



IMPROVED AR/VR TELEPRESENCE APPLICATION IN THE SMARTROOM

Trabajo de Final de Grado
Presentado en la facultad de la
Escola Tècnica d'Enginyeria de Telecomunicació de
Barcelona
Universitat Politècnica de Catalunya
por
Aina Maki

**En cumplimiento parcial de los requisitos para el grado
en INGENIERÍA DE TECNOLOGÍAS I SERVICIOS DE
TELECOMUNICACIÓN**

Tutor: Josep R. Casas

Barcelona, Septiembre 2019



Abstract

This project corresponds to the final degree project carried out at the Technical School of Telecommunications Engineering of Barcelona as a continuation of a project undertaken by the Image Processing Group of the Universitat Politècnica de Catalunya.

From a room recorded with Kinect cameras that capture in RGBD, this project, focused on telepresence conferences, eliminates in real time irrelevant background information such as the walls and ground, with the aim of suppressing the noise of these cloud points in these surfaces and decrement the computational cost.



Resum

Aquest projecte correspon al treball final de grau realitzat a l'Escola Tècnica Superior d'Enginyeria de Telecomunicacions de Barcelona com a continuació d'un projecte emprès pel Grup de Processament d'Imatge de la Universitat Politècnica de Catalunya.

A partir d'una sala grabada amb càmeres Kinect que capturen en RGBD, aquest projecte, enfocat a conferències amb telepresència, elimina en temps real informació irrelevante del segon pla com poden ser les parets i el terra, amb l'objectiu de suprimir el soroll d'aquestes superfícies i decrementar el cost computacional.



Resumen

Este proyecto corresponde al trabajo final de grado realizado en la Escuela Técnica Superior de Ingeniería de Telecomunicaciones de Barcelona como continuación de un proyecto emprendido por el Grupo de Procesamiento de Imagen de la Universitat Politècnica de Catalunya.

A partir de una sala grabada con cámaras kinect que capturan en RGBD, este proyecto, enfocado a conferencias con telepresencia, elimina en tiempo real información irrelevante de segundo plano como pueden ser las paredes y el suelo, con el objetivo de suprimir el ruido de estas superficies y disminuir el coste computacional.



Agradecimientos

Me gustaría dedicar esta sección a mi tutor de la tesis, el profesor Josep R. Casas. Por todo el tiempo que ha invertido en ayudarme, por mostrar siempre disponibilidad para responder a mis dudas y por su gran paciencia y amabilidad.

También me gustaría agradecer a Albert Gil, por guiarme con mis dudas sobre el entorno del grupo de imagen o con ROS (Robotic Operative System).

Finalmente a Arnau Roche, cuya ayuda ha sido decisiva para la finalización de este proyecto. Agradezco su tiempo dedicado y su disposición a ayudar, aportando ideas y conocimientos fundamentales para la resolución final del proyecto.



Historial de revisiones y registros de aprobación

Revisión	Fecha	Propósito
0	22/09/2019	Creación del documento
1	25/09/2019	Revisión del documento
2	28/09/2019	Revisión del documento
3	06/10/2019	Revisión final

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Aina Maki	ainamaki1@gmail.com
Josep Ramon Casas	josep.ramon.casas@upc.edu

Escrito por:		Revisado y aprobado por:	
Date	22/09/2019	Date	06/10/2019
Name	Aina Maki	Name	Josep R. Casas
Position	Autor del Proyecto	Position	Supervisor del Proyecto

Tabla de contenidos

Abstract	1
Resum.....	2
Resumen.....	3
Agradecimientos.....	4
Historial de revisiones y registros de aprobación	5
Tabla de contenidos	6
Lista de Figuras	8
Lista de Tablas	9
1. Introducción.....	10
1.1. Objetivos del proyecto	10
1.2. Planificación del proyecto	11
2. Proyecto de Telepresencia	12
2.1. Contribuciones previas al proyecto de Telepresencia	12
2.1.1. Fusión de datos en 3D para aplicaciones interactivas, por Arantxa Casanova Pagà [1]	12
2.1.2. Visualización objetiva de la experiencia de usuario en VR y AR, por Eduard Torres Romero [2]	13
2.1.3. Visualización objetiva de una escena de realidad aumentada, por Elisabet Bayó Puxan [3]	16
3. Desarrollo del proyecto.....	17
3.1. Robotic Operative System (ROS).....	17
3.1.1. Definición de ROS.....	17
3.1.2. Manejo de ROS.....	17
3.1.3. Procedimiento de aprendizaje	18
3.1.3.1. Paquetes.....	18
3.1.3.2. Publisher y Subscriber	18
3.1.3.3. Rviz	18
3.1.3.4. Point Cloud Library (PCL).....	20
3.1.3.5. Rosbag.....	21
3.1.3.6. Nodelets.....	22
3.1.3.7. RANSAC	23
4. Resultados	24
4.1. RANSAC de forma iterativa	25
4.2. Cropping inicial de la sala y RANSAC del suelo	28

4.3. RANSAC en croppings de paredes y suelo sumado al cropping del resto de la sala	29
5. Presupuesto	30
5.1. Coste de los componentes	30
5.2. Coste del personal.....	30
5.3. Coste total	30
6. Conclusiones y futuras mejoras.....	31
Bibliografía:	32
Glosario	33

Lista de Figuras

FIGURA 1: WORK PACKAGES	11
FIGURA 2: DIAGRAMA DE GANTT (1).....	11
FIGURA 3: DIAGRAMA DE GANTT (2).....	11
FIGURA 4: INCREMENTO EN LA DETECCIÓN DE DISTANCIA, POR A. CASANOVA	13
FIGURA 5: PROYECCIÓN DE DIFERENTES OBJETOS 3D SOBRE UNA IMAGEN 2D CALIBRADA CON EL TABLERO, POR E. TORRES	13
FIGURA 6: ESQUEMA DE OBTENCIÓN DE "FOREGROUND" DE UNA IMAGEN A PARTIR DE UNA REFERENCIA, POR E. TORRES	14
FIGURA 7: CORRECTA INSERCIÓN A PARTIR DEL ANÁLISIS DE PROFUNDIDAD, POR E. TORRAS.....	14
FIGURA 8: EN LA IZQUIERDA, NUBE DE PUNTOS VISUALIZADA PARA CALIBRAR EL SISTEMA. EN LA DERECHA, CÁMARA EXTERNA DEL PROCESO DE CALIBRACIÓN, POR E. TORRAS	15
FIGURA 9: CORRECTA INSERCIÓN DEL TABLERO DE AJEDREZ, POR E. TORRAS	15
FIGURA 10: REPRESENTACIÓN DE LA SALA DE CAPTURA Y LA SALA DE VISUALIZACIÓN, POR E. BAYÓ.....	16
FIGURA 11: REPRESENTACIÓN DEL FLUJO ENTRE PUBLISHER Y SUBSCRIBER [8]	17
FIGURA 12: MENÚ DEL DISPLAY TYPE	19
FIGURA 13: SELECCIÓN DEL TOPIC EMITIDO.....	19
FIGURA 14: MARCADOR DE UN CILINDRO VISUALIZADO EN RVIZ.....	19
FIGURA 15: PRISMA TRAS MODIFICAR COLOR Y OPACIDAD	20
FIGURA 16: MARCADOR DE UNA ESFERA EN RVIZ.....	20
FIGURA 17: CREACIÓN Y VISUALIZACIÓN DE UNA NUBE DE PUNTOS EN FORMA DE PLANO	20
FIGURA 18: CREACIÓN Y VISUALIZACIÓN DE UNA NUBE DE PUNTOS EN FORMA DE CILINDRO.....	21
FIGURA 19: VISUALIZACIÓN EN RVIZ DE LA SMARTROOM.....	21
FIGURA 20: EN LA IZQUIERDA, LA VISUALIZACIÓN COMPLETA DE LA SALA. EN LA DERECHA, LA NUBE DE PUNTOS RESULTANTE DE REALIZAR EL CROPPING, DONDE SE PUEDE APRECiar PARTES DEL SUEL0 AFECTADAS Y PARTES SIN AFECTAR DEBIDO A LA INCLINACIÓN DE LA SALA	22
FIGURA 21: CAPTURA COMPLETA DE LA SALA	24
FIGURA 22: RANSAC DE 1 PLAN0.....	25
FIGURA 23: RANSAC DE 2 PLANOS.....	25
FIGURA 24: RANSAC DE 3 PLANOS.....	26
FIGURA 25: RANSAC DE 4 PLANOS, CON ERROR	26
FIGURA 26: EN LA IZQUIERDA, INCORRECTA CLASIFICACIÓN DE LA PERSONA COMO PLANO. EN LA DERECHA, CORRECTA CLASIFICACIÓN DE LA PARED.....	27
FIGURA 27: CROPPING DE LAS PAREDES DE LA SALA.....	28
FIGURA 28: RESULTADO DE APLICAR RANSAC Y DETECTAR Y ELIMINAR EL SUEL0	28
FIGURA 29: RESULTADO DE APLICAR RANSAC A CADA UNO DE LOS CROPPINGS DE PAREDES Y SUEL0 SUMADO AL CROPPING DEL RESTO DE LA SALA.....	29



Lista de Tablas

TABLA 1: COSTE DE LOS COMPONENTES.....	30
TABLA 2: COSTE DEL PERSONAL	30
TABLA 3: COSTE TOTAL	30

1. Introducción

Con la evolución de la tecnología, uno de los objetivos de la humanidad ha sido mejorar la comunicación entre personas separadas por una cierta distancia, o dicho de otra forma, mejorar la comunicación entre personas que no se encuentran en un mismo lugar.

Entre estas soluciones podemos encontrar las cartas, llamadas telefónicas, los mensajes de texto, videollamadas (skype, hangouts), etc.

Cada una de ellas presenta unas ciertas características que determinan la calidad de la experiencia obtenida durante la obtención del mensaje. Por ejemplo, la diferencia que presentan los mensajes de texto respecto a las cartas corresponde a la instantaneidad de envío de la información o del mensaje que se desea comunicar del emisor al receptor.

Las características que podemos anhelar en una comunicación pueden ser la instantaneidad, la comunicación a dos bandas, la visualización del emisor o receptor durante la comunicación o incluso la percepción de hallarse físicamente en la misma localización que el emisor o receptor durante la comunicación.

Esta última característica corresponde a la telepresencia, que corresponde a una serie de tecnologías que permiten a una persona sentir que está presente o hacer parecer que está presente en un lugar diferente de su ubicación real.

El departamento de Teoría del Señal y Comunicaciones, específicamente el Grupo de Procesamiento de Imagen de la UPC, está desarrollando un proyecto de telepresencia al cual diferentes investigadores y estudiantes han contribuido, incluyendo el caso de este trabajo final de grado (TFG).

1.1. Objetivos del proyecto

Este trabajo de final de grado tiene como objetivo la eliminación de las paredes y el suelo de una sala capturada con cámaras RGBD en tiempo real para suprimir ruido en dichas superficies, que corresponden a información irrelevante durante la transmisión de telepresencia. De este modo, al mismo tiempo se consigue disminuir el coste computacional, puesto que la transmisión de esta parte de la escena en tiempo real constituye la mayor parte de la información que viaja por el canal y es, además, invariante en el tiempo.

1.2. Planificación del proyecto

En el momento de la creación del documento Critical Review se hizo una previsión del curso del trabajo, con sus respectivas creaciones de work packages y el diagrama de Gantt.

No obstante, durante la implementación del proyecto se concretó una implementación que cambió de forma significativa la planificación del proyecto.

A continuación se muestra la planificación que se ha seguido en forma de work packages



Figura 1: Work Packages

En el siguiente diagrama de Gantt se muestran los work packages en una línea temporal.

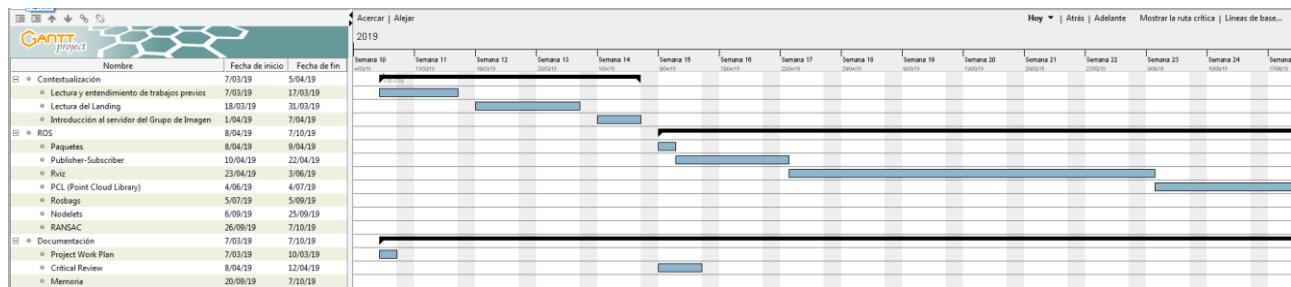


Figura 2: Diagrama de Gantt (1)

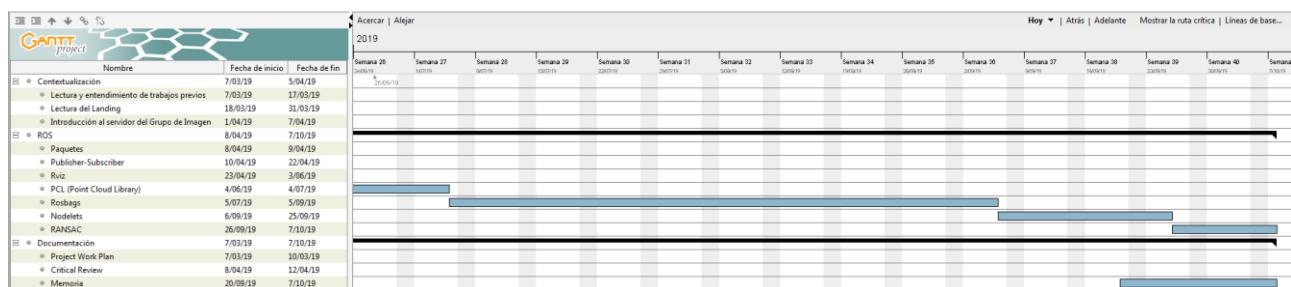


Figura 3: Diagrama de Gantt (2)

2. Proyecto de Telepresencia

El proyecto está montado en la SmartRoom del edificio D5 de Campus Nord UPC. Esta sala está equipada con múltiples sensores RGBD, específicamente 4 en estos momentos, que permiten adquirir escenas 3D en tiempo real.

3 de estos sensores capturan gran parte de la sala, y está organizado para que una persona que se coloque gafas de realidad virtual en la otra parte de la sala quede inmersa en esta escena y pueda visualizar en tiempo real lo que ocurre en ella. La visualización sería la misma aunque el usuario se situara con las gafas en un lugar remoto.

El cuarto sensor captura a la persona que se quiere realizar la inmersión en la escena, o sea la persona que visualiza la sala mediante las gafas de realidad virtual, de tal forma que se puede realizar una inserción de esta persona con Realidad Aumentada en el primer escenario.

Como ya se ha mencionado en apartados anteriores, este proyecto ha sido desarrollado por el Grupo de Procesado de Imagen de la UPC, y este TFG corresponde a una contribución que se realiza en él.

De este mismo modo, diferentes estudiantes han realizado su trabajo final de grado o máster aportando módulos en este proyecto, y a continuación se detallan y se explican tres trabajos realizados en los últimos años para poder comprender de dónde provienen algunas de las funciones que dispone el proyecto y cómo han sido implementadas.

2.1. Contribuciones previas al proyecto de Telepresencia

2.1.1. **Fusión de datos en 3D para aplicaciones interactivas, por Arantxa Casanova Pagà [1]**

En este TFG se presenta la implementación de diversos algoritmos de registro para la fusión de datos 3D de dos sensores RGBD distintos, obteniendo de este modo un aumento del área de detección.

Se toma como referencia un dedo humano que recorre todo el área cubierto por los dos sensores registrados, y se consigue la calibración mediante auto registros de patrones geométricos.

Los resultados obtenidos con los sensores a 120 cm de distancia entre ellos y con un mínimo de 30 cm de superposición fueron de un incremento del 78.5% en la anchura del campo visual de cobertura sin alterar la calidad de registro.

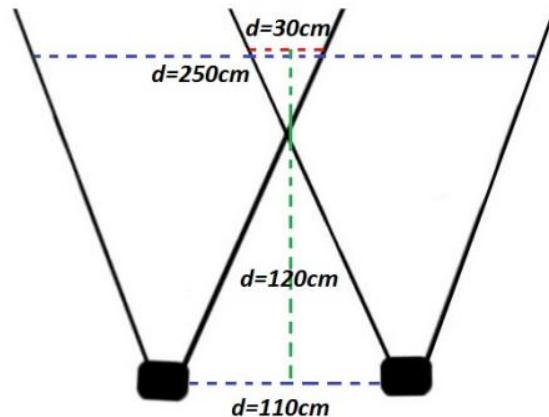


Figura 4: Incremento en la detección de distancia, por A. Casanova [1]

Como podemos observar, gracias al trabajo de esta tesis y actualmente fusionando los datos capturados por 3 sensores RGBD podemos capturar aproximadamente la mitad de la sala SmartRoom.

2.1.2. Visualización objetiva de la experiencia de usuario en VR y AR, por Eduard Torres Romero [2]

El principal objetivo de este TFG es la visualización objetiva (en tercera persona) de la experiencia de usuario en tiempo real durante su navegación en realidad virtual (VR) o realidad aumentada (AR). La visualización se implementa mediante la inserción de elementos virtuales en la imagen de una cámara externa, que deberán tener posición, escala y rotación coherentes con la percepción subjetiva del usuario.

Para alcanzar este objetivo, los pasos que se han seguido son los siguientes:

1. Estudio de los principales problemas, posibles soluciones y métodos susceptibles de ser usados.
2. Inserción de elementos 3D sobre imágenes 2D. Este punto sirve para comprobar la correcta calibración de la cámara e interpretar la geometría 3D de ésta.

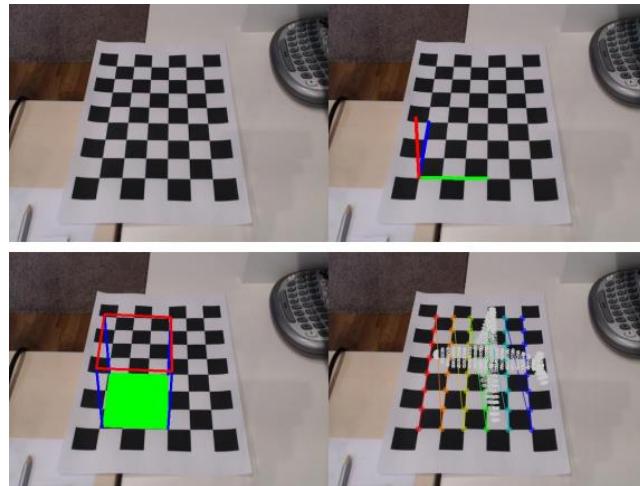


Figura 5: Proyección de diferentes objetos 3D sobre una imagen 2D calibrada con el tablero, por E. Torres [2]

3. Segmentación foreground-background de la imagen obtenida de la cámara y de la escena virtual. De este modo se puede realizar un ensamblaje de capas que simulan la integración de ambas escenas.

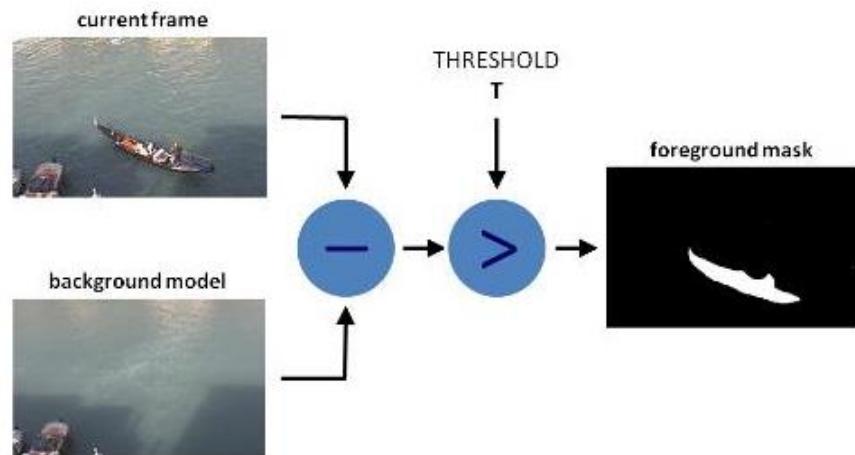


Figura 6: Esquema de obtención de "foreground" de una imagen a partir de una referencia, por E. Torres [2]

4. Adaptación del sistema añadiendo un análisis exhaustivo de la escena para realizar solapamientos de mayor complejidad.



Figura 7: Correcta inserción a partir del análisis de profundidad, por E. Torres [2]

5. Generación de la nube de puntos global que fusiona la recibida con la generada a partir de la cámara de profundidad.

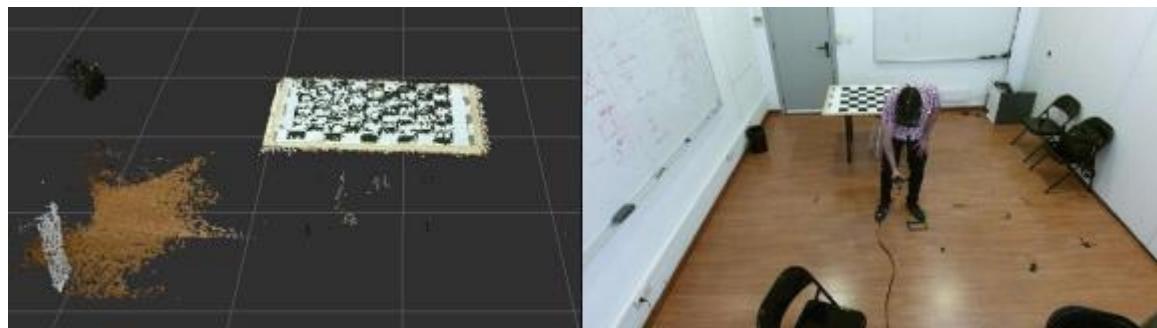


Figura 8: En la izquierda, nube de puntos visualizada para calibrar el sistema. En la derecha, cámara externa del proceso de calibración, por E. Torres [2]



Figura 9: Correcta inserción del tablero de ajedrez, por E. Torres [2]

Con la implementación de este TFG, un usuario puede observar mediante la ayuda de un monitor la SmartRoom junto con elementos inseridos con realidad aumentada, correspondientes a la escena que el usuario de la sala observa des de las gafas mediante realidad virtual.

2.1.3. Visualización objetiva de una escena de realidad aumentada, por Elisabet Bayó Puxan [3]

En este TFG se presenta la implementación de un bloque de visualización en realidad aumentada.

Para ello, el procedimiento ha sido el siguiente:

1. Se hace un registro entre la nube de puntos de la sala de visualización y la nube de puntos de la sala de captura, y de este modo conseguimos ver de forma objetiva la interacción de la usuaria y la sala remota.

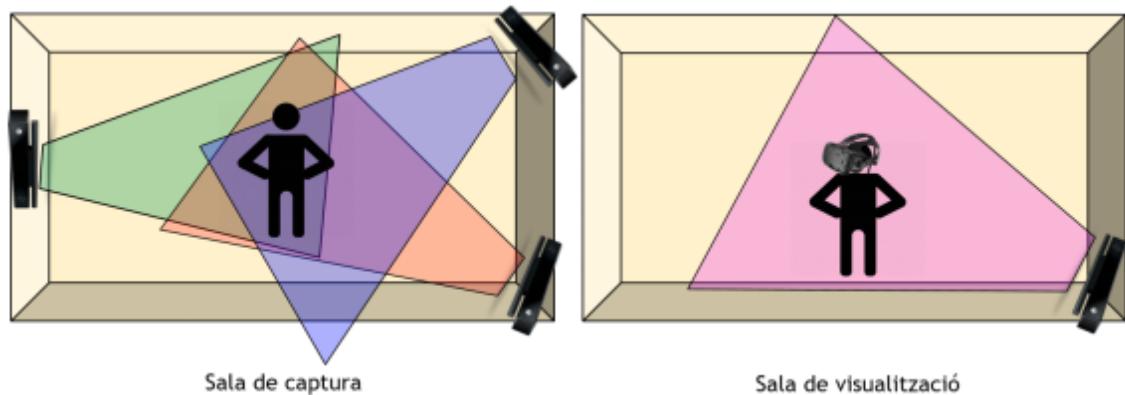


Figura 10: Representación de la sala de captura y la sala de visualización, por E. Bayó [3]

2. Se hace un recorte de la nube de puntos de la sala de visualización, dejando solamente la parte que nos interesa de dicha sala para el momento de fusión con la nube de puntos de la sala remota.
3. Unión de ambas nubes de puntos.
4. Comprobación con las gafas de realidad virtual, en las que se visualizan ambas salas unidas (con elementos de ambas).

3. Desarrollo del proyecto

Para la realización de este proyecto ha sido necesario el aprendizaje del uso de la plataforma ROS [4][5] (Robotic Operative System), programado en C++.

3.1. Robotic Operative System (ROS)

3.1.1. Definición de ROS

ROS [6] es un framework para el desarrollo de software para robots, con un entorno flexible que facilita la colaboración entre diferentes desarrolladores.

ROS provee los servicios estándar de un sistema operativo tales como abstracción del hardware, control de dispositivos de bajo nivel, implementación de funcionalidad de uso común, paso de mensajes entre procesos y mantenimiento de paquetes. Está basado en una arquitectura de grafos donde el procesamiento toma lugar en los nodos que pueden recibir, mandar y multiplexar mensajes de sensores, control, estados, planificaciones y actuadores, entre otros. La librería está orientada para un sistema UNIX (Ubuntu (Linux)).

ROS tiene dos partes básicas: la parte del sistema operativo, ros, y ros-pkg, una suite de paquetes aportados por la contribución de usuarios que implementan las funcionalidades tales como localización y mapeo simultáneo, planificación, percepción, simulación, etc.

3.1.2. Manejo de ROS

ROS funciona a base de nodos [7], que corresponden a archivos ejecutables dentro de un paquete ROS. Utilizan una biblioteca cliente para comunicarse con otros nodos y pueden utilizar o proporcionar algún servicio.

Esta comunicación se hace a través de los tópicos, y los nodos pueden publicar o suscribirse a ellos.

Para permitir la comunicación entre nodos, es necesario tener un roscore corriendo, pero en el caso que se ejecuten distintos nodos en diferentes máquinas, es necesario arrancar un roscore a una máquina y también especificar un puerto. Éste corresponde al master, y se encarga de iniciar la comunicación entre los distintos nodos, diferenciando los que son publicados o suscritos y de recordar los tópicos .

Para poner un ejemplo, un procedimiento habitual sería un sensor que captura información, indica que quiere publicar un tópico al máster. A continuación el visualizador indica que quiere suscribirse a ese tópico, y el máster conecta a estos dos nodos para que haya comunicación entre ellos de forma directa.

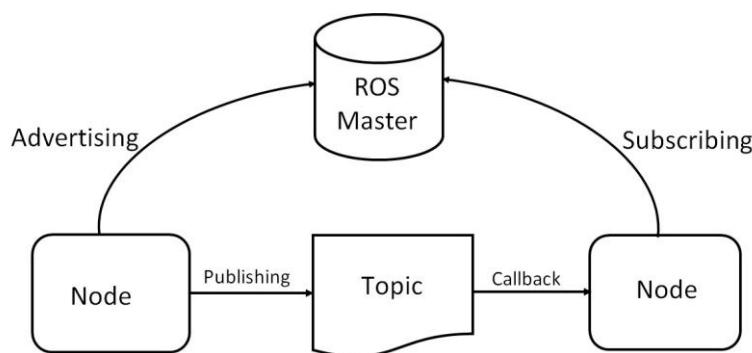


Figura 11: Representación del flujo entre Publisher y Subscriber [8]

3.1.3. Procedimiento de aprendizaje

En este apartado se detallarán las diferentes herramientas que se han usado en ROS en el mismo orden que en el procedimiento de aprendizaje [9] que se ha establecido en este proyecto. De este modo se puede apreciar la construcción del entorno de trabajo y los pasos indispensables para alcanzar los objetivos propuestos en este TFG.

3.1.3.1. Paquetes

El primer paso para entrar en contacto con ROS es la creación de un paquete [10].

El software en ROS está organizado en paquetes. Tienen como objetivo proveer funcionalidades útiles, fáciles de ejecutar y de reutilizar por otros softwares.

Los paquetes pueden contener nodos de ROS, librerías independientes de ROS, datasets, archivos de configuración, etc.

3.1.3.2. Publisher y Subscriber

Como se ha explicado anteriormente, los nodos se diferencian entre publishers y subscribers, dependiendo de si publican o se suscriben a un tópico.

El mejor modo de entender esta interacción es creando estos dos nodos y hacer que interactúen.

Para ello, dentro de un paquete, creamos dos source files. En uno de ellos hacemos que envíe un mensaje de texto cada intervalo de tiempo definido, que corresponderá al nodo del publisher. El otro simplemente espera a que haya un mensaje de entrada mediante una función de callback.

En el momento de ejecución

```
amaki@nx1:~/catkin_ws>> rosrun beginner_tutorials listener
```

El subscriber espera a tener mensajes de entrada. Una vez ponemos en marcha al publisher:

```
amaki@nx1:~/catkin_ws>> rosrun beginner_tutorials talker &
[2] 26962
```

```
amaki@nx1:~/catkin_ws>> [ INFO] [1569762391.925624767]: hello world 0
```

3.1.3.3. Rviz

Una vez comprobado que se entiende la nomenclatura entre publishers y subscribers, es interesante empezar a visualizar mensajes, no sólo de texto sino también formas, grabaciones, etc. Este TFG trabaja con datos en 3D, formados con una nube de puntos. Para poder visualizar este tipo de información necesitamos la herramienta Rviz.

Rviz [11] es un visualizador 3D que muestra datos de sensores y estado de información de ROS en tiempo real.

Se trata de un nodo subscriber, que como hemos indicado anteriormente, se suscribe a un tópico que está siendo publicado.

En este caso, debemos indicar inicialmente qué tipo de mensajes son los que se espera recibir (display type), y a continuación a qué tópico se quiere suscribir

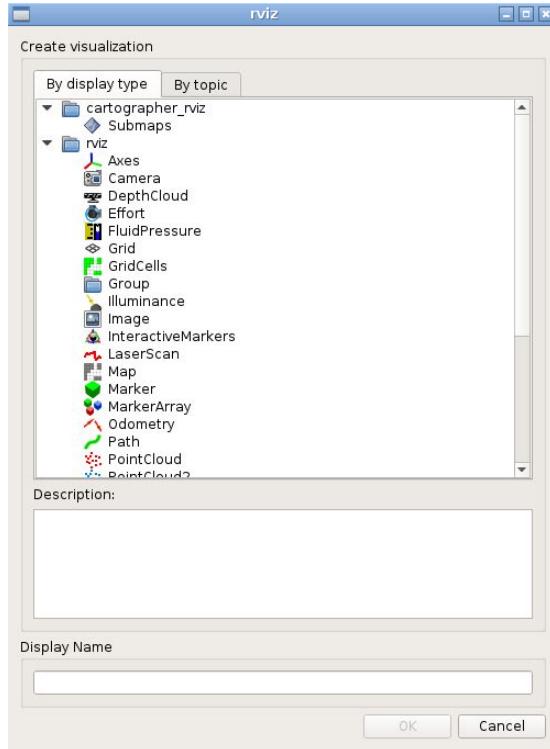


Figura 12: Menú del Display Type



Figura 13: Selección del topic emitido

Existen múltiples paquetes de ejemplo para poder visualizar formas básicas ya definidas para familiarizarse con este visualizador. Si se ejecuta, lo que se puede observar es lo siguiente:

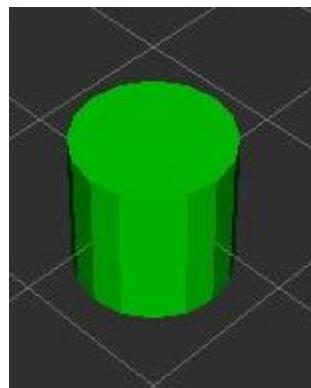


Figura 14: Marcador de un cilindro visualizado en Rviz

En este punto, es interesante alterar el código para cambiar características de las formas básicas ya creadas, como por ejemplo el color, opacidad, etc. para familiarizarse más con la creación de mensajes en ROS.

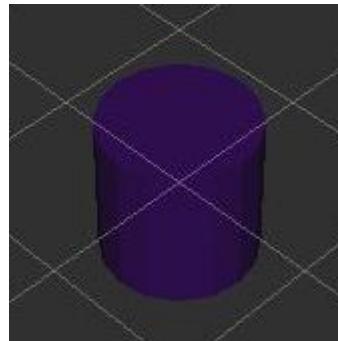


Figura 15: Prisma tras modificar color y opacidad

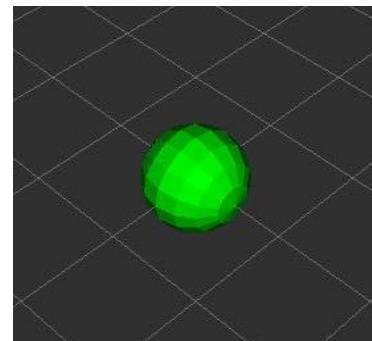


Figura 16: Marcador de una esfera en Rviz

3.1.3.4. Point Cloud Library (PCL)

Dado que la información que se desea procesar son una serie de nubes de puntos, es importante también familiarizarse con ellos.

Para iniciar el contacto con las nubes de puntos, la creación de formas básicas y visualización con Rviz es un buen acercamiento.

Para la creación de nubes de puntos podemos utilizar la librería Point Cloud Library o PCL [12]. Se trata de una librería open source con algoritmos para el procesamiento de nubes de puntos y procesamiento de geometría en 3D.

A partir de las herramientas que nos proporciona esta librería, se puede lograr escribir un código para la creación de nubes de puntos, indicando la localización deseada de cada punto y definir de este modo la forma deseada

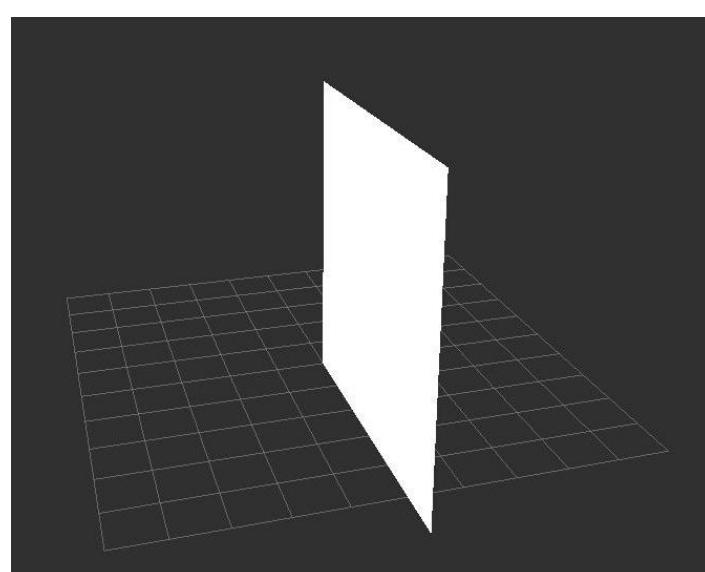


Figura 17: Creación y visualización de una nube de puntos en forma de plano

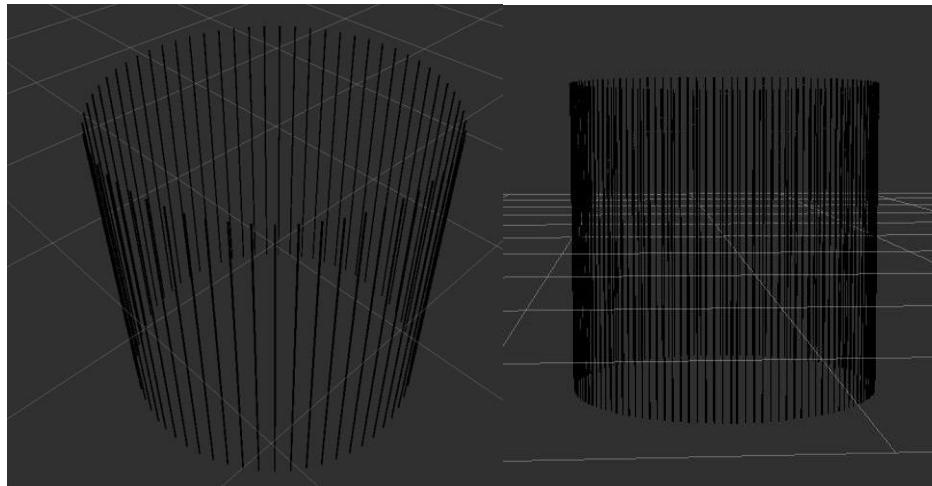


Figura 18: Creación y visualización de una nube de puntos en forma de cilindro

3.1.3.5. Rosbag

El siguiente paso para continuar es el trabajo con nubes de puntos dinámicos.

Los rosbags [13] son formatos de fichero para almacenar datos de mensajes. Existen diversas herramientas para almacenar, procesar, analizar y visualizar estos ficheros.

Por lo tanto, se han utilizado los rosbags para grabar escenas en la SmartRoom y posteriormente visualizarlas.

Una ventaja que podemos apreciar trabajando con rosbags es que de una sola grabación se puede crear diferentes tópicos, correspondientes a la grabación original o a la grabación tras realizar algún procesado, sin alterar el archivo original. Dependiendo del tópico al que se suscriba el subscriber se podrá visualizar una resultado distinto.

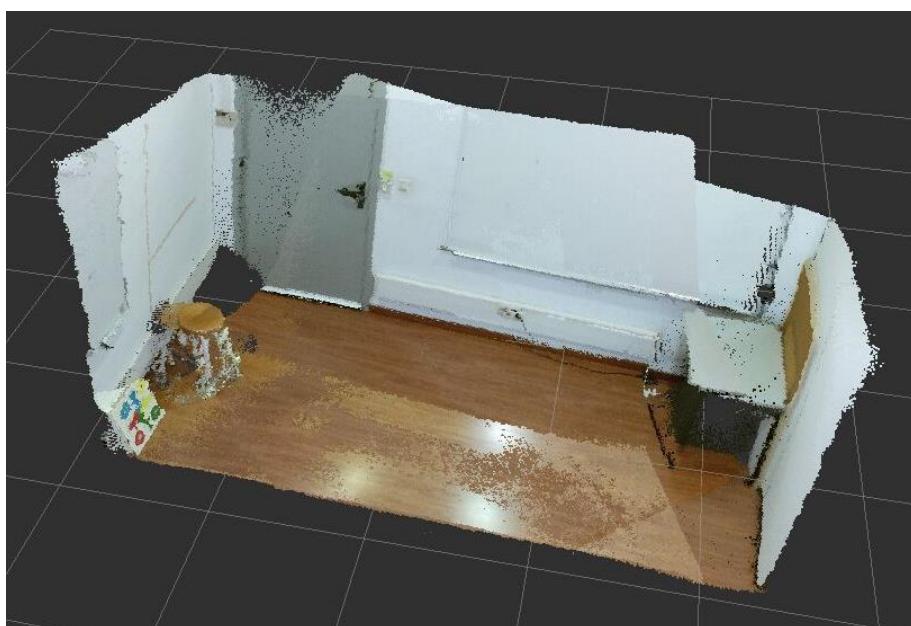


Figura 19: Visualización en Rviz de la Smartroom

3.1.3.6. Nodelets

El objetivo de este TFG es la detección y eliminación de las paredes y el suelo de una sala grabada con cámaras kinect a tiempo real. Como se explicó en el apartado **¡Error! No se encuentra el origen de la referencia.**, la intención es eliminar información de segundo plano irrelevante para la aplicación de telepresencia, disminuyendo los costes de transmisión y computación para el tratamiento de las nubes de puntos en tiempo real.

Dado que los nodos se envían mutuamente información a través de la tarjeta de red, cuando los nodos se ejecutan en una misma máquina y el tamaño de los mensajes es considerable como es en el caso de las nubes de puntos, es preferible la utilización de nodelets. Éstos son una versión más elaborada de los nodos de ROS que ahorran la transmisión de datos a través de la tarjeta de red cuando publisher y subscriber se encuentran en la misma máquina. Los nodelets permiten realizar el mismo proceso empleando *shared memory*, es decir, enviando simplemente el puntero de memoria (la posición de memoria en la que se puede encontrar), y agilizando la comunicación en este caso particular.

Como primera estrategia, se ha realizado un recorte (cropping) básico en la sala. En este cropping se pueden elegir los valores mínimo y máximo de las coordenadas x, y, z que se desea conservar, resultando un prisma rectangular con los costados paralelos a los ejes de coordenadas.

El problema con esta estrategia es su rigidez, puesto que la sala no corresponde a un prisma rectangular y la orientación de la sala tampoco tiene porqué resultar exactamente paralela a los ejes de coordenadas de la nube de puntos.

Cuando se realiza un cropping ajustando los valores, se pueden observar áreas donde se hace una correcta eliminación de las paredes o el suelo, pero se observan partes en que debido a la inclinación relativa de los ejes de coordenadas y la sala, los valores deberían ser más elevados (el recorte más grosero) para que esas partes no permanezcan en la nube de puntos procesada.

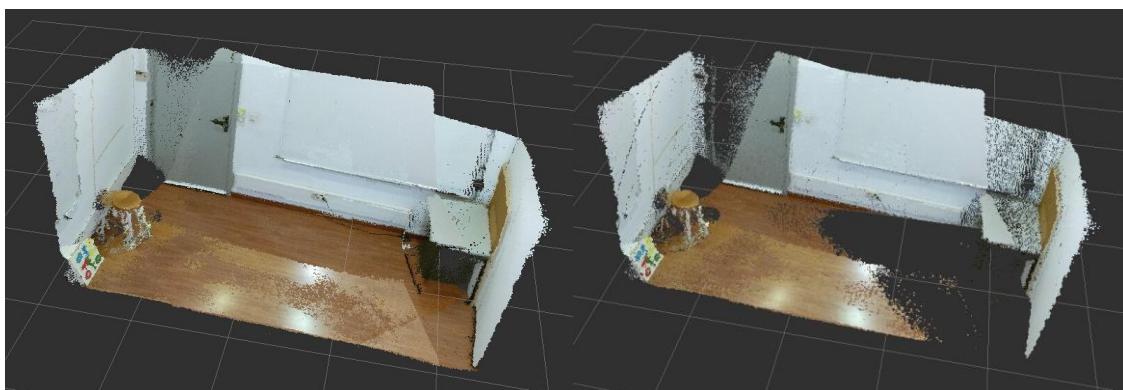


Figura 20: En la izquierda, la visualización completa de la sala. En la derecha, la nube de puntos resultante de realizar el cropping, donde se puede apreciar partes del suelo afectadas y partes sin afectar debido a la inclinación de la sala.

En el caso de determinar unos valores máximo y mínimo que aseguren la eliminación completa de las paredes y el suelo, se estaría eliminando también elementos que no corresponden al objetivo.

Por ese mismo motivo se propone una segunda estrategia basada en el análisis geométrico, que permitirá la detección de las paredes y suelo para su eliminación.



3.1.3.7. RANSAC

Para la detección de las paredes y el suelo utilizaremos RANSAC[14].

Random sample consensus (RANSAC) es un método iterativo para calcular los parámetros de un modelo matemático de un conjunto de datos observados que contiene ruido y outliers (valores atípicos). Es un algoritmo no determinista en el sentido de que produce un resultado razonable sólo con una cierta probabilidad, mayor a medida que se permiten más iteraciones que aplicadas imagen a imagen podrían aumentar significativamente el tiempo de computación.

Los datos consisten en "inliers", es decir, los datos cuya distribución se explica por un conjunto de parámetros del modelo, aunque pueden estar sujetos a ruido, y "outliers" "valores atípicos", que son datos que no encajan en el modelo.

En el caso particular de la SmartRoom, los inliers corresponden a los planos geométricos correspondientes a las paredes y el suelo, mientras los outliers corresponde al resto de la sala con todos los objetos contenidos en ella.

4. Resultados

Para mostrar los resultados obtenidos, se irán comparando respecto a la captura completa de la sala, correspondiente a suelo, 3 paredes, diversos objetos en los costados (un tablero, un taburete y una mesa) y una persona en el centro de la sala.

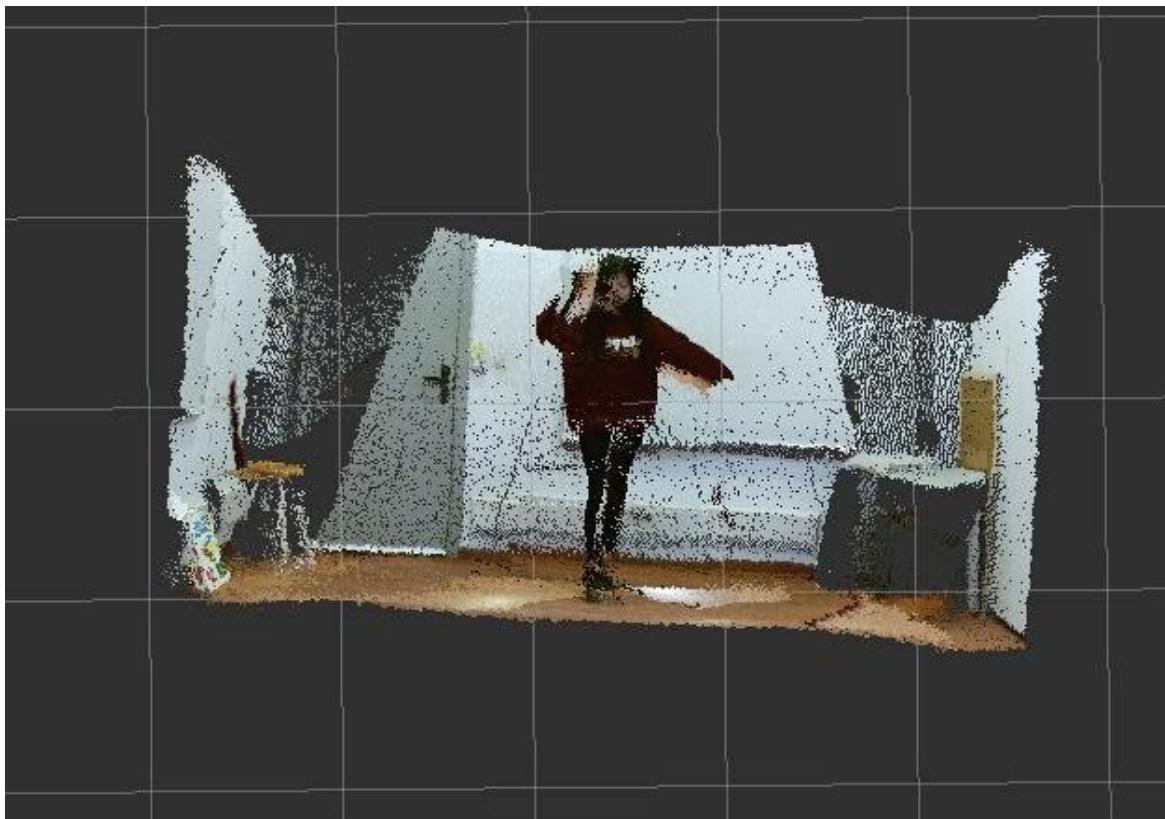


Figura 21: Captura completa de la sala

Considerando el objetivo de eliminar las paredes y el suelo de esta sala, y considerando el tiempo de computación y la eficiencia de la eliminación, proponemos 3 posibles resoluciones:

1. Detectar y eliminar con RANSAC de forma iterativa los diferentes planos hasta eliminar las tres paredes y el suelo indeseados.
2. Aplicar un cropping inicial en la sala, ajustando los valores al máximo para no perder información relevante de la sala (los objetos de los costados) pero al mismo tiempo eliminar las paredes. Posteriormente aplicar RANSAC para detectar el suelo y eliminarlo.
3. Dividir la sala en 5 partes, cada una de las paredes, el suelo y el resto de la sala. Aplicar RANSAC en los croppings de las paredes y suelo y juntar el resultado con el cropping del resto de la sala.

4.1. RANSAC de forma iterativa

Para probar este método se ha ido eliminando planos de forma progresiva, para poder observar el correcto funcionamiento.

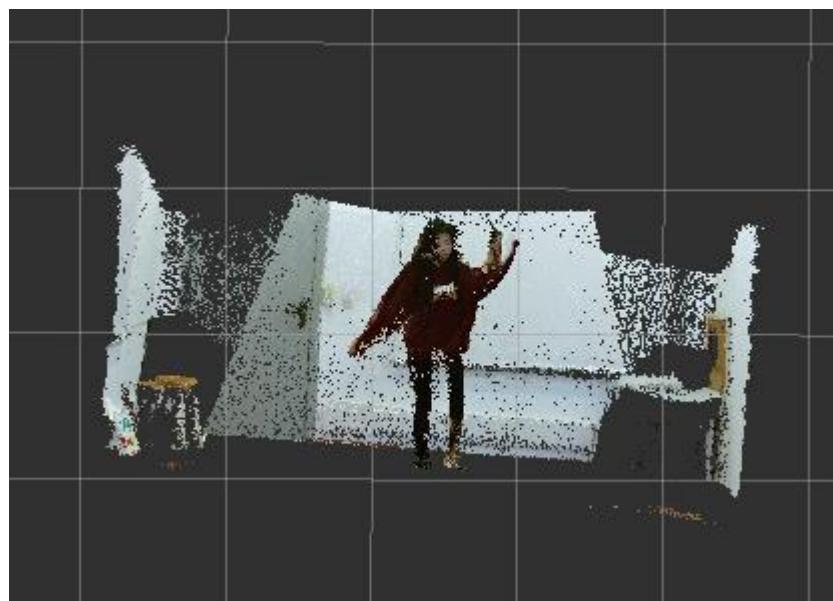


Figura 22: RANSAC de 1 plano

Cuando aplicamos RANSAC una vez, el plano detectado y eliminado es el suelo. Este comportamiento es predecible ya que se trata de la superficie plana más amplia y más regular.

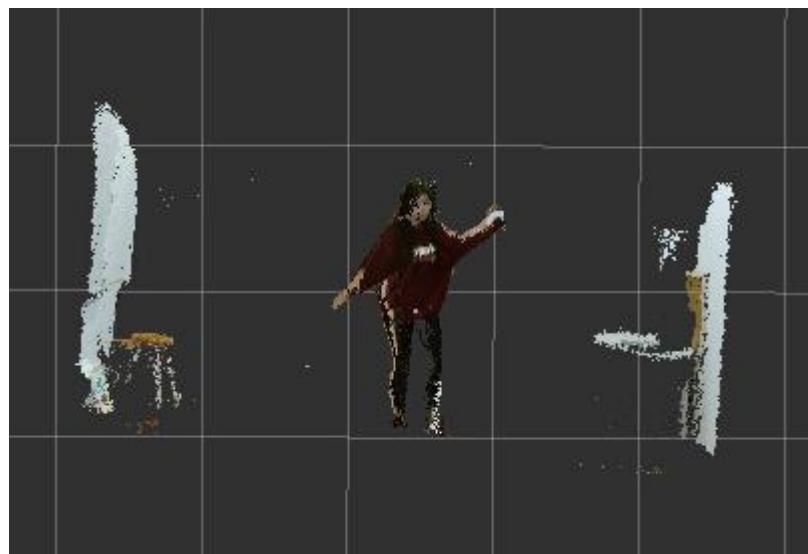


Figura 23: RANSAC de 2 planos

En la segunda iteración se detecta y elimina la pared frontal. Como en el anterior caso, se trata de la superficie más grande que restaba de las 3.

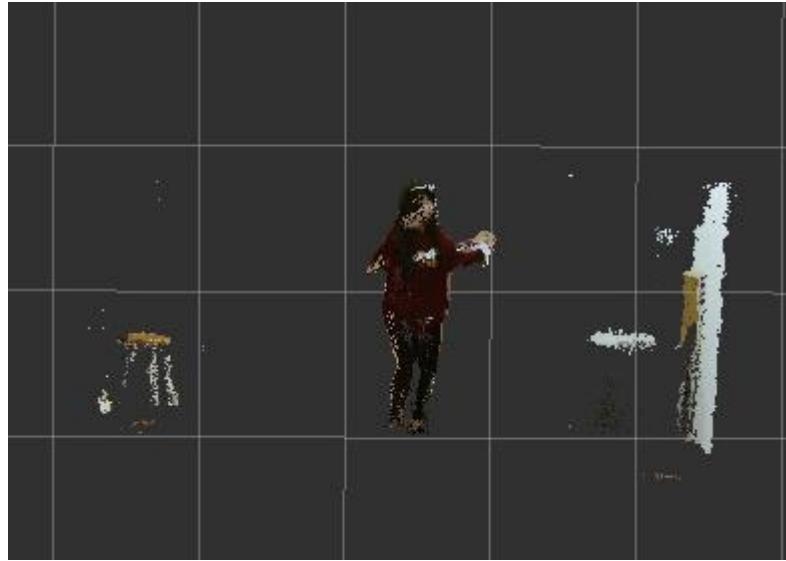


Figura 24: RANSAC de 3 planos

En la tercera iteración la pared de la izquierda ha sido eliminada. Es importante observar que el taburete sigue intacto y que solamente se ha eliminado la pared.

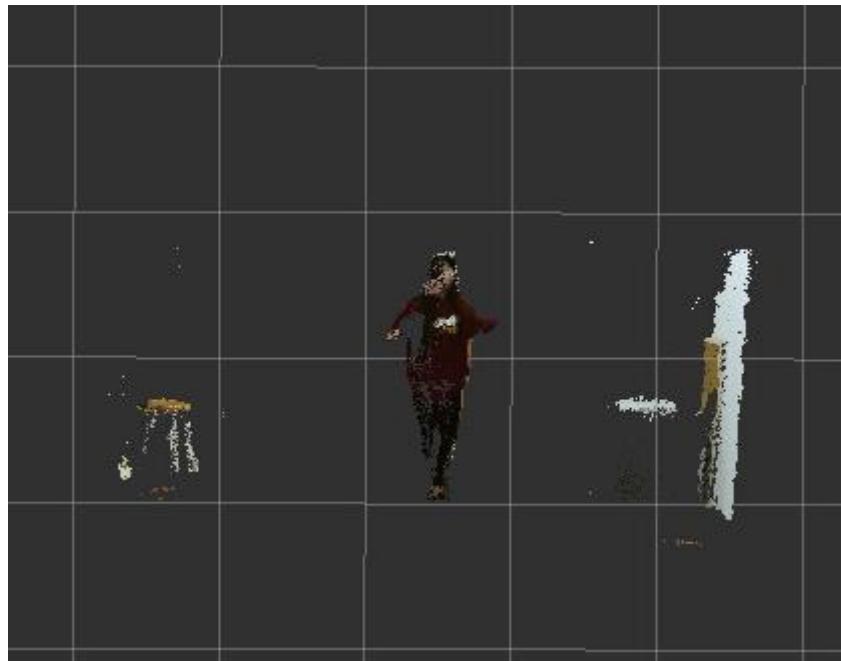


Figura 25: RANSAC de 4 planos, con error

En la 4^a iteración observamos que la detección de la 3^a pared no se realiza correctamente, y que por lo tanto no ha sido eliminada de la nube de puntos. Incluso ajustando los parámetros no se consiguen mejores resultados.

Esto es debido a que la pared restante contiene pocos puntos proporcionalmente a los del resto de la sala y tan poca superficie que RANSAC no es capaz de encontrarlo. Es más, confunde otras superficies clasificándolas momentáneamente como planos.

Para mostrar este hecho, se realiza una visualización de lo que RANSAC clasifica como plano en la 4^a iteración. Lo que se observa varía en el tiempo de forma muy inestable (en

algunos frames elimina la pared y en otros no, puesto que el procesado de cada frame es independiente):

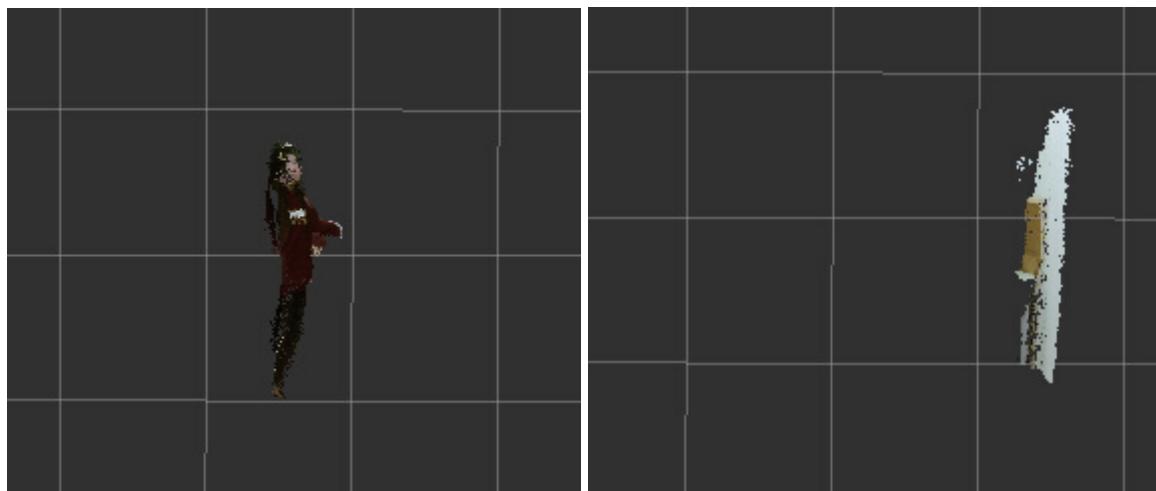


Figura 26: En la izquierda, incorrecta clasificación de la persona como plano. En la derecha, correcta clasificación de la pared.

Cabe decir que este método no es el ideal. Primeramente, como acabamos de observar, no es capaz de detectar todos los planos.

Además, el proceso de aplicar RANSAC de forma iterativa 4 veces supone un alto coste computacional y un ralentizado del procesado, suponiendo algunas paradas en el visionado de la grabación, que aparece congelado en algunas veces cuando se producen pérdidas de frames (frame drops: frames no analizados para mantener el procesado en tiempo real del sensor).

4.2. Cropping inicial de la sala y RANSAC del suelo

Con este método resolvemos el problema que teníamos en el anterior caso, debido a que se aplica RANSAC a una muestra de nube de puntos mucho más reducida.

El resultado de aplicar el cropping es el siguiente:

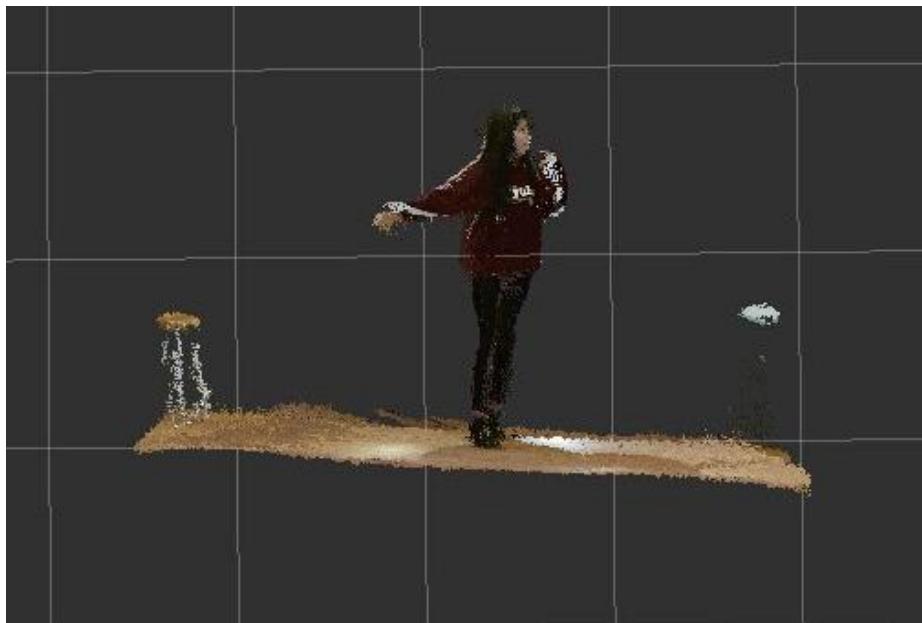


Figura 27: Cropping de las paredes de la sala

Lo que sí que debemos tener en cuenta en este caso es que perdemos un poco de información de la sala. Seguimos viendo el taburete pero debido al cropping hemos perdido información del tablero.

A continuación se aplica RANSAC para detectar y eliminar el plano del suelo.

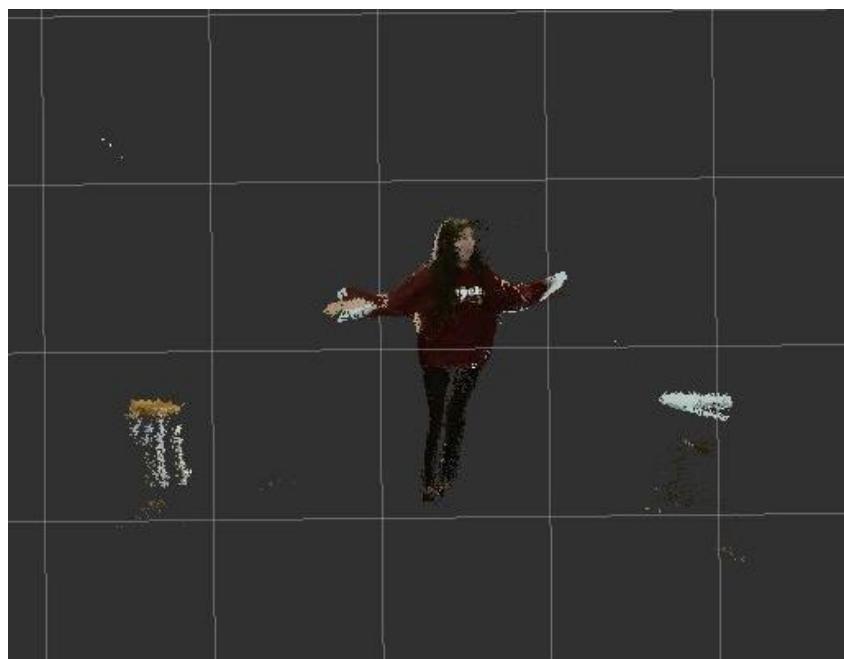


Figura 28: Resultado de aplicar RANSAC y detectar y eliminar el suelo

4.3. RANSAC en croppings de paredes y suelo sumado al cropping del resto de la sala

A diferencia del método de aplicar RANSAC de forma iterativa, con este método reducimos el coste computacional al reducir considerablemente la cantidad de puntos a aplicar RANSAC.

Además, dado que en cada cropping se observa mayoritariamente la pared o el suelo que se quiere eliminar, nos aseguramos que éste será el plano que será detectado y eliminado, evitando de este modo el otro problema que surgió en el caso de aplicar RANSAC de forma iterativa, en el que no se consiguió detectar la última pared de forma correcta.

El resultado que se ha obtenido es el siguiente:

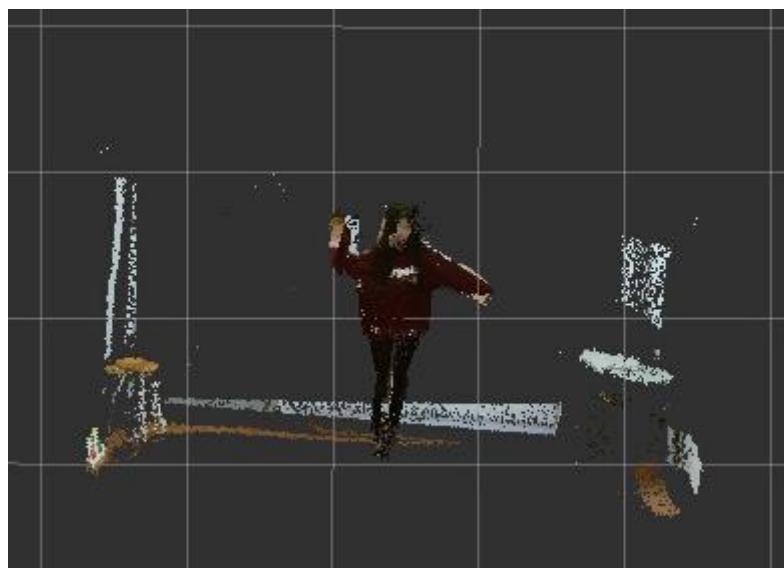


Figura 29: Resultado de aplicar RANSAC a cada uno de los croppings de paredes y suelo sumado al cropping del resto de la sala

Como se puede observar, con este resultado obtenemos partes de paredes y suelo que no han sido eliminadas. Esto es debido a que en el momento que se aplica RANSAC en un cropping y se elimina la superficie detectada, siguen quedando los restos de las otras paredes o suelo que existen dentro del cropping realizado. De este modo, cuando se suman todos los croppings, se pueden observar estos restos.

Comparando este método con el método de realizar un crop inicial de la sala y posteriormente aplicar RANSAC para eliminar el suelo podemos decir que en este caso se conservan mejor los elementos que se encuentran en la sala, concretamente los que se sitúan apoyados en las paredes. En la imagen podemos observar que se preserva perfectamente el taburete y el tablero de parchís.

5. Presupuesto

5.1. Coste de los componentes

Teniendo en cuenta que se trata de la continuación de un proyecto ya implementado, el material ha sido proporcionado por el GPI de la UPC. No obstante, a continuación se detalla el coste de cada componente utilizado.

Componentes	Unidades	Coste	Total
Cámara Kinect v2	3	119,95 €	359,85 €
Ordenador portátil	1	1.100,00 €	1.100,00 €
Ordenador	2	700,00 €	1.400,00 €
Total			2.859,85 €

Tabla 1: Coste de los componentes

Aunque en el proyecto de Telepresencia se llegan a utilizar 4 cámaras kinect, en este proyecto solamente se utilizan 3 para capturar la sala.

El ordenador portátil ha sido utilizado para el desarrollo del proyecto. Los dos ordenadores corresponden a las máquinas que se utilizan para la configuración de desarrollo del sistema. Una se utiliza como interfaz principal. Desde ella se controlan las cámaras de la sala, entre otras funciones.

El ordenador restante sirve para conectar las gafas de realidad virtual y tiene instalado el software necesario para su uso.

5.2. Coste del personal

En la siguiente tabla se detalla el coste total del personal.

Personal	Horas	Coste (€/h)	Seguridad social	Total
Becario	540	8	1.296,00 €	5.616,00 €
Tutor del proyecto	50	25	375,00 €	1.625,00 €
Técnico del proyecto	15	25	112,50 €	487,50 €
Total				7.728,50 €

Tabla 2: Coste del personal

El trabajo del becario corresponde a mi trabajo realizado durante el proyecto.

El trabajo del tutor, como el nombre indica, corresponde al trabajo realizado por el tutor o mentor del proyecto, Josep R. Casas.

El trabajo del técnico corresponde al trabajo realizado por Albert Gil, respondiendo dudas y solucionando errores de la parte más técnica del proyecto.

5.3. Coste total

Así pues, el coste total del proyecto realizado en este TFG se estima que es aproximadamente de 10.600€.

Coste de los componentes	Coste del personal	Coste Total
2.859,85 €	7.728,50 €	10.588,35 €

Tabla 3: Coste total

6. Conclusiones y futuras mejoras

Podemos concluir que se puede utilizar un método u otro según las necesidades.

En el caso que la información que se quiera preservar sea únicamente la persona, el método de realizar un cropping inicial y posteriormente aplicar RANSAC para detectar y eliminar el suelo corresponderá a la mejor aproximación.

No obstante, si queremos preservar elementos de la sala muy cercanas a las paredes de forma totalmente intacta, tenemos dos opciones. El primer método consiste en aplicar RANSAC de forma iterativa, en cuyo caso deberemos tener en cuenta que habrá un plano que no se detectará correctamente. Por lo tanto, es aconsejable aplicar solamente 3 veces RANSAC y mantener la última pared en la grabación, para evitar la eliminación de clasificaciones erróneas como el caso en que se detecta a la persona como plano. El segundo método consiste en realizar distintos croppings, aplicar RANSAC a cada uno de ellos y juntar los resultados. En este caso tenemos que tener en cuenta que no tendremos una eliminación perfecta de las paredes y del suelo y que seguiremos observando restos en los bordes de las superficies.

Como futuras mejoras podemos considerar aplicar este proyecto directamente con la información captada por los sensores. Hasta el momento se ha aplicado los procesamientos en grabaciones capturadas en la sala, pero sería interesante aplicar directamente sobre la información captada a cada momento por los sensores.

Para finalizar, a continuación se detallaran las distintas competencias genéricas que se han trabajado durante la realización de este trabajo.

G8. Capacidad para identificar, formular y resolver problemas de ingeniería:

Durante la realización de este proyecto ha sido necesario adaptarse a las circunstancias del momento dado que se utilizaba un framework (ROS) nunca visto. El procedimiento de plantear opciones, ejecutarlas y cambiarlas ha sido habitual, hasta llegar finalmente la solución deseada.

G10. Experimentalidad y conocimiento de herramientas e instrumentos:

Todo el proyecto se ha realizado en el grupo de imagen, utilizando sus servidores. Además, para realizar grabaciones con los sensores, realizar procesados en ellos y obtener resultados ha sido necesario conocer y utilizar las herramientas e instrumentos habilitados en la SmartRoom.

Bibliografía:

- [1] A. Casanova Pagà, «Fusion of 3D data for interactive applications,» TFG, TelecomBCN Jul 2016 (<http://hdl.handle.net/2117/89001>)
- [2] E. Torres Romero, «Visualización objetiva de la experiencia de usuario en VR y AR,» TFG, TelecomBCN Jun 2017 (<http://hdl.handle.net/2117/106477>)
- [3] E. Bayó Puxan, «[Visualització objectiva d'una escena de realitat augmentada,» TFG, TelecomBCN Jul 2018 (<http://hdl.handle.net/2117/123497>)
- [4] M. Quigley et al., «ROS: an open-source Robot Operating System», presentat a ICRA Workshop on Open Source Software, 2009
- [5] «CSE481C: Capstone Software - Robotics». [En línia]. Disponible a: <https://courses.cs.washington.edu/courses/cse481c/>.
- [6] «Sistema Operativo Robótico». [En línea]. Disponible a: https://es.wikipedia.org/wiki/Sistema_Operativo_Rob%C3%B3tico
- [7] «Understanding ROS Nodes». [En línea]. Disponible a: https://erlerobotics.gitbooks.io/erlerobot/content/es/ros/tutorials/understanding_ros_nodes.html
- [8] «ROS Publisher and Subscriber Image ». [En línea]. Disponible a: https://www.designnews.com/sites/default/files/Design%20News/ROS_System_Model.jpg
- [9] «ROS/Tutorials - ROS Wiki». [En línea]. Disponible a: <http://wiki.ros.org/ROS/Tutorials>
- [10] «Packages. ROS Filesystem Concepzió». [En línea]. Disponible a: <http://wiki.ros.org/Packages>
- [11] «Rviz (ROS Visualization)». [En línea]. Disponible a: <http://sdk.rethinkrobotics.com/wiki/Rviz>
- [12] «What is PCL». [En línea]. Disponible a: <http://pointclouds.org/>
- [13] «Bags». [En línea]. Disponible a: <http://wiki.ros.org/Bags>
- [14] «RANSAC, Random sample consensus». [En línea]. Disponible a: <https://es.wikipedia.org/wiki/RANSAC>



Glosario

TFG: Trabajo Final de Grado

UPC: Universitat Politècnica de Catalunya.

GPI: Grupo de Procesamiento de Imagen.

3D: Tres dimensiones.

RGBD: Red, Green, Blue and Distance.

ROS: Robotic Operative System.

VR: Virtual Reality.

AR: Augmented Reality.

PCL: Point Cloud Library.

RANSAC: Random Sample Consensus.