



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Doctorado en Ingeniería.

RECONSTRUCCIÓN DE UN MODELO 3D A PARTIR DE UN OBJETO CON
SUPERFICIE NO RÍGIDA POR MEDIO DE REDES NEURONALES

Tesis

Que como parte de los requisitos para obtener el Grado de
Doctor en Ingeniería

Presenta

M.C. Luis Rogelio Román Rivera.

Dirigido por:

Dr. Jesús Carlos Pedraza Ortega

Dr. Jesús Carlos Pedraza Ortega
Presidente

Dr. Juan Manuel Ramos Arreguín
Secretario

Dr. Marco Antonio Aceves Fernandez
Vocal

Dr. Manuel Toledano Ayala
Suplente

Dr. Ramón Gerardo Guevara González
Suplente

Centro Universitario Querétaro, Qro.
Febrero 2024
México



Dirección General de Bibliotecas y Servicios Digitales
de Información



RECONSTRUCCIÓN DE UN MODELO 3D A PARTIR DE
UN OBJETO CON SUPERFICIE NO RÍGIDA POR MEDIO
DE REDES NEURONALES

por

Luis Rogelio Román Rivera

se distribuye bajo una [Licencia Creative Commons
Atribución-NoComercial-SinDerivadas 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Clave RI: IGDC-116006

RESUMEN

El objetivo de esta tesis es obtener una nube de puntos 3D a partir de una secuencia de imágenes RGB-D. Las imágenes tomadas por una cámara RGB-D generan dos capas de información de la misma escena, una capa a color y una capa que representa la profundidad de la escena, en la escena se encuentra un objeto dinámico, es decir que se encuentra moviéndose y deformándose, de tal manera que imágenes consecutivas del mismo objeto representan diferentes posiciones y deformaciones geométricas del mismo respecto a la primer toma capturada en orden cronológico. La reconstrucción 3D del objeto considera las deformaciones del mismo y para alcanzar este objetivo se utilizan diferentes estrategias para el mejoramiento de la calidad de las imágenes como lo son la calibración de la cámara con la que se capturan los datos de color y profundidad, así como la correspondencia de la información contenida en ambas capas de datos, para poder capturar la geometría del objeto en deformación también se consideran redes neuronales profundas para el seguimiento de características clave y para la representación de la información en 3D.

(Palabras clave: Redes Neuronales profundas, RGB-D, reconstrucción 3D, visión por computadora)

SUMMARY

The objective of this thesis is to obtain a 3D point cloud from a sequence of RGB-D images. Images taken by a camera RGB-D generates two layers of information from the same scene, a color layer and a layer that represents the depth of the scene, in the scene there is a dynamic object, that is that is moving and deforming, in such a way that consecutive images of the same object They represent different positions and geometric deformations of the same with respect to the first shot. captured in chronological order. The 3D reconstruction of the object considers its deformations and To achieve this objective, different strategies are used to improve the quality of the images. such as the calibration of the camera with which the color and depth data are captured, as well as the correspondence of the information contained in both data layers, in order to capture the geometry of the object in deformation, it is also consider deep neural networks for tracking key features and for information representation in 3D.

(Key words: Deep Learning, RGB-D, 3D reconstruction, computer vision)

A mi familia

RECONOCIMIENTOS

Agradezco especialmente a mi familia por todos los sacrificios que representaron acompañarme en el transcurso del programa de Doctorado, a mis hijos con los que dejé de ir a jugar en muchas ocasiones debido a que estaba ocupado con compromisos del doctorado, a mi esposa, ya que además de trabajar y cumplir con mi horario habitual, tuve que dedicar fines de semana, días libres y vacaciones para avanzar en mi doctorado, le agradezco por ser comprensiva a pesar de las adversidades que vivimos y por estar siempre a apoyándome.

Agradezco a mi director tesis, por todas las atenciones, el seguimiento, el tiempo dedicado, pero sobretodo por todo el conocimiento y enseñanzas compartidas, sin duda alguna un pilar y apoyo muy importante y clave para poder terminar el programa del doctorado con éxito.

Agradezco al sínodo por todas las atenciones brindadas durante este trabajo de investigación. Agradezco a todos mis profesores por la cátedra impartida, sin duda alguna sumaron mucho en mi formación.

Agradezco a la Universidad Autónoma de Querétaro y a la Facultad de Ingeniería, por abrir sus puertas y haberme brindado la oportunidad de crecer en conocimiento y cambiar mi vida para bien.

Haber concluido el programa de Doctorado en Ingeniería es una satisfacción y un logro personal que requirió haber salido de un estado de confort, dedicación, tiempo, esfuerzo, recursos, muchos sacrificios, pero sin duda alguna este esfuerzo se lo dedico a mi familia.

ÍNDICE GENERAL

Resumen	2
Summary	3
Reconocimientos	5
Lista de Tablas	8
Lista de Figuras	9
Lista de acrónimos	13
1. INTRODUCCIÓN	13
1.1. Inteligencia Artificial y Redes Neuronales	13
1.2. Redes Neuronales Artificiales (ANN) <i>Artificial Neural Networks</i>	14
1.2.1. Historia de las ANN	14
1.2.2. Redes Neuronales en problemas de Visión	15
Redes Neuronales Profundas	16
Redes Neuronales Convolucionales	16
1.3. Cámaras de tipo RGB-D	17
1.3.1. Incertidumbres en cámaras RGB-D	21
1.3.2. Evolución de las cámaras RGB-D, correspondientes a los canales de color en formato RGB y a la profundidad.	22
1.3.3. Cámaras Intel Realsense	22
1.3.4. Sistema iPhone X TrueDepth	23
1.3.5. Calibración de las cámaras Red, Green, Blue, Depth por sus siglas en inglés. Que corresponden a Rojo, Verde, Azul y Profundidad (RGB-D)	24
Error geométrico	25
Error de alineación	25
1.4. Valores atípicos y ruido en la información de la capa de profundidad de las cámaras RGB-D.	27
1.5. Estado del arte.	28
1.6. Reconstrucción 3D a partir de imágenes RGB-D.	35
1.6.1. Color y Apariencia	35
1.6.2. Escenas estáticas	36
1.6.3. Escenas dinámicas	36
1.7. Justificación	38

1.7.1.	Importancia del tema	38
1.8.	Descripción del problema	39
1.8.1.	Reconstrucción 3D de objetos dinámicos con superficies deformables mediante datos RGB-D	39
2.	MARCO TEÓRICO	42
2.1.	Fundamentación Teórica	42
2.1.1.	Coordenadas homogéneas	42
2.1.2.	Coordenadas homogéneas a no-homogéneas	43
2.2.	Modelo de proyección de cámara	43
2.3.	Parámetros intrínsecos de la cámara	43
2.3.1.	Longitud focal	44
2.3.2.	Centro óptico y punto principal	44
2.3.3.	Inclinación (<i>skew</i>)	44
2.4.	Matriz de parámetros intrínsecos	45
2.4.1.	Coordenadas del mundo 3D a coordenadas homogéneas 2D	45
2.4.2.	Tamaño de pixel (<i>pixel pitch</i>)	46
2.4.3.	Matriz inversa de parámetros intrínsecos	46
2.5.	Modelos de distorsión	47
2.5.1.	Modelo de distorsión radial	47
2.5.2.	Distorsión radial de un lente	47
2.5.3.	Cálculo iterativo de la distorsión radial inversa	47
2.5.4.	Fórmula exacta para el cálculo de la distorsión radial inversa	48
2.5.5.	Modelo de distorsión radial polinomial	49
2.5.6.	Coordenadas del mundo 3D a coordenadas de imagen.	50
2.5.7.	Transformación de coordenadas del mundo 3D a coordenadas de imagen ideales (sin distorsión)	50
2.5.8.	Modelo de cámara estenopéica (<i>Pinhole camera model</i>)	50
2.6.	Modelo general de una cámara de tipo RGB-D	51
2.6.1.	Transformación de coordenadas de la capa de profundidad a coordena- das del mundo en 3D (sin distorsión)	52
2.6.2.	Modelo para representar la esfera con un tamaño conocido.	53
2.6.3.	Puntuación Z.	54
2.7.	Captura de la escena y objeto a reconstruir	54
2.7.1.	EMF completo	55
2.7.2.	EMF angular ó velocidad angular de la cámara	55
2.8.	Bloque NeRF	55
2.8.1.	Bases teóricas NeRF	55
2.9.	Detección y Seguimiento de puntos clave de objetos dinámicos con superficies deformables	57
2.9.1.	Código latente de apariencia	57
2.9.2.	Código de deformación	57
2.9.3.	Función pérdida	58
2.10.	Redes Neuronales Convolucionales	58
2.11.	Conjunto de datos.	59

2.12. Valores atípicos y ruido en nubes de puntos en 3D	59
2.12.1. Evaluación del modelo	60
3. METODOLOGÍA	62
3.1. Software y hardware utilizado	62
3.2. Cámara RGB-D.	63
3.3. El método RANSAC y el balón de basketball	65
3.4. Calibración de la cámara RGB-D	69
3.4.1. Método para la calibración de la cámara RGB-D	69
3.5. Limpieza de valores atípicos y ruido en nubes de puntos en 3D	72
3.5.1. Red neuronal basada en PointCleanNet	72
3.5.2. Conjunto de datos para la limpieza de valores atípicos y ruido.	74
3.5.3. Arquitectura neuronal propuesta para la limpieza de valores atípicos y ruido.	75
3.5.4. Entrenamiento de la red neuronal propuesta para reducir valores atí- picos y ruido.	77
3.5.5. Metodología reconstrucción 3D de un objeto dinámico	78
4. RESULTADOS	80
4.1. Detección de la esfera de basketball	80
4.2. Calibración de la cámara RGB-D	89
4.3. Valores atípicos y ruido en nubes de puntos 3D	92
4.4. Reconstrucción 3D	96
4.4.1. Nubes de puntos 3D	96
4.4.2. Comparativas	98
4.4.3. Valores 3d a medidas en el mundo real	102
4.4.4. Características del modelo 3D generado	103
5. CONCLUSIONES Y TRABAJO FUTURO	104
5.1. Detección de una esfera representada por un balón de basketball	104
5.2. Calibración de la cámara RGB-D	104
5.3. Valores atípicos y ruido en la nube de puntos 3D	104
5.4. Reconstrucción 3D del objeto dinámico en escena	105
Apéndices	112
.1. Artículo I	112
.2. Artículo II	114
.3. Artículo III	116
.4. Requisito manejo de la lengua inglés	118
.5. Proyecto FONDEC 2021-2023	120
.6. Productos obtenidos	122

ÍNDICE DE TABLAS

1.1. Estado del Arte de reconstrucción 3D	30
3.1. Intel Realsense D435(Keselman et al., 2017).	63
3.2. Conjunto de datos utilizado para entrenamiento y validación.	74
3.3. Conjunto de datos de entrenamiento y validación	77
4.1. Error RMSE en los centros detectados con RANSAC y ajustados con Z-score	87
4.2. Comparación respecto a otros métodos	88
4.3. Parámetros Intrínsecos cámara capa RGB	90
4.4. Parámetros de rotación cámara RGB-D.	90
4.5. Parámetros de traslación cámara RGB-D.	90
4.6. Inferencias en nubes de puntos 3D con baja densidad de elementos	93
4.7. Comparativo considerando F1-score	94
4.8. Comparativo considerando Chamfer Loss.	95
4.9. Características de las nubes de puntos 3D.	103
1. Productos obtenidos durante el programa de Doctorado en Ingeniería.	122

ÍNDICE DE FIGURAS

1.1. Taxonomía de la Inteligencia Artificial (IA) basada en (Alom et al., 2019).	14
1.2. Historia de las Redes Neuronales 1943-2012 basado en (Alom et al., 2019).	15
1.3. Avances Redes Neuronales 2012 - 2017	16
1.4. Red Neuronal Convolutacional (Nagi et al., 2011)	17
1.5. Sensor RGB-D Kinect V2	17
1.6. Cámara realsense RSD435	18
1.7. Cámara Structure Core de Occipital	18
1.8. Cámara de Stereolabs, ZED	19
1.9. Cámara con tecnología TrueDepth	19
1.10. Ejemplo de captura con cámara RGB-D	20
1.11. Incertidumbre y rango de medición de sistemas y metodos de sensado 3D (Rosin et al., 2019)	21
1.12. Sensor RGB-D Kinect (Han et al., 2013).	22
1.13. Sensor Realsense R200 (Keselman et al., 2017).	23
1.14. Sensores Realsense D415 y D435 (Giancola et al., 2018).	23
1.15. Sensor iPhone X TrueDepth (LeCompte et al., 2019).	24
1.16. Ejemplo generado por cámara TrueDepth.	24
1.17. Distorsión de barril.	25
1.18. Error de alineación.	26
1.19. Ejemplo del proceso manual en el método de Staranowicz y otros (A. Staranowicz et al., 2014; A. N. Staranowicz et al., 2015).	27
1.20. Línea del tiempo reconstrucción 3D con datos de color y profundidad	28
1.21. Colormap Optimization (Zhou & Koltun, 2014)	35
1.22. Patch Based Optimization (Bi et al., 2017)	35
1.23. Strand-Accurate Multi-View Hair Capture (Nam et al., 2019)	36
1.24. KinectFusion (Newcombe et al., 2011)	36
1.25. Fusion4D (Dou et al., 2016)	37
1.26. Holoportation (Orts-Escolano et al., 2016)	37
1.27. Objeto dinámico vista de una cámara RGB-D (Innmann et al., 2016)	39
1.28. Objeto dinámico reconstruido una sola cámara RGB-D (Newcombe et al., 2015)	40
2.1. Imagen basada en (“Explaining Homogeneous Coordinates and Projective Geometry”, 1970).	42
2.2. Modelo de proyección de cámara.	44
2.3. Parámetro de inclinación	45
2.4. Modelo de cámara estenopéica (Tsai, 1987)	50

2.5. Modelo de cámara de profundidad basado en (Liu et al., 2020)	52
3.1. Nube de puntos 3D	64
3.2. Representación de profundidad	64
3.3. Diagrama de flujo para encontrar una esfera en una nube de puntos.	65
3.4. Escena experimento 1	66
3.5. Imagen de profundidad correspondiente a la escena del experimento 1	67
3.6. Visualización del archivo PLY del experimento 1 en el software Meshlab	67
3.7. Configuración de la cámara y tripié.	69
3.8. Conjunto de imágenes para la corrección del error geométrico.	70
3.9. Metodología propuesta para la calibración de la cámara RGB-D.	70
3.10. Se requieren de 5 pares de imágenes color-profundidad.	71
3.11. Etapas de reducción de ruido de PointCleanNet como se describen en (Rakotosaona et al., 2020).	72
3.12. Arquitectura PCPNet como se describe en (Guerrero et al., 2018).	73
3.13. Ejemplo del conjunto de datos, el mapa de calor muestra la profundidad de la escena.	74
3.14. Reconstrucción 3D del balón de basketball.	75
3.15. Nube de puntos correspondiente a la reconstrucción del balón de basketball con ruido Gaussiano de 1×10^{-3} desviaciones estándar	75
3.16. Bloque lineal convolucional básico propuesto.	76
3.17. Arquitectura neuronal propuesta basada en PointCleanNet (Rakotosaona et al., 2020).	77
3.18. Metodología para la reconstrucción de un objeto 3D dinámico en una nube de puntos.	78
4.1. Correcta detección del balón de basketball en el experimento 1 mediante Meshlab	80
4.2. Mayor detalle de la esfera detectada.	80
4.3. Esfera ajustada con valores atípicos en rojo, visualización mediante Meshlab	81
4.4. Escena del experimento 2	82
4.5. Escena del experimento 2, información del archivo PLY visualizada en Meshlab.	82
4.6. Correcta detección en la escena del experimento 2.	83
4.7. Detección incorrecta.	83
4.8. Tercer experimento con el balón en una nueva posición.	84
4.9. Escena en escala de colores jet, se muestra la profundidad de la escena en el experimento 3.	84
4.10. Visualización de la escena del experimento 3 sin detectar el balón.	85
4.11. Balón de basketball correctamente detectado en el 3er experimento	85
4.12. Ejemplo de valores Z-score antes de realizar el filtrado y encontrar el barycentro en el experimento 1.	86
4.13. Acercamiento de la esfera en el experimento 1 con valores a remover con Z-score en rojo, visualización mediante Meshlab.	86

4.14. a) Escena Red, Green, Blue por sus siglas en inglés. Que corresponden a los canales Rojo, Verde y Azul (RGB), b) puntos 3D reprojectados a una imagen en 2D, c) detección CircleNet, d) Detección mediante el método propuesto reprojectando un círculo a una imagen 2D, e) Detección mediante el método propuesto en 3D.	87
4.15. Se observa la diferencia entre el tripié y el balón debido al filtrado de valores atípicos mediante z-core	88
4.16. Experimento adicional sosteniendo el balón.	89
4.17. Valores de profundidad arrojados por QUES method (Fathian et al., 2018) ajustado por medio de spline.	91
4.18. Comparativo cualitativo respecto al trabajo de (A. N. Staranowicz et al., 2015) a) contra la propuesta realizada b)	91
4.19. Inferencia con PointCleanNet.	92
4.20. Inferencia con el modelo propuesto	92
4.21. Inferencia con la nube de puntos patron	92
4.22. Comparación cualitativa contra otros métodos en el estado del arte.	93
4.23. Nubes de puntos generadas, datos crudos I	96
4.24. Nubes de puntos generadas, datos crudos II	97
4.25. Comparativa cualitativa modelo 3D contra objeto real I	98
4.26. Comparativa cualitativa modelo 3D contra objeto real II	99
4.27. Comparativa cualitativa modelo 3D contra objeto real III	100
4.28. Modelo 3D generado filtrando valores atípicos	101
4.29. Conversión al tamaño del objeto en el mundo real.	102

LISTA DE ACRÓNIMOS

ANN Redes Neuronales Artificiales (*ANN*) *Artificial Neural Networks*

ASIC *Application-Specific Integrated Circuit* por sus siglas en inglés, Circuito Integrado de Aplicación Específica

CNN Redes neuronales convolucionales por sus siglas en inglés *Convolutional Neural Networks*

CPU Central Processing Unit por sus siglas en inglés, Unidad de Procesamiento Central

EMFs Effective Multi-view factors, por sus siglas en inglés factores de capturas efectivas multi-vista

GDDR6 Graphics Double Data Rate 6 por sus siglas en inglés

IA Inteligencia Artificial

IoT *Internet of Things* por sus siglas en inglés. Que corresponden a Internet de las Cosas

LTS Long Term Support por sus siglas en inglés

MB Mega Byte por sus siglas en inglés

MLP Multi Layer Perceptron por sus siglas en inglés

NeRF NEural Radiance Field ó campos de luminosidad neuronales por sus siglas en inglés

Pinhole-Camera-Model Pinhole camera model por sus siglas en inglés, modelo de cámara estenopéica

Pinhole Centro Óptico

QST Quaternion Spatial Transformer, Transformer Espacial basado en cuaterniones

QuEsT Quaternion Based Camera Pose Estimation, Estimación de la posición de la cámara basada en cuaterniones

RANSAC *RANdom SAmples Consensus*, consenso de muestras aleatorias.

RGB-D Red, Green, Blue, Depth por sus siglas en inglés. Que corresponden a Rojo, Verde, Azul y Profundidad

RGB Red, Green, Blue por sus siglas en inglés. Que corresponden a los canales Rojo, Verde y Azul

RMSE Root Mean Square Error, Error cuadrático medio

SDF Signed Distance Function por sus siglas en inglés, función de distancia orientada

SLAM *Simultaneous Localization And Mapping* por sus siglas en inglés, localización y mapeo simultáneos

STN Spatial Transformer Network, Red Espacial de modelo Transformer

cIOU Circle Intersection Over Union por sus siglas en inglés

mAP Mean Average Precision por sus siglas en inglés

vCPU virtual Central Processing Unit por sus siglas en inglés, CPU virtual

1. INTRODUCCIÓN

1.1. Inteligencia Artificial y Redes Neuronales

Durante el proceso de investigación de esta tesis, el tema de la IA se encuentra vigente en temas de la vida cotidiana y muchos aspectos de la vida humana, no solamente en países de primer mundo sino en todo alrededor del planeta. Se genera y almacena información por medio de toda clase de sensores y dispositivos *Internet of Things* por sus siglas en inglés. Que corresponden a Internet de las Cosas (IoT). Tostadoras, licuadoras y lavadoras de ropa ya incorporan algoritmos de IA.

Ciudades como Shenzhen en China ahora monitorean a sus ciudadanos por medio de cámaras de video y utilizan los datos generados por éstas cámaras para procesar en tiempo real el reconocimiento facial por medio de redes neuronales, identificando automáticamente a las personas y ejecutando diversas políticas sociales en base al comportamiento de cada individuo. Sin duda alguna en este periodo vivimos una línea delgada donde temas de ciencia ficción se vuelven temas de ciencia moderna y en gran parte por los modelos neuronales que se van proponiendo por los investigadores.

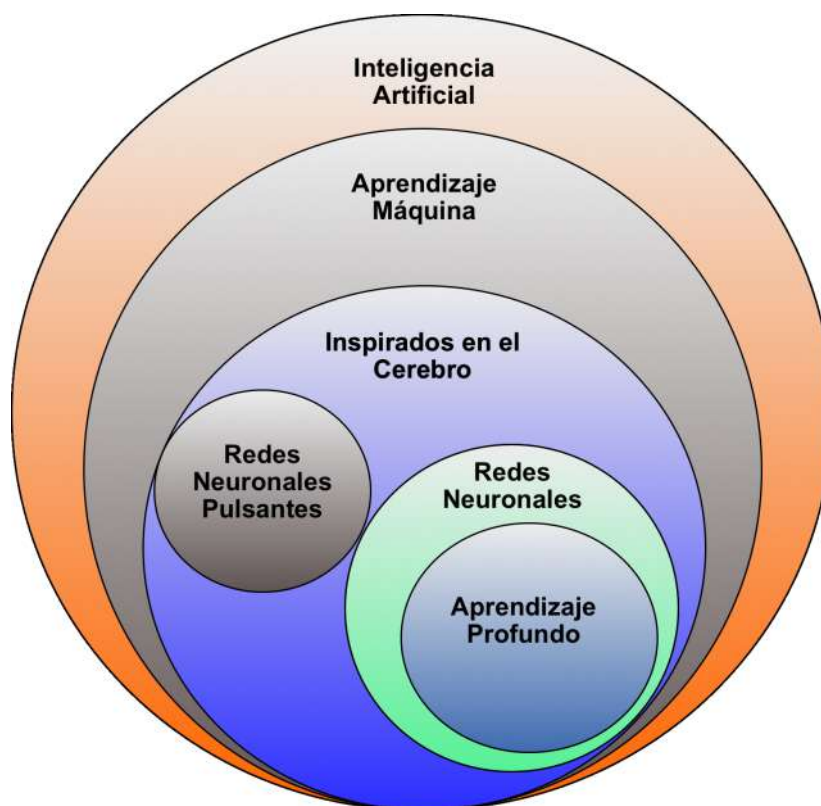


Figura 1.1: Taxonomía de la IA basada en (Alom et al., 2019).

El estudio y diseño de métodos computacionales que pueden realizar tareas que normalmente requieren de inteligencia humana, es la llamada IA (Hu et al., 2019), en los últimos años se ha popularizado en gran parte debido a la ciencia ficción pero también y más importante por logros que se hicieron realidad gracias al trabajo en investigación científica en el tema. En 1997 *DeepBlue*, una super computadora propiedad de la empresa IBM logró ganarle al campeón de ajedrez Gary Kasparov, posteriormente en 2016 *AlphaGo* un programa creado por la empresa Google usando redes neuronales logra ganarle a uno de los mejores jugadores profesiones de *go*, el jugador chino *Fan Hui* (Silver et al., 2016), los cuales son solo algunos logros que se hicieron populares y que han puesto de moda el tema de la inteligencia artificial.

Sin embargo las técnicas inspiradas en el cerebro han comenzado a prevalecer debido a los resultados obtenidos y a su amplia variedad de campos de aplicación Figura 1.1.

1.2. Redes Neuronales Artificiales (ANN) *Artificial Neural Networks*

Las Redes Neuronales Artificiales (ANN) *Artificial Neural Networks* (ANN) son empiezan a surgir debido a la investigación relacionada al cerebro humano en los años cuarentas.

1.2.1 Historia de las ANN

En 1943 se propone que los eventos neuronales y las relaciones entre ellos se pueden tratar de manera lógica propocional describiendo el comportamiento de una red en estos

términos (McCulloch & Pitts, 1943). En 1958 se plantea un modelo probabilístico para el almacenamiento y organización de información en un sistema nervioso hipotético llamado perceptrón (Rosenblatt, 1958). En 1969 se plantean las limitantes del perceptrón (Minsky & Papert, 1969) y se congelan los avances en el tema durante 10 años aproximadamente (Alom et al., 2019). En 1985 se propone el algoritmo de propagación reversa *backpropagation* y se revitaliza el tema (Ackley et al., 1985). En 1988 se propone una red neuronal jerárquica llamada neurocognitrón con capacidad de reconocimiento de patrones visuales (Fukushima, 1988). En 1998 se propone una Redes neuronales convolucionales por sus siglas en inglés Convolutional Neural Networks (CNN) para el reconocimiento de patrones de texto utilizando técnicas de gradiente descendiente para el aprendizaje de la arquitectura de red con resultados superiores a todo lo conocido hasta el momento (LeCun et al., 1998). En 2006 se soluciona el problema para el entrenamiento de redes con capas profundas (Hinton et al., 2006) y se obtienen mejores resultados en reducción de dimensionalidad de datos que el método de análisis de componentes principales (Hinton & Salakhutdinov, 2006). En 2012 se produce una explosión en el área de las redes neuronales debido a las anteriores aportaciones y además que el hardware necesario permitió el entrenamiento de las redes profundas en menor tiempo, la disponibilidad de datos necesarias para el aprendizaje se encuentran disponibles de manera dispersa en Internet (Alom et al., 2019), Figura 1.2.

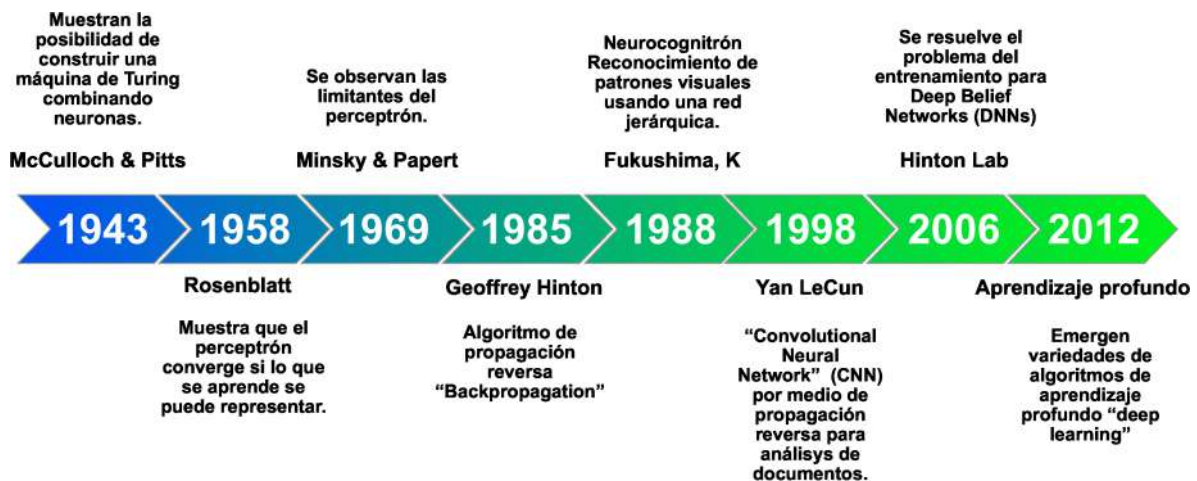


Figura 1.2: Historia de las Redes Neuronales 1943-2012 basado en (Alom et al., 2019).

1.2.2 Redes Neuronales en problemas de Visión

A partir del 2012 se ha observado una transición importante de los métodos tradicionales en problemas de Vision por computadora a la utilización de métodos con redes neuronales Figura 1.3.

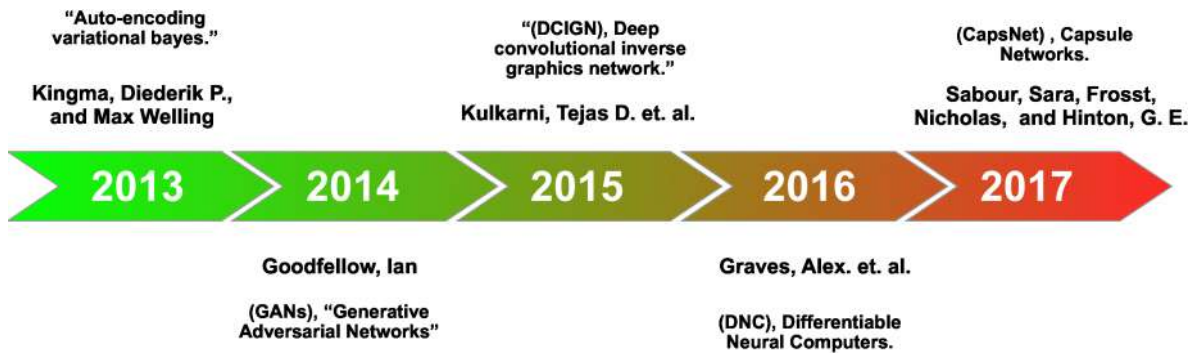


Figura 1.3: Avances Redes Neuronales 2012 - 2017

Redes Neuronales Profundas

Una red neuronal profunda es un algoritmo que aprende una función de clasificación no lineal sobre una colección de datos etiquetados y se compone de las siguientes etapas (Wilmanski et al., 2016):

- Capa de entrada, la cuál consiste en la lectura de los datos con un tamaño definido.
- Capas escondidas, realizan la extracción y acumulación de características de los datos procesados por la capa de entrada o por otra capa escondida.
- Capa de salida, la cuál provee una clasificación.

La topología de la red se refiere a como está diseñada la red, cantidad de capas incluidas entrada y salida, cantidad y tipos de neuronas.

Redes Neuronales Convolucionales

Las CNN emplean una operación de convolución en lugar de una multiplicación de matrices planas como en las redes neuronales tradicionales, consisten principalmente de 3 tipos de capas Figura 1.4. (Ioannidou et al., 2017):

- Capa Convolutacional.
- Capa de Agrupación.
- Capa totalmente conectada.

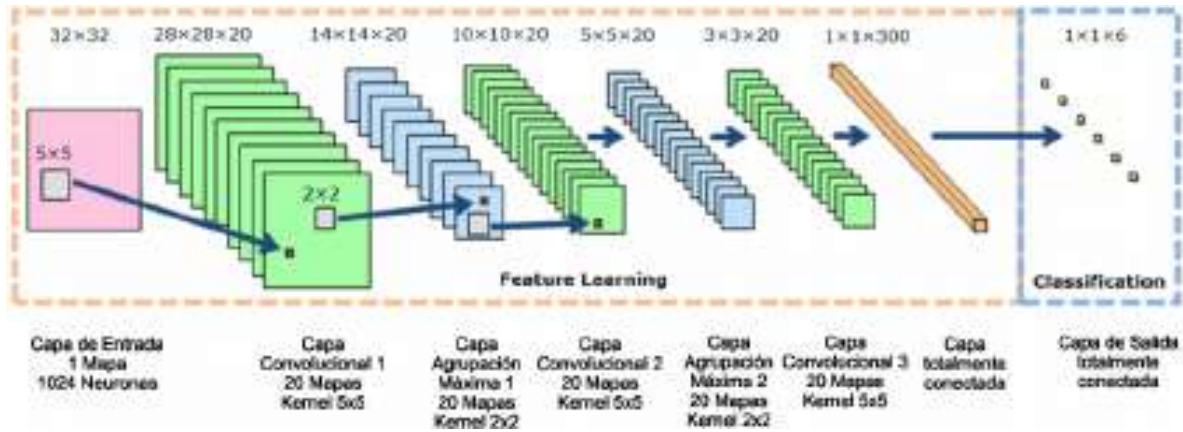


Figura 1.4: Red Neuronal Convolutiva (Nagi et al., 2011)

1.3. Cámaras de tipo RGB-D

Las cámaras RGB-D se han convertido en un sensor común en el área de visión por computadora debido a su popularidad y disponibilidad en el mercado, su tamaño y costo. Microsoft introduce al mercado el Kinect V1 para su uso en videojuegos con la consola Xbox 360 en 2010. Kinect cuenta con tecnología de profundidad 3D utilizando luz estructurada (Huang et al., 2021). Tiene un rango visual desde 0.8 hasta 4 metros, produciendo imágenes de 640 x 480 y es cuando la cámara puede ser usada en una computadora personal e inicia el aprovechamiento de la misma para investigación científica (Han et al., 2013), (Slavcheva et al., 2017). Dos años después, Kinect V2 reemplaza al Kinect V1 en la consola Xbox One, con un rango visual desde 0.5 hasta 4.5 metros y una resolución de 512 x 424, a 1080p en la capa RGB (Giancola et al., 2018; Han et al., 2013).



249 mm - ancho x 66 mm largo x 67 mm - alto
Kinect V2

Figura 1.5: Sensor RGB-D Kinect V2

Después del éxito del Kinect diferentes alternativas de cámaras RGB-D aparecieron. Intel liberó su cámara Realsense (Han et al., 2013; Keselman et al., 2017), reduciendo

el tamaño y consumo energético (Giancola et al., 2018) y proporcionando un kit de desarrollo que permitió trabajar con los datos producidos por la cámara directamente desde una computadora personal para producir nubes de puntos 3D, Figura 1.6. La cámara Intel RealSense D415 RGB-D tiene un rango visual desde 0.16 hasta 10 metros y produce imágenes de profundidad con una resolución de 1280 x 720. La cámara RealSense D435 tiene un campo visual desde 0.2 hasta 4.5 metros. En estas cámaras, la profundidad es calculada mediante *Application-Specific Integrated Circuit* por sus siglas en inglés, Circuito Integrado de Aplicación Específica (ASIC) (Giancola et al., 2018).



90 mm - ancho x 25 mm - largo x 25 mm - alto
Intel Realsense D435

Figura 1.6: Cámara realsense RSD435

A su vez, occipital liberó su cámara Structure Core (“Structure by Occipital - Give Your iPad 3D Vision”, s.f.; Zollhöfer, 2019), Figura 1.7 persiguiendo una filosofía similar a Intel pero ampliando la disponibilidad de desarrollo al ecosistema de Apple. La cámara Structure Core de Occipital tiene un campo visual desde 0.3 hasta 5 metros, imágenes y produce imágenes de profundidad con una resolución de 1280 x 960 (“Structure by Occipital - Give Your iPad 3D Vision”, s.f.).



109 mm - ancho x 18 mm - largo x 24 mm - alto
Structure Core

Figura 1.7: Cámara Structure Core de Occipital

La empresa Stereolab propuso su cámara RGB-D ZED utilizando una estrategia diferente, generando la capa de profundidad a partir de disparidad de píxeles desde dos cámaras RGB integradas, Figura ???. La cámara de Stereolabs, ZED, tiene un rango visual desde 0.5 hasta 20 metros, y produce imágenes de profundidad de una resolución de 4416 x 1242. En la

Figura 1.8 se muestra una cámara ZED 2i, cuenta con un acelerómetro, giroscopio, barómetro, magnetómetro y sensor de temperatura (Neupane et al., 2021).



175 mm - ancho x 33 mm - largo x 30 mm - alto
Zed Camera

Figura 1.8: Cámara de Stereolabs, ZED

Apple incorpora a sus teléfonos móviles la tecnología TrueDepth para autenticación y desbloqueo de pantalla durante el año 2018 Figure 1.9. Apple embebe en el iPhone la tecnología TrueDepth, con el propósito principal de autenticación mediante la geometría en 3D de los rostros y sus texturas (LeCompte et al., 2019).



~34.5 mm - ancho x ~4.5 mm alto
TrueDepth Camera

Figura 1.9: Cámara con tecnología TrueDepth

Este tipo de cámaras produce como salida de datos, una imagen a color y una imagen de profundidad, ambas imágenes describiendo la escena que es capturada por la cámara, Figura 1.10. Actualmente existen diferentes marcas y modelos de cámaras RGB-D, y es común que entre diferentes cámaras se encuentren diferentes características y limitaciones. Recientemente este tipo de cámaras ha sido embebido en dispositivos móviles como en (LeCompte et al., 2019) popularizando la tecnología y el uso de nubes de puntos 3D. Algunos formatos para guardar nubes de puntos 3D son el formato XYZ y el formato PLY, en donde se incluye información espacial describiendo cada punto en tres dimensiones y en ocasiones se puede incluir metadatos como color. Estos archivos pueden variar en tamaño y densidad de puntos y

estas características dependen mayormente de la marca y modelo de la cámara que se utiliza para generar dichos archivos. Se pueden encontrar miles de puntos en una escena capturada en una sola toma y de manera general la complejidad para el procesamiento de esta información se incrementa proporcionalmente con la calidad de la cámara y de la información que esta produce, a mayor detalle mayor densidad de puntos.

En parte izquierda de la Figura 1.10 se encuentra un ejemplo de la representación de profundidad en una imagen en tonos de color rojo, los objetos cercanos aparecen en color rojo claro y los objetos lejanos en rojo oscuro. En la parte derecha de la imagen, se muestra como la cámara de profundidad mediante una cámara RGB embebida captura la información de la capa de color correspondiente a la misma escena.



Imagen capa profundidad

Imagen capa color

Figura 1.10: Ejemplo de captura con cámara RGB-D

En el área de investigación de visión por computadora se pueden encontrar novedosos métodos que utilizan este tipo de información para resolver problemas en diferentes campos, por ejemplo, en reconstrucción 3D (Dou et al., 2016), localización y mapeo simultáneos (SLAM) como en (Rakotosaona et al., 2020; Wasenmüller et al., 2016). Existen diferentes conjuntos de datos ó bases de datos públicas que son recopiladas y organizadas para facilitar el trabajo de investigación utilizando información de diferentes escenarios representados en nubes de puntos 3D (Firman, 2016).

La detección de objetos en las nubes de puntos 3D es comúnmente usada para iniciar procesos posteriores como la calibración de cámaras (C. Zhang & Zhang, 2014), el registro de nubes de puntos (J. Yang et al., 2015), la agrupación de diversos objetos, la compresión de nubes de puntos 3D.

Esferas, conos, cilindros son utilizados como objetos geométricos debido a la simplicidad de ser representados y/o modelados matemáticamente (Liu et al., 2020; A. Staranowicz et al., 2013).

El *RANdom SAmple Consensus*, consenso de muestras aleatorias. (RANSAC) (Derpanis, 2010), es un método estocástico, el cuál utiliza muestras produciendo parámetros específicos para ser ajustados en una función costo y posteriormente listar las mejores propuestas o candidatos a solución (Fischler & Bolles, 1981). RANSAC (Derpanis, 2010), ha mostrado resultados prometedores localizando primitivas geométricas en nubes de puntos en tres

dimensiones (Schnabel et al., 2007), como tal puede ser adaptado en la búsqueda de parámetros específicos y puede acelerar el tiempo de cómputo debido a su naturaleza estocástica.

1.3.1 Incertidumbres en cámaras RGB-D

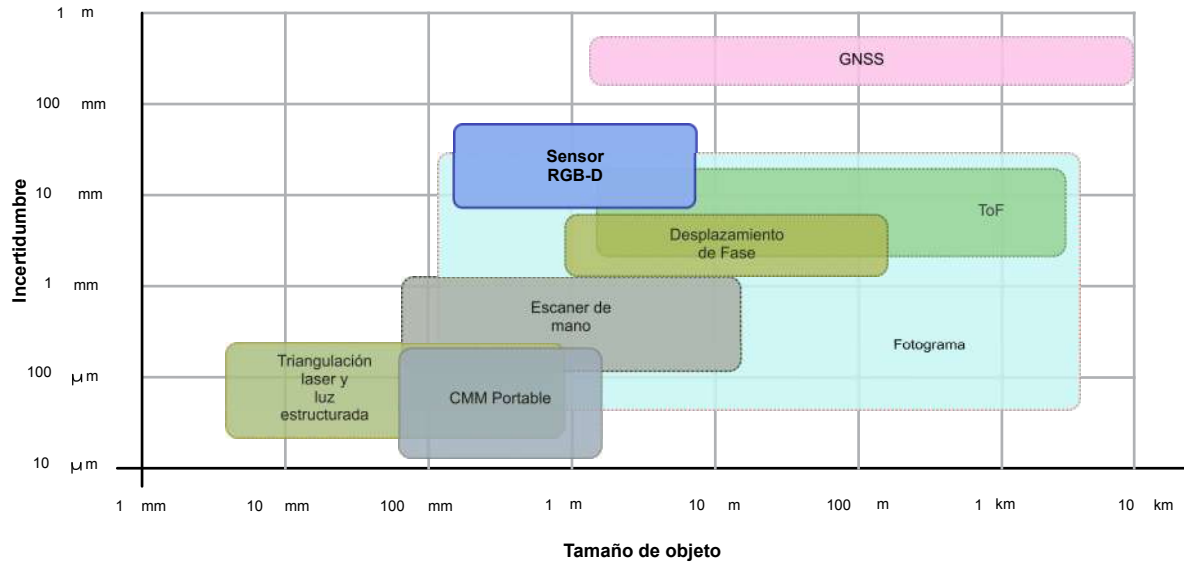


Figura 1.11: Incertidumbre y rango de medición de sistemas y métodos de sensado 3D (Rosin et al., 2019)

En la información proporcionada por una cámara RGB-D, ambas capas de datos, color y profundidad, corresponden en un área que se sobrelapa dadas las matrices de rotación y traslación adecuadas. El proceso de calibración de una cámara generalmente provee estas matrices como parámetros de fábrica sin embargo, en algunas aplicaciones los parámetros de fábrica no son suficientemente adecuados. Además considerando que diferentes cámaras RGB-D de la misma marca y modelo muestran diferentes parámetros de calibración, se observan diferentes errores aún comparando cámaras muy similares capturando las mismas escenas, mostrando distintos niveles de precisión y exactitud. En áreas de investigación como reconstrucción 3D, navegación 3D, robótica, visión por computadora, se utilizan datos de este tipo de cámara donde es importante asegurar y minimizar errores de calibración específicos para cada aplicación, con errores menores a los proporcionados por los parámetros de fábrica.

Se observa que el tamaño de los objetos que una cámara RGB-D de manera general puede capturar va desde los 20 cm hasta alrededor de los 10 m, Figura 1.11

1.3.2 Evolución de las cámaras RGB-D, correspondientes a los canales de color en formato RGB y a la profundidad.



Figura 1.12: Sensor RGB-D Kinect (Han et al., 2013).

El sensor Kinect V1 o Kinect 360 Figura 1.12 consta de los siguientes componentes:

- Una cámara RGB que provee imágenes de 680 x 480 píxeles en 30 cuadros por segundo y con opción de proveer imágenes RGB de 1280 x 1024 píxeles en 10 cuadros por segundo.
- Sensor RGB-D que consiste de un proyector infrarojo y de una cámara infraroja con un rango de 0.8m a 3.5m, 30 cuadros por segundo y un campo de visión de 57° horizontales y 47° verticales.
- Soporte motorizado con 27° de libertad horizontal y vertical.

1.3.3 Cámaras Intel Realsense

Intel comercializó cámaras RGB-D Realsense estéreoescópicas en 2015 como el modelo R200 y R400 reduciendo el tamaño e incluyendo hardware especializado dentro de la misma cámara para el cálculo y mejora de la calidad de la imagen de profundidad (Keselman et al., 2017).

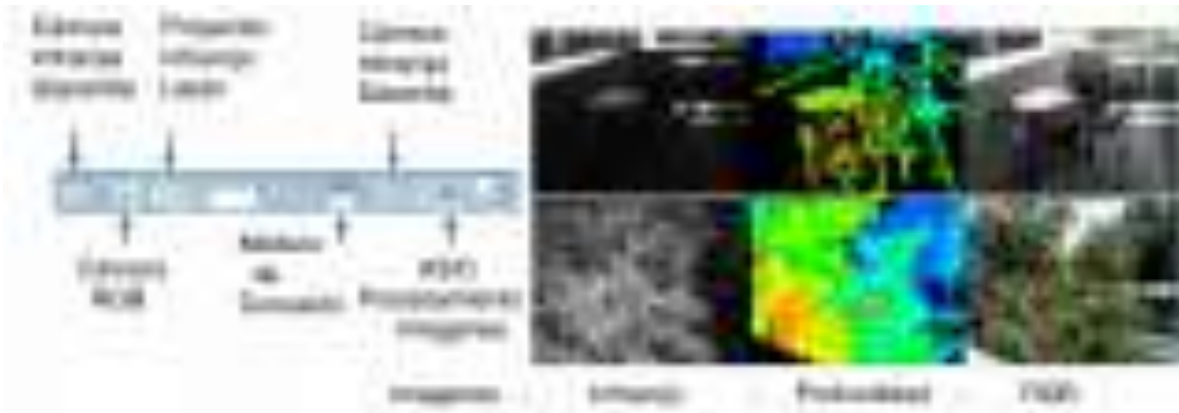


Figura 1.13: Sensor Realsense R200 (Keselman et al., 2017).

El modelo Realsense R200 Figura 1.13 incluye 3 cámaras, un par de cámaras infrarrojas estéreo y una cámara RGB, las cámaras estéreo están separadas 77 milímetros y capturan imágenes de 640 x 480, el campo de visión es de 60° horizontales y 45° verticales y las cámaras pueden soportar 30, 60 y 90 cuadros por segundo, la cámara de color es FullHD 1920 x 1080 con un campo de visión de 70° horizontales y 43° a 30 cuadros por segundo en FullHD y a mayores velocidades con menor resolución y se incluye un proyector infrarrojo laser que proyecta un patrón fijo para calcular la profundidad con las cámaras estéreo. Intel continúa activamente renovando su tecnología y ofreciendo alternativas con mejores prestaciones conforme avanza el tiempo como lo son las cámaras Realsense D415 y D435 Figura 1.14, el rango de la cámara D415 es de 0.16m a 10.0m y el rango de la cámara D435 es de 0.2m a 4.5m, diferente cambio de visión, del modelo D415 es de 63.4° horizontales y 40.4°, del modelo D435 es de 85.2° horizontales y 58° (Giancola et al., 2018).



Figura 1.14: Sensores Realsense D415 y D435 (Giancola et al., 2018).

1.3.4 Sistema iPhone X TrueDepth

En 2018 Apple libera el dispositivo móvil iPhone X que provee información 3D por medio de un sensor llamado TrueDepth Figura 1.15 y 1.16, basado en el principio de

luz estructurada, la funcionalidad de la cámara es similar a la del kinect pero reducida en tamaño e integrada a un dispositivo de telefonía móvil, el propósito original del dispositivo es el reconocimiento facial para desbloquear el dispositivo móvil sin embargo al igual que el kinect la tecnología empieza a ser utilizada con otros objetivos entre ellos de investigación (LeCompte et al., 2019).



Figura 1.15: Sensor iPhone X TrueDepth (LeCompte et al., 2019).



Figura 1.16: Ejemplo generado por cámara TrueDepth.

1.3.5 Calibración de las cámaras RGB-D

Un proceso de calibración de fábrica generalmente provee un conjunto de parámetros que permiten corresponder las capas de color y profundidad que generan las cámaras RGB-D, mediante matrices de rotación y de traslación. Sin embargo para algunas aplicaciones como la reconstrucción 3D, la navegación 3D, robótica, visión por computadora por mencionar algunas áreas de investigación, dichos parámetros de fábrica no son lo suficientemente precisos y exactos. Además, aún con la misma marca de cámara y el mismo modelo pero

diferentes dispositivos es común observar errores de calibración muy diferentes generando imágenes con diferencias aún de la misma escena tomadas en la misma posición. Considerando las características disponibles de las cámaras RGB-D y sus limitaciones, reducir errores de calibración es un paso esencial para utilizar los datos generados por dichas cámaras en aplicaciones que requieren mayor delicadeza.

Error geométrico

En la capa de color RGB, el error geométrico es producido por imperfecciones en la manufactura de los lentes causando efectos de barril y/o distorsiones concavas (C. Zhang & Zhang, 2014), mostrando curvas en lugar de líneas perfectas. Este tipo de distorsión afecta la geometría de los objetos en escena como se muestra en la Figura 1.17 donde las tablas del librero y la línea de la viga aparecen curvadas cuando deberían de ser rectas.



(a)
Distorsión de barril las líneas
aparecen como curvas

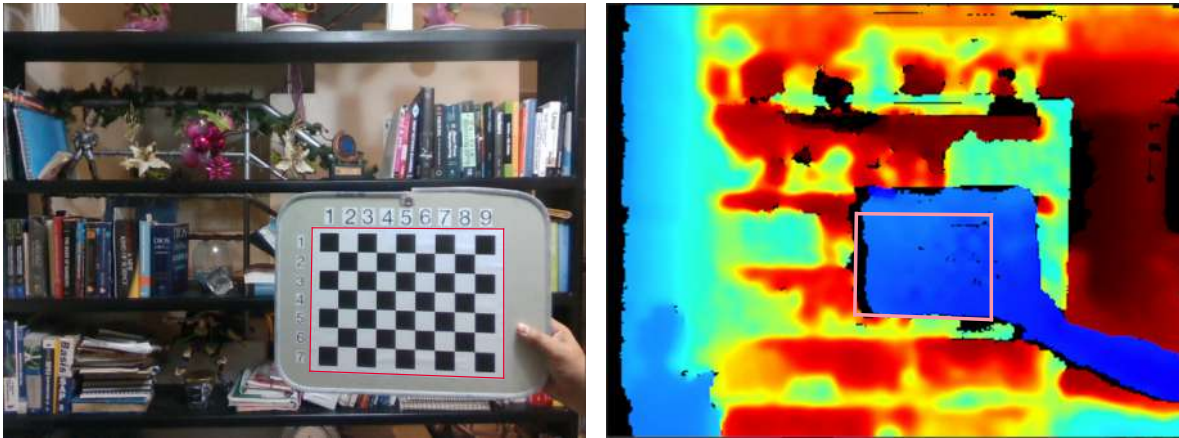
(b)
imagen corregida
las líneas aparecen como líneas

Figura 1.17: Distorsión de barril.

Cabe destacar que este tipo de error se acentúa más en la periferia de la imagen y en menor proporción hacia el centro de la misma.

Error de alineación

Cuando se alinean las dos capas de información que generan las cámaras RGB-D, la capa de color y la capa de profundidad se puede encontrar un error de alineación y correspondencia de los colores, texturas y orillas de los diferentes objetos contra los datos de los mismos objetos en la capa de profundidad (C. Zhang & Zhang, 2014), como se puede apreciar en la Figura 1.18 se aprecia en un recuadro rojo que las orillas de la hoja con el patrón de tablero ajedrez impreso, no corresponde a la posición correcta en la capa de profundidad.



Alineación y correspondencia entre las imágenes de color y profundidad

Figura 1.18: Error de alineación.

A la izquierda de la Figura 1.18 la imagen de la capa de color, a la derecha la representación en colores jet de la capa de profundidad. Un protocolo de calibración sencillo y fácil de replicar provee una manera de mantener a las cámaras RGB-D dentro de las precisiones y exactitudes deseadas para aplicaciones específicas. En el estado del arte, se encuentran diferentes protocolos de calibración con novedosos métodos que pueden ajustar los parámetros de calibración para afinar la precisión y exactitud mas allá de los valores de fábrica. Sin embargo requieren de una escena controlada y de instrumentos delicados, así como de experiencia en el operador que realizar el protocolo de calibración (Darwish et al., 2017), lo que provoca que el proceso de calibración sea difícil de reproducir especialmente para un operador sin experiencia. Recientemente han empezado a surgir métodos de calibración novedosos e innovadores los cuáles utilizan una esfera como geometría patrón para la calibración, esta geometría se tiene que detectar en ambas capas de información, color y profundidad, pero aún se requiere de un considerable número de muestras durante el protocolo de calibración, el trabajo propuesto en (A. N. Staranowicz et al., 2015), requiere de 130 pares de imágenes color-profundidad y posteriormente se requiere de intervención manual para seleccionar y enmascarar las posiciones de la esfera en escena.

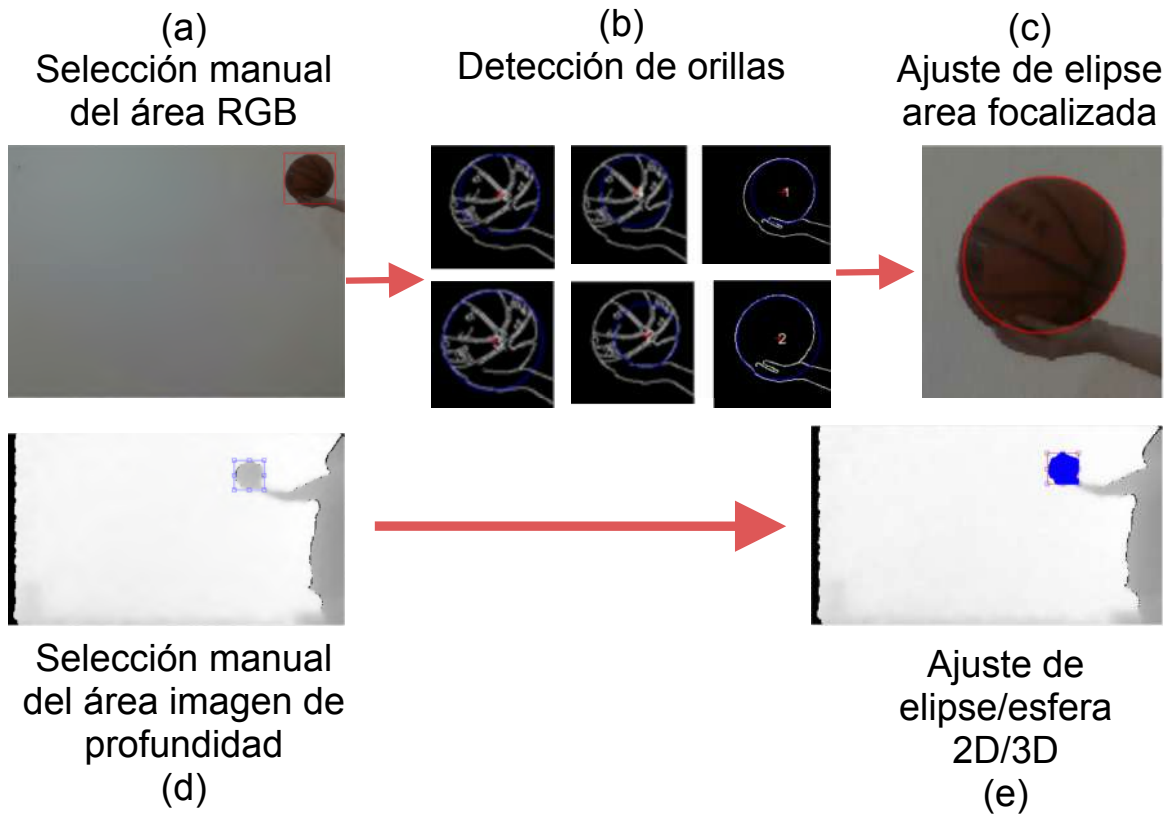


Figura 1.19: Ejemplo del proceso manual en el método de Staranowicz y otros (A. Staranowicz et al., 2014; A. N. Staranowicz et al., 2015).

Como se observa en la Figura 1.19 los pares de imágenes requieren de un proceso manual de selección del área donde se encuentra el balón en la imagen de color y donde se encuentra el círculo gris en la imagen que representa la capa de profundidad.

1.4. Valores atípicos y ruido en la información de la capa de profundidad de las cámaras RGB-D.

Las nubes de puntos en 3D generadas por las cámaras RGB-D usualmente contienen valores atípicos y ruido. Los valores atípicos se visualizan como discontinuidades arbitrarias en la geometría de los objetos y el ruido como puntos ubicados aleatoriamente sin respetar la continuidad de las geometrías de los objetos en escena. El ruido presente puede ocasionar diversos problemas en los cálculos que requieren una buena precisión de la geometría de los objetos en escena (Rakotosaona et al., 2020).

1.5. Estado del arte.

En la Figura 1.20 se observa como ha avanzado el estado del arte en el tema de reconstrucción 3D con imágenes que contienen información de color y profundidad, se seleccionan algunos trabajos de relevancia para el trabajo de investigación, (Zollhöfer, Nießner et al., 2014a) propone reconstruir un modelo 3D utilizando una plantilla previamente capturada de manera estática parametrizando la plantilla del modelo para el seguimiento de puntos clave. (Newcombe et al., 2015) propone utilizar *Simultaneous Localization And Mapping* por sus siglas en inglés, localización y mapeo simultáneos (SLAM) como método de seguimiento de características a las escenas a reconstruir, posteriormente se brinda un seguimiento a deformaciones en los objetos mediante el seguimiento de características en la propuesta de (Innmann et al., 2016).

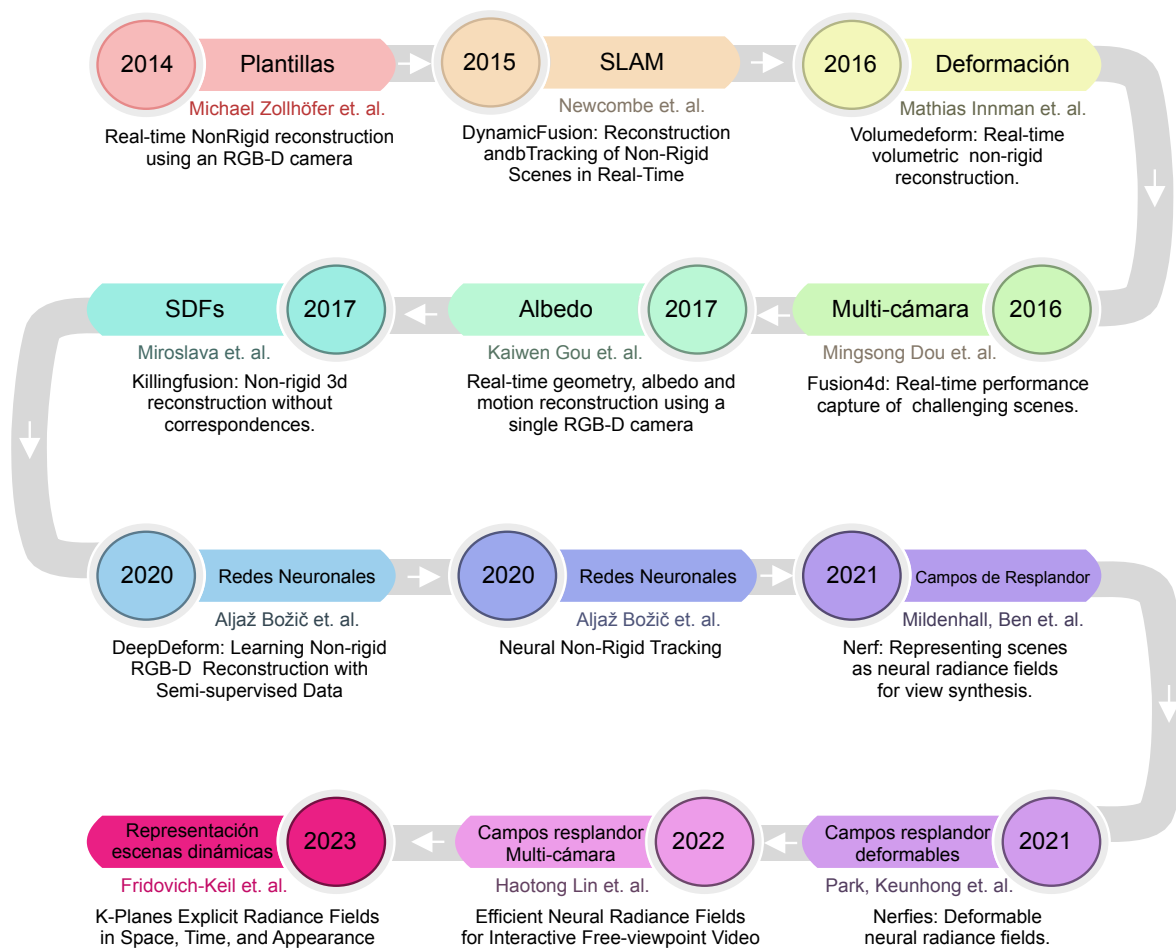


Figura 1.20: Línea del tiempo reconstrucción 3D con datos de color y profundidad

Para sobrellevar los retos hasta el momento en el seguimiento de las características de los objetos que contienen articulaciones y cuya superficie esta en constante cambio en la captura de los datos (Dou et al., 2016) utiliza múltiples cámaras para capturar un objeto desde los alrededores, mitigando los retos del seguimiento a las deformaciones, cada captura contiene la información necesaria para la reconstrucción desde diferentes perspectivas. Se propone una técnica diferente utilizando iluminación Albedo (Guo et al., 2017) sin embargo el método requiere de una ambientación controlada, mas tarde se utilizan Signed Distance Function por sus siglas en inglés, función de distancia orientada (SDF) para modelar y reconstruir los objetos de interes en el trabajo de (Slavcheva et al., 2017), se observa un estancamiento en el avance de propuestas en el año 2017 y se visualiza utilizar redes neuronales profundas para superar los retos presentes en ese momento (Ioannidou et al., 2017).

En el trabajo de (Bozic et al., 2020) se propone el uso de redes neuronales para solucionar el seguimiento de puntos clave en geometrías cuya superficie se deforma de una imagen a otra y se propone una base de datos para el entrenamiento y comparación de redes neuronales en este problema, posteriormente (Božič et al., 2020) propone un modelo neuronal atacando el problema del seguimiento de puntos clave.

En el año 2021 (Mildenhall et al., 2021) se proponen una nueva manera de representar una escena 3D utilizando campos de resplandor *Radiance Fields* brindando una ventaja sobre las propuestas realizadas hasta el momento capturando escenas estáticas complejas con características de iluminación que no habian sido capturadas con la fidelidad presentada en sus resultados, posteriormente en 2021 (Park et al., 2021) propone capturar objetos con superficies deformables e introduce una nueva estrategia para el seguimiento de las deformación acorde a la nueva propuesta basada en campos de resplandor, sin embargo continuan las áreas de oportunidad en el seguimiento de las deformaciones, en el año 2022 (Lin et al., 2022) explora en su propuesta la captura utilizando multiples cámaras y campos de radiación mejorando los resultados en la captura de objetos con superficies deformables. Se propone una manera de representar las escenas dinámicas y estáticas con mejoras al utilizar campos de radiación en el trabajo de (Fridovich-Keil et al., 2023).

El estado del arte se consolida en la Tabla 1.1.

Tabla 1.1: Estado del Arte de reconstrucción 3D

Identificación	Objetivo General	Categorías	Métodos	Resultados
(Zollhöfer, Nießner et al., 2014b)	Capturar el movimiento de un objeto dinámico siguiendo cambios en un modelo estático previamente capturado.	Reconstrucción 3D dinámica Imágenes RGB-D Color Apariencia	Se utiliza un modelo geométrico capturado previamente simplificando el seguimiento de deformaciones a puntos clave capturados en el modelo o plantilla.	Se propone un método para el seguimiento de objetos dinámicos conservando la apariencia generada de un modelo previo (Zollhöfer, Nießner et al., 2014a).
(Innmann et al., 2016)	Reconstrucción de un objeto dinámico a partir de una sola cámara RGB-D sin necesidad de un modelo previo.	Reconstrucción 3D dinámica Imágenes RGB-D Correspondencias. Mallado triangular.	Se usa modelo canónico generado de manera estática donde la geometría y el movimiento se siguen en base a color y profundidad, el modelo dinámico agrega nueva información que se registra en el modelo estático a su vez.	Se propone un método de reconstrucción 3D dinámica sin requerir un modelo previo y que utiliza una sola cámara RGB-D (Innmann et al., 2016).
(Guo et al., 2017)	Se propone un método para reconstrucción de objetos dinámicos sin necesidad de un modelo previo mediante el uso de un solo sensor RGB-D.	Reconstrucción 3D dinámica Imágenes RGB-D Color Apariencia	Utiliza un modelo de seguimiento de movimiento basado en el reflejo Lambertiano	Se propone un método que utiliza la geometría del objeto y un mapa albedo para el seguimiento de movimiento en un objeto dinámico sin necesidad de múltiples cámaras (Guo et al., 2017).

Continuación

Identificación	Objetivo General	Categorías	Métodos	Resultados
(Bi et al., 2017)	Mejorar la apariencia del modelo 3D por medio de texturas capturadas de imágenes RGB alineando el contenido de parches de las imágenes	Reconstrucción 3D Imágenes RGB-D Color Apariencia	Función costo que considera dos propiedades a) Cada imagen objetivo debe de ser similar a su imagen origen b) Cada imagen objetivo proyectada debe de ser fotométricamente consistente	Se presenta un sistema de optimización global basado en parches para mapeo de texturas basadas en imágenes donde se corrigen errores de alineación ocasionados por imprecisiones de geometría, posiciones de la cámara y distorsiones ópticas (Bi et al., 2017).
(Zollhöfer et al., 2018)	Se analizan los recientes desarrollos en reconstrucción de escenas basadas en imágenes RGB-D.	Reconstrucción 3D Imágenes RGB-D Reconstrucción de Escenas estáticas Captura de Escenas dinámicas Color Apariencia	Reconstrucción de escenas. Estructuras de datos. Seguimiento. Asociación de datos. Propiedades de los métodos. Velocidad en línea y fuera de línea. Tipo de entrada, una sola vista, múltiples, arreglos. Tratamiento de Deformaciones.	Se propone áreas de trabajo futuro en la reconstrucción 3D de escenas dinámicas y objetos dinámicos (Zollhöfer et al., 2018): Datasets de escenas dinámicas. Oclusiones. Múltiples elementos. deformaciones mayores. No dependencia de la captura de un modelo estático. Una sola cámara RGB-D Apariencia y Color con escenas dinámicas. Uso de aprendizaje profundo.

Continuación

Identificación	Objetivo General	Categorías	Métodos	Resultados
(Aoki et al., 2019)	Registrar una nube de puntos 3D adicional a una ya existente alineando los puntos utilizando una red neuronal recurrente.	Reconstrucción 3D. Registro de puntos. Nube de puntos. ICP <i>iterative closest point</i> .	Reconstrucción de escenas. Estructuras de datos.	Se propone una red neuronal recurrente para el registro de puntos que no requiere del cómputo previo de correspondencias de puntos como los métodos basados en ICP, la red neuronal propuesta puede ser integrada a un flujo de redes neuronales (Aoki et al., 2019).
(Brachmann & Rother, 2019)	Obtener un conjunto de datos libre de inconsistencias <i>outliers</i> por medio de una red neuronal.	Estadística. Limpieza de datos. Preprocesamiento de datos.	RANSAC <i>Random sample consensus</i> . Redes Neuronales. Correspondencias de patrones.	Se propone un modelo neuronal como extensión al método RANSAC aplicable a tareas de visión por computadora como búsqueda de correspondencias (Brachmann & Rother, 2019).
(Aliev et al., 2019)	Obtener un modelo basado en puntos y metadatos que represente una visualización fotorealista por medio de una red neuronal.	Reconstrucción 3D. Estructuras de datos. Nubes de puntos 3D. Redes Neuronales. Apariencia.	Redes Convolucionales Renderizado con Redes Neuronales	Se propone un modelo neuronal que registra una nube de puntos 3D con una apariencia fotorealista que permite calcular nuevas vistas desde distintos ángulos (Aliev et al., 2019).

Continuación

Identificación	Objetivo General	Categorías	Métodos	Resultados
(Bozic et al., 2020)	Reconstruir modelos 3D con redes neuronales de manera semi supervisada.	Reconstrucción 3D Imágenes RGB-D Datos para entrenamiento de redes neuronales Seguimiento de características clave.	Base de datos con etiquetas en los puntos clave para el entrenamiento de redes neuronales.	Se propone una base de datos con imágenes etiquetadas con el propósito de identificar características clave en diferentes secuencias de capturas en el tiempo donde los objetos de interés dentro de las capturas, se van deformando y entre imágenes hay una diferencia de posiciones de esos puntos claves.
(Božič et al., 2020)	Reconstruir modelos 3D con redes neuronales siguiendo puntos clave en la superficie de los objetos a reconstruir.	Reconstrucción 3D Imágenes RGB-D Modelos para seguimiento de características clave.	Modelo matemático para el seguimiento de puntos clave.	Se propone un modelo matemático para el seguimiento de puntos clave que facilita al modelo neuronal realizar la correspondencia entre los puntos clave de una imagen a otra.
(Mildenhall et al., 2021)	Representación de una escena 3D por medio de campos de radiación y la reconstrucción de vistas no capturas por medio de una red neuronal MLP.	Reconstrucción 3D Imágenes RGB Campos de radiación.	Modelo matemático para la representación de la escena en 3D y entrenamiento de una red neuronal.	Se propone un modelo matemático para la representación de una escena en 3D utilizando campos de radiación e interpolado de la escena por medio de una red neuronal MLP por medio de la cual se pueden crear nuevas proyecciones desde puntos de vista no capturados.

Continuación

Identificación	Objetivo General	Categorías	Métodos	Resultados
(Park et al., 2021)	Representación por medio del uso de campos de radiación de objetos dinámicos deformables.	Reconstrucción 3D Imágenes RGB-D Representacion de campos de radiacion de objetos dinámicos.	Modelo de deformación para facilitar la representación por medio de campos deformables	Se propone modelo para el seguimiento de deformaciones en un rostro humano para poderlo representar utilizando campos de radiación.
(Lin et al., 2022)	Uso de multiples cámaras para la creación de campos de radiación de objetos dinámicos deformables.	Reconstrucción 3D Imágenes RGB-D Representacion de campos de radiacion de objetos dinámicos Multiples cámaras	Representación por medio de campos deformables de datos capturados por múltiples cámaras.	Se propone un método para representar escenas dinamicas capturadas por múltiples cámaras al mismo tiempo utilizando campos de radiación.
(Gao et al., 2022a)	Revisión del estado del arte respecto a la representación de escenas dinamicas con campos de radiación.	Reconstrucción 3D Imágenes RGB-D Representacion de campos de radiacion de objetos dinámicos Una sola cámara.	Se propone una métrica para medir la calidad de los datos.	Se propone una métrica para comparar la calidad de los datos respecto a las superficies a reconstruir de un objeto dinámico en escena mientras la información es capturada..
(Fridovich-Keil et al., 2023)	Modelo para representar una escena dinamica en el tiempo.	Reconstrucción 3D Imágenes RGB-D Representacion de campos de radiacion de objetos dinámicos Representacion de cambios en la escena el el tiempo.	Se propone un modelo matemático para reprsentar una escena y sus cambios en el tiempo.	Se propone un método matemático que utiliza 5 dimensiones para representar cambios en una escena en el tiempo y en el espacio propio de la escena capturada.

1.6. Reconstrucción 3D a partir de imágenes RGB-D.

La disponibilidad a bajo precio de dispositivos RGB-D que permiten la captura de datos en 3D ha permitido el avance en el desarrollo de soluciones para el consumidor promedio de dispositivos como celulares potenciando el avance en el estado del arte de las siguientes áreas (Zollhöfer et al., 2018):

1.6.1 Color y Apariencia

La apariencia y color juega un importante rol tanto en la reconstrucción 3D en escenas estáticas como en escenas dinámicas, en su mayoría soluciones al problema se han propuesto en el espacio de la imagen RGB y recientemente empiezan a surgir propuestas para aprovechar la información de profundidad y realizar mejoras importantes de apariencia en los modelos generados (Zollhöfer et al., 2018).



Figura 1.21: Colormap Optimization (Zhou & Koltun, 2014)

En la Figura 1.21 se observa el método de optimización de mapas de color *Colormap Optimization* el cuál utiliza tanto la información RGB desde varias puntos de referencia de la cámara como la información de profundidad y mediante la optimización de una función energética se genera una textura de mejor calidad a menor energía en la función (Zhou & Koltun, 2014).



Figura 1.22: Patch Based Optimization (Bi et al., 2017)

El método basado en la optimización de parches Figura 1.22 logra eliminar defectos en las imágenes por objetos que en distintos puntos de referencia ocasionan artefactos de error en las imágenes (Bi et al., 2017).

Los métodos anteriores producen excelentes resultados con superficies regulares o continuas, sin embargo para superficies irregulares o estructuras como el cabello también existen alternativas como en la Figura 1.23 donde se observa como se reconstruye el cabello de una cabeza humana utilizando estructuras de líneas para aproximar la información RGB de múltiples vistas del mismo objeto (Nam et al., 2019).



Reconstrucción 3D de cabello

Figura 1.23: Strand-Accurate Multi-View Hair Capture (Nam et al., 2019)

1.6.2 Escenas estáticas

Una cámara RGB-D recorre una escena donde no existe movimiento y su entorno esta compuesto de objetos estáticos, la metodología de reconstrucción más común esta basada en el registro de la información 3D proporcionada por el sensor y mediante el seguimiento de la posición de la cámara se estima el registro y fusión de la información 3D adicional proporcionada subsecuentemente en cada cuadro considerado. Como salida se recupera el modelo de un objeto o una escena (Zollhöfer et al., 2018), un ejemplo de la captura de un objeto se muestra en la Figura 1.24 por el método Kinect Fusion (Newcombe et al., 2011).

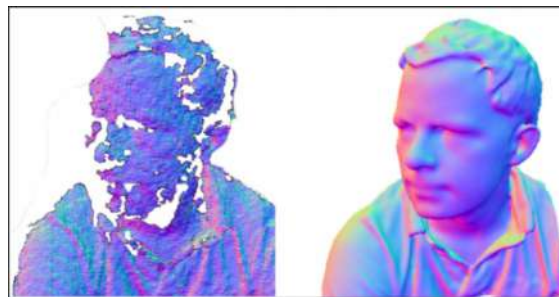


Figura 1.24: KinectFusion (Newcombe et al., 2011)

1.6.3 Escenas dinámicas

Sin embargo es muy común encontrar escenas dinámicas donde objetos dinámicos con superficies deformables son de importancia en la escena misma como personas en movimiento, objetos deformables, objetos interactuando entre ellos y una combinación de

objetos ó características de objetos estáticos y dinámicos. Diversos campos requieren de los ambientes dinámicos como lo son la medicina, la creación de contenido digital, animación por computadora, interacción hombre máquina entre otras (Zollhöfer et al., 2018)

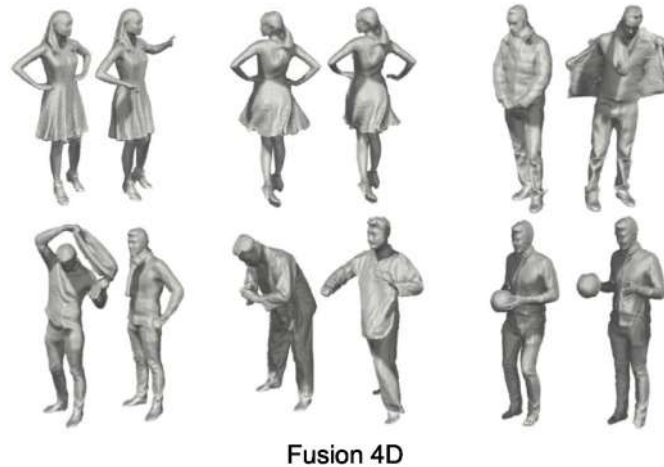


Figura 1.25: Fusion4D (Dou et al., 2016)

Se observa el método Fusion4D Figura 1.25 el cuál utiliza múltiples vistas del objeto capturado para seguir las deformaciones del mismo (Dou et al., 2016).



Figura 1.26: Holoportation (Orts-Escolano et al., 2016)

El método Holoportation (Orts-Escolano et al., 2016) está basado en Fusion4d (Dou et al., 2016) y utiliza una configuración de cámaras RGB-D para rodear al o los objetos a reconstruir, con el objeto de mantener actualizado un modelo 3D que refleje los cambios dinámicos de las formas de los objetos Figura 1.26.

1.7. Justificación

1.7.1 Importancia del tema

Debido a la miniaturización y comercialización de dispositivos móviles, las cámaras RGB-D se encuentran cada vez con mayor disponibilidad y menor costo en el mercado, como lo es el caso de la cámara TrueDepth Figura 1.15 y Realsense Figura 1.14, el beneficio y alcance en la población en general al utilizar las cámaras RGB-D para reconstrucción 3D representa un importante impacto en la generación de modelos 3D en el usuario final, el uso de los datos proporcionados por este tipo de cámaras puede generar beneficios de manera más general al no requerir de un hardware dedicado o de un estudio especializado.

Las escenas dinámicas con objetos con superficies deformables son abundantes de manera natural y de gran utilidad en gran número de aplicaciones. Los modelos de puntos 3D que se persigue generar por medio de la información arrojada por las cámaras RGB-D pueden ser utilizados en diversas aplicaciones que van desde aplicaciones en el área de la salud, al poder por ejemplo generar una férula de una manera más ágil a un método tradicional por medio de la generación de un modelo 3D de manera no invasiva y sin contacto directo con el paciente y sin necesidad de instalaciones especializadas, en robótica y navegación al reconocer por medio de los modelos 3D escenas con obstáculos a evadir y/o mapeado de objetos de interés para poder ser manipulados por ejemplo con algún brazo robótico. En la industria se puede aplicar para reconocer geometrias y descartar defectos en algunas líneas de producción, hasta en la industria del entretenimiento y la generación de contenido digital 3D, el proyecto de investigación puede ser fácilmente adaptado en procesos que requieren de modelos 3D como entrada.

Si bien una cámara RGB-D provee un flujo de información de imágenes de profundidad y color con restricciones variadas dependiendo de la marca y modelo de la cámara disponible, la reconstrucción 3D de una escena flexible y dinámica a partir de la información entregada por este tipo de cámaras aún representa un área de oportunidad de investigación debido a las variadas condiciones en las escenas capturadas y a las características de la información RGB-D proporcionada por este tipo de tecnologías.

Los esfuerzos en el estado del arte varían desde métodos que requieren constelaciones de cámaras especiales RGB-D que se encuentran solamente en estudios profesionales y/o en entornos controlados y que por lo regular requieren una inversión fuera del alcance del rango de aplicaciones necesarias hoy en día, los métodos encontrados en el estado del arte aún tienen diversas limitantes como considerar la generación de una plantilla capturada a partir de un objeto estático con superficie sin deformaciones de manera previa y/o establecer un entorno controlado de iluminación y considerar solo cierto tipo de materiales en los objetos a capturar.

Se pretende mediante el presente proyecto continuar activamente la investigación para expandir el estado de arte en los métodos que utilizan información RGB-D para la reconstrucción 3D de objetos con superficies no rígidas e incorporar redes neuronales y algoritmos de inteligencia artificial con el objetivo de generar mejores resultados.

1.8. Descripción del problema

1.8.1 Reconstrucción 3D de objetos dinámicos con superficies deformables mediante datos RGB-D

El flujo de información generado por uno o varios sensores RGB-D, profundidad y color de una escena, proporcionan una secuencia temporal donde ninguno, uno, muchos objetos y/o la escena completa pueden ser estáticos y/o dinámicos, el foco del presente trabajo es sobre objetos dinámicos con superficies deformables.



Objeto dinámico (muñeco minion).

Figura 1.27: Objeto dinámico vista de una cámara RGB-D (Innmann et al., 2016)

Se observa una secuencia de imágenes Figura 1.27 donde se visualiza un objeto dinámico cuya superficie se va deformando conforme pasan los cuadros y el tiempo en las diferentes tomas, el resultado esperado es la reconstrucción 3D de dicho objeto no importando que su geometría en su superficie no se conservó estática durante la captura de información, el método usado es *VolumeDeform* (Innmann et al., 2016).

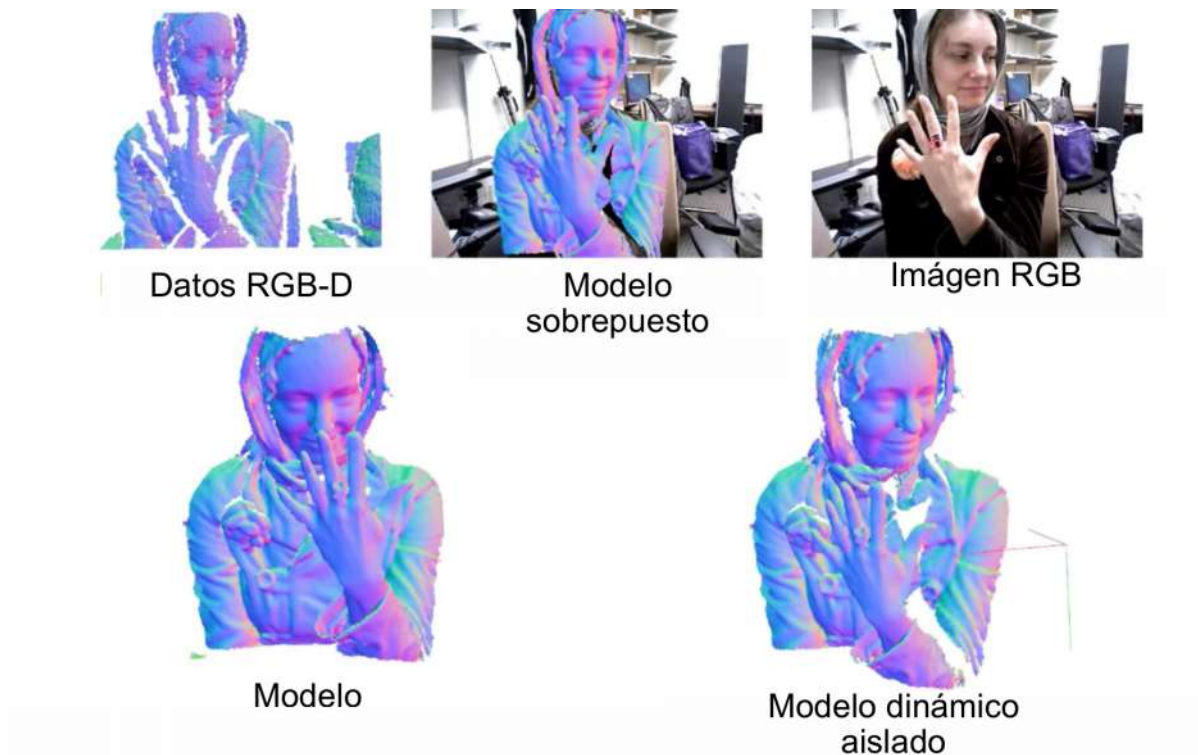


Figura 1.28: Objeto dinámico reconstruido una sola cámara RGB-D (Newcombe et al., 2015)

Se puede apreciar mayor detalle Figura 1.28 de la naturaleza del problema, en base a la información RGB-D generada de manera secuencial en el tiempo, esta es utilizada para generar un modelo 3D de la superficie del objeto dinámico en escena, en este caso un busto humano donde se visualiza el movimiento y deformación de la superficie de la mano, por un lado se actualiza un modelo estático llamado modelo canónico y por otra parte se actualiza un modelo dinámico cuya superficie esta en constante cambio y que es al que se da seguimiento del movimiento, el método utilizado es *DinamicFusion* (Newcombe et al., 2015).

En el estado del arte encontrado se observan como áreas de oportunidad los siguientes puntos:

- Capacidad para manejar mayores deformaciones en las superficies de los objetos dinámicos.
- Mejora en la apariencia en la reconstrucción del modelo 3D respecto a los objetos originales, eliminar defectos debido a los cambios de geometría..
- Mejor manejo y reconstrucción de múltiples objetos dinámicos.
- Capacidad para manejar la interacción de varios objetos dinámicos mientras son reconstruidos.
- Uso de un solo sensor RGB-D y sin arreglos fijos o constelaciones de sensores.
- Uso de la información RGB en conjunto con la información de profundidad.

El seguimiento de puntos clave en objetos dinámicos con superficies deformables representa el principal objeto de estudio y se identifica que a partir de las posibles soluciones a este problema se pueden desencadenar propuestas para mejorar resultados de uno a varios de los puntos anteriores.

2. MARCO TEÓRICO

2.1. Fundamentación Teórica

2.1.1 Coordenadas homogeneas

Las coordenadas homogeneas son un sistema de coordenadas usadas en geometría proyectiva. El infinito se puede representar por un valor finito, rotaciones y traslaciones pueden ser representadas por una operación de matrices, las coordenadas homogeneas pueden ser usadas para representar espacios de dimensiones arbitrarias.

En el caso de una cámara proyectando una imagen, la última coordenada puede representar la distancia entre una imagen siendo proyectada y la cámara que la proyecta. Si la distancia es grande, la imagen será grande también, si la distancia es pequeña entonces la imagen proyectada también será pequeña.

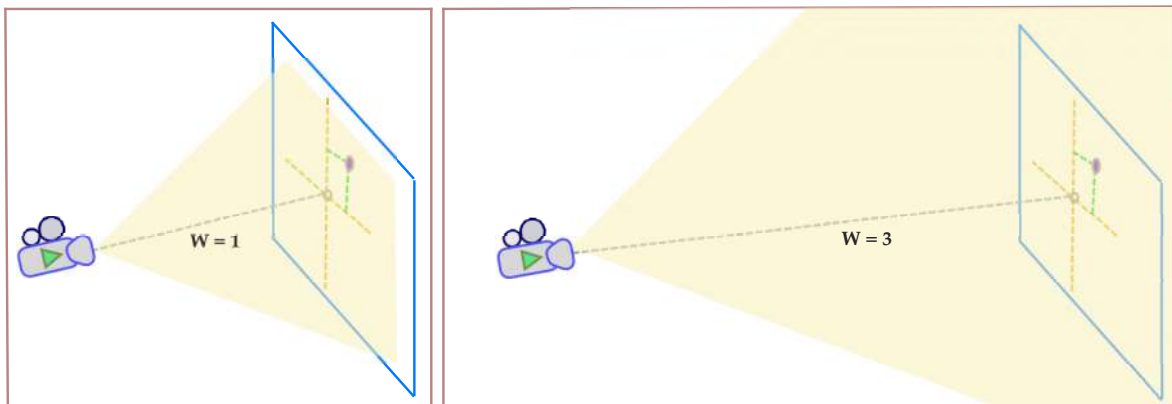


Figura 2.1: Imagen basada en (“Explaining Homogeneous Coordinates and Projective Geometry”, 1970).

Para convertir de coordenadas cartesianas x, y que representan coordenadas en la imagen a coordenadas homogeneas X, Y que representan coordenadas en el mundo 3D, se requiere un número extra k como se muestra en las Ecuaciones 2.1, 2.2, 2.3, 2.4:

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.1)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.2)$$

Usualmente se agrega el número 1 y se multiplica por un número real:

$$\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} k \cdot x \\ k \cdot y \\ k \end{bmatrix} \quad (2.3)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} k \cdot X \\ k \cdot Y \\ k \cdot Z \\ k \end{bmatrix} \quad (2.4)$$

2.1.2 Coordenadas homogéneas a no-homogéneas

Para convertir coordenadas homogéneas a coordenadas no-homogéneas un último componente w y W es usado para dividir todos los demás y posteriormente se descarta.

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \rightarrow \begin{bmatrix} x/w \\ y/w \end{bmatrix} \quad (2.5)$$

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \rightarrow \begin{bmatrix} X/W \\ Y/W \\ Z/W \end{bmatrix} \quad (2.6)$$

2.2. Modelo de proyección de cámara

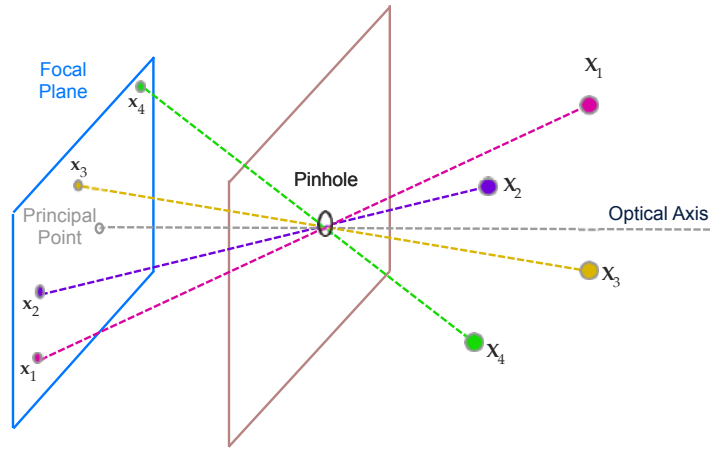
Cada uno de los píxeles en una imagen $\mathbf{x}_i^j = [x_i^j \ y_i^j]^\top$ donde j corresponde a el j -ésimo cuadro tomado por la cámara e i corresponde a el i -ésimo punto en las coordenadas del mundo $\mathbf{X}_i = [X_i \ Y_i \ Z_i]^\top$, el modelo de proyección de cámara convierte de coordenadas en el mundo 3D a coordenadas de una imagen en 2D:

$$\mathbf{x}_i^j = \mathcal{F}^j(\mathbf{X}_i) \quad (2.7)$$

Para aplicar el modelo de transformación proyectiva además del modelo de cámara estenopéica (*pinhole camera*), se requieren los parámetros extrínsecos y de distorsión de la cámara. *distortion of points are needed.*

2.3. Parámetros intrínsecos de la cámara

Los parámetros intrínsecos describen la relación entre el sistema de coordenadas del mundo y el sistema de coordenadas de una imagen. El modelo matemático utiliza cinco parámetros, longitud focal (*focal length*) en las direcciones x y y , punto principal (*principal*



Modelo de cámara "Pinhole"

Figura 2.2: Modelo de proyección de cámara.

point) en las direcciones x y y , y el valor de inclinación (*skew*) entre la dirección x y y (Conrady, 1919), (Duane, 1971).

2.3.1 Longitud focal

En el modelo de cámara estenopéica, la longitud focal f es la distancia entre el agujero y el plano focal a lo largo del eje óptico. Los planos focales x , y pueden tener diferentes longitudes focales, la longitud focal del eje y puede ser modificada por α :

$$f_y = \alpha \cdot f \quad (2.8)$$

$f_x = f_y(\alpha = 1)$ es valida en un modelo de cámara estenopéica ideal, sin embargo las distorsiones en los lentes y los defectos de manufactura afectan. La interpretación de longitudes focales no idénticas es entendida como la forma no cuadrada de los pixeles.

2.3.2 Centro óptico y punto principal

El centro óptico Centro Óptico (Pinhole) esta ubicado en el origen del sistema de coordenadas del mundo 3D y el plano de la imagen, en donde la imagen virtual es formada, se desplaza por medio del eje óptico. La coordenada de la intersección del eje óptico con el plano de la imagen es llamado punto principal.

2.3.3 Inclinación (*skew*)

Un factor de inclinación s distinto a cero, implica que los ejes x y y de la cámara no son perpendiculares entre ellos.

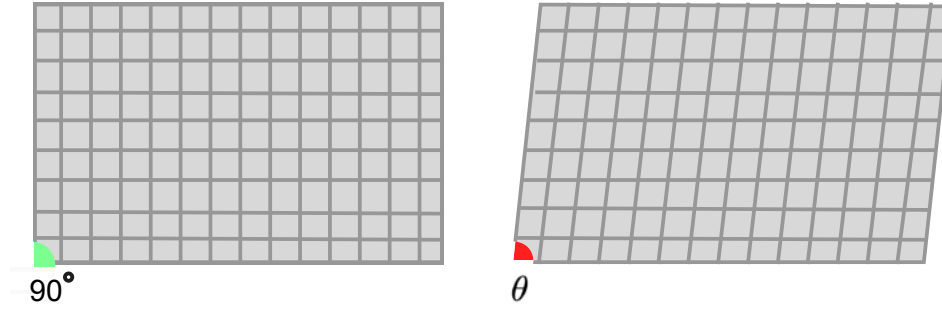


Figura 2.3: Parámetro de inclinación

Una posibilidad para tener un valor de parámetro de inclinación es el sensor montado de manera perpendicular al eje óptico (Szeliski, 2010).

2.4. Matriz de parámetros intrínsecos

La matriz de parámetros intrínsecos definida por \mathbf{K} es una matriz triangular superior usada para transformar coordenadas del mundo a coordenadas homogéneas de una imagen en 2D. Existen 2 formas generales equivalentes de la matriz de parámetros intrínsecos, donde pp_x y pp_y definen las coordenadas del punto principal, algunas veces se pueden encontrar como c_x y c_y (Tordoff & Murray, 2004).

$$\mathbf{K} = \begin{bmatrix} f & s & pp_x \\ 0 & f \cdot \alpha & pp_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

$$\mathbf{K} = \begin{bmatrix} f_x & s & pp_x \\ 0 & f_y & pp_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

2.4.1 Coordenadas del mundo 3D a coordenadas homogéneas 2D

Para obtener coordenadas homogéneas en 2D a partir de coordenadas del mundo en 3D se aplica la siguiente Ecuación:

$$\begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad (2.12)$$

El cálculo se describe a continuación:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.13)$$

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x \cdot X + c_x \cdot Z \\ f_y \cdot Y + c_y \cdot Z \\ Z \end{bmatrix} \quad (2.14)$$

Entonces se convierte a coordenadas no-homogeneas x', y' :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{f_x \cdot X + c_x \cdot Z}{Z} \\ \frac{f_y \cdot Y + c_y \cdot Z}{Z} \end{bmatrix} = \begin{bmatrix} f_x \cdot \frac{X}{Z} + c_x \\ f_y \cdot \frac{Y}{Z} + c_y \end{bmatrix} \quad (2.15)$$

2.4.2 Tamaño de pixel (*pixel pitch*)

El tamaño del pixel es la distancia en milímetros del centro del pixel al centro del pixel adyacente, a menor tamaño de pixel mayor densidad de pixeles. Para convertir pixeles a milímetros se utiliza la siguiente Ecuación:

$$f[\text{mm}] = f[\text{pixels}] \cdot p \left[\frac{\mu\text{m}}{\text{pixel}} \right] \cdot \frac{1[\text{mm}]}{1000[\mu\text{m}]} \quad (2.16)$$

2.4.3 Matriz inversa de parámetros intrínsecos

Para transformar puntos sin distorsión en una imagen la matriz inversa de parámetros intrínsecos es utilizada:

$$\mathbf{K}^{-1} = \frac{1}{f^2 \cdot \alpha} \begin{bmatrix} f \cdot \alpha & -s & cy \cdot s - cx \cdot f \cdot \alpha \\ 0 & f & -cy \cdot f \\ 0 & 0 & f^2 \cdot \alpha \end{bmatrix} = \frac{1}{f_x \cdot f_y} \begin{bmatrix} f_y & -s & cy \cdot s - cx \cdot f_y \\ 0 & f_x & -cy \cdot f_x \\ 0 & 0 & f_x \cdot f_y \end{bmatrix} \quad (2.17)$$

Cuando el valor de inclinación s es igual a 0:

$$\mathbf{K}^{-1} = \frac{1}{f_x \cdot f_y} \begin{bmatrix} f_y & 0 & -cx \cdot f_y \\ 0 & f_x & -cy \cdot f_x \\ 0 & 0 & f_x \cdot f_y \end{bmatrix} = \frac{1}{f^2 \cdot \alpha} \begin{bmatrix} f \cdot \alpha & 0 & -cx \cdot f \cdot \alpha \\ 0 & f & -cy \cdot f \\ 0 & 0 & f^2 \cdot \alpha \end{bmatrix} \quad (2.18)$$

Cuando α es igual a 1:

$$\mathbf{K}^{-1} = \frac{1}{f^2} \begin{bmatrix} f & -s & cy \cdot s - cx \cdot f \\ 0 & f & -cy \cdot f \\ 0 & 0 & f^2 \end{bmatrix} \quad (2.19)$$

Cuando α es igual a 1 y el valor de inclinación s es igual 0:

$$\mathbf{K}^{-1} = \frac{1}{f} \begin{bmatrix} 1 & 0 & -cx \\ 0 & 1 & -cy \\ 0 & 0 & f \end{bmatrix} \quad (2.20)$$

2.5. Modelos de distorsión

La calibración geométrica de una cámara requiere de modelos de distorsión que describan las desviaciones de una cámara real que utilice lentes y difiera del modelo de cámara estenopéica ideal.

2.5.1 Modelo de distorsión radial

Los modelos de distorsión radial convierten entre los puntos distorsionados a los puntos ajustados usando el centro de distorsión (x_c, y_c) y una función $f(r)$:

$$x' = f(r) \cdot \cos(\theta) + x_c \quad (2.21)$$

$$y' = f(r) \cdot \sin(\theta) + y_c \quad (2.22)$$

mediante:

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad (2.23)$$

donde r representa la distancia del punto distorsionado al centro de distorsión.

2.5.2 Distorsión radial de un lente

(X_d, Y_d) son las coordenadas distorsionadas en un plano de imagen, (X_u, Y_u) son las coordenadas reales.

$$X_d + D_x = X_u \quad (2.24)$$

$$Y_d + D_y = Y_u \quad (2.25)$$

$$D_x = X_d (\kappa_1 r^2 + \kappa_2 r^4 + \dots)$$

$$D_y = Y_d (\kappa_1 r^2 + \kappa_2 r^4 + \dots) \quad (2.26)$$

$$r = \sqrt{X_d^2 + Y_d^2}$$

D_x, D_y son la distorsión en el componente x y en el componente y Los coeficientes de distorsión se muestran como k_i en las Ecuaciones eq:RLDX a 2.26.

2.5.3 Cálculo iterativo de la distorsión radial inversa

En el cálculo iterativo de la distorsión radial inversa intervienen la coordenada en el mundo real X con componentes

$$\begin{aligned}
X &= (X_i, Y_i) \\
P &= (p_x, p_y) \\
\hat{D} &= (D_x, D_y) \\
z &= \mathbf{K}^{-1} \hat{D}_x
\end{aligned} \tag{2.27}$$

Iterar hasta converger:

$$\begin{aligned}
r &= \|z\|^2 \\
m &= \kappa_1 r^2 + \kappa_2 r^4 \\
z &= \frac{\mathbf{K}^{-1} \hat{D}_x}{1+m}
\end{aligned} \tag{2.28}$$

Entonces:

$$X = \frac{z + Pm}{1 + m} \tag{2.29}$$

2.5.4 Fórmula exacta para el calculo de la distorsión radial inversa

!!! DETALLAR VARIABLES y CITAR

$$x' = x + \bar{x} (k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) + [p_1 (r^2 + 2\bar{x}^2) + 2p_2 \bar{x} \bar{y}] (1 + p_3 r^2 + \dots) \tag{2.30}$$

$$y' = y + \underbrace{\bar{y} (k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots)}_{\text{distorsión Radial}} + \underbrace{[p_2 (r^2 + 2\bar{y}^2) + 2p_1 \bar{x} \bar{y}]}_{\text{Distorsión Tangencial}} (1 + p_3 r^2 + \dots) \tag{2.31}$$

Donde $\bar{x} = x - x_0, \bar{y} = y - y_0, r = \sqrt{\bar{x}^2 + \bar{y}^2}$. La distorsión tangencial no se considera (Drap & Lefèvre, 2016).

mediante $a_0 = 1, a_1 = k_1, \dots, a_n = k_n$ y una secuencia dada a_1, \dots, a_4 la siguiente relación se obtiene:

$$b_0 = 1 \tag{2.32}$$

y para $n \geq 0$:

$$b_n = - \sum_{k=1}^4 a_k q(n-k) - \sum_{\substack{j+k=n \\ 0 \leq k \\ 1 \leq j \leq 8k}} b_k p(j, 2k) \tag{2.33}$$

donde se usan los siguientes coeficientes intermedios:

$$p(j, k) = \sum_{\substack{n_1 + \dots + n_k = j \\ 0 \leq n_i \leq 4}} a_{n_1} \dots a_{n_k} \quad (2.34)$$

$$q(k) = - \sum_{j=1}^4 a_j q(k-j) \quad (2.35)$$

2.5.5 Model de distorsión radial polinomial

El modelo de distorsión radial polinomial utiliza el polinomio:

$$f(r) = r \cdot (1 + p_1 \cdot r + p_2 \cdot r^2 + \dots + p_N \cdot r^N) = r \cdot \left(1 + \sum_{n=1}^N p_n \cdot r^n \right) \quad (2.36)$$

basado en el modelo de cámara estenopéica y corrección de distorsión en lente de (Tsai, 1987):

2.5.6 Coordenadas del mundo 3D a coordenadas de imagen.

La Ecuación 2.37 representa la transformación rígida desde el sistema de coordenadas para el mundo en 3D donde R es una matrix de rotación de 3×3 , Ecuación 2.38 y T es el vector de traslación eq. 2.39, R y T son parámetros a ser calibrados.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \quad (2.37)$$

$$R \equiv \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (2.38)$$

$$T \equiv \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (2.39)$$

2.5.7 Transformación de coordenadas del mundo 3D a coordenadas de imagen ideales (sin distorsión)

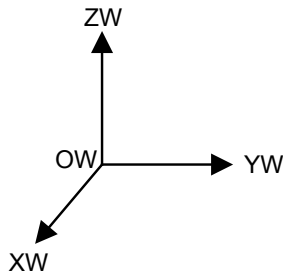
Usando la proyección de perspectiva con un modelo de geometria de cámara estenopéica, el parametro a ser calibrado es la longitud focal efectiva f

$$X_u = f \frac{x_c}{z_c} \quad (2.40)$$

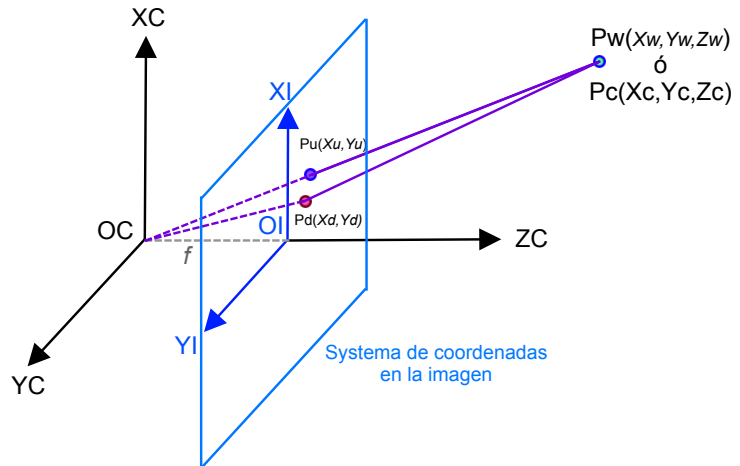
$$Y_u = f \frac{y_c}{z_c} \quad (2.41)$$

2.5.8 Modelo de cámara estenopéica (*Pinhole camera model*)

Coordenadas del mundo en 3D



Coordenadas de la imagen en la cámara



Geometría de la cámara mediante perspectiva de proyección y distorsión radial de lente

Figura 2.4: Modelo de cámara estenopéica (Tsai, 1987)

El Pinhole camera model por sus siglas en inglés, modelo de cámara estenopéica (Pinhole-Camera-Model) que se muestra en la Figura 2.4 está basado en (Tsai, 1987), donde (x_w, y_w, z_w) son las coordenadas en 3D del punto P_w en el sistema de coordenadas del mundo 3D, (x_c, y_c, z_c) son las coordenadas en 3D del punto P_c en el sistema de coordenadas 3D de la cámara. El sistema de coordenadas 3D de la cámara, esta centrado en el centro óptico OC cuyo eje ZC es el mismo que el eje óptico, (XI, YI) es el centro de las coordenadas de la imagen en OI , donde el plano de la imagen intersecta con el eje ZC . YI es paralela a YC y XI es paralela a XC , f es la distancia entre el plano de imagen y el centro óptico OC (X_u, Y_u) es el sistema de coordenadas de imagen de (x_w, y_w, z_w) si un modelo de cámara estenopéica ideal es usado, (X_d, Y_d) es la coordenada de P debido a la distorsión del lente. La unidad (X_f, Y_f) en el sistema de coordenadas usado en la computadora, es el número de pixeles en la imagen discreta adicional a los parametros que requieren ser incluidos y a la calibración que relaciona el sistema de coordenadas de la imagen en el frente del plano al sistema de coordenadas de imagen en la computadora (Tsai, 1987).

2.6. Modelo general de una cámara de tipo RGB-D

Sea $\{D\}$ un punto 3D general en el sistema de coordenadas de una cámara en la capa de profundidad ${}^D\mathbf{X} = [x^D, y^D, z^D]^T$ y la proyección de punto en el sistema de coordenadas cartesianas de la imagen de color $\{R\}$, sea el plano de la imagen ${}^R\mathbf{p} = [u^R, v^R]^T$. Las coordenadas de la imagen son obtenidas en el siguiente proceso de tres pasos (Liu et al., 2020):

Los puntos en 3D ${}^D\mathbf{X}$ en el sistema coordenado $\{D\}$ son transformados primero en el sistema coordenado $\{R\}$ por medio de la Ecuación refeq:toR:

$${}^D\mathbf{R} = {}^R\mathbf{T}^D\mathbf{X} \quad (2.42)$$

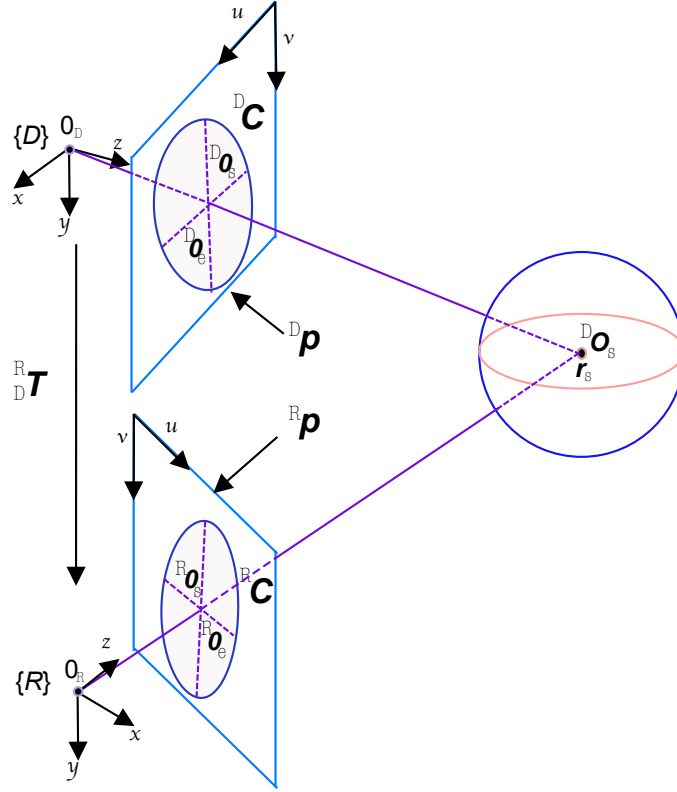


Figura 2.5: Modelo de cámara de profundidad basado en (Liu et al., 2020)

En el modelo general de la cámara de profundidad que se muestra en la Figura 2.5, $({}^D\mathbf{o}_e, {}^R\mathbf{o}_e)$ es el centro de la elipse y $({}^D\mathbf{o}_s, {}^R\mathbf{o}_s)$ es el centro de proyección de la esfera, $\{R\}$ el sistema de coordenadas de la imagen a color, $\{D\}$ el sistema de coordenadas de profundidad, ${}^R_D\mathbf{R} = ({}^R_D\mathbf{R}, {}^R_D\mathbf{T})$ la matriz de rotación y traslación entre los dos sistemas coordenados, ${}^D\mathbf{X}$ el punto 3D en el sistema coordenado de la cámara de profundidad, ${}^R\mathbf{C}$, ${}^D\mathbf{C}$ representan la proyección de la elipse de la esfera en un sistema coordenado de dos dimensiones, ${}^R\mathbf{P}$, ${}^D\mathbf{P}$ los puntos de la esfera proyectados en los planos de dos imágenes, ${}^D\mathbf{o}_s$ y \mathbf{r}_s el centro y radio de la esfera respectivamente.

2.6.1 Transformación de coordenadas de la capa de profundidad a coordenadas del mundo en 3D (sin distorsión)

La búsqueda de información específica en la capa de profundidad requiere que los puntos en 3D almacenados sean convertidos mediante una proyección desde coordenadas en 2D a coordenadas en 3D. Las cámaras de tipo RGB-D usualmente contienen un parámetro de conversión llamado unidad de profundidad (*depth unit*) que transforma valores almacenados en la capa de profundidad a valores expresados en unidades de longitud del Sistema Internacional de Unidades (SI), metros **m**). En la cámara utilizada en el presente proyecto el valor de la unidad de profundidad es de 0.001 y la información de la capa de profundidad se almacena de manera binaria utilizando enteros de 16-bit. Se utiliza la matriz de intrínsecos para proyectar los puntos en 2D a puntos en 3D como sigue:

1. Convertir de números enteros contenidos en el archivo de la imagen I a información

de profundidad en metros Z requiere multiplicar cada valor entero en la imagen de profundidad por la unidad de profundidad Ecuación 2.43 :

$$Z(I_{ij})_{ij} = I_{ij} \times 0,001 \quad (2.43)$$

2. Proyectar de coordenadas en 2D a coordenadas en el mundo en 3D (sin distorsión, considerando que ya se ha corregido con anterioridad) como en la Ecuación 2.44 :

$$\begin{aligned} X(I_{ij})_{ij} &= (i - c_x)/f_x \times 0,001 \\ Y(I_{ij})_{ij} &= (i - c_y)/f_y \times 0,001 \end{aligned} \quad (2.44)$$

3. Finalmente la nube de puntos 3D se representa como coordenadas (X, Y, Z).

2.6.2 Modelo para representar la esfera con un tamaño conocido.

Se utiliza una Ecuación que permite modelar una esfera donde en su superficie encontramos 3 puntos diferentes x_i, y_i, z_i con un centro en común O con coordenadas x_c, y_c, z_c donde:

$$O = (x_c, y_c, z_c) \quad (2.45)$$

$$r^2 = (x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 \quad (2.46)$$

donde r es el radio de la esfera. Expandiendo la Ecuación 2.46 para 3 puntos diferentes:

$$\begin{aligned} r^2 &= (x_1 - x_c)^2 + (y_1 - y_c)^2 + (z_1 - z_c)^2 \\ r^2 &= (x_2 - x_c)^2 + (y_2 - y_c)^2 + (z_2 - z_c)^2 \\ r^2 &= (x_3 - x_c)^2 + (y_3 - y_c)^2 + (z_3 - z_c)^2 \end{aligned} \quad (2.47)$$

entonces en la Ecuación 2.47 cada Ecuación es igual r por lo que igualamos en el siguiente orden:

$$\begin{aligned} (x_1 - x_c)^2 + (y_1 - y_c)^2 + (z_1 - z_c)^2 &= \\ (x_2 - x_c)^2 + (y_2 - y_c)^2 + (z_2 - z_c)^2 &= \\ (x_3 - x_c)^2 + (y_3 - y_c)^2 + (z_3 - z_c)^2 &= \\ (x_1 - x_c)^2 + (y_1 - y_c)^2 + (z_1 - z_c)^2 &= \\ (x_3 - x_c)^2 + (y_3 - y_c)^2 + (z_3 - z_c)^2 &= \\ (x_2 - x_c)^2 + (y_2 - y_c)^2 + (z_2 - z_c)^2 &= \end{aligned} \quad (2.48)$$

Coordenadas de puntos 3D 1 = Coordenadas de puntos 3D 2, Coordenadas de puntos 3D 3 = Coordenadas de puntos 3D 1, Coordenadas de puntos 3D 3 = Coordenadas de puntos 3D 2, y desarrollando el sistema de Ecuaciones 2.48 obtenemos:

$$\begin{aligned}
& 2x_c(x_2 - x_1) + 2y_c(y_2 - y_1) + 2z_c(z_2 - z_1) \\
& \quad + (x_1^2 - x_2^2 + y_1^2 - y_2^2 + z_1^2 - z_2^2) = 0 \\
& 2x_c(x_3 - x_1) + 2y_c(y_3 - y_1) + 2z_c(z_3 - z_1) \\
& \quad + (x_1^2 - x_3^2 + y_1^2 - y_3^2 + z_1^2 - z_3^2) = 0 \\
& 2x_c(x_3 - x_2) + 2y_c(y_3 - y_2) + 2z_c(z_3 - z_2) \\
& \quad + (x_2^2 - x_3^2 + y_2^2 - y_3^2 + z_2^2 - z_3^2) = 0
\end{aligned} \tag{2.49}$$

simplificando la Ecuación 2.49 y reordenando en la forma $Ax = b$ obtenemos:

$$\begin{aligned}
& \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) & 2(z_2 - z_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) & 2(z_3 - z_1) \\ 2(x_3 - x_2) & 2(y_3 - y_2) & 2(z_3 - z_2) \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \\
& = \begin{bmatrix} x_1^2 - x_2^2 + y_1^2 - y_2^2 + z_1^2 - z_2^2 \\ x_1^2 - x_3^2 + y_1^2 - y_3^2 + z_1^2 - z_3^2 \\ x_2^2 - x_3^2 + y_2^2 - y_3^2 + z_2^2 - z_3^2 \end{bmatrix}
\end{aligned} \tag{2.50}$$

la Ecuación 2.50 puede ser resuelta para calcular el radio mediante 3 puntos en la superficie de la esfera.

2.6.3 Puntuación Z.

Para eliminar valores atípicos en el conjunto de puntos que se consideran como parte de la esfera se utiliza la puntuación Z (Salgado et al., 2016), para eliminar valores por debajo de un umbral definido de -1.5 y por arriba de 1.5.

$$z = \sum (X_i - \bar{X})/\sigma \tag{2.51}$$

donde X_i es el valor medido, \bar{X} es el promedio de los valores y σ es la desviación estandar.

2.7. Captura de la escena y objeto a reconstruir

Con el objetivo de reconstruir un modelo en 3D a partir de la captura del objeto por medio de una cámara RGB-D se requiere garantizar la correcta captura de la geometría de dicho objeto considerando que su superficie se encuentra en constante deformación, por lo que se requiere medir la calidad de la captura de la escena y del objeto para poder garantizar la completitud de la información requerida para la reconstrucción de dicho objeto, ya que debido a oclusiones y a la geometría del mismo objeto puede no ser posible realizar una reconstrucción adecuada de dicho objeto. Con este propósito se utiliza en el presente trabajo de investigación la métrica Effective Multi-view factors, por sus siglas en inglés factores de capturas efectivas multi-vista (EMFs) de (Gao et al., 2022b).

Se asume que durante la captura del objeto por medio de la cámara RGB-D:

- El objeto a capturar se mueve a una velocidad aproximadamente constante.
- La cámara permanece fijada en la captura del objeto.
- La distancia de la cámara al objeto permanece aproximadamente constante.

2.7.1 EMF completo

Considerando una toma monocular mediante una cámara RGB-D de una escena dinámica \mathbb{E} en un tiempo determinado \mathcal{T} , en cada toma $t \in \mathcal{T}$ sea \mathbf{O}_t la posición de la cámara en el mundo 3D real. Se considera cada punto \mathbf{x}_t en el dominio de las superficies en la escena observada $\mathbb{S}_t^2 \subset \mathbb{R}^3$, se define el movimiento escena-cámara como el radio relativo esperado ω como en la Ecuación 2.53:

$$\Omega = \mathbb{E}_{t,t+1 \in \mathcal{T}} \left[\mathbb{E}_{\mathbf{x}_t \in \mathbb{S}_t^2} \left[\frac{\|\mathbf{o}_{t+1} - \mathbf{o}_t\|}{\|\mathbf{x}_{t+1} - \mathbf{x}_t\|} \right] \right] \quad (2.52)$$

donde el denominador $\mathbf{x}_{t+1} - \mathbf{x}_t$ denota el flujo de la escena en 3D y el numerador $\mathbf{o}_{t+1} - \mathbf{o}_t$ denota el movimiento de la cámara en 3D en un tiempo t .

El flujo de la escena en 3D se puede estimar mediante el flujo óptico de la escena en 2D y mediante el mapa de profundidad proporcionado por la cámara de tipo RGB-D.

2.7.2 EMF angular ó velocidad angular de la cámara

Se asume que en la captura de la escena la cámara esta dirigida a un punto en específico \mathbf{a} en el espacio de la escena y que esta dirección de captura se conserva indiferentemente del movimiento de la cámara, esta situación es usual cuando se enfoca un objeto y este es rodeado para capturarlo, este punto específico se puede calcular triangulando todos los ejes ópticos de las capturas en una velocidad de captura N , definimos la velocidad angular de la cámara ω en la Ecuación 2.53 como sigue:

$$\omega = \mathbb{E}_{t,t+1 \in \mathcal{T}} \left[\arccos \left(\frac{\langle \mathbf{a} - \mathbf{o}_t, \mathbf{a} - \mathbf{o}_{t+1} \rangle}{\|\mathbf{a} - \mathbf{o}_t\| \cdot \|\mathbf{a} - \mathbf{o}_{t+1}\|} \right) \right] \cdot N \quad (2.53)$$

2.8. Bloque NeRF

2.8.1 Bases teóricas NeRF

NEural Radiance Field ó campos de luminosidad neuronales por sus siglas en inglés (NeRF), en su forma original (Mildenhall et al., 2021), se destina a representar escenas en 3D de una manera aproximada mediante un modelo de redes neuronal, los campos de luminosidad describen una densidad de volumen y color por cada punto y en todas las direcciones dentro de un espacio de una escena, esto se describe mediante la siguiente Ecuación 2.54

$$F(\mathbf{x}, \theta, \phi) \rightarrow (\mathbf{c}, \sigma) \quad (2.54)$$

donde una red Multi Layer Perceptron por sus siglas en inglés (MLP) es representada por F_{Θ} , y θ y ϕ las entradas de la red, $\mathbf{x} = (x, y, z)$ representan coordenadas en la escena a representar, (θ, ϕ) representan el azimuth y perspectiva o vista en ángulos polares pero también se encuentran en la literatura representados mediante un vector unidad en el plano cartesiano 3D mediante $\mathbf{d} = (d_x, d_y, d_z)$, $\mathbf{c} = (r, g, b)$ representa el color en componentes rojo, verde y azul, y la densidad de volumen se respresenta mediante el símbolo σ .

Esta representación neuronal asume una escena estática capturada con múltiples cámaras al mismo tiempo en diferentes direcciones restringiendo la predicción de la densidad de volumen en la escena de manera independiente a la dirección de la vista de la escena donde

el color si depende de la dirección de la vista de la escena. En el modelo NeRF original se consideran 2 fases en el diseño de la red:

- La primera etapa toma como entrada \mathbf{x} y como salida σ y un vector de características dimensionales.
- En la segunda etapa, el vector de características se concatena con la dirección de la vista \mathbf{d} y se pasa para entrar a un segundo MLP cuya salida es \mathbf{c}

Para la síntesis de nuevas vistas (no capturadas por las diferentes cámaras), se realiza la inferencia con el modelo NeRF ya entrenado para cada pixel en la imagen se crean rayos de cámara (líneas imaginarias que pasan por el foco de la cámara, el plano de la imagen y la escena) $C(\mathbf{r})$ y se genera un muestreo de puntos. Por cada muestra de puntos utilizando la dirección del rayo de cámara y ubicación, se extrae el color y densidad y se utiliza el renderizado de volumen como en (Kajiya & Von Herzen, 1984) para producir la imagen a partir de la aproximación del color y densidad mediante la cuadratura de la Ecuación 2.55.

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^N \alpha_i T_i \mathbf{c}_i \quad (2.55)$$

donde \mathbf{T} es la transmisión acumulada que representa la probabilidad de que el rayo viaje sin ser interceptado desde la posición inicial a la posición final en el trazo.

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \quad (2.56)$$

donde N es el número de segmentos iguales en los que se divide el rayo de la cámara de donde se toma la muestra δ_i es la distancia de la muestra i a la muestra $\beta + 1$, α_i representa la transparencia en el punto muestreado i y se obtiene con la Ecuación 2.57

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i) \quad (2.57)$$

La profundidad puede ser aproximada por la acumulación de la transmisión de los rayos de cámara como en la Ecuación 2.58

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^N \alpha_i t_i T_i \quad (2.58)$$

donde t_i es el segmento del rayo calculado. Para cada pixel en la imagen se utiliza una función perdida que considera un error cuadrado fotométrico descrito en la Ecuación 2.59 la cuál se utiliza para el entrenamiento de la red neuronal NeRF:

$$L = \sum_{\mathbf{r} \in R} \left\| \hat{C}(\mathbf{r}) - C_{gt}(\mathbf{r}) \right\|_2^2 \quad (2.59)$$

donde $C_{gt}(\mathbf{r})$ es el valor verdadero de referencia contenida en los pixeles de las imágenes del conjunto de datos para entrenamiento asociado al rayo \mathbf{r} y R es el conjunto de rayos necesarios para sintetizar la imagen.

NeRF originalmente emplea codificación posicional γ para mejorar la reconstrucción del detalle fino como se muestra en la Ecuación 2.60

$$\gamma(v) = (\sin(2^0\pi v), \cos(2^0\pi v), \dots, \sin(2^{N-1}\pi v), \cos(2^{N-1}\pi v)) \quad (2.60)$$

donde N es la dimensionalidad del codificado, originalmente $N = 10$ para el vector \mathbf{x} y $N = 4$ para el vector \mathbf{d} como en (Mildenhall et al., 2021).

2.9. Detección y Seguimiento de puntos clave de objetos dinámicos con superficies deformables

La detección de características ó puntos clave en los objetos dinámicos es fundamental para poder seguir las deformaciones de sus superficies y registrar adecuadamente en un mapa 3D ya sea en un nuevo modelo 3D o en un modelo existente adicionar la nueva información que se va adquiriendo conforme se descubren secciones no visibles anteriormente con conectividad a la frontera o periferia registrada en los modelos 3D canónicos reconstruidos (Innmann et al., 2016) y (Newcombe et al., 2015) Figura 1.28.

Se busca diseñar un método de reconstrucción 3D robusto a deformaciones en las superficies de los objetos dinámicos, a oclusiones, a interacciones de los objetos del modelo 3D.

2.9.1 Código latente de apariencia

Para obtener una mayor resiliencia del bloque NERF a cambios de iluminación cada imagen en el conjunto de datos generado se asocia a un vector de valores reales o código latente ℓ_i^a de longitud n^a reemplazando c_i por $c_i(t)$ como se describe en (Martin-Brualla et al., 2021).

2.9.2 Código de deformación

Se codifica una transformación rígida de un eje de rotación $S = (\mathbf{f}; \mathbf{v}) \in \mathbb{R}^6$ donde \mathbf{r} denota una rotación y \mathbf{v} el origen fijo de la rotación. como se muestra en la Ecuación 2.61

$$e^{[\mathbf{r}]} = \mathbf{I} + \frac{\sin \theta}{\theta} [\mathbf{r}]_x + \frac{1 - \cos \theta}{\theta^2} [\mathbf{r}]_x^2 \quad (2.61)$$

donde $[\mathbf{x}]_x$ es una matriz simétrica sesgada, también conocida como el producto punto de un vector \mathbf{x} de tamaño 3 como en la Ecuación 2.62

$$[\mathbf{x}]_x = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \quad (2.62)$$

La traslación codificada por S puede recuperarse con $\mathbf{p} = \mathbf{G}\mathbf{v}$ con \mathbf{G} representado con la Ecuación 2.63

$$\mathbf{G} = \mathbf{I} + \frac{1 - \cos \theta}{\theta^2} [\mathbf{r}]_x + \frac{\theta - \sin \theta}{\theta^3} [\mathbf{r}]_x^2 \quad (2.63)$$

El exponencial de S puede ser expresado en una matriz homogénea de la forma $e^S \in \mathbf{SE}(3)$ como en la Ecuación 2.64

$$e^S = \begin{pmatrix} e^r & \mathbf{p} \\ 0 & 1 \end{pmatrix} \quad (2.64)$$

donde el punto deformado $\mathbf{x}' = e^S \mathbf{x} = e^r \mathbf{x} + \mathbf{p}$, esta codificación de deformación esta basada en la transformación rígida de (Lynch & Park, 2017).

2.9.3 Función pérdida

Se utiliza una función perdida basada en una energía elástica para considerar la deformación de los objetos como se describe en (Park et al., 2021) en la Ecuación 2.65

$$L_{elastica}(\mathbf{X}) = \|\log \Sigma\|_F^2 \quad (2.65)$$

y la regularización del fondo que se asume estático y con una cantidad de puntos característicos K como en la Ecuación 2.66

$$L_{fondo} = \frac{1}{K} \sum_{k=1}^K \|T(\mathbf{x}_k) - \mathbf{x}_k\|_2 \quad (2.66)$$

2.10. Redes Neuronales Convolucionales

Las redes neuronales convolucionales se caracterizan por ser robustas en la detección y abstracción de patrones en imágenes (LeCun et al., 1998), por lo que se consideran como una opción para la detección de patrones.

Una red convolucional esta compuesta de capas convolucionales de la forma $\mathbf{g} = C_{\Gamma}(\mathbf{f})$ las cuales actuan en una entrada dimensional $\mathbf{f}(x) = (f_1(x), \dots, f_p(x))$ aplicando un conjunto de filtros $\Gamma = (\gamma_{l,\nu}), l = 1, \dots, q, \nu = 1, \dots, p$ y una función de activación ξ típicamente $\xi(z) = \max(0, z)$ o ReLu *Rectified Lineal Unit* Ecuación 2.67, produciendo una salida de q dimensiones $\mathbf{g}(x) = (g_1(x), \dots, g_q(x))$ llamada mapa de características Ecuación 2.68 (Bronstein et al., 2017)

$$\xi(z) = \max(0, z) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad (2.67)$$

$$g_l(x) = \xi \left(\sum_{\nu=1}^p (f_{\nu} \star \gamma_{l,\nu})(x) \right) \quad (2.68)$$

La Ecuación 2.69 denota una operación de convolución (Bronstein et al., 2017).

$$(f \star \gamma)(x) = \int_{\Omega} f(x - x') \gamma(x') dx' \quad (2.69)$$

A la capa cuya función es disminuir la resolución se le considera como una capa de agrupación o *pooling* $\mathbf{g} = P(\mathbf{f})$ y se define como Ecuación 2.70.

$$g_l(x) = P(\{f_l(x') : x' \in \mathcal{N}(x)\}), \quad l = 1, \dots, q \quad (2.70)$$

Donde $\mathcal{N}(x) \subset \Omega$ es un conjunto alrededor de x y P es una permutación donde se obtiene el valor máximo en el caso de *maxpooling* (Bronstein et al., 2017). Una red convolucional se compone de múltiples capas de convolución y agrupamiento cuya jerarquía se representa por Ecuación 2.71:

$$U_{\Theta}(f) = (C_{\Gamma^{(K)}} \cdots P \cdots \circ C_{\Gamma^{(2)}} \circ C_{\Gamma^{(1)}})(f) \quad (2.71)$$

Donde $\Theta = \{\Gamma^{(1)}, \dots, \Gamma^{(K)}\}$ es un vector de parámetros de la red neuronal, se dice que el modelo es profundo cuando está compuesto de varias capas ver Figura 1.4.

El entrenamiento de la red se obtiene minimizando una función costo sobre un conjunto de datos de entrenamiento $\{f_i, y_i\}_{i \in \mathcal{I}}$ dada por Ecuación 2.72:

$$\min_{\Theta} \sum_{i \in \mathcal{I}} L(U_{\Theta}(f_i), y_i) \quad (2.72)$$

El valor de la función costo $L(x, y) = \|x - y\|$ para una salida respecto al valor verdadero de un dato de entrenamiento es igual a la diferencia entre el valor predicho y el valor verdadero.

Al término del entrenamiento y con la red neuronal ya entrenada se espera $U(f) \approx y(f)$ que al aplicar y utilizar la red en valores no antes usados durante el entrenamiento se obtenga como resultado valores muy cercanos a los esperados.

La minimización de la función costo se realiza por medio de propagación reversa y comúnmente se utiliza gradiente descendiente estocástico (LeCun et al., 1998).

2.11. Conjunto de datos.

Se utilizarán bases de datos públicas para el entrenamiento de las redes neuronales siempre que sea posible, sin embargo también se considera la creación de conjuntos de datos RGB-D con etiquetado particular a la(s) características que se requieran inferir.

Algunas tareas específicas de redes neuronales con imágenes RGB-D que cuentan con bases de datos públicas (Firman, 2016) son:

- Objetos aislados, conjunto de datos BigBIR (Singh et al., 2014)
- Seguimiento de la cámara y reconstrucción de escenas, conjunto de datos ICL-NUIM (Handa et al., 2014).
- Etiquetado Semántico, conjunto de datos NYUv2 (Silberman et al., 2012).
- Seguimiento, conjuntos de datos Kinect Tracking Precision dataset (Munaro et al., 2012) y Princeton Tracking Benchmark (Song & Xiao, 2013)
- Datos Sintéticos, conjunto de datos SceneNet (Handa et al., 2016)

2.12. Valores atípicos y ruido en nubes de puntos en 3D

El ruido en la nube de puntos se describe en la Ecuación 2.73. Donde \mathbb{P}^l representa una nube de puntos en 3D, \mathbb{P} representa la superficie ideal de la muestra con p_i sobre la superficie escaneada del objeto, n_i es el ruido agregado y \odot es el conjunto de valores atípicos

que se encuentran presentes en la nube de puntos en 3D.

$$\mathbb{P}' = \{p'_i\} = \{p_i + n_i\}_{p_i \in \mathbb{P}} \cup \{o_j\}_{o_j \in \mathbb{O}} \quad (2.73)$$

$$\tilde{o}_i > 0,5 \quad (2.74)$$

La Ecuación 2.74 representa la probabilidad de encontrar valores atípicos \tilde{o} . Establece que un punto es agregado al conjunto de valores atípicos si la probabilidad es mayor a 0.5 .

Cabe destacar que las capas convolucionales Conv1D de la Figura 3.16 aplican una convolución de 1 dimensión a la señal de entrada compuesta por multiples planos de entrada respresentada por la Ecuación 2.75

$$out(N_i, C_{out_j}) = bias(C_{out_j}) + \sum_{k=0}^{C_{in}-1} weight(C_{out_j}, k_i) * input(N_i, k) \quad (2.75)$$

Donde $*$ es un operador de correlación cruzada válido, N_i es el tamaño de lote (batch size), y C denota el número de canales. Además, capas adicionales se utilizaron para la normalización por lotes (batch norm), como en la Ecuación 2.76.

$$y = \frac{x - E[x]}{\sqrt{Var[X] + \epsilon}} * \gamma + \beta \quad (2.76)$$

Donde γ y β son parámetros vector de aprendizaje de tamaño C (C es el tamaño de la entrada), la media y la desviación estandard son calculadas por dimensión sobre los mini-lotes (son una pequeña muestra de los datos). El valor γ es configurado en 1 y los β a 0. The standard deviation is calculated using the biased estimator.

2.12.1 Evaluación del modelo

Al reducir los valores atípicos de una nube de puntos en 3D se debe considerar que los puntos removidos corresponden a los verdaderos positivos (True positive) y que el numero de puntos removidos de manera erronea se cuantifica como falsos positivos (False Positive) (Visa et al., 2011) Estas dos métricas contribuyen a valor de exactitud (Accurracy) en la Ecuación 2.77, El valor de recuperación (Recall) se visualiza en la Ecuación 2.78, la métrica F1-Score en la Ecuación 2.79, y el error cuadrático medio (MSE mean square error) en la Ecuación 2.80.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (2.77)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (2.78)$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (2.79)$$

$$mse = \frac{\sum_{i=1}^n (y_i - y'_i)^2}{n} \quad (2.80)$$

Se considera la distancia Chamfer como una métrica de evaluación, la distancia Chamfer es una métrica universalmente reconocida en varias tareas relacionadas con nubes de puntos en 3D, se utiliza en métodos basados en el vecino mas cercano (nearest-neighbor) (Wu et al., 2021) La distancia Chamfer entre dos conjuntos de puntos S_1 y S_2 se define en la Ecuación:

$$d_{CD}(S_1, S_2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2 \quad (2.81)$$

Donde cada punto $x \in S_1$ encuentra su vecino mas cercano en S_2 y al contrario, posteriormente todos los puntos y pares de distancias son promediados para producir un valor de forma-distancia.

3. METODOLOGÍA

3.1. Software y hardware utilizado

En este trabajo se utilizó una estación de trabajo con las siguientes características, Central Processing Unit por sus siglas en inglés, Unidad de Procesamiento Central (CPU) AMD Ryzen 5600x con 6 núcleos, 12 hilos de procesamiento, 3.7 GHz, 32 Mega Byte por sus siglas en inglés (MB) Caché L3, 3 MB Caché L2, 32 GB RAM, tarjeta gráfica NVIDIA GeForce RTX 3060 TI con 8 GB Memoria Graphics Double Data Rate 6 por sus siglas en inglés (GDDR6), 4864 núcleos CUDA, sistema operativo Ubuntu 20.04.2 Long Term Support por sus siglas en inglés (LTS), Contenedores Docker 19.03.8, controlador CUDA version 11.2. Python 3.9 como lenguaje de programación. Adicionalmente se utilizó una laptop Macbook air (Retina 13-inch, 2020) con CPU 1.1 GHz Quad-Core Intel Core i5, 8 GB de RAM y tarjeta gráfica integrada Intel Iris Plus Graphics 1536 MB, sistema operativo Mac OS Big Sur 10.13 Beta y Python 2.9.

El software utilizando en la estación de trabajo fue:

- Docker container Linux Engine Versión 19.03.8
- Ubuntu 20.04.2 LTS
- MeshLab versión 2020.12
- Pycharm versión 2020.1
- Jupyter notebook
- Pycharm versión 2020.1

Las imágenes de profundidad producidos por la cámara en los experimentos contienen 101,760 puntos en 3D por cada captura de una escena. La escena en la imagen de profundidad esta representada en enteros y tiene que ser convertida mediante un valor de 0.001 que viene configurado de fábrica en la cámara RGB-D como unidad de profundidad. Los valores de tipo entero dentro de la imagen de profundidad se convirtieron a metros utilizando la fórmula 2.43.

Se utiliza un equipo de bajo desempeño para la creación de conjuntos de datos para la eliminación de valores atípicos y ruido, las características del equipo son: Laptop HP Pavilion 4 Gb RAM, 500 Gb HDD, GTX1650, Rizen 7. El software utilizado para el preprocesamiento de nubes de puntos y visualización en el proceso de limpieza de valores atípicos y ruido es:

- MeshLab versión 2020.12

- Anaconda Python 3.7 64-Bit
- Jupyter notebook
- Pycharm versión 2020.1

El pre-procesamiento de las nubes de puntos fue realizado cargando la nube de puntos a la GPU, para la visualización de las nubes de puntos en 3D se utilizó Meshlab (Edelmers et al., 2021), donde las nubes de puntos contenían 140 mil puntos y utilizaban 532 MB de memoria RAM.

Adicionalmente se utilizó Google Colab para el entrenamiento e inferencia en el proceso de eliminación de valores atípicos y ruido en las nubes de puntos en 3D.

El paquete de Google Colab incluyó las siguientes especificaciones:

- K80, P100, T4 24 GB GPU 2 x vCPU
- 358 Gb disk space
- Python 3.7

3.2. Cámara RGB-D.

Durante la captura de las escenas se utilizó una cámara RGB-D Intel Realsense D435 con las siguientes características (Tabla 3.1):

Tabla 3.1: Intel Realsense D435(Keselman et al., 2017).

Características	Descripción
Rango de operación	~0.11 m - 10 m
Interfaz de conexión	USB tipo C
Dimensiones	90 mm x 25 mm x 25 mm
Resolución de profundidad	1280 x 720

Los factores intrínsecos de fábrica los proveen metadatos en la configuración de propia cámara, los parámetros de distorsión se encuentran en cero y las imágenes capturadas pueden variar de tamaño dependiendo de la conexión y tipo USB usado. Las imágenes presentes se crearon principalmente utilizando una Raspberry Pi 4 con los siguientes tamaños:

- Capa de datos RGB 640x480
- Capa de datos de profundidad 424x240

Las imágenes de profundidad contienen 101,760 puntos en cada captura de una escena. Los datos en la capa de profundidad se almacenan con datos de tipo entero, los cuales tienen que ser convertidos mediante un valor unidad de profundidad para la conversión a metros, este valor es proporcionado en los datos de la propia cámara en la configuración de fábrica, en este caso el valor es: 0,001 mediante la Ecuación 2.43,

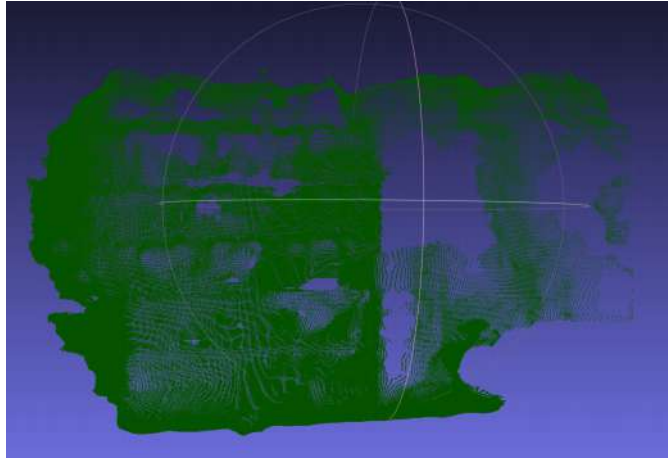


Figura 3.1: Nube de puntos 3D

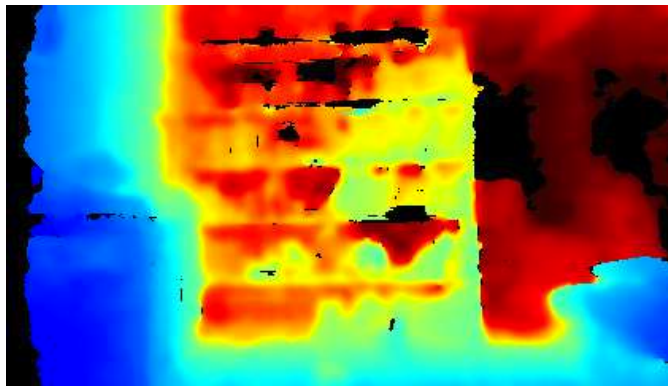


Figura 3.2: Representación de profundidad

Y se realiza una proyección desde coordenadas 2D a coordenadas en 3D mediante la Ecuación 2.44 representando la nube de puntos 3D con coordenadas (x, y, z) . Las dimensiones se representan en metros con números de punto flotante (0,876, 0,1234, 0,345). El valor máximo y mínimo de profundidad depende directamente de la escena capturada y de las características de la cámara RGB-D.

En la Figura 3.1 se muestra una nube de puntos 3D de una escena compleja en composición de objetos, un librero a 2 metros de distancia de la cámara RGB-D, y en el librero se destacan múltiples volúmenes en un mismo color. Se muestra información contenida en un archivo PLY (Figura 3.1) que es desplegada mediante el programa Meshlab, se observa que todos los puntos en 3D se muestran en color verde y no hay una distinción clara de los objetos presentes en la escena, se observa de manera predominante la estructura de un librero. En la Figura 3.2 se aprecian de manera más clara las diferentes capas de profundidad debido a la representación en colores del azul al rojo conforme aumenta la distancia desde la cámara RGB-D.

Se visualiza la toma de una escena mediante la cámara RGB-D (Figura 3.1), se muestra una imagen a color representando la profundidad, el color rojo es mayor distancia desde la cámara, el color azul representa menor distancia desde la cámara en la Figura 3.2). Se procura

capturar objetos a diferentes distancias en cada experimento.

3.3. El método RANSAC y el balón de basketball

Se utiliza un balón de basketball de tamaño estandar número 7 con dimensiones aproximadas de 35 cm de circunferencia y un radio de 12 cm.

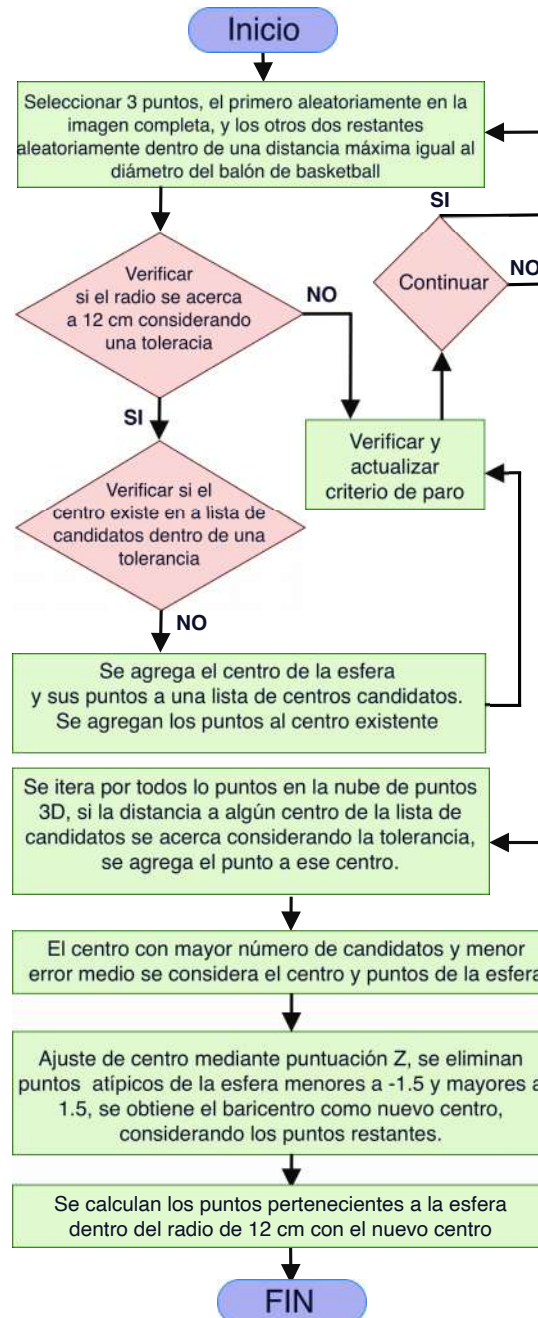


Figura 3.3: Diagrama de flujo para encontrar una esfera en una nube de puntos.

En el algoritmo que se muestra en la Figura 3.3 como primer paso, tres puntos son seleccionados de manera aleatoria de una nube de puntos en 3D, se obtiene el centro por

medio de la Ecuación 2.50 y el radio por medio de la Ecuación 2.46. En un segundo paso, se verifica que el radio se encuentre en una tolerancia $\epsilon = 1\text{cm}$ alrededor del valor del radio del balón de basketball de 12 cm. Si se encuentra dentro de la tolerancia se verifica que el centro de la esfera que representa el balón ya exista en una lista de centros candidatos, si el centro no existe en esta lista, se agrega a la lista junto con los puntos correspondientes a la superficie de la esfera. Si el radio no se encuentra dentro de la tolerancia, se itera nuevamente hasta que una condición de paro se cumpla, la cuál puede ser un límite de iteraciones. En el experimento se utilizaron 1000 iteraciones como criterio de paro. Si el criterio de paro se alcanza, se procesan todos los puntos en la nube de puntos 3D y se verifican las distancias respecto a los centros registrados, si la distancia a algún centro se encuentra dentro de la tolerancia, cada punto evaluado se agrega a la lista de puntos de superficie correspondiente al centro que corresponda.

Una vez la clasificación de puntos a sus correspondiente centros termina, la lista de centros es ordenada con respecto a la cantidad de puntos de superficie que cada centro tiene y con respecto al error en la tolerancia del radio de la esfera, se selecciona el centro con mayor número de puntos y menor error.

Posteriormente se realiza un ajuste al centro seleccionado, para esto se utiliza Z-score, para eliminar valores atípicos, discriminando datos con un valor de Z-score menor a -1.5 y mayor a 1.5. Con los datos que se conservan se calcula el barycentro y se toma como nueva posición para el centro de la esfera junto a todos los puntos restantes como puntos en la superficie de la esfera.

Se selecciona una escena con una pared relativamente plana y rugosa al fondo y mediante un balón de basketball que se coloca entre la pared y a aproximadamente 25 cm de la cámara realsense D435, se generan capturas de la escena en color RGB como se muestra en la Figura 3.4



Figura 3.4: Escena experimento 1

La captura de la profundidad de la escena también es realizada con la cámara realsense D435, la cuál es convertida a un archivo PLY mediante las Ecuaciones 2.43 y 2.44. La escena se despliega en colores que van del azul al rojo como se muestra en la Figura 3.5 y posteriormente el archivo PLY se visualiza mediante la herramienta meshlab como se muestra en la Figura 3.6.

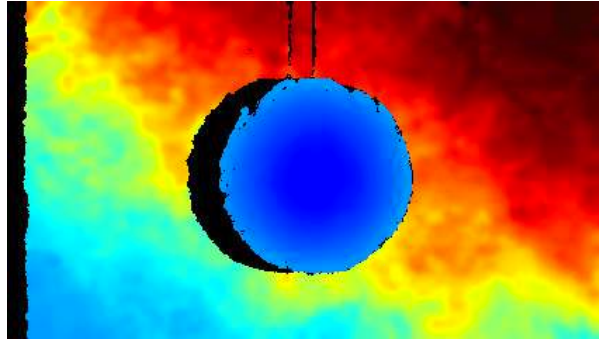


Figura 3.5: Imagen de profundidad correspondiente a la escena del experimento 1

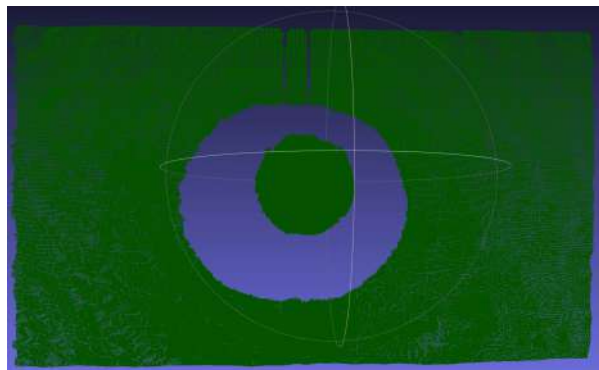


Figura 3.6: Visualización del archivo PLY del experimento 1 en el software Meshlab

Finalmente se propone el siguiente algoritmo:

Algoritmo 1 Ajuste de una esfera con tamaño conocido por medio de RANSAC en una nube de puntos en 3D

Se configuran valores iniciales, diámetro, radio, tolerancias, criterio de paro

$\epsilon \geq 0,012$

$criteriodeparo \geq 100000$

$radio \geq 0,12$

$diameter \leftarrow 0,024$

1: Muestreo de tres puntos en 3D

a).- El primer punto es seleccionado de manera aleatorio del espacio total de datos en 3D.

b).- El segundo y tercer puntos se seleccionan de un espacio menor dentro del *diámetro* cercano al primer punto.

2: Se resuelven los parámetros en 2.50 en la forma $Ax = B$,

if el centro de una esfera se encuentra **then**.

3: Se determina si los tres puntos se encuentran dentro de la tolerancia ϵ con respecto a el *radio*.

if Si **then**

El centro se ajusta mediante Z-score.

El centro ajustado se agrega a la lista de centros.

Se itera al paso 1 conforme al criterio de paro *criteriodeparo*.

else if no **then**

Se itera al paso 1 conforme al criterio de paro *criteriodeparo*.

end if

else if no **then**

Se verifica si el criterio de paro *criteriodeparo* se ha alcanzado.

if Si **then**

Continúa al paso 4.

else if no **then**

continúa iterando al paso 3.

end if

end if

4.- Se itera en todos los puntos del espacio de puntos en 3D y se verifica se encuentren dentro de la tolerancia ϵ del *radio* para cada punto en la lista de centros candidatos.

if Si **then**

Se agrega el punto al centro correspondiente en la lista de centros candidatos.

end if

5: Se verifica que la lista de centros candidatos no se encuentre vacía,

if Si **then**

Ninguna esfera se encontró y el método requiere ejecutarse nuevamente.

go to step 1

else if No **then**

Se itera la lista de centros candidatos.

Se selecciona el centro con mayor cantidad de puntos candidatos que se ajuste de mejor manera al tamaño del balón de basketball.

end if

6: Se ha encontrado la esfera con sus puntos de su superficie en la imagen de profundidad.

3.4. Calibración de la cámara RGB-D

El protocolo de calibración propuesto requiere de la configuración de un tripié con una barra de aluminio para estabilizar la cámara de manera horizontal, mediante un cable USB tipo C se conecta la cámara al equipo Raspberry Pi 4. Se verifica la correcta alineación vertical de la cámara con un nivel de burbuja y de esta manera se puede comenzar a capturar imágenes, la disposición y acomodo de la cámara se muestra en la Figura 3.7



Figura 3.7: Configuración de la cámara y tripié.

Es importante notar que la posición final de la cámara debe de configurarse antes de la toma de las imágenes evitando desacomodar la configuración durante la captura de datos, esta característica hace conveniente que capturar un número pequeño de imagenes sea mucho mejor.

Para la calibración de la cámara se utiliza el balón de basketball.

3.4.1 Método para la calibración de la cámara RGB-D

La primera fase del método propuesto que se muestra en la Figura 3.9 utiliza el método propuesto por (Z. Zhang, 2000), para corregir la distorsión geométrica de las imagenes de color y obtener la matriz de de parámetros intrínsecos, en este paso, 10 imágenes mostrando un patrón de tablero de ajedrez son suficientes, por lo que se requiere tomar al menos 10 imágenes donde el tablero de ajedrez sea visible como se muestra en la Figura 3.8.

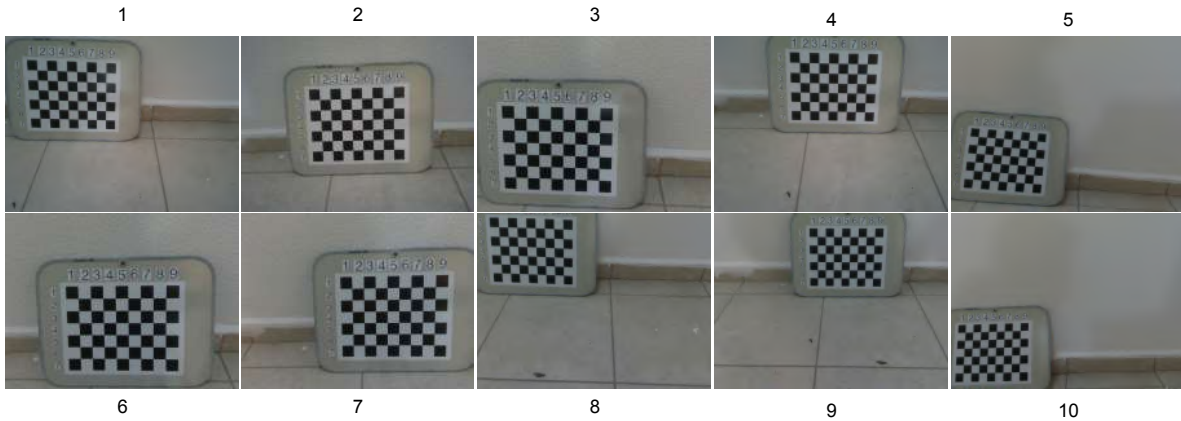


Figura 3.8: Conjunto de imágenes para la corrección del error geométrico.

Los detalles de implementación para la corrección del error geográfico están basados en (Burger, 2016) y se puede descargar código de ejemplo desde (Enazoe, 2022)

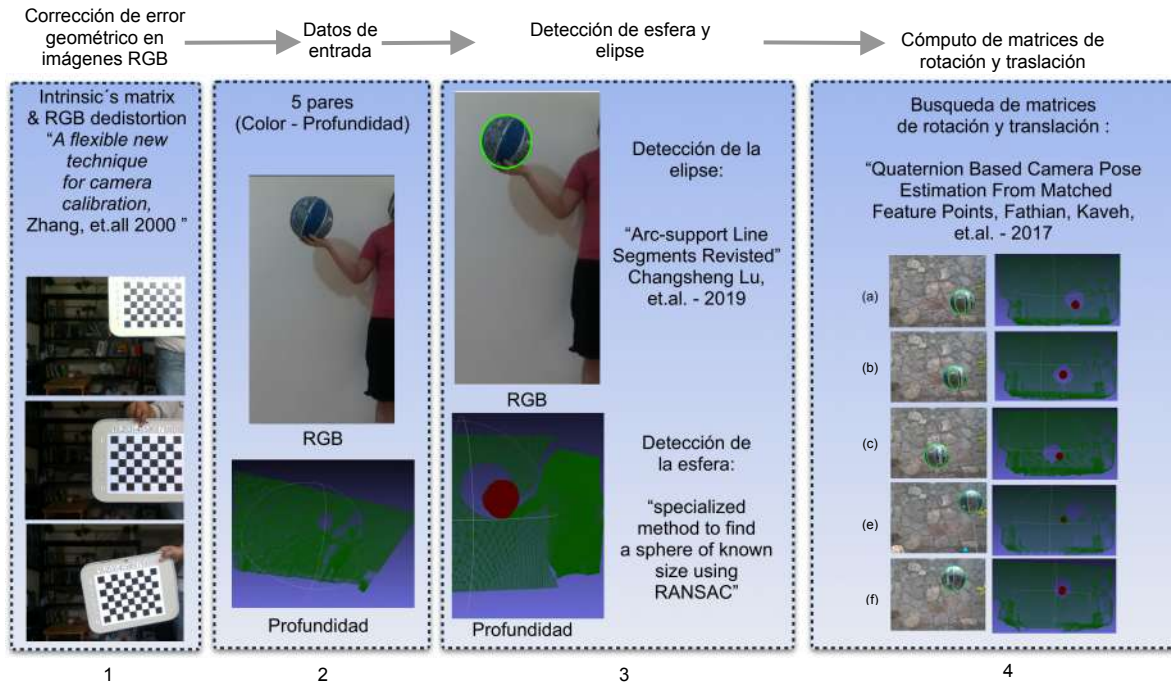


Figura 3.9: Metodología propuesta para la calibración de la cámara RGB-D.

En seguida el método de la Figura 3.9 requiere de 5 pares de imágenes color-profundidad donde el balón de basketball sea visible en la escena en todas las imágenes como se muestra en la Figura 3.10.

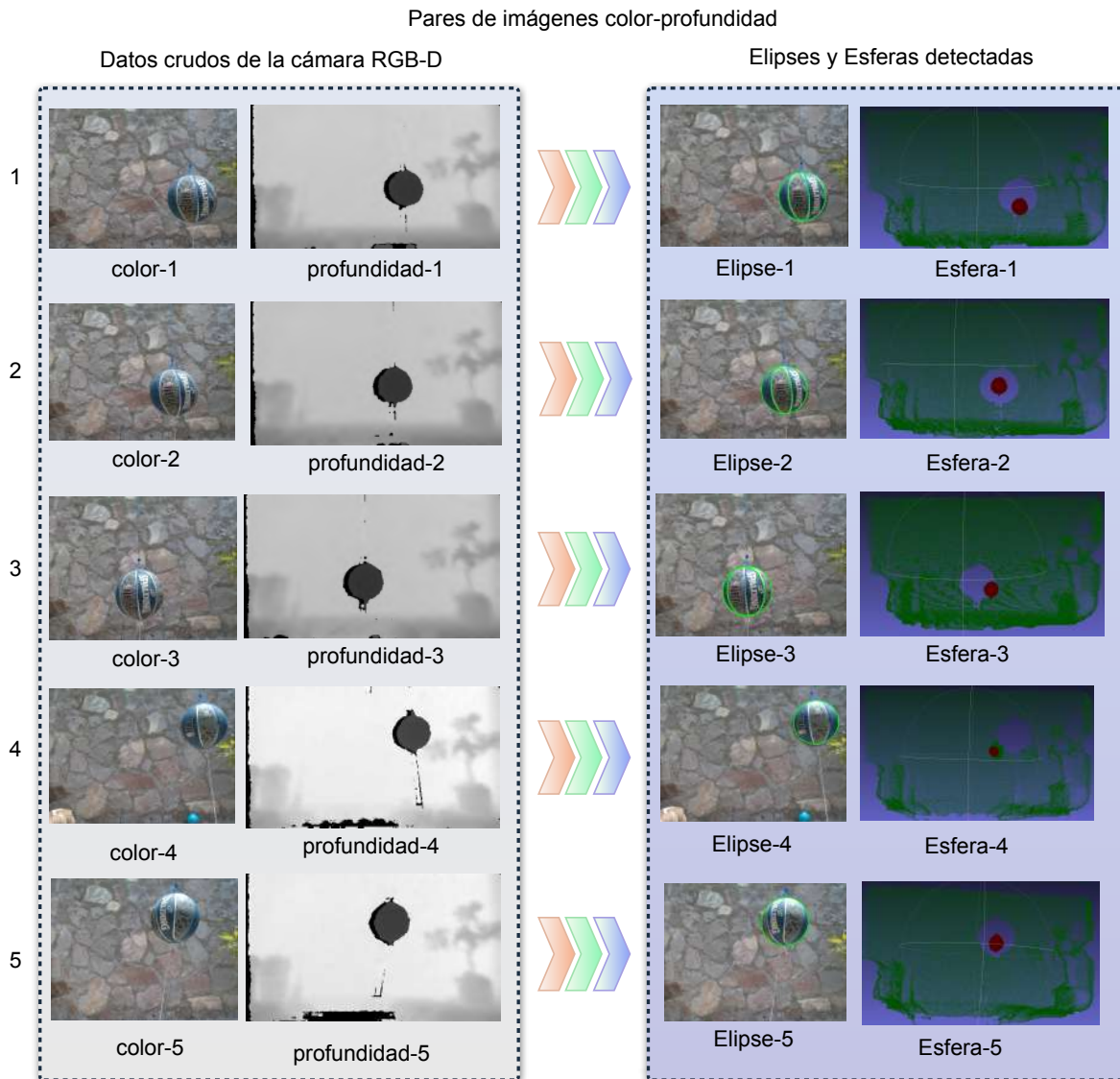


Figura 3.10: Se requieren de 5 pares de imágenes color-profundidad.

Las elipses se localizan en las imágenes de color por medio del método Arc-support Line Segments (Lu et al., 2019). En la capa de profundidad se localizan las esferas en la capa de color mediante nuestro método. Finalmente se utiliza el método Quaternion Based Camera Pose Estimation, Estimación de la posición de la cámara basada en cuaterniones (QuEsT) (Fathian et al., 2018) para obtener las matrices de rotación y traslación para poder alinear y corresponder ambas capas de información color-profundidad. El protocolo de calibración propuesto sigue los siguientes pasos:

- Corregir error geométrico con al menos 10 imágenes con un patrón de tablero de ajedrez.
- Localizar círculos y esferas en al menos 5 pares de imágenes color-profundidad con un balón de basketball visible.

- Obtener la matriz de rotación y traslación mediante el método QuEsT (Fathian et al., 2018).

3.5. Limpieza de valores atípicos y ruido en nubes de puntos en 3D

3.5.1 Red neuronal basada en PointCleanNet

PointCleanNet (Rakotosaona et al., 2020) es un modelo neuronal basado en redes convolucionales diseñado considerando dos etapas principales para la limpieza de los elementos de una nube de puntos en 3D, a su vez esta arquitectura esta basada en el modelo neuronal PCPNet (Guerrero et al., 2018) que estima características locales de manera robusta y utiliza esta información para limpiar el ruido de la nube de puntos en 3D, como se muestra en la Figura 3.11.

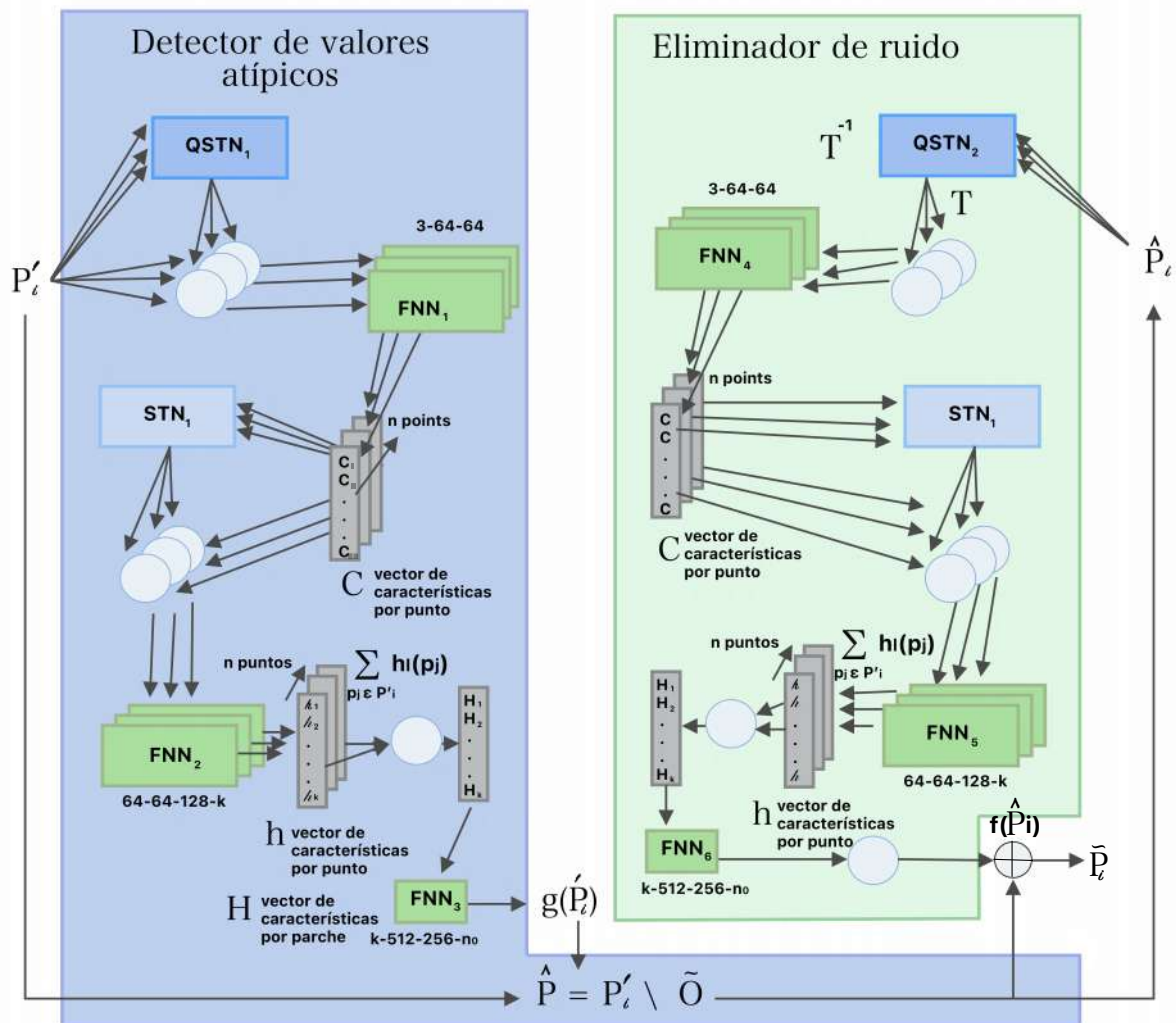


Figura 3.11: Etapas de reducción de ruido de PointCleanNet como se describen en (Rakotosaona et al., 2020).

Las etapas de reducción de ruido permiten procesar las nubes de puntos sin perder características relevantes.

Para la reducción de valores atípicos la red neuronal esta basada en PCPNet (Guerrero et al., 2018) como se muestra en la Figura 3.12 donde los bloques Spatial Transformer Network, Red Espacial de modelo Transformer (STN) y Quaternion Spatial Transformer, Transformer Espacial basado en cuaterniones (QST) son utilizados para estimar propiedades de las formas locales de los objetos como sus normales y la curvatura que presentan los puntos 3D en conjunto, proporcionando una mejor definición en la forma y densidad de los puntos 3D.

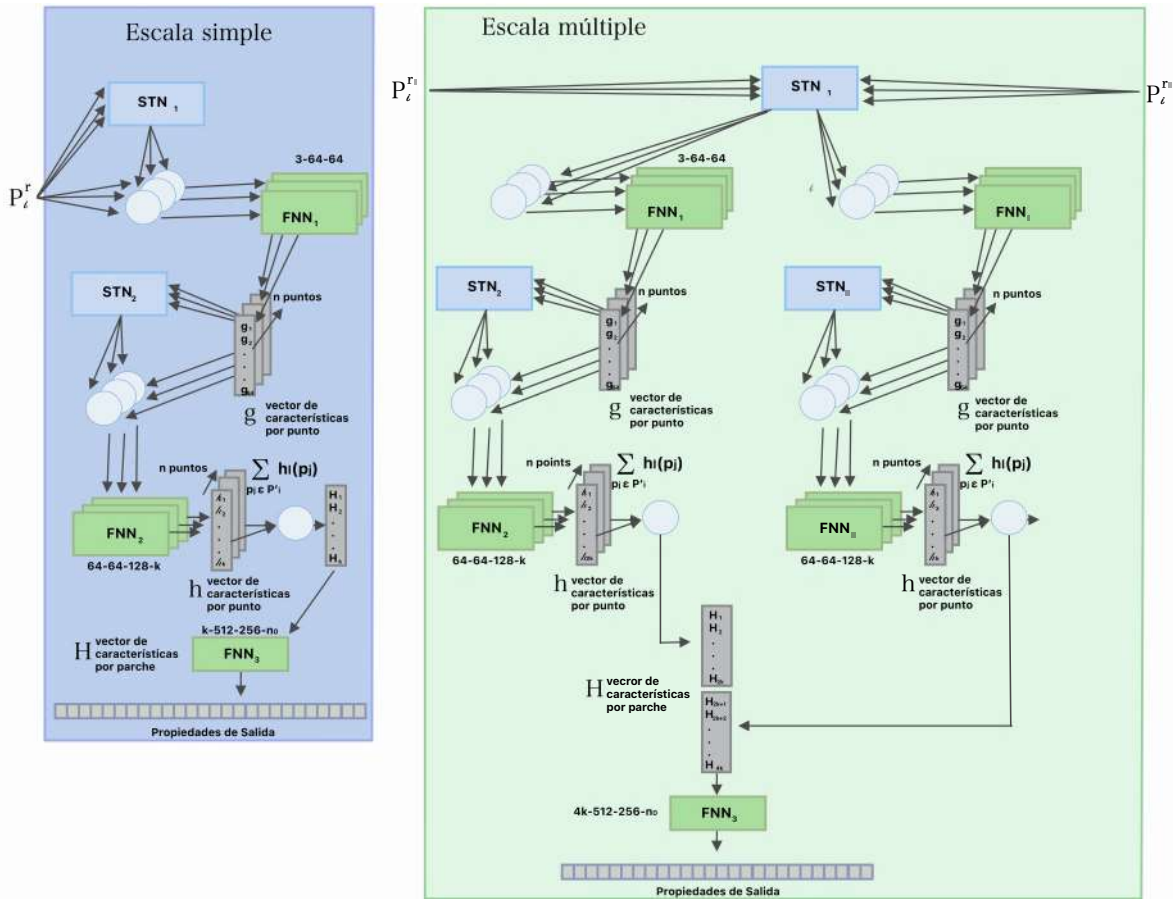


Figura 3.12: Arquitectura PCPNet como se describe en (Guerrero et al., 2018).

La red PCPNet aprende las características de la nube de puntos 3D en espacios locales como un conjunto de puntos

Mediante un dataset propio se realiza la clasificación de valores atípicos en nubes de puntos de diferentes densidades. Se genera ruido en nubes de puntos agregando ruido Gaussiano con una desviación estandar de 0.01 % respecto a la superficie de las Figuras originales dentro de la nube de puntos en una diagonal dentro una caja que contiene dichas Figuras.

En total, se utilizaron 40 nubes de punto como conjunto de entrenamiento para disminución de ruido con 8 niveles de ruido para 5 Figuras.

El preprocesamiento fue realizado cargando las nubes de puntos en la memoria de la GPU, para la visualización se utilizó Meshlab (Cignoni et al., 2011) donde una nube de

puntos de 140,000 elementos utilizó 532 MB de RAM.

Adicionalmente se utilizó Google Colab para el entrenamiento en inferencia, el paquete seleccionado tuvo las siguientes características: K80, P100, T4 24GB GPU 2 x vCPU, 358 Gb disco duro, Python 3.7

3.5.2 Conjunto de datos para la limpieza de valores atípicos y ruido.

Se creó un conjunto de datos propio muy similar a los disponibles por PointCleanNet (Rakotosaona et al., 2020), con la principal diferencia que los datos creados tienen una menor densidad, se dividen en dos secciones, una dedicada al entrenamiento con treinta nubes de puntos en 3D y otra sección para pruebas con diez nubes de puntos en 3D como se muestra en la Tabla 3.2.

Tabla 3.2: Conjunto de datos utilizado para entrenamiento y validación.

Nombre conjunto de datos	Puntos de la escena	Nube de puntos
Conjunto de datos de PointCleanNet	100,000	28
Conjunto de datos de Balón	15,843	40

El conjunto de datos contiene nubes de puntos sin ruido para como patrón de referencia y nubes de puntos con diferentes magnitudes de ruido blanco y valores atípicos. Un ejemplo de la captura de escena se muestra en la Figura 3.13, donde posteriormente se filtran los puntos que no pertenecen al balón y se agrega ruido gaussiano para posteriormente realizar la reconstrucción 3D como se aprecia en la Figura 3.14.

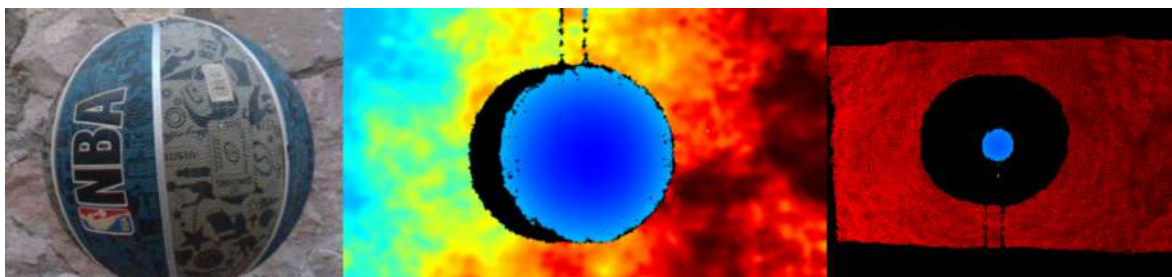


Figura 3.13: Ejemplo del conjunto de datos, el mapa de calor muestra la profundidad de la escena.

Se obtiene solo una semi-esfera como se muestra en la Figura 3.14 debido a la posición de la cámara respecto al balón de basketball, la imagen corresponde a una nube de puntos que contiene 15,843 puntos como nuestra imagen patrón de referencia.

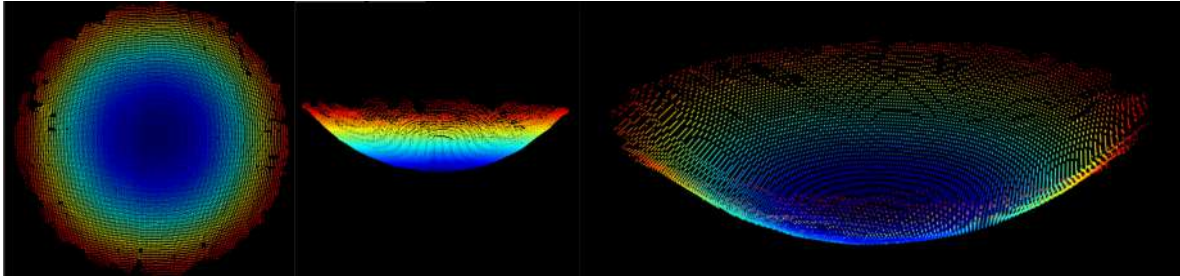


Figura 3.14: Reconstrucción 3D del balón de basketball.

La imagen de la Figura 3.15 corresponde a la reconstrucción 3D con un ruido Gaussiano de 1×10^{-3} desviación estándar.

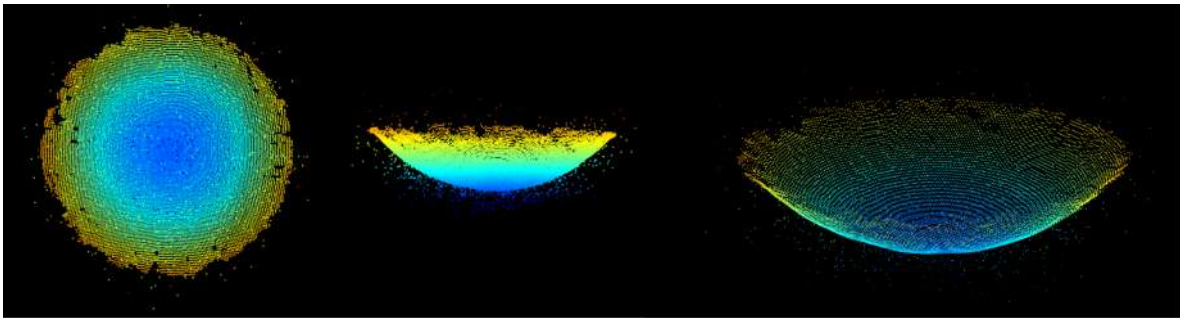


Figura 3.15: Nube de puntos correspondiente a la reconstrucción del balón de basketball con ruido Gaussiano de 1×10^{-3} desviaciones estándar

3.5.3 Arquitectura neuronal propuesta para la limpieza de valores atípicos y ruido.

Se muestra un bloque básico convolucional de una dimensión en la Figura 3.16, PointCleanNet (Rakotosaona et al., 2020) utiliza los bloques a) y c), comienza con una capa convolucional encargada de extraer los mapas de características, y después aplica varias transformaciones para producir nuevos mapas de características, la capa de información es reducida utilizando normalización por lotes (batch normalization) reduciendo la dimensionalidad de los mapas de características y obteniendo un máximo valor de cada uno de los mapas para transformar los parches. Una capa de acceso directo (shortcut layer) es utilizada para aprender características significantes de capas superiores. En los bloques propuestos b) y d) se ha adicionado una capa convolucional para obtener una extracción de características mayor, la cuál nos ayuda a identificar valores atípicos y presencia de ruido en la nube de puntos 3D.

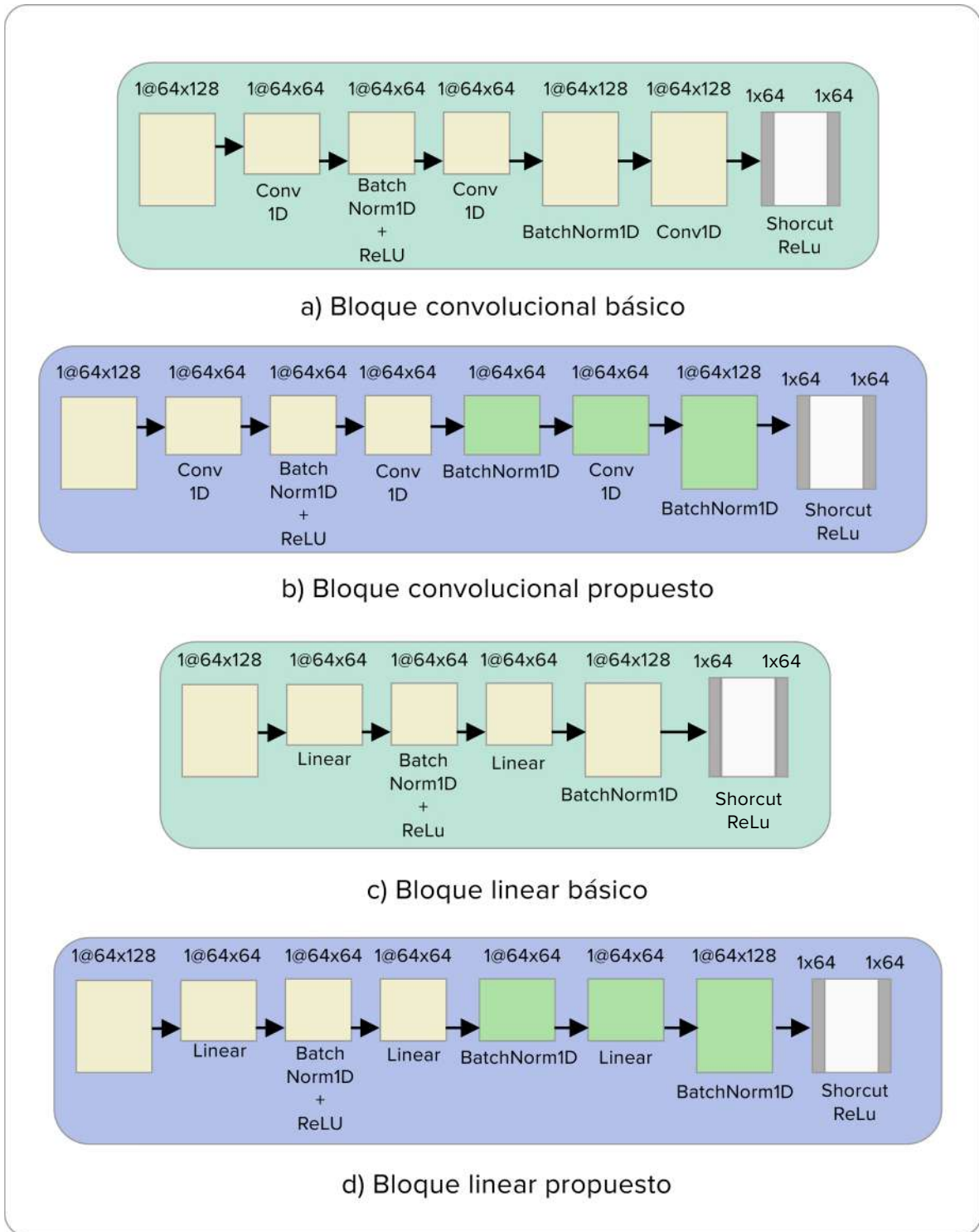


Figura 3.16: Bloque lineal convolucional básico propuesto.

El bloque básico de la Figura 3.16 c) representa un bloque FNN (Feed Forward Network) de PointCleanNet, en el bloque d) se observa que se ha agregado una capa lineal que provee un soporte adicional para identificar patrones locales en la nube de puntos.

Se comienza con una capa convolucional para la extracción de mapas de caracteris-

ticas para considerar características de valores atípicos y ruido, posteriormente se normaliza por lotes reduciendo la salida de las capas y la dimensionalidad de los mapas de características, extrayendo cada vez mapas mas pequeños y obteniendo valores maximos por cada uno de ellos, después se transforman los parches mediante una capa de acceso directo, la cuál ayuda a aprender características de alto nivel que pudiesen ser perdidas durante la normalización por lotes. De esta manera las capas de acceso directo obtienen características significativas de capas superiores.

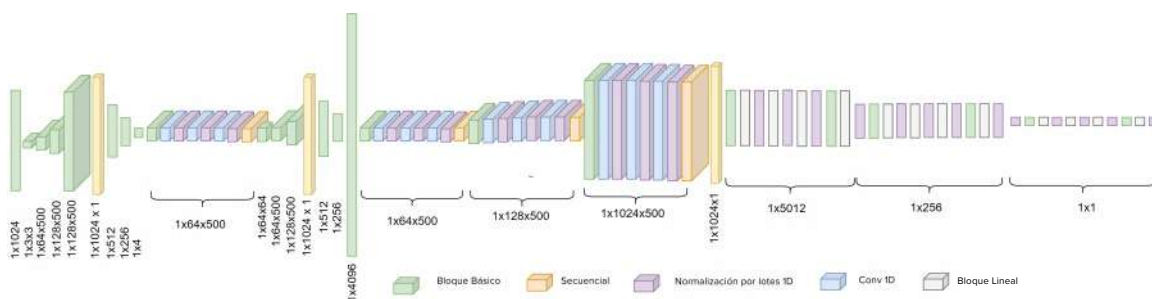


Figura 3.17: Arquitectura neuronal propuesta basada en PointCleanNet (Rakotosaona et al., 2020).

3.5.4 Entrenamiento de la red neuronal propuesta para reducir valores atipicos y ruido.

Se requirió de los siguientes pasos para el entrenamiento de la red neuronal:

- Para el entrenamiento utilizar nubes de puntos en 3D sin ruido y con ruido.
- El conjunto de datos requiere diferenciar entre puntos de un objeto y cuales no (etiquetado)
- El conjunto de datos requiere de referencias para permitir el cálculo de la matriz de confusión.
- El conjunto de datos se divide entre pruebas y validación
- Las nubes de puntos en 3D se procesan para la sección de entrenamiento y para la de validación.

El conjunto de datos se divide como se muestra en la Tabla 3.3:

Tabla 3.3: Conjunto de datos de entrenamiento y validación

Sección del conjunto de datos	Número de nubes de puntos
Entrenamiento	30
Pruebas	10

3.5.5 Metodología reconstrucción 3D de un objeto dinámico

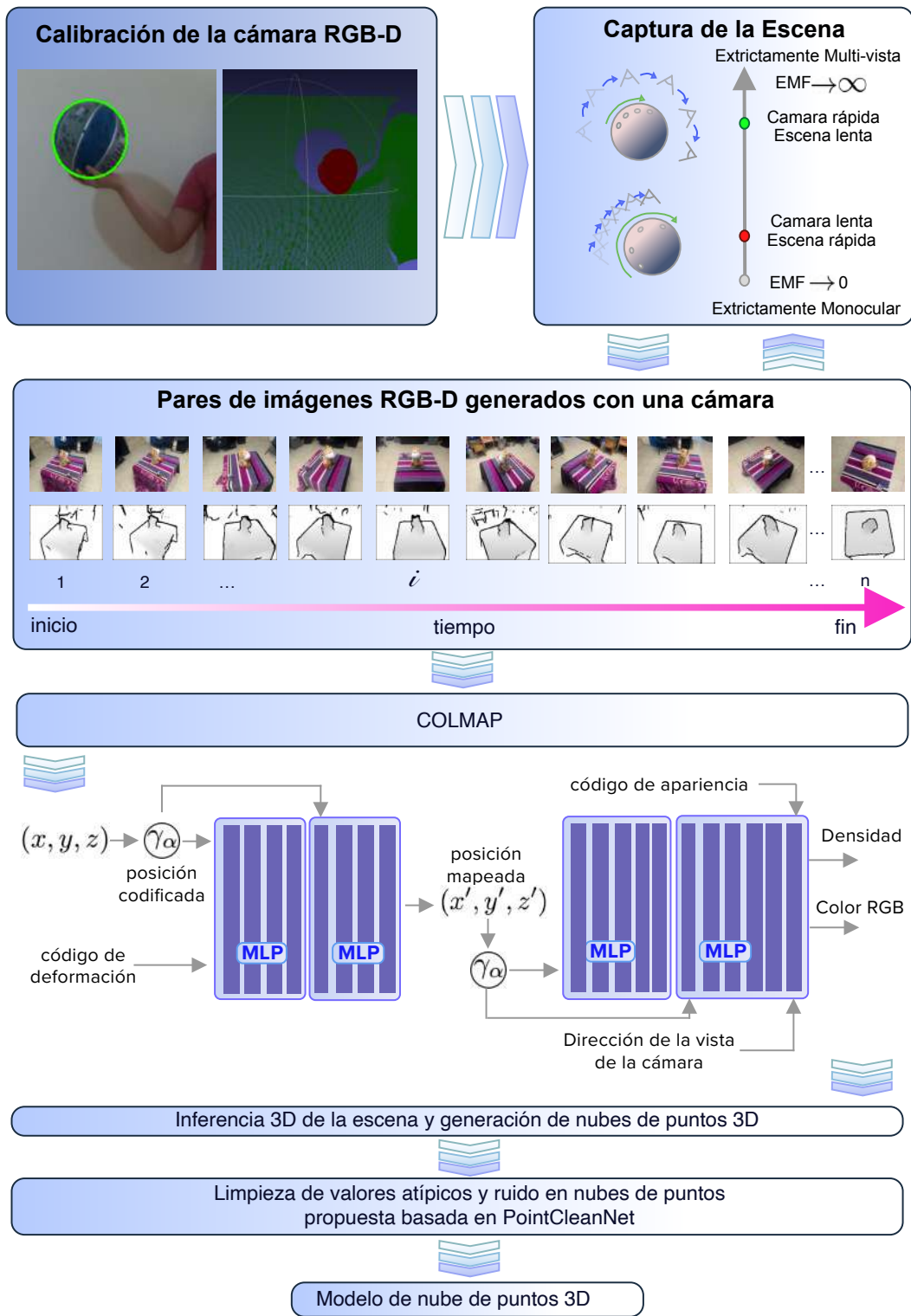


Figura 3.18: Metodología para la reconstrucción de un objeto 3D dinámico en una nube de puntos.

En la metodología propuesta en la Figura 3.18 se utilizan datos generados por una cámara RGB-D de la siguiente manera:

- La cámara RGB-D se calibra mediante los métodos propuestos eliminando errores geométricos y de correspondencia entre los datos de las capas de color y profundidad generados.
- Se realiza la captura de la escena, colocando como eje de rotación un objeto que se busca reconstruir en un modelo de nubes de puntos 3D. Es necesario capturar el objeto alrededor del mismo.
- Se asegura que la captura de la escena sea de una calidad aceptable para la reconstrucción del objeto deseado mediante la métrica EMFs (Gao et al., 2022a). Entre mayor sea el valor EMFs se tendrá mayor información del objeto a reconstruir a pesar de que su superficie se esté deformando en la escena. Si el valor es muy pequeño, la escena no será candidata para una reconstrucción exitosa, se asumen capturas de escena con valores EMFs de 0.7 como mínimo.
- La captura de la escena es ordenada cronológicamente en su captura, en pares de imágenes color-profundidad. En el presente trabajo de investigación se utilizaron 1000 pares imágenes como promedio en las capturas de la escena.
- Cada par de imágenes requiere información de parámetros intrínsecos y extrínsecos de la cámara como parte de la entrada al método NeRF por lo que se utiliza el método COLMAP (Schönberger & Frahm, 2016) para generar dichos parámetros.
- Se generan códigos de apariencia y posición basados en el método (Bojanowski et al., 2017) y un código de deformación basado en (Park et al., 2021) considerando tanto las capas de color como de profundidad.
- En una primera etapa se obtienen las posiciones desplazadas debido a las deformaciones del objeto respecto a una referencia canónica, dichas posiciones desplazadas son la entrada para el siguiente bloque basado en el método NeRF (Mildenhall et al., 2021).
- Se exporta la representación de la escena en 3D capturada por medio de los bloques NeRF a una nube de puntos 3D, utilizando el método propuesto por (Ma et al., 2023).
- Se realiza una limpieza de valores atípicos y ruido mediante una metodología propuesta basada en PointCleanNet (Rodríguez et al., s.f.).
- Se obtiene un modelo de nubes de puntos en 3D.

4. RESULTADOS

4.1. Deteccion de la esfera de basketball

La correcta detección del balón de basketball se muestra en las Figuras 4.1 y 4.2 como una geometría esférica en un primer experimento con buenas condiciones en la escena donde el balón de basketball se distingue facilmente del resto de la escena debido a que no está ocluida por otros objetos. Los puntos en 3D que pertenecen a la superficie de la esfera se muestran coloreados en rojo para distinguirlos del resto de puntos 3D en la escena.

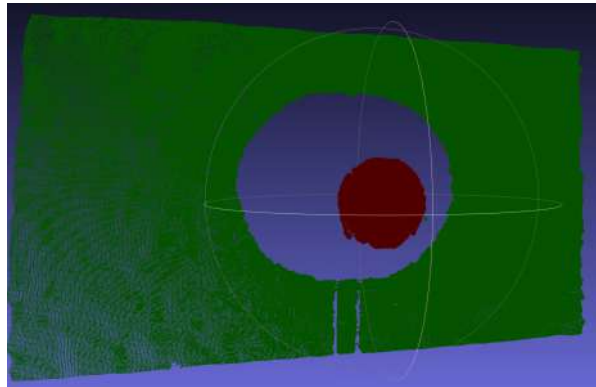


Figura 4.1: Correcta detección del balón de basketball en el experimento 1 mediante Meshlab

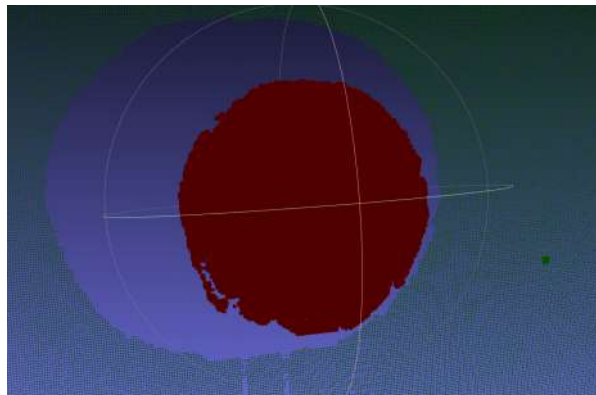


Figura 4.2: Mayor detalle de la esfera detectada.

La tolerancia ϵ define la cantidad de valores atípicos en la cercanía a la superficie de la esfera. En este experimento se utilizaron 1000 iteraciones como criterio de paro para

buscar en la imagen completa en cada iteración, en el experimento se tomaron diferentes candidatos a solución en los cuales se genera un centro de esfera con sus puntos de superficie correspondientes, así, diferentes soluciones se proponen siempre que se satisfaga el modelo matemático propuesto en la Ecuación 2.50 dentro de la tolerancia establecida. Sin embargo el candidato con el mayor número de puntos cuyo radio se aproxime con un menor error a la medida del radio de 12cm , es la que se selecciona como solución. En este experimento los diferentes candidatos generados corresponden al mismo objeto en escena, así que todos los centros candidatos son muy similares, sin embargo el método converge al centro con mejores condiciones y menor error y se ajusta mediante Z-score 2.51, los puntos que resultan pertenecer a la superficie de la esfera del centro ganador se comparan en un a lista y todos aquellos con un valor z-core menor a $-1,5$ y mayores a $1,5$ son descartados pues se consideran valores atípicos, los valores $-1,5$ y $1,5$ fueron seleccionados mediante una heurística basada en un menor de tipo Root Mean Square Error, Error cuadrático medio (RMSE). Con los puntos filtrados se calcula un nuevo centroide y una nueva esfera. Como se puede apreciar en la Figura 4.3 se observan los puntos atípicos en rojo y los puntos filtrados en azul.

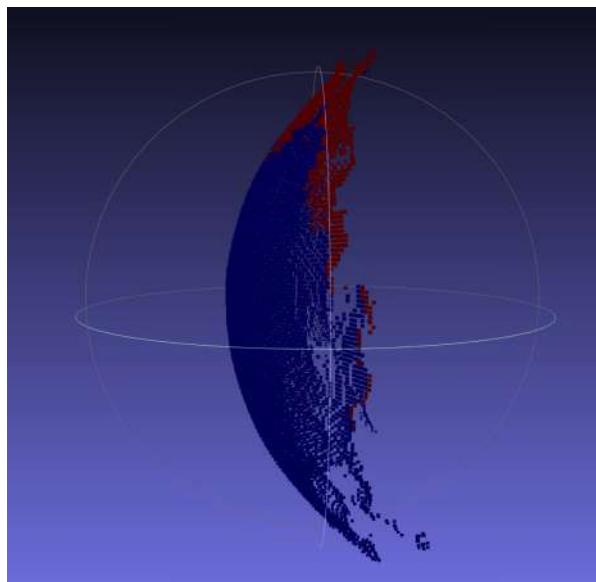


Figura 4.3: Esfera ajustada con valores atípicos en rojo, visualización mediante Meshlab

En un segundo experimento, se genera una imagen de una escena con la misma cámara RGB-D realsense D435, pero esta vez con mayor complejidad en la escena misma, en la parte central de la imagen se encuentra un librero con varios objetos con diversas geometrías, algunas cercanas a una esfera, en la escena se encuentra el balón de basket ball arriba de un tripié, el método es capaz de detectar la esfera, la escena del segundo experimento se muestra en la Figura 4.4.



Figura 4.4: Escena del experimento 2

La imagen de profundidad de la escena se muestra en:

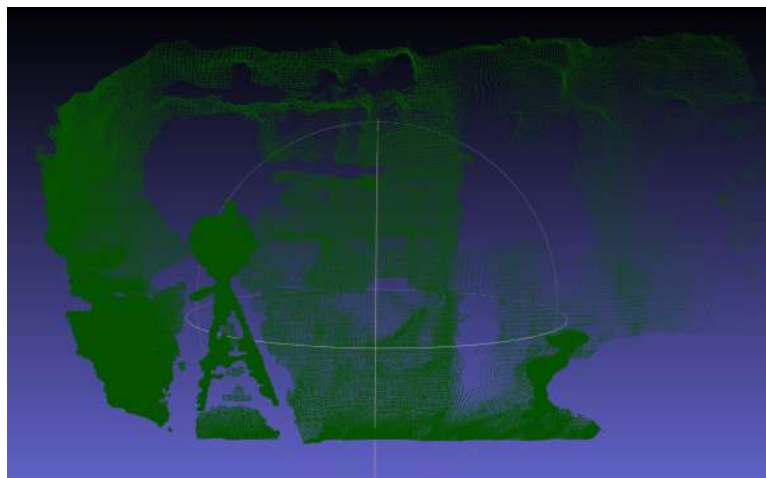


Figura 4.5: Escena del experimento 2, información del archivo PLY visualizada en Meshlab.

La visualización de la correcta detección de la esfera que representa al balón de basketball en la escena con mayor complejidad Figura 4.6 donde se muestra que los valores atípicos se encuentran cercanos a la superficie de la esfera y pudiesen pertenecer al tripié donde el balón se encuentra ubicado. En este experimento se utilizaron 10,000 iteraciones como criterio de paro. Se encontraron diferentes centros candidatos a esfera como solución los cuales también satisfacen la Ecuación 2.50 dentro de la tolerancia establecida, sin embargo en este experimento se demuestra el buen uso del método RANSAC para discriminar falsos positivos eliminando aquellos candidatos con un error mayor y seleccionando al mejor candidato que se acercó al radio de 12cm Figura 4.6

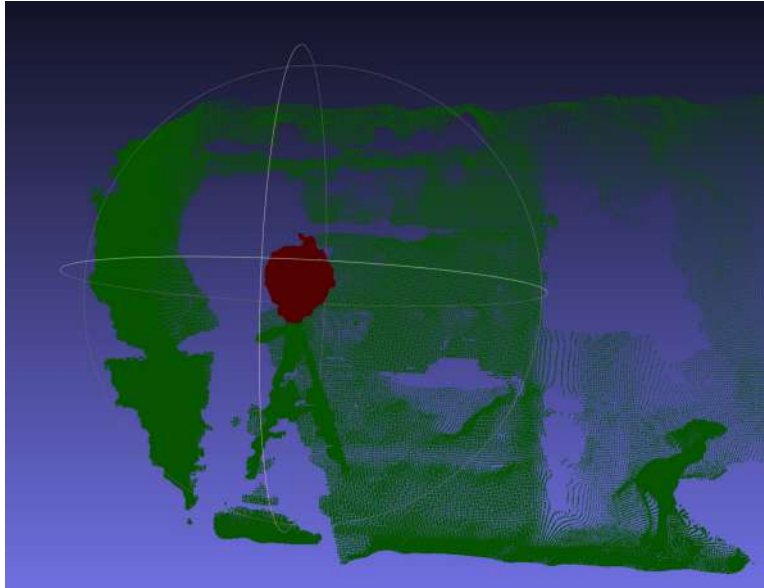


Figura 4.6: Correcta detección en la escena del experimento 2.

Se observa que los falsos positivos pueden incrementar si la tolerancia es mayor y si las muestras en la escena se toman sin la suficiente frecuencia y sin considerar el espacio y dimensiones del objeto a detectar, en este caso en un espacio cercano a la superficie de una esfera con radio de $12cm$. Como se muestra en la Figura 4.7, se observa una detección incorrecta, se iluminan los puntos en 3D falsos positivos en rojo, esta detección cumplió con los criterios de aceptación para considerar como una posible solución, sin embargo es evidente que no existe una esfera en este caso.

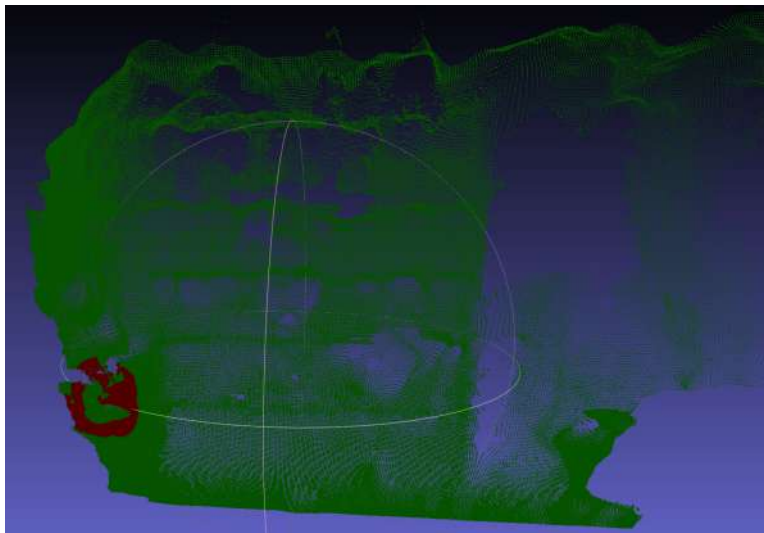


Figura 4.7: Detección incorrecta.

Una diferencia respecto al experimento 1 es la distancia a la que el balón se encuentra respecto a la cámara y una segunda diferencia es la complejidad de la escena donde los

falsos positivos aumentan, sin embargo el método propuesto es robusto ante estas características de la escena.

En un tercer experimento el balón fue posicionado en otra parte de la escena como se muestra en la Figura 4.8 donde también se aprecian diferencias en los valores de profundidad debido a este cambio como se muestra en las Figuras 4.9 y 4.10.



Figura 4.8: Tercer experimento con el balón en una nueva posición.

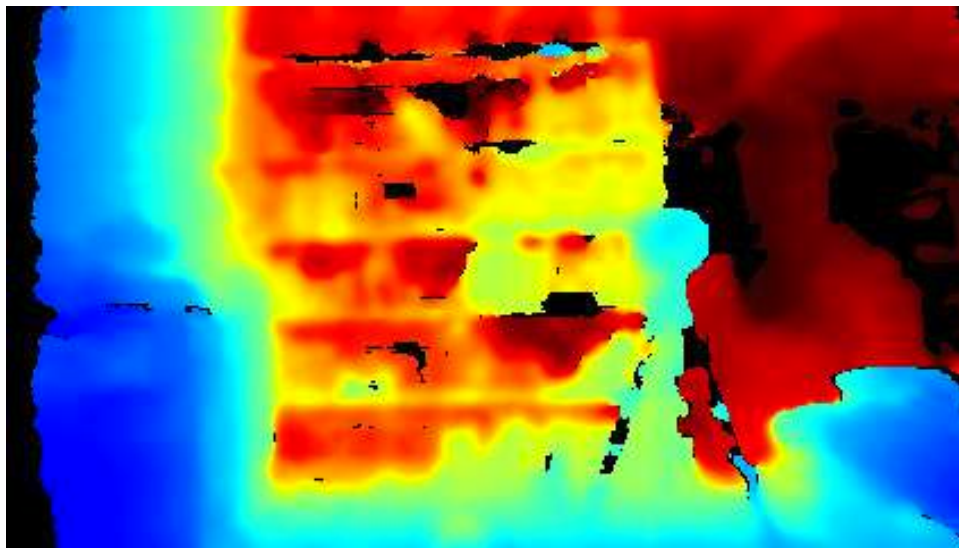


Figura 4.9: Escena en escala de colores jet, se muestra la profundidad de la escena en el experimento 3.

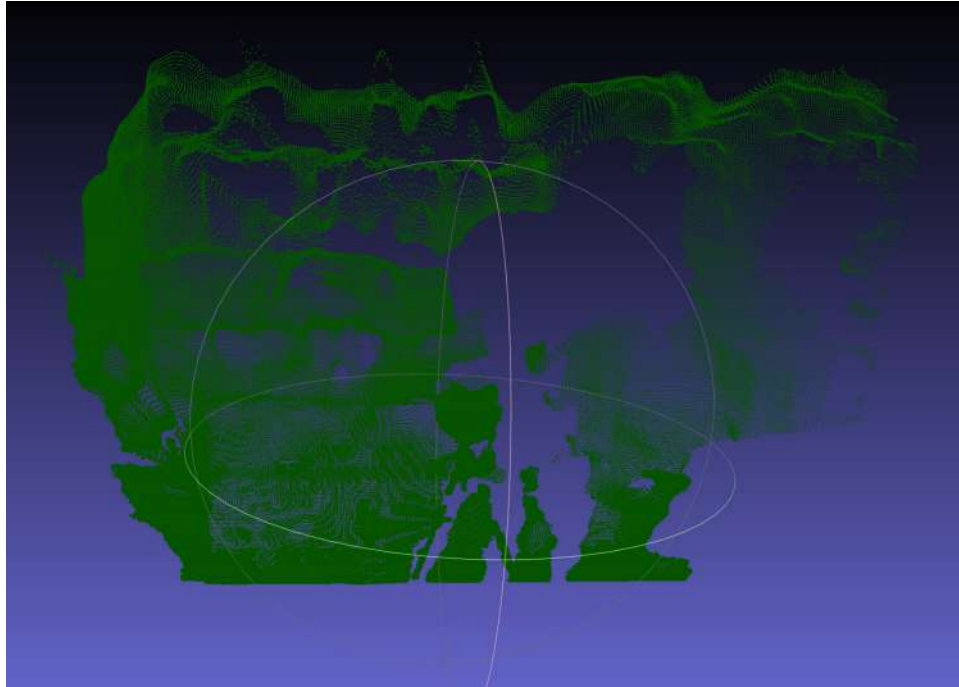


Figura 4.10: Visualización de la escena del experimento 3 sin detectar el balón.

El método propuesto detecta correctamente el balón en la escena como se muestra en la Figura 4.11

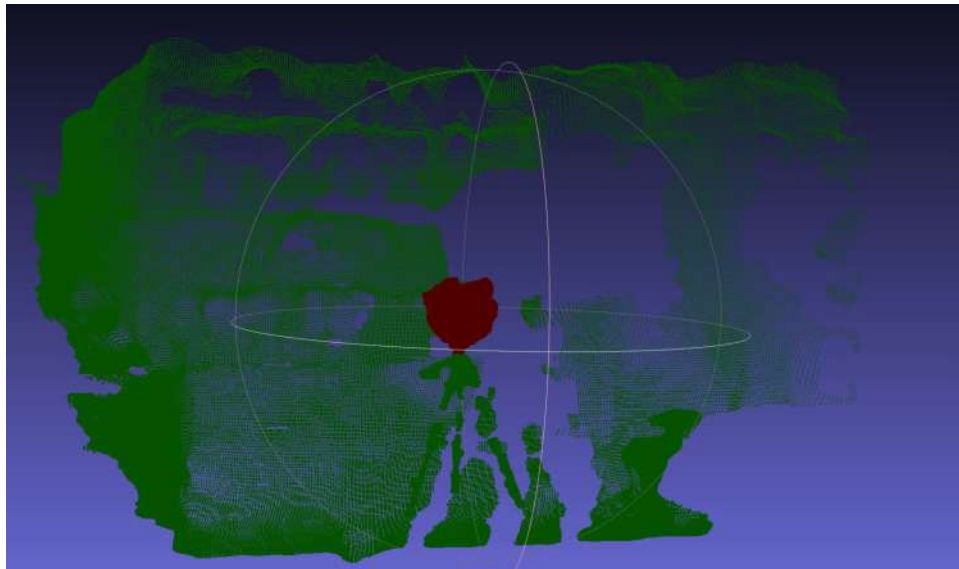


Figura 4.11: Balón de basketball correctamente detectado en el 3er experimento

Una vez que se ha detectado el mejor candidato a esfera en el experimento 3 se procede a ajustar mediante Z-score y se aprecia el resultado en la Figura 4.15.

La Figura 4.12 muestra que el mayor número de elementos se encuentra entre los valores $-1,5$ y $1,5$ en los tres ejes cartesianos.

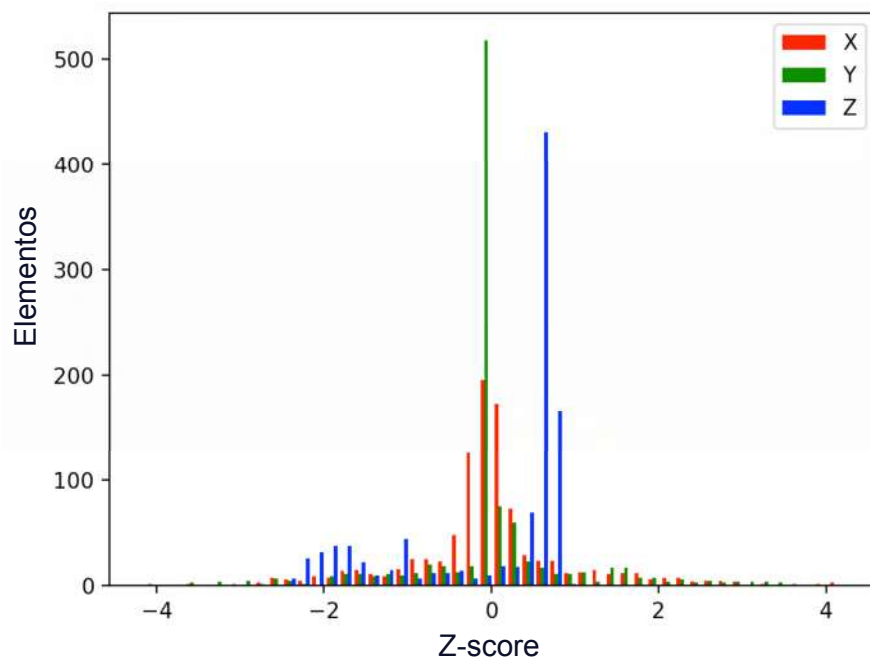


Figura 4.12: Ejemplo de valores Z-score antes de realizar el filtrado y encontrar el barycentro en el experimento 1.

En la Figura 4.13 se observa visualmente la diferencia entre valores atípicos que fueron filtrados (en rojo) a los valores que se considerando como aceptados a la superficie de la esfera (azules).

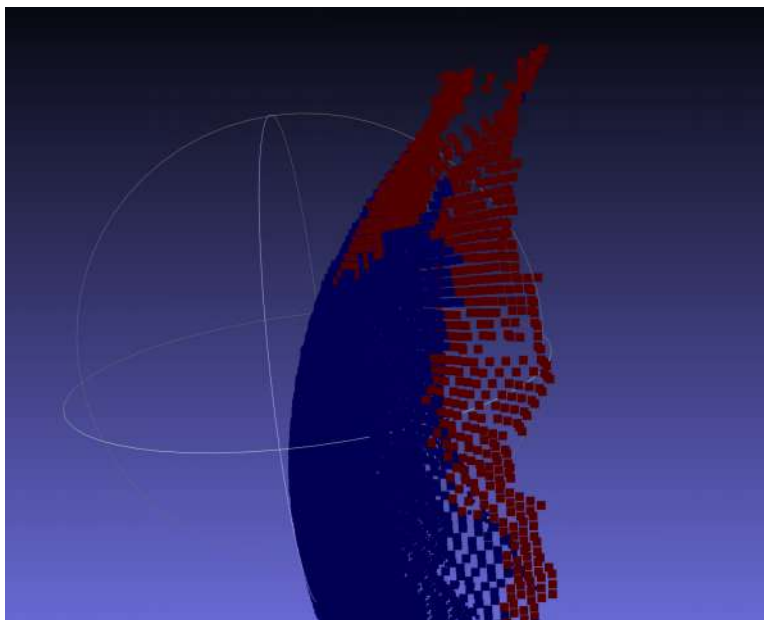


Figura 4.13: Acercamiento de la esfera en el experimento 1 con valores a remover con Z-score en rojo, visualización mediante Meshlab.

Tabla 4.1: Error RMSE en los centros detectados con RANSAC y ajustados con Z-score

Escena y umbral z	RMSE RANSAC	RMSE centro RANSAC puntos ajustados	RMSE Barycentro puntos ajustados
E1 z 3	0.007343368	0.007343368	0.032156230
E1 z 2	0.007343368	0.007343368	0.020370757
E1 z 1.5	0.007343368	0.007343368	0.004612417
E2 z 1.5	0.366534813	0.357131966	0.041731429
E3 z 1.5	0.033177435	0.033539393	0.027763554

La Tabla 4.1 muestra el RGB-D obtenido durante el experimento 1 en metros, observe que una vez que se han removido los valores atípicos, un nuevo valor aún menor de RGB-D puede ser obtenido comparando las distancias al nuevo centro que se obtiene calculando el barycentro con los puntos filtrados.

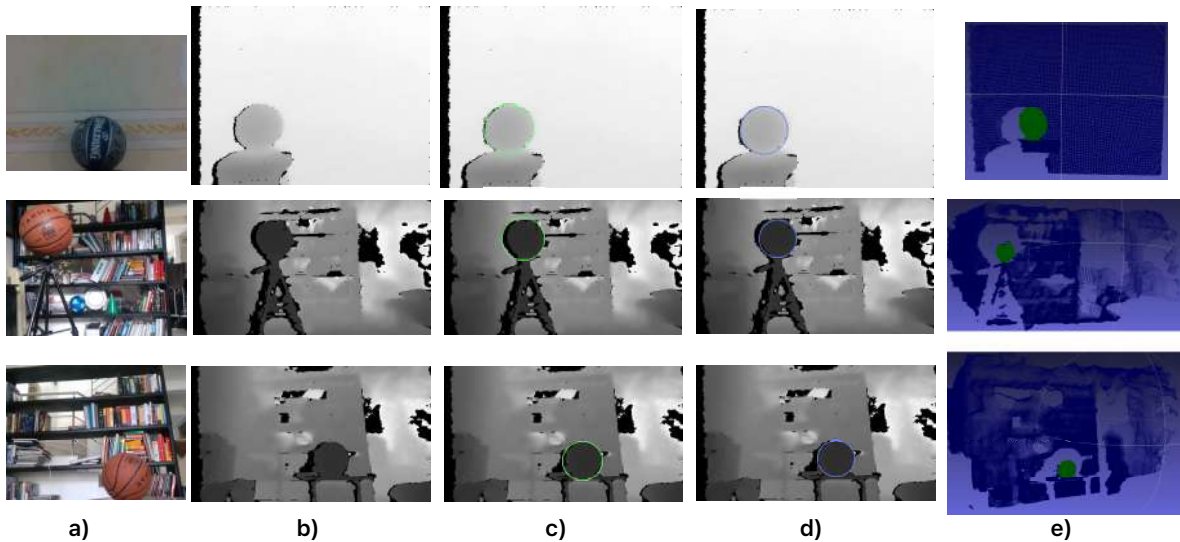


Figura 4.14: a) Escena RGB, b) puntos 3D reprojectados a una imagen en 2D, c) detección CircleNet, d) Detección mediante el método propuesto reprojectando un círculo a una imagen 2D, e) Detección mediante el método propuesto en 3D.

Para poder comparar contra el estado del arte que está basado en redes neuronales (Nguyen et al., 2022), se creó un conjunto de datos de 200 imágenes reprojectando nubes de puntos 3D, estas imágenes contienen 2 principales tipos de escenas, el primer tipo contiene un área clara y limpia de objetos con una pared plana al fondo y el segundo tipo de escena contiene múltiples objetos y diferentes profundidades en la escena como se muestra en la Figura 4.14. Todas las imágenes fueron enmascaradas de manera manual para obtener valores Circle Intersection Over Union por sus siglas en inglés (cIOU) tal como se muestra en (H. Yang et al., 2020). Mean Average Precision por sus siglas en inglés (mAP) (Everingham et al., 2009) se considera en la Tabla 4.2 como métrica. Para obtener los valores cIOU en el método propuesto, el centro de la esfera fue reprojectado de 3D a 2D y por medio del radio

de la esfera, el centro también fue reproyectado para calcular la circunferencia de los círculos dibujados en color azul en la Figura 4.14.

Tabla 4.2: Comparación respecto a otros métodos

Método	mAP	mAP.50cIOU	mAP.75cIOU
CircleNet-HG	0.491	0.843	0.512
SphereDetection (nuestra propuesta)	0.512	0.894	0.529

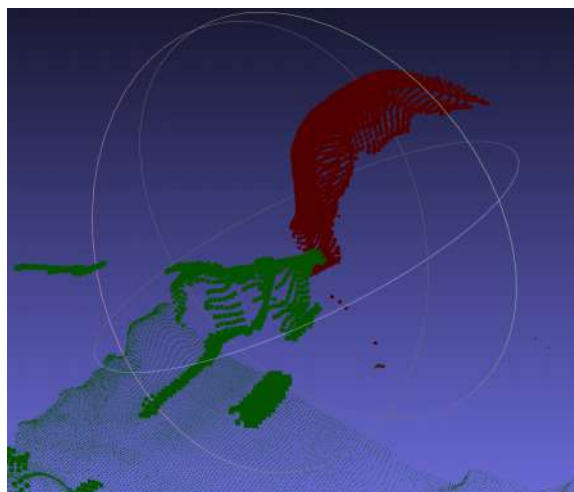


Figura 4.15: Se observa la diferencia entre el tripié y el balón debido al filtrado de valores atípicos mediante z-core

En experimentos posteriores se observa que la detección del balón es robusta siempre y cuando el balón sea visible y no importando que existan otros objetos en escena como en la Figura 4.16.

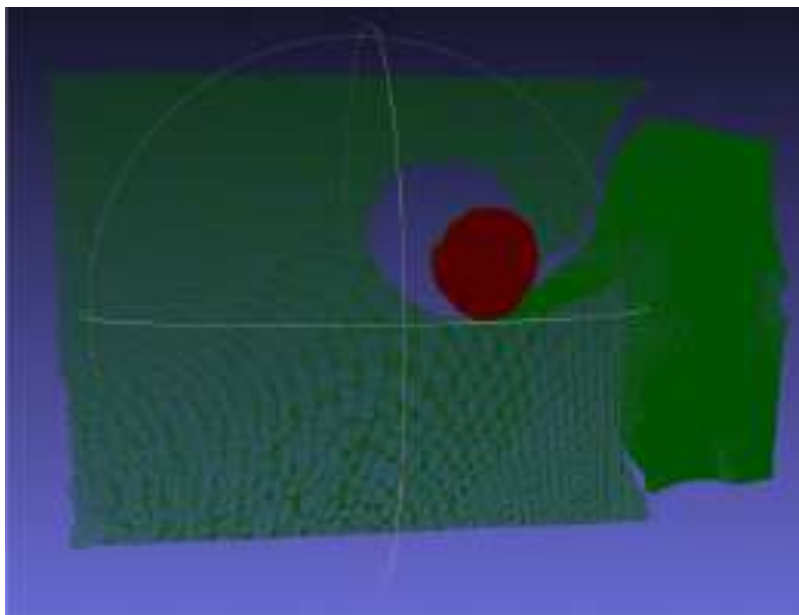


Figura 4.16: Experimento adicional sosteniendo el balón.

4.2. Calibración de la cámara RGB-D

En los experimentos las comparaciones realizadas con (A. N. Staranowicz et al., 2015) se utilizaron un mínimo de imágenes mediante la herramienta libre proporcionada en el trabajo de referencia. La toma de las muestras requirió tomar imágenes de manera consecutiva y el balón de basketball tenía que quedarse quieto entre cada toma de pares de imágenes color-profundidad. La corrección del error geométrico fue mandatoria en todos los métodos para la mejora de los resultados. Un detalle importante es que las muestras recomendadas en el método (A. N. Staranowicz et al., 2015) es arriba de 130 para producir un mejor resultado tal como se reporta en su trabajo, al costo de incrementar notablemente la dificultad de ejecutar el protocolo de calibración. Cuando se comparan las tareas del operador, la toma de cada imagen requiere de la intervención manual posterior en el trabajo de referencia para asegurar una correcta detección de la elipse/círculo en la imagen a color y en la de profundidad. El método propuesto realiza una búsqueda en la imagen completa tanto en la capa de color gracias a (Lu et al., 2019) como en la imagen de profundidad debido al método propuesto para la detección de una esfera utilizando RANSAC, ambas estrategias no requieren de una intervención manual del operador del protocolo de calibración simplificando el mismo. Se tomaron mas muestras de las necesarias para obtener un número mínimo de muestras válidas en el método de (A. N. Staranowicz et al., 2015), sin embargo la propuesta realizada requiere menos pasos y muestras en comparación. Los experimentos realizados usaron escenas con condiciones no ideales, con iluminación natural con luz solar indirecta, la escena contiene texturas no regulares en el fondo. El método propuesto muestra una capacidad superior para tolerar condiciones no ideales.

Tabla 4.3: Parámetros Intrínsecos cámara capa RGB .

Parámetros	F_x	F_y	C_x	C_y	$Skew$
valor RGB de fábrica	421.46	421.46	461.683	236.524	0.0
Zhang et.al.	556.28364	555.65570	325.71797	253.41162	0.0
Staranowicz et. al.	549.1128	550.8689	322.2678	248.1722	-2.126733

En la Tabla 4.3 se muestran los parámetros intrínsecos de fábrica de la cámara RGB-D, se observa que tanto el trabajo de (A. N. Staranowicz et al., 2015) como la propuesta requieren ajustar dichos parámetros, en nuestro caso se utilizar (C. Zhang & Zhang, 2014), se observa también que los parámetros de fábrica se encuentran lejos de los requeridos para los resultados presentados.

Tabla 4.4: Parámetros de rotación cámara RGB-D.

Parámetro	X	Y	Z	W
Fábrica	0.0005242	0.0002236	0.0044973	-0.9999897
Fathian et.al.	0.4188	0.6221	0.5903	-0.2985
Staranowicz et. al.	-0.0089974	-0.0025626	-0.0029216	0.999952

Tabla 4.5: Parámetros de traslación cámara RGB-D.

Parámetro	X	Y	Z
Fábrica	0.0146951	-0.000142742	0.000202387
Fathian et.al.	-0.0811	0.2968	0.4095
Staranowicz et. al.	0.013768	-0.0098439	0.011041

Los valores de rotación y traslación se muestran en las Tablas 4.4 y 4.5, es importante notar que los valores generados por el método QUEsT (Fathian et al., 2017) se muestran en una escala respecto al tamaño real de la escena por lo que se requiere interpolar por medio de un valor para aproximar la proyección 3D a una imagen en 2D, para este propósito se utilizó una regresión spline como se muestra en la Figura 4.17 para interpolar el error de profundidad con la escala de valores producidos por el método QUEsT.

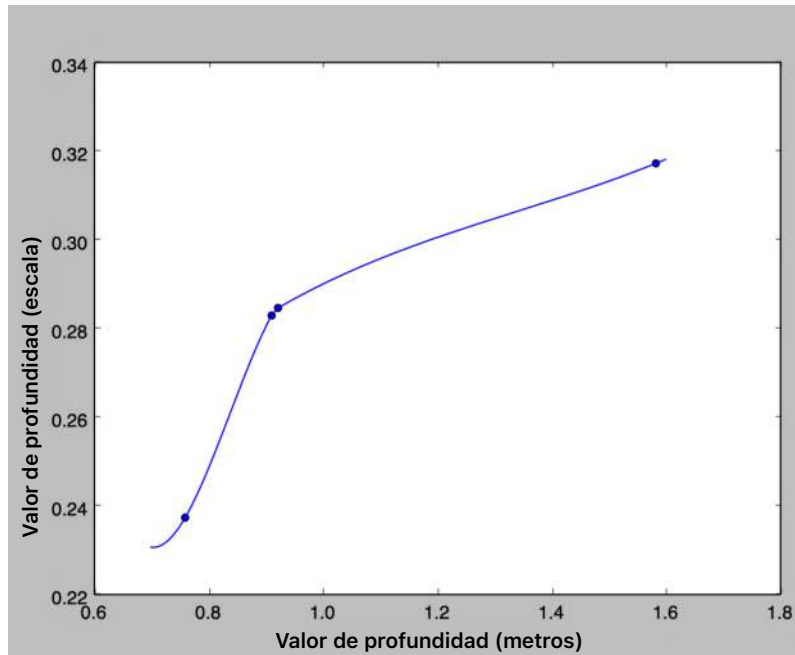


Figura 4.17: Valores de profundidad arrojados por QUEsT method (Fathian et al., 2018) ajustado por medio de spline.

Observese que la línea ajustada no muestra un desplazamiento uniforme de los valores de profundidad este comportamiento es esperado (Rosin et al., 2019).

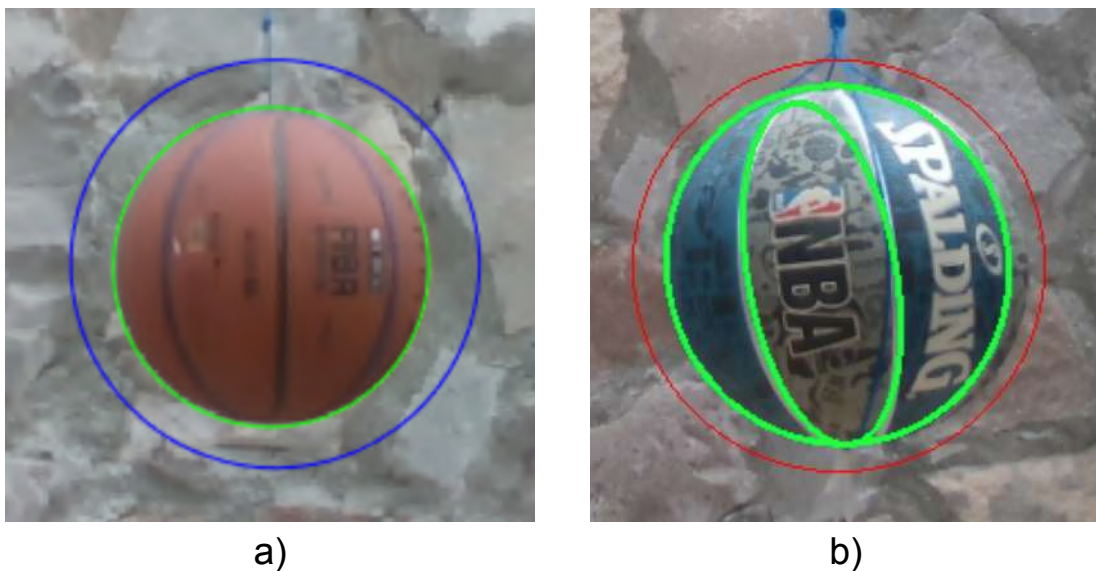


Figura 4.18: Comparativo cualitativo respecto al trabajo de (A. N. Staranowicz et al., 2015) a) contra la propuesta realizada b)

En la Figura 4.18 se observa la proyección de la detección del círculo en el trabajo de (A. N. Staranowicz, s.f.) en color azul (a); y la proyección de la detección de la esfera en nuestra propuesta en rojo (b), observese que en nuestro caso el círculo se acerca más a la

circunferencia del balón respecto al trabajo de referencia mostrando un resultado favorable con una cantidad de imágenes mucho menor.

4.3. Valores atípicos y ruido en nubes de puntos 3D

Se realizan dos inferencias en la misma nube de puntos para procesar y eliminar valores atípicos y ruido con el objetivo de mostrar de manera cualitativa los resultados comparados contra el modelo base PointCleanNet (Rakotosaona et al., 2020) tal como se muestra en la Figura 4.19 se observa la inferencia mediante PointCleanNet.

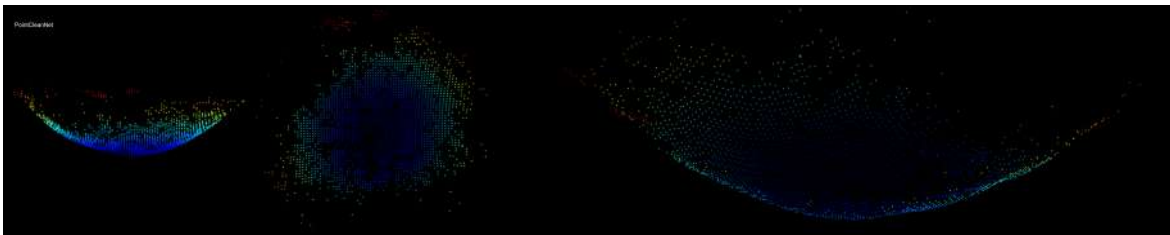


Figura 4.19: Inferencia con PointCleanNet.

En seguida se realiza una inferencia con el modelo neuronal propuesto, se observa en la Figura 4.20

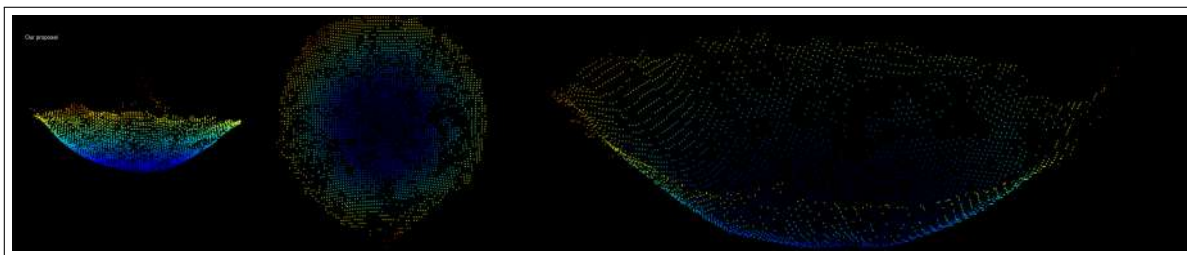


Figura 4.20: Inferencia con el modelo propuesto

Observese la diferencia contra la nube de puntos patrón en la Figura 4.21.

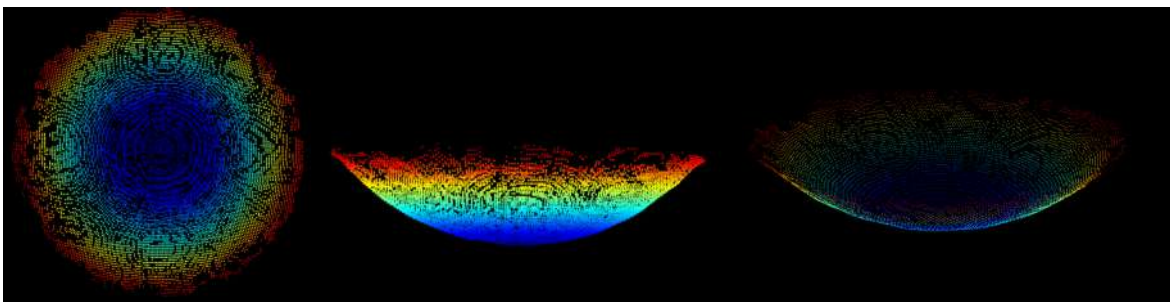


Figura 4.21: Inferencia con la nube de puntos patrón

Se realizan diferentes diferencias con bajas densidades de puntos como se muestra en la Tabla 4.6

Tabla 4.6: Inferencias en nubes de puntos 3D con baja densidad de elementos

Nube de puntos	Total valores típicos	Total valores atípicos
0	15843	1584
1	15843	3168
2	10625	1062
3	10625	125
4	5296	529
5	5296	1059
6	3916	391
7	3916	783
8	3048	304
9	3048	609

Las nubes de puntos mostradas en la Figura 4.22 se reconstruyeron despues de realizar inferencias mediante PointCleanNet, Luo,S y otros y nuestro modelo.

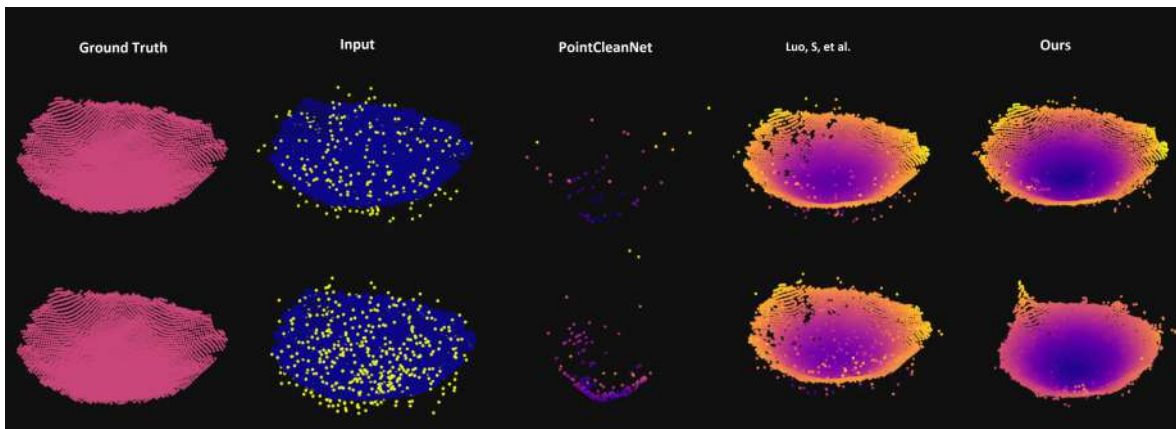


Figura 4.22: Comparación cualitativa contra otros métodos en el estado del arte.

La primer nube de puntos contiene 10,750 puntos, de los cuales 125 son valores atípicos con una desviación estandard del 10%. La segunda nube de puntos contiene 3657 puntos de los cuales 609 son valores atípicos con una desviación estandard del 20%. Se observa que PointCleanNet tiene el peor desempeño en comparación de Luo,S y otros y nuestra propuesta utilizando una baja densidad de elementos en la nube de puntos en 3D. La exactitud de PointCleanNet es de 16,3% contra 60,6% de nuestro método tal como se muestra en la Tabla 4.7

Tabla 4.7: Comparativo considerando F1-score

Método	Número de nube de puntos	Precisión	Sensitividad	F1-score	Exactitud	Error cuadrático medio
PointCleanNet	0	0.086038	0.105429	0.094752	0.81689	0.18311
	1	0.162519	0.171086	0.166692	0.71495	0.28505
	2	0.095279	0.210923	0.131263	0.74630	0.25370
	3	0.163770	0.199529	0.179890	0.69678	0.30322
	4	0.092794	0.676749	0.163209	0.36979	0.63021
	5	0.171616	0.693107	0.275112	0.39135	0.60865
	6	0.090674	0.987212	0.166093	0.10007	0.89993
	7	0.167146	0.964240	0.284906	0.19345	0.80655
	8	0.090692	1.000000	0.166302	0.09069	0.90931
	9	0.166667	1.000000	0.285714	0.16735	0.83265
Nuestra propuesta	0	0.091530	0.431818	0.151043	0.55879	0.44121
	1	0.164659	0.517677	0.249848	0.48198	0.51802
	2	0.091441	0.412429	0.149692	0.57423	0.42577
	3	0.171462	0.485176	0.253379	0.52345	0.47655
	4	0.096508	0.465028	0.159844	0.55605	0.44395
	5	0.170346	0.488196	0.252565	0.51849	0.48151
	6	0.100656	0.588235	0.171898	0.48549	0.51451
	7	0.170197	0.574713	0.262620	0.46223	0.53777
	8	0.091949	0.634868	0.160633	0.39827	0.60173
	9	0.162986	0.605911	0.256874	0.41619	0.58381

Se muestra de manera cuantitativa la comparación contra el estado del arte en la Tabla 4.8 utilizando la métrica de la distancia de Chamfer. Se puede observar que el modelo propuesto tiene una menor pérdida con respecto a los trabajos de referencia, esto es debido al desplazamiento que la nube de puntos experimenta durante la inferencia, mostrando que nuestro modelo solo afecta aquellos puntos que son clasificados como valores atípicos.

Tabla 4.8: Comparativo considerando Chamfer Loss.

Número de nube de puntos	Pérdida Chamfer PointCleanNe	Pérdida Chamfer Luo, S, et. al.	Pérdida Chamfer propuesta
0	0.07782	0.16389	0.02181
1	0.07782	0.21744	0.02863
2	0.13977	0.20516	0.05945
3	4.55815	0.20388	0.02294
4	0.70367	0.26565	0.04464
5	0.73141	0.20699	0.02136
6	0.11121	0.18111	0.02035
7	0.15080	0.24181	0.05381
8	6.05349	0.29295	0.05481
9	2.94945	0.26633	0.05060

Los resultados muestran que el modelo propuesto se desempeña de mejor manera con una cantidad de elementos menor en la nube de puntos 3D u con una mayor proporción de valores atípicos contra valores reales.

4.4. Reconstrucción 3D

4.4.1 Nubes de puntos 3D

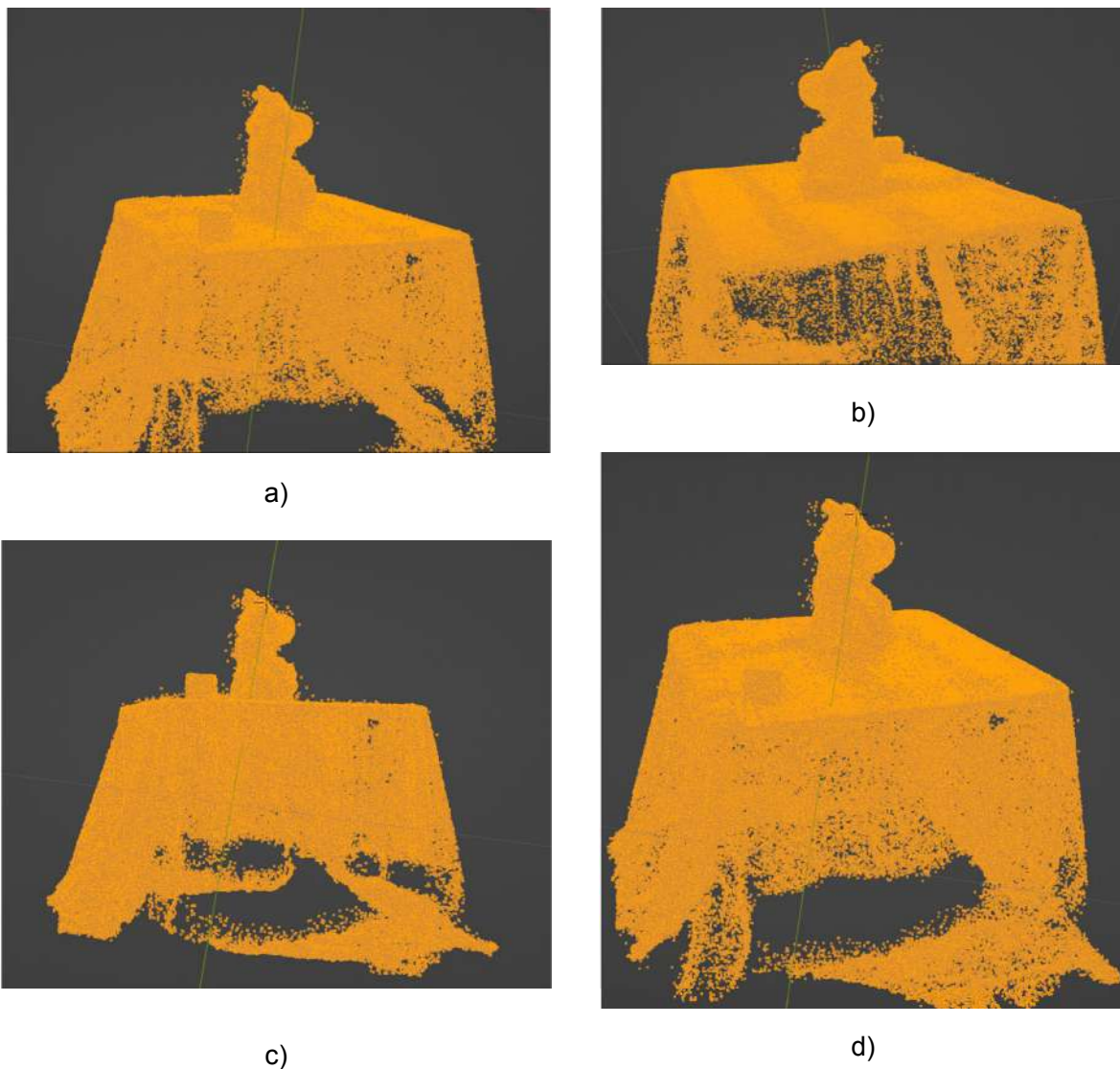
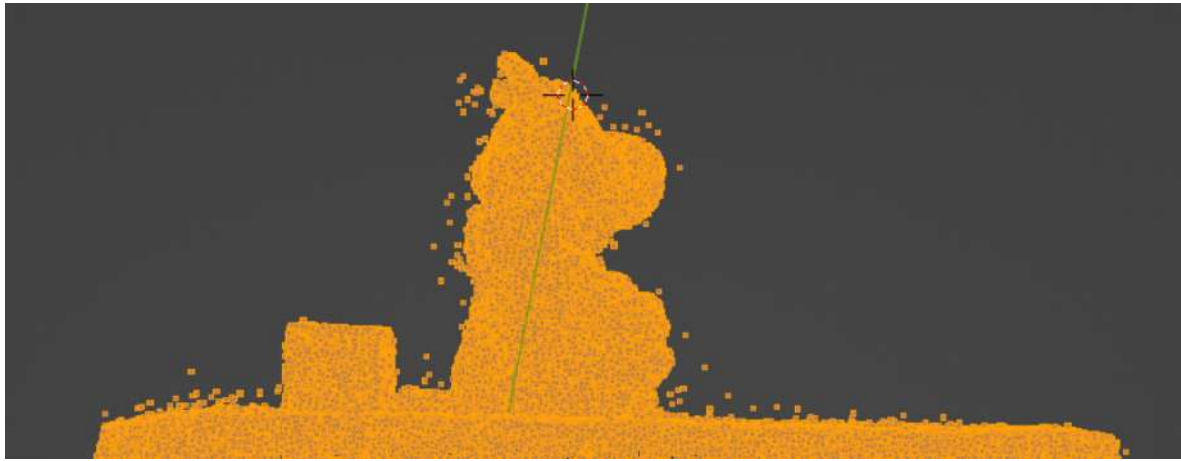
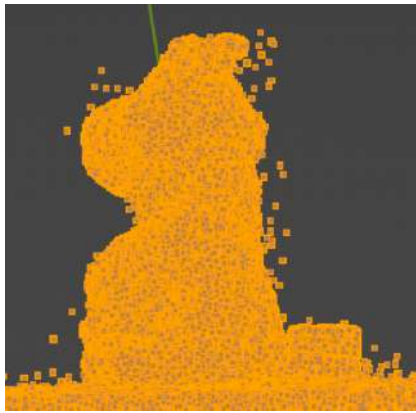


Figura 4.23: Nubes de puntos generadas, datos crudos I

Se muestra en la Figura 4.23 mediante el programa Blender la nube de puntos generados en diferentes perspectivas, en el inciso a) se aprecia un cubo de rubik al frente sobre la mesa y mas al fondo la figura del oso, en el inciso b) primero se observa el oso y mas al fondo el cubo de rubik, en los incisos c) y d) se observan desde otra perspectiva la misma nube de puntos con detalles de la mesa y la cobija que la cubría, se alcanzan a apreciar incluso pliegues. Es importante destacar que el oso de peluche se encontraba en movimiento constante al adquirir el conjunto de imágenes para alimentar el método de reconstrucción 3D propuesto.



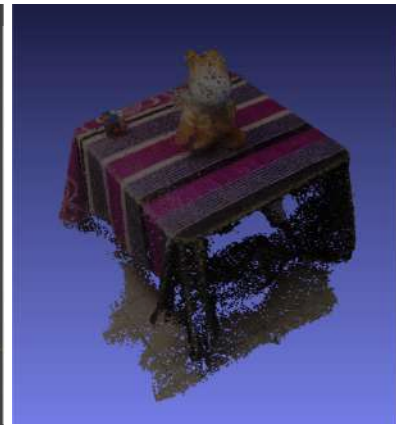
a)



b)



c)



f)

Figura 4.24: Nubes de puntos generadas, datos crudos II

En la Figura 4.24 se observan puntos 3D atípicos con mayor claridad tanto en el modelo del oso de peluche como en la superficie de la mesa en el inciso a), es importante notar que el movimiento del oso de peluche es una fuente de origen para este ruido en la reconstrucción del modelo 3D del objeto en escena. Se aprecia en el inciso b) que las partes del oso de peluche cuyo movimiento fue mas acentuado durante la captura de la escena contienen mayor número de puntos atípicos. En el inciso c) se aprecia una visualización en color negro para facilitar la visualización del ruido únicamente en el oso de peluche. En el inciso f) se observa la nube de puntos con color, notandose que los detalles finos se aprecian con mayor calidad en los objetos que se mantuvieron estáticos en la escena, la mesa con la cobija, el cubo rubik y parte del suelo.

4.4.2 Comparativas

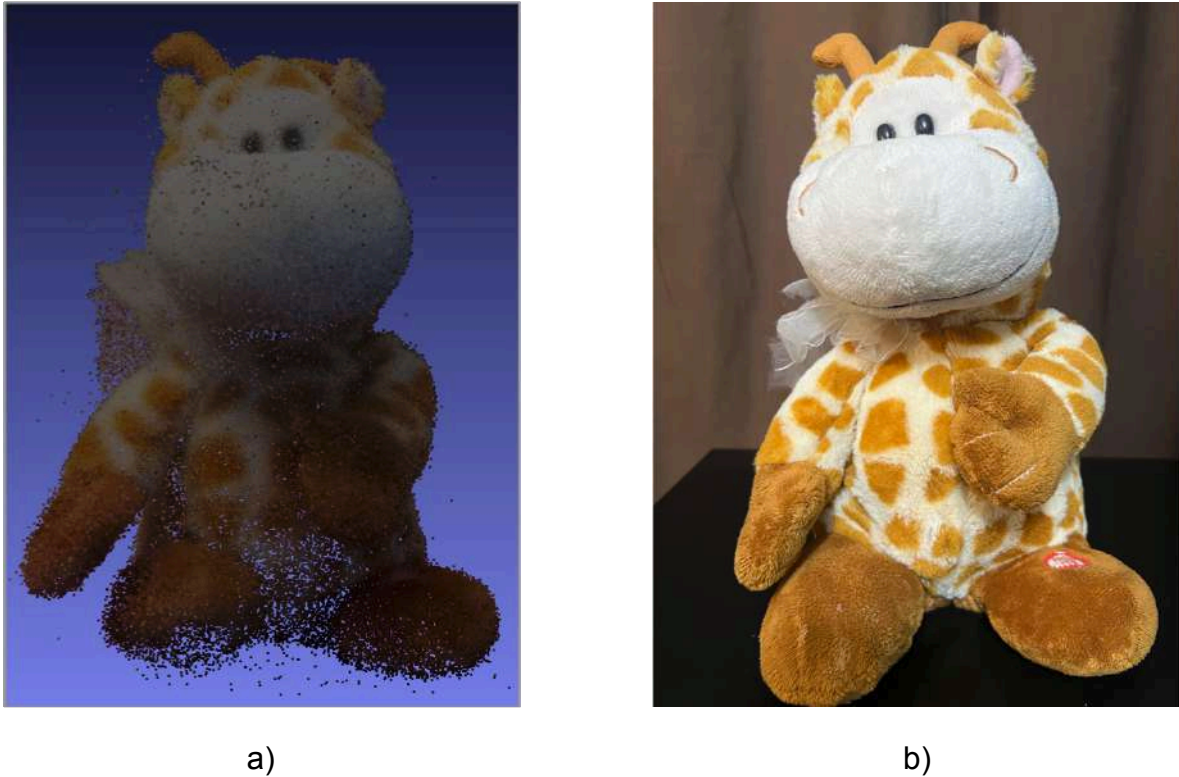
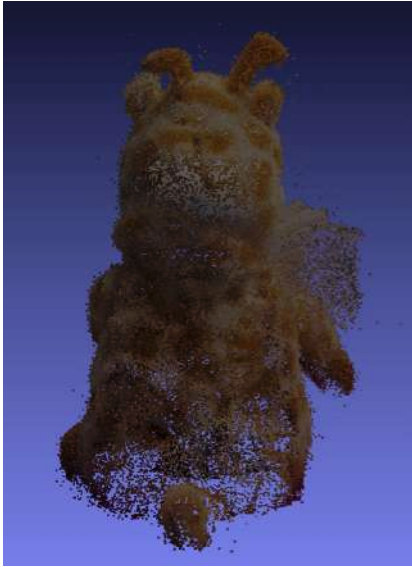


Figura 4.25: Comparativa cualitativa modelo 3D contra objeto real I

Se muestra una comparativa cualitativa entre en la Figura 4.25 entre el modelo 3D generado inciso a) y la imagen del oso de peluche real inciso b). Debido a la naturaleza del método propuesto se conserva información del color del objeto capturado que puede ser desplegado para una mejor visualización, el color se conserva por cada punto como una propiedad adicional en el modelo 3D generado. El inciso a) muestra visualmente una característica del método propuesto en la densidad de los puntos generados en las diferentes áreas o secciones del modelo 3D, en las piernas del oso se aprecia una menor densidad de puntos así como en ciertas partes del rostro, esto no necesariamente es indeseado, ya que como tál se visualizan únicamente puntos y no un modelo 3d de mallas con textura donde no se verían espacios vacíos. Los lugares con menor densidad muestran zonas con menor diversidad de perspectivas en el conjunto de datos generado durante la captura de la escena. La posición de la cabeza, manos y piernas del oso real varían debido a que son las partes móviles y no se quedan en una misma posición. Observese que la sección de la mascada del oso de peluche se observa una dispersión considerable de los puntos en esa área, esto es causa de la transparencia del material de la mascada y muestra una de las ventajas de utilizar NeRF (Mildenhall et al., 2021) como un bloque del método propuesto en este trabajo de investigación. La posición del modelo generado en 3D del inciso a) es la posición canónica la cuál es la primera posición en la secuencia de la captura de la escena, esta primera posición es la que se considera para dar el seguimiento a las deformaciones de la superficie del oso durante la captura de la escena.



a)



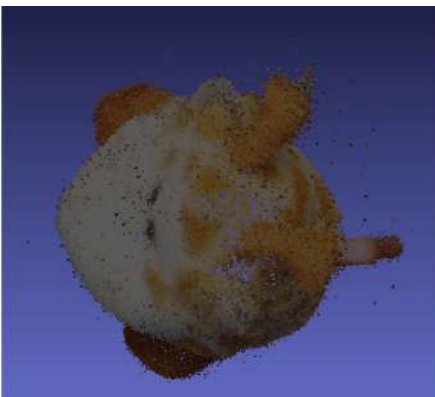
b)



c)



d)



e)



f)

Figura 4.26: Comparativa cualitativa modelo 3D contra objeto real II

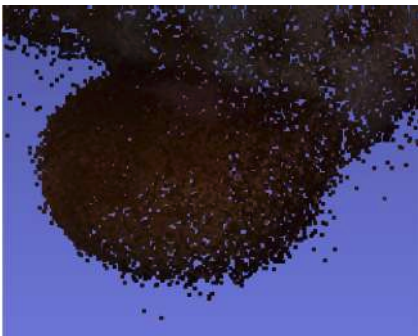
Las condiciones de iluminación al momento de la captura de la escena contra las condiciones de iluminación de la muestra del objeto real son diferentes y se observa en la Figura 4.26 una iluminación mas acentuada en la fotos del objeto real en los incisos b), d), f) contra las fotos del modelo 3D en los incisos a), c) y e).



a)



b)



c)



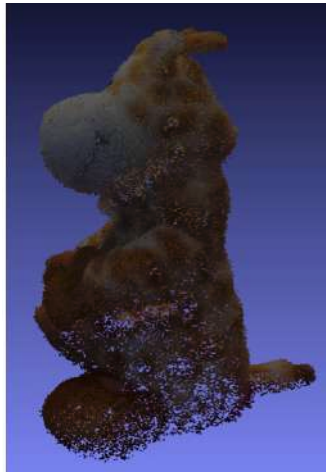
d)

Figura 4.27: Comparativa cualitativa modelo 3D contra objeto real III

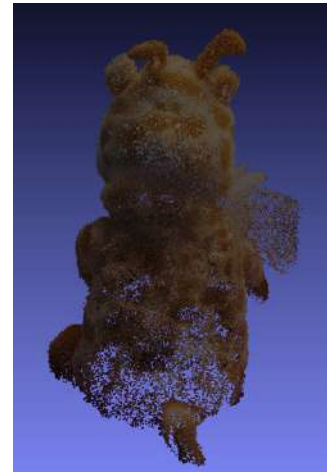
El modelo 3D generado hasta este punto conserva los valores atípicos que el propio proceso genera en sus primeros bloques, por ejemplo se observa que fuera del objeto en los incisos a) y c) de la Figura 4.27 se encuentran los valores atípicos.



a)



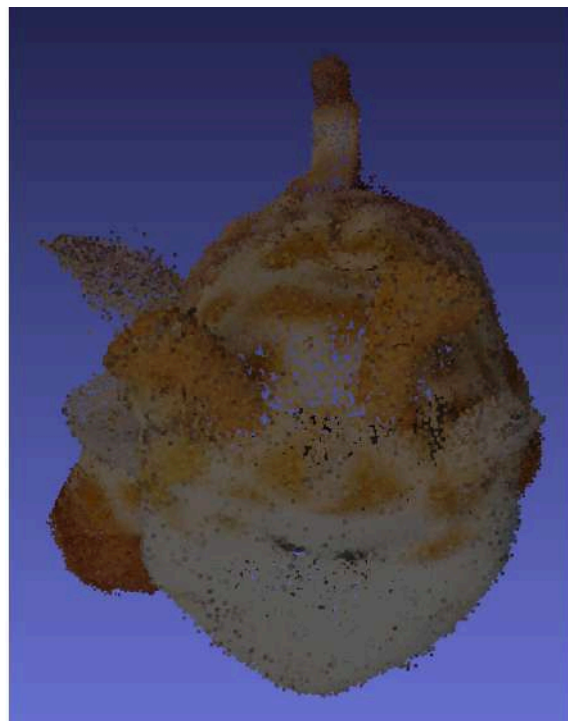
b)



c)



d)



e)

Figura 4.28: Modelo 3D generado filtrando valores atípicos

En el modelo propuesto se utiliza una etapa de limpieza de valores atípicos mediante redes neuronales, como se aprecia en la Figura 4.28 incisos a) al e) se disminuyen los puntos que se encuentran fuera de la superficie del oso de peluche. Nótese que la zona con mayor cantidad de valores atípicos remanentes en el método propuesto después de la etapa de filtrado es en la region de la mascada, las características de transparencia y el movimiento de la mascada durante la captura y la densidad de los puntos aumentan la dificultad de la etapa de filtrado.

4.4.3 Valores 3d a medidas en el mundo real

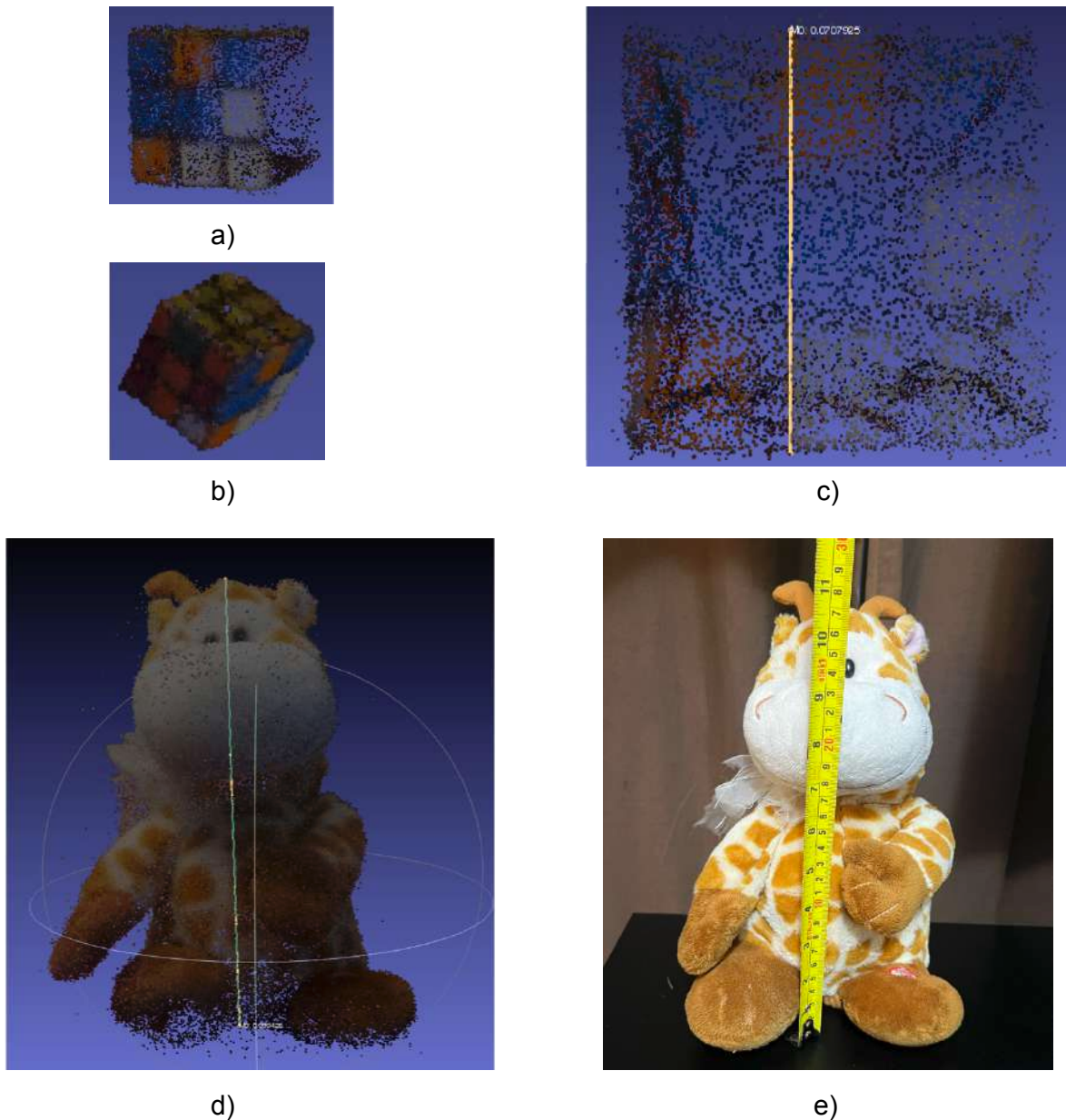


Figura 4.29: Conversión al tamaño del objeto en el mundo real.

Los valores de profundidad en las nubes de puntos se representan mediante una escala de valores que no necesariamente representan una unidad de medida en el mundo real. En el método propuesto y por la naturaleza del protocolo de captura de la escena es posible aproximar la escala a valores de medida en el mundo real mediante un objeto patrón como lo puede ser un balón de basketball o una objeto geométrico que pueda ser identificado fácilmente en escena y cuyas medidas nos permitan tener una escala referencia para convertir los valores de profundidad a medidas en el mundo real.

En la Figura 4.29 se observa en los incisos a) y b) un cubo de rubik como objeto

patrón cuyos lados miden de manera similar aproximadamente 57 milímetros. En el inciso c) se observa un valor numérico aproximado de 0.0707925 y en el inciso d) se observa un valor aproximado en la altura del modelo en 3D del oso de: 0.339428, mediante una regla de 3 se puede obtener una altura en el mundo real aproximada de 276 milímetros. Observese en el inciso e) una medición aproximada de 27 centímetros.

La toma de estas medidas se realizó mediante un proceso manual por lo que se tiene que considerar un error de operación tanto en la captura de la distancia del modelo 3D así como de la altura del oso en 3D, sin embargo se demuestra que se pueden obtener mediciones cercanas a la realidad.

4.4.4 Características del modelo 3D generado

En la Tabla 4.9 se muestran la cantidad de puntos generados en los modelos 3D de la escena con la mesa, el cubo rubik y el oso de peluche, solo el oso de peluche y solo el oso de peluche después de haber filtrado los puntos 3D atípicos.

Tabla 4.9: Características de las nubes de puntos 3D.

Nube de puntos	Puntos de la escena
Escena oso, mesa y cubo rubik	948,436
Modelo 3D oso con ruido	214,741
Modelo 3D oso sin ruido	211,217

La nube de puntos 3D de la escena completa no se muestra, debido a que solo se extrae un cubo de información de la representación de la escena con **NERF!** (**NERF!**) donde solo se contiene el modelo de puntos 3D mostrado en la Figura 4.23.

5. CONCLUSIONES Y TRABAJO FUTURO

5.1. Detección de una esfera representada por un balón de basketball

Se propone un método utilizando algoritmos clásicos de inteligencia artificial donde se detecta un balón de basketball correspondiente a una geometría esférica utilizando el método RANSAC y se propone un modelo matemático para obtener el radio y centro de la esfera a partir de 3 puntos ubicados en la superficie de la geometría a encontrar, el valor predefinido del balón de basketball es usado como un valor objetivo para el modelo matemático propuesto y mediante una tolerancia se convergen diferentes propuestas de solución generadas de manera aleatoria hasta encontrar la solución correcta.

Los experimentos muestran ventajas al utilizar datos en 3D respecto a otros métodos en el estado del arte los cuales buscan en un espacio de 2D.

El método propuesto muestra oportunidades para trabajo futuro como lo es la paralelización del algoritmo en la etapa de muestreo y convergencia, mejoras en la heurística para ajustar tolerancias contra el ruido presente en la escena, la profundidad y la complejidad de la misma.

5.2. Calibración de la cámara RGB-D

Se propone un método para la calibración de cámaras RGB-D que a comparación del método de referencia en el estado del arte utiliza información directamente en la capa 3D y no requiere reproyectar a un espacio en 2D, se requiere menor esfuerzo y complejidad en el protocolo de calibración así como una cantidad menor de muestras de imágenes. Los experimentos realizados muestran que un protocolo de calibración con menor complejidad produce mejores resultados en condiciones no ideales y más cercanos a la realidad.

La capa de profundidad también está sujeta a un error de calibración, en el presente trabajo se utilizó una regresión por medio de spline, en un trabajo futuro se ajustará este tipo de error mediante una estrategia de calibración de manera separada.

5.3. Valores atípicos y ruido en la nube de puntos 3D

Se puede concluir por los experimentos realizados que a una mayor dispersión de ruido, mayor complicación para los modelos neuronales para remover los valores atípicos de las nubes de puntos en 3D, el modelo propuesto mejora la exactitud de las figuras utilizadas en nuestro propio conjunto de datos y con respecto a las propuestas en el estado del arte especialmente en nubes de baja densidad, las cuales permiten trabajar de manera ágil los escenarios presentados ya que requieren de una menor capacidad de cómputo.

Como trabajo futuro se considerarán nubes de puntos de alta densidad, técnicas de aumento de datos en, diferentes distribuciones de ruido, se reducirá la muestra de la nube de puntos explorando técnicas como filtrado por rejilla de voxels.

5.4. Reconstrucción 3D del objeto dinámico en escena

Se propone un método que considera la calidad de los datos recabados mediante el protocolo de captura de la información utilizando una métrica que permite discriminar una buena captura para la reconstrucción de un objeto cuya superficie se está deformando en el transcurso de la toma de datos mediante una sola cámara, se utiliza como bloque básico la arquitectura neuronal NeRF y se utiliza modificaciones a este modelo neuronal para dar seguimiento a las deformaciones del objeto en escena. Se obtiene una reconstrucción 3D del objeto deseado mediante una proyección de nube de puntos mediante la inferencia de la escena con el modelo y se filtran valores atípicos y ruido de la nube de puntos resultante, se obtiene una nube de puntos fiel al tamaño, forma y color al modelo original en el mundo real a pesar de las deformaciones y textura del objeto.

Como trabajo futuro se considerará un estudio de ablación para el entrenamiento de los bloques modificados NeRF propuestos, así como la adaptación del método a otras propuestas de representación de campos de radiación de luminosidad en el estado del arte. Se considerará adaptar la base de datos generada para estudios posteriores y afín a publicar mediante un artículo de manera libre los datos generados en la captura de las escenas. La purga y disminución del tamaño de los distintos bloques de redes neuronales queda pendiente para un trabajo futuro con la finalidad de trasladar la inferencia a dispositivos de menores prestaciones a los utilizados en el presente trabajo de investigación.

BIBLIOGRAFÍA

- Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive science*, 9(1), 147-169.
- Aliev, K.-A., Ulyanov, D., & Lempitsky, V. (2019). Neural Point-Based Graphics. *arXiv preprint arXiv:1906.08240*.
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3), 292.
- Aoki, Y., Goforth, H., Srivatsan, R. A., & Lucey, S. (2019). Pointnetlk: Robust & efficient point cloud registration using pointnet. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7163-7172.
- Bi, S., Kalantari, N. K., & Ramamoorthi, R. (2017). Patch-based optimization for image-based texture mapping. *ACM Trans. Graph.*, 36(4), 106-1.
- Bojanowski, P., Joulin, A., Lopez-Paz, D., & Szlam, A. (2017). Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*.
- Bozic, A., Zollhofer, M., Theobalt, C., & Nießner, M. (2020). Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7002-7012.
- Božič, A., Palafox, P., Zollhöfer, M., Dai, A., Thies, J., & Nießner, M. (2020). Neural Non-Rigid Tracking. *arXiv preprint arXiv:2006.13240*.
- Brachmann, E., & Rother, C. (2019). Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses. *arXiv preprint arXiv:1905.04132*.
- Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4), 18-42.
- Burger, W. (2016). Zhang's camera calibration algorithm: in-depth tutorial and implementation. *HGB16-05*, 1-6.
- Cignoni, P., Ranzuglia, G., Callieri, M., Corsini, M., Ganovelli, F., Pietroni, N., Tarini, M., et al. (2011). MeshLab.
- Conrady, A. (1919). Lens-systems, decentered. *Monthly notices of the royal astronomical society*, 79, 384-390.
- Darwish, W., Tang, S., Li, W., & Chen, W. (2017). A New Calibration Method for Commercial RGB-D Sensors. *Sensors*, 17(6), 1204. <https://doi.org/10.3390/s17061204>
- Derpanis, K. G. (2010). Overview of the RANSAC Algorithm. *Image Rochester NY*, 4(1), 2-3.
- Dou, M., Taylor, J., Kohli, P., Tankovich, V., Izadi, S., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S. R., Kowdle, A., Escolano, S. O., Rhemann, C., & Kim, D. (2016).

- Fusion4D. *ACM Transactions on Graphics*, 35(4), 1-13. <https://doi.org/10.1145/2897824.2925969>
- Drap, P., & Lefèvre, J. (2016). An exact formula for calculating inverse radial lens distortions. *Sensors*, 16(6), 807.
- Duane, C. B. (1971). Close-range camera calibration. *Photogramm. Eng*, 37(8), 855-866.
- Edelmers, E., Kazoka, D., & Pilmane, M. (2021). Creation of Anatomically Correct and Optimized for 3D Printing Human Bones Models. *Applied System Innovation*, 4(3), 67.
- Enazoe. (2022). *enazoe/camera calibration cpp: c detail implementation of camera calibration*. https://github.com/enazoe/camera_calibration_cpp
- Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., & Zisserman, A. (2009). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2), 303-338. <https://doi.org/10.1007/s11263-009-0275-4>
- Explaining Homogeneous Coordinates and Projective Geometryx. (1970). <https://www.tomdalling.com/blog/modern-opengl/explaining-homogenous-coordinates-and-projective-geometry/>
- Fathian, K., Ramirez-Paredes, J. P., Doucette, E. A., Curtis, J. W., & Gans, N. R. (2017). Quaternion based camera pose estimation from matched feature points. *arXiv preprint arXiv:1704.02672*.
- Fathian, K., Ramirez-Paredes, J. P., Doucette, E. A., Curtis, J. W., & Gans, N. R. (2018). Quest: A quaternion-based approach for camera motion estimation from minimal feature points. *IEEE Robotics and Automation Letters*, 3(2), 857-864.
- Firman, M. (2016). RGBD datasets: Past, present and future. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 19-31.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381-395.
- Fridovich-Keil, S., Meanti, G., Warburg, F. R., Recht, B., & Kanazawa, A. (2023). K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. *CVPR*.
- Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2), 119-130.
- Gao, H., Li, R., Tulsiani, S., Russell, B., & Kanazawa, A. (2022a). Monocular Dynamic View Synthesis: A Reality Check. *NeurIPS*.
- Gao, H., Li, R., Tulsiani, S., Russell, B., & Kanazawa, A. (2022b). Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35, 33768-33780.
- Giancola, S., Valenti, M., & Sala, R. (2018). *A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies*. Springer.
- Guerrero, P., Kleiman, Y., Ovsjanikov, M., & Mitra, N. J. (2018). Pcpnet learning local shape properties from raw point clouds. *Computer graphics forum*, 37(2), 75-85.
- Guo, K., Xu, F., Yu, T., Liu, X., Dai, Q., & Liu, Y. (2017). Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Transactions on Graphics (TOG)*, 36(3), 32.

- Han, J., Shao, L., Xu, D., & Shotton, J. (2013). Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5), 1318-1334.
- Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., & Cipolla, R. (2016). Understanding real world indoor scenes with synthetic data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4077-4085.
- Handa, A., Whelan, T., McDonald, J., & Davison, A. J. (2014). A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. *2014 IEEE international conference on Robotics and automation (ICRA)*, 1524-1531.
- Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786), 504-507.
- Hu, Y., Li, W., Wright, D., Aydin, O., Wilson, D., Maher, O., & Raad, M. (2019). Artificial Intelligence Approaches. *Geographic Information Science & Technology Body of Knowledge*, 2019(Q3). <https://doi.org/10.22224/gistbok/2019.3.4>
- Huang, X., Zhang, Y., & Xiong, Z. (2021). High-speed structured light based 3D scanning using an event camera. *Optics Express*, 29(22), 35864-35876.
- Innmann, M., Zollhöfer, M., Nießner, M., Theobalt, C., & Stamminger, M. (2016). VolumeDeform: Real-time volumetric non-rigid reconstruction. *European Conference on Computer Vision*, 362-379.
- Ioannidou, A., Chatzilari, E., Nikolopoulos, S., & Kompatsiaris, I. (2017). Deep learning advances in computer vision with 3d data: A survey. *ACM Computing Surveys (CSUR)*, 50(2), 20.
- Kajiya, J. T., & Von Herzen, B. P. (1984). Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3), 165-174.
- Keselman, L., Woodfill, J. I., Grunnet-Jepsen, A., & Bhowmik, A. (2017). Intel (r) realsense (tm) stereoscopic depth cameras. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1267-1276.
- LeCompte, M. C., Chung, S. A., McKee, M. M., Marshall, T. G., Frizzell, B., Parker, M., Blackstock, A. W., & Farris, M. K. (2019). Simple and Rapid Creation of Customized 3-dimensional Printed Bolus Using iPhone X True Depth Camera. *Practical radiation oncology*.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Lin, H., Peng, S., Xu, Z., Yan, Y., Shuai, Q., Bao, H., & Zhou, X. (2022). Efficient Neural Radiance Fields for Interactive Free-viewpoint Video. *SIGGRAPH Asia Conference Proceedings*.
- Liu, H., Qu, D., Xu, F., Zou, F., Song, J., & Jia, K. (2020). Approach for accurate calibration of RGB-D cameras using spheres. *Optics Express*, 28(13), 19058-19073.
- Lu, C., Xia, S., Shao, M., & Fu, Y. (2019). Arc-support line segments revisited: An efficient high-quality ellipse detection. *IEEE Transactions on Image Processing*, 29, 768-781.
- Lynch, K. M., & Park, F. C. (2017). *Modern robotics*. Cambridge University Press.
- Ma, D., Cao, J., & Chen, Z. (2023). Point Cloud Rendering via Multi-plane NeRF. *Computer Graphics International Conference*, 199-210.

- Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., & Duckworth, D. (2021). Nerf in the wild: Neural radiance fields for unconstrained photo collections. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7210-7219.
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1), 99-106.
- Minsky, M., & Papert, S. (1969). An introduction to computational geometry. *Cambridge tracts in computer science*, HIT.
- Munaro, M., Basso, F., & Menegatti, E. (2012). People tracking within groups with RGB-D data. *Intelligent Robots and Systems (IROS)*.
- Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., & Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 342-347.
- Nam, G., Wu, C., Kim, M. H., & Sheikh, Y. (2019). Strand-Accurate Multi-View Hair Capture. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 155-164.
- Neupane, C., Koirala, A., Wang, Z., & Walsh, K. B. (2021). Evaluation of depth cameras for use in fruit localization and sizing: Finding a successor to kinect v2. *Agronomy*, 11(9), 1780.
- Newcombe, R. A., Fox, D., & Seitz, S. M. (2015). Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 343-352.
- Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., & Fitzgibbon, A. W. (2011). Kinectfusion: Real-time dense surface mapping and tracking. *ISMAR*, 11, 127-136.
- Nguyen, E. H., Yang, H., Deng, R., Lu, Y., Zhu, Z., Roland, J. T., Lu, L., Landman, B. A., Fogo, A. B., & Huo, Y. (2022). Circle Representation for Medical Object Detection. *IEEE Transactions on Medical Imaging*, 41(3), 746-754. <https://doi.org/10.1109/tmi.2021.3122835>
- Orts-Escalano, S., Dou, M., Tankovich, V., Loop, C., Cai, Q., Chou, P. A., Mennicken, S., Valentin, J., Pradeep, V., Wang, S., Kang, S. B., Rhemann, C., Kohli, P., Lutchyn, Y., Keskin, C., Izadi, S., Fanello, S., Chang, W., Kowdle, A., ... Khamis, S. (2016). Holoportation. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*. <https://doi.org/10.1145/2984511.2984517>
- Park, K., Sinha, U., Barron, J. T., Bouaziz, S., Goldman, D. B., Seitz, S. M., & Martin-Brualla, R. (2021). Nerfies: Deformable neural radiance fields. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5865-5874.
- Rakotosaona, M.-J., La Barbera, V., Guerrero, P., Mitra, N. J., & Ovsjanikov, M. (2020). Pointcleannet: Learning to denoise and remove outliers from dense point clouds. *Computer graphics forum*, 39(1), 185-203.

- Rodríguez, I. S., Ortega, J. C. P., Rogelio, L., Rivera, R., Arreguín, J. M. R., & Hurtado, E. G. (s.f.). Caracterización de valores atípicos en nube de puntos en 3D para la reducción del tiempo de ejecución en memoria.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408. <https://doi.org/10.1037/h0042519>
- Rosin, P. L., Lai, Y.-K., Shao, L., & Liu, Y. (2019). *RGB-D Image Analysis and Processing*. Springer.
- Salgado, C. M., Azevedo, C., Proença, H., & Vieira, S. M. (2016). Noise versus outliers. *Secondary analysis of electronic health records*, 163-183.
- Schnabel, R., Wahl, R., & Klein, R. (2007). Efficient RANSAC for point-cloud shape detection. *Computer graphics forum*, 26(2), 214-226.
- Schönberger, J. L., & Frahm, J.-M. (2016). Structure-from-Motion Revisited. *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor segmentation and support inference from rgb-d images. *European Conference on Computer Vision*, 746-760.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489. <https://doi.org/10.1038/nature16961>
- Singh, A., Sha, J., Narayan, K. S., Achim, T., & Abbeel, P. (2014). Bigbird: A large-scale 3d database of object instances. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 509-516.
- Slavcheva, M., Baust, M., Cremers, D., & Ilic, S. (2017). Killingfusion: Non-rigid 3d reconstruction without correspondences. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1386-1395.
- Song, S., & Xiao, J. (2013). Tracking revisited using RGBD camera: Unified benchmark and baselines. *Proceedings of the IEEE international conference on computer vision*, 233-240.
- Staranowicz, A., Brown, G. R., Morbidi, F., & Mariottini, G. L. (2013). Easy-to-use and accurate calibration of rgb-d cameras from spheres. *Pacific-Rim Symposium on Image and Video Technology*, 265-278.
- Staranowicz, A., Brown, G. R., Morbidi, F., & Mariottini, G. L. (2014). Easy-to-Use and Accurate Calibration of RGB-D Cameras from Spheres.
- Staranowicz, A. N. (s.f.). astaranowicz/DCCT: Depth-Camera Calibration Toolbox (RGB-D Calibration ToolBox). <https://github.com/astaranowicz/DCCT>
- Staranowicz, A. N., Brown, G. R., Morbidi, F., & Mariottini, G.-L. (2015). Practical and accurate calibration of RGB-D cameras using spheres. *Computer Vision and Image Understanding*, 137, 102-114.
- Structure by Occipital - Give Your iPad 3D Vision. (s.f.). <https://structure.io/structure-core/specs>
- Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.

- Tordoff, B., & Murray, D. W. (2004). The impact of radial distortion on the self-calibration of rotating cameras. *Computer Vision and Image Understanding*, 96(1), 17-34.
- Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4), 323-344.
- Visa, S., Ramsay, B., Ralescu, A. L., & Van Der Knaap, E. (2011). Confusion matrix-based feature selection. *MAICS*, 710, 120-127.
- Wasenmüller, O., Meyer, M., & Stricker, D. (2016). CoRBS: Comprehensive RGB-D benchmark for SLAM using Kinect v2. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1-7.
- Wilmanski, M., Kreucher, C., & Lauer, J. (2016). Modern approaches in deep learning for SAR ATR. *Algorithms for synthetic aperture radar imagery XXIII*, 9843, 98430N.
- Wu, T., Pan, L., Zhang, J., Wang, T., Liu, Z., & Lin, D. (2021). Density-aware chamfer distance as a comprehensive metric for point cloud completion. *arXiv preprint arXiv:2111.12702*.
- Yang, H., Deng, R., Lu, Y., Zhu, Z., Chen, Y., Roland, J. T., Lu, L., Landman, B. A., Fogo, A. B., & Huo, Y. (2020). CircleNet: Anchor-Free Glomerulus Detection with Circle Representation. En *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020* (pp. 35-44). Springer International Publishing. https://doi.org/10.1007/978-3-030-59719-1_4
- Yang, J., Li, H., Campbell, D., & Jia, Y. (2015). Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11), 2241-2254.
- Zhang, C., & Zhang, Z. (2014). Calibration between depth and color sensors for commodity depth cameras. En *Computer vision and machine learning with RGB-D sensors* (pp. 47-64). Springer.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11), 1330-1334.
- Zhou, Q.-Y., & Koltun, V. (2014). Color map optimization for 3D reconstruction with consumer depth cameras. *ACM Transactions on Graphics (TOG)*, 33(4), 155.
- Zollhöfer, M. (2019). Commodity RGB-D sensors: Data acquisition. En *RGB-D Image Analysis and Processing* (pp. 3-13). Springer.
- Zollhöfer, M., Nießner, M., Izadi, S., Rehmann, C., Zach, C., Fisher, M., Wu, C., Fitzgibbon, A., Loop, C., Theobalt, C., et al. (2014a). Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics (ToG)*, 33(4), 156.
- Zollhöfer, M., Nießner, M., Izadi, S., Rehmann, C., Zach, C., Fisher, M., Wu, C., Fitzgibbon, A., Loop, C., Theobalt, C., & et al. (2014b). Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics*, 33(4), 1-12. <https://doi.org/10.1145/2601097.2601165>
- Zollhöfer, M., Stotko, P., Görlitz, A., Theobalt, C., Nießner, M., Klein, R., & Kolb, A. (2018). State of the Art on 3D Reconstruction with RGB-D Cameras. *Computer graphics forum*, 37, 625-652.

APÉNDICES

0.1. Artículo I

Article

Reduced Calibration Strategy Using a Basketball for RGB-D Cameras

Luis-Rogelio Roman-Rivera ^{*,†}, Israel Sotelo-Rodríguez [†], Jesus Carlos Pedraza-Ortega [†],
Marco Antonio Aceves-Fernandez [†], Juan Manuel Ramos-Arreguín [†] and Efrén Gorrostieta-Hurtado [†]

Facultad de Ingeniería, Universidad Autónoma de Querétaro, Cerro de las Campanas S/N, Querétaro C.P. 76010, Mexico; isotelo17@alumnos.uaq.mx (I.S.-R.); carlos.pedraza@uaq.mx (J.C.P.-O.); marco.aceves@uaq.mx (M.A.A.-F.); juan.ramos@uaq.mx (J.M.R.-A.); efrén.gorrostieta@uaq.edu.mx (E.-G.H.)

* Correspondence: lroman26@alumnos.uaq.mx

† These authors contributed equally to this work.



Citation: Roman-Rivera, L.-R.; Sotelo-Rodríguez, I.; Pedraza-Ortega, J.C.; Aceves-Fernandez, M.A.; Ramos-Arreguín, J.M.; Gorrostieta-Hurtado, E. Reduced Calibration Strategy Using a Basketball for RGB-D Cameras. *Mathematics* **2022**, *10*, 2085. <https://doi.org/10.3390/math10122085>

Academic Editors: Xiangtao Zheng, Jinchang Ren and Ling Wang

Received: 19 May 2022

Accepted: 13 June 2022

Published: 16 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: RGB-D cameras produce depth and color information commonly used in the 3D reconstruction and vision computer areas. Different cameras with the same model usually produce images with different calibration errors. The color and depth layer usually requires calibration to minimize alignment errors, adjust precision, and improve data quality in general. Standard calibration protocols for RGB-D cameras require a controlled environment to allow operators to take many RGB and depth pair images as an input for calibration frameworks making the calibration protocol challenging to implement without ideal conditions and the operator experience. In this work, we proposed a novel strategy that simplifies the calibration protocol by requiring fewer images than other methods. Our strategy uses an ordinary object, a know-size basketball, as a ground truth sphere geometry during the calibration. Our experiments show comparable results requiring fewer images and non-ideal scene conditions than a reference method to align color and depth image layers.

Keywords: RGB-D camera; RGB-D camera calibration; spherical object; 3D reconstruction; sphere detection

MSC: 65D19; 53C38

1. Introduction

RGB-D cameras are becoming popular due to their availability, size, and accessible cost; there are different choices in the market from different brands. Microsoft popularized this type of camera with the Kinect [1] camera used initially for gaming (Figure 1a) in 2010, and its 3D depth technology used structured light [2]. It had a visual range from 0.8 to 4 m, producing 640×480 depth images. However, scientific research was later possible thanks to a personal computer's Kinect software development kit allowing this camera to be connected directly to a personal computer. Data produced by the Kinect could finally be processed for scientific purposes [3]. Two years later, Kinect V2 replaced Kinect V1 for the Xbox One gaming console with a range from 0.5 to 4.5 meters, a resolution of 512×424 , and a 1080p RGB camera [3,4].







After Kinect, other RGB-D alternatives appeared. Intel released its Realsense camera line [3,5] reducing camera size and power requirements [4], but providing a software development kit to work with the data directly from a personal computer, allowing production of 3D cloud points quickly (Figure 1b). Intel Realsense D415 RGB-D camera has a visual range from 0.16 to 10 m and produces depth images with a resolution of 1280×720 . Realsense D435 has a visual range from 0.2 to 4.5 m. In these cameras, depth computation is performed by an integrated ASIC (Application-Specific Integrated Circuit) [4].

Conversely, Occipital offered its structure camera [6,7] (Figure 1c), Structure core camera from Occipital has a visual range from 0.3 to 5 m, and it produces depth images

0.2. Artículo II

Article

A Robust Sphere Detection in a Realsense Point Cloud by USING Z-Score and RANSAC

Luis-Rogelio Roman-Rivera ^{†,*}, Jesus Carlos Pedraza-Ortega [†], Marco Antonio Aceves-Fernandez [†],
Juan Manuel Ramos-Arreguín [†], Efrén Gorrostieta-Hurtado [†] and Saúl Tovar-Arriaga [†]

Facultad de Ingeniería, Universidad Autónoma de Querétaro, Cerro de las Campanas S/N,
Querétaro C.P. 76010, Mexico

* Correspondence: lroman26@alumnos.uaq.mx

† These authors contributed equally to this work.

Abstract: Three-dimensional vision cameras, such as RGB-D, use 3D point cloud to represent scenes. File formats as XYZ and PLY are commonly used to store 3D point information as raw data, this information does not contain further details, such as metadata or segmentation, for the different objects in the scene. Moreover, objects in the scene can be recognized in a posterior process and can be used for other purposes, such as camera calibration or scene segmentation. We are proposing a method to recognize a basketball in the scene using its known dimensions to fit a sphere formula. In the proposed cost function we search for three different points in the scene using RANSAC (Random Sample Consensus). Furthermore, taking into account the fixed basketball size, our method differentiates the sphere geometry from other objects in the scene, making our method robust in complex scenes. In a posterior step, the sphere center is fitted using z-score values eliminating outliers from the sphere. Results show our methodology converges in finding the basketball in the scene and the center precision improves using z-score, the proposed method obtains a significant improvement by reducing outliers in scenes with noise from 1.75 to 8.3 times when using RANSAC alone. Experiments show our method has advantages when comparing with novel deep learning method.

Keywords: 3D point cloud; RANSAC; sphere detection; RGB-D cameras; z-score

MSC: 65D19



check for updates

Citation: Roman-Rivera, L.-R.;

Pedraza-Ortega, J.C.;

Aceves-Fernandez, M.A.;

Ramos-Arreguín, J.M.;

Gorrostieta-Hurtado, E.;

Tovar-Arriaga, S. A Robust Sphere

Detection in a Realsense Point Cloud

by Using Z-Score and RANSAC.

Mathematics **2023**, *11*, 1023. [https://](https://doi.org/10.3390/math11041023)

doi.org/10.3390/math11041023

Academic Editors: Xiangtao Zheng,

Jinchang Ren and Ling Wang

Received: 22 December 2022

Revised: 11 February 2023

Accepted: 15 February 2023

Published: 17 February 2023



Copyright: © 2023 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license ([https://](https://creativecommons.org/licenses/by/4.0/)

[creativecommons.org/licenses/by/](https://creativecommons.org/licenses/by/4.0/)

[4.0/](https://creativecommons.org/licenses/by/4.0/)).

1. Introduction

RGB-D cameras have become a common sensor in the area of computer vision [1,2], and the popularity started with the Microsoft Kinect output on their Xbox video game console, when the camera could be used on a personal computer, scientists started taking advantage of it for research [3]. This type of camera produces, as data output, a color image and an depth image, both images describing the scene that is captured on camera. Currently, there are different brands and models of RGB-D cameras, and it is common for different cameras to offer different characteristics and limitations [4]. Recently, this type of camera has been embedded in mobile devices as in [5] popularizing the technology and use of 3D point clouds. Some formats for saving 3D point clouds are the XYZ format and the PLY format, where spatial information is included describing each point in three dimensions and sometimes metadata, such as color, can be included. These files can vary in size and density of points and these depend mostly on the camera that is being used to generate such files, thousands of points can be found in a scene captured in a single shot, and, generally, the complexity of the processing of this information is increased proportionally with the quality of the camera and of the information it produces, the greater the detail, the greater point density. Novel methods can be found that use this

0.3. Artículo III



ACCEPTANCE LETTER

Conferences and Workshops in Telematics and Computing
WITCOM 2023

Dear

Luis Rogelio Román Rivera, Jesus Carlos Pedraza-Ortega, Israel Sotelo-Rodriguez, Ramón Gerardo Guevara-González and Manuel Toledano-Ayala

Paper ID: 5772

Title: 3D point cloud outliers and noise reduction using neural networks

We are delighted to inform you that the paper referenced above has been accepted for presentational presentation at WITCOM 2023, subject to fulfilling the requirements listed below.

Congratulations on your acceptance!

Please carefully review this email as it contains important information regarding the inclusion of your paper in the Proceedings of WITCOM 2023, which will be published in the Springer CCIS Series of Journals (<https://www.springer.com/series/7899>)

Yours faithfully

Félix Mata

Chair WITCOM conferences 2023

0.4. Requisito manejo de la lengua inglés



A QUIEN CORRESPONDA:

La que suscribe, Directora de la Facultad de Lenguas y Letras, hace **C O N S T A R** que

ROMAN RIVERA LUIS ROGELIO

Presentó el **Examen de Manejo de la Lengua** efectuado el día veintisiete de junio de dos mil veintidós, en el cual obtuvo la siguiente calificación:

8-

Se extiende la presente a petición de la parte interesada, para los fines escolares y legales que le convengan, en el Campus Aeropuerto de la Universidad Autónoma de Querétaro, el día seis de julio de dos mil veintidós.



Atentamente,
"Enlazar Culturas por la Palabra"

DRA. ADELINA VELÁZQUEZ HERRERA

AVH/japa*CL*FLL-C.-1399

0.5. Proyecto FONDEC 2021-2023



Centro Universitario, junio 30 de 2023
Oficio Núm: 3376

DR. JESUS CARLOS PEDRAZA ORTEGA
FACULTAD DE INGENIERIA
PRESENTE

Me permito hacer de su conocimiento que el **H. Consejo Universitario**, en sesión ordinaria del 29 de junio de 2023, realizada en el Auditorio Fernando Díaz Ramírez, dió la siguiente resolución al **INFORME FINAL** del proyecto de investigación del cual usted es responsable:

Título	Inicio / Término	Colaboradores / Participantes	Resolución
<p>Detección de microcalcificaciones en mamografías, utilizando técnicas de aprendizaje máquina y aprendizaje profundo. FIN202124</p> <p>Financiamiento: I. Interno, I.3 Convocatoria con recursos financieros de la UAQ FONDEC-UAQ 2021 \$95,000 MXN.</p>	<p>Septiembre 2021 / Febrero, 2023</p>	<p>Efren Gorrostieta Hurtado, Juan Manuel Ramos Arreguin, Luis Antonio Salazar Licea, Marco Antonio Aceves Fernandez, Saul Tovar Arriaga / Edgar Rodrigo Lopez Silva, Gerardo Treviño Valdes, Israel Sotelo Rodriguez, Jose Gustavo Alfaro Montufar, Victor</p> <p>Beltran Barrera, Mayra Azucena Cintora Garcia, Luis Rogelio Roman Rivera.</p>	<p>INFORME FINAL APROBADO</p>

Aprovecho la ocasión para enviarle un cordial saludo.

ATENTAMENTE
"EDUCO EN LA VERDAD Y EN EL HONOR"

DR. JAVIER AVILA MORALES
Secretario Académico

Dra. Ma. Guadalupe Flavia Loarca Piña.-Directora de Investigación y Posgrado
 Expediente
rfiores

Dictamen:202314108

0.6. Productos obtenidos

Tabla 1: Productos obtenidos durante el programa de Doctorado en Ingeniería.

Tipo de producto	Nombre	Tipo de Participación
Artículo en revista indizada	Reduced Calibration Strategy Using a Basketball for RGB-D Cameras	Primer Autor
Artículo en revista indizada	A Robust Sphere Detection in a Realsense Point Cloud by USING Z-Score and RANSAC	Primer Autor
Artículo en congreso internacional	3D Point Cloud Outliers and Noise Reduction Using Neural Networks	Primer Autor
Artículo en congreso internacional	Caracterización de valores atípicos en nube de puntos en 3D para la reducción del tiempo de ejecución en memoria	Colaborador
Artículo en congreso internacional	Red neuronal convolucional de baja latencia para la estimación de profundidad monocular	Colaborador
Tesis Maestría	Reducción de valores atípicos y presencia de ruido en nube de puntos en 3D utilizando técnicas de aprendizaje profundo	Director de Tesis
Tesis Maestría	Estimación de profundidad a partir de imágenes monoculares mediante una arquitectura CNN en sistemas embebidos	Colaborador en el sínodo