



DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

EXPERIENCIA E INVESTIGACIÓN APLICADA EN VISIÓN
COMPUTACIONAL Y SENSORES DE PROFUNDIDAD

INFORME DE PRÁCTICA PROFESIONAL

Estudiante:	Samuel Ignacio Bugueño Córdova
Supervisor:	Rodrigo Verschae Tannenbaum
Establecimiento:	Universidad de O'Higgins
Organización:	Laboratorio de Robótica y Sistemas Inteligentes
Santiago, Chile	

Índice de Contenidos

1. Introducción	1
2. Marco Teórico	2
2.1. Cámaras RGB	2
2.2. Cámaras ToF	2
2.3. Cámaras de polarización	4
2.4. Modelo pinhole	5
2.5. Modelo de lentes	5
2.6. Modelo de perspectiva central	6
2.7. Calibración intrínseca y extrínseca	7
2.8. Proyección entre cámara RGB y ToF	8
2.9. Pose en 3D	8
2.10. ArUco Marker	9
2.11. Métodos de registro locales y globales	10
2.12. Cálculo de normales en nube de puntos	10
2.13. Reconstrucción de superficie de Poisson	11
3. Prodecimientos y experimentos	12
3.1. Instalación ArenaSDK Software	12
3.2. Conexión y armado cámaras RGB, Tof y polarización	12
3.3. Uso de ArenaView	13
3.4. Calibración Cámara RGB + ToF	15
3.4.1. Calibración intrínseca cámara RGB	15
3.4.2. Calibración extrínseca entre cámara RGB y ToF	16
3.4.3. Proyección de información RGB a cámara ToF	18
3.4.4. Toma de datos	19
3.5. Toma de datos objetos a distinta distancia	19
3.6. Captura de datos de objetos en distintos puntos con ArUco Marker	20
3.7. Superposición de nube de puntos	21
3.8. Capturas de cámara de polarización con cereza	23
4. Resultados	26
4.1. Calibración Cámara RGB + ToF	26
4.2. Captura ToF de objeto a distinta distancia	28
4.3. Superposición de nubes de puntos de distintos puntos de vista de objetos	33
5. Análisis de resultados	35
5.1. Calibración Cámara RGB + ToF	35
5.2. Captura ToF de un objeto a distinta distancia	35
5.3. Superposición de nubes de puntos de distintos puntos de vista de objetos	36
6. Conclusión	37
7. Bibliografía	38

Lista de Figuras

1.	Diagrama funcionamiento cámara RGB	2
2.	Ilustración funcionamiento cámara ToF	3
3.	Diagrama de sensores usados para capturar información de cada grado de polarización	4
4.	Diagrama modelo de lentes para cámaras	6
5.	Diagrama de modelo de perspectiva central en cámaras	7
6.	Capturas realizadas a objetivo para realizar calibración intrínseca	8
7.	Ejemplo de matriz de rotación	9
8.	ArUCo Marker individual	9
9.	Captura con reconocimiento de ArUco markers	10
10.	Imagen cámaras ToF y RGB armadas entre sí y conectadas sobre un trípode	13
11.	Objetivo de calibración	16
12.	Captura para calibración extrínseca entre cámaras	17
13.	Captura de prueba para proyección entre información RGB y cámara ToF	18
14.	Captura con cámara ToF a distinta distancia de objeto	20
15.	Setup completo de capturas con cámara de polarización para cereza.	24
16.	Posición de la cámara de polarización y del servomotor con la cereza en el setup.	24
17.	Objetos usados a lo largo de los experimentos	26
18.	Captura dinosaurio interior 5m	31
19.	Captura dinosaurio exterior 6m	31
20.	Captura dinosaurio interior 7m	32
21.	Captura dinosaurio exterior 8.1m	32
22.	Mesh de jarrón	34

Lista de Tablas

1.	Comparación de capturas a diferentes distancias (0.1m - 1m)	28
2.	Comparación de capturas a diferentes distancias (0.1m - 2m)	29
3.	Comparación de capturas a diferentes distancias (3m - 4m)	30
4.	Mesh objetos experimentos (superior izquierda: dinosaurio, superior derecha: liso-form, inferior izquierda: caja e inferior derecha: taza negra)	33

1. Introducción

Chile ha sido conocido históricamente como un país exportador de minerales, como el cobre, y de productos agrícolas como las cerezas. Sin embargo, la preponderancia de esta en los últimos años ha aumentado significativamente por las demandas del mercado chino en cada año nuevo. No obstante, los productores agrícolas se enfrentan a muchos desafíos a lo largo del proceso de sembrado y de cosecha, lo que dificulta el poder mantener y aumentar la producción de cerezas. Por lo tanto, para mejorar el rendimiento de los productos agrícolas y facilitar el trabajo de los agricultores es necesario el uso de tecnologías para mejorar las técnicas usadas. Es en esta necesidad que el área de visión computacional ofrece una gran cantidad de oportunidades para facilitar la cosecha de las cerezas, como el conteo de estas frutas junto con la evaluación de su estado.

Aunque el área de visión computacional es muy versátil para varias áreas de la robótica, existen características de las cerezas que a veces son difíciles de capturar adecuadamente o pueden tener mucha incertidumbre. Es por esta razón que además de las cámaras tradicionales (también llamadas RGB) se deben usar otros tipos de cámaras con otras propiedades y beneficios como las cámaras ToF y las cámaras de polarización.

A lo largo de este informe, se va a abordar la investigación y los experimentos realizados en las cámaras ToF y de polarización de la marca LucidVisions para así comprender mejor su funcionamiento junto con su aplicación en el área anteriormente mencionada. En este informe primero se va a hacer una revisión bibliográfica de las cámaras ToF y de polarización en general, junto con la documentación en particular de las cámaras a usar. Luego, se va a especificar la forma en que se realizaron los diversos procedimientos usados en la cámara y experimentos realizados. Después, se muestra los resultados de los experimentos y los datos procesados para a continuación analizar dichos resultados. Finalmente, se va a realizar una conclusión sobre la investigación y los experimentos realizados.

2. Marco Teórico

En este marco teórico, se va a abordar el funcionamiento de las cámaras a usar, los modelos usados para representarlos y los métodos usados en los experimentos realizados. Además, se van a mencionar todas las herramientas de software y físicas que se usaron para mejorar los resultados obtenidos de la cámara.

2.1. Cámaras RGB

Las cámaras RGB corresponden a las cámaras "tradicionales". Estas cámaras están compuestas principalmente por un lente, un film y el cuerpo de la cámara en sí. La cámara funciona al concentrar los haces de luz por medio del lente al interior del cuerpo de la cámara. Una vez los haces de luz se encuentran dentro de la cámara, el film reacciona con la luz que entra para así capturar la imagen de la escena. En caso de las cámaras digitales, se reemplaza el film por un sensor que permite almacenar la información de la luz en un array de píxeles [1]. A pesar de tener varios formatos, uno de los más comunes es el formato BGR, en donde cada píxel tiene información de tres canales, azul, verde y rojo.

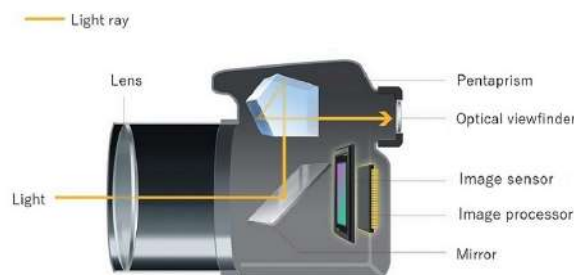


Figura 1: Diagrama funcionamiento cámara RGB

La cámara RGB usada en los experimentos realizados corresponde a la cámara Tritón CC a color [2].

2.2. Cámaras ToF

Las cámaras ToF o "Time of Flight" corresponden a cámaras activas que permiten la medición de la distancia respecto al entorno. Esto se realiza por medio de la emisión de luces infrarrojas al ambiente y medición del tiempo que toma la luz reflejada para volver a la cámara. Con la medición del tiempo, se puede calcular la distancia de la cámara respecto a un objeto en particular de la escena [3]. Estas cámaras son similares a LIDAR, pero este último sensor usa pulsos láser más potentes que las cámaras ToF, lo que les permite tener una mayor precisión y alcance. Mientras que las cámaras ToF además de pulsos láser pueden usar señales de luz continua, suelen ser más convenientes económicamente y entregan una buena resolución a corta y mediana distancia [4].

Existen 2 tipos de cámaras ToF: directa e indirecta. Las cámaras ToF directas se encargan de medir directamente el tiempo que toma un haz de luz infrarrojo ir y volver de la colisión con el objeto (usando un temporizador), mientras que las cámaras ToF indirectas miden el desfase

entre las ondas emitidas y reflejadas . Dentro de las cámaras ToF indirectas, existen 2 métodos para enviar el laser infrarrojo: pulsado o por una onda continua. Este último método es el que se presenta en las cámaras a usar [5].

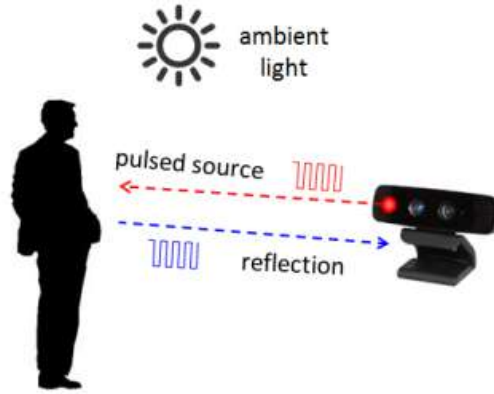


Figura 2: Ilustración funcionamiento cámara ToF

Las cámaras usan el anterior método presentan 4 ventanas (o microframes) con un desfase de 90° entre ellas (o sea, representan 0°, 90°, 180° y 270°). Al recibir la luz infrarroja reflejada, esta se acumula en forma de carga eléctrica en cada uno de estos frames o ventanas. Al ponderar cuánto de cada desfase contribuye a la onda reflejada, se puede determinar la distancia de la escena respecto a la cámara (d) y el ángulo de desfase entre la onda emitida y la reflejada (ϕ), como se muestra en la siguiente ecuación [6]:

$$\phi = \arctan \left(\frac{Q_3 - Q_4}{Q_1 - Q_2} \right)$$

$$d = \frac{c}{4\pi f} \phi$$

Sin embargo, estas cámaras suelen sufrir problemas con objetos que son oscuros, transparentes o que son reflectantes, debido a que generan problemas con la reflexión de las ondas emitidas. Además, se pueden generar varios problemas en ciertas esquinas y en intersecciones entre superficies por la reflexión especular que se genera, por lo que se entregan datos erróneos sobre la distancia. No obstante, por medio de varios tratamientos, se pueden generar mejores capturas reduciendo el ruido y aumentando la precisión de la información de distancia [7] y [8]. Para lo anterior, se recomienda ver el link en la referencia [8].

Esta información se puede guardar como un mapa de profundidad en forma de una imagen 2D en escala de grises. En este mapa se asocia a cada pixel una determinada distancia respecto a la cámara. Esto a su vez se puede generar una nube de puntos, en donde se muestra la posición 3D de cada punto (incluyendo su distancia) [3].

La cámara de Lucid Vision Helios2 Ray corresponde a una cámara ToF indirecta [9]. Mientras que el resto de las cámaras de la línea Helios2 emite luz infrarroja en la longitud de onda de 850 nm para uso en interiores (al ser la banda con menos interferencia en interiores), la cámara Helios2 Ray emite luz con una longitud de onda de 940 nm para usar mejor en exteriores. Si bien la cámara Helios2 Ray está diseñada para el uso en exteriores, se puede usar también en interiores con un poco de mayor imprecisión que en exteriores [10]. Esta cámara permite obtener los resultados como

nubes de puntos 3D, mapas de profundidad, entre otros. Además, la unidad de medida usada para calcular la profundidad de la escena corresponde a milímetros.

2.3. Cámaras de polarización

La luz tiene una característica importante conocida como polarización. Esta propiedad corresponde a la dirección respecto a la que oscila la onda electromagnética de la luz. La luz puede ser clasificada en polarizada (la luz oscila solo respecto a un plano) y no polarizada (la luz oscila respecto a varios planos). La mayoría de las fuentes de luz emiten luz no polarizada, incluyendo el Sol. Hay distintos tipos de polarización de la luz, como la polarización lineal (oscila respecto a un eje), elíptica o circular. No obstante, la cámara usada en los experimentos solo detecta luz polarizada de forma lineal.

Existen diversos fenómenos que pueden polarizar la luz de forma natural, como la reflexión en un cierto ángulo respecto a un material y la refracción [11]. La forma en la que se refleja y refracta la luz en un material afecta las características de la luz polarizada, como el grado de polarización de la luz en el caso de la reflexión. Es por eso que la luz polarizada entrega importante información sobre las características y la forma de un objeto en comparación de una cámara RGB normal. [12]

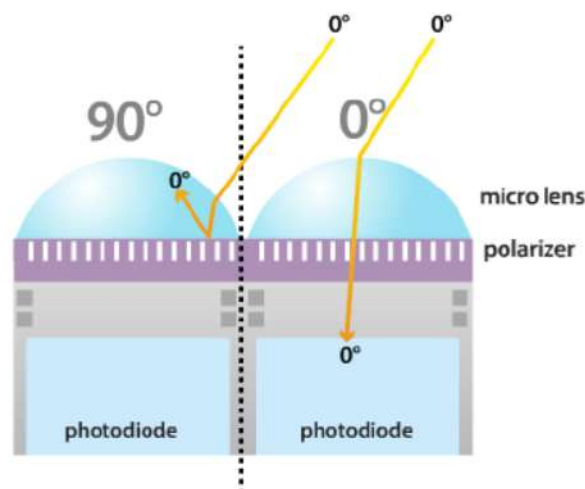


Figura 3: Diagrama de sensores usados para capturar información de cada grado de polarización

No obstante, para poder estudiarla adecuadamente, se debe poder obtener la luz polarizada de forma artificial, para lo cual se usa un polarizador. Estos pueden clasificarse a su vez en diversos tipos, pero los que en la cámara se usa polarizadores dentro de su sensor en forma de cuadrícula con distintos ángulos. En función de esto, se permite que la luz que oscila de forma perpendicular a esta grilla va a pasar por el polarizador. Este sensor presenta 4 ángulos en su polarizador (0° , 45° , 90° y 135°). Es debido a esta propiedad, que un porcentaje de luz que entra en cada polarizador va a poder pasar por este, siendo que el 100 % de la luz pasa si la dirección de oscilación es completamente perpendicular a la grilla y este porcentaje va a ir disminuyendo progresivamente hasta llegar a 0 % cuando la dirección de propagación es completamente paralela al de la grilla [13].

Gracias a estos 4 polarizadores en el sensor, se puede obtener varias propiedades importantes al juntar la información que se obtiene de cada uno por separado. Dos de las características más

importantes de la luz polarizada corresponde al grado de polarización y el ángulo de polarización. El grado de polarización corresponde a la razón entre la intensidad de la luz polarizada y la luz no polarizada que llega al polarizador. Por otro lado, el ángulo de polarización corresponde a la dirección de la máxima polarización de la luz recibida. Para poder obtener estas propiedades, es necesario obtener los parámetros de Stokes, los cuales son usados para describir el estado de la luz polarizada. Los parámetros de Stokes describen las siguientes características: S_0 representa la intensidad total de la luz recibida, S_1 y S_2 representan la cantidad de luz que está polarizado de forma lineal y horizontal y S_3 representa la cantidad de luz que está polarizada circularmente, no obstante esta última no se va a usar al ser rara la polarización circular y solo usar un polarizador perpendicular. Estos parámetros se calculan por las ecuaciones descritas a continuación:

$$\begin{aligned} S_0 &= I = P_0^\circ + P_{90}^\circ \\ S_1 &= Q = P_0^\circ - P_{90}^\circ \\ S_2 &= U = P_{45}^\circ - P_{135}^\circ \end{aligned}$$

Se debe considerar que las intensidades de la luz en cada uno de los polarizadores respectivos se suman y restan. Una vez que se obtienen estos parámetros, se puede calcular el ángulo de propagación Θ (AoP) y el grado de polarización Π (DoLP) con las siguientes ecuaciones [14]:

$$\begin{aligned} \Pi &= \frac{\sqrt{S_1^2 + S_2^2}}{S_0} \\ \Theta &= \arctan \frac{S_2}{S_1} \end{aligned}$$

Es por medio del cálculo de los parámetros anteriores y las características de esta cámara que su uso resulta útil en una gran cantidad de aplicaciones, como la detección de rasguños, la eliminación de brillo en cámaras, la detección de personas en automóviles, la inspección de paquetes, entre otros [13].

La cámara de polarización utilizada para los experimentos corresponde al modelo Phoenix 5.0 [16].

2.4. Modelo pinhole

El modelo pinhole es la forma clásica de modelar cómo las cámaras capturan una escena en 3D y la proyectan en una imagen 2D. Este modelo se basa en el principio más simple para construir una cámara fotográfica, que consiste en crear un agujero en una habitación o caja cerrada y oscura. La imagen capturada se proyecta invertida en el fondo de la cámara. [17]

2.5. Modelo de lentes

A diferencia del modelo pinhole, en lugar de usar un agujero para direccionar los haces de luz, se utiliza una lente cóncava para obtener una mejor calidad en las imágenes [17] y [18].

En este modelo, existen los siguientes elementos geométricos. En primer lugar, está el centro de la cámara, representado por la letra O . Luego, se definen dos planos: uno correspondiente a la cámara y otro en el que se proyecta la imagen [17] y [18].

También se define el punto focal como el punto en el que todos los rayos de luz que pasan por la lente se intersectan. Además, la distancia focal se define como la distancia entre el centro de la cámara y el punto focal. El eje que conecta el centro de la cámara con el punto focal se denomina eje óptico [17] y [18].

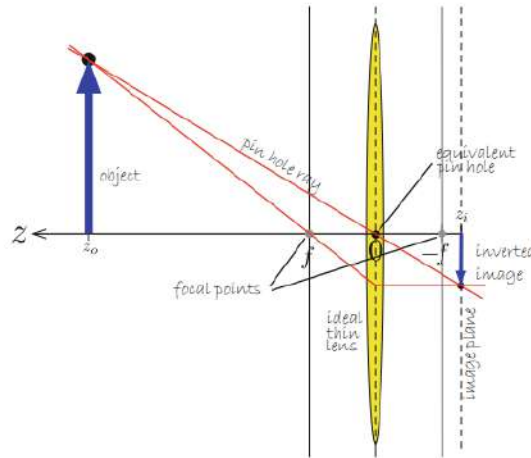


Figura 4: Diagrama modelo de lentes para cámaras

No obstante, por el uso de lentes, se introducen también distorsiones en la imagen, siendo la más común la llamada distorsión radial, la cual produce una variación en la magnificación de la imagen en función de la distancia al eje óptico [17] y [18].

2.6. Modelo de perspectiva central

Este es un modelo más generalizado que incluye los modelos anteriores. Aquí se aprovechan las definiciones geométricas y las derivaciones de los dos modelos previos, pero, a diferencia de estos, se considera que el plano de la imagen está al frente de la cámara en lugar de atrás (aunque sigue estando en la distancia focal) [17].

A su vez, se puede hacer una proyección del centro de la cámara al plano de la imagen, lo que se denomina punto principal o C' , con coordenadas (c_x, c_y) . Este punto se usa posteriormente para definir la posición de los píxeles en la imagen.

También cabe mencionar que se define un marco de referencia o sistema de coordenadas de la cámara en el punto C , con el eje z apuntando hacia adelante de la cámara, el eje y hacia abajo y el eje x también hacia abajo. Mientras tanto, el marco de referencia de la imagen proyectada se define en el punto C' , donde el eje z de la cámara se intersecta con el plano de la imagen [18].

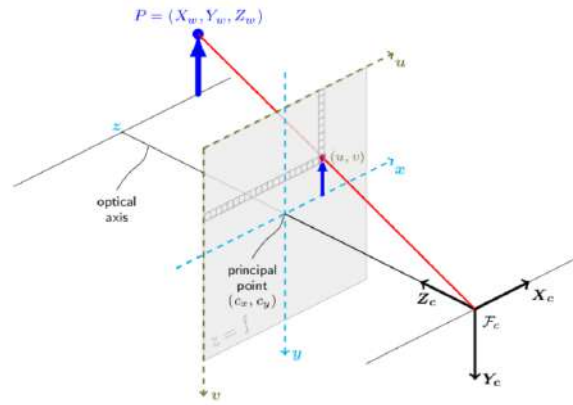


Figura 5: Diagrama de modelo de perspectiva central en cámaras

Finalmente, en este modelo se pueden obtener las proyecciones del mundo a la imagen 2D por medio de la siguiente transformación:

$$sm' = A[R|t]M'$$

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

En la anterior transformación, se tiene que:

- (X, Y, Z) son las coordenadas de un punto 3D en el espacio de coordenadas globales.
- (u, v) son las coordenadas del punto de proyección en píxeles.
- A es la matriz de la cámara, también conocida como matriz de parámetros intrínsecos.
- (c_x, c_y) es el punto principal, que suele estar en el centro de la imagen.
- f_x, f_y son las distancias focales expresadas en unidades de píxel.

Se tiene que la matriz R y la matriz T corresponden a la matriz de rotación y la matriz de traslación del sistema de referencia global con respecto al de la cámara. Juntas, estas matrices conforman la matriz extrínseca de la cámara [19].

2.7. Calibración intrínseca y extrínseca

La calibración corresponde al proceso de encontrar los parámetros intrínsecos y extrínsecos de una cámara, junto con los parámetros que definen la distorsión del lente de la cámara [17].

Para esto, se deben realizar varias capturas de un objetivo, ya sea en 3D o en 2D, como un tablero de ajedrez, un tablero ChArUco o patrones circulares. A partir de estas capturas, se deben detectar los patrones de los objetos para encontrar puntos importantes en estos y, así, comparar cómo se capturan esos puntos en la cámara con respecto a su posición real en el mundo 3D [20].

Existen diversos algoritmos que realizan esta función, y el algoritmo estándar de OpenCV sigue los siguientes pasos [19]:

1. Dependiendo de la configuración que se use para la función `calibrate`, se definen los valores iniciales de la matriz de calibración intrínseca.
2. Se usa un algoritmo llamado `SolvePnP` para calcular la pose inicial de la cámara (parámetros extrínsecos). Para esto, se busca disminuir los errores de reproyección entre los puntos que se detectan en la cámara respecto a su posición real en 3D. Por lo tanto, se usan los valores iniciales de la matriz de calibración y luego se emplea el algoritmo de optimización Levenberg-Marquardt.
3. Una vez que se tiene la pose inicial, se busca minimizar los errores de reproyección de la cámara de forma similar al método `SolvePnP` para así calcular los parámetros intrínsecos de la cámara.

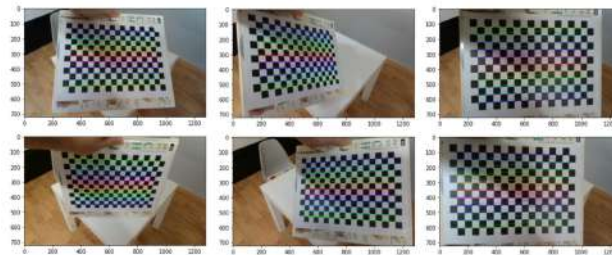


Figura 6: Capturas realizadas a objetivo para realizar calibración intrínseca

2.8. Proyección entre cámara RGB y ToF

Un objetivo importante que se va a realizar a lo largo de los experimentos es poder añadir información de la cámara RGB a la cámara ToF. El procedimiento consiste primero en encontrar la correspondencia entre los puntos de la nube de puntos que es capturada por la cámara ToF y la cámara RGB, y luego tomar la información de la imagen RGB y asociarla a la nube de puntos [21].

Para realizar el primer paso, se va a usar la función `ProjectPoints2` del módulo de OpenCV. Esta función permite proyectar un punto en coordenadas 3D a un punto en coordenadas 2D, para lo cual requiere conocer los parámetros intrínsecos de la cámara RGB y los parámetros extrínsecos (posición relativa) de la cámara ToF respecto a la cámara RGB. Una vez que se tienen dichos valores, se pueden proyectar los puntos de la cámara ToF a sus píxeles respectivos en la imagen RGB [21].

Finalmente, se debe acceder a cada píxel en la cámara RGB y sus valores en cada uno de sus canales para luego asignárselos a sus puntos respectivos en la cámara ToF [21].

2.9. Pose en 3D

Existen varias formas de representar la pose en 3D de un objeto respecto a un marco de referencia global. No obstante, para los experimentos y el procesamiento de los datos, se va a preferir usar la matriz de transformación homogénea. Esto se puede representar a partir de la siguiente transformación [17]:

$$\begin{pmatrix} {}^A X \\ {}^A Y \\ {}^A Z \\ 1 \end{pmatrix} = \begin{pmatrix} {}^A \mathbf{R}_B & {}^A \mathbf{t}_B \\ \mathbf{0}_{1 \times 3} & 1 \end{pmatrix} \begin{pmatrix} {}^B X \\ {}^B Y \\ {}^B Z \\ 1 \end{pmatrix}$$

Esta matriz está compuesta por 2 submatrices: una de traslación (3x1) y otra de rotación (3x3). Estas matrices permiten definir el nuevo marco de referencia a partir del definido originalmente. Por un lado, la matriz de traslación define el vector que conecta el origen del centro del nuevo sistema de referencia con el antiguo, y cada columna de la matriz de rotación define la dirección de los nuevos ejes de la matriz en función de las coordenadas anteriores. Esto se muestra en el ejemplo a continuación [17]:

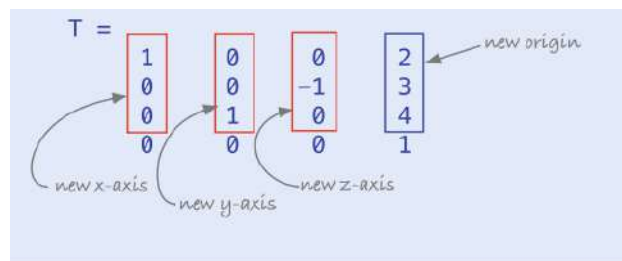


Figura 7: Ejemplo de matriz de rotación

2.10. ArUco Marker

Un ArUco Marker corresponde a un marcador fiducial cuadrado binario, es decir, a un marcador compuesto por un borde negro y blanco en su interior. Un ArUco Marker está compuesto por un gran borde de color negro y una serie de bloques binarios en su interior. El borde negro sirve para mejorar el reconocimiento de estos marcadores, y la codificación binaria permite obtener la identificación de los mismos [22].



Generated marker

Figura 8: ArUCo Marker individual

Estos marcadores son ampliamente usados en el reconocimiento de pose debido a su capacidad de corrección de errores, fácil reconocimiento y a la presencia de sus 4 vértices. Además, tienen un paquete especializado con una serie de funciones que permiten construir el ArUco Marker de acuerdo con el tamaño de bits deseado y el diccionario que se desea usar, identificar el marcador específico a utilizar y obtener la pose de este marcador respecto a la cámara, así como proyectar los ejes respecto al centro del marcador [22].

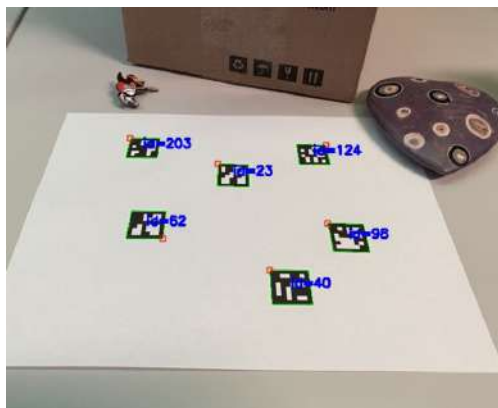


Figura 9: Captura con reconocimiento de ArUco markers

2.11. Métodos de registro locales y globales

El registro de nubes de puntos corresponde al problema de estimar la transformación que se debe aplicar entre 2 nubes de puntos capturadas de la misma escena para poner los puntos correspondientes entre estas en la misma posición y unirlos en una nube de puntos 3D completa [23].

Los métodos de registro buscan encontrar rasgos distintivos entre las nubes de puntos para encontrar la correspondencia entre estas, usando como función objetivo minimizar el error de proyección geométrica. Hay diversos enfoques para realizar estos métodos, pero se usarán a lo largo de este informe los métodos de registro mediante optimización. Además, se pueden clasificar en métodos globales (no requieren una cierta alineación inicial) y locales (mejoran una transformación inicial y requieren estar alineados desde el principio) [23].

Estos métodos se caracterizan por usar algoritmos de optimización para estimar la transformación entre las 2 nubes de puntos. Tienen 2 pasos: encontrar la correspondencia entre las nubes de puntos (por medio de la diferencia de posición en las nubes y sus características) y estimar la transformación necesaria para superponer adecuadamente las nubes de puntos [24].

En particular, se va a usar el método conocido como ICP. Este método corresponde a un método de registro local. Junto con diversos métodos de optimización disponibles, se usan 3 tipos de métricas distintas para calcular la distancia: punto a punto, punto a plano y plano a plano. El primero busca los pares de puntos más cercanos basados en la distancia en coordenadas o características, el segundo minimiza la distancia ortogonal entre un punto y el plano en otro (usando la normal del punto) y el último usa un enfoque similar al primero, pero usando las normales de los puntos [24].

2.12. Cálculo de normales en nube de puntos

Existen diversos métodos que permiten calcular las normales en una nube de puntos, pero la plataforma usada a lo largo de la experimentación, llamada Cloud Compare, utiliza los métodos de Ajuste de Planos con Mínimos Cuadrados (Least Squares - LS), Ajuste de Superficie Cuadrática (Quadric Fitting - QUADRIC) y Triangulación Local (TRI). En el uso de estos métodos, también se usa Octree, que corresponde a un tipo de estructura de árbol para agrupar los datos. El uso de esta estructura permite calcular las normales de la nube de puntos a partir de la vecindad alrededor de un punto en un cierto radio [25] y [26].

2.13. Reconstrucción de superficie de Poisson

Este es un método que se usa para el procesamiento de datos a lo largo de los experimentos y un plugin dentro de la plataforma CloudCompare. Para el uso de este método, se debe calcular de antemano las normales de la nube de puntos, y estas deben ser consistentes y tener poco ruido. Como resultado de este método, se obtiene un mesh a partir de la nube de puntos [27].

Este método busca enfocar el problema de la reconstrucción de superficie de una nube de puntos en encontrar una función escalar χ que cumpla la siguiente relación con un campo vectorial \vec{V} que satisface la siguiente ecuación [28]:

$$\Delta\chi \equiv \nabla \cdot \nabla\chi = \nabla \cdot \vec{V}$$

Este método se caracteriza, además, por poder adaptarse bien al ruido de la nube de puntos y puede manejar toda la información de la nube a la vez [28].

3. Prodecimientos y experimentos

3.1. Instalación ArenaSDK Software

Las cámaras ToF Helios2 Ray y RGB Tritón (CC, o sea, a color) de Lucid Vision Labs se pueden usar directamente por código con C, C++ y Python o por medio de una interfaz gráfica entregada por el fabricante llamada ArenaView. Ambas formas permiten realizar capturas de las cámaras, modificar parámetros de la cámara y visualizar las capturas. No obstante, si se desea una forma más gráfica en intuitiva de usar las cámaras, se recomienda usar ArenaView.

Para usar las cámaras por medio de los primeros 2 métodos, se requiere descargar los paquetes y el software de Lucid Visions Lab por medio del siguiente link: <https://thinklucid.com/downloads-hub/> [29]. Para esto, primero se debe crear una cuenta en la plataforma de Lucid Visions Lab (así como para leer más material respecto a las cámaras) y descargar el software respectivo para el sistema operativo del computador en el que se van a usar las cámaras. Se recomienda crear un ambiente en Conda para evitar problemas en las versiones de los paquetes requeridos para el uso del software de ArenaSDK.

Al descargar este paquete, se va a encontrar la GUI que corresponde a Arena View, los paquetes de Arena que se usan para conectarse a las cámaras y modificar sus parámetros para los distintos lenguajes y códigos de ejemplo en C, C++ y Python en la siguiente dirección del computador: C:\ProgramData\Lucid Vision Labs\Examples\src\ (en caso de usar Windows).

Para obtener más información sobre las configuraciones necesarias para realizar la instalación se puede ir al siguiente link: <https://support.thinklucid.com/arena-sdk-documentation/> [30].

Sin embargo, si también se quiere usar las cámaras con un software de terceros como ROS, se recomienda dirigirse al siguiente link: <https://support.thinklucid.com/#3rd-party> [31].

3.2. Conexión y armado cámaras RGB, Tof y polarización

Primero se va a abordar la forma de conexión entre la cámara y el computador de forma independiente. Luego, se va a mostrar cómo se puede instalar las cámaras juntas para obtener información al mismo tiempo de la escena.

En el caso de la cámara ToF, solo es necesario atornillar el adaptador del trípode (en el caso del laboratorio, ya está incluido). Para armar la cámara RGB y de polarización, se requiere primero sacar el tapa para el polvo que tiene la cámara. Luego, se debe tomar el lente de la cámara y atornillarlo a la ranura con el sensor que posee la cámara. Posteriormente, se debe tomar el tubo del lente de la cámara, colocarlo sobre la lente y atornillarlo también en la cámara. En el caso de los componentes usados en el RisLab, solo se tiene un lente y un tubo para la lente para tanto las cámaras RGB como la de polarización, por lo que solo se puede usar una de estas cámaras a la vez. Para más detalles para el armado de estas cámaras, se puede ir a los siguientes links: <https://support.thinklucid.com/getting-started/#gs-Helios-Connect> y <https://www.youtube.com/watch?v=KyQD-z0Km00> [32] y [33].

Para la conexión de las cámaras RGB y ToF se requieren 2 cables: uno de alimentación llamado M8 Power Supply Adapter (<https://thinklucid.com/product/m8-gpio-power-supply-na-eu/> [35]) y otro de Ethernet para transmisión de datos llamado M12 to RJ45 IP67 Cat6a Cable (<https://thinklucid.com/product/m12-to-rj45-ip67-cat6a-cable-amphenol-2-0m-dark-green/>) [36]. Ambas cámaras tienen dos ranuras en la parte trasera de la cámara, una para el cable M8 y otro

para el cable M12. Para insertarlos, se debe seguir la forma especial de cruz que tiene el cable en su interior con la ranura de la cámara. En el caso especial en que se tenga un PoE switch en el computador, solo es necesario insertar el cable de conexión M12 a las cámaras. No obstante, por conocimiento empírico, **se recomienda el usar la cámara ToF Helios2 Ray con tanto el cable de alimentación como el ethernet, para evitar problemas de alimentación que puedan afectar el funcionamiento de la cámara.**



Figura 10: Imagen cámaras ToF y RGB armadas entre sí y conectadas sobre un trípode

Por otro lado, en el caso de la cámara de polarización Phoenix, es necesario solo el uso de un cable ethernet Cat6a para poder conectar la cámara con el computador. En este caso, no es necesario ningún cable de alimentación. Aunque se recomienda poner atención la orientación en la que se conecta, puesto que debe ser de manera que la luz parpadeante que aparece al conectarse debe estar hacia arriba. Más información en el siguiente link: <https://support.thinklucid.com/getting-started/#gs-phoenix-connect> [36].

Finalmente, se debe considerar que de los componentes presentes en el laboratorio, solo existe un adaptador de trípode que está instalado en la cámara ToF. Por lo tanto, es requerido el montar sobre la cámara ToF la cámara RGB o de polarización. Para esto, se tiene una montura especial que se usa para atornillar las cámaras RGB y de polarización sobre la ToF. Primero, se atornilla la montura a la cámara RGB y de polarización usando los espacios más al interior de la montura con los tornillos más pequeños a disposición en el paquete del laboratorio. Una vez atornillado la cámara RGB y de polarización a la montura, se debe atornillar la montura a la cámara ToF, usando los tornillos más grandes que están disponibles en el paquete de las cámaras. Se debe considerar, que esta montura tiene una orientación específica para atornillarse correctamente a la cámara Helios, por lo que si no se toma esto en cuenta con anterioridad, la cámara RGB y de polarización pueden tener la orientación equivocada respecto a la cámara ToF.

3.3. Uso de ArenaView

La plataforma de ArenaView tiene 6 ventanas principales. Estas se pueden observar a continuación:

- Cameras: en donde aparecen las cámaras conectadas y se pueden seleccionar para realizar capturas, grabar en tiempo real y configurar la cámara.
- Logs: en donde se entregan notificaciones del funcionamiento de la plataforma errores.

- Options en donde se pueden configurar y realizar diversas funciones tanto en el envío, en el procesamiento y en la forma de guardar los datos de las cámaras. Incluye la opción de usar Jupyter Labs.
- Features: incluye los nodos con los parámetros usados para configurar la captura y envío de los datos de la cámara. Los valores parámetros pueden ser modificados directamente al modificar el dato que se encuentra inmediatamente al lado del nombre del nodo. Además, tiene un filtro con un buscador para buscar el nodo o característica que se desea modificar (si se coloca "favs" muestra los nodos favoritos). También, permite escoger con un botón si se quiere ver la lista de nodos "simple (básica) o "complete" (completa).
- Properties: esta ventana muestra información más completa del nodo, entregando la descripción de la función del nodo en particular y valores más específicos que se presentan dentro de dicho nodo.
- Pagina de inicio o "ventana principal": puede incluir varias pestañas al mismo tiempo para visualizar las capturas o grabaciones realizadas por las cámaras. Además, puede abrir pestañas con jupyter lab y con información de documentación sobre el uso de ArenaView y ArenaSDK.

Junto con lo anterior, existen 2 barras en la parte superior izquierda. La barra vertical incluye los botones que se usan para realizar capturas, empezar grabaciones y terminar transmisión de las cámaras. Para esto, se debe seleccionar la cámara a usar, activarla presionando el botón a la derecha del nombre y luego presionar el botón con la instrucción que se quiere realizar. Además, en esta barra hay botones que permiten guardar y navegar en las capturas y grabaciones de la cámara.

Por otro lado, en la barra horizontal se pueden configurar características de la propia plataforma como las ventanas que se pueden visualizar (en View), entrar a documentación de ayuda y poder copiar, pegar y deshacer las acciones dentro de la GUI.

Finalmente, se van a mencionar algunas secciones y nodos que se van a usar dentro de la pestaña de "Options" y "Features". En primer lugar, en "Options" está la sección de RGB depths. Esta sección y sus subsecciones permiten realizar los pasos necesarios para la calibración intrínseca y extrínseca de la cámaras RGB y ToF para posteriormente proyectar la información entre estas.

Por otro lado, en el caso de la cámara Tritón RGB no se vió necesario a lo largo de la experimentación modificar más nodos que Gain y Gain Automático. Estos nodos modifican la ganancia de la información recibido por la cámara, y por lo tanto, el brillo con el cual se captura las imágenes. En caso de tener la iluminación adecuada se recomienda mantener el Gain Automático como continuo. Si no se tiene la iluminación adecuada o el nivel de brillo afecta seriamente el color de las imágenes, se recomienda modificar el gain de forma adecuada. Si al usar por primera vez la cámara Tritón se observa todo de color negro, se debe ajustar el Gain Automático en continuo (después se puede cambiar a discreción del usuario).

En el caso de la cámara ToF Helios2 Ray, existen una serie de nodos que son importantes para la captura de información de la cámara. Estos se describen a continuación:

- Exposure: determina el tiempo usado para calcular la profundidad en cada captura. Mientras menor tiempo de exposición, más ruidoso va a ser la captura, pero evita la sobresaturación en tomas cercanas y con objetos de color blanco o reflectantes (posiblemente sea mejor para objetos a mayor velocidad). Mientras mayor tiempo de exposición, más preciso son las mediciones realizadas y se obtiene mejor información para objetos oscuros.

- Modo de distancia: indica la máxima distancia que la cámara ToF puede capturar. Esto varía el nivel de precisión que posee la cámara. Se recomienda ajustarlo bien para evitar ruido en la cámara, lo que se conoce como el efecto "wrap around" o "ambiguity".
- Confidence threshold: corresponde a un filtro que permite eliminar los píxeles en función del nivel de confianza que se tiene en la precisión con que se calculó la profundidad de cada pixel. Permite filtrar el ruido en las capturas.
- Image Accumulation: indica cuántas imágenes se promedian para obtener la captura final. Mientras mayor sea, mejor será la calidad de la captura realizada (mejor definición y precisión), aunque va a tomar mayor tiempo para poder realizar la captura.
- Rango: esto se controla en la sección de Bit-depth en "Options". Indica cuál es el rango de distancias en que los píxeles van a ser considerados y van tener una coloración. Los píxeles fuera del rango van a tener un color negro.
- Flying pixel filter: otro tipo de filtro que se encarga de eliminar los píxeles que están lejanos a otros.

A continuación, se va a entregar los links para profundizar sobre el manejo de ArenaView y el uso de los "Features" para la cámara ToF: <https://support.thinklucid.com/arena-sdk-documentation/>, <https://www.youtube.com/watch?v=f9dCo6SlCFU> y <https://www.youtube.com/watch?v=1ZHcsbEsL6o> [30], [37] y [38].

3.4. Calibración Cámara RGB + ToF

Existen dos métodos para realizar la calibración entre las cámaras RGB y ToF de Lucid Vision. Uno es mediante la plataforma ArenaView (<https://support.thinklucid.com/knowledgebase/using-lucid-3d-rgb-> [39] y otro mediante el código de ejemplo en C:\ProgramData\Lucid Vision Labs\Examples\src (<https://support.thinklucid.com/app-note-helios-3d-point-cloud-with-rgb-color/>) [40]. A continuación, se explicarán los métodos utilizados por ambas cámaras para realizar esta calibración.

3.4.1. Calibración intrínseca cámara RGB

Para realizar la calibración intrínseca de la cámara RGB, se debe utilizar el objetivo de calibración entregado por el proveedor, disponible en el siguiente enlace: https://arenasdk.s3-us-west-2.amazonaws.com/LUCID_target_whiteCircles.pdf. Este es un objetivo de calibración circular, en el que se calcula la distancia entre los centros de los círculos blancos. **Es importante tener en cuenta que, a lo largo de la experimentación, se observó que, aunque en la calibración se considera que la distancia entre los centros de los círculos es de 50 mm, la distancia real es de 52 mm. Esta corrección se aplicó en el resto del código utilizado durante la experimentación.**

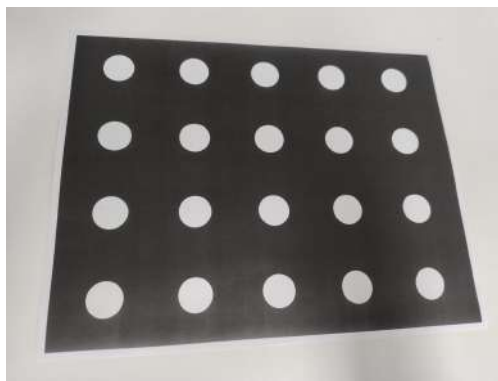


Figura 11: Objetivo de calibración

Durante el proceso de calibración intrínseca, se deben tomar 10 capturas del objetivo de calibración. Primero, se recomienda enfocar manualmente el lente de la cámara RGB usando la perilla, para ajustarlo a la distancia de captura. Luego, se debe colocar el objetivo de manera que la fila de 4 círculos esté horizontal, y mover el objetivo en distintos ángulos y distancias respecto a la cámara, abarcando idealmente todo el ángulo de visión de la cámara.

Este proceso se realiza de manera *automática* si se usa el código (C:\ProgramData\Lucid Vision Labs\Examples\src\Python Source Code Examples\py_HLTRGB_1_calibration.py), pero en ArenaView, se debe activar la cámara Tritón, presionar el botón de "stream" para grabar, y luego dirigirse a la pestaña "Options". Allí, se debe acceder a "Calibration", que se encuentra debajo de "RGB depth". Una vez verificado que el objetivo de calibración es el correcto, se debe presionar el botón "Grab Image" para tomar las capturas necesarias. En la ventana de "Log", se podrá verificar si la imagen fue correctamente capturada (es decir, si los círculos dentro del objetivo fueron detectados). Una vez obtenidas las 10 capturas, se debe presionar el botón "Complete Calibration" para finalizar el proceso.

Una vez tomadas estas capturas, se calculan los parámetros intrínsecos de la cámara, primero convirtiendo la imagen a escala de grises y luego utilizando la función `calibrate` del módulo de OpenCV https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga687a1ab946686f0d85ae0363b5af1d7b para calcular los valores intrínsecos de la cámara RGB [41]. Dependiendo de si se obtienen estos valores mediante código o mediante la GUI de ArenaView, se puede guardar el archivo como un archivo .yaml o .json. En el último caso, el archivo se guarda en la dirección C:\ProgramData\Lucid Vision Labs\ArenaView, con el nombre de serie de la cámara calibrada, para ser usado posteriormente sin tener que repetir el proceso.

3.4.2. Calibración extrínseca entre cámara RGB y ToF

Posteriormente, se debe calcular la posición relativa de la cámara ToF respecto a la cámara RGB, lo que se conoce como los parámetros extrínsecos entre las cámaras. Para ello, se debe tomar una captura con la cámara RGB del objetivo de calibración y otra captura con la cámara ToF, para evaluar la diferencia en posición entre ambas cámaras.

En el proceso de calibración extrínseca, se debe colocar el objetivo con las 4 filas de círculos de forma horizontal y las 5 columnas de forma vertical, además de asegurarse de que el objetivo esté bien enfocado en el caso de la cámara Tritón. **Es necesario considerar que se debe colocar el objetivo de calibración a una distancia adecuada y centrado entre ambas**

cámaras. También se recomienda situar el objeto sobre una superficie firme y no mover el objetivo entre capturas para evitar distorsiones en los resultados obtenidos. Luego, se debe realizar una captura con la cámara Helios configurando el formato de la cámara en `Coord3D_ABCY16`, el cual contiene información tanto de posición como de la intensidad de cada píxel (*puede indicar coloración o profundidad*). Una vez obtenida esta captura, se debe realizar una captura con la cámara Tritón en la configuración RGB8, con el objetivo de calibración en la misma posición que en el caso anterior.

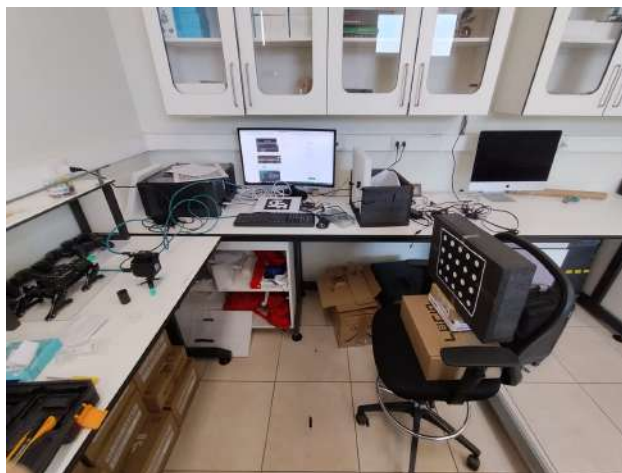


Figura 12: Captura para calibración extrínseca entre cámaras

Este proceso se realiza de forma automática mediante uno de los códigos de ejemplo (`C:\ProgramData\Lucid Vision Labs\Examples\src\Python Source Code Examples\py_HLTrGB_2_orientation.py`), pero en caso de realizarse en ArenaView, se debe dirigirse a la pestaña "Opciones" e ir a "Orientation" que está justo debajo de "Calibration". Una vez allí, se deben seguir los pasos indicados en la lista de esta sección, de arriba hacia abajo. Estos pasos son los siguientes:

1. Seleccionar la cámara ToF y luego ir a la pestaña de "Características" o "Features" para seleccionar el formato de píxel a `Coord3D_ABCY16` en el campo de "PixelFormat".
2. Regresar a la pestaña de "Orientation" y realizar la captura del objetivo de calibración.
3. Dentro de esa misma pestaña, seleccionar "Luminance False Coloration" y ajustar el "Coloring Algorithm" a "raw".
4. Una vez asegurado que el objetivo de calibración está dentro de la captura, ajustar el rango de coloración en la barra de abajo para asegurarse de que el objetivo sea visible y los círculos dentro de este se vean completamente blancos.
5. Presionar el botón "OK" y el botón "Pre-orient Helios2" para finalizar el proceso. Los logs indicarán si el proceso fue exitoso.

Para evitar errores, se recomienda cortar cualquier stream activo al momento de presionar los botones para orientar o calibrar. A continuación, se detallan los pasos para la cámara Tritón:

1. Seleccionar y activar la cámara Tritón, ajustando el formato de sus píxeles a RGB8, modificando el "PixelFormat" de la misma forma que se hizo con la cámara Helios.

2. Tomar una captura del objetivo y asegurarse de que este se vea correctamente.
3. Después de verificar esto, presionar el botón "OK" para completar el proceso para la cámara Tritón, así como el botón "Pre-orient Triton".
4. Presionar el botón "Complete Orientation" para finalizar el procedimiento de orientación.

Una vez que se tienen las capturas de ambas cámaras, se modifican para que la captura de la cámara RGB esté en escala de grises, mientras que la imagen de la cámara Helios se normaliza. Luego, se detectan los círculos de cada captura y se utiliza el método solvePnP de OpenCV para calcular los parámetros extrínsecos de la cámara. Dependiendo de si se obtienen estos valores mediante código o mediante la GUI de ArenaView, se puede guardar este archivo como un archivo .yaml o .json. En el último caso, el archivo se guarda en la dirección `C:\ProgramData\Lucid Vision Labs\ArenaView`, con el nombre de serie de la cámara Helios, los parámetros intrínsecos de la cámara Tritón y los parámetros extrínsecos de la cámara ToF respecto a la cámara RGB, para poder ser usado posteriormente sin necesidad de repetir el proceso.

3.4.3. Proyección de información RGB a cámara ToF

Una vez que se tengan calculados los parámetros extrínsecos e intrínsecos entre ambas cámaras, se puede realizar la proyección de la información de color de la cámara RGB sobre la información de profundidad de la cámara ToF. Para ello, se utiliza la función de OpenCV `projectPoints`, usando como entradas los parámetros previamente calculados. Este proceso se puede realizar automáticamente utilizando el código de ejemplo en `C:\ProgramData\Lucid Vision Labs\Examples\src\Python Source Code Examples\py_HLTRGB_3_overlay.py`, o mediante la plataforma ArenaView. En este caso, se debe ir a la pestaña 'RGB Depth' en la barra de 'Options', activar el botón 'Enable' y configurarlo como 'true'. Luego, se deben activar tanto la cámara Helios como la Tritón. Después, se selecciona la cámara Helios y se presiona el botón de grabar. Finalmente, se debe ir a la barra de 'Options', seleccionar la pestaña 'Pixel Formats' y presionar '3DView' bajo la sección 'Display View'.



Figura 13: Captura de prueba para proyección entre información RGB y cámara ToF

3.4.4. Toma de datos

Después de revisar de forma reiterada este proceso para mejorar los resultados de la proyección, se obtuvieron capturas de 5 objetos, presentados a continuación:

Durante la toma de datos, para cada objeto se realizaron capturas de la cámara RGB en formato RGB 8 (guardadas como imágenes), de la cámara ToF en formato `Coord3D_ABCY16` (guardadas como nubes de puntos en archivos `.ply`), y también una nube de puntos con la información proyectada entre ambas cámaras, en formato `Coord3D_ABCY16` (guardada como archivo `.ply`).

Como parte de las recomendaciones finales de este experimento, se deben considerar los problemas en el cálculo de profundidad generados por los bordes, así como cuando se coloca un objeto sobre los planos visibles en la escena de la cámara. Esto puede generar errores en la información capturada por la cámara ToF, creando una especie de 'sombra' o silueta en el resto de los planos. Además, se recomienda **configurar adecuadamente el modo de distancia de la cámara**, ya que, si la escena se extiende más allá del rango de la cámara, algunas partes pueden aparecer como ruido cerca de la cámara, dificultando la lectura de los datos. Por ello, se aconseja usar el modo de distancia con la máxima distancia posible entre la cámara y el resto de la habitación cuando se realicen capturas en interiores. Este ajuste no es tan crítico en exteriores.

3.5. Toma de datos objetos a distinta distancia

Para la toma de datos a distinta distancia, se consideraron dos escenarios: uno dentro del laboratorio y otro en el exterior a la luz. En el caso del laboratorio, solo se pudieron capturar datos a la máxima distancia disponible, que era de unos 4 metros. En el exterior, se realizaron capturas hasta la máxima distancia mencionada en las especificaciones de la cámara, es decir, 8.3 m. A continuación, se muestra el setup utilizado para las capturas tanto en interiores como en exteriores, colocando la cámara ToF sobre un trípode para facilitar su desplazamiento y capturando únicamente el objeto correspondiente a la figura de dinosaurio.

Debido a limitaciones en los cables disponibles para alimentar las cámaras en el exterior, solo se pudo usar la cámara ToF para capturar información de profundidad de las escenas. Por esta razón, tanto en interiores como en exteriores, solo se utilizó la cámara Helios2 Ray. La información se guardó en formato `Coord3D_ABCY16` y como nubes de puntos en archivos `.ply`. Además, se obtuvo una foto con un teléfono móvil para dar una idea de cómo era la escena al momento de la captura. Debido a que los parámetros de las cámaras que brindan mayor claridad y precisión varían según la distancia (como la exposición), se adjuntaron los parámetros principales modificados en un archivo `.txt`.

Para realizar las capturas, se mantuvo fija la cámara y se movió la plataforma sobre la cual se encontraba el objeto. Además, para añadir precisión a los datos obtenidos, se utilizó una huincha de medir para calcular la distancia entre el objeto y el lente de la cámara.



Figura 14: Captura con cámara ToF a distinta distancia de objeto

3.6. Captura de datos de objetos en distintos puntos con ArUco Marker

Este experimento consiste en tomar capturas desde distintos puntos de un objeto fijo usando la cámara ToF. Para ello, se situó el objeto sobre una silla y se montaron la cámara ToF y la cámara RGB (sobre la ToF) en un trípode. Luego, se movió el trípode alrededor del objeto para capturar distintas partes del objeto, tratando de cubrir la mayor parte de su superficie, exceptuando la parte inferior. Esto se puede observar a continuación:

Además, para poder sobreponer las capturas obtenidas en forma de nubes de puntos, se utilizó un ArUco Marker colocado debajo del objeto a escanear. Este marcador se dejó fijo debajo del objeto y se usó para obtener la posición relativa del ArUco Marker respecto a la cámara utilizando la función `solvePnP` y `aruco.DetectorParameters` de OpenCV. Esto permite alinear todas las capturas con el marco de referencia del ArUco Marker para que estén lo más alineadas posible. La siguiente imagen ilustra este proceso:

El procedimiento del experimento para cada pose deseada de un objeto es el siguiente:

1. Colocar la cámara en la posición deseada para realizar la captura con el trípode, mirando hacia el objeto (aproximadamente a unos 30-40 cm de distancia del objeto, tratando de variar en unos 45° de forma horizontal entre cada captura).
2. Retirar el objeto del campo de visión de la cámara y dejar el ArUco Marker en su lugar.
3. Conectar y activar la cámara Triton en ArenaView.
4. Presionar el botón de grabar y ajustar el Gain en la pestaña de "Features" de la cámara Triton para obtener una imagen clara (asegurándose de que los bordes del ArUco Marker se vean claramente en color negro).
5. Cerrar la pestaña de ArenaView.

6. Abrir el entorno de Conda desde la barra de comandos del sistema operativo, dirigirse a la carpeta en la que se desea almacenar las capturas y ejecutar el código con `python lector_aruco_RGB.py`.
7. Introducir en la barra de comandos la dirección en la que se desea guardar la captura de la pose, luego seleccionar la cámara a usar y, finalmente, ingresar la medida del lado del ArUco Marker en milímetros.
8. Aparecerá la imagen capturada y se proyectarán los ejes sobre el ArUco Marker. Una vez verificado que la posición y orientación de los ejes es correcta, se debe presionar cualquier tecla para salir de la ventana.
9. Escribir "y" si se considera que los ejes se capturaron correctamente; de lo contrario, presionar cualquier otra tecla.
10. La imagen capturada con la cámara RGB y la pose relativa del ArUco Marker respecto a la cámara se guardarán en la carpeta especificada.
11. Colocar el objeto en su lugar, conectar el cable de alimentación a la cámara ToF, activarla y realizar la captura de la nube de puntos.
12. Guardar la nube de puntos en la carpeta junto con la pose del ArUco Marker respecto a la cámara.

Una vez realizado este procedimiento, se guardarán las nubes de puntos en el formato de píxeles `Coord3D_ABCY16` y como archivos `.ply`. Para obtener una mejor calidad en los resultados de cada pose, es recomendable configurar los parámetros de la cámara ToF, como el aumento de la acumulación de imágenes, el modo de distancia, la exposición, y especialmente el filtro de umbral (*threshold*) para reducir el ruido en la nube de puntos. Si se desea evitar modificar constantemente el Gain de la cámara Triton, se recomienda realizar el experimento en interiores y mantener una iluminación constante (utilizando gain automático).

3.7. Superposición de nube de puntos

Para visualizar adecuadamente las nubes de puntos capturadas y poder editarlas, se usará el software CloudCompare. Esta plataforma permite cargar las nubes de puntos obtenidas en el experimento anterior y editarlas, mostrando los cambios realizados en tiempo real, lo que facilita su superposición y la creación de un escaneo completo de los objetos. Las herramientas de CloudCompare utilizadas para este procedimiento son las siguientes:

- **Clonar:** Copia de las nubes de puntos.
- **Segmentación:** Permite encerrar en un polígono la parte de la nube de puntos que se desea conservar y excluir la parte exterior al polígono, generando una nueva nube de puntos.
- **Transformación:** Aplica una transformación en términos de traslación y rotación para colocar la nube de puntos en una pose determinada. En este caso, se utilizará la transformación homogénea, introduciendo las matrices de rotación y traslación.

- **Mover:** Permite ajustar manualmente la rotación y la traslación de la nube de puntos utilizando el mouse.
- **Registro con ICP:** Algoritmo de registro local utilizado en CloudCompare para alinear de forma precisa nubes de puntos que ya han sido transformadas, con el fin de que se superpongan adecuadamente. Se especifica qué parte de la nube se desea alinear.
- **Merge:** Combina dos nubes de puntos en una sola nube.
- **Calcular normales (Octree):** Calcula las normales de la nube de puntos a partir de los puntos vecinos (se puede seleccionar el número de puntos vecinos utilizados), lo que permite crear un mesh a partir de la nube de puntos.
- **Reconstrucción de superficie de Poisson (Poisson Recon):** Genera la superficie de un mesh a partir de la nube de puntos y las normales calculadas previamente. Se pueden configurar los niveles utilizados por el algoritmo y el tipo de superficie para la reconstrucción.

Para un mejor uso de esta plataforma, se recomienda consultar los siguientes enlaces que explican algunas partes del procedimiento a continuación: <https://www.youtube.com/watch?v=m43usERF33M> y <https://www.youtube.com/watch?v=0OcN-lNChlA>.

A continuación, se describen los pasos realizados para crear el escaneo completo de un objeto a partir de las capturas con la cámara ToF:

1. Cargar todas las capturas de las nubes de puntos de un determinado objeto en CloudCompare. Se debe tener en cuenta que la posición de la cámara en la nube de puntos estará en el origen del sistema de coordenadas global y la nube de puntos tendrá la orientación de los ejes dispuestos en OpenCV (eje x hacia la derecha, y hacia abajo y z hacia adelante respecto a la cámara).
2. Seleccionar todas las nubes de puntos y clonarlas para editarlas con mayor facilidad (en caso de cualquier error, se pueden copiar las nubes originales nuevamente).
3. Ejecutar el código llamado `transformacion_inversa_marco_tof_final.py` e introducir las direcciones absolutas de los archivos con las poses calculadas con la cámara Tritón para cada captura del objeto. Este código obtiene la pose del ArUco Marker respecto a la cámara ToF (en lugar de la Tritón) y calcula la inversa de la pose para obtener la transformación que pasa la nube de puntos del sistema de referencia de la cámara al sistema de referencia del ArUco Marker. Estas transformaciones se guardarán en un archivo `.txt` en la dirección especificada en la barra de comandos.
4. Seleccionar la nube de puntos que se desea mover y copiar y pegar su transformada respectiva dentro de la herramienta de transformación para colocarla en la pose deseada. Se recomienda guardar esta nube para evitar problemas en caso de cometer un error en los pasos posteriores.
5. Realizar la segmentación de forma manual utilizando la herramienta de segmentación para dejar el sector deseado de la nube de puntos.
6. Seleccionar en el visualizador dos nubes de puntos al mismo tiempo y realizar el ajuste fino de forma manual entre estas nubes mediante la herramienta de mover.

7. Aplicar el registro ICP entre ambas nubes de puntos para mejorar el ajuste entre ellas. Dado que las capturas abarcan partes diferentes del objeto y tienen solo una sección que se superpone, se recomienda usar una superposición de aproximadamente un 50 % (se puede ajustar según sea necesario).
8. Una vez que se haya asegurado que las nubes de puntos estén alineadas correctamente, se debe usar la herramienta *Merge* para combinarlas en una sola. Se recomienda segmentar los bordes de las capturas antes de hacer el merge y usar el filtro ROI para eliminar el ruido de las nubes de puntos.
9. Realizar de forma iterativa los pasos 6 a 8 hasta que todas las nubes de puntos estén unidas en una sola nube completa del objeto.
10. Usar la herramienta de cálculo de normales Octree para obtener las normales necesarias para crear el mesh de la nube. Se recomienda ajustar el número de puntos vecinos utilizados para obtener mejores resultados.
11. Opcional: colorear toda la nube de puntos antes de realizar el siguiente paso para distinguir mejor las partes del objeto (por ejemplo, en el caso de usar solo la cámara ToF sin la RGB).
12. Seleccionar la nube con las normales obtenidas y utilizar la herramienta de *Poisson Recon* para construir la superficie del objeto. Es importante tener en cuenta que, al configurar un nivel más bajo en el algoritmo, el mesh tendrá menos detalle pero estará menos afectado por el ruido.

Este procedimiento se repite con cada nube de puntos hasta obtener un mesh completo de cada captura. Debido a ciertos márgenes de error en la medición de distancia entre la cámara y el ArUco Marker, puede haber algunas desviaciones que impidan que las nubes de puntos se alineen perfectamente después de aplicar la transformación, por lo que será necesario realizar ajustes manuales. Sin embargo, si se verificó correctamente la orientación al capturar el ArUco Marker, se debería obtener la orientación correcta de la nube de puntos después de la transformación. Finalmente, los meshes calculados pueden guardarse como archivos .ply y ser subidos y editados en CloudCompare.

3.8. Capturas de cámara de polarización con cereza

Para realizar la captura con la cámara de polarización, se deben seguir los siguientes pasos:

1. Colocar la cámara de polarización sobre la cámara ToF como soporte (en lugar de la cámara RGB). Luego, se debe establecer el setup de la cámara. Para esto, primero se conecta la cámara ToF al trípode.
2. Ensamblar dos placas, una de color azul y otra de color rojo (controlador RAMPS1.4), uniéndolas mediante sus pines.
3. Conectar una fuente de poder switching al enchufe. Esta fuente debe estar conectada, a su vez, a la placa Arduino de color rojo, y la placa roja debe conectarse mediante un cable USB al computador.

4. Tomar el servomotor de color rojo en el que se colocará la cereza y conectarlo mediante un cable con varios pines a la placa Arduino roja (este cable debe conectarse a pines específicos dentro de la placa).
5. Colocar la cereza sobre el servomotor, insertándola en las patas que sobresalen de este (para ello, se deben hacer unos agujeros en la cereza).
6. Una vez colocada la cereza sobre el servomotor, este debe ser instalado dentro de una estructura de plumavit de color negro, en donde se encuentran las luces que se utilizarán con la cámara.
7. Conectar un enchufe de la estructura a la alimentación.
8. Conectar la cámara a la alimentación y al computador, ubicándola a la entrada de la estructura y apuntando hacia la cereza (idealmente, a la misma altura que la cereza).



Figura 15: Setup completo de capturas con cámara de polarización para cereza.



Figura 16: Posición de la cámara de polarización y del servomotor con la cereza en el setup.

Una vez que el setup esté listo, se abre Anaconda Prompt en el computador y se activa el ambiente de **ariel** utilizando el comando **conda activate ariel**. A continuación, se debe seleccionar el código a utilizar. Antes de ejecutar este código, asegúrese de que entre poca luz en la estructura, idealmente apagando las luces si se encuentra en un lugar cerrado. Luego, se debe encender una de las luces del setup. Finalmente, ejecute el código seleccionado, especificando la carpeta en la que se guardarán las capturas, la luz que está encendida en la estructura y la configuración deseada para la cámara.

4. Resultados

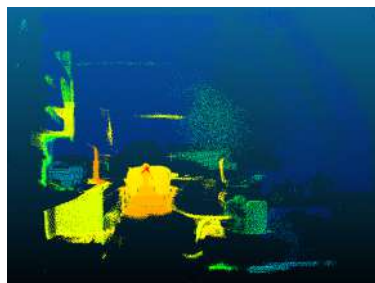
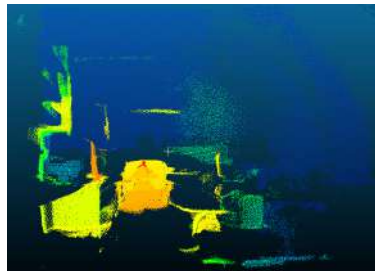
Los objetos que se utilizarán a lo largo de los experimentos se muestran en la siguiente imagen:

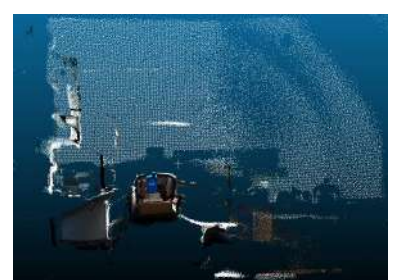
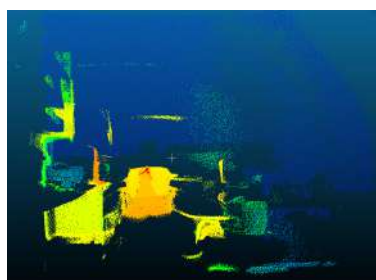
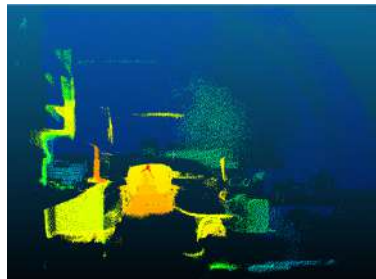
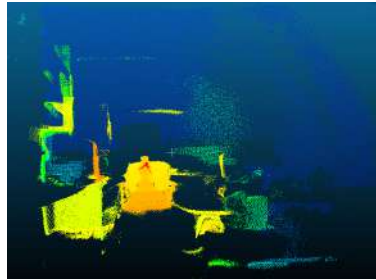


Figura 17: Objetos usados a lo largo de los experimentos

4.1. Calibración Cámara RGB + ToF

A continuación, se presenta una imagen para cada objeto (por fila): primero, la imagen de la cámara RGB; luego, la nube de puntos de la cámara ToF; y finalmente, la superposición entre la información RGB y la nube de puntos de la cámara ToF. El primer objeto es un juguete de dinosaurio, el segundo es un desodorante Lisoform, el tercero es un jarrón, el cuarto es una taza negra y, finalmente, el último es una caja azul.





4.2. Captura ToF de objeto a distinta distancia

En las siguientes imágenes se presentarán las capturas obtenidas a distintas distancias. En la primera columna se muestran las capturas realizadas en interiores y en la segunda columna, las capturas realizadas en el exterior. Cada fila corresponde a una determinada distancia, la cual se especificará en la parte inferior de cada imagen. El objeto utilizado para realizar las capturas es una figura de dinosaurio, aunque en algunas capturas se le colocó una cereza entre sus espinas para verificar si se podía distinguir adecuadamente.

Además, después de la primera tabla, se muestran las imágenes obtenidas en el exterior desde 5 metros hasta la máxima distancia disponible para la cámara ToF (Tabla 15 hasta la Tabla 18).

Tabla 1: Comparación de capturas a diferentes distancias (0.1m - 1m)


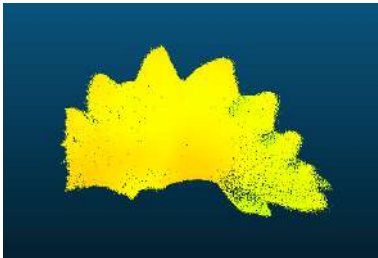
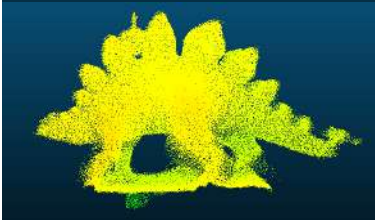
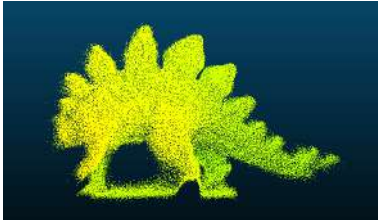
Distancia	Captura Interior	Captura Exterior
0.1m		
0.2m		

Tabla 2: Comparación de capturas a diferentes distancias (0.1m - 2m)

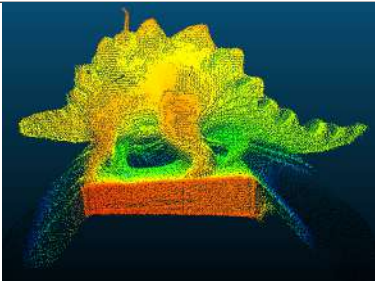

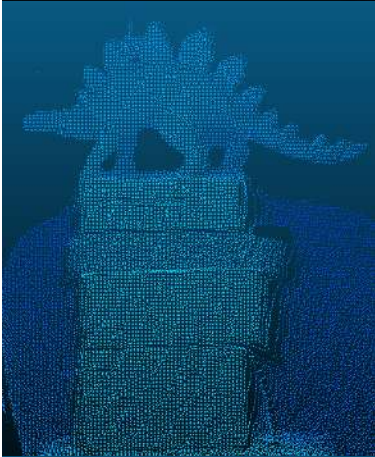
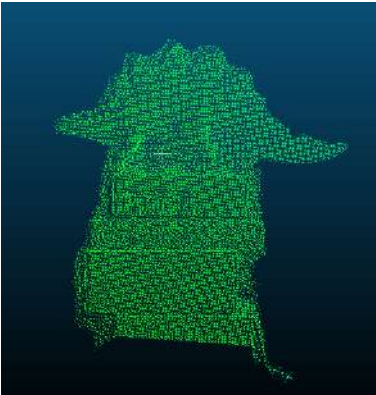
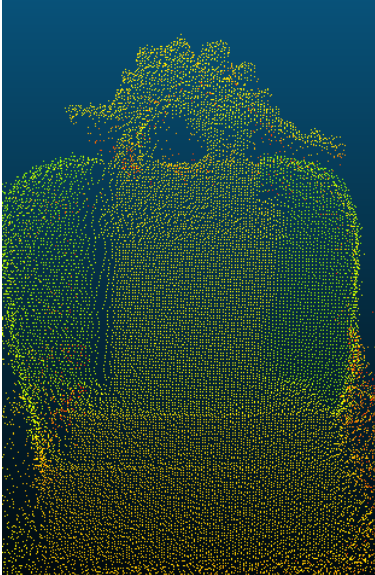
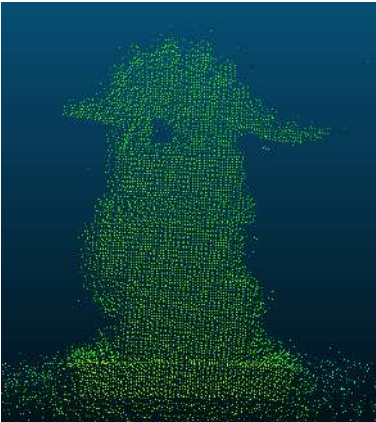
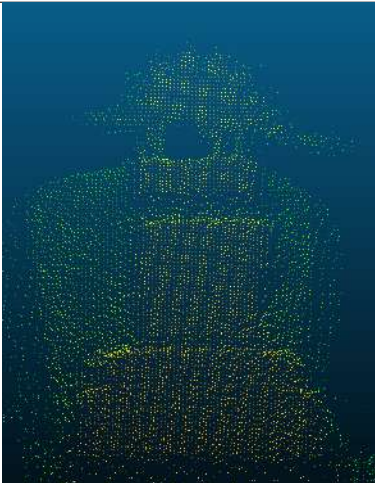

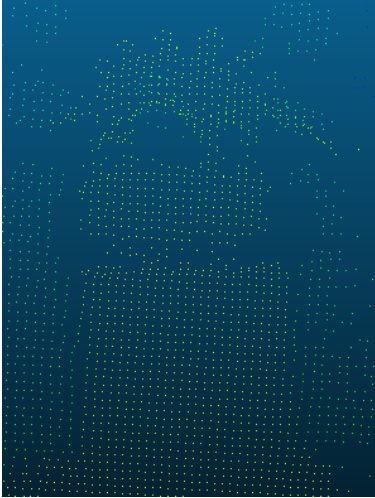

Distancia	Captura Interior	Captura Exterior
0.3m		
1m		
2m		

Tabla 3: Comparación de capturas a diferentes distancias (3m - 4m)

Distancia	Captura Interior	Captura Exterior
3m		
4m		

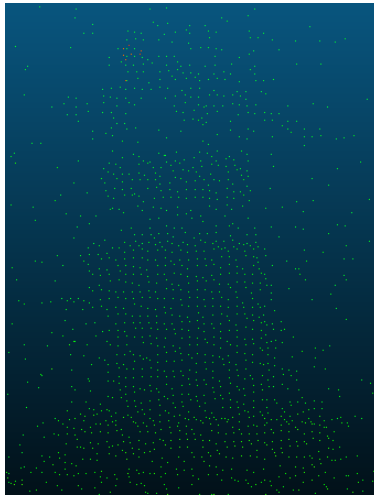


Figura 18: Captura dinosaurio interior 5m

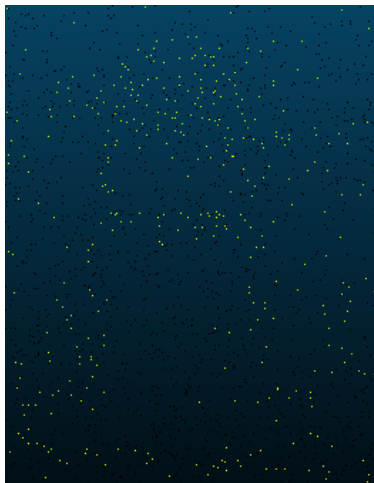


Figura 19: Captura dinosaurio exterior 6m

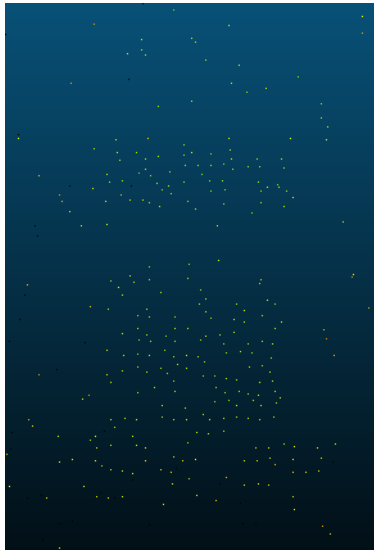


Figura 20: Captura dinosaurio interior 7m

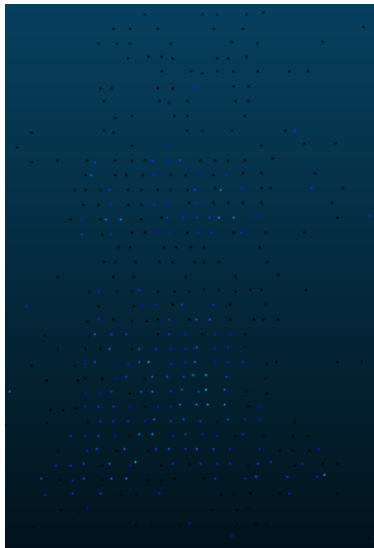
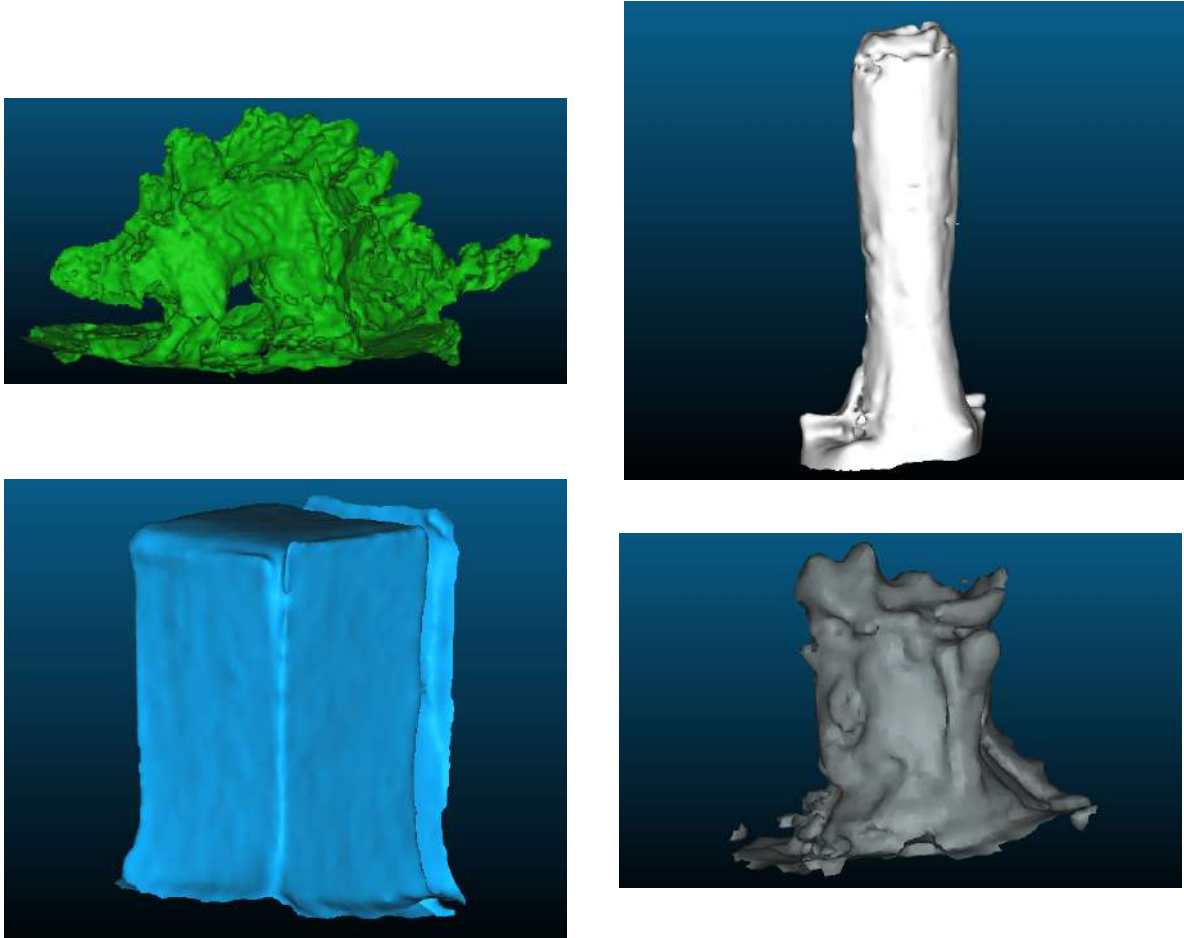


Figura 21: Captura dinosaurio exterior 8.1m

4.3. Superposición de nubes de puntos de distintos puntos de vista de objetos

A continuación, se presentan los meshes creados a partir de las capturas realizadas desde distintos puntos de vista de los objetos. A continuación, se muestran las capturas de los resultados:

Tabla 4: Mesh objetos experimentos (superior izquierda: dinosaurio, superior derecha: lisoform, inferior izquierda: caja e inferior derecha: taza negra)



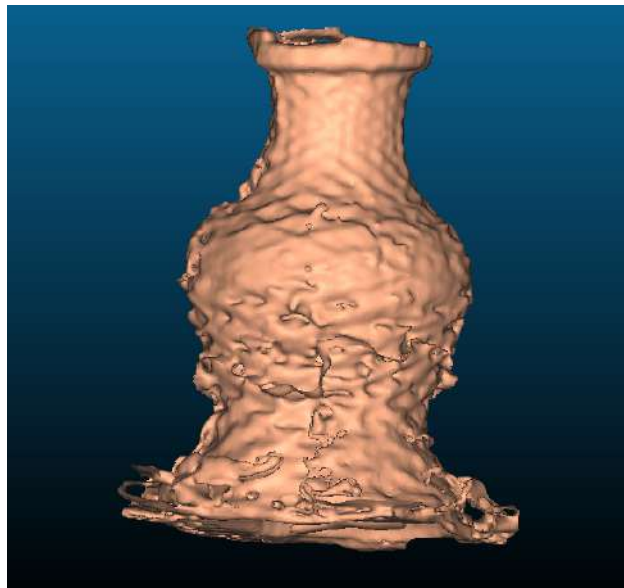


Figura 22: Mesh de jarrón

5. Análisis de resultados

A continuación, se van a analizar los resultados obtenidos en los diversos experimentos realizados a lo largo de la práctica.

5.1. Calibración Cámara RGB + ToF

En primer lugar, se puede observar que las capturas obtenidas por la cámara ToF de forma independiente son congruentes con lo que se puede observar en las imágenes de la cámara RGB. Aunque se presentan algunas incongruencias y ruido, esto se puede explicar por los defectos en la cámara con ciertos colores, objetos transparentes y los bordes de los objetos. No obstante, se puede notar que todos los píxeles que se encuentran en la nube de puntos sin información RGB también están en la cámara RGB, lo que indica que se pudo realizar una correcta calibración extrínseca entre ambas cámaras.

También se puede apreciar que la información RGB proyectada en la nube de puntos presenta una buena coherencia al compararla con la imagen de la cámara RGB. Este comportamiento es bastante homogéneo entre las capturas de los distintos objetos utilizados. Aunque no se puede ver de forma adecuada en estas capturas, se presentan algunas deformaciones e imprecisiones en la nube de puntos de ciertos objetos, especialmente el desodorante Lisoform (sobreexposición) y la taza negra (mucha desconfianza y mucho ruido). Sin embargo, al presentarse tanto en la nube de puntos sin información RGB, se puede concluir que el proceso de captura sigue siendo relativamente válido. Finalmente, se puede mencionar que hay algunos espacios en blanco en la nube de puntos, pero esto se debe a la distancia entre la cámara y los objetos, lo que impide que la cámara pueda ver los objetos que están atrás.

5.2. Captura ToF de un objeto a distinta distancia

Al comparar las capturas tomadas con la cámara en el interior y el exterior, se puede observar que el alcance mínimo de la cámara en el exterior es bastante similar al alcanzado en el interior. A pesar de que en la captura a 0.1 m en el interior no se puede ver el dinosaurio, sino su interior, se puede observar su contorno claramente. Esta diferencia entre ambas capturas se puede explicar por el hecho de que, accidentalmente, al realizar la captura del dinosaurio en el interior a 0.1 m, se usó un mayor tiempo de exposición que el de la captura en el exterior, por lo que el fenómeno presentado en el interior se puede asociar a la sobreexposición de la cámara. Además, se puede mencionar que no se puede comparar la máxima distancia en interiores con la del exterior debido a la limitación en el tamaño de la habitación en la que se realizaron las capturas en el interior.

También se puede observar que, en general, no hay grandes diferencias entre las tomas tanto en el interior como en el exterior, tanto en el número de puntos que rebotan del dinosaurio como en la precisión de la figura. Sin embargo, al aumentar la distancia en el exterior, se observa un mayor ruido de fondo, además de una mayor imprecisión en la profundidad asignada a las partes de los objetos, de forma bastante homogénea entre ambos entornos (lo que implica una menor confianza en los píxeles calculados). Por lo tanto, se puede colegir que, si bien el diseño de la cámara debería funcionar mejor en exteriores que en interiores, no se presentan grandes diferencias entre estos cuando se capturan objetos no reflectivos ni muy oscuros.

Por otro lado, se puede deducir que la máxima distancia en la que se puede diferenciar adecuadamente las partes de la figura corresponde a los 2 metros, por lo que se podría considerar como el

máximo alcance para realizar aplicaciones de mayor precisión, aunque se recomienda usar menores distancias debido a lo disperso que se vuelve la nube de puntos a esta distancia.

5.3. Superposición de nubes de puntos de distintos puntos de vista de objetos

A partir de los resultados, se puede observar que se puede crear una buena reconstrucción de los objetos al poder combinar las capturas de los objetos con la cámara ToF. Sin embargo, se presenta un considerable ruido dentro de las capturas, lo que genera una serie de relieves y texturas que no se aprecian en el objeto real. Esto se puede asociar principalmente a la falta de mayor precisión en la posición del objeto relativa a la cámara, proporcionada por el código que reconoce el ArUCo Marker, lo que provoca que no se puedan superponer de manera perfecta las nubes de puntos, incluso con el uso del método ICP y el ajuste manual entre las nubes.

Por otro lado, también contribuye al ruido en las imágenes los distintos fenómenos ópticos que afectan a las nubes de puntos generadas por la cámara ToF, como el ruido generado en el borde y los problemas con las superficies reflectantes y de color oscuro. Esto se observa de forma más pronunciada en el mesh de la taza negra, puesto que esta es de color oscuro y, a la vez, reflectante, tiene una superficie redonda y detalles con bordes bastante pronunciados. Es por esta razón que, incluso con las distintas configuraciones intentadas con la taza, se generaron capturas muy ruidosas y con poca exactitud y confianza en los valores de profundidad estimados por la cámara.

Finalmente, los meshes en 3D que se crearon tuvieron que ser cambiados de color al procesar los datos, puesto que no se pudo usar la cámara RGB al mismo tiempo que la cámara ToF para superponer la información entre estas.

6. Conclusión

Por medio de la investigación y los experimentos realizados, se obtuvo una comprensión más profunda del funcionamiento de las cámaras ToF y de polarización, sus beneficios y desventajas, la forma de usarlas en conjunto con otras cámaras y cómo manejar de forma específica las cámaras de Lucid Vision. Además, a través de los experimentos realizados, se pudo verificar el rendimiento de las cámaras en un ambiente real en comparación con lo establecido en la documentación. Respecto a lo anterior, se puede observar que, a pesar de que se presentan algunas diferencias, se comprueba la calidad de las cámaras utilizadas.

Después de practicar con las cámaras empleadas, se pudo familiarizar con las restricciones y limitaciones que posee cada una de ellas. Asimismo, se demostró que, con el conocimiento adecuado, estas limitaciones y problemas pueden ser gestionados de mejor manera. Además, se pudo notar que la implementación de estas cámaras en conjunto, su alcance y resolución son adecuadas para caracterizar de forma bastante completa los objetos estudiados. Es por lo anterior que se puede concluir que las cámaras en el laboratorio son adecuadas para su uso en la agricultura, en particular, para la cosecha y verificación de calidad de las cerezas, aunque se requieren varias consideraciones para su aplicación.

A pesar de las restricciones de los experimentos realizados debido a la duración de la práctica, se deduce que hay una gran cantidad de aplicaciones y experimentos que se pueden realizar con estas cámaras para seguir comprobando sus capacidades y encontrar nuevas aplicaciones dentro del área de la agricultura. Finalmente, se agradece la oportunidad de aprender sobre estas cámaras y su uso, además de poder contribuir a esta área en desarrollo en Chile.

7. Bibliografía

1. "How Cameras Work," HowStuffWorks. [Online]. Available: <https://electronics.howstuffworks.com/camera.htm>. [Accessed: 16-Mar-2025].
2. "Triton 32 MP IMX265," Lucid Vision Labs. [Online]. Available: <https://thinklucid.com/product/triton-32-mp-imx265/>. [Accessed: 16-Mar-2025].
3. L. Li, "Time-of-Flight Camera – An Introduction," Texas Instruments, Technical White Paper SLOA190B, Jan. 2014, revised May 2014. [Online]. Available: <https://www.ti.com/lit/wp/sloa190b/sloa190b.pdf>. [Accessed: 16-Mar-2025].
4. "Time-of-Flight Sensor vs. LiDAR: What Are the Differences?" PMT. [Online]. Available: <https://pmt-fl.com/time-of-flight-sensor-vs-lidar-what-are-the-differences/>. [Accessed: 16-Mar-2025].
5. S. Feng, "Understanding the Principle of 3D Time-of-Flight Camera Technology," Goer-Micro, Nov. 2024. [Online]. Available: <https://industry.goermicro.com/blog/tech-briefs/understanding-the-principle-of-3d-time-of-flight-camera.html>. [Accessed: 16-Mar-2025].
6. "Helios2 Ray Outdoor Time-of-Flight 3D Camera," Lucid Vision Labs. [Online]. Available: <https://thinklucid.com/product/helios2-ray-outdoor-tof-ip67-3d-camera/>. [Accessed: 16-Mar-2025].
7. J. Smith, "Advancements in 3D Imaging Technologies," arXiv preprint arXiv:2303.14840, Mar. 2023. [Online]. Available: <https://arxiv.org/pdf/2303.14840>. [Accessed: 16-Mar-2025].
8. "Understanding Time-of-Flight Cameras," YouTube. [Online]. Available: <https://www.youtube.com/watch?v=f9dCo6SlCFU>. [Accessed: 16-Mar-2025].
9. "Helios2 Ray Outdoor ToF IP67 3D Camera," Lucid Vision Labs. [Online]. Available: <https://thinklucid.com/product/helios2-ray-outdoor-tof-ip67-3d-camera/>. [Accessed: 16-Mar-2025].
10. "Time-of-Flight Camera Technology Explained", YouTube. [Online]. Available: <https://www.youtube.com/watch?v=I-QFjwmiwgk>. [Accessed: 16-Mar-2025].
11. "Polarization Explained: The Sony Polarized Sensor," Lucid Vision Labs. [Online]. Available: <https://thinklucid.com/tech-briefs/polarization-explained-sony-polarized-sensor/>. [Accessed: 16-Mar-2025].
12. M. T. Eismann, "Pola4All: Survey of Polarimetric Applications and an Open-Source Polarization Camera Dataset," Journal of Electronic Imaging, vol. 33, no. 1, p. 010901, Jan. 2024. [Online]. Available: <https://doi.org/10.1117/1.JEI.33.1.010901>. [Accessed: 16-Mar-2025].
13. "PolarSens: Polarization Image Sensor," Sony Semiconductor Solutions Corporation. [Online]. Available: <https://www.sony-semicon.com/en/technology/industry/polarsens.html>. [Accessed: 16-Mar-2025].
14. "Polarization Cameras: A New Dimension of Vision," The Imaging Source, White Paper, Version 1.1, 2022. [Online]. Available: https://s1-dl.theimagingsource.com/api/2.5/packages/publication/whitepaper-camera/wppolcam/e6dbe889-59bd-5130-9824-30722186ab3d/wppolcam_1.1.en_US.pdf. [Accessed: 16-Mar-2025].

15. "Polarized Camera Resource Center," Lucid Vision Labs. [Online]. Available: <https://thinklucid.com/polarized-camera-resource-center/>. [Accessed: 16-Mar-2025].
16. "Phoenix 5.0 MP Polarization Model IMX264MZR/MYR," Lucid Vision Labs. [Online]. Available: <https://thinklucid.com/product/phoenix-5-0-mp-polarization-model-imx264mzrmyr/>. [Accessed: 16-Mar-2025].
17. P. Corke, Robotics, Vision and Control: Fundamental Algorithms in MATLAB, 2nd ed. Springer, 2017. [Online]. Available: <https://doi.org/10.1007/978-3-319-54413-7>. [Accessed: 16-Mar-2025].
18. K. Hata and S. Savarese, "CS231A Course Notes 1: Camera Models," Stanford University. [Online]. Available: https://web.stanford.edu/class/cs231a/course_notes/01-camera-models.pdf. [Accessed: 16-Mar-2025].
19. "Camera Calibration and 3D Reconstruction," OpenCV Documentation. [Online]. Available: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html. [Accessed: 16-Mar-2025].
20. "Camera Calibration," OpenCV Documentation. [Online]. Available: https://docs.opencv.org/4.x/d4/d94/tutorial_camera_calibration.html. [Accessed: 16-Mar-2025].
21. "Helios 3D Point Cloud with RGB Color," Lucid Vision Labs. [Online]. Available: <https://support.thinklucid.com/app-note-helios-3d-point-cloud-with-rgb-color/>. [Accessed: 16-Mar-2025].
22. "Detection of ArUco Markers," OpenCV Documentation. [Online]. Available: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html. [Accessed: 16-Mar-2025].
23. "Global Registration," Open3D Documentation. [Online]. Available: https://www.open3d.org/html/tutorial/pipelines/global_registration.html. [Accessed: 16-Mar-2025].
24. J. Doe, "Advanced Techniques in Computer Vision," arXiv preprint arXiv:2103.02690, Mar. 2021. [Online]. Available: <https://arxiv.org/pdf/2103.02690>. [Accessed: 16-Mar-2025].
25. "Computing Normals", CloudCompare Wiki. [Online]. Available: <https://cloudcompare.org/doc/wiki/index.php?title=Normals%5CCompute>. [Accessed: 16-Mar-2025].
26. "CloudCompare Octree", CloudCompare Wiki. [Online]. Available: https://www.cloudcompare.org/doc/wiki/index.php/CloudCompare_octree. [Accessed: 16-Mar-2025].
27. "Poisson Surface Reconstruction (Plugin)", CloudCompare Wiki. [Online]. Available: [https://www.cloudcompare.org/doc/wiki/index.php/Poisson_Surface_Reconstruction_\(plugin\)](https://www.cloudcompare.org/doc/wiki/index.php/Poisson_Surface_Reconstruction_(plugin)). [Accessed: 16-Mar-2025].
28. H. Hoppe, "Poisson Surface Reconstruction", Microsoft Research. [Online]. Available: <https://hhoppe.com/poissonrecon.pdf>. [Accessed: 16-Mar-2025].
29. "Downloads Hub", Lucid Vision Labs. [Online]. Available: <https://thinklucid.com/downloads-hub/>. [Accessed: 16-Mar-2025].

30. "Arena SDK Documentation", Lucid Vision Labs Support. [Online]. Available: <https://support.thinklucid.com/arena-sdk-documentation/>. [Accessed: 16-Mar-2025].
31. "Third-Party Software", Lucid Vision Labs Support. [Online]. Available: <https://support.thinklucid.com/#3rd-party>. [Accessed: 16-Mar-2025].
32. "Getting Started with Helios Connect", Lucid Vision Labs Support. [Online]. Available: <https://support.thinklucid.com/getting-started/#gs-Helios-Connect>. [Accessed: 16-Mar-2025].
33. "Introduction to Lucid Vision Cameras", YouTube. [Online]. Available: <https://www.youtube.com/watch?v=KyQD-z0Km00>. [Accessed: 16-Mar-2025].
34. "M8 GPIO Power Supply (NA/EU)", Lucid Vision Labs. [Online]. Available: <https://thinklucid.com/product/m8-gpio-power-supply-na-eu/>. [Accessed: 16-Mar-2025].
35. "M12 to RJ45 IP67 Cat6a Cable (2.0m, Dark Green)", Lucid Vision Labs. [Online]. Available: <https://thinklucid.com/product/m12-to-rj45-ip67-cat6a-cable-amphenol-2-0m-dark-green/>. [Accessed: 16-Mar-2025].
36. "Getting Started with Phoenix Connect", Lucid Vision Labs Support. [Online]. Available: <https://support.thinklucid.com/getting-started/#gs-phoenix-connect>. [Accessed: 16-Mar-2025].
37. "Understanding Time-of-Flight Cameras", YouTube. [Online]. Available: <https://www.youtube.com/watch?v=f9dCo6SlCFU>. [Accessed: 16-Mar-2025].
38. "Lucid Vision Labs Product Overview", YouTube. [Online]. Available: <https://www.youtube.com/watch?v=1ZHcsbEsL6o>. [Accessed: 16-Mar-2025].
39. "Using Lucid 3D RGB IP67 Kit in ArenaView", Lucid Vision Labs Support. [Online]. Available: <https://support.thinklucid.com/knowledgebase/using-lucid-3d-rgb-ip67-kit-in-arenaview/>. [Accessed: 16-Mar-2025].
40. "Camera Calibration and 3D Reconstruction", OpenCV Documentation. [Online]. Available: https://docs.opencv.org/4.x/d9/d0c/group___calib3d.html#ga687a1ab946686f0d85ae0363b5af1d7b. [Accessed: 16-Mar-2025].
41. "Helios 3D Point Cloud with RGB Color", Lucid Vision Labs Support. [Online]. Available: <https://support.thinklucid.com/app-note-helios-3d-point-cloud-with-rgb-color/>. [Accessed: 16-Mar-2025].