



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



VIRTUAL REALITY INTERACTION IN TELEPRESENCE

**Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya**

**Grau en
Enginyeria de Tecnologies i Serveis de Telecomunicació**

Mireia de Gracia Forés

Tutor: Josep Ramon Casas Pla
Barcelona, Gener 2021

Abstract

The content of this document presents a solution to the main objective of the project: the creation of a bidirectional telepresence system.

This system should allow two people to see each other through virtual reality glasses, in real time, as if they were facing each other, in 3D and with consistent geometric proportions, as if they were in the same room. It must also allow the users to move around within the scene.

The different options that have been considered will be displayed, as well as the implementation strategy chosen. The final system is integrated with ROS, Linux, Unity, HTC Vive virtual reality glasses and one computer per user. The reason for this choice is detailed later in this report, but we can advance that it is for technical reasons that allow the continuity of this project.

Finally, some examples of operation with captures made on the corresponding devices are included.

Resum

El contingut d'aquest document presenta una solució al principal objectiu del projecte: la creació d'un sistema bidireccional de telepresència.

Aquest sistema ha de permetre que dues persones es puguin veure mitjançant unes ulleres de realitat virtual, en temps real, una davant de l'altra, en 3D i amb proporcions geomètriques coherents, com si estiguessin a la mateixa sala. També ens ha de permetre la mobilitat dins de l'escena generada.

Es mostraran les diferents opcions que han estat considerades, així com la solució que s'ha escollit. El sistema final està integrat amb ROS, Linux, Unity, ulleres de realitat virtual HTC Vive i un ordinador per usuari. El motiu d'aquesta elecció es detalla més endavant en aquest informe, però podem avançar que és per qüestions tècniques que permeten la continuïtat d'aquest projecte.

Finalment, s'inclouen alguns exemples de funcionament amb captures realitzades als corresponents dispositius.

Resumen

El contenido de este documento presenta una solución al principal objetivo del proyecto: la creación de un sistema bidireccional de telepresencia.

Este sistema debe permitir que dos personas puedan verse mediante unas gafas de realidad virtual, en tiempo real, una delante de la otra, en 3D y con proporciones geométricas coherentes, como si estuvieran en la misma sala. También debe permitir la movilidad dentro de la escena generada.

Se mostrarán las diferentes opciones que han sido consideradas, así como la solución elegida. El sistema final está integrado con ROS, Linux, Unity, gafas de realidad virtual HTC Vive y un ordenador por usuario. El motivo de esta elección se detalla más adelante en este informe, pero podemos adelantar que es por cuestiones técnicas que permiten la continuidad de este proyecto.

Finalmente, se incluyen algunos ejemplos de funcionamiento con capturas realizadas a los correspondientes dispositivos.

A la meva família i amigues, que han estat un suport psicològic molt important al llarg d'aquesta etapa que finalitza amb aquesta tesi de final de grau.
Moltes gràcies iaio, tenies raó, no havia d'afluixar.

Agraïments

Vull expressar el meu sincer agraïment al supervisor Josep Ramon Casas Pla, per la seva bona predisposició des del començament del projecte. Ha estat un plaer poder treballar amb ell i el seu suport i supervisió han estat clau en la finalització amb èxit del sistema. Gràcies per aquesta oportunitat, Josep, he après molt al teu costat.

També a l'Albert Gil Moreno que, tot i que ara és extern a la Smart Room i la UPC, ha ofert la seva ajuda en la resolució de conflictes de software que han anat sortint al llarg del camí. Gràcies Albert.

Historial de revisions i d'aprovació

Revisió	Data	Objectiu
0	29/12/2020	Creació del document
1	12/01/2021	Revisió del document
2	21/01/2021	Revisió del document
3	23/01/2021	Última revisió i aprovació del document

LLISTA DE PARTICIPANTS DEL DOCUMENT

Nom	e-mail
Mireia de Gracia Forés	mireia.de.gracia@estudiantat.upc.edu
Josep Ramon Casas Pla	josep.ramon.casas@upc.edu

Escrit per:		Revisat i aprovat per:	
Data	29/12/2020	Data	23/01/2021
Nom	Mireia de Gracia Forés	Nom	Josep Ramon Casas Pla
Posició	Project Author	Posició	Project Supervisor

Taula de continguts

Abstract	1
Resum	2
Resumen	3
Agraïments	5
Historial de revisions i d'aprovació.....	6
Taula de continguts	7
Llistat de figures:.....	9
Llistat de taules:.....	10
1. Introducció.....	11
1.1. Finalitat del projecte	11
1.2. Requisits i especificacions	12
1.3. Metodologia i procediment	12
1.4. Pla de Treball	13
1.5. Desviacions i incidències	14
2. Estat de l'art de la tecnologia emprada en aquest projecte:.....	15
2.1. ROS (Robot Operating System).....	15
2.2. SteamVR (Steam Virtual Reality)	15
2.3. HTC Vive.....	16
2.4. Unity 3D.....	17
2.5. Kinect.....	17
2.6. Exemples d'investigacions actuals o anteriors	18
3. Metodologia i desenvolupament del projecte:.....	19
3.1. Creació d'un prototip amb un ordinador, un sensor i unes ulleres	20
3.1.1. Connexió amb les ulleres de realitat virtual	21
3.1.2. Plugin de rviz	22
3.1.3. Unity 3D.....	23
3.1.4. OpenVR	26
3.2. Creació del prototip bidireccional.....	26
4. Resultats	28
4.1. Rviz.....	28
4.1.1. Eliminació de fons en rviz	30

4.2.	Unity	32
4.2.2.	Eliminació de fons en Unity	34
4.3.	Llistat del codi que s'adjunta	36
5.	Pressupost	37
6.	Conclusions i desenvolupaments futurs:	38
	Bibliografia:	41
	Apèndix	42
A.1.	Work Packages	42
A.2.	Diagrama Gantt	48
A.3.	Comandes del terminal	50
A.4.	Documentació	51

Llistat de figures:

Il·lustració 1 - Captura de l'usuari mitjançant els sensors, imatge de la pàgina oficial de SteamVR.	16
Il·lustració 2 - Producte HTC Vive	16
Il·lustració 3 - Exemple de núvol de punts, Sala SmartRoom	17
Il·lustració 4 - Dues usuàries amb el respectiu material, cadascuna a la seva sala. Il·lustració de font pròpia.....	19
Il·lustració 5 - Simulació d'imatge estreta del món virtual. Il·lustració de font pròpia.....	19
Il·lustració 6 - Exemple del que veuria la usuària 2 segons el cas anterior. Il·lustració de font pròpia.	20
Il·lustració 7 - Esquema del sistema amb rviz plugin	22
Il·lustració 8 - Esquema del sistema amb Unity.....	23
Il·lustració 9 - Connexions de ROS executant el projecte de Unity	25
Il·lustració 10 - Esquema del sistema bidireccional complet.....	26
Il·lustració 11 - Exemple de captura amb rviz, càmera user1	28
Il·lustració 12 - Exemple de captura amb rviz, càmera user2	29
Il·lustració 13 - Eliminació de fons user1	30
Il·lustració 14 - Eliminació de fons user 2	31
Il·lustració 15 - Exemple de visualització a les ulleres de realitat virtual amb rviz_plugin	32
Il·lustració 16- Exemple de captura de l'espai amb Unity, user2.....	33
Il·lustració 17 - Exemple de captura d'entorn, user1	33
Il·lustració 18 - Eliminació de fons a Unity, càmera user1	34
Il·lustració 19 - Eliminació de fons a Unity, càmera user2.....	35
Il·lustració 20 - Simulació de la visualització a les ulleres HTC Vive	35
Il·lustració 21 - Accions disponibles amb SteamVR dins de Unity.....	36
Il·lustració 22 – Milestones (fet amb el programa Ganttter).....	47
Il·lustració 23 - Gantt (part 1)	48
Il·lustració 24 - Gantt (part 2)	48
Il·lustració 25 - Gantt (part 3)	49
Il·lustració 26 - Exemple documentació SteamVR per Linux	51
Il·lustració 27- Exemple documentació plugin rviz.....	52
Il·lustració 28 - Exemple documentació instal·lació ROS Melodic.....	53

Llistat de taules:

Taula 1 - Taula de requisits	12
Taula 2 - Taula d'especificacions	12
Taula 3 – Costos del projecte	37
Taula 4 - Work Packages.....	46

1. Introducció

Aquest treball de final de grau, té com a propòsit l'exploració i la creació d'un sistema de telepresència 3D bidireccional. La telepresència es podria definir com la "translació" virtual d'un individu d'un espai a un altre mitjançant l'ús de la tecnologia audiovisual. Diem que el moviment és virtual perquè, físicament, la persona no s'ha mogut de la seva sala, però es genera la sensació d'estar a un espai diferent. Quan parlem d'un sistema bidireccional, el que volem és poder transportar dues persones alhora a un mateix espai. És a dir, que els fluxos de dades (imatge i so) d'ambdues convergeixin en un nou espai audiovisual *tecnològic* i, per tant, la comunicació entre elles sigui possible *com si* estiguessin físicament al mateix espai.

Les eines que emprarem per fer el projecte seran sensors Kinect, ROS, SteamVR/Unity/rviz i les ulleres HTC. ROS és un software que permet la creació d'aplicacions distribuïdes per robòtica. En aquest projecte, ROS fa de node d'unió entre tots els elements del sistema de telepresència: és qui captura l'stream de dades dels sensors Kinect2 perquè pugui ser processat i finalment visionat per les ulleres de realitat virtual. Les ulleres en qüestió que utilitzarem són de la casa HTC, i per tal que la informació pugui ser mostrada a les ulleres utilitzarem el motor de renderització SteamVR combinat amb Unity o rviz per tal de visualitzar el contingut a les ulleres o, alternativament, en un entorn 3D d'una pantalla plana.

1.1. Finalitat del projecte

Aquest projecte de final de grau té com a missió principal crear un nou sistema de telepresència 3D bidireccional en realitat virtual, totalment integrat en Linux al laboratori "Smart Room" pel Grup de Processat d'Imatge (GPI) del departament de Teoria del Senyal i Comunicacions (TSC) i per IDEAI-UPC (Intelligent Data Science and Artificial Intelligence Research Center).

La Smart Room és un laboratori dotat de diferents sensors visuals i acústics, així com altres equipaments que permeten l'experimentació en els camps de la realitat virtual, la realitat augmentada, el processat i el mapejat 3D de l'entorn. Per a la creació d'aquest sistema bidireccional es necessiten 2 sensors Kinect v2, dues ulleres HTC Vive i dos ordinadors amb sistema operatiu Linux.

1.2. Requisits i especificacions

S'han definit una sèrie de requisits que un possible usuari podria esperar del sistema a desenvolupar. Es mostren a la taula següent.

Taula de requeriments
Un sistema que permeti la comunicació amb un altre usuari.
Un sistema que permeti veure i interactuar amb l'altra persona.
Un sistema que doni la sensació que els dos usuaris són a la mateixa sala.
Un sistema immersiu en temps real.
Un sistema compatible amb les ulleres de realitat virtual HTC Vive.

Taula 1 - Taula de requisits

Pel que respecta a les especificacions tècniques del sistema bidireccional de telepresència en 3D, el prototip final hauria d'aconseguir el següent:

Taula d'especificacions
Prototip senzill d'aquest primer sistema nou.
Bidireccionalitat en temps real.
Sistema operatiu Linux i un sol ordinador per usuari.
Plataforma ROS i continuïtat del projecte, sense dependre de versions específiques de programari.
Compatibilitat amb HTC Vive.
Minimitzar el "delay" per tal que la sensació d'immersió sigui realista.
Incorporació d'objectes de realitat augmentada.
El sistema ha de respectar la configuració i les proporcions geomètriques de l'escena d'interacció.

Taula 2 - Taula d'especificacions

1.3. Metodologia i procediment

Aquest treball és una continuació de la línia de recerca en el camp de la telepresència a la Smart Room de la UPC. Però cal destacar que aquest projecte comença des de zero sense aprofitar els mòduls de software existents. Els projectes anteriors es basaven en l'obtenció dels punts de la sala, fins arribar a la creació d'un núvol de punts 360° mitjançant el registre dels núvols de punts generats per tres sensors Kinect diferents.

A més de no haver aconseguit mai la bidireccionalitat, objectiu principal del present projecte, tampoc s'havia integrat mai tot en un sol ordinador per usuari. Anteriorment es necessitaven diferents ordinadors/servidors i diferents sistemes operatius: Linux per córrer ROS i la gestió dels streams, i Windows per al renderitzat i la visualització de l'escena (núvol de punts) a les ulleres HTC. Un dels requisits importants és que el sistema de telepresència sigui gestionat per ROS, que únicament està disponible per Linux. Això facilita la interconnexió de tots els elements sense passar per una comunicació per xarxa Linux-Windows. D'aquí l'interès per explorar la possibilitat de renderitzar i visualitzar amb les ulleres només amb Linux i ROS, donat que això permet la integració compacta i flexible de tots els mòduls del sistema.

La bidireccionalitat té un interès molt obvi i és que una comunicació perquè sigui completa necessita un feedback. En la demo antiga es capturava un núvol de punts que podia ser reproduït a les ulleres de realitat virtual però no permetia incorporar un segon usuari, i aquest només tenia l'opció d'observar a través d'una pantalla el mateix que veia l'usuari amb les ulleres HTC Vive.

També, a diferència de projectes anteriors, el sistema permet la incorporació d'objectes de realitat augmentada, a més de la possibilitat de crear atmosferes o sales de reunió virtuals, així com desplaçaments de l'usuari per l'escena. La tècnica utilitzada per eliminar el fons del núvol de punts és una eina de processat que tampoc s'havia utilitzat prèviament en altres versions del projecte de telepresència.

Altres elements que no s'havien fet a la sala i que s'han utilitzat en el projecte actual, són el concepte d'interconnectivitat i filtrat amb Unity3D de l'investigador associat a la Universitat de Manchester, Dr. Inmo Jang; i el plugin de rviz d'André Gilerson, Institute for Mineral Resources Machine Technology, RWTH Aachen University.

1.4. Pla de Treball

A [l'Annex A.1](#) es poden trobar els Work Packages i Milestones detallats, i a continuació el [diagrama Gantt](#) del projecte.

Work Breakdown Structure

WP1: Instal·lació del sistema operatiu necessari per començar a treballar al primer ordinador (Linux GNOME 18.04, ROS Melodic amb llibreries, SteamVR, Unity)

WP2: Aconseguir capturar i visionar el núvol de punts d'un Kinect2 i ROS (PC1).

WP3: Creació del node per visionar les dades amb les ulleres de realitat virtual HTC Vive (PC 1).

WP4: Duplicar la instal·lació del sistema operatiu (PC2)

WP5: Capturar i visionar la captura en 3D amb un Kinect2 i ROS (PC2)

WP6: Creació del node per visionar segones ulleres de realitat virtual. (PC2)

WP7: Creuar el visionat-subscripció de nodelets per tal de crear la bidireccionalitat, creant una xarxa local. Subscriure el nodelet de les HTC Vive 1 al nodelet del núvol de punts de 2 i viceversa.

WP8: Correcció del visionat perquè sigui geomètricament correcte.

WP9: Test i creació de l'script final

WP10: Documentació corresponent al Treball de Final de Grau.

1.5. Desviacions i incidències

La principal desviació respecte el pla original va ser durant les primeres setmanes. Es van estudiar diferents solucions per poder aconseguir la bidireccionalitat del sistema (un dels objectius principals). Com que aquest projecte es un treball d'exploració – el que volem aconseguir, no existeix – aquestes diferències respecte el pla original entren dins del procés de recerca de solucions. Cal contrastar diferents opcions i la viabilitat d'aquestes.

Apart d'aquests impediments més tècnics, la situació pandèmica actual ha propiciat el canvi en l'ordre dels work packages original. S'ha prioritzat deixar un equip completament instal·lat i funcionant amb la visualització del núvol 3D amb les ulleres de realitat virtual. Un cop aconseguit això, enlloc de passar a treballar el núvol de punts amb la transformacions geomètriques necessàries, s'ha procedit a la clonació d'aquest sistema al PC2 (un laptop) per tal de poder avançar feina des de casa i, en cas de confinament, poder seguir treballant.

Les ulleres de realitat virtual 2 (pel PC2) encara no han arribat, i per aquest motiu la simulació final haurà de ser en aquest ordinador via el programa Unity. També s'han hagut de crear nous comptes de correu i usuaris per poder instal·lar en aquesta segona màquina el programari de Unity.

Al llarg del desenvolupament del projecte s'han provat diferents possibles solucions per aconseguir l'objectiu de bidireccionalitat. En aquest sentit, hi ha proves que han requerit molt temps i, en canvi, no seran utilitzades per la versió final del projecte. Però aquest temps extra invertit, també s'ha vist compensat per altres tasques que, a priori, havien d'ocupar més temps i han estat més ràpides de dur a terme.

2. Estat de l'art de la tecnologia emprada en aquest projecte:

A continuació introduiré els conceptes clau per entendre el desenvolupament d'aquest projecte.

2.1. ROS (Robot Operating System)

Robot Operating System (ROS) [1] és un programari que permet el desenvolupament de software per robòtica. El seu funcionament es basa principalment en un node central (`roscore`) que gestiona tots els processos. Cada procés o acció es desenvolupa en un node i genera un tòpic que es pot publicar al node central perquè un altre terminal s'hi pugui subscriure.

Imaginem el cas del sensor 3D, Kinect 2. Un cop està inicialitzat el node central, la captura que fa el sensor és un procés que genera un tòpic (per exemple `kinect2/points`). Aquest tòpic es pot publicar mitjançant un node de publicació (`publisher`). D'aquesta forma, amb un visualitzador ens podem subscriure a aquest tòpic per tal de visionar el núvol de punts.

En aquest projecte, els dos ordinadors comparteixen un mateix node central, ja que ambdós màquines es troben en la mateixa xarxa amb direccions IP estàtiques. Això permet que cada ordinador es pugui subscriure al núvol de punts contigu.

Entre les llibreries que ofereix ROS, cal destacar la llibreria PCL [2] que permet la transformació del núvol de punts (per eliminar el fons, per exemple); la llibreria `iai_kinect`, que permet la connexió del Kinect amb l'ordinador; la llibreria `tf`, que utilitzem per poder aplicar una sèrie de matrius de transformació al núvol de punts (rotació, translació...); i altres eines com `rviz`, el visualitzador per defecte de ROS i `rqt_graph`, una eina que mostra una gràfica amb totes les interconnexions establertes amb ROS i els diferents processos.

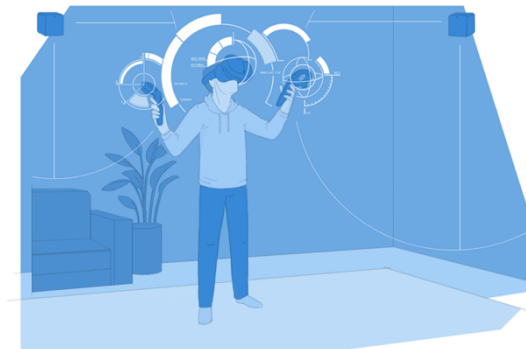
La versió de ROS utilitzada és ROS Melodic i el sistema operatiu en ambdós màquines és Ubuntu 18.04.

2.2. SteamVR (Steam Virtual Reality)

Steam Virtual Reality [3] és una secció del software de videojocs per ordinador Steam que permet l'execució d'experiències immersives en realitat virtual, com per exemple videojocs o mons virtuals que es poden explorar. Per poder utilitzar aquesta tecnologia és imprescindible l'ús d'un equip de realitat virtual, com per exemple HTC Vive.

Gràcies a uns sensors que es col·loquen a la sala, el software de Steam és capaç de determinar l'espai disponible a la nostra habitació i habilita una "zona de joc" per la qual l'usuari es podrà desplaçar. Amb les ulleres i els comandaments que porta el jugador a les

mans, els sensors són capaços de determinar la posició d'aquest en tot moment (*Il·lustració 1*).



Il·lustració 1 - Captura de l'usuari mitjançant els sensors, imatge de la pàgina oficial de SteamVR.

2.3. HTC Vive

HTC Vive [4] és un equipament conformat per unes ulleres de realitat virtual, que es connecten a l'ordinador, i dos comandaments sense fil, un per cada mà. Els comandaments tenen una sèrie de botons que permeten la interacció de l'usuari amb l'escena virtual que està visionant a les ulleres de realitat virtual: desplaçaments, agafar objectes, fer zoom...

S'ha optat per utilitzar aquest model d'ulleres de realitat virtual perquè és compatible, mitjançant SteamVR, amb Linux i aquest era un dels requisits del projecte.



Il·lustració 2 - Producte HTC Vive

2.4. Unity 3D

Unity 3D [5] és un programa per dissenyar principalment videojocs, animacions i experiències immersives en 3D. Actualment és molt utilitzat per crear escenes de realitat virtual, ja que és compatible amb moltes ulleres de VR, a més de tenir suport en un gran nombre de plataformes.

Unity permet la connexió amb ROS mitjançant nodes subscriptors. D'aquesta forma, es pot traslladar el núvol de punts en el seu visor per tal de crear les escenes pertinents a posteriori.

2.5. Kinect

El Kinect [6] és un sensor de Microsoft (ja descatalogat) conformat per una càmera RGB, un sensor de profunditat (infraroig) i un micròfon. Aquest sensor ofereix diferents sortides (càmera de profunditat; núvol de punts hd, sd o qhd; o simplement imatge) que poden ser visionades a posteriori amb visualitzadors com per exemple rviz (el visualitzador de ROS) o d'altres. El model utilitzat és el Kinect v2, amb les següents prestacions tècniques:

- Definició: 1920x1080, 30fps
- FOV: 70° H , 60° V

i permet capturar núvols de punts com el de la *Il·lustració 3*.



Il·lustració 3 - Exemple de núvol de punts, Sala SmartRoom

2.6. Exemples d'investigacions actuals o anteriors

Altres alumnes de la UPC que han treballat anteriorment en el camp de la Telepresència a la smartroom amb el sistema antic unidireccional que emprava Linux per la captura i Windows per al renderitzat a les ulleres són:

- Ivan Moreno [7], que va treballar sobre l'eliminació de fons amb temps real amb eines de PCL (RANSAC),
- Aina Maki [8], que va aplicar eines de *cropping* i RANSAC per a l'eliminació del terra i parets
- Elisabet Bayó [9], que va introduir el concepte de càmera objectiva per a la visualització per terceres persones de l'escena percebuda per l'usuari de les ulleres amb coherència geomètrica
- Eduard Torres [10], que va explorar en el registre entre l'usuari i l'escena percebuda a les ulleres, posant la base per al desenvolupament de la càmera objectiva amb coherència geomètrica
- Arantxa Casanova [11], que va investigar i implementar el registre de tres sensors Kinect per tal de generar una escena 3D de 360° (120°x3) observable des de totes les direccions.

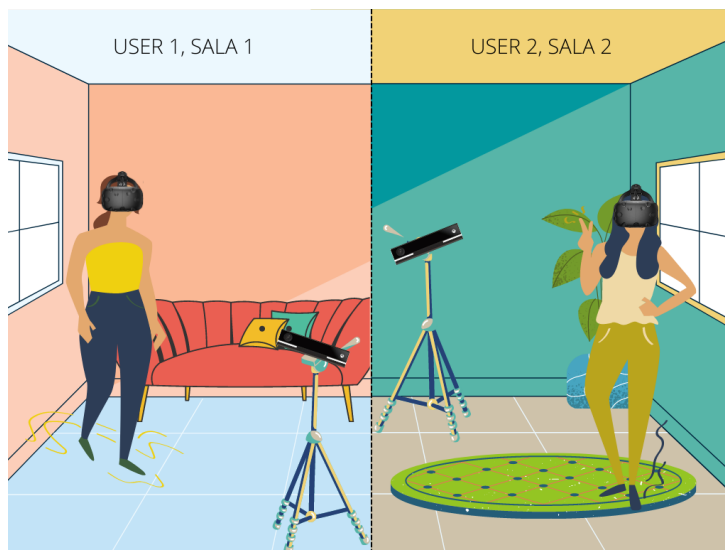
Molts dels projectes de telepresència actuals fan referència una de les primeres definicions de telepresència: l'operació de màquines de forma remota, mitjançant un model de simulació de l'aparell en qüestió, com si ho estiguem fent directament amb les nostres mans. En són exemples: Building a virtual world in ROS based on the robot's perception[12]; ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots[13]. Un exemple de projecte encarat a la telepresència de persones és Holoportation de Microsoft[14], que pretén capturar l'entorn en 3D per tal d'aconseguir una experiència immersiva de l'usuari en un món híbrid (virtual però incorporant objectes reals de l'entorn o persones).

La importància de la integració de tot el sistema en Linux recau en què és un fet indispensable per poder acostar el projecte de telepresència de la Smart Room a altres projectes més grans a nivell de pressupost, com el de Microsoft Holoportation, però amb menys recursos (dos sensors, dos ordinadors i el software necessari), per tant, més senzill i amb menor cost, per tal d'aconseguir una transició més ràpida a l'aplicació pràctica.

3. Metodologia i desenvolupament del projecte:

El que es vol aconseguir és el següent:

Dos usuaris en dues sales diferents (*Il·lustració 4*), cadascú equipat amb un Kinect v2, unes HTC Vive i un ordinador Linux es troben en un espai de reunió virtual visualitzant-se un a l'altre com si fossin a la mateixa sala, o un davant de l'altre. A continuació s'exemplifica el funcionament del sistema objectiu amb una sèrie de gràfics.



Il·lustració 4 - Dues usuàries amb el respectiu material, cadascuna a la seva sala. Il·lustració de font pròpia.

En la *Il·lustració 5* podem observar un exemple d'execució, les dues usuàries es troben en un món virtual. Aquest món virtual pot variar: pot ser una sala de reunions, un espai natural com en el gràfic, o simplement un espai buit.



Il·lustració 5 - Simulació d'imatge extreta del món virtual. Il·lustració de font pròpia.

En aquest projecte no es contempla la creació d'un espai de reunions virtual i es deixa com una possible línia futura del projecte. Les usuàries podran, a més, desplaçar-se per l'escena i interactuar amb els elements d'aquesta. Per tal que les usuàries es puguin veure l'una a l'altra en un mateix espai, es processa i elimina el fons de cada sala i així s'evita que es transmetin en la connexió objectes, respectant així la privadesa o intimitat de l'espai personal de cadascuna.



Il·lustració 6 - Exemple del que veuria la usuària 2 segons el cas anterior. Il·lustració de font pròpia.

Finalment, a la *Il·lustració 6* mostrem un exemple del que veuria la usuària 2 a través de les seves ulleres de realitat virtual. La peculiaritat d'aquest sistema és que no veuria una animació de la persona, sinó la persona real conformada per un núvol de punts, a diferència de tots els sistemes que incorporen avatars en què es mostren gràfics o personatges ficticis que representen una persona. A la imatge clarament no es pot apreciar, però observaria un món en 3D, amb la usuària real. Un aspecte rellevant és la geometria 3D de l'escena que s'aconsegueix amb la combinació de sensors 3D i ulleres de realitat virtual. Quan un usuari s'acosta o s'allunya o es mou al voltant de l'interlocutor, l'observa a través de les ulleres com si realment estigués present davant seu, mantenint la geometria 3D de l'escena, com hem esmentat.

3.1. Creació d'un prototip amb un ordinador, un sensor i unes ulleres

Com hem pogut veure en l'apartat anterior, el sistema que es vol crear és simètric. Per aquest motiu la primera part del treball es centra en la creació d'un sistema més senzill conformat per un Kinect v2, unes ulleres HTC Vive i un ordinador. Hi ha dos grans blocs diferenciables segons la seva utilitat: un mòdul `publisher` de ROS que captura i envia, conformat per Kinect v2 i ROS a l'ordinador; i un segon mòdul que rep aquesta informació i la mostra, conformat per un `subscriber` de ROS i el processat del núvol de punts.

Aquest procés de captura, enviament, publicat i processat no s'havia fet mai amb un sol ordinador, sinó que els mòduls es desenvolupava en servidors diferents. Per tal que la integració en una sola màquina fos possible, es van efectuar una sèrie de tests facilitats per la pròpia pàgina de SteamVR per comprovar si era compatible amb Linux. Després de testejar i instal·lar per primer cop SteamVR en Linux, es va procedir a la compilació de ROS Melodic (versió molt estable pel sistema operatiu Ubuntu 18.04) amb diferents packages necessaris com per exemple la llibreria PCL (habilita el tractament del núvol de punts), `iai_kinect` (permet la connexió del Kinect2 amb ROS), entre d'altres.

La connexió entre Kinect2 i ROS es fa mitjançant un nodelet que envia tot l'stream de dades obtingut amb el sensor cap a la màquina. És a la màquina on podem utilitzar aquesta informació i, per exemple, visualitzar-la amb `rviz` (el visualitzador per defecte de ROS). ROS permet, a més, crear altres nodes de publicació i subscripció als diferents tòpics generats. Per tal de processar el núvol de punts, es crea un `publisher` de la informació que arriba del Kinect2 i un `subscriber` se subscriu a aquest tòpic (li direm `kinect2/qhd/points`). Aquest node pot, per exemple, llençar un script d'una llibreria de ROS perquè executi un codi sobre aquest núvol de punts i faci alguna acció amb ell (rotació, zoom, eliminar punts...).

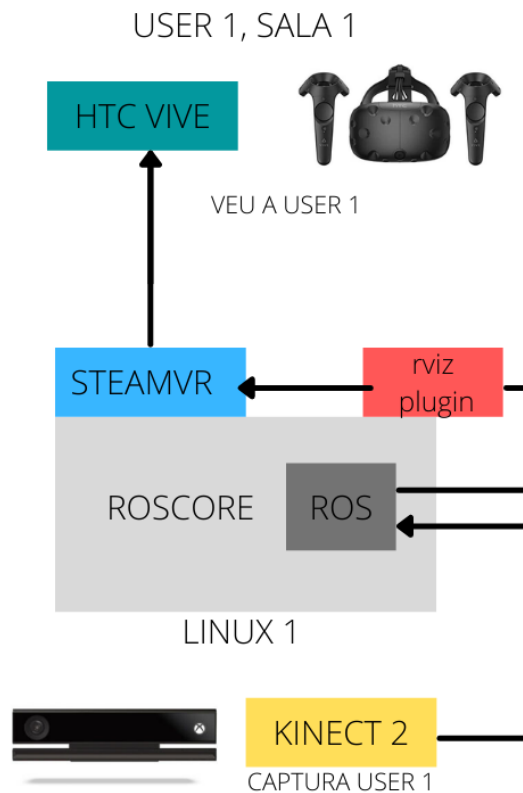
3.1.1. Connexió amb les ulleres de realitat virtual

En aquesta primera versió no eliminem el fons per qüestions de simplificació del procediment, així que un cop tenim la informació de Kinect2 disponible a ROS s'ha de buscar com fer la transmissió del feed a les ulleres de realitat virtual. Això és complicat perquè les ulleres de realitat virtual necessiten que la informació estigui renderitzada per tal d'entendre-ho i poder-ho representar. Per aquest motiu era imprescindible que SteamVR fos compatible i funcional amb Linux, perquè SteamVR fa de vincle directe amb les ulleres HTC Vive.

La dificultat es basa en què la transmissió de dades entre elements gestionats per ROS és senzilla, gràcies a l'ús de nodes que permeten publicar o subscriure als diferents tòpics o dades generades. Kinect2 es comunica amb ROS mitjançant un nodelet. També mitjançant un nodelet és possible la comunicació amb `rviz`, etc. En canvi, si necessitem comunicar-nos amb les ulleres necessitem algun pont que permeti aquesta transmissió, ja que no formen part de ROS, i per tant hem de buscar la forma d'enviar la informació de forma que les ulleres la puguin reproduir.

Després d'una extensa recerca sobre les diferents formes en què es podia establir la connexió amb les ulleres de realitat virtual des de ROS, les opcions disponibles van resultar bàsicament tres: un plugin de `rviz`, Unity 3D o OpenVR.

3.1.2. Plugin de rviz



Il·lustració 7 - Esquema del sistema amb rviz plugin

El visor rviz permet la visualització de diferents tipus de dades obtingudes amb ROS. És un script que permet la subscripció als diferents tòpics que són visibles pel `roscore`, i especificant el tipus de dada que es tracta, es pot arribar a mostrar per pantalla.

La idea d'aquest plugin és original d'André Gilerson [15] de RWTH Aachen University, que va desenvolupar aquesta solució per un projecte de mineria. El que fa aquest plugin és afegir una opció de visionat a rviz. Quan seleccionem aquesta opció, sempre amb SteamVR obert prèviament, la mateixa imatge que tenim a la pantalla pròpia de rviz es duplica a les ulleres.

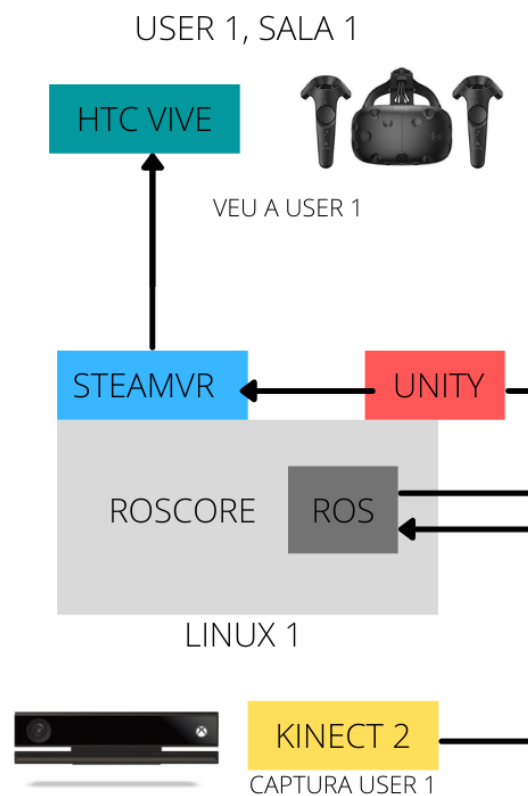
El millor és que el retard que hi ha entre el moviment davant del Kinect2 i el visionat d'aquest a les ulleres de realitat virtual, és mínim, perquè no hi ha cap processat extra i, a més, es tracta d'un sistema conformat íntegrament per eines de ROS.

La problemàtica d'aquesta opció és que no permet el desplaçament per l'escena, actua simplement com una "segona pantalla". Tampoc permetria la futura inserció d'elements de realitat augmentada o la interacció dels usuaris. Un altre punt en contra és que aquest plugin

va ser creat per un projecte universitari i a dia d'avui no té un manteniment assegurat, així doncs, no garanteix una possible continuïtat d'aquest sistema de telepresència bidireccional a la sala.

Tot i que no era exactament el que buscàvem pel nostre projecte, aquesta opció s'ha deixat muntada, s'ha testejat i està funcionant.

3.1.3. Unity 3D



Il·lustració 8 - Esquema del sistema amb Unity

La segona opció era l'ús de Unity 3D per realitzar la unió amb les ulleres HTC Vive. Aquesta opció, a diferència de l'anterior, és molt més atractiva pel fet que Unity 3D és un programa en constant desenvolupament. És una eina cada cop més utilitzada per dissenyadors de videojocs i escenes immersives, a part de ser compatible amb una llarga llista de dispositius. Per aquest motiu, la continuïtat del projecte es veu garantida.

Per contra, sí que és cert que la connexió amb aquest programa és considerablement més complicada, ja que no forma part de ROS i s'ha de renderitzar el núvol de punts un cop s'ha enviat a Unity. A més, la transmissió de dades tant grans com un núvol de punts a Unity3D, pot suposar un augment considerable en el *delay* des de que es produeix el moviment davant del sensor fins que es transmet.

Inspirada en la idea de Inmo Jang [16], la connexió amb Unity 3D des de ROS és possible gràcies a l'ús de la llibreria ROS# i una sèrie *d'assets* de Unity. ROS# permet la comunicació dins de Unity. Recordem que Unity no forma part de ROS i no té implícits els tòpics, per això és important crear un node `publisher` que els faci disponibles.

Per fer possible la connexió de ROS amb aquest programari, el que s'ha fet és crear un projecte de Unity 3D amb 3 objectes principals: `RosConnector`, `PointCloudRenderer` i `PointCloudOrigin`.

Al `RosConnector` tenim dos elements, un en què indiquem la IP de l'ordinador en qüestió que ens envia la informació (el núvol de punts) i el `PointCloudSubscriber`, on hi indiquem el nom del tòpic al que ens volem subscriure (per exemple `kinect2/hd/points`, o el nom del núvol processat). En el cas simple que estem creant, la IP de l'ordinador seria la de la pròpia màquina que estem utilitzant.

`PointCloudRenderer` s'encarrega de crear la imatge. El que fem és subscriure aquest renderitzador al connector que ens transmet el núvol de punts, i definim quina mida volem que tingui el punt en qüestió. Aquest codi crea una matriu amb informació de color i posició per cada punt.

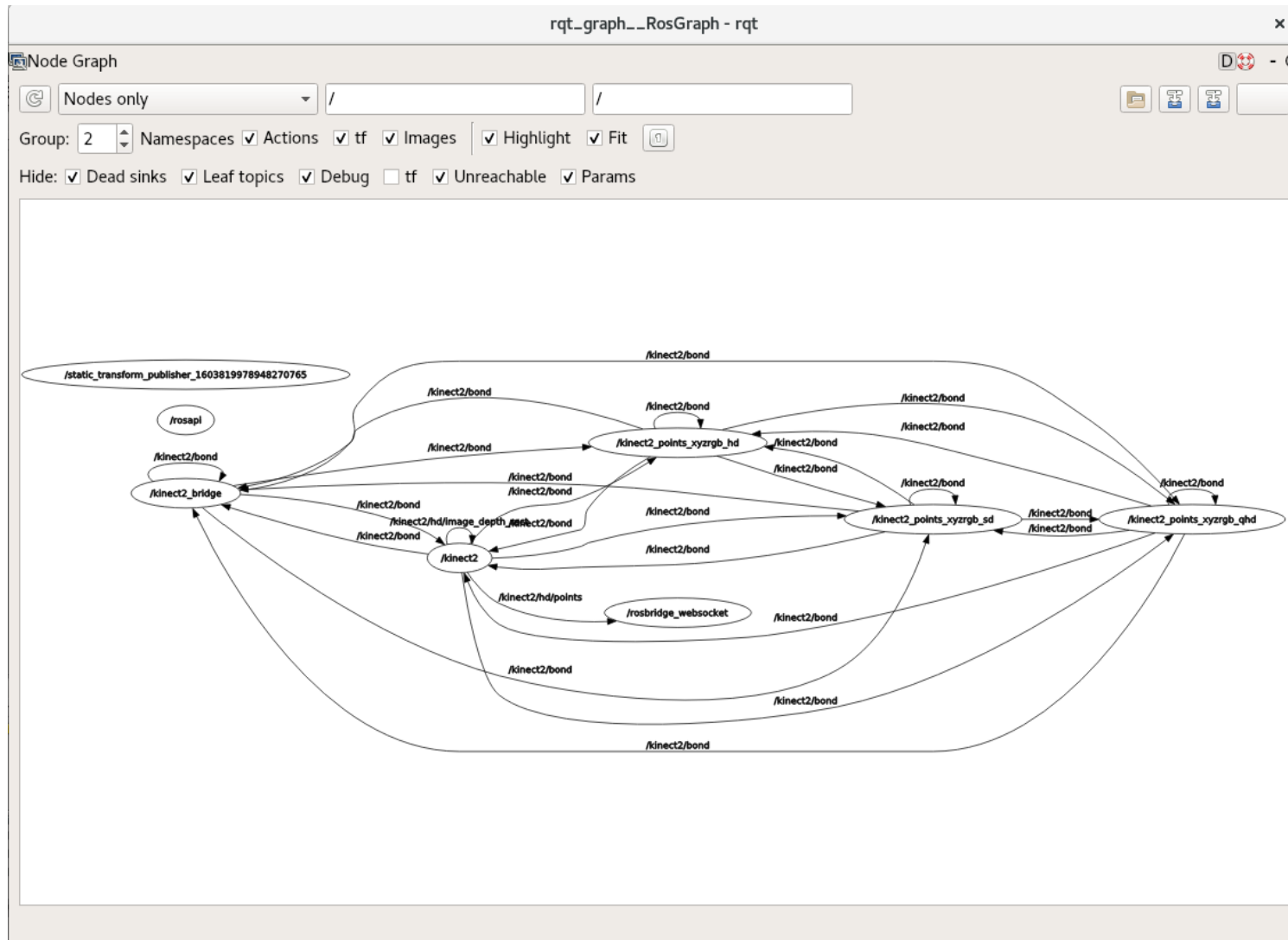
`PointCloudOrigin` ens serveix per poder aplicar les transformacions corresponents al núvol de punts al qual ens hem subscrit.

Un cop tenim això hauríem de ser capaços de visionar el nostre núvol de punts a la pantalla de Unity. Per connectar-ho amb HTC VIVE només cal instal·lar els assets de fàbrica que hi ha disponibles a Unity 3D corresponents a SteamVR.

El gran avantatge d'utilitzar Unity 3D per crear el sistema bidireccional és la versatilitat que ofereix. Es poden crear escenes virtuals, interaccions (desplaçaments, moviments, accions) a més d'inserir objectes de realitat augmentada (i accions amb aquests). Com si es tractés d'un videojoc però amb la persona incorporada a dins.

Un dels inconvenients, degut precisament a la mida dels núvols de punts, és que el temps de retard entre el moviment i la visualització d'aquest en les ulleres de realitat virtual és major que en el cas anterior del plugin de `rviz`. Hi ha una part física amb la qual no hi podem fer res: la velocitat de la xarxa. Per contra, sí que es pot tractar el núvol de punts per tal d'eliminar el fons (per exemple) així com altres tècniques de processat que disminueixen significativament el *delay* en la transmissió d'informació, ja que es transmeten menys punts i per tant la mida d'aquest és menor.

Per tot el que ofereix aquesta tecnologia (la continuïtat del projecte en un futur, el compliment de tots els requisits, i la gran varietat de camps a explorar), aquesta opció ha estat l'escollida per crear el sistema bidireccional.



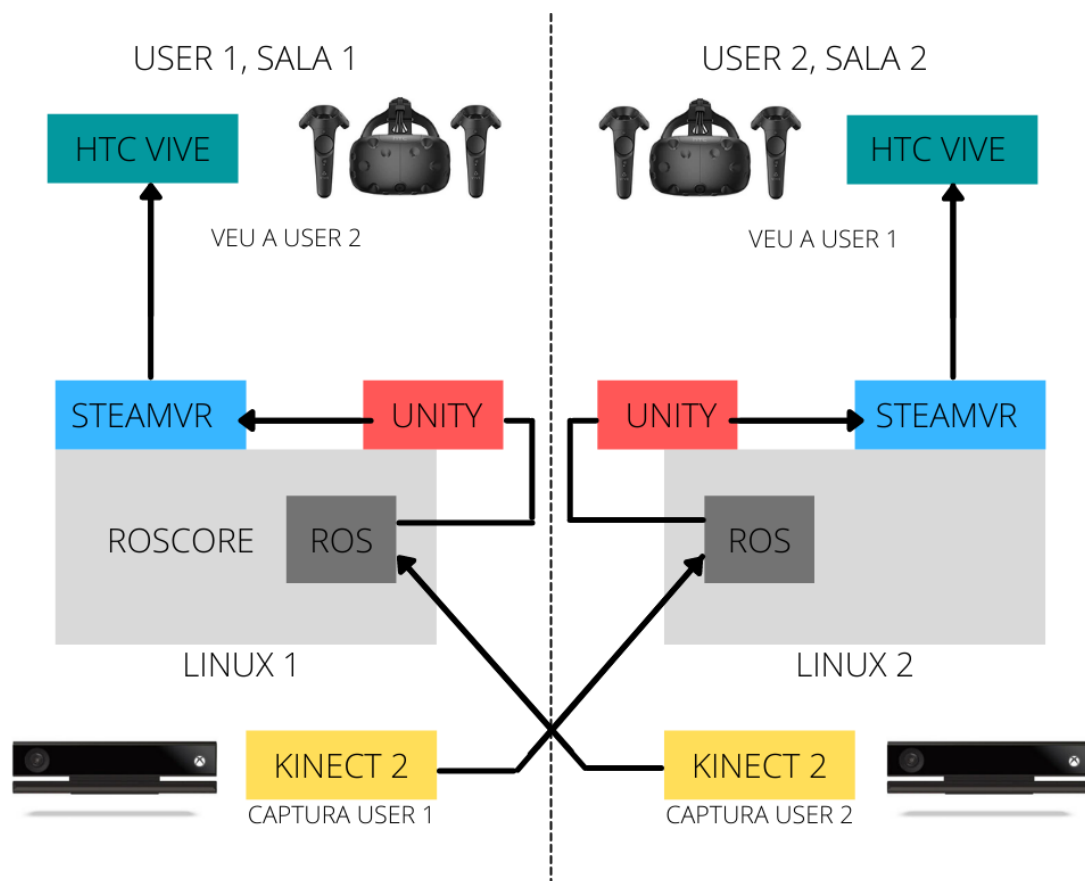
Il·lustració 9 - Connexions de ROS executant el projecte de Unity

3.1.4. OpenVR

Hi ha una tercera opció contemplada, que era l'ús de la llibreria OpenVR conjuntament amb OpenGL. Aquesta opció es va descartar perquè requeria de versions antigues molt específiques de programari i, per tant, seria un pas enrere pel projecte.

3.2. Creació del prototip bidireccional

Com que, com indicàvem, el sistema bidireccional és simètric, un cop aconseguit un prototip funcional de les connexions amb un sistema més senzill s'ha de duplicar el sistema per aconseguir la bidireccionalitat, afegint algunes modificacions, com es mostra a la *Il·lustració 10*.



Il·lustració 10 - Esquema del sistema bidireccional complet

Perquè la comunicació amb ROS entre les màquines sigui possible cal que ambdós ordinadors formin part de la mateixa xarxa i les seves adreces IP siguin estàtiques, per així poder realitzar les referències pertinents en el codi dels nodes subscriptors. El que ens

interessa és que hi hagi un únic `roscore`, un únic “nucli” de ROS, que gestioni tots els tòpics, tant els generats pel primer sensor Kinect2 com pel segon. És a dir, tindrem la informació capturada a les dues sales disponible a ROS. D’aquesta forma podrem crear la informació dels núvols de punts (enviant al segon usuari el del primer, i viceversa) per tal que es puguin veure l’un a l’altre. També perquè els dos ordinadors reconeguin quin és el master de ROS, s’ha de configurar el fitxer `bashrc` d’aquests per especificar l’adreça IP pare.

Un cop s’ha repetit tot el procés 3.1. a la nova màquina, cal efectuar una sèrie de canvis en la configuració. Per exemple, s’han de crear nous scripts que permetin la captura simultània sense conflictes de dos Kinect2, ja que si no es fa això, ROS entra en conflicte al tenir dos tòpics amb el mateix identificador. Per tant s’han de duplicar els `publishers` també.

També, en els respectius projectes de Unity que es troben a cada màquina, per tal que la visualització sigui creuada cal canviar el tòpic pel contigu. És a dir, l’ordinador 1 ha de portar com a tòpic el generat pel Kinect2 de l’altre usuari i al contrari. A més, la IP que s’ha d’especificar al `RosConnector` en ambdós projectes de les màquines és la mateixa, la del ROS Master (en el nostre cas, el PC 1).

Com que les segones ulleres de realitat virtual que es van demanar per aquest projecte no han arribat, el test del sistema bidireccional s’ha dut a terme en un extrem amb el visualitzador del propi Unity.

El que s’observa un cop s’ha aconseguit crear els núvols de punts i la visualització bidireccional és possible, és que el temps de retard entre un moviment de l’usuari 1 i la percepció d’aquest a les ulleres de l’usuari 2, és tant gran que faria impossible la comunicació (s’estima que l’ordre de magnitud d’aquest retard ha de ser de mil·lisegons, i el resultat obtingut donava un retard d’uns quants segons) . Per aquest motiu, arribats a aquests punt és imprescindible plantejar-se diferents eines de processat, com per exemple l’eliminació del fons, per reduir considerablement la mida de les dades a transmetre i , per tant, el temps d’execució.

Per tal d’eliminar el fons m’he basat també en una idea de Imno Jang, [17] que capturava un altre tipus de càmera per eliminar background. La dificultat és que ell no ho feia de forma bidireccional ni amb el Kinect2, per tant s’han hagut de crear nous nodes que s’adaptin al nostre sensor a més de fer possible l’eliminació probabilística simultània en ambdós màquines, sense que ROS entri en conflicte.

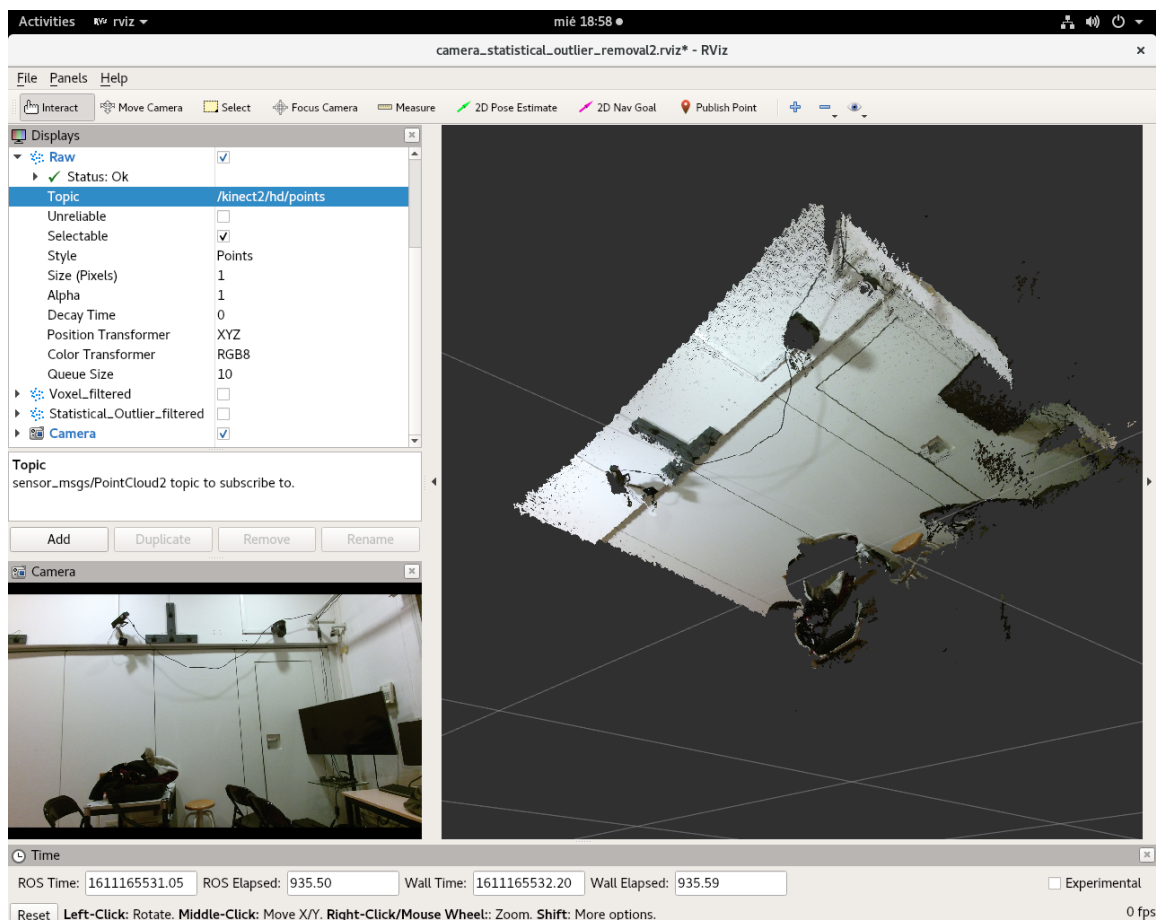
4. Resultats

A continuació es mostren els resultats obtinguts en la nostra exploració en la temàtica de la construcció d'un prototip del nou sistema de telepresència bidireccional en 3D. Començaré amb els resultats corresponents a l'ús de rviz com a solució, incorporant l'eliminació de fons i un exemple de visió amb les ulleres. Acabaré amb una estructura similar però amb el projecte creat de Unity 3D.

4.1. Rviz

A continuació es mostren els resultats de capturar amb Kinect2 i utilitzar el visor rviz de ROS per mostrar els núvols de punts.

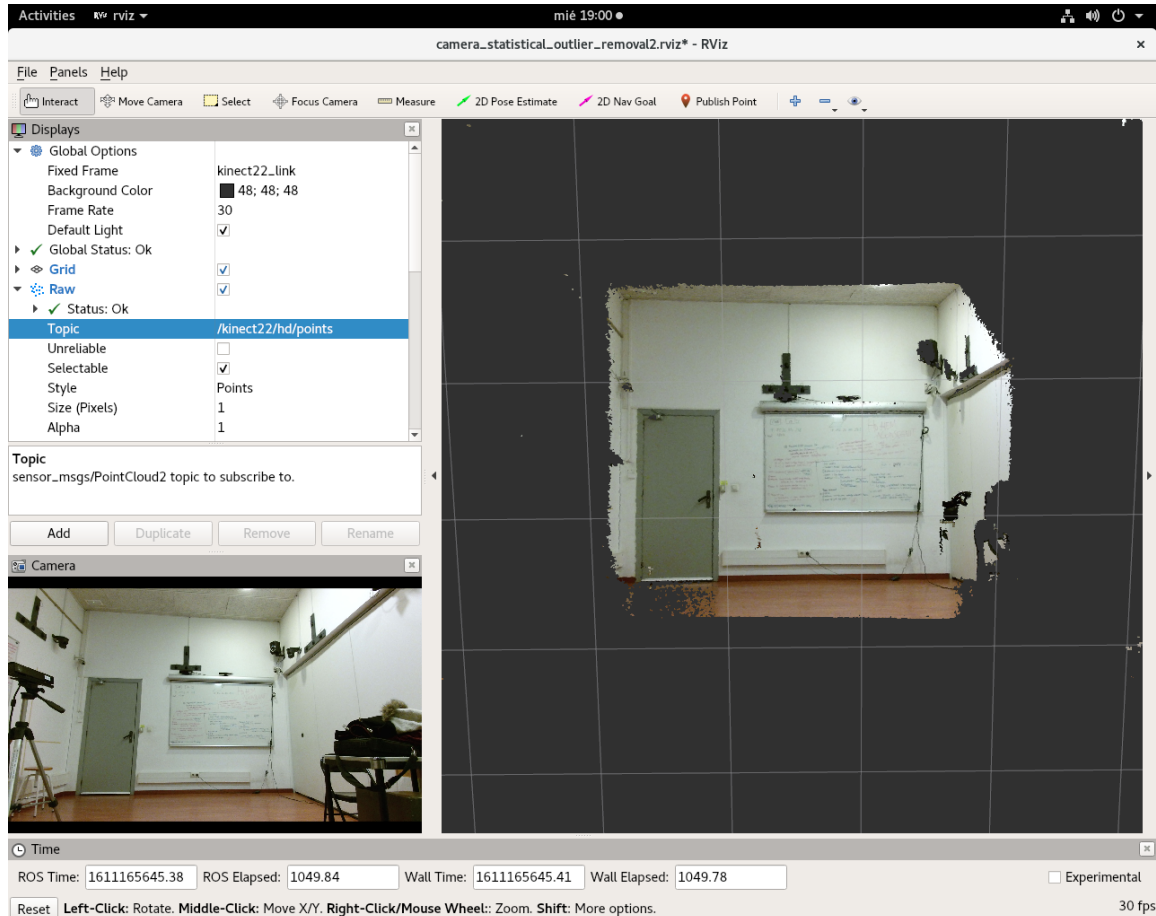
En la *il·lustració 11*, he decidit inclinar el núvol de punts per tal que es pugui apreciar millor la captura de profunditat. A la meitat dreta tenim el núvol de punts corresponents a l'ordinador 1 (user 1). Es pot observar que la part corresponent a la taula i cadires està sortint cap a fora. Per aquest motiu hi ha ombres al núvol de punts, perquè els punts es troben més endavant i el propi objecte fa ombra respecte l'òptica del sensor. Per poder comparar, a l'extrem esquerre hi ha la càmera pròpia del sensor Kinect2.



Il·lustració 11 - Exemple de captura amb rviz, càmera user1

A continuació hi ha un exemple similar però amb el sensor del segon ordinador (user 2). Com en el cas anterior, en l'extrem esquerre podem observar la pròpia càmera del sensor i a la meitat dreta el núvol de punts representat.

En ambdós casos podem apreciar que el camp visual de la càmera és significativament més ampli horitzontalment que en el cas del sensor de profunditat.



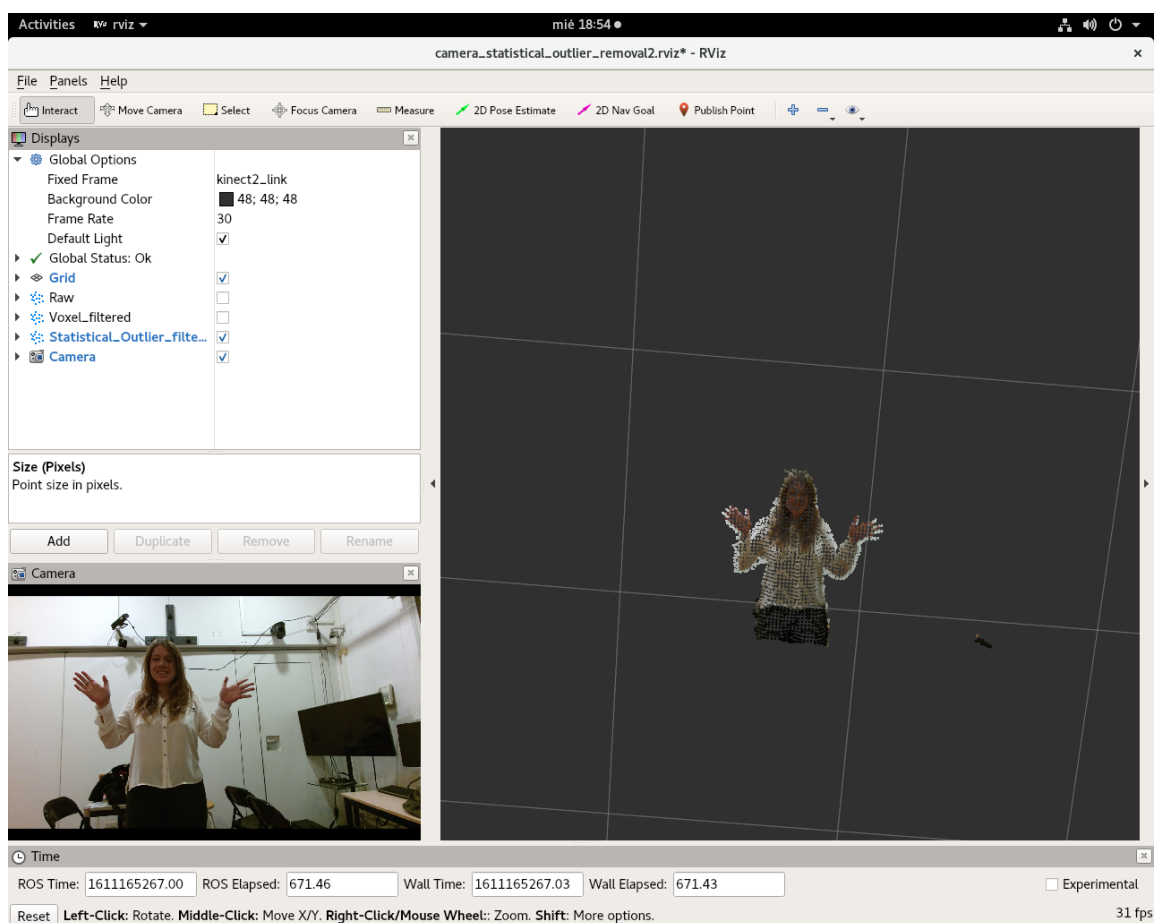
Il·lustració 12 - Exemple de captura amb rviz, càmera user2

4.1.1. Eliminació de fons en rviz

Com que el sistema de visualització amb rviz estava implementat, també se li ha aplicat la eliminació de fons probabilística, a més d'un filtrat per vòxels.

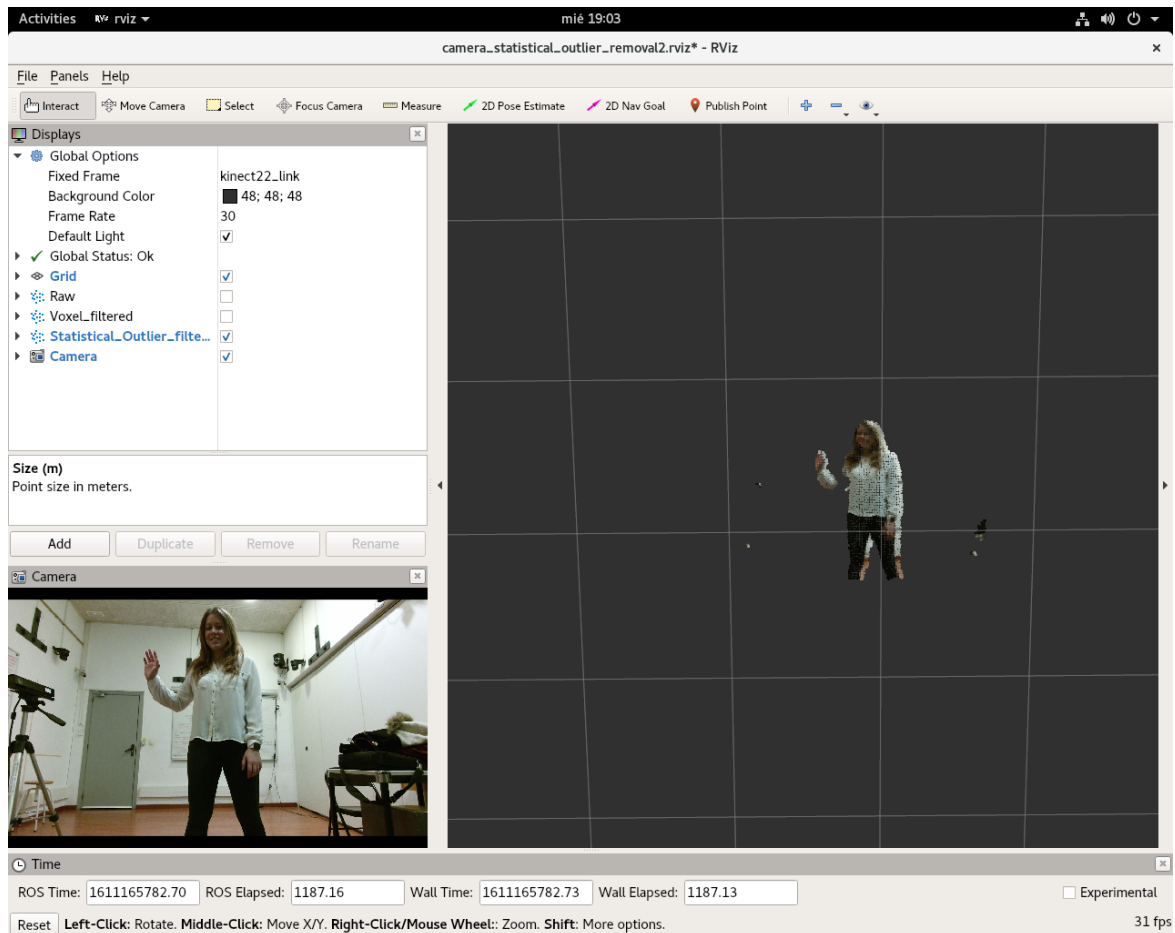
Aquesta tècnica empra un processat basat amb supervoxels (`voxel_grid filtering` de la llibreria PCL). Aquells voxels en els quals no hi ha suficients punts, són eliminats i els altres es conserven en forma de supervoxels. Això implica que enlloc de treballar amb 300.000 punts, per exemple, treballem amb 50.000, reduint considerablement la mida del núvol resultant. Sobre els supervoxels resultants es calculen estadístiques de profunditat i moviment, per fer una eliminació de fons (background subtraction) [18] amb un llindar amb desviació acceptable. D'aquesta forma, s'eliminen tots els supervoxels que no compleixen el requisits, considerant que formen part del fons.

El millor d'aquesta tècnica és que, indiferentment de la posició de la càmera, pot analitzar el fons i eliminar-lo. De fet, si durant la transmissió es mou la càmera, també s'elimina. Això permet que, tot i que l'usuari s'estigui movent molt, no apareixen vestigis no corresponents a la persona.



Il·lustració 13 - Eliminació de fons user1

Com es pot veure la separació i eliminació del fons no és perfecta, ja que resten alguns punts sense eliminar, sobretot al voltant de la persona, que podrien pertànyer al fons (degut a la poca precisió espacial de la tècnica basada en supervoxels). Aquesta tècnica de tractament del núvol de punts no s'havia implementat abans en cap TFG previ. Per això, es podria millorar en un futur treball de final de grau.

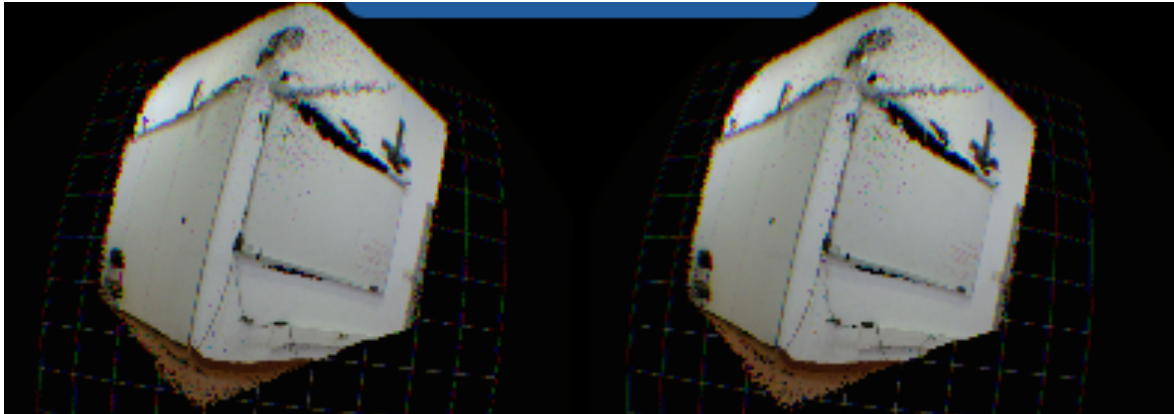


Il·lustració 14 - Eliminació de fons user 2

Podem observar en aquest cas que hi ha alguns punts corresponents a elements molt pròxims al sensor i no s'eliminen (els objectes que surten de la taula). El Kinect2 té una peculiaritat, i és que al no ser un sensor extremadament precís, necessita una distància mínima per tal de poder distingir la distància dels objectes de forma correcta.

Una de les solucions que s'han estudiat és l'ús del plugin desenvolupat per André Gilerson [15]. Aquesta opció no permet el desplaçament pel núvol de punts o espai, simplement la visualització.

El resultat de visionar un núvol de punts a les ulleres de realitat virtual, seria el següent:

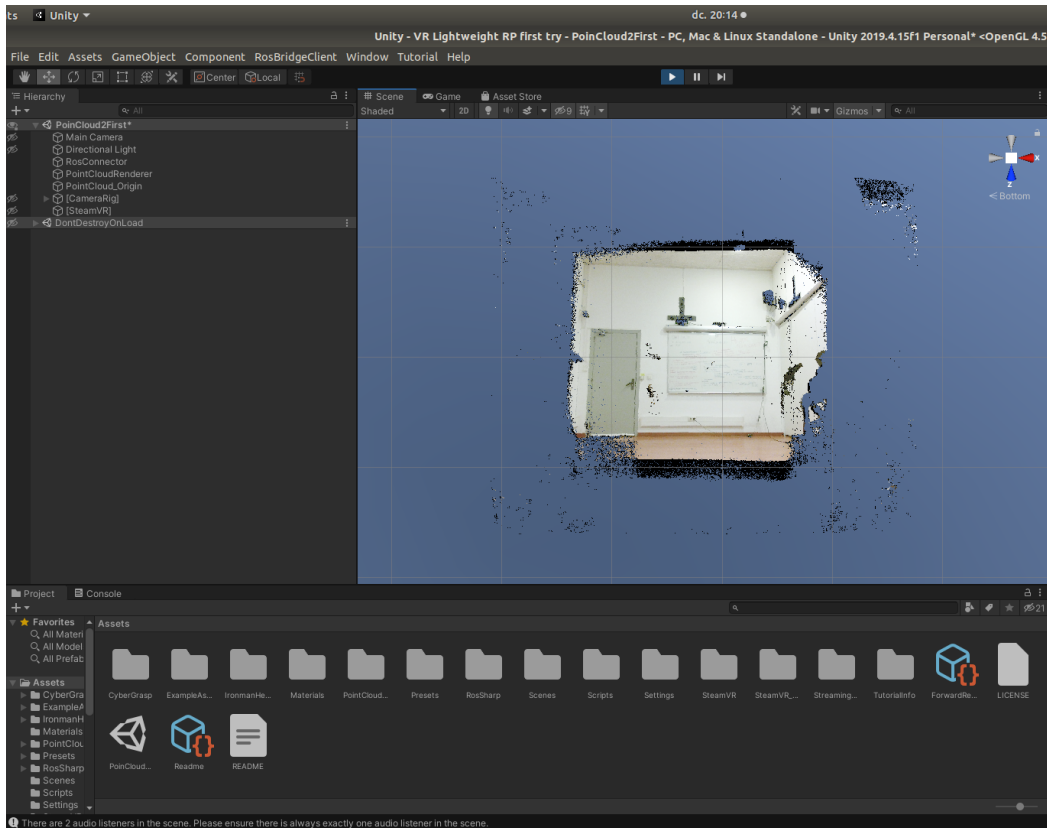


Il·lustració 15 - Exemple de visualització a les ulleres de realitat virtual amb rviz_plugin

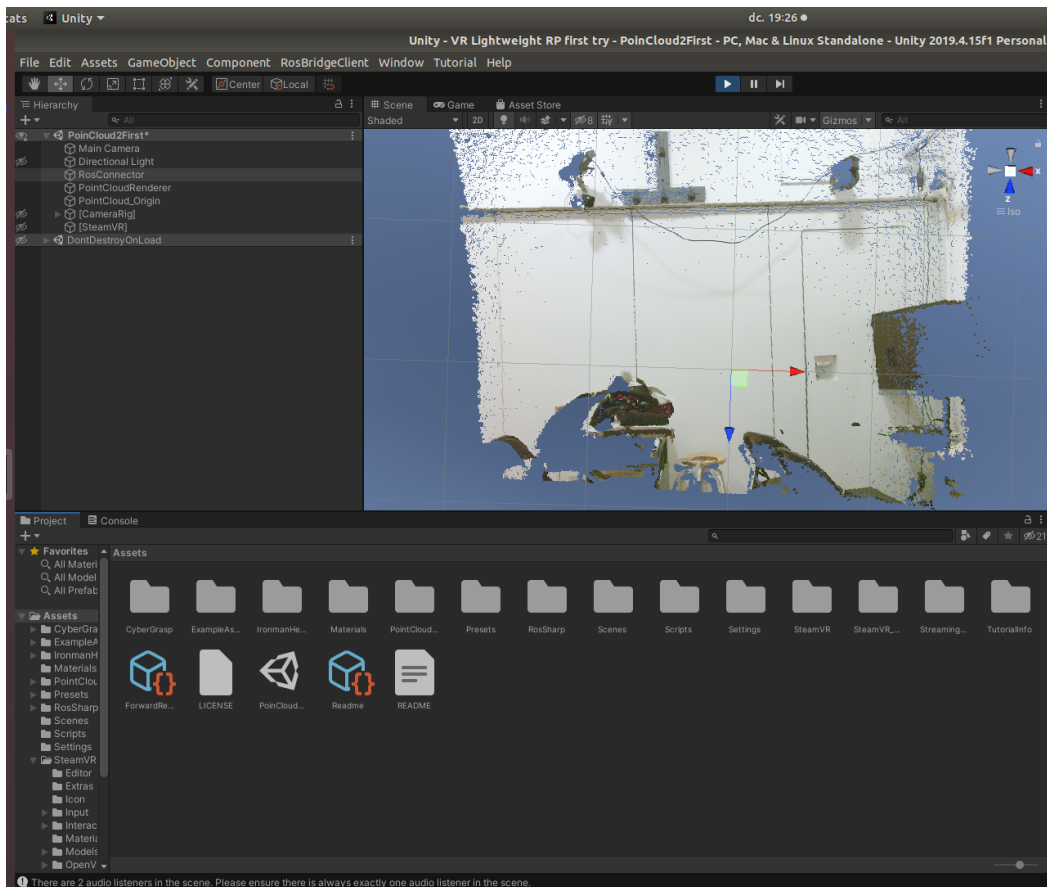
4.2. Unity

La segona opció considerada, i definitiva, per crear el sistema bidireccional és Unity. A continuació es mostra la representació del núvol de punts capturat pels dos Kinect2. Es pot apreciar que hi ha una diferència en quant a qualitat del núvol de punts respecte el visualitzador rviz. Això es deu al fet que rviz porta un renderitzador incorporat, adaptat al tipus de dades que li entren.

En canvi, en el projecte de Unity 3D no hi ha res, i s'ha de crear tot de zero, inclòs el renderitzador. Una possible línia futura seria buscar la forma d'optimitzar aquesta funcionalitat per tal que la qualitat del núvol de punts aconseguit sigui el més fidel possible al generat.



Il·lustració 16- Exemple de captura de l'espai amb Unity, user2



Il·lustració 17 - Exemple de captura d'entorn, user1

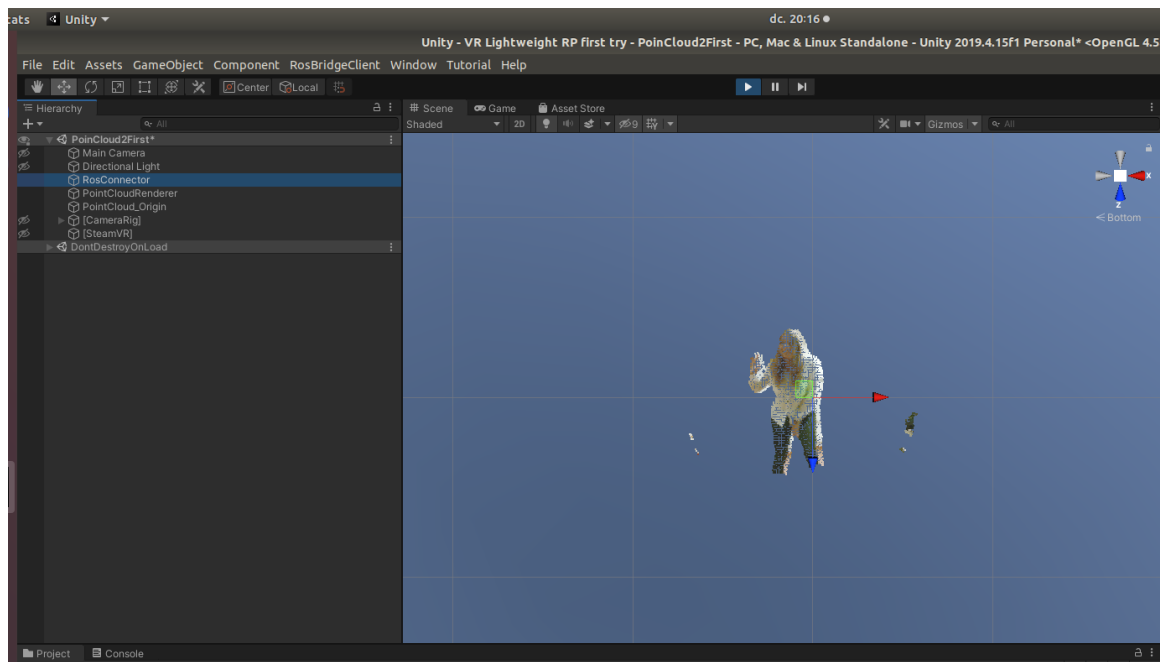
4.2.2. Eliminació de fons en Unity

A més, una altra gran diferència entre l'opció Rviz (4.1.) i l'actual de Unity és que el temps de retard que hi ha entre que es produeix un moviment davant del sensor i aquest es pot visualitzar al projecte de Unity pertinent, és molt més elevat. Estem parlant que en el cas de rviz no hi ha pràcticament delay i en Unity, si transmetem tot un núvol de punts sense filtrar, aquesta diferència pot arribar a ser de 6 segons. Això es deu al fet que un núvol de punts és realment un paquet de dades molt gran i, al no ser una eina pròpiament intriguada en ROS, la transmissió que s'ha de dur a terme cap a aquesta plataforma ralentitza el procés.

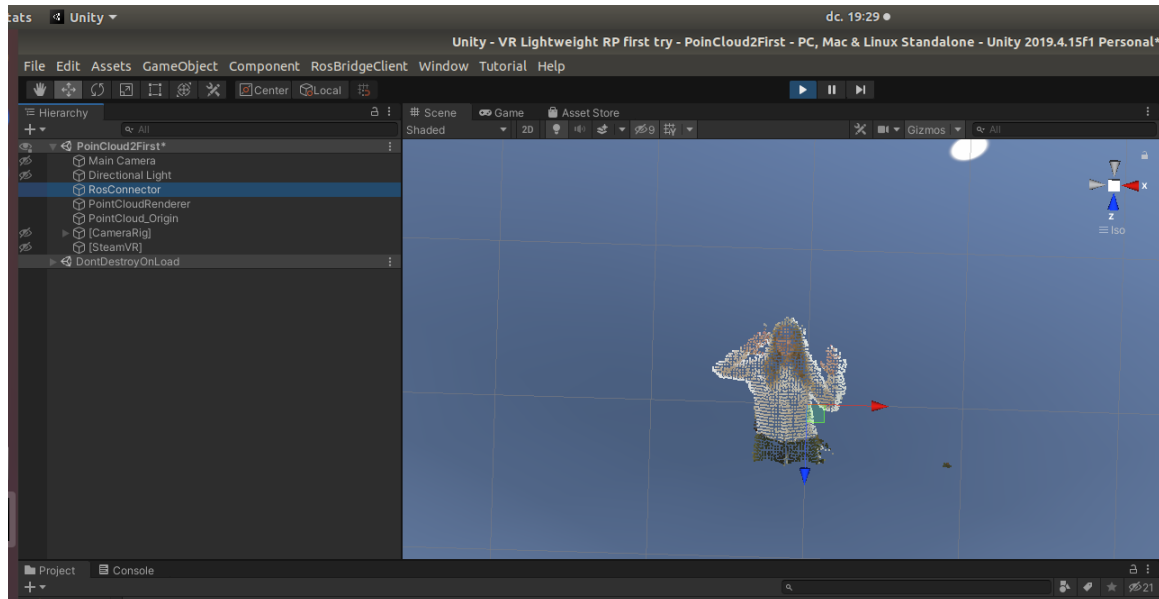
Va ser en aquest punt que es va decidir investigar com millorava aquest temps de diferència entre moviment i visualització si enlloc de transmetre tot el núvol de punts aquest es filtrava prèviament. Per una comunicació bidireccional, no té massa sentit transmetre el fons de la sala, per exemple, i es poden aplicar altres mètodes de filtrat com el que ofereix la llibreria de pcl ROS (`voxel_grid filtering`) per tal de reduir el nombre de punts que conformen la figura.

Aplicant el filtrat probabilístic del fons i el filtrat previ per vòxels, s'ha comprovat que no hi ha diferència de performance entre el visor rviz i el projecte Unity, pel que respecta al temps fins mostrar un moviment.

Aquesta opció d'integrar Unity, a més de les eines de filtrat, la creació del sistema bidireccional, o l'ús íntegre de Linux, són noves d'aquest TFG i no s'havien utilitzat prèviament en altres treballs en el grup d'Imatge.



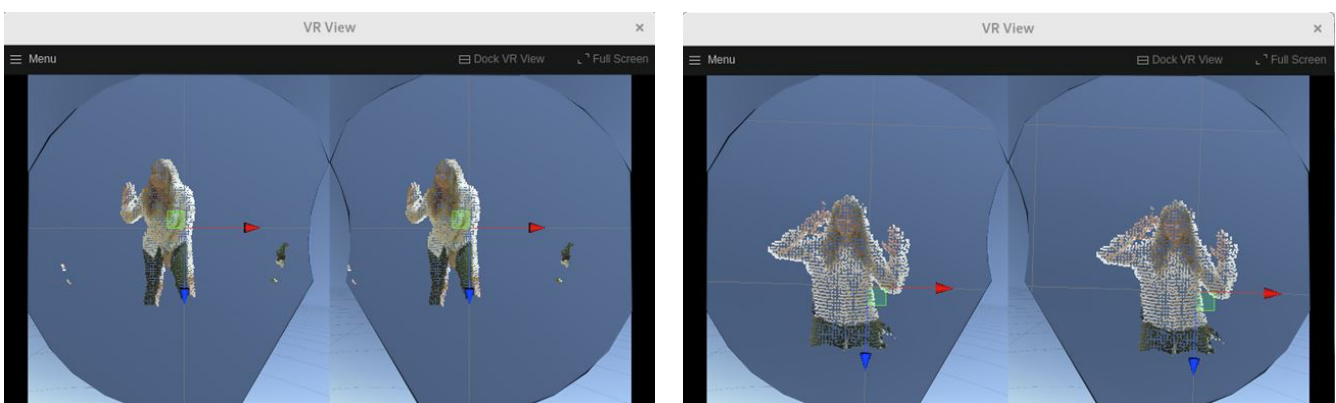
Il·lustració 18 - Eliminació de fons a Unity, càmera user1



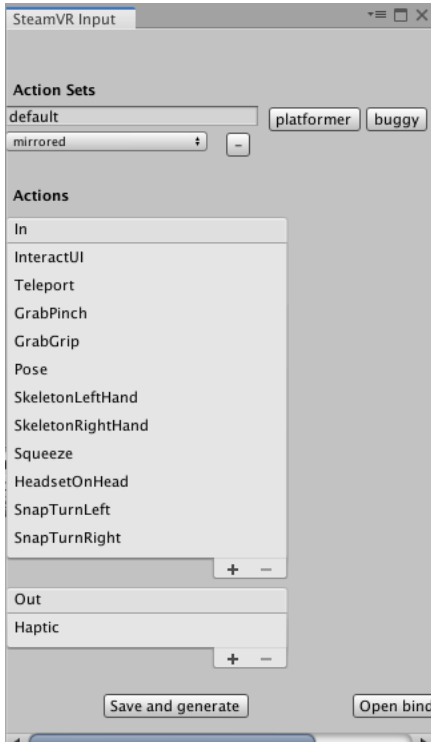
Il·lustració 19 - Eliminació de fons a Unity, càmera user2

Els espais que es poden apreciar entre els punts que conformen la meua imatge filtrada en la imatge superior es deuen al filtrat per vòxels. Segurament millorant aquesta eina de filtrat es podria optimitzar el sistema perquè no apareguessin buits entre els punts i fos una imatge molt més contínua. La optimització del filtrat no era un dels objectius d'aquest treball i, un cop s'ha obert aquest nou front en quant a tècniques de processat, queda pendent per un proper treball la possibilitat de millorar aquesta metodologia.

Com s'ha comentat prèviament en aquest document, les ulleres de realitat virtual que es van demanar per aquest projecte no han arribat a temps. Per aquest motiu s'adjunta una imatge del que podria ser la simulació de visualització que tindria un usuari connectat al projecte de Unity 3D. El fons que ara es mostra blau (per defecte a Unity) es podria modificar o crear-ne un de virtual per tal que l'usuari es desplaci per un espai en concret.



Il·lustració 20 - Simulació de la visualització a les ulleres HTC Vive



Il·lustració 21 - Accions disponibles amb SteamVR dins de Unity

A més de tot el que s'ha esmentat, el projecte que s'ha creat està preparat perquè es puguin desenvolupar diferents accions d'interacció entre els usuaris de forma senzilla amb Unity i SteamVR. La creació de les accions específiques no es contemplava en el marc d'objectius del treball i és una proposta possible de cara a futurs treballs.

Gran part de la documentació sobre els processos d'instal·lació dels diferents software necessaris per aquest projecte s'ha creat a la wiki del grup d'Imatge: imatge.upc.edu. I n'adjuntem les captures en [l'Annex A.4.](#)

4.3. Llistat del codi que s'adjunta

A la carpeta zip que s'adjunta amb aquest document es pot trobar:

Carpeta `catkin_ws`: espai de compilació dels packages de ROS. Aquest inclou:

- `perception_pcl`: conté tots els scripts creats necessaris per poder realitzar l'eliminació de fons de forma bidireccional, és a dir, simultàniament en els dos núvols de punts i en temps real.
- `rviz_vive`: el codi i fitxers de configuració que fan possible la visualització del núvol de punts a les ulleres HTC Vive des de rviz.
- `iai_kinect2`: conté els arxius que fan possible la comunicació entre els sensors Kinect2 i ROS.
- `geometry`: permet aplicar transformacions geomètriques al núvol de punts.

Carpeta Unity: el projecte necessari per reproduir aquest prototip, amb els assets i scripts necessaris per captar el núvol de punts. Només s'adjunta un projecte perquè en ambdós màquines és idèntic, modificant el tòpic al que se subscriu el `RosConnector`.

Carpeta `steamvr_lib_plug`: una selecció de les llibreries de SteamVR necessàries per córrer el plugin de rviz que permet la visualització a les ulleres de realitat virtual.

També a [l'Annex A.3.](#) hi ha disponibles les comandes a executar per tal d'executar aquest sistema bidireccional.

5. Pressupost

Per aquest projecte es van demanar unes ulleres extra HTC Vive que no han arribat, així que es considerarà només el material utilitzat. La resta de hardware ha estat utilitzat durant un període de 5 mesos i es podrà utilitzar durant els pròxims projectes. Per aquest motiu, el cost d'aquests serà l'amortització proporcional a aquests mesos.

Tant el sensor Kinect, els ordinadors i les ulleres tenen una vida útil de 3 anys per l'ús intensiu en el camp d'investigació. Per facilitar els càlculs, calcularem l'amortització mensual.

Es pot estimar que el temps destinat a fer aquest TFG és equivalent a 4 hores diàries, mitja jornada, 5 dies a la setmana des del Setembre de 2020 fins al Gener de 2021, ambdós inclosos. Això suposa un total de 80 hores mensuals. Les hores de feina d'una enginyera júnior tenen un cost aproximat de 9 euros l'hora, indiferentment de les tasques realitzades. D'altra banda, les hores destinades pel tutor tenen un cost aproximat de 25€ l'hora. Considerant que hem fet unes 15 reunions d'una hora i mitja de durada, són 22,5 hores dedicades a la revisió d'aquest treball.

El programari i llibreries que he utilitzat (ROS, Unity3D, Linux, SteamVR) és de llicència lliure i, per tant, el seu cost és 0€.

Components	Cost mensual	Cost total
Kinect 2 v2 (2 unitats)	$(250*2)/36 = 13,89€$	69,45€
Laptop Gaming	$1.899/36 = 52,75€$	263,75€
HTC Vive	$700/36 = 19,45€$	97,25€
PC Torre	$1.300/36 = 36,12€$	180,6€
Enginyera júnior en mitja jornada	$9*80 = 720€$	3600€
ROS, Unity3D, Linux, SteamVR, llibreries	0€	0€
Supervisió del tutor del TFG	No aplica	565,5€
Total		4776,14€

Taula 3 – Costos del projecte

6. Conclusions i desenvolupaments futurs:

Resumim aquí el treball que s'ha dut a terme al llarg d'aquest quadrimestre, seguint la línia d'investigació de la telepresència, integrat únicament amb Linux i gestionat amb el software de robòtica ROS. El que s'ha aconseguit és dissenyar i implementar un sistema en què dos usuaris en dues sales diferents es puguin veure mitjançant unes ulleres de realitat virtual un davant de l'altre com si es trobessin físicament a la mateixa sala, respectant les proporcions geomètriques. Per fer-ho s'ha utilitzat un ordinador Linux, unes ulleres HTC Vive i un sensor Kinect 2 per usuari. Després d'haver discutit diferents opcions per poder dur a terme la connexió amb les ulleres de realitat virtual, s'ha optat per l'ús de Unity 3D per la seva versatilitat, a més d'obrir moltes línies futures que garanteixen la continuïtat d'aquest projecte.

Un cop s'ha exposat la metodologia i els resultats del treball que s'ha dut a terme, es pot concloure que s'han assolit tots els objectius marcats :

- S'ha creat, des de zero, un primer prototip d'aquest nou sistema bidireccional de telepresència en 3D, com es pot observar en els diferents resultats mostrats a l'apartat 4 d'aquest document
- S'ha aconseguit un sistema integrat únicament en Linux, com s'especificava en la descripció del TFG.
- El sistema en qüestió, s'ha desenvolupat amb ROS i no depèn de versions específiques de software, facilitant així la continuïtat del projecte de Telepresència. Unity és un software en continu desenvolupament, així com SteamVR.
- El sistema és compatible amb HTC Vive, com s'ha pogut observar per exemple en l'apartat 4.1.
- El sistema no té *delay* entre el moviment i la transmissió d'aquest, com consta en l'apartat 4.2. d'aquest document i, per tant, la sensació d'immersió és completament realista.
- El sistema creat permet la incorporació d'objectes amb realitat virtual (Unity 3D).
- El sistema respecta les proporcions geomètriques.

La principal preocupació a l'hora d'utilitzar Unity era que, apart que no s'havia utilitzat prèviament en el projecte de Telepresència, es desconeixia fins a quin punt el *delay* podia afectar la comunicació. Ha quedat demostrat que amb eines de processat del núvol de punts aquest punt conflictiu desapareix i, per tant, l'ús d'aquesta tecnologia ha estat una decisió encertada. Addicionalment Unity permet la incorporació nativa d'objectes de realitat virtual 3D i amb animacions.

La primera conclusió arrel d'aquest projecte és que un sistema bidireccional de telepresència podria ser molt útil en una gran varietat de situacions socials i compleix amb moltes necessitats que tenim avui en dia. Després de la situació epidemiològica que hem viscut, hagués permès la proximitat de les persones amb les seves famílies, respectant les mesures sanitàries de seguretat, establint una connexió que clarament es perd amb les

videotrucades, ja que no es conserva la geometria i no tenim la sensació d'estar realment a la mateixa sala. Són molts els casos de gent que s'ha sentit sola durant el confinament per l'aïllament social que això comporta, així com les seqüeles emocionals i psicològiques que això ha suposat. També podria ser interessant per reunions laborals en un entorn virtual d'una sala de reunions. Primerament, perquè no es transmetria el fons de les cases de cada treballador i amb la geometria 3D del sistema podríem mantenir el contacte visual natural (si un/a interlocutor/a mira un altre/a interlocutor/a, aquest/a segon/a veu que l'està mirant a ell/a). Aquesta conservació de la geometria permet, doncs, als interlocutors de sistemes virtuals per a reunions, ser conscients de si les persones ens estan mirant quan parlem, com passa en una conversa normal i, per tant, si estem connectant o ens estem comunicant eficientment amb la resta de persones reunides. Més enllà de tot això, s'evitarien molts viatges innecessaris amb avions i per tant l'empremta de carboni darrere dels desplaçaments a llarga distància, a més de la separació de la família que comporta per a les treballadores que viatgen.

Més enllà de les possibles conclusions d'utilitat, cal destacar que Microsoft ha deixat de fabricar aquests sensors Kinect perquè va suposar per l'empresa una pèrdua milionària: no es van vendre tant com a perifèrics de la consola Xbox per a la que es van dissenyar, sinó que la principal demanda va ser dels laboratoris de recerca, que no els adquirien junt amb la consola. En aquest sentit, podria donar la sensació que no s'està apostant pels núvols de punts. Hi ha diferents motius pels quals això pot estar succeint:

- Hi ha d'altres fabricants (Assus, Occipital/Structure, RealSense, Orbec, PMD...) que han pres el relleu en els sensors.
- El tipus de dades que en resulten (els núvols de punts) són molt diferents a altres dades audiovisuals com la imatge, el vídeo i el so, donat que els núvols de punts són dades no denses (sparse) i no estan tan estudiades en termes d'eficiència de processament, i compressió per a la transmissió.
- El volum de dades que s'ha de transmetre amb cada núvol de punts és realment gran, molt superior a qualsevol altra multimèdia utilitzada correntment i, per això, es necessita una connexió d'internet altament veloç, així com targetes gràfiques potents per processar el resultat
- Precisament pel motiu que es necessiten targetes gràfiques potents i/o eines de computació avançades, és més difícil que un usuari normal faci ús d'aquest tipus de dades amb les seves eines domèstiques, i això propicia que no sigui investigat
- Per contra, és curiós que la mateixa NASA estigui utilitzant sensors LIDAR en els seus prototips, que creen núvols de punts amb molta més precisió mitjançant radar, per obtenir informació, per exemple, de Mart, i així poder recórrer el planeta a posteriori des de la Terra amb unes ulleres de realitat virtual.

Amb això el que vull expressar és que, pot semblar que els núvols de punts avui en dia no siguin una temàtica amb massa importància entre els usuaris, però els investigadors estan desenvolupant eines realment interessants, com la manipulació a distància de robots, o el que acabem de comentar de la NASA i, en aquest sentit, sí crec que els núvols de punts

és un camp que mereix seguir sent investigat, per totes les possibilitats que ens ofereixen. Tenir informació de profunditat podria ser de gran ús en molts casos.

Pel que respecta a ROS, ha demostrat que és una eina molt potent que permet la gestió en robòtica distribuïda. Es podria considerar un propi sistema operatiu que facilita la coordinació dels diferents elements que pugui haver en un sistema. Pel gran potencial que presenta, investigadors arreu del món estan desenvolupant software i això ens fa pensar que seguirà creixent i guanyant noves funcionalitats o compatibilitats amb altre programari, com ha passat per exemple amb la connexió de Unity3D i ROS mitjançant ROS#.

El projecte de telepresència sempre permet millores o noves versions, ja que és una tecnologia canviant i que avança any rere any. En aquest sentit es podria intentar incorporar la idea de la demo anterior que permetia la captura de tres sensors simultanis en la comunicació bidireccional. De fet, el sistema creat en aquest projecte permetria que en un extrem hi hagués aquesta captura efectuada per tres sensors. El que caldria és simplement afegir l'ordinador que controla aquesta captura a la mateixa xarxa que es troben els dos ordinadors utilitzats durant aquest treball. Com que l'únic que necessitem és el tòpic generat pels sensors, es podria visionar en un extrem aquesta captura realitzada pels tres Kinect. S'hauria d'aprofundir, però, en com s'hauria de modificar el codi de l'eliminació de fons aplicat en aquest projecte per tal d'utilitzar el mateix sistema amb els tres sensors. També es podria crear una nova captura amb tres Kinect a l'altre extrem i hi hauria dues imatges 360°.

Una altra línia podria ser explorar diferents eines de processat de núvols de punts per tal de millorar el filtrat del núvol de punts per aconseguir un bon rendiment amb Unity, així com afegir diferents elements de realitat augmentada, accions amb SteamVR o crear espais de reunió interactius.

Durant la recerca vaig poder comprovar que és possible enviar un feed d'àudio mitjançant ROS (les ulleres de realitat virtual incorporen auriculars i micròfon). En aquest sentit seria interessant crear una comunicació verbal per facilitar la interacció entre els usuaris.

Bibliografia:

- [1] “ROS.org | About ROS.” <https://www.ros.org/about-ros/> (accessed Jan. 02, 2021).
- [2] “pcl - ROS Wiki.” <http://wiki.ros.org/pcl> (accessed Jan. 21, 2021).
- [3] “SteamVR.” <http://steamvr.com> (accessed Jan. 02, 2021).
- [4] “VIVE European Union | Discover Virtual Reality Beyond Imagination.” <https://www.vive.com/eu/> (accessed Jan. 21, 2021).
- [5] U. Technologies, “Unity Real-Time Development Platform | 3D, 2D VR & AR Engine.” <https://unity.com/> (accessed Jan. 03, 2021).
- [6] “Kinect: desarrollo de aplicaciones de Windows.” <https://developer.microsoft.com/es-es/windows/kinect/> (accessed Jan. 21, 2021).
- [7] I. Moreno Vega, “Mixed Reality (AR/VR) demonstrator,” *Mixed Reality (AR/VR) demonstrator with object selection in real-time on ROS*, Feb. 2020, Accessed: Jan. 22, 2021. [Online]. Available: <https://upcommons.upc.edu/handle/2117/178164>.
- [8] A. Maki, “Improved AR/VR Telepresence Application in the SmartRoom,” *Mejora en una Aplicación VR/AR de Telepresencia en la SmartRoom*, Sep. 2019, Accessed: Jan. 22, 2021. [Online]. Available: <https://upcommons.upc.edu/handle/2117/174562>.
- [9] E. Bayó Puxan, “Visualització objectiva d’una escena de realitat augmentada,” *Objective visualization of an augmented reality scene*, Jul. 2018, Accessed: Jan. 22, 2021. [Online]. Available: <https://upcommons.upc.edu/handle/2117/123497>.
- [10] L. Mora Ballestar, “3D surface reconstruction in telepresence systems,” *Reconstrucció de superfícies 3D en sistemes de telepresencia*, Jun. 2017, Accessed: Jan. 22, 2021. [Online]. Available: <https://upcommons.upc.edu/handle/2117/109805>.
- [11] A. Casanova Pagà, “Fusion of 3D data for interactive applications,” *Fusión de datos 3D para aplicaciones interactivas*, Jul. 2016, Accessed: Jan. 22, 2021. [Online]. Available: <https://upcommons.upc.edu/handle/2117/89001>.
- [12] A. Sun, “Building a virtual world in ROS based on the robot’s perception.” p. 32.
- [13] D. Whitney, E. Rosen, D. Ullman, E. Phillips, and S. Tellex, “ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Oct. 2018, pp. 1–9, doi: 10.1109/IROS.2018.8593513.
- [14] “Holoportation,” *Microsoft Research*. <https://www.microsoft.com/en-us/research/project/holoportation-3/> (accessed Jan. 22, 2021).
- [15] AndreGilerson. (2020). AndreGilerson/rviz_vive. RVIZ Plugin for the HTC Vive [C++]. https://github.com/AndreGilerson/rviz_vive (Original work published 2018)
- [16] inmojang. (2020). inmo-jang/perception_pcl. Point Cloud Filtering Examples [C++]. https://github.com/inmo-jang/perception_pcl (Original work published 2019)
- [17] inmojang. (2020). inmo-jang/unity_assets. Useful Unity assets for robotics and virtual reality [C#]. https://github.com/inmo-jang/unity_assets (Original work published 2020)
- [18] “Foreground detection - Wikipedia.” https://en.wikipedia.org/w/index.php?title=Foreground_detection&oldid=100098282

Apèndix

A.1. Work Packages

Work Breakdown Structure

WP1: Instal·lació del sistema operatiu necessari per començar a treballar (PC1).

1. Instal·lar Linux GNOME
2. Instal·lar ROS (Melodic) + llibreries
3. Instal·lar Steam + SteamVR Linux

WP2: Aconseguir capturar i visionar el núvol de punts d'un Kinect2 i ROS (PC1).

1. Connectar Kinect2 amb Linux
2. Capturar amb Kinect2 i ROS + visualitzar el núvol de punts creat
3. Utilitzar tf i millorar captura Kinect2

WP3: Visionar les dades amb les ulleres de realitat virtual HTC Vive (PC 1).

1. Crear el bridge per enviar les dades
2. Enviar núvol de punts cap a les ulleres
3. Comprovar la visualització

WP4: Duplicar la instal·lació del sistema operatiu (PC2)

1. Instal·lar Linux GNOME.
2. Instal·lar ROS (Melodic) + llibreries
3. Instal·lar Steam + SteamVR (Linux)

WP5: Aconseguir capturar i visionar la captura en 3D amb un Kinect2 i ROS (PC2)

1. Connectar Kinect2 amb Linux
2. Capturar amb Kinect2 i ROS + visualitzar el núvol de punts creat
3. Utilitzar tf i millorar captura Kinect2

WP6: Visionat segones ulleres de realitat virtual. (PC2)

1. Crear el bridge per enviar les dades
2. Enviar núvol de punts cap a les ulleres
3. Comprovar la visualització

WP7: Creuar el visionat-subscripció de nodelets per tal de crear la bidireccionalitat.

1. Subscriure el nodelet de les HTC Vive 1 al nodelet del núvol de punts de 2.
2. Subscriure el nodelet de les HTC Vive 2 al nodelet del núvol de punts 1.
3. Confirmar que es poden visionar de forma creuada els núvols de punts.

WP8: Correcció del visionat perquè sigui geomètricament correcte.

1. Modificar les tf aplicades per tal que cada usuari vegi al seu davant a l'altra persona.
2. Correcció de dimensions.

WP9: Test i creació de l'script final

1. Testar tot el sistema i comprovar que funciona la bidireccionalitat correctament.
2. Crear script que executi el procés automàticament.

WP10: Documentació corresponent al Treball de Final de Grau.

1. Proposal and Work Plan
2. Project Critical Review
3. Final Report

Project: Virtual Reality Interaction in Telepresence	WP ref: 1	
Major constituent: Software	Sheet n of m	
Short description:	Planned start date: 2-Oct-2020 Planned end date:5-Oct-2020	
Instal·lació del sistema operatiu necessari per començar a treballar (PC1).	Start event:2-Oct-2020 End event:5-Oct-2020	
Internal task T1: Instal·lar Linux GNOME. Internal task T2: Instal·lar ROS (Melodic) + lliberies Internal task T3: Instal·lar Steam + SteamVR (Linux)	Deliverables:	Dates:

Project: Virtual Reality Interaction in Telepresence	WP ref: 2	
Major constituent: Simulation	Sheet n of m	
Short description:	Planned start date: 6- Oct-2020 Planned end date: 13-Oct-2020	
Aconseguir capturar i visionar el núvol de punts d'un Kinect2 i ROS (PC1).	Start event: 6-Oct-2020 End event:13-Oct-2020	
Internal task T1: Connectar Kinect2 amb Linux Internal task T2: Capturar amb Kinect2 i ROS + visualitzar el núvol de punts creat Internal task T3: Utilitzar tf i millorar captura Kinect2	Deliverables:	Dates:

Project: Virtual Reality Interaction in Telepresence	WP ref: 3	
Major constituent: Software	Sheet n of m	
Short description:	Planned start date: 14-Oct-2020 Planned end date: 6-Nov-2020	
Visionar les dades amb les ulleres de realitat virtual HTC Vive (PC 1).	Start event:13-Oct-2020 End event:4-Nov-2020	
Internal task T1: Crear el bridge per enviar les dades Internal task T2: Enviar núvol de punts cap a les ulleres Internal task T3: Comprovar la visualització	Deliverables:	Dates:

Project: Virtual Reality Interaction in Telepresence	WP ref: 4	
Major constituent: Software	Sheet n of m	
Short description:	Planned start date:9-Nov-2020 Planned end date:10-Nov-2020	
Duplicar la instal·lació del sistema operatiu (PC2)	Start event:9-Nov-2020 End event:10-Nov-2020	
Internal task T1: Instal·lar Linux GNOME. Internal task T2: Instal·lar ROS (Melodic) + llibreries Internal task T3: Instal·lar Steam + SteamVR (Linux)	Deliverables:	Dates:

Project: Virtual Reality Interaction in Telepresence	WP ref: 5	
Major constituent: Simulation	Sheet n of m	
Short description:	Planned start date:11-Nov-2020 Planned end date:13-Nov-2020	
Aconseguir capturar i visionar la captura en 3D amb un Kinect2 i ROS (PC2)	Start event:5 - Nov- 2020 End event:16 - Nov - 2020	
Internal task T1: Connectar Kinect2 amb Linux Internal task T2: Capturar amb Kinect2 i ROS + visualitzar el núvol de punts creat Internal task T3: Utilitzar tf i millorar captura Kinect2	Deliverables:	Dates:

Project: Virtual Reality Interaction in Telepresence	WP ref: 6	
Major constituent: Software	Sheet n of m	
Short description:	Planned start date:16-Nov-2020 Planned end date:20-Nov-2020	
Visionat segones ulleres de realitat virtual. (PC2)	Start event:16-Nov-2020 End event: esperem ulleres	
Internal task T1: Crear el bridge per enviar les dades Internal task T2: Enviar núvol de punts cap a les ulleres Internal task T3: Comprovar la visualització	Deliverables:	Dates:

Project: Virtual Reality Interaction in Telepresence	WP ref: 7	
Major constituent: Software	Sheet n of m	
Short description:	Planned start date:23-Nov-2020 Planned end date:27-Nov-2020	
Creuar el visionat-subscripció de nodelets per tal de crear la bidireccionalitat.	Start event:20 - Nov - 2020 End event:4 - Des - 2020	
Internal task T1: Subscriure el nodelet de les HTC Vive 1 al nodelet del núvol de punts de 2. Internal task T2: Subscriure el nodelet de les HTC Vive 2 al nodelet del núvol de punts 1. Internal task T3: Confirmar que es poden visionar de forma creuada els núvols de punts.	Deliverables:	Dates:








Project: Virtual Reality Interaction in Telepresence	WP ref: 8	
Major constituent: Simulation	Sheet n of m	
Short description:	Planned start date:30-Nov-2020 Planned end date:11-Des-2020	
Correcció del visionat perquè sigui geomètricament correcte.	Start event: 7- Des - 2020 End event: 18 - Des - 2020	
Internal task T1: Modificar les tf aplicades per tal que cada usuari vegi al seu davant a l'altra persona. Internal task T2: Correcció de dimensions.	Deliverables:	Dates:

Project: Virtual Reality Interaction in Telepresence	WP ref: 9	
Major constituent: Test	Sheet n of m	
Short description:	Planned start date:14-Des-2020 Planned end date:21-Des-2020	
Test i creació de l'script final	Start event: 21 - Des - 2020 End event: 24 - Des - 2020	
Internal task T1: Testar tot el sistema i comprovar que funciona la bidireccionalitat correctament. Internal task T2: Crear script que executi el procés automàticament.	Deliverables:	Dates:

Project: Virtual Reality Interaction in Telepresence	WP ref: 10	
Major constituent: Documentació	Sheet n of m	
Short description:	Planned start date:14-Set-2020 Planned end date:26-Gen-2021	
Documentació corresponent al Treball de Final de Grau.	Start event: 14-Set-2020 End event: 26-Gen-2021	
Internal task T1: Proposal and Work Plan Internal task T2: Project Critical Review Internal task T3: Final Report	Deliverables:	Dates:
	Proposal and Workplan	23-Oct-2020
	Project Critical Review	1 - Des - 2020
	Final Report	26-Gen-2021

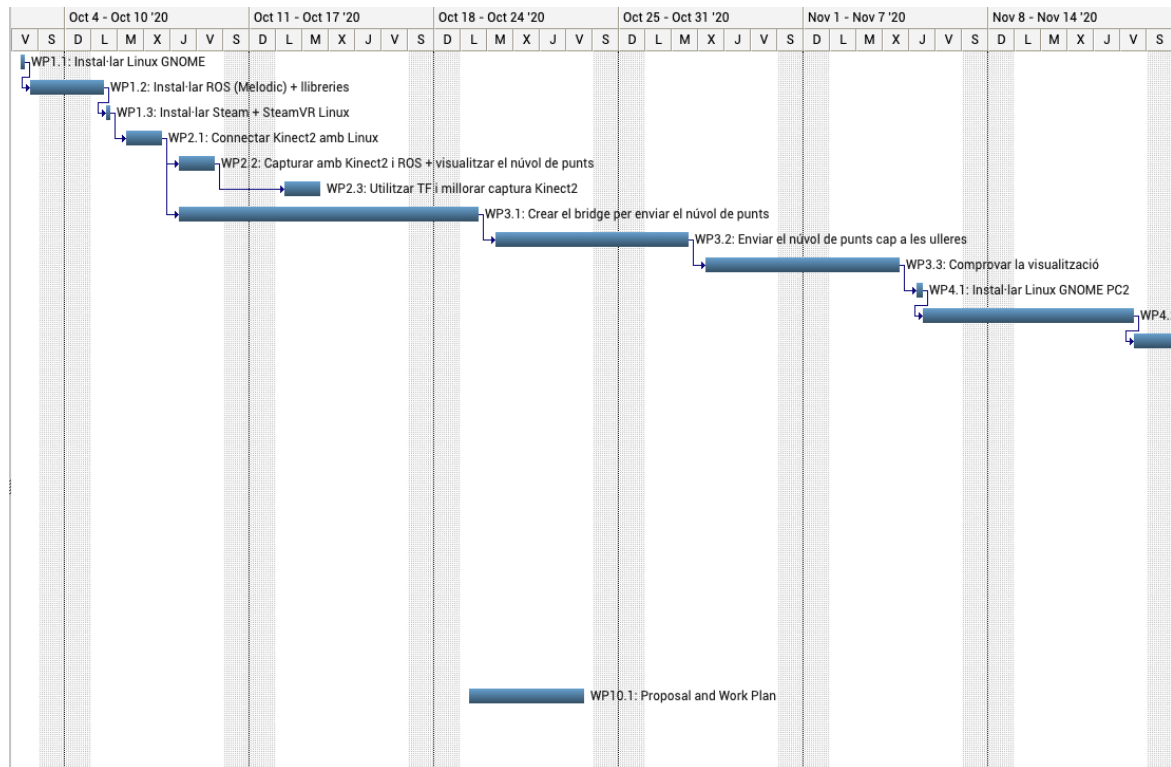
Taula 4 - Work Packages

Milestones

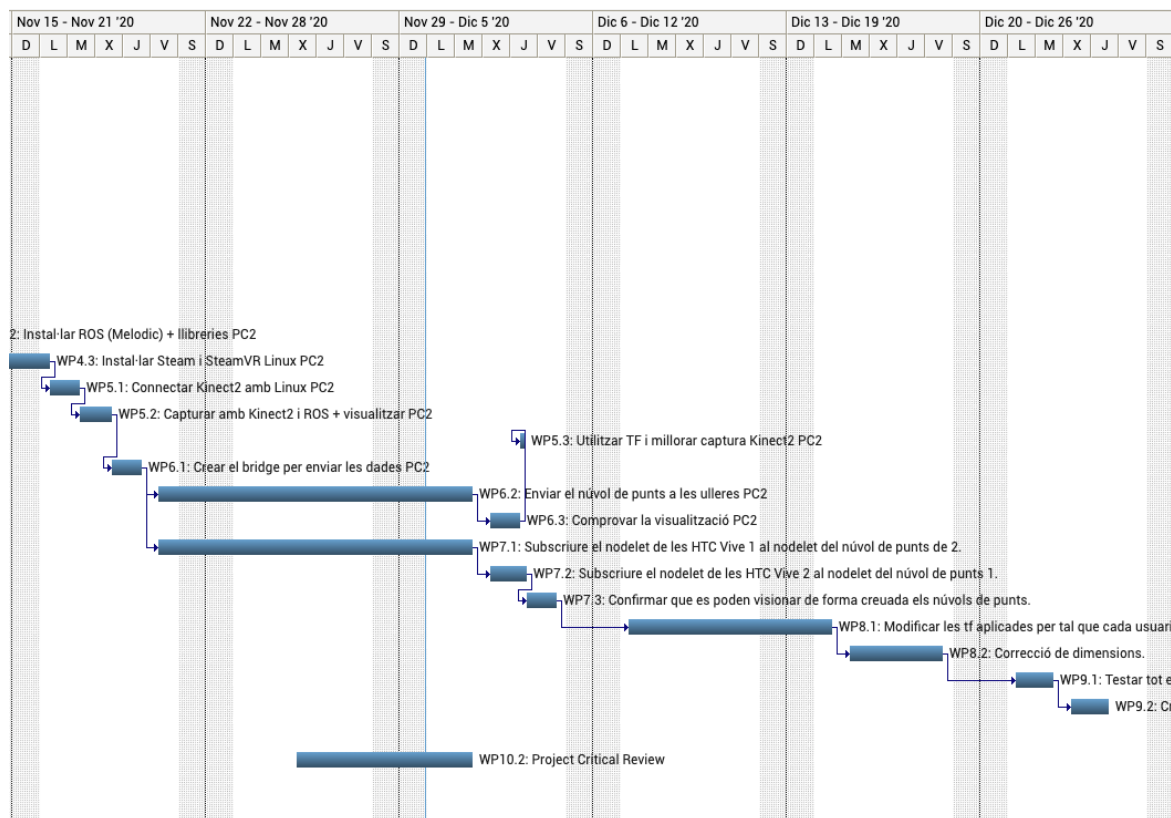
		Nombre	Duración	Inicio	Fin	Predecessoras
1		WP1.1: Instal·lar Linux GNOME	0.25día	10/02/2020	10/02/2020	
2		WP1.2: Instal·lar ROS (Melodic) + llibreries	0.25día	10/02/2020	10/05/2020	1
3		WP1.3: Instal·lar Steam + SteamVR Linux	0.25día	10/05/2020	10/05/2020	2
4		WP2.1: Connectar Kinect2 amb Linux	1día	10/06/2020	10/07/2020	3
5		WP2.2: Capturar amb Kinect2 i ROS + visualitzar el núvol de punts	1día	10/08/2020	10/09/2020	4
6		WP2.3: Utilitzar TF i millorar captura Kinect2	1día	10/12/2020	10/13/2020	5
7		WP3.1: Crear el bridge per enviar el núvol de punts	4días	10/08/2020	10/19/2020	4
8		WP3.2: Enviar el núvol de punts cap a les ulleres	3días	10/20/2020	10/27/2020	7
9		WP3.3: Comprovar la visualització	3días	10/28/2020	11/04/2020	8
10		WP4.1: Instal·lar Linux GNOME PC2	0.3día	11/05/2020	11/05/2020	9
11		WP4.2: Instal·lar ROS (Melodic) + llibreries PC2	3días	11/05/2020	11/13/2020	10
12		WP4.3: Instal·lar Steam i SteamVR Linux PC2	0.3día	11/13/2020	11/16/2020	11
13		WP5.1: Connectar Kinect2 amb Linux PC2	0.6día	11/16/2020	11/17/2020	12
14		WP5.2: Capturar amb Kinect2 i ROS + visualitzar PC2	0.7día	11/17/2020	11/18/2020	13
15		WP5.3: Utilitzar TF i millorar captura Kinect2 PC2	0.2día	12/03/2020	12/03/2020	18
16		WP6.1: Crear el bridge per enviar les dades PC2	0.6día	11/18/2020	11/19/2020	14
17		WP6.2: Enviar el núvol de punts a les ulleres PC2	4días	11/20/2020	12/01/2020	16
18		WP6.3: Comprovar la visualització PC2	0.6día	12/02/2020	12/03/2020	17
19		WP7.1: Subscriure el nodelet de les HTC Vive 1 al nodelet del núv	4días	11/20/2020	12/01/2020	16
20		WP7.2: Subscriure el nodelet de les HTC Vive 2 al nodelet del núv	0.9día	12/02/2020	12/03/2020	19
21		WP7.3: Confirmar que es poden visionar de forma creuada els núv	0.6día	12/03/2020	12/04/2020	20
22		WP8.1: Modificar les tf aplicades per tal que cada usuari vegi al se	3días	12/07/2020	12/14/2020	21
23		WP8.2: Correcció de dimensions.	2días	12/15/2020	12/18/2020	22
24		WP9.1: Testar tot el sistema i comprovar que funciona la bidireccio	1día	12/21/2020	12/22/2020	23
25		WP9.2: Crear script que executi el procés automàticament.	1día	12/23/2020	12/24/2020	24
26		WP10.1: Proposal and Work Plan	2.5días?	10/19/2020	10/23/2020	
27		WP10.2: Project Critical Review	2.5días?	11/25/2020	12/01/2020	
28		WP10.3: Final Report	7.5días?	01/05/2021	01/25/2021	

Il·lustració 22 – Milestones (fet amb el programa Ganter)

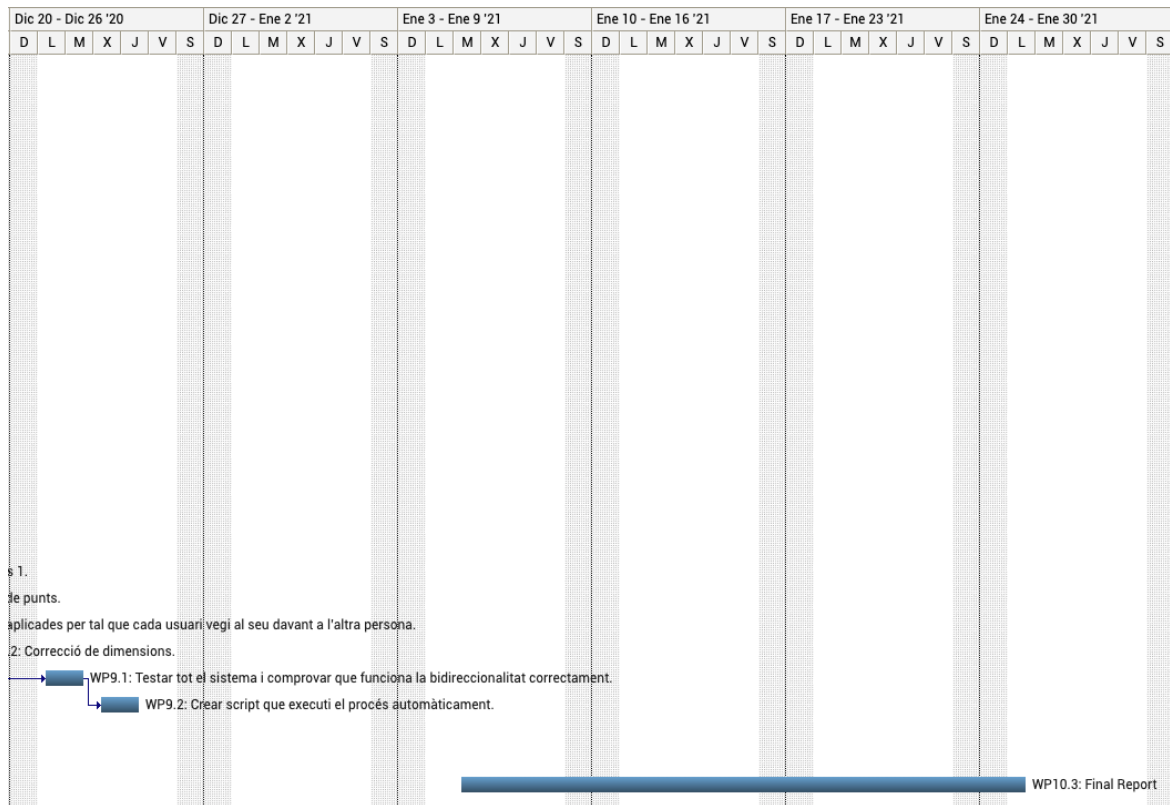
A.2. Diagrama Gantt



Il·lustració 23 - Gantt (part 1)



Il·lustració 24 - Gantt (part 2)



Il·lustració 25 - Gantt (part 3)

A.3. Comandes del terminal

PC 1:

Inicialitzem el core de ROS:

```
. ~/catkin_ws/devel/setup.bash  
roscore
```

Connectem amb el Kinect2 del PC1, publicant la TF i el núvol de punts

```
. ~/catkin_ws/devel/setup.bash  
&  
roslaunch kinect2_bridge kinect2_bridge.launch publish_tf:=true  
fps_limit:=30  
&  
. ~/catkin_ws/devel/setup.bash  
&  
roslaunch tf_static_transform_publisher 0 0 0 0 0 0 camera_link  
kinect2_link 100
```

PC 2:

Connectem amb el Kinect2 del PC2, publicant la TF i el núvol de punts

```
. ~/catkin_ws/devel/setup.bash  
&  
roslaunch kinect2_bridge kinect22_bridge.launch publish_tf:=true  
fps_limit:=30  
&  
. ~/catkin_ws/devel/setup.bash  
&  
roslaunch tf_static_transform_publisher 0 0 0 0 0 0 camera_link  
kinect22_link 100
```

per unity:

Un cop hem fet els passos del bloc 1:

Eliminem el fons de les captures d'ambdós Kinects i creem un pont amb unity amb rosbridge

```
. ~/catkin_ws/devel/setup.bash  
roslaunch pcl_ros camera_statistical_outlier_removal.launch  
&  
roslaunch pcl_ros camera_statistical_outlier_removal2.launch  
&  
roslaunch rosbridge_server rosbridge_websocket.launch  
  
. ~/catkin_ws/devel/setup.bash  
roslaunch rqt_graph rqt_graph
```

Obrim el projecte de Unity

per plugin directe a les ulleres htc vive des de rviz:

Un cop hem fet els passos del bloc 1:

```
. ~/catkin_ws/devel/setup.bash  
~/steam/steam/ubuntu12_32/steam-runtime/run.sh roslaunch rviz rviz
```



A.4. Documentació

wiki: PrjTelepresence / documentation / SteamVRforLinux

Up | Start Page | Index | History | Watch This

SteamVR for Linux

Author: Mireia de Gracia Forés

Passos per la instal·lació de SteamVR a Linux

1. Comprovar sistemes operatius compatibles amb el Test que facilita Steam

⇒ <https://store.steampowered.com/app/690390/Test/>

(Els sistemes compatibles el 5 d'Octubre de 2020 són Gnome, Unity i KDE. En cas de no tenir instal·lat qualsevol d'aquests escriptoris consultar l'Anex al final d'aquesta pàgina).

2. Comprovar que tenim els **drivers** instal·lats de Nvidia, AMD i Intel.

Els comandaments a utilitzar són:

- ubuntu-drivers devices

-- insertar foto mail--

Si estem d'acord amb les recomanacions, podem instal·lar tots els drivers recomanats utilitzant:

```
sudo ubuntu-drivers autoinstall
```

En cas contrari, podem instal·lar el driver desitjat utilitzant el comandament:

```
sudo apt install nvidia-360 (per exemple)
```

Finalment reiniciem el nostre sistema perquè els canvis siguin efectius.

3. Instal·lar **Steam**

Per instal·lar Steam a Ubuntu és tant senzill com escriure

```
sudo apt install steam
```

Un cop tenim instal·lat Steam ens demanarà entrar-hi amb el nostre usuari. Fem un Log In.

4. Instal·lar **SteamVR**

Quan ja tenim Steam funcionant al nostre Linux, hem de connectar-hi les ulleres de realitat virtual. Amb l'equip connectat al PC i a corrent, ens apareixerà un missatge a Steam: "Using a VR Set?" i clicarem a "Install SteamVR".

Un cop finalitzada la instal·lació de SteamVR procedirem a definir l'espai disponible de la sala amb la cal·libració del propi SteamVR. Podem seguir també els tutorials disponibles per entendre com funciona aquesta tecnologia.

A la configuració de la VR activarem la càmera de les nostres ulleres.

Annex: Instal·lar Gnome

Per instal·lar Gnome sobre Ubuntu 18.04 Bionic Beaver Linux:

- sudo apt install taskel
- sudo taskel install ubuntu-desktop
- sudo service gdm3 start

En el seu defecte, si es prefereix un escriptori Gnome Vanilla i no la versió full com s'ha indicat anteriorment, es pot fer:

- sudo apt install gnome-session gdm3

Il·lustració 26 - Exemple documentació SteamVR per Linux



Plugin de Rviz per HTC Vive

Author: Mireia de Gracia

Podem trobar el plugin al repositori d'Andre Gilerson. [⇒ https://github.com/AndreGilerson/rviz_vive](https://github.com/AndreGilerson/rviz_vive)

Aquest plugin permet visualitzar el núvol de punts directament des d'RVIZ cap a les ulleres de realitat virtual HTC Vive. EL que es visualitza és el GRID. Amb el moviment del cap es produeixen modificacions similars a les que podríem fer amb el mouse

Aquest plugin s'ha provat amb ROS Melodic, i les versions més noves de SteamVR i OpenVR (Octubre 2020).

Els passos per instal·lar aquest plugin són els següents:

- 1- Dins de la carpeta src del nostre espai de treball catkin_ws (creat anteriorment, si no ha estat creat revisar els tutorials de ROS), clonem el repositori del plugin.
- 2- Cal clonar en un altre directori (Home per exemple) el repositori d'OpenVR actual. [⇒ https://github.com/ValveSoftware/openvr](https://github.com/ValveSoftware/openvr)
- 3- Un cop clonat el repositori OpenVR i rviz_vive, dins del directori de rviz_vive obrim el CMakeList i modifiquem la línia 30 on hi ha el path del OpenVR (aquí hi posem el Path on hem clonat el OpenVR al nostre PC).
- 4- Guardats els canvis, anem des del terminal a catkin_ws i fem catkin_make.
- 5- Ja està instal·lat, cal recordar que perquè funcioni cal fer ./catkin_ws/devel/setup.bash al terminal abans de córrer rviz
- 5.1- NOTA: és bo reiniciar el PC un cop s'ha instal·lat el plugin. NOTA 2: connectar el headset abans d'inicialitzar rviz
- 6- Al terminal on volem obrir rviz, a més de fer el pas 5 cal executar-ho de la següent forma perquè funcioni: (es pot consultar la bibliografia sobre SteamVR a Github) [⇒ https://github.com/ValveSoftware/SteamVR-for-Linux](https://github.com/ValveSoftware/SteamVR-for-Linux)

```
~/steam/steam/ubuntu12_32/steam-runtime/run.sh rosrn rviz rviz
```

*** NOTA: és possible que tinguem el següent error quan intentem executar el pas 6:**

```
CURL_OPENSSL_4 not found....
```

Per solucionar_ho: sudo apt-get update sudo apt-get install libcurl4-openssl-dev

```
cd /home/$USER/.steam/ubuntu12_32/steam-runtime/pinned_libs_64 mv libcurl.so.4 libcurl.so.4.bak ln -s /usr/lib/x84_64-linux-gnu/libcurl.so.4 libcurl.so.4
```

Il·lustració 27- Exemple documentació plugin rviz



ROS Melodic

Author: Mireia de Gracia Forés

A continuació es detallen els passos per la instal·lació de ROS Melodic sobre Ubuntu 18.04. Aquesta versió de ROS està disponible des del 2018 fins al 2023. Passada aquesta data és serà necessari instal·lar un nou ROS.

1. Instal·lar el repositori de ROS Melodic

El primer que farem és configurar el repositori de ROS perquè l'instal·lador trobi els paquets d'aquest software.

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu bionic main" > /etc/apt/sources.list.d/ros-melodic.list'
```

2. Afegir la clau de ROS

Utilitzarem apt-key per afegir la clau.

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

3. Actualitzar localment els packages ROS

```
sudo apt update
```

Observarem que efectivament hi ha un GET cap a l'URL de ROS, i posteriorment un HIT.

4. Instal·lar el paquet de ROS Melodic

Hi ha diferents paquets disponibles per Ubuntu 18.04. El primer que instal·larem és un "full desktop" (inclou PCL i opencv). En cas de necessitar expansions, s'instal·laran més endavant.

```
sudo apt install ros-melodic-desktop-full
```

Com que pel nostre projecte és imprescindible l'ús dels nodelet, també instal·larem el metapaquet "base", que conté diferents paquets opcionals que necessitarem.

```
sudo apt install ros-melodic-base
```

5. Entorn ROS

Un cop hem instal·lat els paquets, hem de configurar l'entorn perquè reconegui els fitxers de programa, ja que quan s'han instal·lat no s'han guardat a un directori que ROS sàpiga localitzar.

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
```

6. Comprovar la instal·lació

Ara podem escriure roscd al terminal per veure que ens ha redirigit al directori /opt/ros/melodic

Per iniciar un daemon de ROS podem executar roscore. Mentre roscore s'executa no podem fer res més en aquest terminal, per això utilitzarem roscore list, i per sortir /rosout

Il·lustració 28 - Exemple documentació instal·lació ROS Melodic