todas sus medidas y geometría. En la ilustración 24-3 se observa la implementación del sistema de reconstrucción.



Ilustración 24-3: Implementación del sistema de reconstrucción.

Realizado por: Castro, Gabriel; 2023

Cabe resaltar que el sistema está configurado para funcionar de acuerdo con el diseño, alguna variación en la altura, ancho y ángulo de las cámaras afectara al funcionamiento del sistema. Esto se da debido a la ausencia de algún algoritmo automático para la unión de dichas nubes, por lo que el ajuste y unión es de manera manual mediante la matriz A establecida para el sistema diseñado. En la sección siguiente se discutieron las medidas del sistema de acuerdo con los resultados arrojados en cada etapa del sistema.

3.4 Verificación: pruebas unitarias y método de evaluación del sistema.

En esta sección se abordó la verificación mediante pruebas unitarias, las pruebas unitarias consisten en la comparación de los resultados hasta la etapa de reconstrucción, y la evaluación del sistema

que compruebe la validez del sistema por medio de la comparación de las medidas reales y virtuales de los objetos. Para la verificación del funcionamiento de cada etapa, se hubo planteado la toma de un solo objeto como imagen de referencia, a partir de este objeto se evaluaron los resultados correspondientes a cada etapa, partiendo desde la toma de imágenes buscando encontrar que parámetros de imagen son los más adecuados al capturar fotos de cada vista parcial del objeto, siguiendo el diagrama de flujo se realizó la calibración de cada cámara, por consiguiente, se tiene la rectificación, en este apartado se realizó la verificación de los procesos anteriores mediante el análisis de las imágenes rectificadas con las imágenes sin rectificar.

En la disparidad, la cual corresponde a una de las etapas más cruciales del sistema, se analizó que mapa de disparidad presenta mejores resultados en cuanto a la profundidad del objeto, por último, la reconstrucción es la etapa final del sistema, en esta parte se visualiza la digitalización parcial del objeto.

3.4.1 Configuración de los parámetros de imagen.

En este apartado se analizaron los parámetros de imagen como ganancia, brillo, contraste y cuál es la mejor relación entre estos parámetros. En este apartado la validez de estos parámetros se realizó mediante el último proceso de reconstrucción y triangulación, por ahora solo nos centramos en el resultado final de reconstrucción para validar el brillo, la ganancia y el contraste de las imágenes. Para identificar el valor adecuado de cada uno de los parámetros de imagen se realizó la captura de fotos variado cada parámetro a la vez, los valores establecidos de los parámetros que no cambian se establecieron con los datos por defecto que vienen configurados de fabrica en la cámara y estos valores son ganancia 24 de un rango de [0 48], de brillo en 120 de un rango de [0 240], y contraste en 116 de un rango de [0 254].

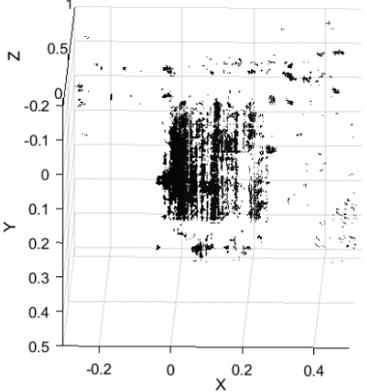
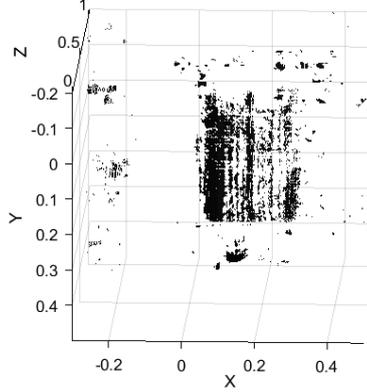
La ganancia definida como un factor numérico de conversión entre el número de fotoelectrones capturados en un dato numérico definido entre un valor de [0 48], que es el valor de luminiscencia de cada píxel y que por lo tanto determina su nivel de brillo o de luz. En la tabla 2-3 se muestra la variación de la ganancia mediante imágenes de la cámara izquierda del sensor estéreo y el objeto reconstruido, con el brillo y el contraste en su valor por defecto. El valor de la ganancia es variado cada 8 unidades desde 0 hasta su valor máximo 48. Las pruebas realizadas se han tomado en un cuarto con una luminiscencia de referencia de 21-22 lux, estas medidas se tomaron mediante las aplicaciones móviles de luxómetro *Lux Light Meter Pro* y *Lux Light Meter Free*, en la ilustración 25-3, se observan estas aplicaciones.

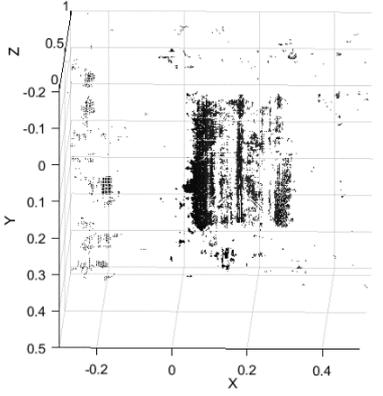
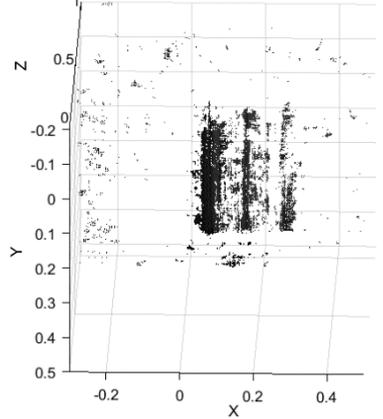
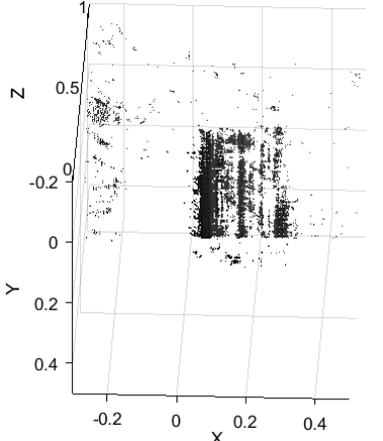
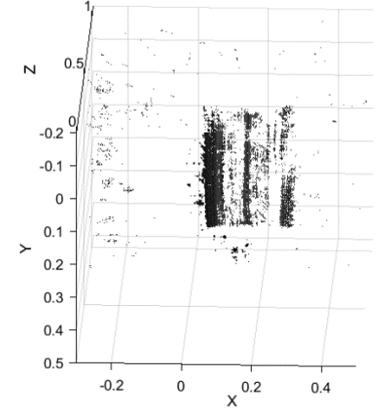


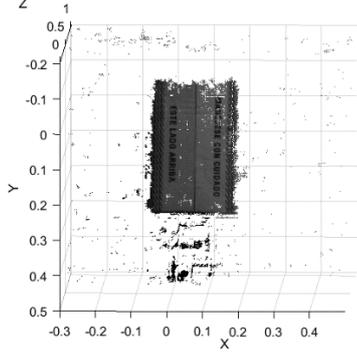
Ilustración 25-3: Luxómetros lux Light Meter Pro y Free.

Relizado por: Castro, Gabriel; 2023

Tabla 2-3: Variación de la ganancia de la cámara.

Ganancia	Imagen	Nube de puntos del objeto.
0		
8		

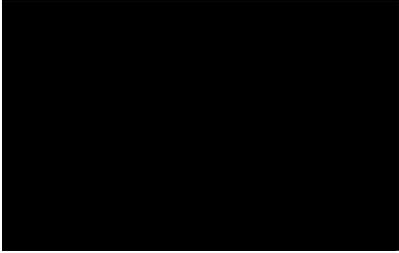
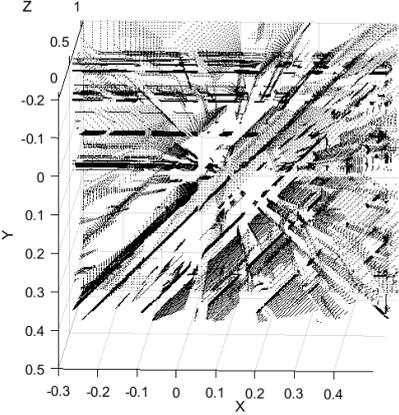
16		
24		
32		
40		

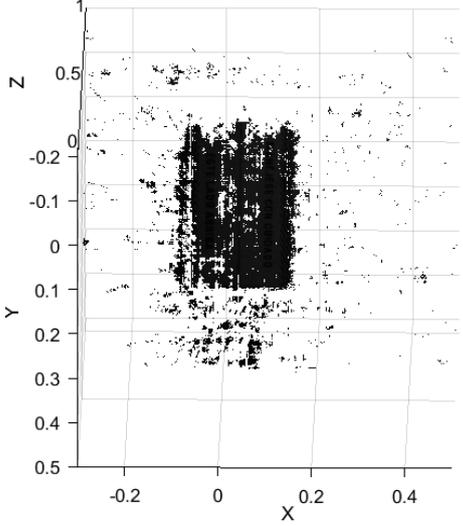
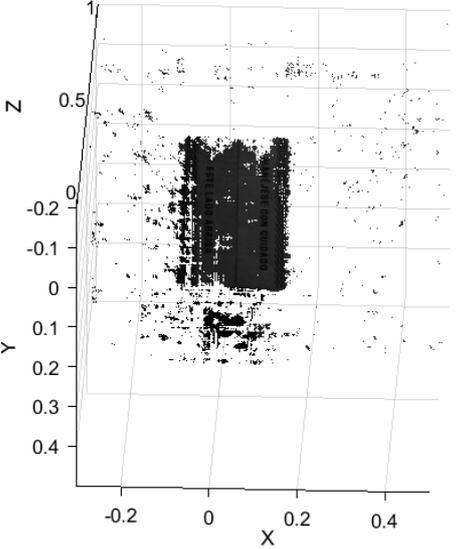
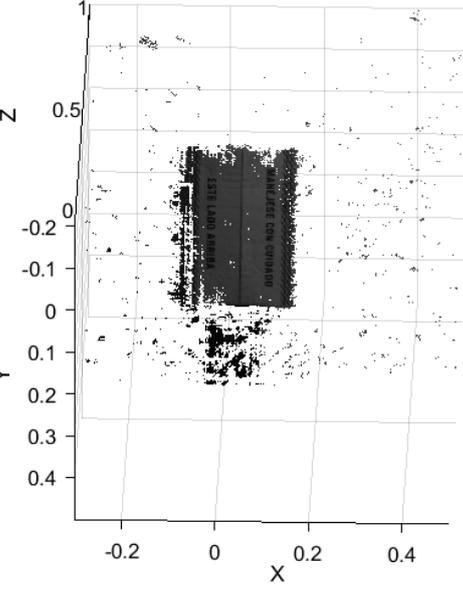
48		
----	---	--

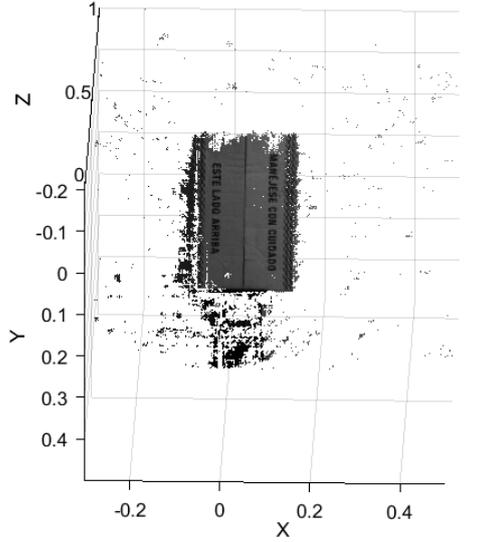
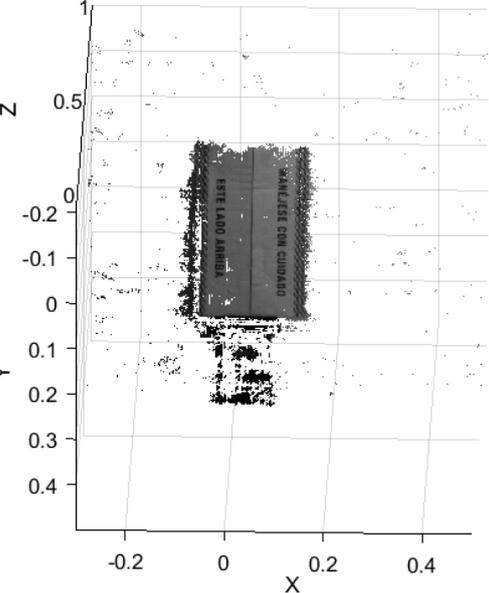
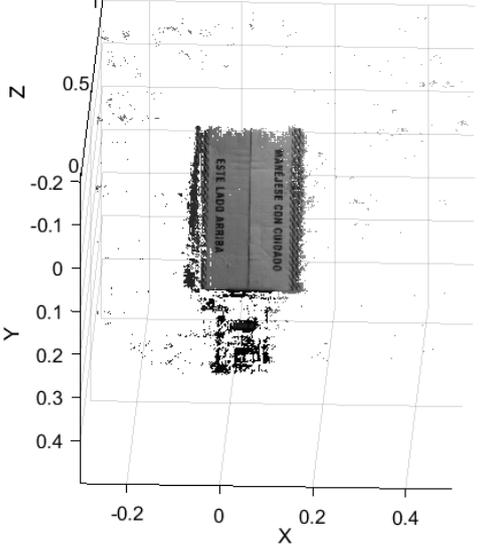
Realizado por: Castro, Gabriel; 2023

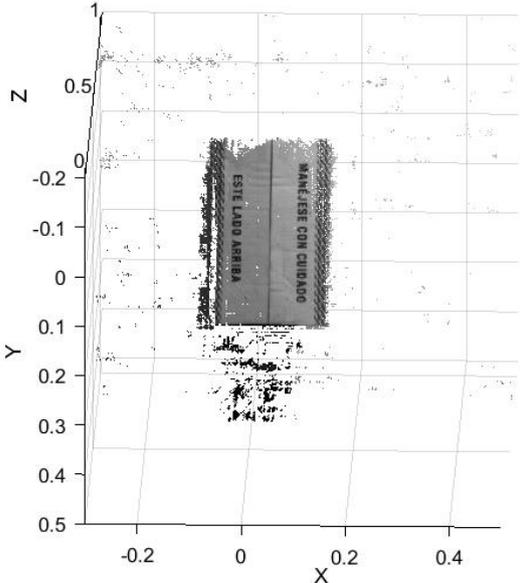
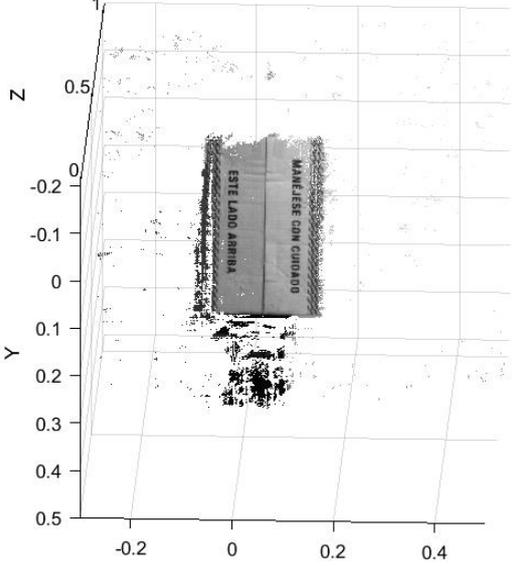
La tabla 2-3 muestra los resultados de la reconstrucción, con una ganancia de 0 hasta una ganancia de 48 que es su valor máximo, desde 0 hasta 40 el objeto no se pudo diferenciar, lo que implica que el valor de ganancia más óptimo es de 48, otras cámaras como las ZED requieren de un valor de iluminación lo suficientemente alto y estable, aunque el objeto fue tomado en un ambiente constante y buena iluminación, en imágenes en escala de grises es importante una gran variedad de niveles de grises para poder diferenciar los colores y por ende la profundidad del objeto, en las imágenes con una ganancia de 0 hasta 16 la imagen no se diferenciaba del marco de la imagen, pero desde la 24 hasta la 40 la reconstrucción del objeto fue nula en su mayor parte, lo que requiere un nivel de factor de iluminación o ganancia de 48 para reconstruir los objetos. Desde este apartado en adelante se tomó un valor de ganancia de 48 para la búsqueda de los parámetros restantes. Para la búsqueda del valor de brillo se ha establecido la ganancia en su valor máximo y el contraste en 116, la variación del brillo se realizó, en unidades de 30. En la tabla 3-3 se muestra la variación de este dato de configuración. Tomando en cuenta que el brillo se extiende en un rango de [0 240].

Tabla 3-3: Variación del brillo de la cámara.

Brillo	Imagen	Nube de puntos del objeto.
0		

30		
60		
90		

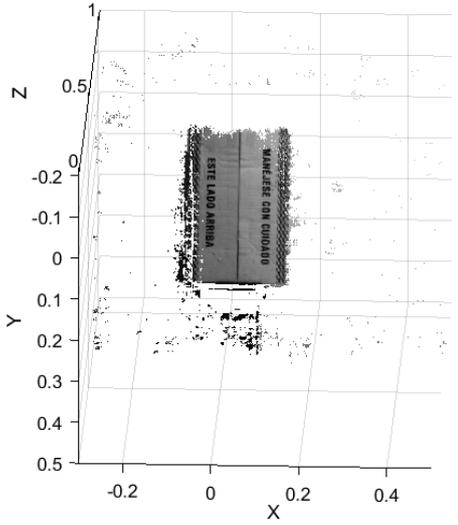
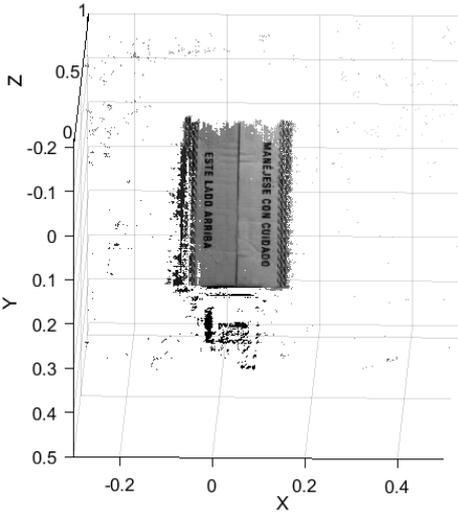
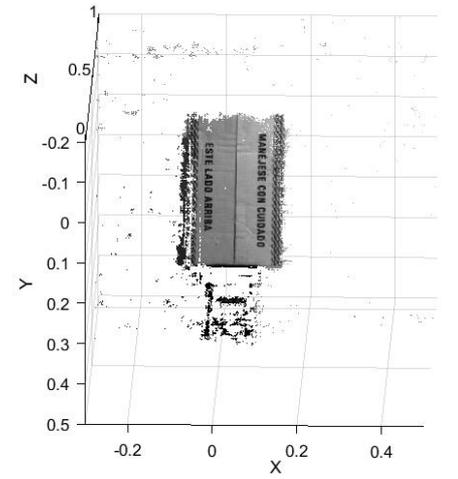
120		
150		
180		

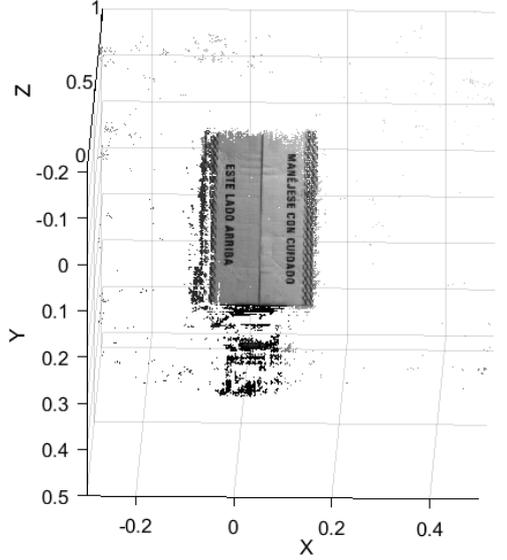
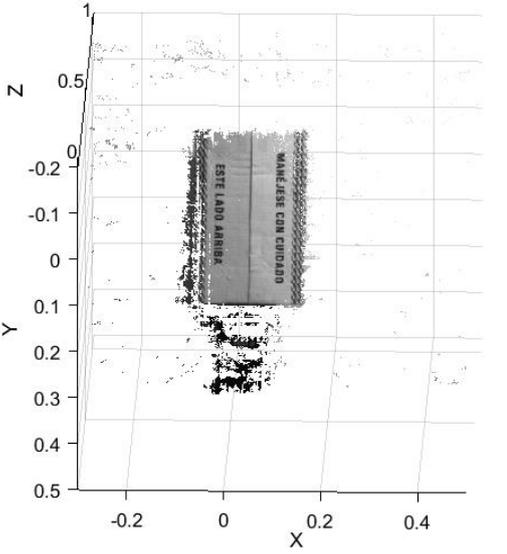
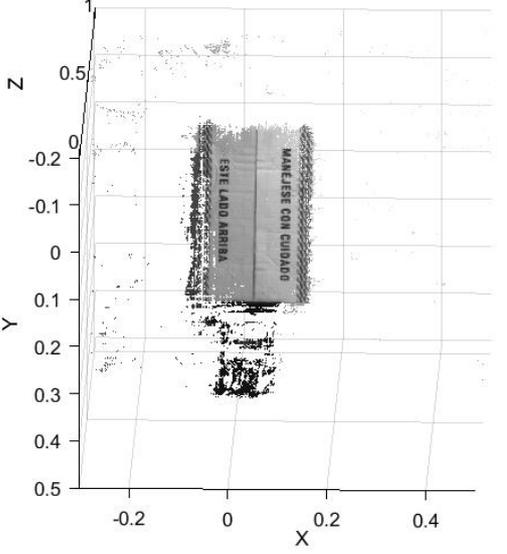
210		
240		

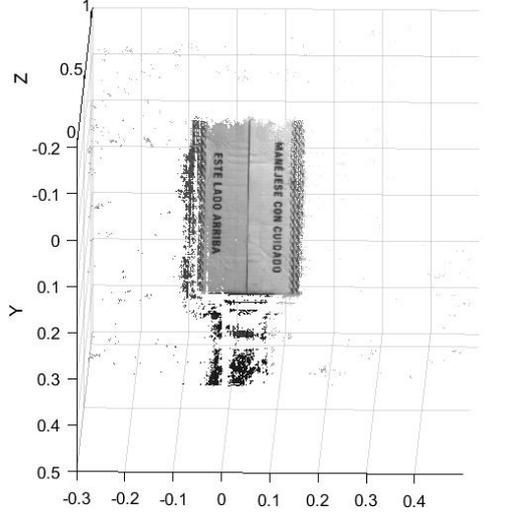
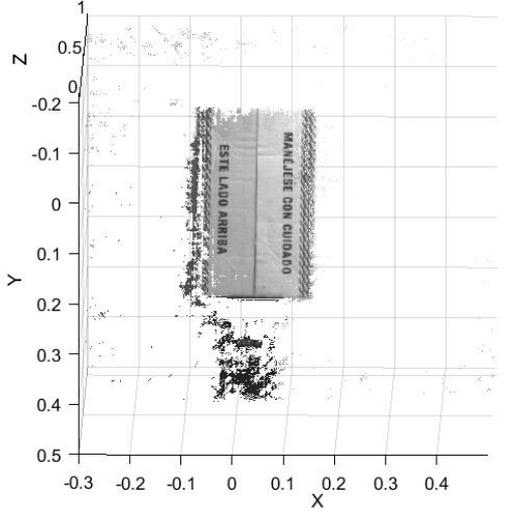
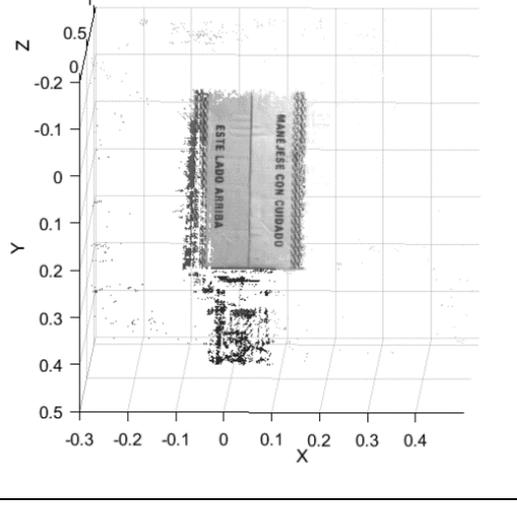
Realizado por: Castro, Gabriel; 2023

Para el parámetro de operación del brillo fue bastante similar a la de la ganancia, con un valor de [0 60] el objeto no se reconstruyó correctamente debido a la similitud de los pixeles con el marco de la imagen, mientras que desde [90 240] el objeto se diferenció del fondo de la imagen, con el valor más alto de brillo el objeto pareció reconstruirse en su mejor manera, aunque el ruido generado ocasiona que el objeto no se muestre en su forma natural. Para la búsqueda del último parámetro, es decir el contraste se tomó la ganancia a su máximo valor y el brillo en 240. El contraste ajusta la diferencia entre los tonos más oscuros y claros de una imagen, este parámetro de operación puede ser ajustado en un rango entre [0 254]. Por lo que este fue variado en unidades de 32. En la tabla 4-3 se observa la variación de este valor.

Tabla 4-3: Variación del contraste de la cámara.

Contraste	Imagen	Nube de puntos del objeto.
0		
32		
64		

<p>96</p>		
<p>128</p>		
<p>160</p>		

192		
224		
254		

Realizado por: Castro, Gabriel; 2023

Para el valor del contraste todos los resultados presentaron características similares, se podría trabajar con cualquiera de ellos, pero los datos en el rango de [96 128] mantienen una uniformidad

de puntos en toda la cara del objeto, por lo que este proyecto se tomó el valor de 116 que es el valor que viene por defecto y es definido por el fabricante, y que está dentro del rango aceptable.

3.4.2 Parámetros de calibración.

Para calibrar se tomó en cuenta los consejos dados por (Matlab, 2022) en la sección de calibración, la ilustración 26-3 muestra la captura del patrón de calibración de la cámara izquierda y derecha del sensor estéreo.



Ilustración 26-3: Patrón de calibración, vistas izquierda y derecha.

Realizado por : Castro, Gabriel; 2023

Cada cámara estéreo tiene un identificador único que sirve de referencia para la identificación de estas, este identificador ha sido otorgado por la universidad, se han llamado a los parámetros de calibración tomando en cuenta los últimos dos dígitos de este identificador, así hemos llamado a los parámetros *17calibration.mat* correspondientes a la cámara 16399990000-33950017. En la tabla 5-3 se muestran las matrices: fundamental, esencial, rotación, traslación y error de reproyección de cada cámara estéreo.

Tabla 5-3: Parámetros de calibración.

Cámara	Matriz Fundamental	Matriz Esencial	Matriz de Rotación	Matriz de traslación	Error de Re-proyección
17	3.3333166329783 8e-08 1.1636956392684 4e-05	0.00471112675464 562 1.65047586961452	0.99994693478549 0 0.00051024518984 9441	-119.9151893 63933 -0.545969655 940633	0.148384547239 894

	-0.004225496 39933205 -2.035468235 19011e-05 -3.192124790 39070e-06 0.3258023979335 48 0.0063302264530 3090 -0.322430106 529511 1.3324268791432 3	-0.539695402 136399 -2.881918756 60055 -0.455532085 051980 119.891021165225 0.61180155325675 4 -119.9140331 30215 -0.448419845 891645	0.01028918184061 86 -0.0005492287 47937404 0.99999268103289 0 0.00378632122727 265 -0.0102871745 822416 -0.0037917714 1978415 0.99993989644808 9	-1.648417742 19230	
18	-7.927893264 36390e-08 1.3879324832773 8e-05 -0.002851558 62360991 -1.763405689 424 77e-05 -1.879909374 63051e-06 0.3265554623462 61 0.0054141186762 2160 -0.324340259 925691 0.8018676397676 79	-0.011184053 9256998 1.96239072356208 0.16523777428195 5 -2.492453133 65261 -0.266297674 804705 120.297791683961 0.48093673259596 0 -120.3067735 12862 -0.253724332 190727	0.99997606574099 5 0.00532735661276 961 0.00441443277010 984 -0.0053366804 5668134 0.99998354838291 7 0.00210304665038 591 -0.0044031564 6607239 -0.0021265547 3261422 0.99998804491759 0	-120.3078944 71660 0.16111191270 9676 -1.962765627 53106	0.089813824588 5228
19	6.8606395597349 5e-09 -8.432899382 41777e-06 0.0025599779644 3641 3.8770922371027 9e-06 1.8743765702807 1e-06	0.00091915030597 8721 -1.131031857 72269 0.23599753143132 5 0.52020573562748 2 0.25108637888618 6	0.99998525396745 9 0.00187603267059 324 0.00509630739405 055 -0.0018654183 8990108 0.99999608295121 0	-119.8578329 00851 0.23363993363 6616 1.13152159690 962	0.074898058438 9599

	0.3252931476322 68 -0.000925152 555719638 -0.324063760 496 004 -1.500657164 26114	119.861782025947 -0.010051482 7076135 -119.8578017 28021 0.24891605248146 8	-0.0020866951 0660767 -0.0051002021 3975940 0.00207715759060 027 0.99998483656227 4		
20	5.3559229867550 3e-08 -3.244665493 27348e-05 0.0088703974810 7810 2.6700738742116 3e-05 9.2684663564868 0e-07 0.3055967297320 99 -0.006739468 48612469 -0.304697539 765532 0.3440369152426 26	0.00785520302504 38 -4.7663602273 4058 0.32084945538606 4 3.92566254625363 0.13794290982315 2 121.373504129443 -0.060085066 632722 -121.3437914 52029 0.12745147494167 3	0.99997376698675 9 0.00211435708836 006 0.00692783028190 451 -0.0021070050 1403158 0.99999720955976 7 -0.0010683644 2733879 -0.0069300698 5410807 0.00105373942778 029 0.99997543158071 4	-121.34346241 4517 0.31576463381 7882 4.76670626219 406	0.113877007690 145

Realizado por: Castro, Gabriel; 2023

De acuerdo con (OPENCV, 2017), el error de re-proyección es un buen indicador para estimar cuan exactos son los parámetros encontrados, cuanto más aproximado sea este valor a cero, más precisos serán los parámetros de calibración.

3.4.3 Rectificación de imágenes.

La rectificación de imágenes consiste en ajustar cada par estéreo de modo que estén alineados horizontalmente, la geometría epipolar vista en el capítulo 2 describe el método utilizado para optimizar la coincidencia de píxeles en ambas imágenes, esta geometría consiste en establecer a cada imagen con los mismos puntos característicos del objeto en una línea que corta horizontalmente a los mismo puntos y así realizar la búsqueda de puntos iguales variando cada pixel de manera vertical hasta encontrar el pixel que corresponde a ambas imágenes. Esta

rectificación logra por lo tanto ajustar verticalmente cada imagen de modo que la búsqueda se la realice de manera horizontal y así optimizar la coincidencia de puntos iguales. Una buena coincidencia depende la rectificación de cámara estéreo, ya que a partir de estos parámetros de rectificación se realiza el ajuste de las imágenes, una calibración ineficiente limitaría la rectificación y los procesos posteriores. En la ilustración 27-3 se muestra el resultado de rectificar. (Matlab, 2023) establece que la rectificación estéreo proyecta las imágenes en un plano de imagen común de modo que los puntos correspondientes tienen las mismas coordenadas de fila, lo que resulta como si las dos cámaras estuvieran en paralelo.

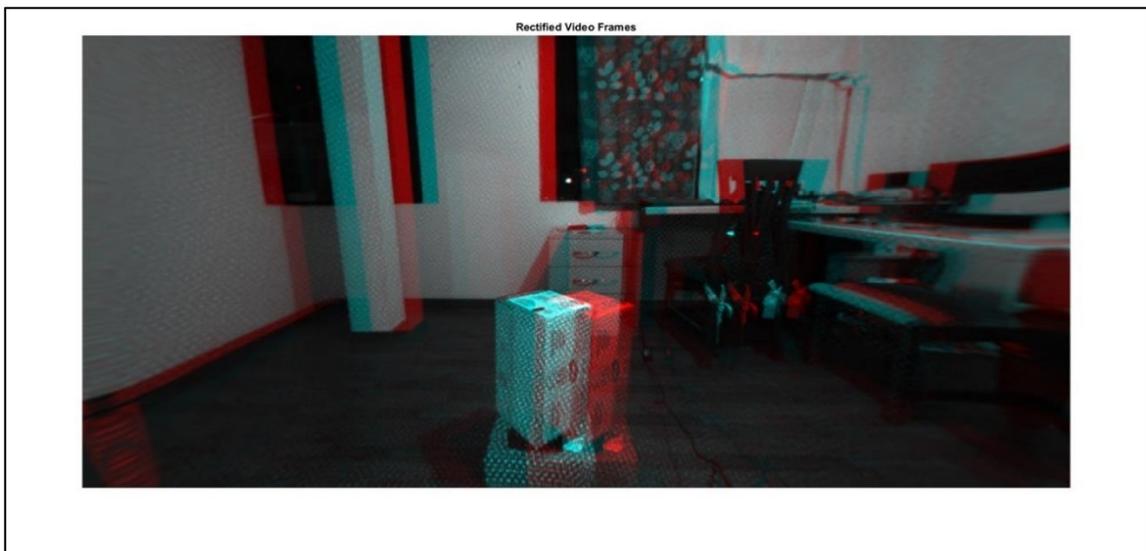


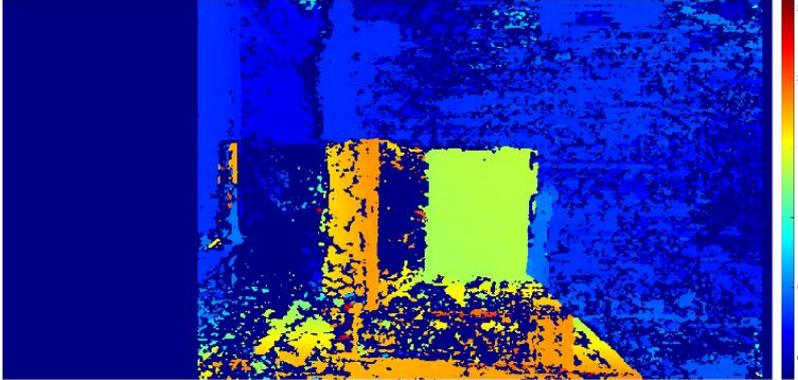
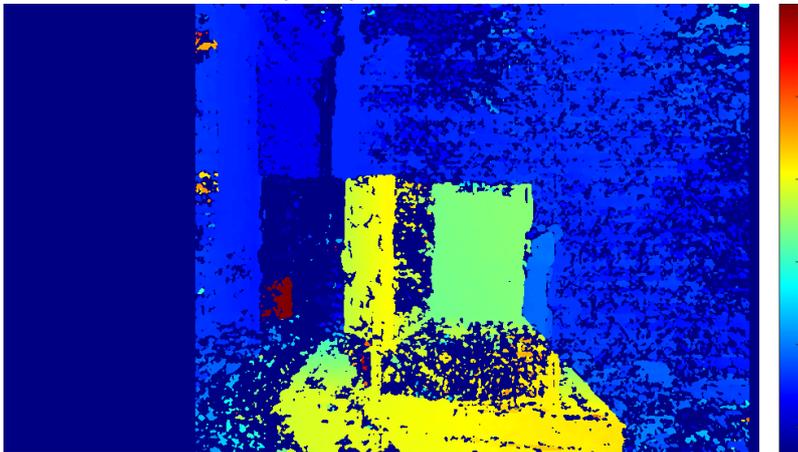
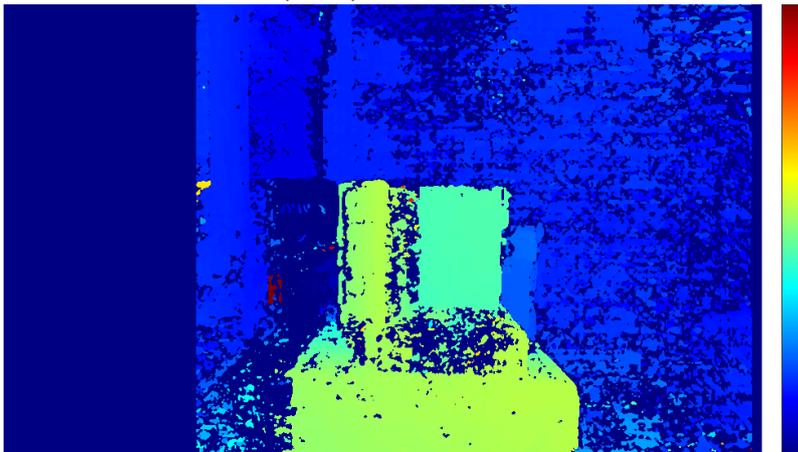
Ilustración 27-3: Rectificación de imágenes.

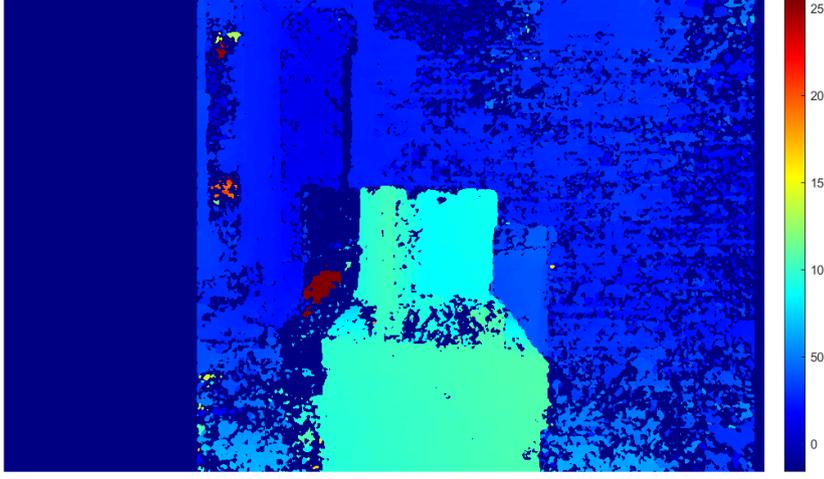
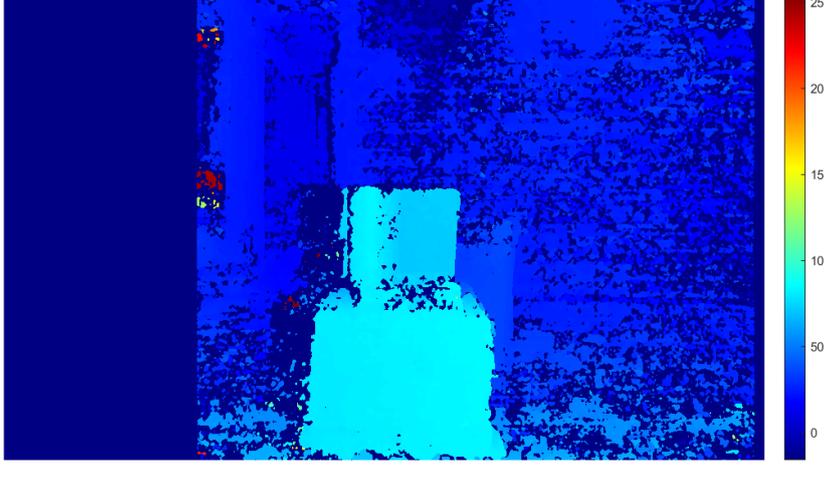
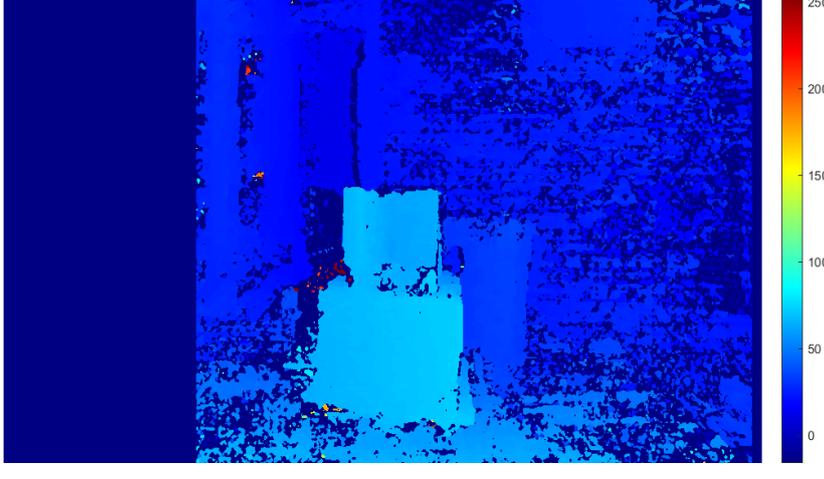
Realizado por: Castro, Gabriel; 2023

3.4.4 *Disparidad.*

En esta subsección se centró en la búsqueda de las medidas del arreglo, básicamente consistió en la toma de imágenes variando la distancia desde el objeto a la cámara, con el fin de encontrar dimensiones cuyos resultados de reconstrucción fueran aceptables. Estas medidas deben obedecer a una buena relación tamaño y distancia, lo que implica que para objetos pequeños a una distancia muy grande provoca pérdida en el número de puntos al reconstruir, y a una distancia muy corta el objeto no se reconstruya uniformemente. En la ilustración 14-3 se observan los objetos de prueba, la medida de distancia de la cámara al objeto se varió cada 5 cm. Estableciendo un valor de partida de 30 cm, para un objeto conformado por dos cajitas de 17x12 cm. En la tabla 6-3 se visualiza el mapa de disparidad de cada distancia. El fabricante ofrece información sobre la medida de profundidad entre la cámara y el objeto y establece que la reconstrucción óptima esta entre 0,5 y 18 metros.

Tabla 6-3: Disparidad vs Distancia.

Distancia [cm]	Mapa de profundidad
30	<p data-bbox="810 349 957 365">Mapa de disparidad usando SGM</p>  <p>The image shows a disparity map for a 30 cm distance. The scene is a textured surface with a central rectangular feature. The color scale on the right ranges from 0 (dark blue) to 250 (red). The central feature is predominantly yellow and green, indicating a disparity of approximately 100-150. The surrounding textured area is mostly blue, indicating a disparity of 0-50.</p>
35	<p data-bbox="759 826 995 842">Mapa de disparidad usando SGM</p>  <p>The image shows a disparity map for a 35 cm distance. The scene is the same as the 30 cm map. The color scale on the right ranges from 0 (dark blue) to 250 (red). The central feature is predominantly yellow and green, indicating a disparity of approximately 100-150. The surrounding textured area is mostly blue, indicating a disparity of 0-50.</p>
40	<p data-bbox="759 1386 995 1402">Mapa de disparidad usando SGM</p>  <p>The image shows a disparity map for a 40 cm distance. The scene is the same as the previous maps. The color scale on the right ranges from 0 (dark blue) to 250 (red). The central feature is predominantly yellow and green, indicating a disparity of approximately 100-150. The surrounding textured area is mostly blue, indicating a disparity of 0-50.</p>

50	<p>Mapa de disparidad usando SGM</p> 
60	<p>Mapa de disparidad usando SGM</p> 
70	<p>Mapa de disparidad usando SGM</p> 

Realizado por: Castro, Gabriel; 2023

Se verifico que por la baja coincidencia que tienen las cámaras del par estéreo al estar a una distancia menor de 70 cm, cada mapa presento incongruencias como huecos y bordes desaliñados. Por lo que se establece esta última distancia como una medida estándar para objetos pequeños y medianos. La altura de cada cámara se ha diseñado en 80 cm, la altura de la base se ha planteado en 10 cm con el fin de escanear las superficies superiores de objetos con una altura máxima de 38 cm de altura, pero la altura de la base puede variar para objetos pequeños. Para la búsqueda de rango de disparidad, el software Matlab ofrece la herramienta *imtool* que trae una regla para medir la separación más grande y pequeña del par estéreo. Con esto se estableció el vector de dos elementos [*min disparity*, *max disparity*] en [0,80], en la ilustración 28-3 se muestra la búsqueda de este parámetro.

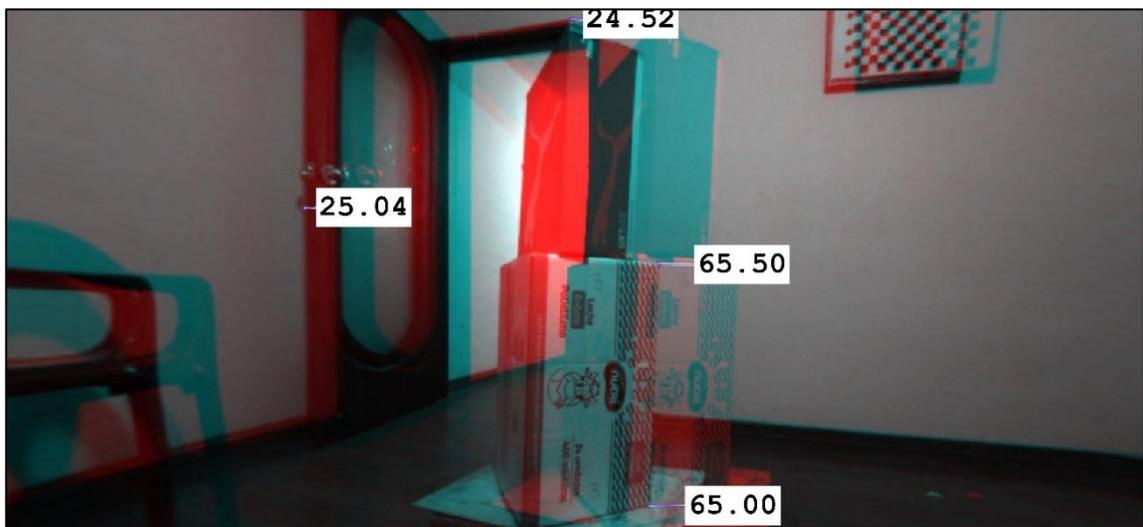


Ilustración 28-3: Rango de disparidad.

Fuente: Castro, Gabriel; 2023

3.4.5 *Reconstrucción.*

La reconstrucción de cada vista parcial del objeto es la consecuencia de la búsqueda de un mapa de disparidad, en la ilustración 29-3 se muestran más vistas correspondientes a 4 caras del objeto. La reconstrucción por lo tanto es el proceso final de este proyecto, las nubes de puntos fueron generadas mediante los pasos anteriores descritos en la metodología del diseño. Cada vista parcial tiene huecos en las partes que la coincidencia de puntos no se da por no existir puntos que relacionen ambas imágenes. Cada vista es de aporte apositivo para el objeto final, la vista que logra captar alguna parte del objeto que otra vista no puede por estar geoméricamente localizada en una esquina del arreglo, combina esa información tridimensional con su vecina vista. Estableciendo puntos de referencias propios del objeto como esquinas, o superficies planas de fácil identificación para la posterior unión de las nubes de puntos.

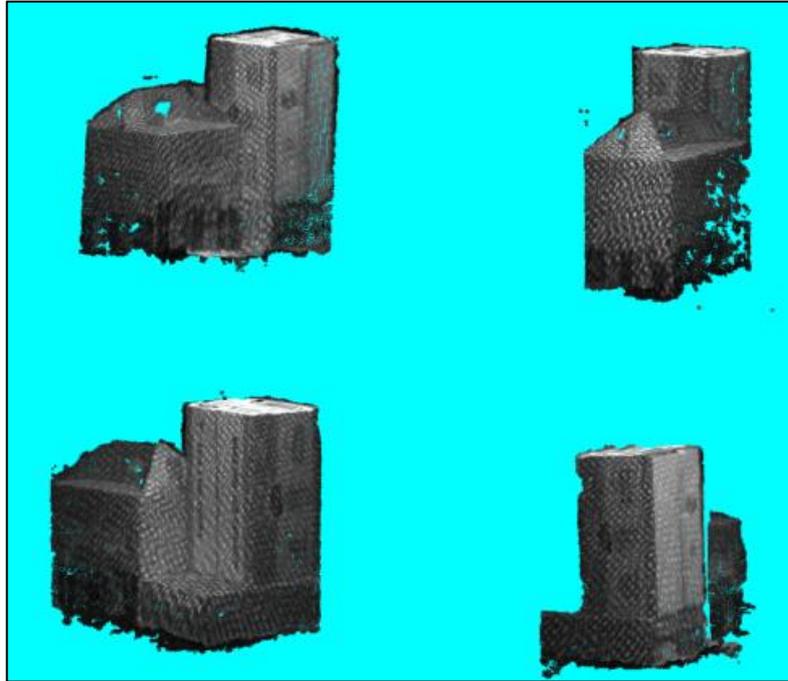


Ilustración 29-3:Reconstrucción 3D de las caras del objeto.

Realizado por: Castro, Gabriel; 2023

3.4.6 Costura de nube de puntos.

La costura y unión de nubes, consistió en establecer las matrices de rotación y traslación de manera que cada una se alinee con su vecina vista. Estas matrices forman la matriz descrita en el apartado 3.2.1.5, llamada matriz A. En la tabla 7-3 se muestran las matrices para cada vista.

Tabla 7-3: Matrices A y B para cada vista del objeto.

Vista	Matriz A				Matriz B				Angulo α grados	Angulo θ grados
1	1	0	0	0	cos(theta)	0	sin(theta)	0	0	0
	0	cos(alfa)	sin(alfa)	0	0	1	0	0		
	0	-sin(alfa)	cos(alfa)	0	-sin(theta)	0	cos(theta)	0		
	0	0	0	1	0	0	0	1		
2	1	0	0	0	cos(theta)	0	sin(theta)	0	0	-90
	0	cos(alfa)	sin(alfa)	0	0	1	0	0		
	0	-sin(alfa)	cos(alfa)	0	-sin(theta)	0	cos(theta)	0		
	0	0	0	1	-.85	0.01	1.5	1		
3	1	0	0	0	cos(theta)	0	sin(theta)	0	0	175
	0	cos(alfa)	sin(alfa)	0	0	1	0	0		
	0	-sin(alfa)	cos(alfa)	0	-sin(theta)	0	cos(theta)	0		
	0	0	0	1	.725	0.01	2.2	1		
4	1	0	0	0	cos(theta)	0	sin(theta)	0	0	90
	0	cos(alfa)	sin(alfa)	0	0	1	0	0		
	0	-sin(alfa)	cos(alfa)	0	-sin(theta)	0	cos(theta)	0		
	0	0	0	1	1.47	0	0.85	1		

Realizado por: Castro, Gabriel; 2023

Cada una de estas matrices ajusta las nubes de cada vista del objeto para convertirla en una sola. La orientación, ubicación es ajustada mediante estas matrices. El arreglo de cámaras está configurado para las dimensiones del diseño propuesto, pero se puede alterar las matrices para otro arreglo con otras dimensiones. La matriz A presenta la variación del objeto con respecto al eje X, y la matriz B presenta la variación del objeto con respecto al eje Y.

3.4.7 Métodos de evaluación del sistema de reconstrucción.

Para la evaluación del sistema de reconstrucción es necesario la búsqueda de parámetros o indicadores de contraste que comprueben el rendimiento del sistema. (Allard y Lavoie, 2014, p. 8) plantean 5 objetivos principales o indicadores para tres tipos de métodos de posicionamiento de escáneres, de los cuales plantean velocidad, simplicidad, flexibilidad, confiabilidad y precisión. En la tabla 8-3 se muestran dichos comparadores, la tabla presenta de manera resumida las principales ventajas y limitaciones de cada método, el sistema de reconstrucción está basado en el método de características naturales, aunque la unión de nubes de puntos se realizó de manera manual. Partiendo de esto se tomó como indicador del rendimiento del sistema a la velocidad de respuesta, la confiabilidad y la precisión que el sistema de reconstrucción presenta.

Tabla 8-3: Indicadores de desempeño de escáneres 3D.

Objetivo Principal	Métodos de Posicionamiento	Limitaciones Principales
Velocidad, Simplicidad	Geometría (Características Naturales)	Precisión, Características y Desempeño dependen del Objeto.
Flexibilidad	Geometría + Objetivos (Híbrido)	Comprometido con la exactitud
Confiabilidad, Precisión	Objetivos	Objetivos fijos en el objeto

Fuente: (Allard y Lavoie 2014, p. 8)

Realizado por: Castro, Gabriel; 2023

Cada parámetro o indicador se define como:

- **Velocidad:** Tiempo de ejecución que tarda el programa de reconstrucción en procesar las imágenes y generar un modelo digitalizado del objeto. Valor medido en segundos.
- **Precisión:** Variación existente entre distintos datos tomados del mismo objeto digitalizado. La medición se realizó en milímetros.

- **Confiabilidad:** Variación existente entre los valores medidos del objeto digitalizado y del objeto real, estos valores fueron tomados en milímetros.

Cada uno de estos parámetros fueron analizados tomando 30 muestras, con el fin de medir el tiempo en que se demora en reconstruir un mismo objeto, estos datos son mostrados en el capítulo de resultados como un indicador del rendimiento del sistema.

3.5 Mantenimiento: Entrega del sistema de reconstrucción.

Este apartado define la entrega del sistema de reconstrucción, sus resultados y limitaciones como parte de la metodología *waterfall*. Estos datos corresponden al capítulo de resultados por lo que estos se expondrán en el capítulo siguiente, en la ilustración 30-3 se observa el arreglo de cámaras del sistema completo.



Ilustración 30-3: Sistema de reconstrucción: arreglo de cámaras.

Realizado por: Castro, Gabriel; 2023

CAPÍTULO IV

4. MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS

4.1 Evaluación del sistema de reconstrucción.

Para la evaluación del sistema mediante el análisis de indicadores, se tomó como referencia la nube de puntos de la ilustración 1-4, que corresponde a un par de cajas apiladas, cuya geometría es conocida.



Ilustración 1-4: Imagen de tres cajas apiladas y su nube de puntos.

Realizado por: Castro, Gabriel; 2023

4.1.1 Velocidad del sistema.

Con el fin de medir el rendimiento mediante la determinación de la velocidad que el sistema se toma en realizar todos los procesos, desde la toma de datos hasta la reconstrucción, se hubo planteado la toma de tiempo de ejecución para tres de los procesos principales del algoritmo: la rectificación de imágenes, procesamiento de imágenes y unión de nube de puntos. Cada dato fue obtenido mediante las funciones de cronometro de *Matlab* con nombre *tic* y *toc*.

4.1.1.1 Tiempo de Rectificación de imágenes.

El diagrama de flujo del sistema comienza con la adquisición, la calibración y la rectificación de imágenes estéreo, estos procesos son de vital importancia y se han discutido algunas razones de su criticidad, debido a que sirve como base para el buen funcionamiento de los procesos

siguientes. Se midieron los tiempos en segundos en que se demoran en ejecutar todos estos procesos como si fuera un solo bloque. En la tabla 1-4 se muestran los datos obtenidos del tiempo de los procesos de rectificación de imágenes.

Tabla 1-4: Tiempo de rectificación de imágenes.

Prueba	Tiempo rectificación (Seg)
1	5.526398
2	5.77437
3	5.206134
4	5.762547
5	5.39715
6	5.699596
7	5.862359
8	5.653544
9	5.371135
10	5.225816
11	5.164164
12	5.277801
13	5.468559
14	5.617935
15	5.528962
16	5.287473
17	5.242821
18	5.193757
19	5.004409
20	5.057853
21	5.104595
22	5.742781
23	5.672124
24	5.60296
25	5.500067
26	5.91062
27	5.129649
28	4.993725
29	5.077579
30	4.957686
Promedio	5.400485633

Realizado por: Castro, Gabriel; 2023

De la tabla anterior se tiene un promedio de tiempo de 5.4 segundos, considerando los procesos de lectura de imágenes, almacenamiento de imágenes en video y rectificación de las imágenes por medio de parámetros de calibración.

4.1.1.2 Tiempo de Procesamiento de imágenes.

De igual manera que el proceso anterior, se calculó el tiempo que le toma al algoritmo de reconstrucción tomar esas imágenes rectificadas, segmentar y orientar cada nube de puntos y, finalmente generar un modelo tridimensional de estas. En la tabla 2-4 se muestran los datos de tiempo de procesamiento de imágenes.

Tabla 2-4: Tiempo de procesamiento de imágenes.

Prueba	Tiempo de procesamiento(seg)
1	5.914695
2	5.321274
3	5.38252
4	5.471255
5	5.559801
6	5.129725
7	5.862359
8	5.323002
9	5.460568
10	5.340865
11	5.449602
12	5.433398
13	5.433109
14	5.6092
15	5.5834
16	5.864588
17	5.243158
18	5.436262
19	5.004409
20	5.606702
21	5.595433
22	5.580899
23	5.38946
24	5.322303
25	5.868681
26	5.631971
27	5.413957
28	5.563656
29	5.600004
30	5.624585
Promedio	5.5006947

Realizado por: Castro, Gabriel; 2023

El tiempo promedio en realizar la reconstrucción parcial de las caras del objeto es de 5.50 segundos, cuya medida es muy parecida a la rectificación de las imágenes.

4.1.1.3 Tiempo de Unión y costura de nube de puntos.

La unión y costura conlleva el movimiento de las nubes, de tal manera que estas queden alineadas para formar un objeto completo. En la tabla 3-4 se muestran estos datos.

Tabla 3-4: Tiempo de unión de nubes de puntos.

Prueba	Tiempo de costura (seg)
1	2.148618
2	1.876799
3	1.652166
4	1.683862
5	1.717918
6	1.766824
7	2.057177
8	2.427553
9	1.815602
10	2.109333
11	1.964525
12	2.203947
13	1.717518
14	2.040669
15	2.465617
16	2.375335
17	2.369195
18	2.520406
19	2.100959
20	1.81559
21	1.957893
22	1.942797
23	1.936897
24	1.600406
25	1.581889
26	1.63822
27	1.647157
28	1.564416
29	1.572165
30	2.011353
Promedio	1.9427602

Realizado por: Castro, Gabriel; 2023

El tiempo promedio de ejecución del algoritmo para unir las nubes de puntos es de 1.94 segundos, lo que resulta en el proceso más rápido debido a que todo el proceso es manual, lo que quiere decir que no calcula la posición de la nube siguiente respecto a la anterior.

4.1.1.4 Tiempo total medio del sistema de reconstrucción.

En la tabla 4-4 se realizó una prueba de normalidad mediante la prueba de Shapiro-Wilk, en la cual se observa que todas las medidas de tiempo siguen una distribución normal.

Tabla 4-4: Prueba de normalidad de Shapiro-Wilk para los tiempos de ejecución del algoritmo.

	<i>Tiempo rectificación (Seg)</i>	<i>Tiempo de procesamiento (Seg)</i>	<i>Tiempo de costura (Seg)</i>
S	0.956434384	0.959474473	0.936031983
p-value	0.234176661	0.282548866	0.064127887
Alpha	0.05	0.05	0.05
normal	yes	yes	yes

Realizado por: Castro, Gabriel; 2023

Para obtener el tiempo total estimado de la velocidad del sistema de reconstrucción desde la lectura de imágenes hasta la unión de nubes de puntos, se realizó la suma aritmética de todos los tiempos debido a que todas las muestras de tiempo siguen una distribución normal. En las ecuaciones 4-1 se observan estas medidas.

$$tiempo_{total} = t_{rectificacion} + t_{procesamiento} + t_{costura} \quad (4.1)$$

$$tiempo_{total} = 5.400485633 + 5.5006947 + 1.9427602 \quad (4.1.1)$$

$$tiempo_{total} = 12.84394053 \quad (4.1.2)$$

Se obtiene una estimación total de tiempo de 12,84 segundos, lo que toma el algoritmo en desarrollar cada proceso. Este indicador puede sugerir aplicaciones para el sistema que necesiten un rápido procesamiento de las imágenes, como por ejemplo reconstrucción en tiempo real.

4.1.2 Precisión del sistema.

La precisión del sistema plantea mostrar los resultados del sistema de reconstrucción en comparación con COLMAP. Este último hace uso de Fotogrametría vista de manera general en el capítulo dos. Como medida de comparación entre los dos métodos, se utilizó la longitud de uno de los lados del objeto de la ilustración 2-4, cuya dimensión real es de 19,5 cm. Como indicador

de la dispersión de los datos entre los dos métodos se utilizó la desviación estándar que establece cuan alejadas están las medidas de su valor medio.

4.1.2.1 Prueba T-Student para medias de dos muestras emparejadas entre medidas obtenidas por este proyecto y el programa COLMAP.

Se tomaron las muestras mediante la herramienta *measuring* de *MeshLab*, tanto COLMAP como el sistema de reconstrucción propuesto entregan nubes de puntos del objeto, lo que permite importar a *MeshLab* y trabajar con sus datos tridimensionales. La tabla 5-4 muestra los datos obtenidos para los dos métodos.

Tabla 5-4: Medidas en centímetros del objeto mediante el sistema propuesto y COLMAP.

N.º Prueba	Sistema Propuesto (cm)	COLMAP (cm)
1	19.5499	19.550266
2	19.70093	19.568886
3	19.7289	19.637812
4	19.6571	19.766804
5	19.9446	19.495119
6	19.8304	19.455891
7	19.8318	19.483997
8	19.3921	19.666887
9	19.4839	19.728706
10	19.4747	19.512014
11	19.5069	19.356665
12	19.51557	19.831213
13	19.5894	19.641481
14	19.3038	19.580454
15	19.5426	19.592411
16	19.3779	19.641422
17	19.4137	19.503412
18	19.75	19.493107
19	19.6889	19.456238
20	19.6776	19.418285
21	19.5498	19.365353
22	20.2073	19.475187
23	19.2253	19.540047
24	19.1263	19.600014
25	19.5097	19.428385
26	19.6121	19.535519
27	19.6024	19.537991

28	19.1243	19.447212
29	19.0661	19.634918
30	18.9654	19.61463
Media	19.53164667	19.55201087
Desviación estándar	0.267572763	0.112198342

Realizado por: Castro, Gabriel; 2023

Con el fin de encontrar la diferencia o igualdad entre estos dos grupos de datos, se realizó una prueba T-Student para medias emparejadas. Con un valor de alfa del 5% de riesgo de que las medias de ambos grupos sean distintos. Se establece las hipótesis nula y alternativa de la prueba pareada como:

1. **Ho:** $MC - MS = 0$ (No existe diferencia significativa entre las medias del sistema propuesto y el software de reconstrucción COLMPAP)
2. **Ha:** $MS < MC$ (La media de los datos obtenidos por COLMAP es mayor que la media de los datos obtenidos por el sistema propuesto).

Antes de realizar la prueba T, es necesario satisfacer dos requisitos como los son: la normalidad de los datos, o lo que es lo mismo verificar si estos se ajustan a una distribución normal y, la homocedasticidad que establece la igualdad de varianzas entres estos dos grupos de datos.

Por medio de la aplicación Excel, se realizaron las pruebas de *Shapiro-Wilk* y *d'Agostino-Pearson* con el fin de verificar si los datos siguen una distribución normal. En la tabla 6-4 se muestran los resultados de estas pruebas de normalidad.

Tabla 6-4: Prueba de distribución normal de las dos muestras pareadas.

Shapiro-Wilk Test			d'Agostino-Pearson		
	Sistema Propuesto (cm)	COLMAP (cm)		Sistema Propuesto (cm)	COLMAP (cm)
W-stat	0.976659122	0.976046601	DA-stat	0.820123861	1.71300794
p-value	0.731492582	0.713545151	p-value	0.663609151	0.42464406
alpha	0.05	0.05	alpha	0.05	0.05
normal	yes	yes	normal	Yes	yes

Realizado por: Castro, Gabriel; 2023

Cada una de las muestras presentan una distribución normal cuyo p-value es mayor que Alpha cumpliendo con la hipótesis de normalidad. Mediante cada una de las pruebas las muestras son

normales. Para satisfacer el segundo requisito de verificar la igualdad de las varianzas, se realizó una prueba *Fisher* para varianzas de dos muestras, esta prueba permite determinar si la varianzas de las dos muestras es igual si la hipótesis nula $H_0: p\text{-value} > 0.5$, o diferentes el caso contrario. En la tabla 7-4 se observa esta prueba.

Tabla 7-4: Prueba F para varianzas de dos muestras.

	Sistema Propuesto (cm)	COLMAP (cm)
Media	19.53164667	19.55201087
Varianza	0.071595183	0.012588468
Observaciones	30	30
Grados de libertad	29	29
F	5.687362711	
P(F<=f) una cola	5.62941E-06	
Valor crítico para F (una cola)	1.860811435	

Realizado por: Castro, Gabriel; 2023

El valor de la probabilidad es menor que 0.5 lo que establece que las varianzas no son iguales. Como tener varianzas iguales no es requisito para poder realizar una prueba T-Student pareada, se realizó la prueba cumpliendo los requisitos necesarios, como normalidad de las muestras, independencia de muestras y que cada muestra sea obtenida de un mismo sujeto. La tabla 8-4 muestra el resultado de la prueba T.

Tabla 8-4: Prueba T-Student para dos muestras pareadas.

	Sistema Propuesto (cm)	COLMAP (cm)
Media	19.53164667	19.55201087
Varianza	0.071595183	0.012588468
Observaciones	30	30
Coefficiente de correlación de Pearson	-0.23679452	
Diferencia hipotética de las medias	0	
Grados de libertad	29	
Estadístico t	-0.355571877	
P(T<=t) una cola	0.362367318	
Valor crítico de t (una cola)	1.699127027	
P(T<=t) dos colas	0.724734636	
Valor crítico de t (dos colas)	2.045229642	

Realizado por: Castro, Gabriel; 2023

El $P(t \leq t)$ es mayor a 0.5, lo que establece que la hipótesis nula es verdadera, por lo tanto, no existe diferencia entre las medias de las dos muestras obtenidas por el sistema de reconstrucción propuesto y el software de reconstrucción basado en SFM, COLMAP.

Por último, en la ilustración 2-4 se observa la dispersión de los datos. Se presenta también la desviación estándar para cada uno de los métodos de reconstrucción. Los datos obtenidos por medio del sistema de reconstrucción propuesto dan como resultado una desviación estándar de 0.267572763.

Los datos obtenidos por medio del sistema de reconstrucción propuesto dan como resultado una desviación estándar de 0.112198342, lo que establece que el software COLMAP presenta menos dispersión media de datos.

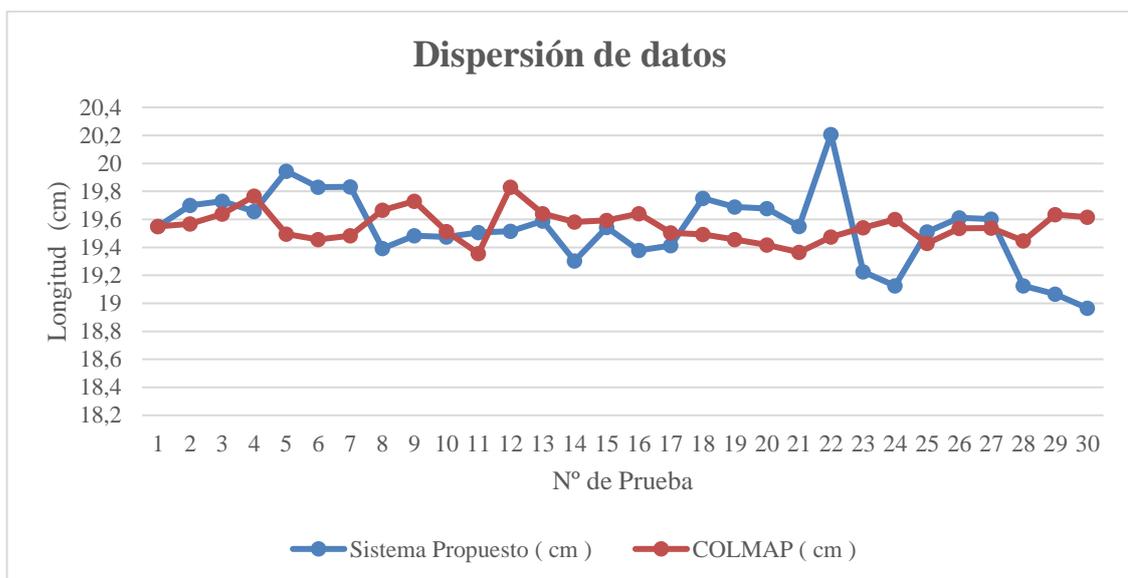


Ilustración 2-4: Dispersión de datos del sistema propuesto y COLMAP.

Realizado por: Castro, Gabriel; 2023

Por medio de las pruebas realizadas, los datos adquiridos por el sistema de reconstrucción volumétrica virtual de objetos y los datos adquiridos por el software de reconstrucción COLMAP, en específico la prueba T-Student dieron a conocer que los datos son idénticos. Pero sin embargo la dispersión de los datos del sistema propuesto es más grande con una desviación estándar de 0.267 en comparación de 0.11 de COLMAP. También se debe tener en cuenta que existen errores humanos de medición para la obtención de ambas muestras.

4.1.3 Mediciones de exactitud.

Con el fin de verificar la exactitud del sistema de reconstrucción, se planteó la realización de una prueba Z con el fin de saber si existe diferencia de una muestra de mediciones obtenidas mediante el sistema propuesto en comparación con un único valor, es a saber la medición real de unos de los lados del objeto, cuyo valor es de 19,5 cm. Con un nivel de significancia del 5%. Se establece las hipótesis nula y alternativa de la prueba Z como:

1. **H₀**: MS = 19.5 cm (No existe diferencia significativa entre la media del sistema propuesto y el valor real)
2. **H_a**: MS ≠ 19,5 cm (Existe diferencia significativa entre la media del sistema propuesto y el valor real).

Los datos de la muestra del sistema propuesto siguen una distribución normal vista en el apartado anterior, por lo que es factible comparar mediante la media de la muestra, en la tabla 9-4 se observa los resultados de la prueba Z.

Tabla 9-4: Prueba Z de una muestra.

Prueba Z de una muestra	Sistema Propuesto (cm)
Media	19.53164667
Varianza (conocida)	0.071595183
Observaciones	30
Media hipotética	19.5
z	0.646455602
P(Z≤z) una cola	0.258992171
Valor crítico de z (una cola)	1.644853627
P(Z≤z) de dos colas	0.517984341
Valor crítico de z (dos colas)	1.959963985

Realizado por: Castro, Gabriel; 2023

El valor de P(Z≤z) es mayor a 0.05, estableciendo a la hipótesis nula como verdadera, que los datos de la muestra obtenidos del sistema de reconstrucción son iguales o aproximados al valor real.

4.1.3.1 Error absoluto entre las medidas obtenidas mediante el sistema de reconstrucción y el dato real.

Para encontrar el error que presenta el sistema de reconstrucción, se realizó la búsqueda del error absoluto entre el sistema de reconstrucción propuesto y la medida real del objeto. En la tabla 10-4 se presenta este error.

Tabla 10-4: Error absoluto entre el sistema propuesto y medida real.

N.º Prueba	Error (mm)
1	0.499
2	2.0093
3	2.289
4	1.571
5	4.446
6	3.304
7	3.318
8	1.079
9	0.161
10	0.253
11	0.069
12	0.1557
13	0.894
14	1.962
15	0.426
16	1.221
17	0.863
18	2.5
19	1.889
20	1.776
21	0.498
22	7.073
23	2.747
24	3.737
25	0.097
26	1.121
27	1.024
28	3.757
29	4.339
30	5.346
Media	2.014133333
Desviación estándar	1.721710133

Realizado por: Castro, Gabriel; 2023

Analizando los datos se obtiene una media de error de 2 mm y una desviación estándar de 1,72 mm, con un error máximo de 7 mm y un mínimo de 0,069 mm.

4.2 Discusión de los parámetros de comparación.

Según los parámetros de comparación y los datos obtenidos se concluye que:

- El sistema de reconstrucción tiene una media de ejecución de 12.84 segundos lo que lo convierte en un sistema de reconstrucción de rápido tiempo de respuesta.
- El error medio del sistema es de 2 mm, aplicable a sistemas que necesiten un margen de error máximo de hasta 2mm.
- El sistema presento resultados aproximados a los resultados obtenidos por el sistema de reconstrucción COLMAP basada en la técnica SFM, que mediante la prueba T, estableció que estos datos son aproximadamente iguales.

4.3 Objetos digitalizados.

El algoritmo del sistema de reconstrucción entrega una nube de puntos densa de los objetos digitalizados, tal como se observa en la ilustración 3-4, estas nubes presentan ruido que ocasionan una ineficiente conversión a malla.

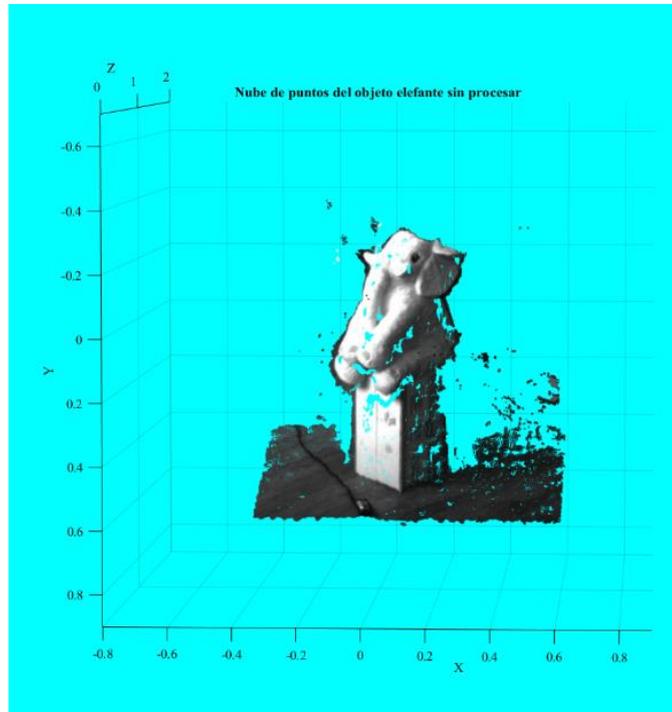


Ilustración 3-4: Nube de puntos de vista lateral de un elefante.

Realizado por: Castro, Gabriel; 2023

Con el fin de eliminar este ruido y computar la malla correspondiente de las nubes de puntos se utilizó el programa Meshlab, este programa permite la conversión de puntos con coordenadas tridimensionales a una superficie continua. El sistema de reconstrucción permite la exportación de las nubes de puntos en formato PLY, como se observa en la ilustración 4-4, que es un formato de almacenamiento de datos tridimensionales, aunque también tiene la posibilidad de exportarse en formato *ascii*.

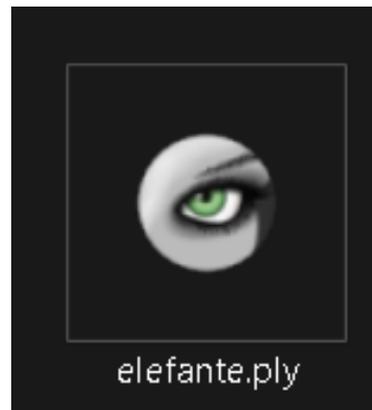


Ilustración 4-4: Archivo PLY.

Realizado por: Castro, Gabriel; 2023

Meshlab permite importante directamente el archivo PLY, en la ilustración 5-4 se observa el área de trabajo del software de edición de mallas.

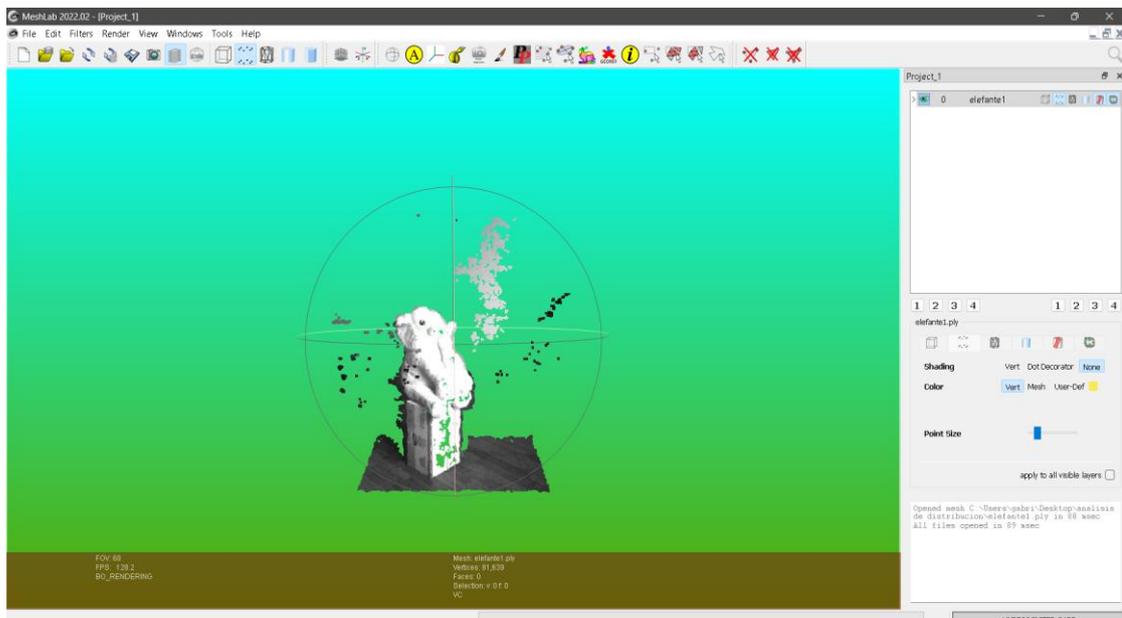


Ilustración 5-4: Ventana de edición de Meshlab.

Realizado por: Castro, Gabriel; 2023

El software permite el recorte de puntos mediante algunas herramientas de selección cuadrada o selección por puntos. Esto es de utilidad debido a que algunas partes del objeto que no son de interés, pero que son de ayuda en el proceso de unión de nubes salen en archivo PLY del sistema de reconstrucción, y que ahora es necesario eliminar. En la ilustración 6-4 se observa el filtrado de puntos para un elefante, vista derecha sin filtrar y vista izquierda filtrada.

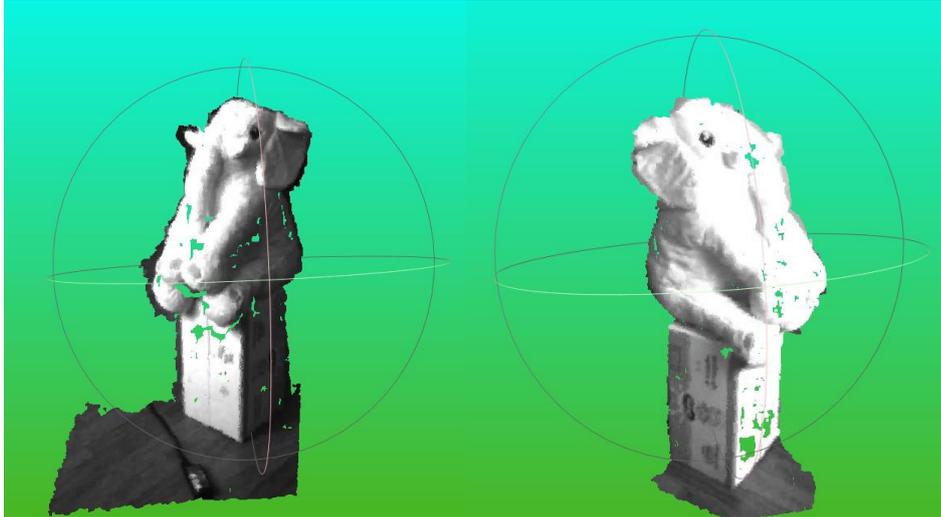


Ilustración 6-4: Nube de puntos de elefante filtrado el ruido.

Realizado por: Castro, Gabriel; 2023

La conversión de una nube de puntos a una malla texturizada puede realizarse mediante la herramienta *compute normals for point set* y la herramienta *Surface Reconstruction: screened poisson*, en la ilustración 7-4 se muestra el objeto convertido a malla.

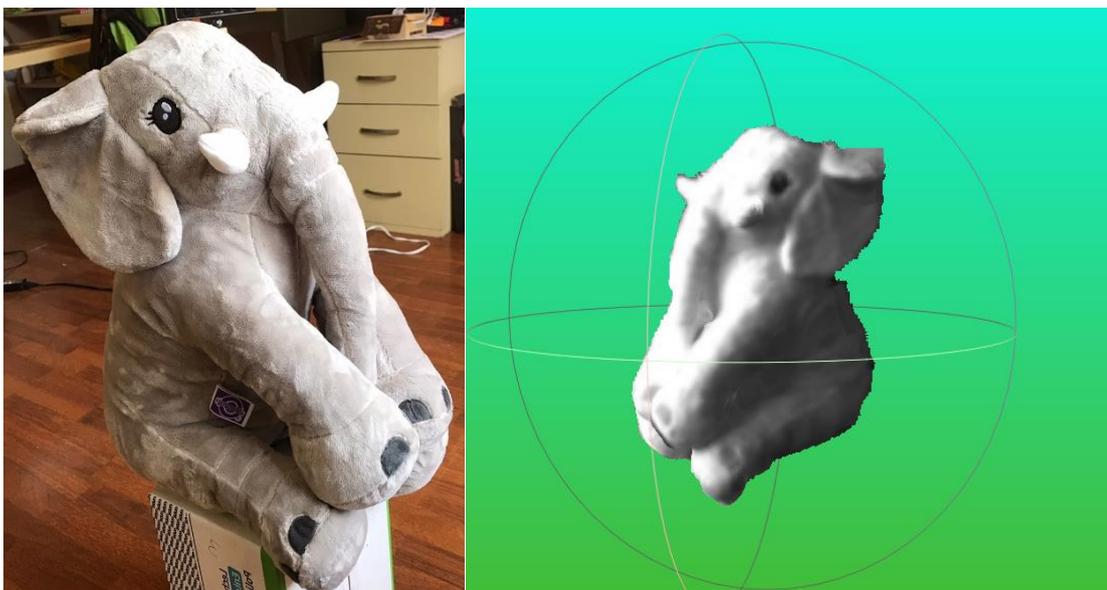


Ilustración 7-4: Malla del objeto elefante.

Realizado por: Castro, Gabriel; 2023

Meshlab permite la exportación de la malla del objeto como archivo de formato polígono (PLY) o de impresión de estereolitografía (STL). En la ilustración 8-4 se observa un violín con vista frontal.

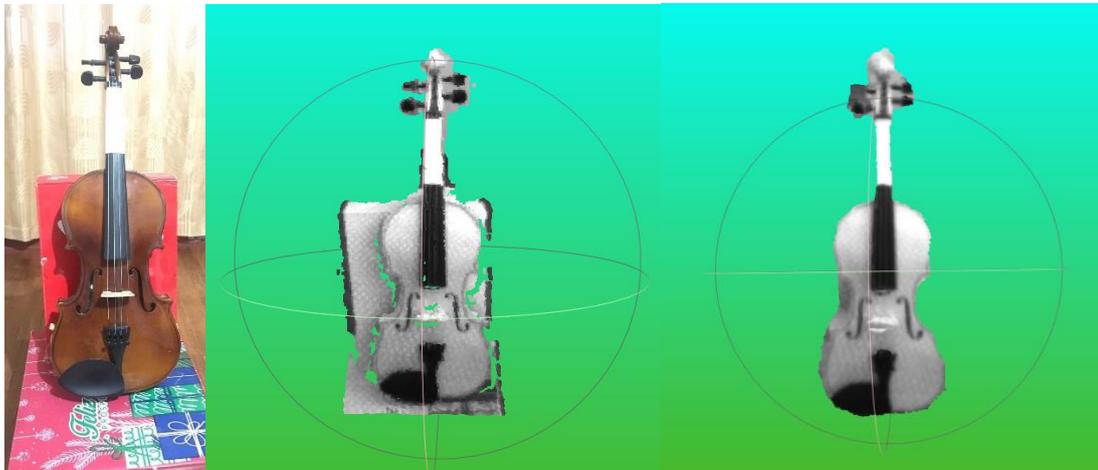


Ilustración 8-4: Reconstrucción de violín, imagen, nube de puntos y malla del objeto respectivamente.

Realizado por: Castro, Gabriel; 2023

Se observa que, en las partes pequeñas de la nube de puntos del violín, como en el clavijero se generan puntos uniformes que no corresponden a la forma original, Meshlab por lo tanto genera la malla de acuerdo con la nube, pero esa malla es errónea comparado con el objeto original. En la ilustración 9-4 se observa la figura de una rana, la nube de puntos y su malla.

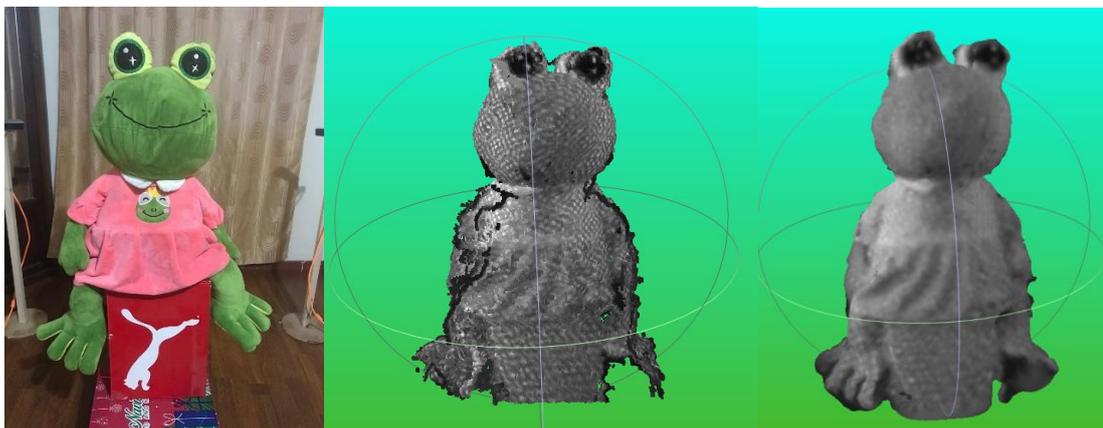


Ilustración 9-4: Reconstrucción de rana: imagen, nube de puntos y malla del objeto respectivamente.

Realizado por: Castro, Gabriel; 2023

Se observa la gran cantidad de ruido en la nube de puntos, esto ocasiona incongruencias en la conversión a malla. También al igual que el violín las partes pequeñas como en el pie de la rana

la nube y malla es uniforme, mientras que en el objeto real hay una separación. Existen también errores de reconstrucción dado por la técnica estero en especial del apartado de correspondencia de puntos. En la ilustración 10-4 se observa la reconstrucción de tres cajitas de diferente tamaño y geometría rectangular.

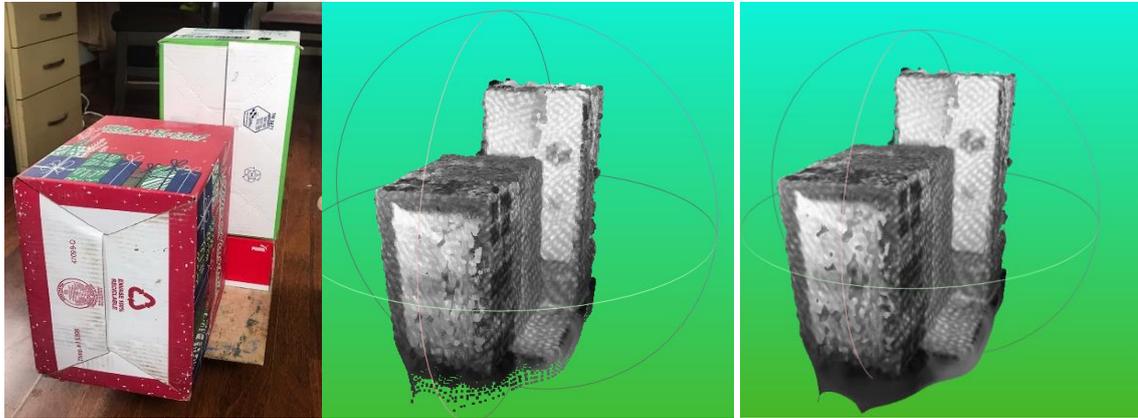


Ilustración 10-4: Reconstrucción de cajas apiladas: imagen, nube de puntos y malla del objeto respectivamente.

Realizado por: Castro, Gabriel; 2023

Por último, se muestra la reconstrucción de una mochila, el color de esta es más oscura, esto ocasiona problemas de correspondencia entre puntos. Los colores oscuros y las superficies brillantes refractan la luz no permitiendo una uniformidad de colores en las dos imágenes estéreo, ocasionando huecos e irregularidades en la reconstrucción. En la ilustración 11-4 se muestra la mochila, la nube de puntos y la malla vista izquierda y derecha respectivamente.

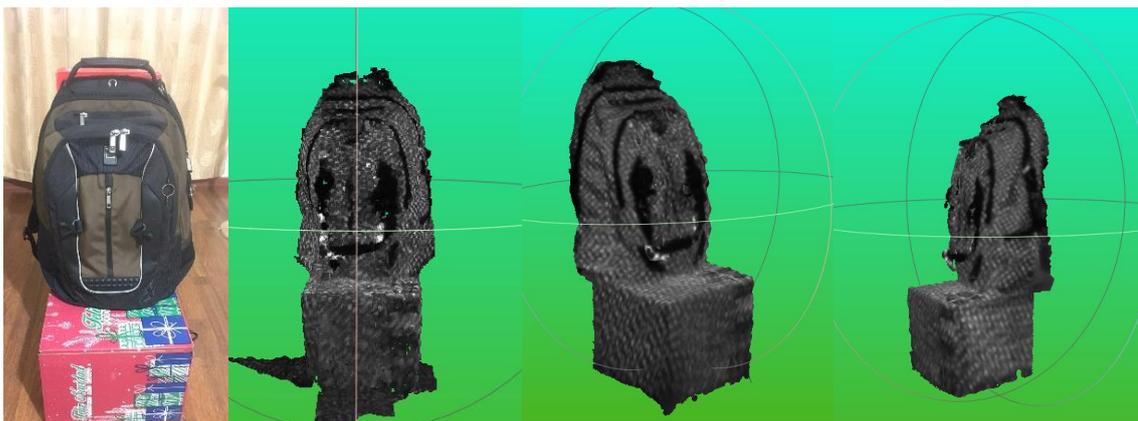


Ilustración 11-4: Reconstrucción de mochila: imagen, nube de puntos y malla vista izquierda y derecha respectivamente.

Realizado por: Castro, Gabriel; 2023

4.4 Limitación del sistema.

El sistema de reconstrucción es más eficiente para objetos con un diámetro mayor a 20 cm, en las partes más pequeñas de los objetos, el sistema no detecta la forma geométrica propia del objeto, por lo tanto, aquellas partes pequeñas aparecen reconstruidas de manera uniforme como en el caso del pie de la rana o el clavijero del violín. El sistema es sensible a la no uniformidad de la luz en el objeto, el algoritmo de correspondencia de píxeles necesita que ambas imágenes estéreo tengan los mismos niveles de luz para poder encontrar el mapa de disparidad, por lo que requiere de un sistema de iluminación uniforme. Los objetos con una superficie lisa y brillante reflejan la luz y generan no uniformidad en las imágenes. La textura de los objetos digitalizados no es uniforme, en la ilustración 12-4 se observa la textura del objeto de la ilustración 10-4, esto se debe a que todos los puntos obtenidos no se encuentran en el mismo valor de profundidad para objetos planos, esto puede ser solucionado mediante el procesamiento de nube de puntos y de mallas texturizadas.

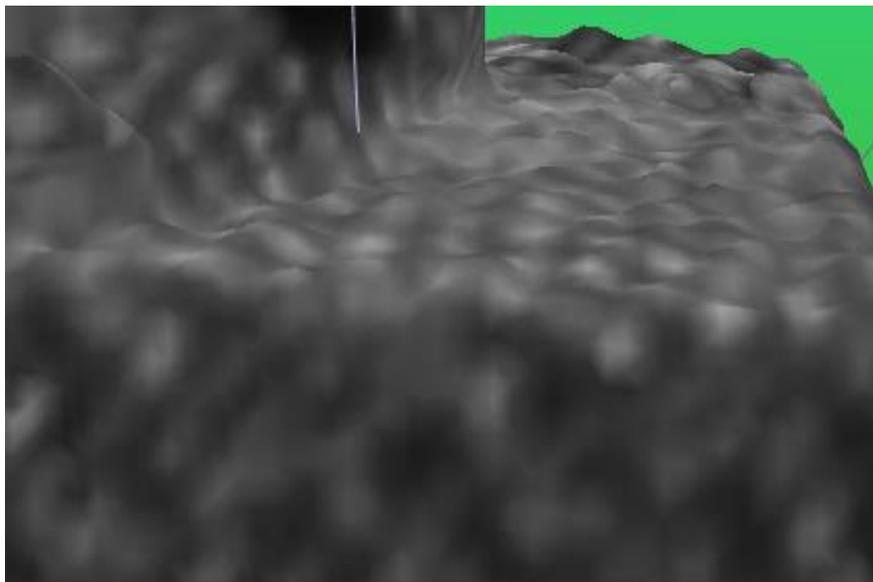


Ilustración 12-4: Textura de objeto digital.

Realizado por: Castro, Gabriel; 2023

4.5 Costos del sistema de reconstrucción.

En la tabla 11-4 se muestran los materiales tecnológicos que forman parte del sistema de reconstrucción, lo más caro del sistema son las cámaras estéreo de la marca MYNT. El computador, al tratarse de cuatro vistas no es necesario de una maquina con una capacidad computacional superior.

Tabla 11-4: Costos de implantación del sistema de reconstrucción.

MATERIAL	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL	Financiamiento
Computador Core i3 8Gb Ram USB 3.0	1	\$400	\$400	TESISTA
cámaras	4	\$140	\$560	ESPOCH
Licencia estudiantil Matlab	1	\$61	\$61	TESISTA
Total			\$1021	

Realizado por: Castro, Gabriel; 2023.

El costo de implementación del sistema de reconstrucción es de 1021 dólares, con un valor de 560 dólares en cámaras que corresponden a la parte de extracción de las características geométricas del objeto.

CONCLUSIONES

La revisión de las técnicas, métodos y tecnologías de reconstrucción de objetos dio paso a la elaboración de un sistema de reconstrucción de objetos virtuales, tomando como base la técnica de visión estéreo.

El sistema se diseñó mediante un arreglo de cuatro vistas estéreo, cada una localizada en la esquina de un arreglo cuadrado, cada vista estéreo mediante la obtención de mapas de profundidad individuales logró la captura parcial de objeto de interés, el algoritmo desarrollado tomó cada vista parcial para convertirla en una sola vista tridimensional completa del objeto.

Se analizaron las imágenes estereoscópicas para configurar los parámetros de operación de las cámaras estéreo: la ganancia, el brillo y el contraste por medio de la variación de cada una de sus magnitudes, con el fin de obtener imágenes que presenten un resultado óptimo de reconstrucción. Se demostró que cada parámetro en su valor máximo dio como resultado nubes de puntos fiables del objeto.

Las imágenes estéreo presentaron distorsiones por fenómenos físicos como errores en la fabricación de las lentes y la perpendicularidad de los sensores con el plano de imagen, esto ocasionó la imposibilidad de un ajuste en los algoritmos que relacionan los píxeles de las imágenes para encontrar la profundidad, fue necesario la calibración de estas cámaras, la calibración recortó, ajustó y rectificó las imágenes mediante el procesamiento de estas, con el objeto de realizar la coincidencia entre el par estéreo y la obtención de un mapa de profundidad aceptable para reconstruir los objetos.

El sistema fue implementado siguiendo un arreglo de cámaras sencillo cuadrado, cada cámara espaciada de la anterior por la misma distancia, mientras más espaciamiento entre cámaras hubo los objetos se reconstruyeron mejor, pero la densidad o la cantidad de puntos del objeto disminuyó debido a la cantidad de píxeles que el objeto ocupa en el marco de imagen, se implementó el sistema a una distancia mínima de 70 cm entre el objeto y cada cámara, para satisfacer esa relación entre densidad de puntos y calidad de reconstrucción.

Mediante la evaluación se logró obtener medidas similares a los objetos reales, estas medidas diferían en pequeñas longitudes en orden de los dos milímetros de media debido al ajuste manual de las nubes, resultando en un sistema útil para aplicaciones con este rango de error.

RECOMENDACIONES

Se recomienda la revisión del estado del arte sobre otras técnicas de reconstrucción pasivas y activas.

Es recomendable realizar el diseño de un sistema de adquisición de imágenes automático con una sola cámara estéreo y una plataforma de giro, para reducir el uso de varias cámaras. De modo que se pueda optimizar el cableado y obtener más vistas para una reconstrucción más densa.

Es recomendable evaluar los parámetros de cámara para cada entorno de reconstrucción, y objetos, debido a que a cada escenario de reconstrucción presentar diferentes características al escenario estudiado.

Se recomienda el uso de la técnica estéreo acompañada con otra técnica llamada luz estructura, esta técnica de luz consiste en un método activo, añadiendo un elemento como un láser que mida la distancia entre la cámara y los objetos, o también mediante la incorporación de un proyector que proyecte patrones deformables sobre el objeto, mediante el análisis de procesamiento de imágenes pueda encontrarse esta deformación y realizar la búsqueda de la profundidad mediante estéreo de manera más precisa. También el de realizar el procesamiento de imágenes en un entorno y lenguaje dedicado al desarrollo de objetos virtuales como Python y PCL.

La mejora del sistema se puede realizar añadiendo más cámaras de manera equidistante al objeto de manera estática, debido a que la reconstrucción del objeto está configurada de acuerdo con la posición de cada cámara respecto a la cámara anterior, el objeto de esta recomendación es tomar imágenes del objeto y en donde haya huecos, las imágenes continuas puedan mejorar esas limitaciones.

Es recomendable realizar una calibración aceptable, en algunas cámaras estéreo fenómenos de fabricación como la distorsión radial y tangencial son insignificantes, pero en las cámaras de la marca MYNT se aprecia una distorsión muy grande esto dificulta la reconstrucción.

BIBLIOGRAFÍA

ALLARD, P.H; & LAVOIE, J.A. “Differentiation of 3D scanners and their positioning method when applied to pipeline integrity”. [en línea], 2014, pp. 8. Disponible en: https://www.ndt.net/events/ECNDT2014/app/content/Paper/19_Allard.pdf.

ARRIAGA, J.A. Reconstrucción de Perfiles Tridimensionales de Objetos Transparentes [en línea] (Trabajo de titulación). (maestría) Instituto Nacional de Astrofísica, Óptica y Electrónica. Tonantzintla, Puebla. 2013. pp 2-3. [Consulta: 2023-01-15]. Disponible en: <https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/786/1/ArriagaHJA.pdf>.

BASULTO LANTSOVA, A. Emparejamiento de puntos en triangulación multicámara utilizando DIC [en línea]. (Trabajo de titulación). (Grado de Maestro en Ciencias en Ingeniería Electrónica) Tecnológico nacional de México. Celaya, Guanajuato, México.2020. pp. 113.[Consulta:2023-02-17]. Disponible en: https://www.researchgate.net/profile/Alejandro_Barranco_Gutierrez5/publication/349683067_Emparejamiento_de_puntos_en_triangulacion_multicamara_utilizando_DIC/links/603c5abe299bf1cc26fbd13b/Emparejamiento-de-puntos-en-triangulacion-multicamara-utilizando-DIC.pdf.

BIMND. *El Escáner Láser 3D Terrestre ¿Qué tipos hay?* [blog].2020 [Consulta: 15 enero 2023]. Disponible en: <https://www.bimnd.es/el-escaner-laser-3d-terrestre-que-tipos-hay/>.

BITFAB. *Tipos de escaneado 3D a examen: los comparamos.* [blog]. 2020. [Consulta: 15 enero 2023]. Disponible en: <https://bitfab.io/es/blog/tipos-escaneado-3d/>.

CAPPELLARO, R. Comparison of Stereo Visual Inertial Odometry Algorithms for Unmanned Ground Vehicles [en línea] (Trabajo de titulación). (Master of Science Degree in MECHATRONIC ENGINEERING) Politecnico di Torino. Italia. 2019. pp. 2-14. [Consulta: 12 junio 2022]. Disponible en: <https://webthesis.biblio.polito.it/11648/1/tesi.pdf>.

CÁRDENAS QUIROGA, E.A.; et al. “La estereoscopia, métodos y aplicaciones en diferentes áreas del conocimiento”. Revista Científica General José María Córdova [en línea], 2015, 13(16), pp. 201-219. [Consulta: 9 junio 2022]. ISSN 2500-7645, 1900-6586. DOI 10.21830/19006586.37. Disponible en: <https://www.revistacientificaesmic.com/index.php/esmic/article/view/37>.

CASTRO, F. Estereoscopia | IDIS. *IDIS* [en línea]. 1840. [Consulta: 7 marzo 2023]. Disponible en: <https://proyectoidis.org/estereoscopia/>.

CATALÀ SARMIENTO, C. Edición estereoscópica [en línea] (Trabajo de titulación). (Máster Universitario en Postproducción Digital-Màster Universitari en Postproducció Digital) Universitat Politècnica de València. Gandia. 2014. pp.13-15. [Consulta: 13 febrero 2023]. Disponible en: <https://riunet.upv.es/handle/10251/37172>.

CIGNONI, P., CALLIERI, M.; et al. “MeshLab: an Open-Source Mesh Processing Tool”. Eurographics Italian Chapter Conference [en línea], 2008, pp. 1. [Consulta: 16 febrero 2023].
ISSN 978-3-905673-68-5. DOI 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136. Disponible en: <https://diglib.eg.org/bitstream/handle/10.2312/LocalChapterEvents.ItalChap.ItalianChapConf2008.129-136/129-136.pdf?sequence=1&isAllowed=y>.

CONTRERAS, L. *Escáner láser 3D o escáner de luz estructurada: ¿cuál es mejor?* [blog]. 2019. [Consulta: 15 enero 2023]. Disponible en: <https://www.3dnatives.com/es/escaner-laser-3d-escaner-luz-estructurada-mejor-08082019/>.

CREAFORM, L. *¿Qué es una cmm? todo lo que necesita saber sobre las máquinas de medición por coordenadas y sus tipos.* [blog]. 2021. [Consulta: 15 enero 2023]. Disponible en: <https://www.creaform3d.com/blog/es/que-es-una-cmm-todo-lo-que-necesita-saber-sobre-las-maquinas-de-medicion-por-coordinadas-y-sus-tipos/>.

CUESTA, E., FERNÁNDEZ, P.; et al. “Aplicaciones Metrologicas de los láseres por triangulación”. Reed Business Information [en línea], 2014, pp. 59-73. [Consulta: 15 enero 2023]. ISSN 0210-685X. Disponible en: <http://hdl.handle.net/10651/31492>.

EISERT, P.; et al. Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views. *IEEE Transactions on Circuits and Systems for Video Technology* [en línea], 2000, 10(2), pp. 261-277. [Consulta: 7 marzo 2023]. ISSN 1558-2205. DOI 10.1109/76.825726. Disponible en: <https://ieeexplore.ieee.org/document/825726?arnumber=825726>.

EPELBUM, S. Historia de la Estereoscopia y sus Aplicaciones. Archivos de Oftalmología de Buenos Aires [en línea], 2010, 81(2), pp. 62-67. [Consulta: 7 marzo 2023]. Disponible en: <https://docplayer.es/7560209-Historia-de-la-estereoscopia-y-sus-aplicaciones.html>.

FERNANDEZ, R. *Calibración de la cámara OpenCV*. [blog]. 2018. [Consulta: 13 febrero 2023]. Disponible en: <https://unipython.com/calibracion-la-camara-opencv/>.

FURHT, B. *Encyclopedia of Multimedia: Block Matching* [en línea]. Boston, MA: Springer US, pp. 55-56, 2008. [Consulta: 26 julio 2022]. ISBN 978-0-387-78414-4. Disponible en: https://doi.org/10.1007/978-0-387-78414-4_132.

GOLUBEVA, S. *Qué es el escaneo 3D láser* [blog]. 2022. [Consulta: 16 enero 2023]. Disponible en: <https://www.artec3d.com/es/learning-center/laser-3d-scanning>.

HERNANDEZ, C.; & SCHMITT, F. Silhouette and stereo fusion for 3D object modeling. *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings*. [en línea]. 2003, Banff, AB, Canada: s.n., pp. 46-53. [Consulta: 7 marzo 2023]. DOI 10.1109/IM.2003.1240231. Disponible en: <https://ieeexplore.ieee.org/document/1240231>.

HERNÁNDEZ-DÍAZ, M.; et al. “Evaluación morfométrica del complejo estilohioideo mediante tomografía computarizada multicorte”. *Revista Científica Odontológica* [en línea], 2021, (Venezuela), 9(2), pp. 1-9. [Consulta: 16 enero 2023]. DOI 10.21142/2523-2754-0902-2021-056. Disponible en: <https://revistas.cientifica.edu.pe/index.php/odontologica/article/view/910/785>.

HIRSCHMULLER, H. “Stereo Processing by Semiglobal Matching and Mutual Information”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [en línea], 2008, 30(2), pp. 328-341. [Consulta: 15 febrero 2023]. ISSN 1939-3539. DOI 10.1109/TPAMI.2007.1166. Disponible en: https://elib.dlr.de/55367/1/Stereo_Processing-Hirschm%C3%BCller.pdf.

HORNBERG, A. *Handbook of Machine and Computer Vision: The Guide for Developers and Users* [en línea]. 2nd. ed. S.l.: Wiley. 2017. ISBN 978-3-527-41341-6. [Consulta: 15 enero 2023]. Disponible en: <https://elibro.net/es/ereader/epoch/177220?page=1190>.

IDS. *3D-Data with Stereo Vision - IDS Imaging Development Systems GmbH* [blog]. 2014. [Consulta: 13 febrero 2023]. Disponible en: <https://en.ids-imaging.com/whitepaper-details/whitepaper-depth-information-3d-images.html>.

JAKOB, W. *Calibration Best Practices-calib.io* [en línea]. 2018. [Consulta: 13 febrero 2023]. Disponible en: <https://calib.io/blogs/knowledge-base/calibration-best-practices>.

JÁUREGUI EL SO, I. Reconstrucción de objetos en 3D mediante un sistema estéreo binocular [en línea] (Trabajo de titulación). (Grado en Ingeniería Informática) Universidad Pública de

Navarra. 2018. [Consulta: 13 febrero 2023]. Disponible en: <https://academica-e.unavarra.es/xmlui/handle/2454/32849>.

JIMÉNEZ MEDINA, L.M. Algoritmo para reconstrucción 3D de objetos texturizados [en línea] (Trabajo de titulación). (Maestría) Centro de investigación científica y de educación superior de ensenada. Mexico. 2008. [Consulta: 15 febrero 2023]. Disponible en: <http://cicese.repositorioinstitucional.mx/jspui/handle/1007/335>.

JUAREZ, E. *¿Qué es la digitalización 3D?* [blog]. 2022. [Consulta: 15 enero 2023]. Disponible en: <https://www.juarezthompson.mx/post/que-es-la-digitalizacion-3d>.

KAPELLER, C. Evaluation of a 3D reconstruction system comprising multiple stereo cameras [en línea] (Trabajo de titulación). (Diplom-Ingenieur) Technische Universität Wien. Viena-Austria. 2018. pp. 19-20.[Consulta: 15 febrero 2023]. Disponible en: <https://repositum.tuwien.at/handle/20.500.12708/1882>.

KREON. *¿Qué es un brazo de medición? Kreon Technologies* [blog]. 2022. [Consulta: 15 enero 2023]. Disponible en: <https://kreon3d.com/es/que-es-un-brazo-de-medicion/>.

KRYPTON, S. *¿Qué es la cámara 3D? - Definición de Krypton Solid (Actualizado 2023) - Krypton Solid.* [blog]. 2021. [Consulta: 13 febrero 2023]. Disponible en: <https://kryptonsolid.com/que-es-la-camara-3d-definicion-de-krypton-solid/>.

LEE, P.-H., HUANG, J.-W. y LIN, H.-Y. 3D model reconstruction based on multiple view image capture. 2012 International Symposium on Intelligent Signal Processing and Communications Systems [en línea]. 2012, Tamsui, Taiwan: s.n., pp. 58-63. [Consulta: 7 marzo 2023]. DOI 10.1109/ISPACS.2012.6473453. Disponible en: <https://ieeexplore.ieee.org/document/6473453>.

LEÓN, M.Á. *Digitalización 3D. Interempresas* [blog]. 2003.[Consulta: 15 enero 2023]. Disponible en: <https://www.interempresas.net/Plastico/Articulos/8697-Digitalizacion-3D.html>.

LÓPEZ VALLES, J.M.; et al. Conceptos y Técnicas de Estereovisión por Computador. Revista Iberoamericana de Inteligencia Artificial [en línea], 2005, (España), 9(27), pp. 35-62. [Consulta: 15 enero 2023]. ISSN 1137-3601. Disponible en: <https://www.redalyc.org/pdf/925/92502703.pdf>.

MANTECA FERNÁNDEZ, F. Generación de modelos 3D mediante luz estructurada [en línea] (Trabajo de titulación). (Ingeniería de Tecnologías de Telecomunicación) UNIVERSIDAD DE

CANTABRIA. Espana. 2018. pp. 11-13. [Consulta: 16 enero 2023]. Disponible en: <https://repositorio.unican.es/xmlui/handle/10902/15136>.

MARTÍNEZ, M.M. Técnicas de visión estereoscópica para determinar la estructura tridimensional de la escena [en línea] (Trabajo de titulación). (Máster en Ingeniería Informática para la Industria) Universidad Complutense de Madrid. Espana. 2010. pp. 89.[Consulta: 17 febrero 2023]. Disponible en: https://eprints.ucm.es/id/eprint/11350/1/Técnicas_de_visión_estereoscópica_para_determinar_la_estructura_tridimensional_de_la_escena.pdf.

MATHWORKS. *Patrones de calibración.* [en línea]. 2022a. [Consulta: 28 junio 2022]. Disponible en: https://la-mathworks-com.translate.goog/help/vision/ug/calibration-patterns.html?_x_tr_sl=auto&_x_tr_tl=es&_x_tr_hl=es.

MATHWORKS. *Preparar cámara y capturar imágenes.* [blog]. 2022b. [Consulta: 29 junio 2022]. Disponible en: https://la-mathworks-com.translate.goog/help/vision/ug/prepare-camera-and-capture-images.html?_x_tr_sl=auto&_x_tr_tl=es&_x_tr_hl=es.

MATLAB. *What Is Camera Calibration?* - MATLAB & Simulink - MathWorks América Latina. [blog]. 2023. [Consulta: 13 febrero 2023]. Disponible en: <https://la.mathworks.com/help/vision/ug/camera-calibration.html>.

MORÓN A, C. Metodología de trabajo conjunto de escaneado e impresión 3d [en línea] (Trabajo de titulación). (Grado en Ingeniería en Tecnologías Industriales Especialidad en Ingeniería Mecánica) UNIVERSIDAD POLITÉCNICA DE MADRID. Espana. 2018. pp.13-19. [Consulta: 15 enero 2023]. Disponible en: https://oa.upm.es/50348/1/TFG_CARLOS_MORON_ALGUACIL.pdf.

MYNTAI. *Plataformas admitidas — documentación de MYNT® EYE S SDK 2.5.0.* [blog]. 2019a. [Consulta: 10 junio 2022]. Disponible en: https://mynt-eye-s-sdk-readthedocs-io.translate.goog/en/latest/src/sdk/support_platforms.html?_x_tr_sl=auto&_x_tr_tl=es&_x_tr_hl=es.

MYNTAI. *Product Description — MYNT® EYE S SDK 2.5.0 documentation.* [blog]. 2019b. [Consulta: 13 febrero 2023]. Disponible en: https://mynt-eye-s-sdk.readthedocs.io/_/downloads/en/2.3.7/pdf/.

OFTALVIST. *Visión estereoscópica o estereopsis ¿Qué es y cómo afecta?* [blog]. 2019. [Consulta: 13 febrero 2023]. Disponible en: <https://www.oftalvist.es/blog/que-es-la-vision-estereoscopica-o-estereopsis>.

PORTELLA COAGUILA, A.G. Diseño y construcción de un dispositivo para realizar reconstrucción tridimensional basado en un sistema estéreo de dos cámaras [en línea] (Trabajo de titulación). (Ingeniería) Universidad de San Agustín de Arequipa. Facultad de ingeniería de producción y servicios. Escuela Profesional de Ingeniería y Electrónica. Arequipa-Peru. 2015. [Consulta: 7 marzo 2023]. Disponible en: <http://repositorio.unsa.edu.pe/handle/UNSA/3316>.

RESTREPO, C.; et al. Tomografía computarizada de craneo. [en línea]. Radiología e Imágenes diagnósticas. Ecoe ediciones Project. 2019, pp. 175-185. [Consulta: 1 febrero 2023]. ISBN 978-958-8843-81-0. Disponible en: https://www.researchgate.net/publication/339461394_TOMOGRAFIA_COMPUTARIZADA_DE_CRANEO.

REVOPOINT 3D. *Comparing Three Prevalent 3D Imaging Technologies—ToF, Structured Light and Binocular Stereo Vision.* Revopoint 3D Technologies Inc. [blog]. 2019.[Consulta: 13 febrero 2023]. Disponible en: <https://www.revopoint3d.com/comparing-three-prevalent-3d-imaging-technologies-tof-structured-light-and-binocular-stereo-vision/>.

ROCCHINI, C.; et al. A low-cost 3D scanner based on structured light. Computer Graphics Forum [en línea], 2001, 20(3), pp. 299-308. [Consulta: 13 febrero 2023]. DOI 10.1111/1467-8659.00522. Disponible en: https://www.researchgate.net/profile/Roberto-Scopigno/publication/2377692_A_low_cost_3D_scanner_based_on_structured_light/links/5a492cb1458515f6b058bfd2/A-low-cost-3D-scanner-based-on-structured-light.pdf

RODRÍGUEZ, H. *¿Qué es OpenCV y para qué sirve?* [blog]. <https://www.crehana.com> [en línea]. 2021. [Consulta: 10 junio 2022]. Disponible en: <https://www.crehana.com/blog/desarrollo-web/que-es-opencv/>.

SAIDON, M.S.; et al. Developing Stereo Vision System for Object Recognition and Positioning of AMAD-R Mobile Robot. Proceedings of the International Conference on Man-Machine Systems [en línea], 2009. [Consulta: 16 junio 2023]. Disponible en: https://www.researchgate.net/publication/265746633_Developing_Stereo_Vision_System_for_Object_Recognition_and_Positioning_of_AMAD-R_Mobile_Robot.

SANTANDER, U. *Metodologías de desarrollo de software: ¿qué son?* [en línea].2020. [Consulta: 5 junio 2022]. Disponible en: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>.

SAQUIN, C.H. Implementación de un algoritmo para reconstrucción 3D de imágenes submarinas mediante un sistema de visión de bajo costo [en línea] (Trabajo de titulación). (Maestría) Centro de Ingeniería y Desarrollo Industrial. México. 2019. pp. 36-37. [Consulta: 17 febrero 2023]. Disponible en: <https://cidesi.repositorioinstitucional.mx/jspui/handle/1024/385>.

SCHONBERGER, J.L.; & FRAHM, J.M. Structure-from-Motion Revisited. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [en línea]. Las Vegas, NV, USA: IEEE, 2016, pp. 4104-4113. [Consulta: 16 febrero 2023]. ISBN 978-1-4673-8851-1. DOI 10.1109/CVPR.2016.445. Disponible en: <http://ieeexplore.ieee.org/document/7780814/>.

SCHÖNBERGER, J.L.; et al. Pixelwise View Selection for Unstructured Multi-View Stereo. Computer Vision – ECCV 2016 [en línea]. Cham: Springer International Publishing, Lecture Notes in Computer Science, 2016, pp. 501-518. [Consulta: 16 febrero 2023]. ISBN 978-3-319-46486-2. Disponible en: http://link.springer.com/10.1007/978-3-319-46487-9_31.

TARDÓN GARCÍA, L.J. Metodos robustos de reconstrucción de escenas tridimensionales empenado informacion binocular. [en línea] (Trabajo de titulación). (Doctorado). ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN.UNIVERSIDAD POLITÉCNICA DE MADRID. Espana. 1999.[Consulta: 14 febrero 2023]. Disponible en: <https://oa.upm.es/707/>.

TERRYGLEE. *Información general sobre Visual Studio.* [en línea]. 2023.[Consulta: 15 febrero 2023]. Disponible en: <https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide>.

TORRES CIVIT, L. Sistema de reconstrucción de caras 3D basado en modelos y multicámara [en línea] (Trabajo de titulación). (Ingeniería) Universitat Politècnica de Catalunya. Catalunya-Espana. 2007. [Consulta: 7 marzo 2023]. Disponible en: <https://upcommons.upc.edu/handle/2099.1/4510>.

VÁZQUEZ, F. *Metrología y Normalización: Máquina de Medición por Coordenadas (MMC).* [en línea]. 2015. [Consulta: 15 enero 2023]. Disponible en: http://itpmetrologiaynormalizacion.blogspot.com/2015/12/maquina-de-medicion-por-coordenadas-mmc_7.html.

WONG QUIJANO, J.O. Reconstrucción de objetos mediante cámaras estéreo, utilizando múltiples vistas. [en línea] (Trabajo de titulación). (Maestría) Universidad Veracruzana. Mexico. 2013. pp. 4-29. [Consulta: 15 febrero 2023]. Disponible en: <https://cdigital.uv.mx/bitstream/handle/123456789/46841/WongQuijanoJesus.pdf?sequence=2>.

YILMAZ, O; & KARAKUS, F. Stereo and kinect fusion for continuous 3D reconstruction and visual odometry. Department of Computer Engineering, Turgut Özal University [en línea], 2013, pp. 115-118. [Consulta: 16 enero 2023]. ISBN 978-1-4799-3343-3. DOI 10.1109/ICECCO.2013.6718242. Disponible en: https://www.researchgate.net/profile/Ozgur-Yilmaz/publication/269327935_Stereo_and_kinect_fusion_for_continuous_3D_reconstruction_and_visual_odometry/links/567cf4a008aebccc4e0378d6/Stereo-and-kinect-fusion-for-continuous-3D-reconstruction-and-visual-odometry.pdf

ANEXOS

Anexo A Código de adquisición de imágenes estéreo mediante SDK.

```
// limitations under the license.
#include <opencv2/highgui/highgui.hpp>

#include "mynteye/logger.h"
#include "mynteye/api/api.h"

MYNTEYE_USE_NAMESPACE

int main(int argc, char *argv[]) {
    auto &&api = API::Create(argc, argv);
    if (!api) return 1;

    bool ok;
    auto &&request = api->SelectStreamRequest(&ok);
    if (!ok) return 1;
    api->ConfigStreamRequest(request);

    Model model = api->GetModel();

    if (model == Model::STANDARD) {
        // ir control: range [0,160], default 0
        api->SetOptionValue(Option::IR_CONTROL, 100);

        LOG(INFO) << "Set IR_CONTROL to "
            << api->GetOptionValue(Option::IR_CONTROL);
    }

    // MYNTEYE-S210A S2000 S200B don't support this option
    if (model == Model::STANDARD210A ||
        model == Model::STANDARD2 || model == Model::STANDARD200B) {
        LOG(INFO) << "Sorry, MYNTEYE-S210A S2000 S200B don't support ir control";
        return 0;
    }

    api->Start(Source::VIDEO_STREAMING);
```

```
cv::namedWindow("frame");

while (true) {
    api->WaitForStreams();

    auto &&left_data = api->GetStreamData(Stream::LEFT);
    auto &&right_data = api->GetStreamData(Stream::RIGHT);

    cv::Mat img;
    cv::hconcat(left_data.frame, right_data.frame, img);
    cv::imshow("frame", img);

    char key = static_cast<char>(cv::waitKey(1));
    if (key == 27 || key == 'q' || key == 'Q') { // ESC/Q
        break;
    }
}

api->Stop(Source::VIDEO_STREAMING);
return 0;
```

Anexo B Código del sistema de reconstrucción en Matlab.

```
clc
clear;
close all

tic



---


%% almacenamiento de parametros de calibracion
stereoParams{1}=load('17calibration.mat');%
stereoParams{2}=load('18calibration.mat');%
stereoParams{3}=load('19calibration.mat');%
stereoParams{4}=load('20calibration.mat');%
stereoParams{5}=load('20calibration.mat');%
stereoParams{6}=load('20calibration.mat');%
% stereoParams{7}=load('stereoParamsquare.mat');
% stereoParams{8}=load('prueba1-2.mat');
%stereoParams{5}=load('prueba3-2.mat');
%stereoParams{6}=load('prueba2-2.mat');



---


%% creacion de video a partir de un conjunto de imagenes
imageNames = dir(fullfile('dataset', 'left9', '*.png'));
imageNames = {imageNames.name}';
imageNames1 = dir(fullfile('dataset', 'right9', '*.png'));
imageNames1 = {imageNames1.name}';



---


%% Construccion del archivo de video (en blanco)
outputVideo = VideoWriter(fullfile('left.avi'));
outputVideo.FrameRate = 5;
open(outputVideo)
outputVideo1 = VideoWriter(fullfile('right.avi'));
outputVideo1.FrameRate =5;
open(outputVideo1)
% Recorra la secuencia de imágenes, cargue cada imagen
% y luego escríbala en el video
for ii = 1:length(imageNames)
```

```

for ii = 1:length(imageNames)
    img = imread(fullfile('dataset','left9',imageNames{ii}));
    writeVideo(outputVideo,img)
end
close(outputVideo)
for ii = 1:length(imageNames1)
    img1 = imread(fullfile('dataset','right9',imageNames1{ii}));
    writeVideo(outputVideo1,img1)
end
close(outputVideo1)
%% creacion de video que almaneca los frames de cada camara estereo
videoFileLeft = 'left.avi';
videoFileRight = 'right.avi';
%reproduccion de video
videoreproduccion = VideoReader(videoFileLeft);
player = vision.VideoPlayer('Position', [20,200,740 560], 'Name', 'CAMARA IZQUIERDA');
while hasFrame(videoreproduccion)
    frame = readFrame(videoreproduccion);
    step(player,frame);
    pause(1/videoreproduccion.frameRate)
end
release(player);
%% lectura de imagenes estereo
% lectura de datos de los videos left y right
readerLeft = VideoReader(videoFileLeft);
readerRight = VideoReader(videoFileRight);
player = vision.VideoPlayer('Position', [20,200,740 560]);

InL = readFrame(readerLeft);
InR = readFrame(readerRight);

disparityRange = [-16*1 256];
th = [-.4 .6;-.9 1.6;0 1];
i=0;

while hasFrame(readerLeft) && hasFrame(readerRight)
    % Read the frames.
    i=i+1;
    InL = readFrame(readerLeft);
    InR = readFrame(readerRight);
    % Rectify the frames.
    [OutL, OutR] = rectifyStereoImages(InL, InR, stereoParams{1,i}.stereoParams);
    % Convert to grayscale.
    OutLg = rgb2gray(OutL);
    OutRg = rgb2gray(OutR);

    disparityMap1 = disparity(OutLg,OutRg,'BlockSize',5, ...
        'DisparityRange',disparityRange,'UniquenessThreshold',11*1, ...
        'ContrastThreshold',0.05);

    points3D = reconstructScene(disparityMap1, stereoParams{1,i}.stereoParams)./1000;
    points3D = thresholdPC(points3D,th);
    ptCloudRef = pointCloud(points3D, 'Color', OutL);
    player3D = pcplayer([-0.3, .9], [-.7 .7], [0.4 3], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');
    view(player3D, ptCloudRef);
    ContenedorPT{i}=ptCloudRef;
end
release(player);
save('Reconstruccion.mat','ContenedorPT')

```

```
%% almacenamiento de cada en nube en objetos.
```

```
ptCloud1 = ContenedorPT{1};  
ptCloud2 = ContenedorPT{2};  
ptCloud3 = ContenedorPT{3};  
ptCloud4 = ContenedorPT{4};  
ptCloud5 = ContenedorPT{5};
```

```
player3D1 = pcplayer([-0.8, .9], [-.7 .9], [0 1], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');  
view(player3D1, ptCloud1);  
player3D2 = pcplayer([-0.8, .9], [-.9 1], [0 2], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');  
view(player3D2, ptCloud2);  
player3D3 = pcplayer([-0.8, .9], [-.7 .9], [0 2], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');  
view(player3D3, ptCloud3);
```

```
%% preprocesamiento de nubes antes de unir.
```

```
% vista 1
```

```
grados_ejex=-16;  
alfa = grados_ejex*pi/180;  
grados_ejey=45;  
theta = grados_ejey*pi/180;
```

```
A = [1,0,0,0;...  
     0,cos(alfa),sin(alfa),0; ...  
     0,-sin(alfa), cos(alfa),0; ...  
     0 0 0 1];
```

```
tform = affine3d(A);  
ptCloudR1 = pctransform(ptCloud1, tform);
```

```
B = [cos(theta),0,sin(theta),0;...  
     ... .., ... .., ... ..  
     0.9 .015 0.5 1];
```

```
tform = affine3d(B);  
ptCloudR1 = pctransform(ptCloudR1, tform);
```

```
th1 = [.23 .5;-.9 1.6;0.8 1.3]; % quitar ruido de terminaciones del objeto  
ptCloudR11 = thresholdPC(ptCloudR1.Location,th1);  
ptCloudR11 = pointCloud(ptCloudR11,"Color",ptCloudR1.Color);
```

```
player3D1 = pcplayer([-0.8, .9], [-.7 .9], [0 2], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');  
view(player3D1, ptCloudR11);
```

```
% vista 2
```

```
grados_ejex=-16;  
alfa = grados_ejex*pi/180;  
grados_ejey=45;  
theta = grados_ejey*pi/180;
```

```
A = [1,0,0,0;...  
     0,cos(alfa),sin(alfa),0; ...  
     0,-sin(alfa), cos(alfa),0; ...  
     0 0 0 1];
```

```
tform = affine3d(A);  
ptCloudR2 = pctransform(ptCloud2, tform);
```

```
B = [cos(theta),0,sin(theta),0;...  
     0,1,0,0; ...  
     -sin(theta),0, cos(theta),0; ...  
     0.9 .015 0.55 1];
```

```

tform = affine3d(B);
ptCloudR2 = pctransform(ptCloudR2, tform);

th2 = [.3 .7;-.9 1.6;0.8 1.27]; % quitar ruido de terminaciones del objeto
ptCloudR22 = thresholdPC(ptCloudR2.Location,th2);
ptCloudR22 = pointCloud(ptCloudR22,"Color",ptCloudR2.Color);
player3D2 = pcplayer([-0.8, .9], [-.7 .9], [0 2], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');
view(player3D2, ptCloudR22);
% vista 3
grados_ejex=-16;
alfa = grados_ejex*pi/180;
grados_ejey=45;
theta = grados_ejey*pi/180;

A = [1,0,0,0;...
     0,cos(alfa),sin(alfa),0; ...
     0,-sin(alfa), cos(alfa),0; ...
     0 0 0 1];

tform = affine3d(A);
ptCloudR3 = pctransform(ptCloud3, tform);

B = [cos(theta),0,sin(theta),0;...
     0,1,0,0; ...
     -sin(theta),0, cos(theta),0; ...
     0.9 .015 0.5 1];

tform = affine3d(B);
ptCloudR3 = pctransform(ptCloudR3, tform);

th3 = [.2 .7;-.9 1.6;0.8 1.3]; % quitar ruido de terminaciones del objeto
ptCloudR33 = thresholdPC(ptCloudR3.Location,th3);
ptCloudR33 = pointCloud(ptCloudR33,"Color",ptCloudR3.Color);
player3D3 = pcplayer([-0.8, .9], [-.7 .9], [0 2], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');
view(player3D3, ptCloudR33);

```

%% union y costura de nube de puntos

```

% vista 1
grados_ejex=0;
alfa = grados_ejex*pi/180;
grados_ejey=0;
theta = grados_ejey*pi/180;

A = [1,0,0,0;...
     0,cos(alfa),sin(alfa),0; ...
     0,-sin(alfa), cos(alfa),0; ...
     0 0 0 1];

tform = affine3d(A);
ptCloudR111 = pctransform(ptCloudR11, tform);

B = [cos(theta),0,sin(theta),0;...
     0,1,0,0; ...
     -sin(theta),0, cos(theta),0; ...
     0 0 0 1];

```

```

tform = affine3d(B);
ptCloudR111 = pctransform(ptCloudR111, tform);

% player3D5 = pcplayer([-0.8, .9], [-.7 .9], [0 2], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');
% view(player3D5, ptCloudR111);

% vista 2
grados_ejex=0;
alfa = grados_ejex*pi/180;
grados_ejey=-90;
theta = grados_ejey*pi/180;

A = [1,0,0,0;...
     0,cos(alfa),sin(alfa),0; ...
     0,-sin(alfa), cos(alfa),0; ...
     0 0 0 1];

tform = affine3d(A);
ptCloudR222 = pctransform(ptCloudR22, tform);

B = [cos(theta),0,sin(theta),0;...
     0,1,0,0; ...
     -sin(theta),0, cos(theta),0; ...
     -.88 .01 1.53 1];

tform = affine3d(B);
ptCloudR222 = pctransform(ptCloudR222, tform);

mergeSize=0.00001;

ptCloudScene1 = pcmerge(ptCloudR111, ptCloudR222, mergeSize);
player3D6 = pcplayer([-2 2], [0 1], [-2 2.5], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');
view(player3D6, ptCloudScene);

% vista 3
grados_ejex=0;
alfa = grados_ejex*pi/180;
grados_ejey=175;
theta = grados_ejey*pi/180;

A = [1,0,0,0;...
     0,cos(alfa),sin(alfa),0; ...
     0,-sin(alfa), cos(alfa),0; ...
     0 0 0 1];

tform = affine3d(A);
ptCloudR333 = pctransform(ptCloudR33, tform);

B = [cos(theta),0,sin(theta),0;...
     0,1,0,0; ...
     -sin(theta),0, cos(theta),0; ...
     0.725 .01 2.2 1];

tform = affine3d(B);
ptCloudR333 = pctransform(ptCloudR333, tform);

mergeSize=0.00001;
ptCloudScene2 = pcmerge(ptCloudScene1, ptCloudR333, mergeSize);
player3D6 = pcplayer([-2 2], [0 1], [-2 2.5], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');
view(player3D6, ptCloudScene2);

```

```
%% denoise and removing invalid points
objeto = pcdenoise(ptCloudScene1);
objeto = removeInvalidPoints(objeto);

player3D6 = pcplayer([0 1], [0 1], [0.5 1.5], 'VerticalAxis', 'y', 'VerticalAxisDir', 'down');
view(player3D6, objeto);
%% exportacion de nubes
toc
pcwrite(objeto, "mochila", PLYFormat="binary");
```



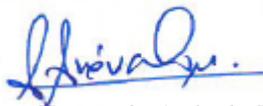
ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 29/ 03 / 2023

INFORMACIÓN DEL AUTOR/A (S)	
Nombres – Apellidos: Gabriel David Castro Dutan	
INFORMACIÓN INSTITUCIONAL	
Facultad: Informática y Electrónica	
Carrera: Telecomunicaciones	
Título a optar: Ingeniero en Electrónica, Telecomunicaciones y Redes	
f. Analista de Biblioteca responsable:	 Ing. Fernanda Arévalo M.



0522-DBRA-UPT-2023