UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Diseño

Análisis de la reconstrucción 3D mediante técnicas de Visión Artificial

Autor: Marc Ribera Vilaplana

Tutor: Eugenio Ivorra Martínez

Titulación: Grado en Ingeniería Electrónica Industrial y Automática

Fecha: Julio de 2021

Quiero expresar mi agradecimiento a todos aquellos compañeros y profesores del grado, que me han formado y visto crecer dentro del ámbito de la ingeniería electrónica. En especial, quiero destacar, la implicación de Eugenio, que ha sabido guiarme y darme el ánimo para afrontar a aquellos problemas que durante el proyecto veía que me superaban.

RESUMEN

Las reconstrucciones tridimensionales realizadas a través de cámaras de visión 3D son de gran utilidad a la hora de recrear objetos o estancias. Obteniendo un modelo tridimensional óptimo, el objeto escaneado se puede visualizar tantas veces como sea necesario, recordando información que a primera vista no es relevante. El presente proyecto recopila las diferentes metodologías aplicadas para crear sistemas de visión artificial, centrándose en la estereovisión, técnica utilizada para el desarrollo del proyecto.

Las cámaras utilizadas para los escaneos son los modelos comerciales *Intel Realsense Depth Camera D415* y la *Microsoft Azure Kinect DK*. Además, el desarrollo de software se ha implementado con *Python 3.7* y *RTABMap*.

Tras la reconstrucción, los modelos son analizados, definiendo sus puntos fuertes, débiles y aquellos aspectos que no difieren en calidad respectivamente. Pudiendo comparar objetivamente los modelos obtenidos con ambas cámaras y demostrando las ventajas de las técnicas utilizadas.

RESUM

Les reconstruccions tridimensionals realitzades amb càmeres de visió 3D son de gran utilitat a l'hora de recrear objectes o estàncies. Mitjançant un model tridimensional òptim, el objecte escanejat pot ser visualitzat tantes vegades con sigui necessari, recordant informació que amb la primera ullada no s'ha considerat rellevant. El projecte acull les diferents metodologies utilitzades per a crear sistemes de visió artificial, centrant-se a la estereovisió, tècnica utilitzada per a desenvolupar el projecte.

Les càmeres utilitzades son els models comercials *Intel Realsense Depth Camera D451* i la *Microsoft Azure Kinect DK*. Respecte el desenvolupament de software realitzat, ha segut implementat mitjançant *Python 3.7* y *RTABMap*.

Una vegada reconstruïts, els models son analitzats, definint els punts forts, febles y aquells aspectes que no varien respectivament. Així es comparen els models obtinguts amb les dues càmeres y es demostren els avantatges de les tècniques utilitzades.

ABSTRACT

Three-dimensional reconstruction done by 3D artificial vision cameras are really used when remembering an object or scene is needed. Having access to an optimised three-dimensional model of the scanned object, allows users to visualize it, as many times as needed, catching missing information not taken on first look. This project gets information about methodologies used when building an artificial vision system, going deeper into stereovision, which is the technique used all along the project.

The 3D cameras used on the project are commercial ones, *Intel Realsense Depth Camera D415* and *Microsoft Azure Kinect DK*. Besides, software development has been done with *Python 3.7* and *RTABMap*.

After the reconstruction, models are analysed, extracting their advantages, drawbacks, and some characteristics where they are not different at all. 3D Models reconstructed from both cameras have been objectively compared assuring their strengths are known.

Índice de documentos

DOCUMENTO 1:

MEMORIA

Documento 1 – MEMORIA DESCRIPTIVA

Listado de Figuras

Listado de Tablas

1. Descripción general del problema

1.1. Objeto

El presente proyecto pretende desarrollar y analizar sistemas de visión artificial de bajo coste. En concreto se centrará en la reconstrucción de diferentes estancias comunes encontradas en edificios empresariales, públicos o viviendas, como baños, salas de reuniones, comedores, cocinas... La reconstrucción de las estancias se desarrollará con dos mecanismos diferentes de reconstrucción 3D, pudiendo así comparar los modelos obtenidos.

Además de utilizar diferentes técnicas de reconstrucción, se utilizarán dos modelos comerciales de cámaras 3D, la *Intel REALSENSE DEPTH CAMERA D415* y la *Microsoft Azure Kinect DK*. Pudiendo obtener así hasta 4 posibles reconstrucciones por cada estancia analizada, si se realiza en análisis con ambas cámaras en la misma habitación.

Como se viene demostrando en los últimos años, los sistemas visión artificial tienen multitud de aplicaciones, viéndose integrados ampliamente en procesos de control de calidad. En cuanto a la visión artificial aplicada en la reconstrucción 3D, encontramos cantidad de utilidades potenciales que aún están por explotar como el análisis de estancias para su posterior tasación económica, donde convertimos el modelo 3D en un archivo de consulta de pequeños detalles que a primera vista pueden pasar desapercibidos o el trazado de planos en planta de las estancias a partir de dicha reconstrucción.

Además, fuera del ámbito de aplicación del proyecto, la reconstrucción 3D puede utilizarse como una herramienta de análisis de objetos permitiendo su posterior recreación, en base al modelo 3D, o visualización, si se combina con técnicas de realidad aumentada.

1.2. Alcance

El presente proyecto contempla la obtención, extracción y procesamiento de las imágenes obtenidas por cámaras de visión 3D, obteniendo como resultado una reconstrucción 3D de los contenidos grabados por las cámaras. En la memoria, se explicarán los métodos utilizados, la

descripción de los parámetros modificados y el posterior análisis de la solución final. Pudiendo ser recreado por toda aquella persona o entidad que requiera de los medios necesarios.

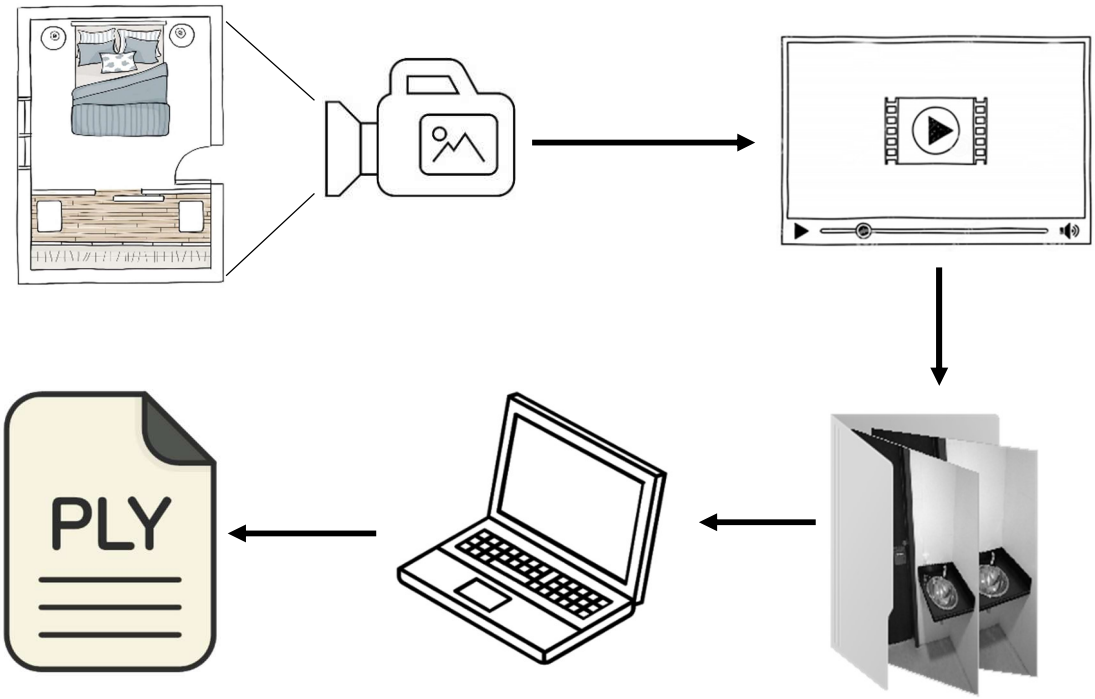


***Figura 1**: Esquema del proceso de obtención del modelo 3D*

2. Visión Artificial

2.1. Origen y concepto

Encontramos el origen de la visión artificial alrededor de los años 60, cuando se empiezan a crear prototipos basados en una serie cámaras que captaban imágenes de un proceso concreto. Posteriormente, las imágenes eran procesadas, accediendo a ciertas estructuras que se encontraban al analizar su contenido.

No es hasta los años 80, cuando junto con la creación de procesadores más rápidos y capaces, se consigue captar, procesar y reproducir imágenes de forma remota, dando lugar a la captura de imágenes y la reducción a sus características visuales básicas.

2.2. Técnicas de lectura 3D

Dentro de la visión artificial, encontramos múltiples formas de agrupar las técnicas de visión 3D, como, por ejemplo, atendiendo a la información obtenida o la forma de obtener dicha información. A la hora de dividir los tipos de técnicas utilizadas en la visión artificial me voy a basar en el principio de funcionamiento de cada sistema [1], además de indicar en cada caso si se trata de una técnica activa o pasiva. Cabe destacar que las técnicas activas son aquellas que

requieren de la emisión de señales para la obtención de los valores de interés, mientras que las pasivas no las emiten.

***Figura 2**: Clasificación de técnicas de VA según principios de funcionamiento, verde pasivo, rojo activo.*

*En esta técnica se utlizan sistemas tanto activos como pasivos

2.2.1. Luz estructurada

Los sistemas basados en luz estructurada disponen de tres elementos fundamentales, un patrón de luz conocido, un sistema de detección y un procesado de los datos obtenidos. La fuente de luz estructurada se encarga de iluminar el objeto de interés, el patrón de luz emitido se ve deformado por la superficie del objeto, siendo esta deformación captada por la cámara y procesada posteriormente.

Existen multitud de patrones de luz que varían en complejidad y coste, el caso más común es un patrón basado en una línea láser que recorre la globalidad del objeto para obtener sus características tridimensionales (**Figura 3**).

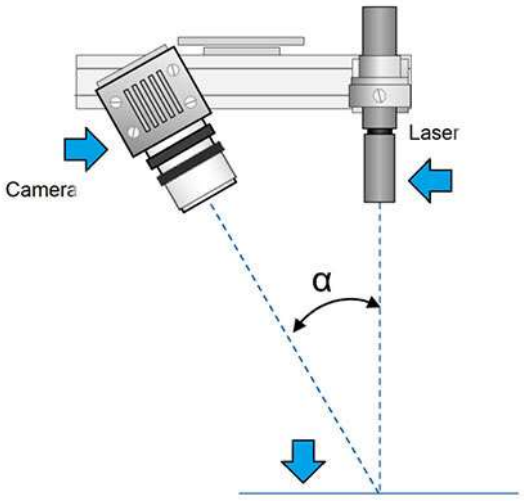


***Figura 3**: Sistema de visión creado mediante técnicas de luz estructurada*

Generalmente se trata de un método muy preciso, independiente de las condiciones de iluminación ambientales y tienen un bajo coste computacional, lo que permite desarrollar sistemas de bajo coste. Sin embargo, no disponen de gran flexibilidad, ya que, deben trabajar con unos rangos y precisiones concretas, requieren de un sistema mecánico para escanear la globalidad del objeto y se debe tener en cuenta el color del patrón de luz junto con el del objeto escaneado, si ambos son del mismo color, puede no distinguirse el patrón del objeto. Debido a sus características, estos sistemas trabajan muy bien bajo entornos controlados, siendo una de las razones por las que se han extendido ampliamente en la industria.

2.2.2. Estereovisión

Este conjunto de técnicas consiste en capturar una misma escena desde dos o más puntos de vista. Las diferentes capturas se pueden realizar con distintas cámaras sincronizadas (**Figura 4**), conociendo la distancia que las separa o con un solo sensor, que captura la escena desde diferentes posiciones. El procedimiento a seguir en este tipo de técnicas es el siguiente:

- Captura de la escena mediante diferentes puntos de vista. La localización de los distintos puntos de vista debe ser conocida o en su defecto calcularse en base a las imágenes tomadas.
- Corrección de distorsiones debidas a la lente. Se pretende obtener una representación fehaciente de la realidad y evitar disparidades de percepción.
- Alineación de las imágenes, no es imprescindible, simplifica el posterior procesamiento debido a la paridad en posición de los puntos característicos de la imagen.
- Se resuelve el llamado, problema de correspondencia, proceso mediante el cual se trata de encontrar los pares característicos de las imágenes. Para ello se utilizan funciones de coste que permiten comparar aquellos puntos con mayor influencia.
- Obtenidos los pares se forma un mapa de disparidad que representa la diferencia entre las coordenadas de las imágenes.
- Por último, si se conocen los parámetros, tanto intrínsecos como extrínsecos de las cámaras, se puede obtener la información 3D mediante el mapa de disparidad.

Como se observa, una de las principales ventajas de estos sistemas es la pasividad, no requiere emitir ningún tipo de señal y tienen un bajo coste económico, con un par de cámaras es más que suficiente para implementar el sistema de visión. El principal inconveniente, es la resolución del

problema de correspondencia en situaciones donde el objeto de interés tiene poca textura, ya que se dificulta la adquisición de puntos característicos sobre la imagen.

***Figura 4**: Bumblebee. Cámara estéreo paralela comercial*

2.2.3. Tiempo de vuelo

En sistemas basados en tiempo de vuelo, la variable de interés es el tiempo. Este transcurre desde que el pulso electromagnético es emitido hasta que vuelve a ser captado por el sensor, habiendo viajado a través de un medio conocido y siendo reflejado previamente en el objeto de interés. Conociendo la velocidad de la onda en el medio que atraviesa, la distancia se puede calcular en función del tiempo transcurrido. Dentro de estos sistemas las diferencias vienen determinadas por la tipología del sensor que recibe la señal reflejada:

- **Emisores de luz a determinadas frecuencias:** se emiten ondas electromagnéticas, generalmente infrarrojo, con unas longitudes de onda muy concretas, midiendo el desfase de estas se obtiene el tiempo transcurrido.
- **Obturadores automáticos:** emitiendo ondas a determinadas frecuencias y sincronizando la apertura y cierre de los obturadores con dichas frecuencias, la distancia al objeto de interés se obtiene mediante la cantidad de luz que se bloquea (obturador cerrado) y el alcance de la onda emitida.
- **Pulsos directos:** en este caso se emiten únicamente pulsos (láser o ultrasonido), recibiendo la cantidad y tiempo que tardan en volver al sensor.

Las principales ventajas de estos sistemas son el bajo tiempo de adquisición, un casi inexistente coste computacional, se obtiene directamente información 3D y la no dependencia de la textura de los objetos. Por el contrario, se caracterizan por tener una muy baja resolución, alto coste económico y dificultades de escaneo en superficies con un alto coeficiente espectral.

2.2.4. Interferometría

Las técnicas de interferometría se basan en el análisis de las señales reflejadas por una superficie, estas señales reflejadas son comparadas con patrones de referencia que permiten extraer la información 3D del objeto. Generalmente se usan señales electromagnéticas con más de una longitud de onda, permitiendo así el aumento de la resolución. Los principales inconvenientes son, su alto coste económico y que solo se puede aplicar en superficies prácticamente planas, lo que a su vez es una ventaja si se aplica para un control de calidad preciso, sobre este tipo de superficies.

2.2.5. Rayos x

Se trata de técnicas que consisten en la emisión de un tipo de radiación (rayos x) sobre una sección del objeto, obteniendo la información 3D en función de la radiación absorbida por el objeto, este puede ser reconstruido si se juntan los diferentes planos.

Estos sistemas permiten obtener información interna y externa de los objetos con una gran precisión. Por otro lado, resultan muy costosos económicamente, los objetos a escanear tienen dimensiones limitadas y la radiación empleada supone un alto riesgo sanitario para los seres humanos. Son utilizados principalmente en medicina y en la industria alimentaria.

2.2.6. Campos magnéticos

Como se viene comentando en los apartados anteriores, las técnicas basadas en campos magnéticos se centran en el estudio de la perturbación de los núcleos atómicos de los objetos a escanear. Los átomos del objeto se alinean con un campo magnético constante y más tarde se perturban con otro campo magnético adicional, siendo los resultados de esta perturbación los que aportan la información 3D del plano del objeto. Principalmente estas técnicas son utilizadas en medicina, aunque hay algunos casos en los que se aplica a la industria.

Estas técnicas presentan una alta precisión, en lo que a la representación tridimensional del objeto respecta, y permite analizar, al igual que en técnicas basadas en rayos x, la estructura interna del mismo. Por el contrario, resultan ser muy caros, lentos en la adquisición de imágenes y debido a la influencia de objetos ferromagnéticos, estos no deben situarse próximos al objeto de interés.

2.2.7. Acomodación de enfoque

Los métodos basados en acomodación de enfoque utilizan la profundidad de campo de la cámara para obtener la información 3D. Todo objeto enfocado sufre un desenfoque que es mayor a mayor apertura de diafragma y siempre es proporcional a la distancia entre el punto enfocado y el resto de los puntos no enfocados. Existen técnicas tanto activas como pasivas, donde el objeto puede analizarse en base a un patrón de luz o simplemente analizar la distorsión de su superficie.

Se trata de un sistema muy económico, pero presenta gran cantidad de desventajas como la baja resolución, una calibración previa de la cámara y un rango de profundidades muy limitadas, además, en el caso pasivo, se ve afectado por cambios de luz.

2.2.8. Conocimiento de la escena

Basándonos en una imagen 2D tomada de un objeto conocido y considerando la calibración efectuada en la cámara, es posible extraer información 3D muy precisa en los ejes x e y. En el caso del eje z (eje cámara-objeto) se pueden medir las distancias conociendo las medidas del objeto de interés y los parámetros de la cámara, pudiendo determinar de esta manera una relación entre la medida del objeto en la fotografía y su profundidad. Para crear mayor definición se suele combinar con técnicas de luz estructurada.

Estos procedimientos son baratos y consiguen unos resultados 3D bien definidos. Sin embargo, debe existir un patrón conocido del objeto analizado, que en muchos casos no es posible obtener.

2.2.9. Análisis de la textura

En objetos con texturas muy irregulares se puede considerar este factor para analizar la inclinación de la superficie de interés, para realizar este análisis se requiere de los parámetros que representan la superficie, siendo estos los ubicados en una imagen 2D. Estas metodologías son muy baratas y sencillas, pero cuando la superficie deja de ser irregular la precisión es muy pobre.

3. Factores a considerar

3.1. Factores exigibles

- Se requiere de un proceso de escaneo rápido y sencillo, sin necesidad de altos conocimientos técnicos.
- Precio máximo de la solución aportada 500 € por equipo.
- Sistema portátil con la posibilidad de realizar escaneos tridimensionales en color
- Errores relativos máximos admisibles, 5%.

3.2. Factores condicionantes

- Utilización de sistema de visión 3D de bajo coste, cámaras 3D *Intel REALSENSE DEPTH CAMERA D415* y *Microsoft Azure Kinect DK*.
- Situación sanitaria actual que impide la obtención de todas las tomas de datos que se hubiesen querido por parte del autor del proyecto. En concreto se ha obtenido una toma por cada estancia a escanear, fue lo que se pudo hacer durante el tiempo que se tuvieron las cámaras en posesión.
- Hardware utilizado para la reconstrucción del sistema, se trata de un portátil Asus F550C que tiene sus limitaciones a la hora de reconstruir los entornos 3D bien por el tiempo de procesamiento de las imágenes o bien por la incompatibilidad de algunos programas, usando versiones más livianas de los mismos.

4. Propuestas alternativas

A continuación, se presentan propuestas alternativas a la utilizada en el proyecto como podrían ser:

- Utilización de métodos de luz estructurada, obviamente son más precisos que el utilizado, pero a la vez más caros, por tanto, han sido desestimados. Sale del ámbito de aplicación del proyecto. Dentro de este conjunto encontramos modelos comerciales como el escáner 3D *Peel 3D Peel 2*
- También encontramos alternativas que combinan diferentes patrones de luz estructurada para aumentar la resolución final de la información 3D obtenida, siendo un ejemplo comercial los modelos *Shining EinScan H* y *HX*, que combinan técnicas de luz

estructurada y triangulación láser (diferentes patrones láser emitidos simultáneamente).

***Figura 5**: Escaners 3D comerciales. Peel 3D Peel 2 (izquierda) y Shining EinScan HX (derecha)*

El principal inconveniente de todos los dispositivos mostrados anteriormente es la dotación económica que hay que realizar para obtenerlos. Por el contrario, las dos cámaras utilizadas en el proyecto (*Azure Kinect DK y Intel Realsense Depth Camera D415*) no superan los 400 € en su precio de venta al público, cifra que se ve multiplicada 15 veces si se intenta comprar alguno de los modelos comentados o similares.

En cuanto a la precisión, como es de esperar no son tan precisas como los modelos comentados, sin embargo, dado el ámbito de aplicación, no importa tener una precisión milimétrica. Las estancias escaneadas son superficies de varios metros cuadrados donde no se requiere de tal exactitud.

5. Descripción del Hardware empleado

Para la implementación del sistema de visión finalmente se han empleado dos cámaras, basadas en técnicas de estereovisión, siendo posible reconstruir los sistemas escaneados para su posterior análisis. Además de las cámaras, se ha utilizado un ordenador personal, para el procesamiento de los datos obtenidos y la reconstrucción del entorno tridimensional.

5.1. Intel REALSENSE DEPTH CAMERA D415

La *Intel REALSENSE DEPTH CAMERA D4515* se trata de una cámara 3D dual multiplataforma. Que consta de distintas cámaras (DEPTH y RGB) que permiten la recreación 3d del conjunto grabado. En cuanto a las especificaciones generales de esta cámara encontramos:

- Rango de trabajo óptimo: entre 0.5 m y hasta 3 m*
- Dimensiones: 99 x 20 x 23 mm
- Objetivo de profundidad:
    - Ángulos de visión FOV (Field of View): 65 x 40 º
    - Resolución máxima de salida: 1820 x 720 p*
    - Velocidad de obturación: hasta 90 fps*
    - Mínima distancia de enfoque: 16 cm*
    - Precisión de la profundidad: menor al 2% (en 2 m)*

- Objetivo RBG:
    - Ángulo sensor FOV: 69 x 42º
    - Resolución máxima de salida: 1920 x 1080 p*
    - Velocidad de obturación: 30 fps*
    - Resolución del sensor: 12 MP

Consultar **[2] Intel RealSense D400 Series Product Family Datasheet para información más concreta*

***Figura 6**: Intel Realsense Depth Camera D415 [2]*

Todas estas indicaciones varían en función de las velocidades de obturación y de la resolución seleccionada al realizar la grabación. Por ello, basándonos en las grabaciones realizadas en el proyecto se establece en los próximos párrafos la configuración concreta utilizada para grabación con la *Intel REALSENSE DEPTH CAMERA D415*.

Como se observa en la hoja de características del producto este puede trabajar con distintas configuraciones en las que se pueden variar principalmente, velocidades de obturación, resolución de salida y en algunos casos la apertura de diafragma (profundidad de campo). En base a la configuración seleccionada, algunas propiedades del producto quedan desactivadas por la incompatibilidad de estas [2]. Destacar que la configuración aplicada influye en las dos cámaras, se requiere del mismo tipo y cantidad de fotogramas por segundo para que la información de la escena sea reconstruida correctamente.

En lo que respecta a la configuración definida para el escaneo de las estancias tenemos:

- Resolución de salida (módulo de profundidad): 640 x 480 p
- Velocidad de obturación (módulo de profundidad): 30 fps
- Resolución de salida (módulo de color): 640 x 480 p
- Velocidad de obturación (módulo de color): 30 fps

Dejando fijos los dos parámetros mencionados, la apertura de diafragma varia adaptando la luz que entra al sensor, para que este pueda tomar la información apropiadamente. Se observa que la configuración seleccionada no es la que más calidad e información aporta al sistema, esto es debido a que se ha seleccionado una configuración que permite obtener una buena calidad en el escaneo, pero no compromete la cantidad de datos obtenidos (tamaño del archivo). De nada sirve obtener un escaneo con mucha calidad e información, si nuestros sistemas de procesamiento no son capaces de soportar la carga computacional que conllevan estos volúmenes de datos.

En cuanto a los ángulos del campo de visión de la cámara, cabe destacar, que estos serán menores a los presentados anteriormente. Esto es debido a la ubicación de ambos objetivos en el cuerpo de la cámara (ver **Figura 7**). Independientemente de la posición del plano de interés, el campo de visión efectivo se ve reducido al ver el mismo plano desde dos puntos diferentes.

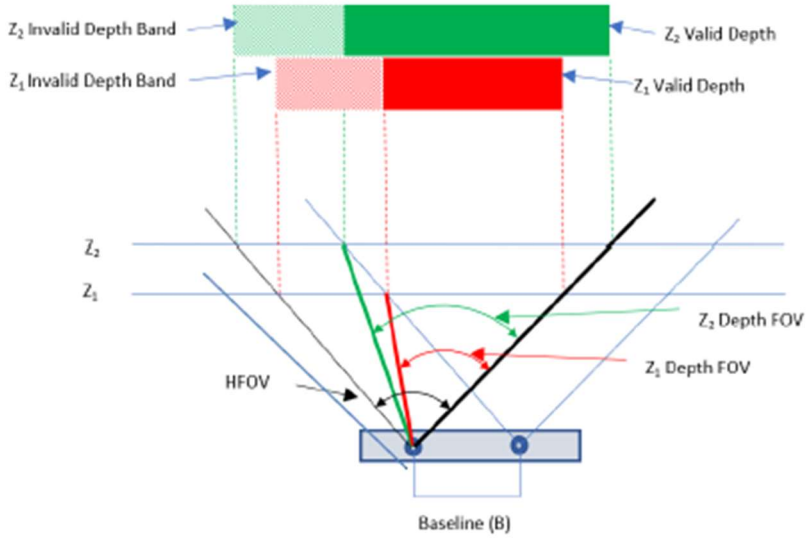


***Figura 7**: Campos de visión de la Intel Realsense D415 [2]*

Destacar la posibilidad de cambiar la calibración de la cámara, aunque esta viene calibrada de fábrica, estos parámetros se pueden modificar mediante un patrón de calibración (patrón de ajedrez) o una superficie plana paralela a la cámara, indicando la distancia entre la cámara y la superficie.

5.2. Microsoft Azure Kinetic DK

La cámara 3D *Microsoft Azure Kinetic DK,* al igual que el modelo presentado anteriormente, consta de dos cámaras, DEPHT y RGB, que permiten, de una forma similar a la anterior, extraer información 3D de la escena a escaneada. Respeto a las especificaciones de la cámara destacan:

- Rango de trabajo óptimo: entre 0.5 m y hasta 5.46 m*
- Dimensiones: 103 x 39 x 126 mm
- Objetivo de profundidad:
    - Ángulos de visión FOI (Field of Interest): 75 x 65 °*
    - Resolución máxima de salida: 640 x 576 p*
    - Velocidad de obturación: hasta 30 fps*
    - Mínima distancia de enfoque: 16 cm*
    - Resolución del sensor: 1 MP
- Objetivo RBG:
    - Ángulo sensor FOV: 90 x 59 °*
    - Resolución máxima de salida: 38840 x 2160 p*
    - Velocidad de obturación: 30 fps*
    - Resolución del sensor: 12 MP

Consultar **[3] Azure Kinect DK Documentation para información más concreta*

***Figura 8**: Microsoft Azure Kinect DK [3]*

Las características comentadas anteriormente dependen del modo y configuración usada ya que, por ejemplo, existen incompatibilidades entre altas resoluciones y velocidades de obturación entre otras.

De la misma manera que en la *Intel REALSENSE DEPTH CAMERA D415* la configuración establecida permite elegir la resolución y la velocidad de obturación quedando libres valores como sensibilidades o aperturas de diafragma de la cámara. Comentar, que no todas las resoluciones y velocidades son compatibles [3] lo que en algunos casos limita la gama de posibilidades de la reconstrucción tridimensional.

En lo que respecta a la configuración definida para el escaneo de las estancias tenemos:

- Resolución de salida: 1920 x 1080 p
- Velocidad de obturación: 30 fps

Si se analiza la configuración seleccionada no es la más exigente, nuevamente y basándose en el mismo argumento comentado en el apartado 5.1, esta configuración presenta una buena relación entre calidad del escaneo realizado y la cantidad de datos obtenidos de la escena.

En cuanto a los ángulos del campo de visión de la cámara se ven modificados en función del modo elegido, variando entre 75 x 65 ° y 120 x 120 ° en el caso de la cámara de profundidad y 90 x 59 ° y 90 x 74.3 ° en el caso de la cámara RGB. Debido a la variación el campo de visión de las distintas cámaras y su posición relativa el campo de interés (FOV) se ve reducido como se observa en la Figura 9.

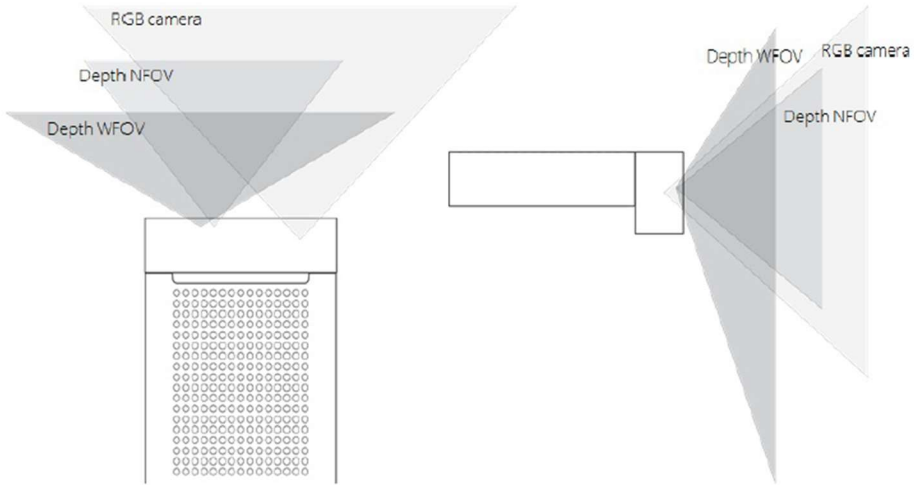


***Figura 9**: Campos de visión horizontales (izquierda) y verticales (derecha) de la Azure Kinect DK [3]*

Al igual que la *Realsense D415*, la *Azure Kinect DK* también permite cambiar la calibración de fábrica de la cámara mediante los patrones comentados en el apartado anterior.

5.3. Asus F550C

Para la gestión y análisis de los datos obtenidos con las dos cámaras expuestas anteriormente se requiere de un dispositivo capaz de tratar tal cantidad de información de una manera rápida y eficaz. Para ello, se ha dispuesto del ordenador personal del alumno, un *Asus F550C*. Este dispositivo dispone de las siguientes características:

- Procesador: Intel Core I7 3537U
- Tarjeta gráfica: NVIDIA GEFORCE 720M
- Memoria RAM: 4 GB
- Disco duro: SanDisk SDD 1TB
- Sistema Operativo: Windows 10 (64 bits)

Como se verá más adelante, en el desarrollo del proyecto, el ordenador utilizado ha sido una fuente de problemas en algunos casos. Para evitar estos problemas se recomienda utilizar un ordenador un poco más potente. La memoria RAM se queda un tanto corta a la hora de procesar las imágenes con los códigos de Python con lo que se recomienda que se aumente a un modelo con 8 GB de RAM, además, para utilizar la versión extendida del programa RTAB-Map (ver **7.2.1 Instalación RTAB-Map**) se requiere de tarjetas gráficas dedicadas, por lo tanto, una tarjeta gráfica con estas características también sería conveniente.

Hay que destacar, que estas mejoras no afectan a la calidad final del proyecto, simplemente facilitan la fluidez y disminuyen los tiempos de procesamiento. Quedando excluidas de los factores condicionantes del proyecto.

6. Escaneo de estancias

Una de las partes más importantes en cualquier proceso de reconstrucción tridimensional es la adquisición de datos del entorno escaneado, si partimos de una información tomada bajo unas condiciones óptimas la reconstrucción puede llegar a serlo. Para escanear adecuadamente nuestros objetos de interés se deben tener en cuenta 4 aspectos fundamentales, la iluminación, la estabilidad, el movimiento y los objetos no deseados.

Una correcta iluminación es vital para obtener correctamente los colores y contrastes de los objetos, quedando de esta manera, definidos por sus límites espaciales y como consecuencia su integración con el entorno. La grabación debe ser estable y suave, evitar los movimientos bruscos para reducir perturbaciones indeseadas que entorpezcan la posterior reconstrucción de la escena.

La escena escaneada debe permanecer en reposo durante el escaneo ya que si la escena contiene movimiento este se representará como una nebulosa de puntos que dejará inservible la reconstrucción obtenida. Por último, evitar en la medida de lo posible objetos reflectantes o transparentes como puedan ser espejos, ventanas, etc. Debido a sus características, son muy difíciles de representar ya que plasman sobre su superficie otros objetos, además existen técnicas específicas para escanear este tipo de escenas (ver apartado **2.2 Técnicas de lectura 3D**).

7. Sistemas de reconstrucción utilizados

Para la reconstrucción de las escenas tridimensionales escaneadas se han utilizado dos métodos de ensamblaje tridimensional distintos. El primero, basado en una librería de *Python* llamada *Open3D* y el segundo mediante programa de reconstrucción en tiempo real *RTAB-Map (Real-Time Appareance-Based Mapping)*.

7.1. Librería Open3D Python

7.1.1. Introducción a la librería

Open 3D es una librería de código abierto que ofrece un software de soporte para gestionar información 3D. La interfaz de Open3D expone estructuras de datos y algoritmos basados en *Python* y *C++* [4]. El código desarrollado en la librería facilita la paralelización de procesos, dejando unos códigos claros y sencillos. Además de ser una librería multiplataforma, dispone de una extensa comunidad de usuarios, lo que proporciona una herramienta de diferenciación notable.

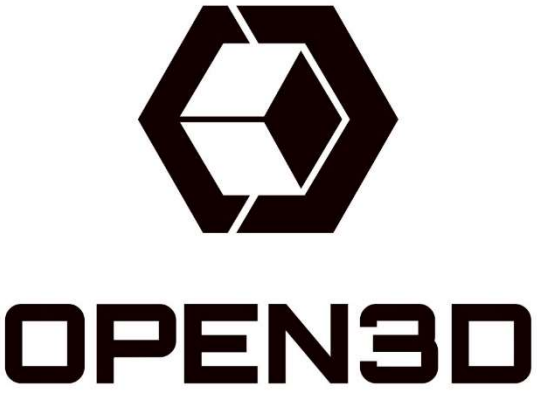


***Figura 10**: Logotipo de la librería Open3D*

7.1.2. Instalación Python y librería

Para la utilización de *Python* en una máquina con Windows 10 se ha optado por utilizar *Pycharm 2020.2.5*, añadiendo una distribución *Anaconda*. Para ejecutar los códigos de *Python* se requiere configurar en *PyCharm* un intérprete de *Python*, basado en la distribución añadida.

Para ello, se abre *Pycharm* y accediendo a **Archivo** → **Configuración** → **Interprete de Python** se pulsa en **Añadir**. En la pestaña emergente se selecciona el intérprete de Conda, definiendo a continuación la versión de *Python* a utilizar y la ubicación del entorno virtual (ver **Figura 11**). Destacar, que un mismo intérprete se puede usar para varios proyectos, aunque mi recomendación personal es trabajar con un intérprete por proyecto.

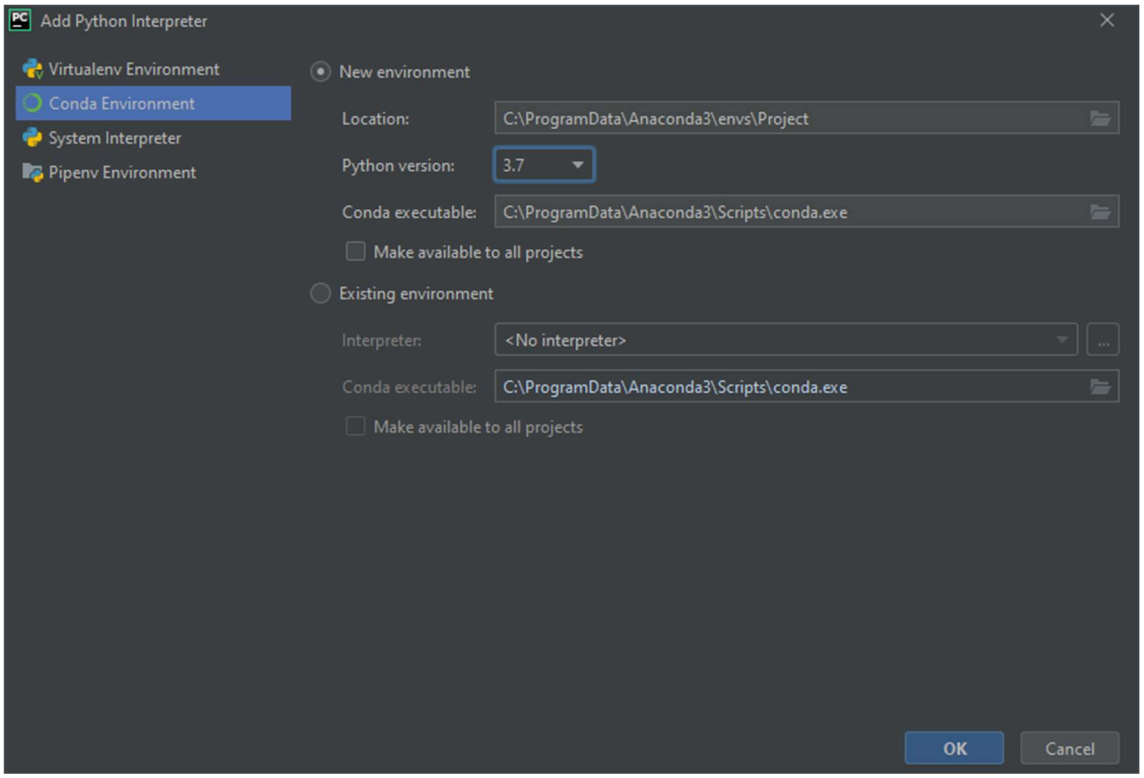


***Figura 11**: Configuración del entorno de Conda creado para el proyecto*

Una vez creado el entorno virtual, es el momento de añadir todas aquellas librerías necesarias para que *Open3D* funcione correctamente. Para ello sobre el Terminal de *PyCharm* se ejecutan las siguientes líneas de código:

- *conda install numpy*
- *conda install -c conda-forge opencv*
- *conda install -c conda-forge librerealsense*
- *conda install -c open3d-admin -c conda-forge open3d*

Estas líneas instalan las librerías Numpy y OpenCV (gestión y manipulación de datos en tiempo real), el wrapper (manejador) del sensor Intel RealSense SDK 2.0 [5] (ubicado en la Intel Realsense D415) y por último la librería Open3D, indispensable para la gestión de la información tridimensional.

Una vez realizado el proceso de instalación se puede comprobar, que este se ha realizado satisfactoriamente importando las diferentes librerías mediante:

- *python -c "import **nombrelibreria** as **nombredeimportacion**"*

```
(TFG_Project_II) C:\Users\marcr\Desktop\Project>python -c "import open3d as o3d"

(TFG_Project_II) C:\Users\marcr\Desktop\Project>
```

***Figura 12**: Comprobación de intalación correcta de la librería Open 3D*

7.1.3. Extracción de imágenes RGB y DEPTH

Como se comenta en el alcance del proyecto (ver **Figura 1**), para obtener la representación 3D de la escena se requiere de las imágenes de color y profundidad de cada uno de los instantes del escaneo.

El proceso de extracción de las imágenes varía en función de la cámara utilizada ya que de la *Intel Realsense D415* se obtiene un archivo .bag, mientras que en la *Azure Kinect DK* se obtiene un archivo *.mkv.*

7.1.3.1. Intel Realsense D415

Para realizar el proceso de extracción a partir del archivo tipo bag proporcionado por la cámara, se ejecuta el archivo *cargar_bag_alignment.py*. Para la correcta ejecución del archivo se debe indicar en las líneas 8, 31 y 33, el nombre del archivo.bag, la resolución y los fotogramas por segundo de la cámara de profundidad y la resolución y los fotogramas por segundo de la cámara de color respectivamente. Tener en consideración que el archivo.bag debe encontrarse en la misma dirección que el archivo ejecutado.

El código, presentado en Anexo 3, analiza y alinea cada uno de los fotogramas analizados guardando los respectivos archivos colorXXXX.jpg y depthXXXX.png, además de mostrar por pantalla una ventana con la secuencia grabada.

Hay que destacar la importancia de la correcta alineación de los fotogramas. Al analizar la escena desde dos puntos de vista, los campos de visión varían, quedando un primer campo de visión común, visible a ambas cámaras, y un segundo campo de visión perdido o muerto, visible por solo una de las dos cámaras.

Mediante el proceso de alineación se asegura que los límites del campo de visión son los mismos en ambos fotogramas, quedando por tanto descrito el campo de visión común.

7.1.3.2. Microsoft Azure Kinect DK

Para extraer las imágenes del archivo mkv obtenido de la *Microsoft Azure Kinect DK* se debe instalar el SDK (software development kit) de la cámara. Una vez instalada se añaden las librerías para que puedan ser utilizadas por la librería Open3d [6] y indicando el nombre del archivo se obtiene la descomposición de imágenes de color y profundidad.

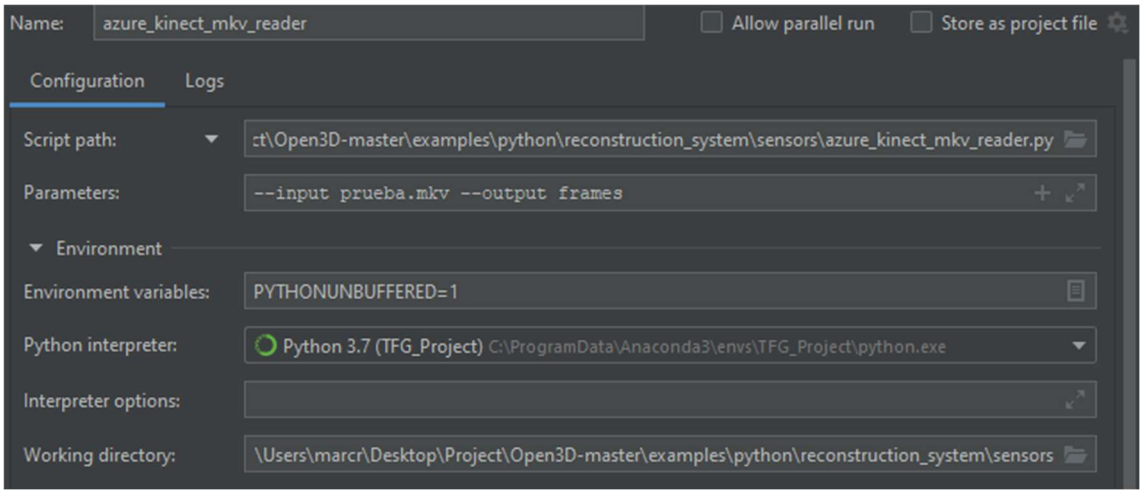


***Figura 13**: Configuración de Pycharm para la extracción de las imágenes*

7.1.4. Reconstrucción del modelo 3D

Para la reconstrucción del modelo 3D se utilizan una serie de archivos de *Python* incluidos en las referencias ([7] Open3D – Reconstruction system), estos se pueden encontrar en los archivos de *GitHub* de la propia librería (*Open3D/examples/Python/reconstruction_system*) [8].

La reconstrucción se divide en cuatro partes fundamentales, la creación, enlace, ajuste e integración de la escena. Cada una de estas fases es independiente de la anterior, pudiendo ser ejecutadas tantas veces como sea necesario de forma individual, antes de pasar al siguiente proceso.

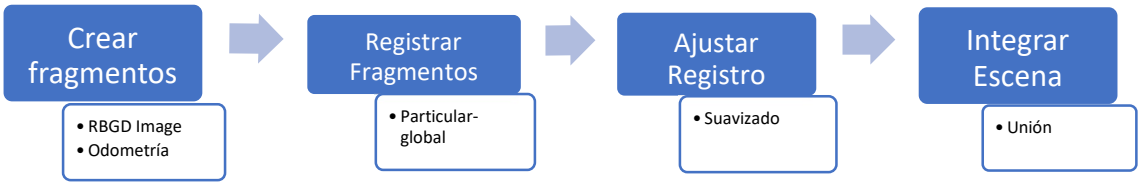


***Figura 14**: Flujograma de las partes de la reconstrucción 3D*

Respecto a la configuración de la reconstrucción, dentro de la carpeta **config** se encuentra un archivo tipo JSON llamado tutorial. En **tutorial.json** se especifica:

- La ubicación relativa de las imágenes, en la dirección establecida deben existir dos carpetas llamadas **depht** y **image** con sus respectivas imágenes dentro.
- La profundidad máxima de la escena grabada
- El tamaño de los puntos tridimensionales ubicados en la reconstrucción.
- La diferencia máxima entre dos profundidades, si la diferencia de profundidad es mayor se consideran dos objetos independientes
- Fiabilidad del lazo cerrado creado por la odometría y el escaneo de la escena.

```
{
    "name": "Open3D reconstruction tutorial http://open3d.org/docs/release
    "path_dataset": "Pictures_banyo/",
    "path_intrinsic": "",
    "max_depth": 3.0,
    "voxel_size": 0.03,
    "max_depth_diff": 0.07,
    "preference_loop_closure_odometry": 0.1,
    "preference_loop_closure_registration": 5.0,
    "tsdf_cubic_size": 3.0,
    "icp_method": "color",
    "global_registration": "ransac",
    "python_multi_threading": true
}
```

***Figura 15**: Configuración guardada en tutorial.json*

7.1.4.1. Creación de los fragmentos

El primer paso para reconstruir la escena es crear pequeños fragmentos de la secuencia RGBD obtenida.

Para ello el código lee el primer par de imágenes RGBD y compara esta imagen con el resto de las imágenes obtenidas mediante el cálculo de la odometría (posición relativa de un punto respecto a otro). Si este proceso se repite para cada una de las imágenes se puede obtener una

alineación basta de las imágenes, esta permite la creación de fragmentos independientes de elementos clave de la escena.

Para acelerar el proceso de creación de fragmentos se utilizan algoritmos de optimización de gráficos, estos permiten encontrar las posiciones de encaje entre imágenes, tomando como validas aquellas posiciones en las que encajan mayor cantidad de puntos.

***Figura 16**: Fragmento 0002 de la escena escaneada con la Intel Realsense D415*

En este caso se establecen como parámetros de entrada los pares RGBD, estos son analizados, como se explica anteriormente y al acabar el proceso se obtiene un fragmento, en forma de nube de puntos, por cada cien pares RGBD.

7.1.4.2. Enlace de los fragmentos

Una vez creados los fragmentos, cada uno de ellos es independiente del resto, por tanto, deben ser transportados a un espacio tridimensional global donde todos ellos se unan y creen una representación de la escena escaneada.

Para transportar los fragmentos a un espacio global estos se alinean utilizando los fragmentos vecinos y agregando la información obtenida en la odometría RGBD. En caso de no encontrar fragmentos vecinos (con factores comunes) se realiza un registro global de la escena donde iterando sobre puntos definidos de los fragmentos se intenta hallar aquellas zonas que no son definidas en ningún fragmento, recordar que estas zonas muchas veces carecen de interés ya que no disponen de información relevante o puntos característicos (paredes lisas, superficies homogéneas).

Como valores de entrada obtenemos la salida del proceso anterior, las nubes de puntos. Sobre estas se calculan las odometrías, guardado las relaciones entre fragmentos, más tarde serán utilizadas para crear la representación global de la escena.

7.1.4.3. Ajuste de los fragmentos

Una vez obtenido un primer *boceto* de la representación tridimensional es el momento de pulir aquellos puntos conflictivos, donde el enlace entre fragmentos puede haber fallado. Para ello se tiene en consideración los parámetros especificados en el archivo **tutorial.json.**

Basándose en el tamaño de los puntos tridimensionales, la distancia máxima entre profundidades y la profundidad máxima escaneada, los objetos son agrupados y definidos con mayor claridad. Además, a partir del análisis del color de los objetos se suavizan los límites de estos quedando debidamente delimitados espacialmente [7].

7.1.4.4. Integración de la escena

El último paso para completar la reconstrucción es, la integración de la escena. En este caso se lee la información obtenida de las alineaciones realizadas entre pares de imágenes y fragmentos y habiendo sido transportados al espacio global de representación se permite reconstruir la escena.

Para reconstruir la escena tal y como se presenta en la **Figura 17**, se leen los datos obtenidos en el enlace y ajuste de los fragmentos, uniendo todos aquellos fragmentos que han sido analizados previamente. Obteniendo una nube de puntos que representa la escena global.

***Figura 17**: Reconstrucción 3D de la escena (primera reconstrucción obtenida)*

Para la ejecución de todos estos apartados se debe ejecutar el archivo **run_system.py** ubicado en *Open3D/examples/Python/reconstruction_system*, especificando la ruta relativa del archivo de configuración y los pasos que se desean realizar, en la **Figura 18** vemos que se han incluido los cuatro pasos en una ejecución, estos también se pueden ejecutar individualmente.

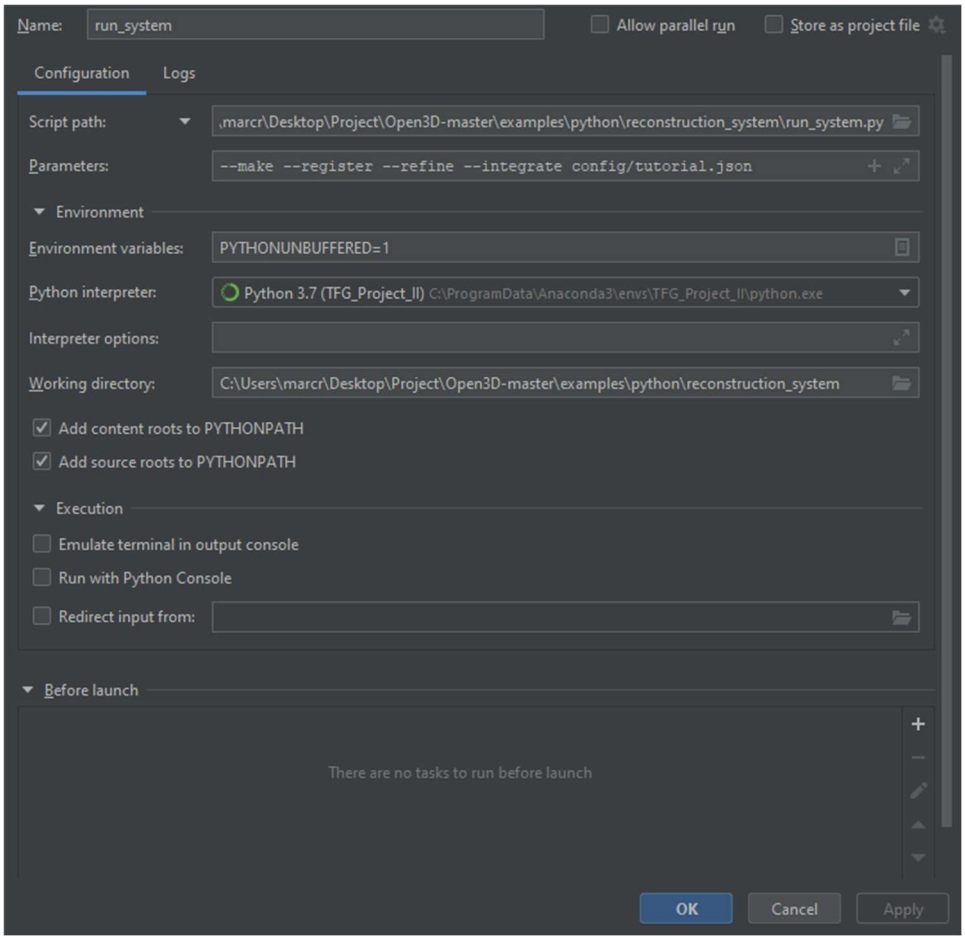


***Figura 18**: Parámetros de ejecución de run.py*

7.1.5. Análisis de los modelos obtenidos

7.1.5.1. Intel Realsense D415

En este análisis me voy a basar en cómo afectan los parámetros de la configuración sobre las reconstrucciones obtenidas. Para ello presento la **Tabla 1** que muestra una relación entre las reconstrucciones obtenidas, **Figuras 17, 19 y 20**, y las configuraciones establecidas. Destacar la primera reconstrucción, que se ha realizado con la configuración por defecto introducida en la documentación de *Open3D*.

Figuras	Profundidad Máxima	Tamaño puntos	Máxima diferencia de profundidad
Figura 17	3.0	0.05	0.07
Figura 19	3.0	0.03	0.09
Figura 20	3.0	0.02	0.05

***Tabla 1**: Relación entre reconstrucciones y configuraciones*

Como se puede observar en la primera reconstrucción **Figura 17**, la estancia se ve bien definida, aunque se observa cierto ruido alrededor de las paredes y ciertas superficies se ven duplicadas.

Si pasamos al siguiente caso, **Figura 19**, se disminuye el tamaño de los puntos que generan la representación y además se aumenta la máxima diferencia de profundidad. A priori estos

cambios deberían aumentar la resolución (menor volumen tridimensional de los puntos) y a su vez descartar menor cantidad de información (mayor diferencia de profundidad).

Por el contrario, al aumentar la diferencia máxima de profundidad entre puntos se dan como validos puntos que están fuera de las superficies de interés, lo que finalmente lleva a la superposición de fragmentos de forma incorrecta y, por tanto, a una reconstrucción no valida.

***Figura 19**: Reconstrucción 3D aumentando la máxima diferencia en profundidad (Intel Realsense D415)*

En la tercera reconstrucción, se trata de aumentar al máximo la resolución disminuyendo tanto el tamaño de los puntos, como la separación máxima entre ellos. Se puede comprobar que la mejora entre la primera reconstrucción es considerable, se elimina gran cantidad del ruido de además de evitar duplicaciones de superficies no deseadas.

Figura 20: Reconstrucción 3D disminuyendo el tamaño y la máxima diferencia de profundidad entre puntos

Como se puede observar en la **Figura 20**, la reconstrucción continúa teniendo algo de ruido, este no es relevante para la visualización de la escena, pero puede causar problemas en futuros usos de la reconstrucción. Para eliminar estos puntos erráticos se pueden usar principalmente dos tipos de filtros, un filtro estadístico, que elimina aquellos puntos que se encuentran en los extremos de la distribución aplicada y filtros de eliminación por radio.

En el primer tipo de filtros se establece un número de puntos y una desviación estándar respecto a la distancia entre los puntos, quedando fuera aquellos que se encuentran más alejados. En el segundo caso, se define un radio y un número de puntos, todo grupo de puntos que en una esfera de dicho radio no contenga en número mínimo de puntos queda eliminado.

En este caso y debido a la existencia de un pasamanos situado junto al WC se ha decidido utilizar un filtro estadístico (ver Anexo 1: cargar_modelo_3d.py). Tras el filtrado se obtiene la reconstrucción de la **Figura 21**.

***Figura 21**: Reconstrucción aplicando un filtro estadístico (Intel Realsense D415)*

7.1.5.2. Microsoft Azure Kinect DK

Con esta cámara se han escaneado dos estancias distintas, la primera, que es la misma que en el caso que la *Intel Realsense D415* y la segunda se trata de una habitación sin luz natural en la que encontramos diferentes superficies reflectantes, como pantallas de ordenadores, o el propio suelo, cuando refleja los focos de luz.

En lo que respecta a los parámetros introducidos en la configuración de la reconstrucción, se ha optado por utilizar las mismas configuraciones que en el caso anterior, para igualar las condiciones de las reconstrucciones. Como era de esperar, la mejor reconstrucción se ha conseguido con la presentada en la última fila de la **Tabla 1**.

Observando la **Figura 22**, se comprueba que aquellas superficies reflectantes como la pila del banyo o el pasamanos de minusválidos, se reconstruyen de una forma muy errática y sin definir el objeto de forma adecuada, esto sucede de la misma manera si se analizan las reconstrucciones de la *Realsense D415,* ver **Figura 21**. Para comprobar que no se trata de un caso concreto se pretende reconstruir una segunda habitación, como se ha comentado anteriormente en esta existen elementos reflectantes.

***Figura 22**: Reconstrucción 3D obtenida (Azure Kinect DK)*

Siguiendo el mismo proceso explicado para la primera reconstrucción se reconstruye la siguiente escena, **Figuras 23, 24 y 25**.

***Figura 23**: Reconstrucción 3D de la segunda escena (Azure Kinect DK) (1/3)*

***Figura 24**: Reconstrucción 3D de la segunda escena (Azure Kinect DK) (2/3)*

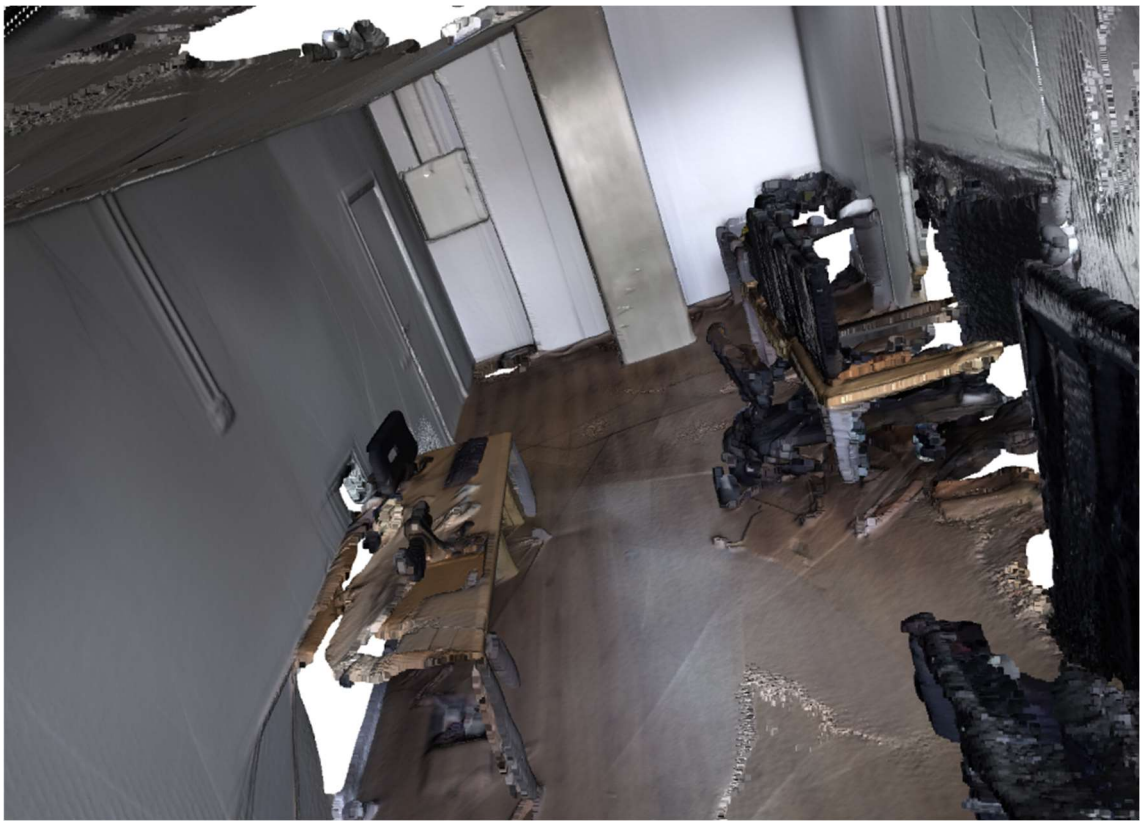


***Figura 25**: Reconstrucción 3D de la segunda escena (Azure kinect DK) (3/3)*

Cabe destacar que los elementos reflectantes se ven vacíos en las zonas brillantes y llenos en las zonas opacas, por ejemplo, en la **Figura 26**, se ve una pantalla de ordenador donde el marco se ha reconstruido correctamente mientras que su interior no queda definido. También se observan defectos en la reconstrucción del suelo en aquellos puntos donde se encuentra el reflejo de los puntos de iluminación, **Figura 25**.

***Figura 26**: Pantalla de ordenador presente en la escena reconstruida*

7.2. RTAB-Map

RTAB-Map es una herramienta de reconstrucción 3D en tiempo real basada en gráficos RGBD, que utiliza un bucle cerrado basado en el aspecto incremental de las imágenes. El detector del bucle cerrado determina, si la imagen proviene de una localización anterior o si establece una nueva localización [9]. Si la hipótesis de nueva localización es aceptada, se añade al gráfico de reconstrucción, utilizando un optimizador de gráficos que minimiza los errores de la reconstrucción.

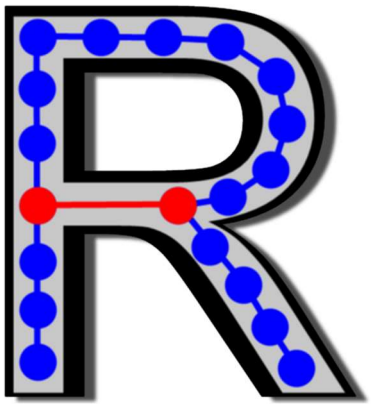


***Figura 27**: Logotipo del progamario RTAB-Map*

Para asegurar las limitaciones del tiempo real, el programa, dispone de un gestor de memoria que se estructura al igual que la memoria humana, memoria de trabajo (Working Memory, WM), corto plazo (Short-term memory, STM), largo plazo (Long-term Memory, LTM) y sensorial (Sensory Memory, SM). De esta forma las localizaciones visitadas recientemente se mantendrán en la memoria de trabajo, mientras que el resto pasarán a la memoria de largo plazo.

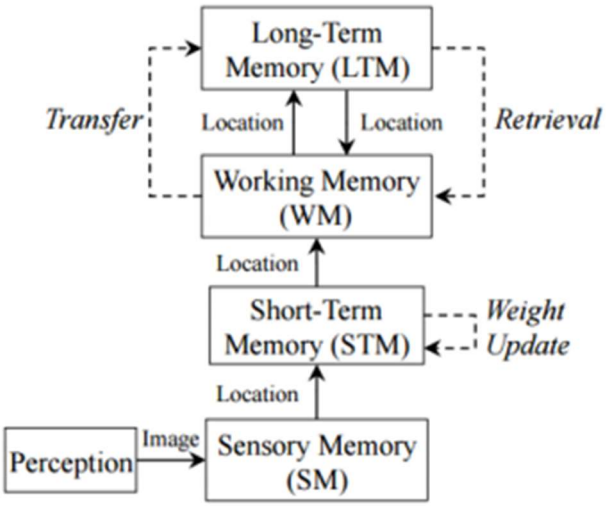


***Figura 28**: Tratamiento de memoria usado por RTAB-Map*

La **Figura 28** presenta el esquema del tratamiento de memoria utilizado por RTAB-Map en su técnica de reconstrucción. El módulo de percepción se encarga de obtener las imágenes, estas son enviadas a la SM, la cual procesa las imágenes y extrae aquella información útil para encontrar un lazo de realimentación, bucle cerrado. En base a la información extraída, se crea una nueva localización que es pasada a la STM, si la localización es similar a las almacenadas en esta memoria, se añade a la localización almacenada y se aumenta el peso de dicha localización.

La memoria de trabajo es la encargada de detectar los bucles cerrados entre ambas localizaciones espaciales. Debido a la gestión de la memoria, en la STM se almacenan únicamente aquellas localizaciones que tienen gran peso (más propensas a crear bucles cerrado), sin embargo, en la LTM se almacenan aquellas localizaciones con menor peso. La STM tiene como función encontrar localizaciones entre imágenes consecutivas, mientras que la memoria a largo plazo almacena la relación entre las localizaciones de menor peso, aquellas que ya han sido escaneadas.

Cabe destacar que la memoria LTM no se utiliza para detectar los bucles cerrados, esta es solo utilizada si la WM no es capaz de encontrar relación alguna entre localizaciones.

7.2.1. Algoritmo de RTAB-Map

En la **Figura 29** se puede observar en forma de seudocódigo el algoritmo de funcionamiento de RTAB-Map.

Algorithm 1 RTAB-Map

1: $time \leftarrow$ TIMENOW() ▷ TIMENOW() returns current time
2: $I_t \leftarrow$ acquired image
3: $L_t \leftarrow$ LOCATIONCREATION(I_t)
4: **if** z_t (of L_t) is a bad signature (using T_{bad}) **then**
5: Delete L_t
6: **else**
7: Insert L_t into STM, adding a neighbor link with L_{t-1}
8: Weight Update of L_t in STM (using $T_{\text{similarity}}$)
9: **if** STM's size reached its limit (T_{STM}) **then**
10: Move oldest location of STM to WM
11: **end if**
12: $p(S_t|L^t) \leftarrow$Bayesian Filter Update in WM with L_t
13: Loop Closure Hypothesis Selection ($S_t = i$)
14: **if** $S_t = i$ is accepted (using T_{loop}) **then**
15: Add loop closure link between L_t and L_i
16: **end if**
17: Join $trash$'s thread ▷ Thread started in TRANSFER()
18: RETRIEVAL(L_i) ▷ LTM $\rightarrow$ WM
19: $pTime \leftarrow$ TIMENOW() $- time$ ▷ Processing time
20: **if** $pTime > T_{\text{time}}$ **then**
21: TRANSFER() ▷ WM $\rightarrow$ LTM
22: **end if**
23: **end if**

***Figura 29**: Seudocódigo explicativo del algoritmo de funcionamiento del RTAB-Map*

Inicialmente se adquiere la imagen y se crea una localización para la misma. Las características de la imagen son representadas por un conjunto de palabras visuales, en el caso de RTAB-Map, el vocabulario visual ha sido entrenado previamente, disminuyendo de esta manera los tiempos de ejecución. Posteriormente, la imagen se transforma a coordenadas y se le aplica un escalado espacial (Scale-Space), esta técnica consiste en reducir el ancho de banda (Piramidal Gaussiana y Laplaciana) [10] de la imagen consiguiendo la uniformidad en los puntos de interés.

Una vez creada la localización, se comprueba si se han extraído suficientes características de la imagen, si el número es menor al valor medio de características extraídas por imagen, se descarta. Sin embargo, si cumple con la condición se introduce en la memoria a corto plazo y se aplicará la llamada, optimización de pesos.

En el momento en que el número de localizaciones almacenadas en la STM llega a su máximo, la última localización almacenada se traslada a la WM. Dentro de la WM la localización se calcula la probabilidad de que se detecte un bucle cerrado mediante un filtro de Bayes. Si el filtro de Bayes devuelve una alta probabilidad entre la localización analizada y alguna almacenada en la WM se establece una nueva relación de bucle cerrado.

Cuando el tiempo de asociación de nuevas localizaciones supera los umbrales establecidos, se entiende que existe poca probabilidad de crear un bucle cerrado entre las localizaciones almacenadas en la WM y las obtenidas de la STM. En este caso, se produce un proceso de transferencia donde las localizaciones con menor probabilidad de ser utilizadas pasan a la LTM. De la misma forma se comprueba si existen localizaciones vecinas en la LTM y estas se recuperan por parte de la WM quedando como localizaciones con posibilidades de establecer una relación de bucle cerrado. El número de localizaciones transferidas entre las distintas memorias depende del número de localizaciones que se añadan desde la STM a la WM.

7.2.2. Instalación RTAB-Map

Para la instalación de este programa se dispone de dos opciones, descargar la versión RTAB-Map 0.20.8 Cunda 11.1 o la versión RTAB-Map 0.20.8 [11]. Si se consultan los requisitos mínimos para el correcto funcionamiento del programa, en la primera versión se requiere de una tarjeta gráfica dedicada, requisito que no se cumple en la máquina de trabajo (Asus F550C), por esta razón se decide descargar e instalar la versión RTAB-Map 0.20.8.

El proceso de instalación no tiene mayor complicación, se descarga un ejecutable que es el encargado de instalar y dejar el RTAB-Map operativo.

7.2.3. Parametrización de la reconstrucción

Para realizar las reconstrucciones se utilizan las imágenes RGB y DEPTH, extraídas de los ficheros de video. Para ello se especifican las carpetas donde se encuentran estas imágenes, además, también se debe especificar la configuración intrínseca de la cámara, la cual permite la posterior reconstrucción tridimensional.

Para introducir estos valores en RTAB-Map ir a *Window/Preferences* y especificar las dos carpetas de imágenes y la ubicación del archivo de calibración tipo YALM.

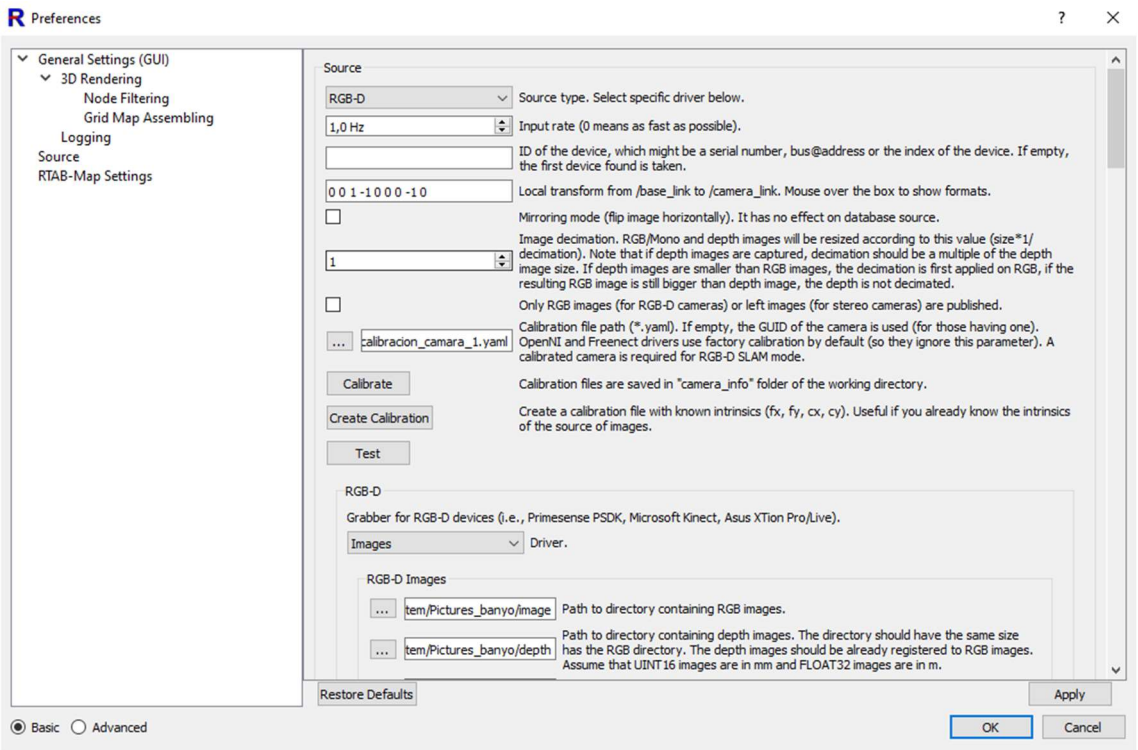


***Figura 30**: Ventana con las preferencias establecidas en para la reconstrucción 3D*

7.2.4. Análisis de los modelos obtenidos

Una vez especificadas las diferentes direcciones se pulsa el botón de inicio. Acto seguido el programa lee los pares de imágenes definiendo un código de color para el estado de cada par (ver **Figura 31**), siendo:

- Azul: característica encontrada en imágenes anteriores. Se aumenta el peso de dicha localización.

- Rojo: característica encontrada en el diccionario entrenado previamente. No existe localización para estas características.
- Verde: característica nueva. Nueva localización guardada.
- Amarillo: característica nueva y que además aparece más de una vez en la imagen.

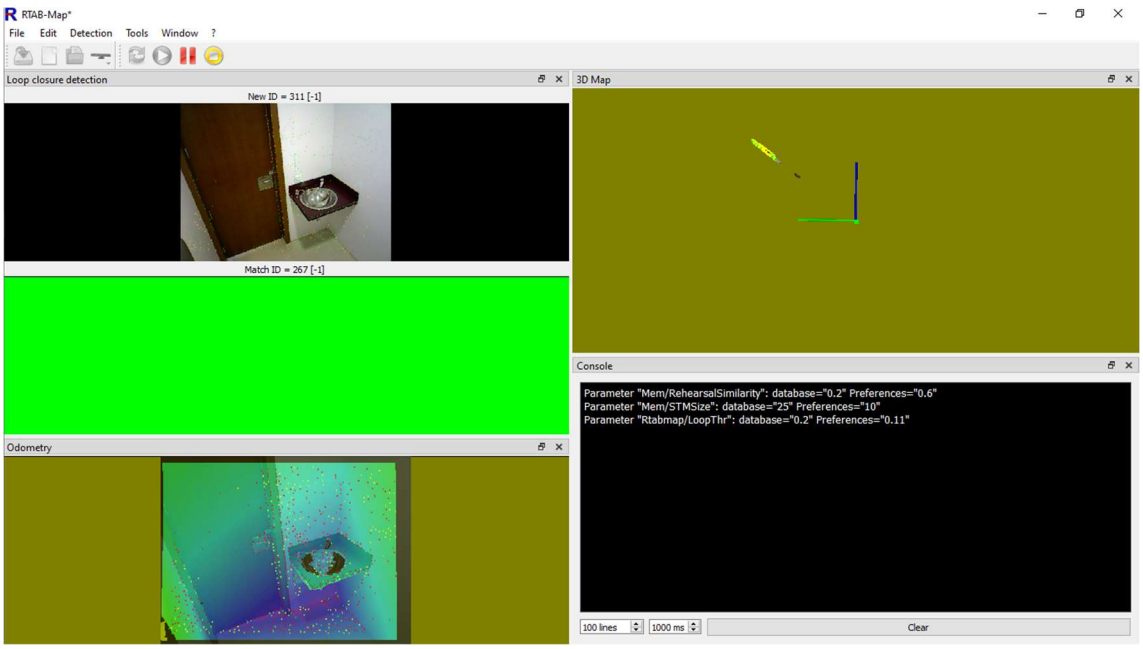


Figura 31: RTAB-Map leyendo y clasificando los pares de imágenes

7.2.4.1. Intel Realsense D415

Cuando se intentan reconstruir los datos obtenidos mediante esta cámara, el bucle de control no encuentra suficientes puntos característicos en los primeros pares de imágenes, esto supone la clasificación negativa en una gran cantidad de imágenes, lo que imposibilita la reconstrucción de la escena.

Como se puede observar en la **Figura 32**, la imagen superpuesta contiene una cantidad pobre de puntos característicos (amarillos), lo que no permite emparejar esta imagen con la ubicada en el fondo. Por el contrario, en la **Figura 33** se observan una gran cantidad de puntos característicos tanto nuevos (amarillos) como ya ubicados previamente (rojos).

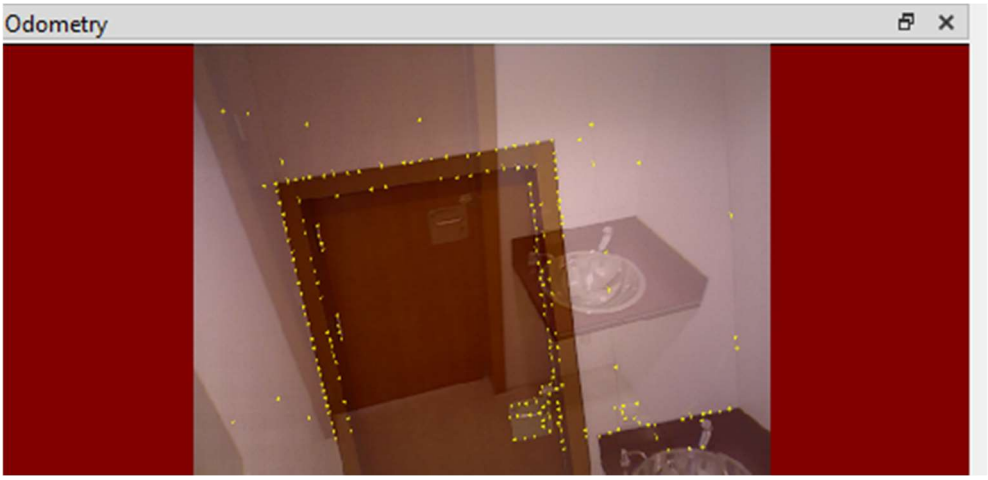


Figura 32: RTAB-Map obteniendo los puntos caracteríticos de la escena (1/2)

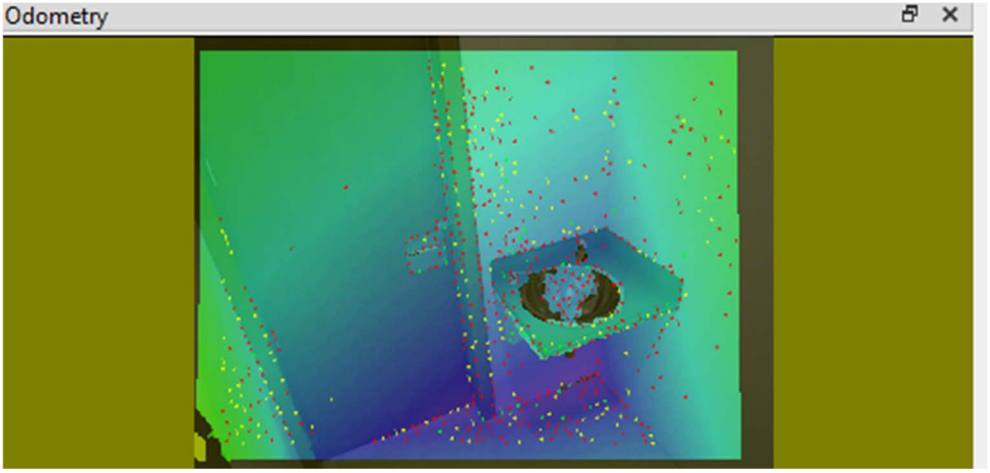


***Figura 33**: RTAB-Map obteniendo los puntos caracteríticos de la escena (2/2)*

Debido a la imposibilidad de obtener una reconstrucción 3D de la escena, por falta de puntos característicos en la misma, se ha probado a aumentar el número de puntos buscados (aumentando también el tiempo de procesamiento), al mismo tiempo, se ha incrementado el umbral de confidencia de las localizaciones (disminuyendo la precisión de la reconstrucción), sin embargo, no ha sido posible obtener puntos característicos suficientes para reconstruir la escena.

Analizando la escena escaneada se observa que existen pocos elementos diferenciales en la estancia, las paredes son lisas y las dimensiones de la escena están próximas a los límites inferiores recomendados por el fabricante. Para poder crear una reconstrucción óptima se recomienda aumentar la entropía de la información introduciendo elementos en la escena que permitan obtener nuevos puntos característicos.

7.2.4.2. Microsoft Azure Kinect DK

Al intentar reconstruir la estancia presentada anteriormente mediante el escaneo realizado con la *Microsoft Azure Kinect DK,* sucede lo mismo que con el escaneo de la *Realsense D415*. En este caso, se empieza escaneando la parte más profunda del aseo (tomando como menos profunda la puerta de entrada). Esto facilita el inicio de la reconstrucción, obteniendo durante los primeros instantes unas reconstrucciones fehacientes de la realidad.

Durante la rotación de la cámara a lo largo de la estancia, se escanean paredes lisas y sin ningún tipo de objeto sobre ellas. Quedando, sobre el campo de visión de la cámara, una superficie blanca con una entropía de información muy baja (mucho orden en la imagen). En este punto el algoritmo no es capaz de reconocer puntos característicos suficientes, imposibilitando la comparación entre localizaciones (actual y conocidas) y como consecuencia se pierde la posición de la cámara.

Cuando el algoritmo es capaz de encontrar nuevos puntos característicos en la imagen (escaneo de la pila y la puerta), la cámara se ha movido lo suficiente como para que el campo de visión haya cambiado completamente, imposibilitando nuevamente la similitud entre localizaciones o en el peor de los casos, creando similitudes entre localizaciones que realmente no existen (si se disminuye mucho el rango de confianza de la similitud).

Como se observa en la **Figura 34**, la reconstrucción no es válida, solo se reconstruye el fondo de la estancia, además, existen altos grados de distorsión en la escena, debidos a los bajos rangos

de confianza utilizados, que dan por válida la hipótesis de bucle cerrado cuando realmente no existe tal relación.

***Figura 34**: Reconstrucción 3D no válida (Azure Kinect DK)*

Para defender la premisa inicialmente establecida, mediante la cual, se determina que, si la estancia tuviese una entropía de información mayor, se encontrarían más puntos característicos y, por tanto, se podría reconstruir apropiadamente la habitación. Se pretende reconstruir la segunda estancia escaneada con la *Azure Kinect DK* (ver **Figuras 23, 24 y 25**). Esta estancia dispone de unas características especiales, tiene zonas de mucha entropía de información y espacios que se encuentran completamente vacíos y que se asimilan espacialmente al primer caso de análisis.

Cuando se intenta reconstruir la estancia, en cierta manera, sucede lo mismo que en el caso anterior, los puntos característicos son escasos en algunas localizaciones y la reconstrucción deja de avanzar. Para demostrar que únicamente fallan las zonas con poca entropía, se han separado los pares de imágenes en dos grandes grupos, aquellos que disponen de una entropía de información elevada y los que no.

Viendo las figuras mostradas a continuación, se demuestra que el algoritmo utilizado por RTABMap funciona correctamente, si se tiene información tridimensional óptima. En la **Figura 35**, se muestra la reconstrucción de la zona donde se encuentran las pantallas, teclados y

diferentes elementos que aportan información relevante a la escena. De la misma manera en la **Figura 36** se presenta una reconstrucción de la zona que queda enfrentada a la anterior.

***Figura 35**: Reconstrucción parcial (1/2) (Azure Kinect DK)*

***Figura 36**: Reconstrucción parcial (2/2) (Azure Kinect DK)*

Quiero destacar que la reconstrucción del suelo, en los lugares donde encontramos reflejos de la iluminación de la habitación, presenta defectos en su reconstrucción, de la misma forma, estos defectos, aparecen en las paredes. Este suceso ya se ha comentado anteriormente en las reconstrucciones obtenidas por la *Intel Realsense D415*.

8. Conclusión

Las técnicas basadas en estereovisión se caracterizan por obtener unas reconstrucciones muy buenas, considerando la inversión económica que supone implementar un sistema de este tipo. Estos sistemas también tienen sus desventajas, las cuales también se tratarán a lo largo de la conclusión.

Como se ha observado en las diferentes reconstrucciones existen dos características diferenciales a la hora de reconstruir las estancias. La primera depende del objeto y de la información que podemos obtener del mismo, la información puede tener mayor o menor entropía, definiendo los objetos lisos, simétricos y ordenados de baja entropía. Por el contrario, los objetos abruptos, desordenados y caóticos se consideran de alta entropía (a mayor entropía de información, mayor cantidad de puntos característicos se podrán encontrar en la imagen escaneada). La segunda, es la iluminación, un factor que por las características del proyecto queda fuera de los parámetros controlados, en las condiciones especificadas, se establece que para el escaneo no se pueden utilizar dispositivos de iluminación externos a los existentes en la escena originalmente, siendo este un factor limitante a la hora de obtener la información tridimensional de la escena.

Como se puede observar en las reconstrucciones, en especial en las realizadas con *RTAB-Map*, las técnicas de estereovisión consiguen obtener la información 3D sorprendentemente buena cuando se cumplen las condiciones comentadas anteriormente. Las superficies bien iluminadas y con alta entropía, como las mesas encontradas en la segunda estancia escaneada, quedan perfectamente definidas ya aportan gran cantidad de información, lo que facilita la adquisición de puntos característicos y por tanto su reconstrucción.

El sistema implementado también tiene sus limitaciones, siendo su punto débil las superficies uniformes (sin puntos característicos) o reflectantes. Se puede comprobar en la **Figura 26** que las superficies reflectantes no son reconstruidas adecuadamente, también encontramos el ejemplo en el suelos o paredes, donde se refleja la luz emitida por los puntos de luz. Obteniendo una reconstrucción irregular en estas superficies. En cuanto a las zonas con baja entropía, se puede afirmar que son el mayor enemigo de estas técnicas, sin los puntos característicos no es posible reconstruir la escena.

En las **Tablas 2 y 3** se observa una comparativa entre las dimensiones medidas en las escenas escaneadas y las medidas obtenidas mediante las reconstrucciones realizadas. Como se puede observar, las dimensiones globales de las estancias son fehacientes a la realidad siendo las desviaciones máximas de las plantas de la reconstrucción (ancho o largo) menores al 5 % en términos relativos. En términos absolutos las desviaciones no varían más allá de 11 cm en el peor de los casos, obteniendo medidas muy cercanas a las dimensiones reales de las estancias.

Origen de las medidas	Ancho (m)	Largo (m)	Diferencia máxima
Reales	1.35	2.22	-
Realsense D415	1.29	2.15	4.44 %
Azure Kinect DK	1.39	2.28	2.96 %

***Tabla 2**: Comparativa de dimensiones entre los modelos 3D del WC*

Origen de las medidas	Ancho (m)	Largo (m)	Diferencia máxima
Reales	2.64	6.80	-
Azure Kinect DK	2.60	6.69	1.62 %

***Tabla 3**: Comparativa de dimensiones entre los modelos 3D de la habitación*

Quiero destacar que debido a la falta de paralelismo entre paredes en los modelos tridimensionales las medidas se han obtenido calculando la media de tres mediciones obtenidas, ambas esquinas y un punto central.

Respecto a futuras áreas de desarrollo del proyecto, se plantea reconstruir las escenas utilizando una iluminación solidaria con la cámara de escaneo, facilitando el aumento de los contrastes en aquellas zonas conflictivas (entropías bajas con iluminación deficiente), lo que implicaría un aumento del número de puntos característicos encontrados a lo largo de la escena. Además, también se podría trabajar en un algoritmo que obtuviera los planos en planta de la estancia escaneada en base al modelo tridimensional obtenido.

9. Referencias

[1] Ivorra Martínez, E. (2015). Desarrollo de técnicas de visión hiperespectral y tridimensional para el sector agroalimentario. [Consulta: 9 de julio de 2021]. Disponible en: https://riunet.upv.es/handle/10251/48541

[2] Intel RealSense D400 Series Product Family Datasheet. [Consulta: 9 de julio 2021]. Disponible en: https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf

[3] Azure Kinect DK Documentation. [Consulta: 9 de julio de 2021]. Disponible en: https://docs.microsoft.com/en-us/azure/kinect-dk/

[4] Open 3D Documentation. [Consulta: 9 de julio de 2021]. Disponible en: http://www.open3d.org/docs/release/

[5] Intel Realsense Python Wrapper – GitHub Project. [Constulta: 9 de julio de 2021]. Disponible en: https://github.com/IntelRealSense/librealsense/tree/master/wrappers/python/examples

[6] Azure Kinect with Open3D [Consulta: 9 de julio de 2021]. Disponible en: http://www.open3d.org/docs/latest/tutorial/Basic/azure_kinect.html#

[7] Open 3D – Reconstruction system. [Constulta: 9 de julio de 201]. Disponible en: http://ww.open3d.org/docs/release/tutorial/reconstruction_system/index.html

[8] Open 3D – Reconstruction system – GgitHub Project. [Consulta: 9 de julio de 2021]. Disponible en: http://www.open3d.org/docs/release/tutorial/reconstruction_system/index.html

[9] RTAB-Map Documentation. [Consulta: 9 de julio de 2021]. Disponible en: http://introlab.github.io/rtabmap/

[10] López Torres P. [Consulta: 9 de julio de 2021]. Disponible en: http://bibing.us.es/proyectos/abreproy/70799/fichero/p_lopez_torres.pdf

[11] RTAB-Map – GitHub Project. https://github.com/introlab/rtabmap/wiki/Tutorials

10. Anexos

10.1. Anexo 1: Código fuente

cargar_bag_alignment.py

```python
import pyrealsense2 as rs
import numpy as np
import cv2
import argparse
import os.path


file_name="stairs"

parser = argparse.ArgumentParser(description="Read recorded bag file. Check the stream resolution, fps and format")
parser.add_argument("-i", "--input", type=str, help="Path to the bag file")# Add argument path to a bag file as an input
args = parser.parse_args(["-i", file_name + ".bag"])# Parse object

print(args.input)#Check the spelling by print the filename
if not args.input:#Error if no parameter have been given
    print("No input paramater have been given.")
    exit()
if os.path.splitext(args.input)[1] != ".bag":#Check bag extension
    print("Only .bag files are accepted")
    exit()
try:
    i = 0
    # Create pipeline
    pipeline = rs.pipeline()

    # Create a config object
    config = rs.config()
    # Tell config that we will use a recorded device from filem to be used by the pipeline through playback.
    rs.config.enable_device_from_file(config, args.input)
    # Configure the pipeline to stream the depth stream
    config.enable_stream(rs.stream.depth, 640, 480, rs.format.z16, 30)
    # config.enable_stream(rs.stream.depth, 1280, 720, rs.format.z16, 30)
    config.enable_stream(rs.stream.color, 640, 480, rs.format.rgb8, 30)
    # config.enable_stream(rs.stream.color, 1920, 1080, rs.format.rgb8, 15)

    # Start streaming from file
    pipeline.start(config)

    # Create opencv window to render image in
    cv2.namedWindow("Depth Stream", cv2.WINDOW_AUTOSIZE)
    cv2.namedWindow("Color Stream", cv2.WINDOW_AUTOSIZE)

    # Create colorizer object
    colorizer = rs.colorizer();
    align_to = rs.stream.color
    align = rs.align(align_to)

    # Streaming loop
    while True:
        # Get frameset of depth
        frames = pipeline.wait_for_frames()
        aligned_frames = align.process(frames)

        # Get depth frame
```

```
        depth_frame = aligned_frames.get_depth_frame()
        color_frame = aligned_frames.get_color_frame()

        # Validate that both frames are valid
        if not depth_frame or not color_frame:
            continue

        # Colorize depth frame to jet colormap
        depth_color_frame = colorizer.colorize(depth_frame)

        # Convert depth_frame to numpy array to render image in opencv
        depth_color_image = np.asanyarray(depth_color_frame.get_data())
        color_image = np.asanyarray(color_frame.get_data())

        # Render image in opencv window
        cv2.imshow("Depth Stream", depth_color_image)
        cv2.imshow("Color Stream", color_image)

        if i < 20:
            #cv2.imwrite(file_name + "_depth_" + str(i) + ".png", depth_color_image)
            file_name = "depth{:04d}.png".format(i)
            cv2.imwrite("Ext_from_bag/"+file_name, depth_color_image)
            file_name = "color{:04d}.jpg".format(i)
            cv2.imwrite("Ext_from_bag/"+file_name, color_image)
            i = i + 1

        key = cv2.waitKey(1)
        # if pressed escape exit program
        if key == 27:
            cv2.destroyAllWindows()
            break

finally:
    pass
```

Cargar_modelo3d.py

```python
import numpy as np
import open3d as o3d

def display_inlier_outlier(cloud, ind):
    inlier_cloud = cloud.select_by_index(ind)
    outlier_cloud = cloud.select_by_index(ind, invert=True)

    print("Showing outliers (red) and inliers (gray): ")
    outlier_cloud.paint_uniform_color([1, 0, 0])
    inlier_cloud.paint_uniform_color([0.8, 0.8, 0.8])
    o3d.visualization.draw_geometries([inlier_cloud],
                                      zoom=0.3412,
                                      front=[0.4257, -0.2125, -0.8795],
                                      lookat=[2.6172, 2.0475, 1.532],
                                      up=[-0.0694, -0.9768, 0.2024])

file_path=''
file_name='banyo_final.ply'
#file_name='integrated_banyo_14_bo_II.ply'
mesh = o3d.io.read_point_cloud(file_path + file_name)

#cl, ind = mesh.remove_statistical_outlier(nb_neighbors=400, std_ratio=2)
mesh2 = mesh.remove_statistical_outlier(nb_neighbors=400, std_ratio=2)
#o3d.io.write_triangle_mesh("banyo_final.ply", cl)
o3d.io.write_point_cloud("banyo_final.ply", cl)
#cl, ind = mesh.remove_radius_outlier(nb_points=16, radius=0.05)

display_inlier_outlier(mesh,ind)
o3d.visualization.draw_geometries([mesh2])

print('Visualización en color creada correctamente')
```

DOCUMENTO 2:

PLIEGO DE CONDICIONES TÉCNICAS

Documento 2 – PLIEGO DE CONDICIONES TÉCNICAS

1. Objeto

El presente pliego de condiciones hace referencia a la reconstrucción tridimensional de diversas estancias mediante técnicas de visión artificial. Se pretende determinar las condiciones técnicas que debe cumplir dicha reconstrucción, además de especificar las condiciones de los componentes utilizados.

2. Componentes

2.1. Cámara de escaneo tridimensional

El escaneo de la escena a reconstruir debe ser realizado mediante sistemas de estereovisión, siendo esta técnica la única empleada durante el proceso. En cuanto a la iluminación de la escena, se debe escanear con las condiciones lumínicas habituales de la estancia, es decir, sin utilizar sistemas de iluminación externos a los instalados en dicha estancia.

Respecto al carácter económico del proyecto, no se debe superar los 1000€ con el pago de los sistemas de adquisición de datos seleccionados, siendo dos el número mínimo de dispositivos de adquisición utilizados. De esta manera se asegura la comparación entre diversas fuentes de información.

El sistema de escaneo tridimensional debe ser de fácil transporte, no superar los 5 kg de masa ni los 0.008 m^3 en volumen. Además, se asegura la posibilidad de escanear escenas en espacios interiores, con superficies totales no inferiores a 5 m^2.

2.2. Reconstrucción tridimensional

La reconstrucción debe ser nítida y clara, sin ruidos que puedan perturbar la información a mostrar. Las medidas globales de las estancias (ancho y largo), no deben varian más de un 5 % respecto a la dimensión real. Considerando estancias de entre 10 y 12 m^2, un error absoluto alrededor de 18 cm.

Además, todos aquellos objetos con un volumen no superior a 0.75 m^3 quedan exentos de cumplir las especificaciones nombradas anteriormente. Esto es debido a que se trata de objetos no relevantes a la hora de representar la globalidad de la escena escaneada.

3. Prueba de servicio

Tras el escaneo y el procesamiento de los datos se deben comprobar las longitudes totales de la estancia y ratificar que respetan los márgenes establecidos en el punto anterior.

4. Condiciones de entrega

A la hora de depositar la entrega, aportar un archivo que contenga la información tridimensional de la estancia escaneada, pudiendo ser el mismo utilizado para posteriores desarrollos sobre el modelo base obtenido.

DOCUMENTO 3:

PRESUSPUESTO

Presupuesto parcial nº 1 Camaras 3D

Código	Ud	Denominación	Cantidad	Precio (€)	Total (€)
RSD415	Ud	Intel Realsense D415	1	309.82	309.82
AKDK	Ud	Microsoft Azure Kinect DK	1	337.35	337.35
TOTAL					647.17

Presupuesto parcial nº 2 Hardware de reconstrucción

Código	Ud	Denominación	Cantidad	Precio (€)*	Total (€)
ASF550C	Ud	Asus F550C	1	250	250
TOTAL					250

*Coste total del equipo en el año de compra (2019) 950€. Aplicando una depreciación anual del 26%, al iniciar el proyecto se encuentra en un valor de 456 €. El proyecto a durando 5 meses, que fijan el coste del equipo en 250 €.

Presupuesto parcial nº 3 Coste personal

Código	Ud	Denominación	Cantidad	Precio (€)	Total (€)
AL1	h	Coste laboral alumno	310	20	6200
TU1	h	Coste laboral Tutor	15	40	600
TOTAL					6800

Presupuesto parcial nº 4 Costes indirectos

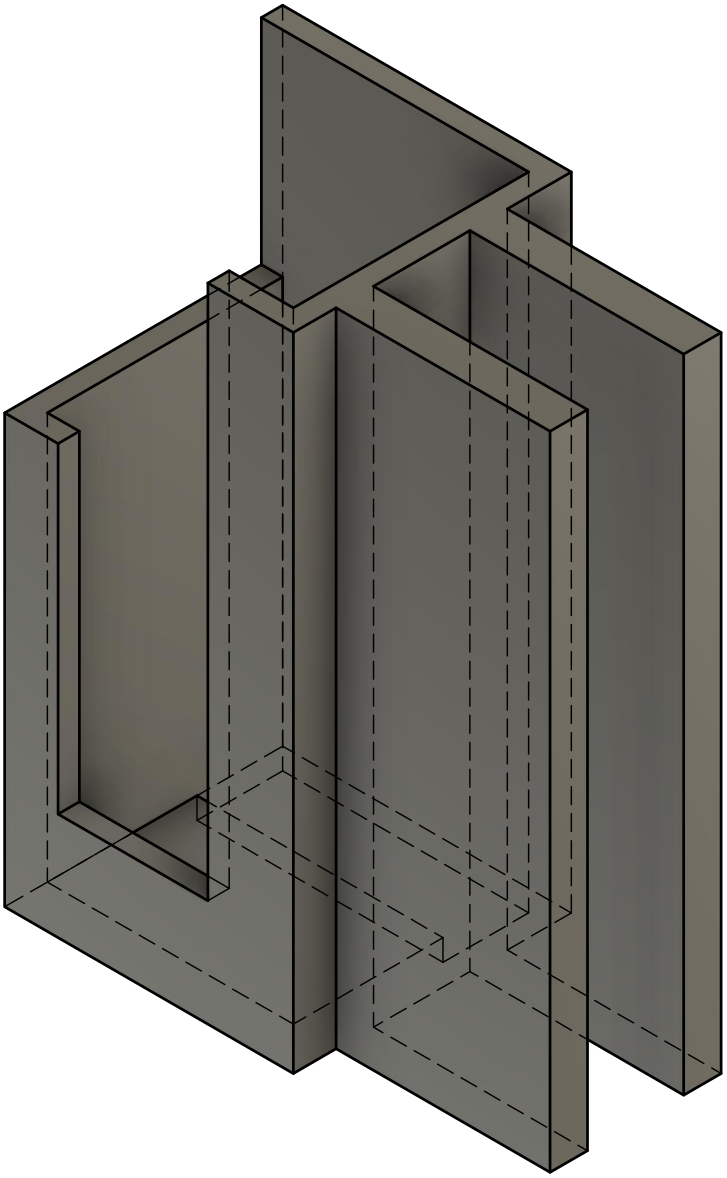
Código	Ud	Denominación	Cantidad*	Precio (€)	Total (€)
CI1	%	Costes Indirectos	7	7697.17	538.80

*Los costes indirectos tratan de estimas los gastos que no dependen directamente del proyecto, se generan mientras este se lleva a cabo. Se trata de costes de tipo muy variado y de dificil estimación, se han presupuestado en torno a un 7% del total del presupuesto.

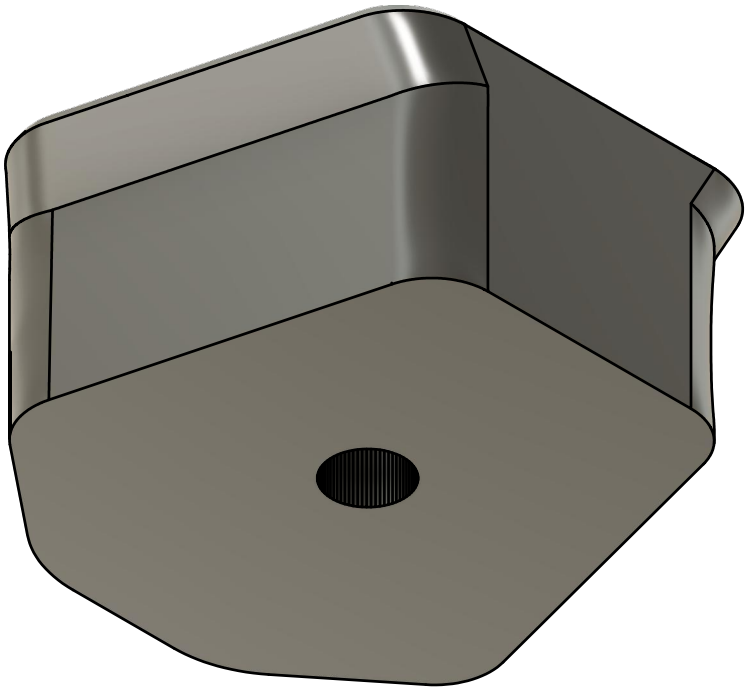
Denominación	Precio (€)
Cámaras 3D	647.17
Hardware de reconstrucción	250
Coste personal	6800
Costes indirectos	538.8
TOTAL	**8235.97**

DOCUMENTO 4:

PLANOS



Proyecto: Análisis de reconstrucción 3D mediante técnicas de VA TITULAR: Marc Ribera Vilaplana Emplazamiento: Universitat Politècnica de València Referencia: MRV202101		Fecha: 01/07/2021 Escala 2/1
Autor: Marc Ribera Vilaplana I.T.I. colegiado nº 00000000	Plano: Soporte Realsense D415	Plano Nº 01



Proyecto: Análisis de reconstrucción 3D mediante técnicas de VA TITULAR: Marc Ribera Vilaplana Emplazamiento: Universitat Politécnica de València Referencia: MRV202102		Fecha: 09/07/2021 Escala 2/1
Autor: Marc Ribera Vilaplana I.T.I. colegiado nº 00000000	Plano: Soporte Kinect DK	Plano Nº 02