



Sistema de reconstrucció de cavitats en 3D

Autor: Adrià Aguilà Pérez

Data: 27/10/2016

Director: Josep Fernández Ruzafa (ESAI)

Titulació: Grau en Enginyeria Informàtica

Especialitat: Enginyeria de Computadors

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC)



Agraïments

Moltes gràcies a totes les persones que m'han anat donant el seu suport al llarg d'aquest projecte, i al departament d'ESAI de la UPC per proporcionar-me accés al material necessari.



Resum

Les instal·lacions del clavegueram són d'una importància extrema per mantenir una correcta situació d'higiene en una ciutat. Un cop construïdes, és necessari fer-ne un manteniment regular per garantir que segueixen complint la seva funció, però degut a la seva naturalesa subterrània i de conductes foscos i estrets, aquesta no sempre és una tasca senzilla.

L'objectiu d'aquest projecte és contribuir al desenvolupament d'una eina que permeti realitzar aquests manteniments d'una forma més senzilla, segura i eficient. Aquesta eina està sent desenvolupada pel departament d'ESAI de la UPC, per petició de l'empresa de Fomento y Contratas (FCC), els encarregats actualment de fer el manteniment del clavegueram de Barcelona. La seva finalitat principal serà l'exploració d'embornals, que tot i ser la part del clavegueram que més a prop es troba de la superfície, no resulten fàcils d'obrir i inspeccionar, motiu pel qual es vol un aparell que pugui realitzar el procés d'exploració de forma directa i sense necessitat d'aixecar-ne la reixa.

El projecte consistirà en desenvolupar una part del *software* d'aquesta eina. Concretament, es centra al voltant del disseny d'un algorisme que permeti convertir un conjunt de punts capturats en un espai en 3D i que representen l'interior d'aquest embornal a una dada numèrica que indicarà quin percentatge de la canonada que connecta amb el sistema de clavegueram principal està obstruïda, perquè l'equip de manteniment pugui valorar si és necessari accedir a l'embornal per fer una neteja o no.



Resumen

Las instalaciones de alcantarillado son de una importancia extrema para mantener una correcta situación de higiene en una ciudad. Una vez construidas, es necesario realizarles un mantenimiento regular para garantizar que siguen cumpliendo su función, pero debido a su naturaleza subterránea y de conductos oscuros y estrecho, ésta no siempre resulta tarea fácil.

El objetivo de este proyecto es contribuir al desarrollo de una herramienta que permita realizar estos mantenimientos de una forma mas sencilla, segura y eficiente. Esta herramienta esta siendo desarrollada por el departamento de ESAll de la UPC, por petición de la empresa Fomento y Contratas (FCC), los actuales encargados de hacer el mantenimiento del alcantarillado de Barcelona. Su finalidad principal será la exploración de imbornales, que aunque són la parte del alcantarillado más cercana a la superficie, no resultan fáciles de abrir e inspeccionar, motivo por el qual se desea un aparato que sea capaz de realizar el proceso de exploración de forma directa y sin necesidad de levantar la reja.

El proyecto consistirá en desarrollar una parte del *software* de esta herramienta. Concretamente, se centra alrededor del diseño de un algoritmo que permita convertir un conjunto de puntos capturados en un espacio de 3D y que representan el interior de este imbornal en un dato numérico que indicará que porcentaje de la cañería que conecta con el sistema principal de alcantarillado está obstruída, para que el equipo de mantenimiento pueda valorar si es necesario acceder al interior para realizar una limpieza o no.



Abstract

In any city, sewer installations are of major importance to help maintain a correct hygienic state. Once built, it's necessary to do regular maintenance on them to guarantee they keep accomplishing their function properly. Provided they are built in an underground System filled with dark and small conduits, this is not usually an easy task.

The aim of this project is to contribute to the development of a tool that aids in doing this maintenance in an easier, safer and more efficient way. This tool is being developed by the UPC's ESAII department by petition of the Fomento and Contratas company, the current responsables for doing Barcelona's city sewer maintenance. It's main purpose is the exploration of scuppers, which even though are the most superficial part of the sewer System, are not particularly easy to open and examine manually. For this reason, the company wants a device capable of doing this exploration process directly and without even needing to lift the case.

This project will be centered around the developing a part of the *software* for this device. In particular, we want to design an algorithm that is capable of transforming a particular set of points in a 3D space which represent the inside of a scupper into a numerical value which indicates the portion of the pipe connection to the main draining system is obstructed. This value will help the maintenance team in taking a decision regarding wether to manually access the scupper for cleaning or not.



Índex

1. Introducció.....	7
1.1 Context.....	7
1.2 Actors implicats	9
1.2.1 Desenvolupador	9
1.2.2 Director i suport	9
1.2.3 Usuaris	10
1.2.4 Beneficiaris	10
1.3 Estat de l'art.....	10
1.3.1 Manuals de referència.....	11
1.3.2 Detecció de línies	11
1.3.3 Detecció de plans	11
2. Formulació del problema i abast	13
2.1 Objectiu	13
2.2 Possibles obstacles	13
2.2.1 Encallar-se en la implementació.....	13
2.2.2 No disponibilitat del PC de treball.....	14
2.2.3 Pèrdua de les dades	14
2.2.4 Problemes amb el <i>hardware</i>	14
2.2.5 Temps d'espera de càlcul i proves.....	14
2.3 Metodologia i rigor.....	14
2.3.1 Metodologia <i>Scrum</i>	14
2.3.2 Entorn de desenvolupament.....	15
2.3.3 Eines de seguiment	15
2.3.4 Mètode de validació.....	15
2.3.5 Avaluació final	16
2.3.5 Revisió de la metodologia.....	16
3. Solució final	17
4. Planificació temporal.....	19
4.1 Especificació de les tasques	19
4.1.1 Gestió del projecte	19
4.1.2 Incorporació i familiarització amb l'entorn de treball.....	20
4.1.3 Implementació del projecte.....	20



4.1.4 Fase de validació.....	22
4.1.5 Fase final.....	22
4.2 Temps per tasca.....	23
4.3 Dependències de tasques.....	23
4.4 Alternatives i pla d'acció.....	24
5. Pressupost.....	25
5.1 Control de gestió.....	28
6. Desviacions.....	29
6.1 Danys greus al capçal.....	29
6.2 Cal·libració i generador de dades.....	30
7. Sostenibilitat.....	32
7.1 Dimensió econòmica.....	32
7.2 Dimensió social.....	33
7.3 Dimensió ambiental.....	33
7.4 Lleis i regulacions.....	34
8. Incorporació i familiarització.....	35
8.1 <i>Microsoft Visual Studio</i>	35
8.2 <i>CloudCompare</i>	36
8.3 El <i>hardware</i>	37
8.4 El codi.....	38
9. Implementació.....	39
9.1 Detecció de rectes.....	39
9.1.1 Càlcul de distàncies mínimes.....	40
9.1.2 Càlcul de desviacions.....	42
9.1.3 Vector de diferències.....	42
9.1.4 Algorisme final.....	43
9.2 Obturació de la canonada.....	48
9.2.1 Detecció de la canonada.....	48
9.2.2 Càlcul del nivell d'obturació.....	48
9.3 Resultat final.....	51
10. Desviacions.....	54
10.1 Cal·libració del sistema.....	54
10.2 Generador de dades.....	58
11. Conclusions i futura feina.....	59
12. Bibliografia.....	61

1. Introducció

1.1 Context

Aquest TFG es desenvolupa amb la col·laboració del Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAII) de la FIB. L'objectiu final del projecte és crear un sistema capaç d'explorar cavitats, analitzar-les, i reproduir-les visualment en un entorn 3D.

Es tracta d'un projecte sol·licitat per l'empresa de FCC (Fomento i Contratas). Aquesta empresa s'encarrega, entre altres coses, de la gestió i el correcte funcionament del clavegueram de la ciutat de Barcelona. L'objectiu principal d'aquest sistema és doncs l'exploració dels embornals de la ciutat, concretament del nivell de sediment del terra i del nivell d'obturació del conducte que comunica amb el clavegueram. En general, per inspeccionar un embornal actualment cal aixecar-ne la tapa i fer-ne un reconeixement manual. Aquest procés és molt lent, perquè obrir l'embornal és costós: calen algunes eines per afluixar la reixa, sol haver-hi cotxes trepitjant la tapa o bloquejant-ne una part del recorregut, etc.

El sistema demanat és una perxa amb càmeres que permetrà explorar ràpidament l'embornal. La idea és que un operari situï la base de la perxa sobre l'embornal i vagi movent-lo amunt i avall, mentre un segon operari dirigeix el procés des d'un ordinador.



Figura 1. Exemple d'un embornal com els que es vol inspeccionar, en el que es poden apreciar els seus components principals (reixa, capsa prismàtica i forat de connexió).

El sistema consta doncs, de dues parts: una primera part *hardware*, composta de perxa i base, i una part *software* per gestionar-lo. Pel que fa a la part física és, com ja s'ha dit, una perxa amb una base per sostenir-lo. Hi ha dues plaques *Arduino*, una incorporada a la perxa i l'altra a la base, que s'encarreguen de llegir la informació dels diversos sensors del sistema: la base té un *encoder* que permet saber la profunditat a la qual està a la perxa i que serà llegida per l'*Arduino* de la base, mentre que a l'extrem de la perxa hi ha dues càmeres, dos *LEDS* vermells i dos *LEDS* per il·luminar. Aquests elements estan separats 180 graus entre ells, de tal forma que a cada costat hi hagi una càmera i dos *LEDS*, un vermell i un per il·luminar. Amb un muntatge així, és fàcil binaritzar la imatge rebuda de les càmeres per detectar exactament el punt vermell i saber a quina distància hi ha la paret, per posteriorment localitzar el forat i el seu nivell d'obturació.



Figura 2. Detall d'un costat del capçal on es veuen, de dalt a baix, la càmera, el LED d'il·luminació i el LED vermell.



Figura 4. Sistema complet, es poden apreciar tots els components. De dalt a baix, perxa, base amb l'encoder,

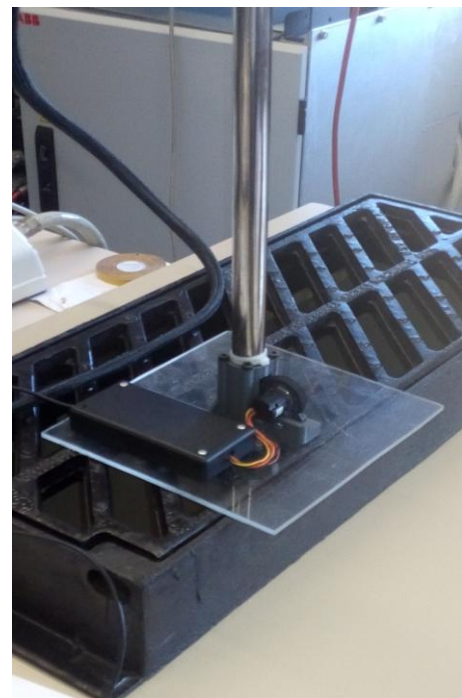


Figura 3. Exemple d'ús del sistema i detall de la base.



La part *software* del projecte és la que em correspon desenvolupar a mi en la seva major part. La idea darrere el *software* és recollir les dades que arriben dels sensors i traduir-les a una representació interpretable. Concretament, rep el conjunt de lectures dels sensors en forma de punts en l'espai i s'han de transformar a un conjunt de plans que reproduïx de la forma més acurada possible l'interior de l'embornal.

Per tractar els punts es fa servir la llibreria *PointCloudLibrary* (amb arxius d'extensió .pcl), que proporciona tant les estructures dels punts en els espais que es vulgui, de les que ens centrarem en la representació (x,y,z) , com estructures de núvols de punts. Aquestes estructures ofereixen la majoria de funcionalitats que ofereix un vector genèric amb l'afegit que, en tractar-se de contenidors específics dels elements bàsics d'aquesta llibreria, proporcionen mètodes per tractar aquests punts que poden resultar útils en algun punt del desenvolupament.

1.2 Actors implicats

Tot projecte té un seguit de persones que hi estan implicades de formes molt diverses, tan de forma directa com indirecta, tan per la banda del seu desenvolupament com del seu posterior consum. A continuació s'exposen tots aquests actors.

1.2.1 Desenvolupador

El desenvolupador, jo mateix, és la persona encarregada de la major part de la càrrega del projecte. La seva tasca no es limita solament a desenvolupar la part tècnica del projecte dins dels límits establerts per ell mateix i pels altres implicats, sinó que hi ha també tot un esforç de documentació que no pot faltar pas, com la redacció de la memòria i la presentació del projecte finalitzat. És també el beneficiari del benefici acadèmic i de formació que comporta la seva realització.

1.2.2 Director i suport

Darrere del desenvolupador hi ha en tot moment el seu director, així com també tot un seguit de persones que es troben directament implicades en el projecte i que poden oferir la seva experiència per guiar el desenvolupador en cas de dificultats. Entre aquestes persones, destacaen Josep Fernández Ruzafa, director del TFG.



1.2.3 Usuaris

Els usuaris, la gent que estarà en contacte directe amb el producte finalitzat, són els equips de reconeixement i manteniment del departament de FCC. Aquests empleats veuran una millora substancial en el procés d'explorar els embornals, evitant-se en la majoria de casos haver-los ni tan sols d'obrir a no ser que els calgui una neteja. En la majoria de casos però, que són només d'exploració, el sistema els resoldrà gran part de la feina.

1.2.4 Beneficiaris

El projecte final té com a beneficiaris, com ja s'ha dit, la gent del departament de FCC. Per una banda, els usuaris abans esmentats, els empleats que faran servir el sistema directament. Per altra banda, però no són els únics que se'n beneficiaran, sinó tota la secció del departament encarregada del manteniment del clavegueram. En tractar-se d'una forma més ràpida i eficient de fer les exploracions, les tasques d'inspecció rutinàries els portaran menys temps i comportaran menys costos, estalvis que podran dedicar a altres tasques.

1.3 Estat de l'art

El projecte permet resoldre un problema concret i amb un sistema físic específic, ja que un sistema que no es pugui introduir per la reixa d'un embornal no és vàlid i això limita molt les opcions.

No obstant, aquest projecte presenta clars elements de visió per computador que es poden aplicar, com l'esmentada binarització de les imatges de càmera rebudes. Tenint en compte que es tracta d'un entorn molt fosc, una binarització simple detectarà un punt lluminós sense problemes.

A nivell de *software*, la lectura dels sensors no presenta cap complicació. El que l'aplicació *software* rebrà és un núvol de punts corresponent a les posicions en les quals s'ha detectat el làser. El conjunt de totes les lectures d'un escombrat generarà un núvol de punts a tractar. La part de tractament de dades, que es correspon amb la meva feina principal dins del projecte, vol acabar convertint un núvol de punts inconnex en un conjunt de plans que s'ajustin a l'interior de l'embornal. Això fa pensar que en un futur, tècniques de reconstrucció de plans puguin ser útils, però les més precises tendeixen a ser molt costoses en temps de computació. A més, el primer que cal fer abans de poder fer res és netejar les dades d'entrada per garantir que totes són rellevants.

Tot i que no hi ha cap projecte similar a aquest, molts dels problemes que es presenten a priori si que s'han tractat i s'exposen a continuació.

1.3.1 Manuals de referència

El llenguatge emprat en el projecte és *C++*. Aquest és un llenguatge complex, i en qualsevol moment es poden oblidar detalls al respecte o necessitar alguna cosa que no s'ha fet servir mai abans. Per resoldre aquests dubtes, es disposarà del manual de referència de *C++*^[1].

A banda d'això, també s'utilitza la llibreria específica *PointCloudCompare*, que té el seu propi manual d'usuari on es reflecteixen totes les capacitats de la llibreria. De la mateixa manera, també hi ha disponible un manual de *CloudCompare*, l'aplicació que es fa servir per visualitzar resultats. Aquests dos manuals^{[2][3]} tindran un ús més freqüent, ja que es tracta d'eines específiques del projecte que no s'han tractat abans.

Tot i que hi ha un munt de referències sobre tot aquest software en moltes pàgines diferents (*StackOverflow* per posar un exemple), el millor és anar sempre a les pàgines de referència oficials.

1.3.2 Detecció de línies

Una de les coses que hi ha necessitat d'identificar, en diversos conjunts de punts que representen una secció d'un prisma, són les rectes que defineixen aquesta secció. A partir d'aquestes rectes, es poden trobar les interseccions que defineixen les cantonades del prisma en qüestió. Existeixen diverses formes d'obtenir una recta a través d'un conjunt de punts, de les quals s'ha seleccionat la transformada de Hough^[4]. Amb aquest mètode, es pot trobar la línia que fa la regressió lineal més òptima d'una sèrie de punts filtrats prèviament per tenir el pendent més freqüent en el conjunt (per tant, són òptims per establir una recta).

Aquest algorisme tracta tota parella de punts en un espai x-y creant la recta $y = mx + n$ que els uneix, i fent servir aquests valors per representar la recta en un espai m-n. Un cop es coneix la localització m-n de cada possible recta del conjunt de punts, la feina està en trobar els paràmetres m i n que millor s'adapten a totes alhora.

1.3.3 Detecció de plans

Quan es tinguin les rectes que representen diferents seccions d'una paret, caldrà recompondre aquesta paret. Una primera idea seria extreure els plans a partir de les rectes que l'han de formar. No obstant, després de cercar informació al respecte, tots els mètodes que apareixen són per generar plans a partir de punts, concretament a través d'una regressió lineal múltiple^[5].



Una regressió lineal simple^[6] fa servir un conjunt de punts en un espai x-y per generar una recta que minimitzi la suma de distàncies dels punts del conjunt a la recta. En aquesta suma de distàncies, que dóna la suma de distàncies a partir dels paràmetres m i n de la recta que es faci servir, es pot trobar la distància mínima derivant parcialment m i n , podent calcular així la recta que millor aproxima tots els punts.

Tornant a la generació de plans, saber això ens ajuda en què l'algorisme es pot aplicar també a la generació d'un pla a partir de punts x-y-z en l'espai. El concepte és el mateix^[7] però amb més variables, i es pot obtenir una fórmula que minimitzi la suma de distàncies de punts al pla. Com que volem generar-lo a partir de rectes, una solució simple és convertir les rectes en segments delimitats, i posteriorment a un conjunt de punts delimitat. Amb aquest conjunt de punts que sabem una recta, es pot fer una regressió lineal múltiple i trobar el pla ideal.



2. Formulació del problema i abast

2.1 Objectiu

L'objectiu final del treball es, com ja s'ha expressat anteriorment, tenir un sistema de reconeixement per visualitzar un entorn 3D concret, un embornal del clavegueram, de tal forma que el procés de reconeixement dels mateixos sigui més eficient, segur i ràpid.

Cal tenir en compte que es tracta d'un entorn molt ben delimitat, cosa que juga a favor d'un disseny eficient. Tot i que hi ha molts tipus d'embornals, hi ha certs patrons comuns a tots ells, i aquesta informació es pot tenir en compte a l'hora de trobar solucions òptimes que no podrien aplicar-se en un altre entorn, com per exemple el mètode que s'ha esmentat abans de binaritzar el punt de llum que genera un *LED*. Aquest procés és trivial perquè estem en un entorn tancat i fosc, i a cel obert potser hauria de replantejar-se.

No obstant, l'abast del projecte pot arribar fins més enllà de l'exploració de clavegueram. Amb una mica de feina per generalitzar les parts que aprofiten l'especificitat de l'entorn, ja sigui a través de canvis *hardware*, *software* o una mica de cada, el projecte final hauria de poder-se adaptar amb més o menys feina per poder acabar sent un sistema de reconeixement 3D que tingui més funcionalitats, com ara explorar coves i pous naturals, per posar un cas.

En definitiva, es tracta d'un projecte amb un objectiu concret i específic, però això no li resta capacitat per ser la base de futurs projectes que ho puguin dur un pas més enllà.

2.2 Possibles obstacles

Al llarg de la vida del projecte, poden sortir diverses dificultats imprevistes, ja siguin internes al projecte o causades per algun element exterior. Prevenir aquestes dificultats abans que apareguin ajudarà a estar preparats si apareixen i a tenir un pla d'acció per salvar-les.

2.2.1 Encallar-se en la implementació

El món de la programació és bastant menys que ideal. Això implica que els codis poden trigar a pensar-se, i trigar encara més a funcionar fins que s'arreglin els *bugs* pertinents.

Solució: dedicar-hi més hores, totes les que es pugui fins que la feina estigui acabada per quan toca.



2.2.2 No disponibilitat del PC de treball

Per desenvolupar la meua part del *software* de reconeixement m'ha estat assignat un ordinador al Departament de Robòtica, on tinc tota la feina que he anat realitzant. En determinades circumstàncies, aquest ordinador podria no estar disponible.

Solució: si és poca estona, es pot esperar. Si l'ordinador ha d'estar inoperatiu uns dies, instal·lar tot el programari en un altre i continuar allà el desenvolupament.

2.2.3 Pèrdua de les dades

Mai s'ha de descartar que algun problema de guardat o del sistema operatiu de l'ordinador provoqui la pèrdua de documents i actualitzacions.

Solució: fer *backups* periòdics de la feina feta.

2.2.4 Problemes amb el *hardware*

La meua feina és desenvolupar un *software* que treballa sobre un *hardware* que ja s'ha fet. No obstant, en algun punt del projecte aquest *hardware* pot deixar de funcionar correctament, o pot necessitar d'alguna millora perquè es pugui seguir avançant el *software*.

Solució: realitzar tota la feina que es pugui que no depengui del *hardware*. En cas que sigui imprescindible per avançar, col·laborar amb els responsables i ajudar en tot el possible per tenir-lo operatiu com més aviat millor.

2.2.5 Temps d'espera de càlcul i proves

El programa triga bastant en compilar i en executar-se, i aquest temps probablement només anirà en augment. L'entorn per visualitzar les solucions també es lent a l'hora de mostrar-les. Per tant, cada cop que es recompila el codi, s'executa i es volen veure els resultats, es gasta bastant de temps.

Solució: entre prova i prova, introduir tants canvis rellevants com sigui possible per no desaprofitar el temps invertit en cada prova.

2.3 Metodologia i rigor

En tot projecte, cal una metodologia de treball constant que permeti portar un bon ritme de feina i no endarrerir-se.

2.3.1 Metodologia *Scrum*

La metodologia *Scrum* es basa en fer reunions curtes però molt constants. Aquest mètode és molt àgil i permet fer un bon seguiment del projecte, a part de permetre moltes adaptacions de la planificació que no es podrien dur a terme si es fes una reunió cada 3 mesos amb el tutor.



Després de parlar-ho amb el director del meu projecte, en Josep Fernández, hem acordat que adaptarem les idees d'aquesta metodologia. Com que el meu entorn de treball es el Departament de Robòtica, tindrem trobades regularment i molt sovint, de manera que es poden anar comentant problemes i dificultats i adaptant els objectius i la direcció del treball.

La idea és comentar en aquestes trobades l'estat actual del projecte, mostrant-li tots els avenços destacables que hi hagi hagut des de la última trobada. En cas que hagi aparegut alguna dificultat de les mencionades abans, es pot comentar per intentar trobar una solució i, en cas que sigui apropiat, ajustar els terminis per tenir-la en compte.

2.3.2 Entorn de desenvolupament

La major part del projecte es realitzarà al Departament de Robòtica. Allà hi ha un PC adaptat a les necessitats del projecte en quan a entorn de desenvolupament i programari instal·lat. A més, aquest ordinador està connectat directament a la part *hardware* del projecte i, per tant, és la representació més propera del conjunt de *hardware* i *software* que representarà el projecte final. Per últim, tot i que és possible treballar amb arxius de dades preses en un altre moment, és convenient disposar del sistema física per fer alguna lectura ocasional.

2.3.3 Eines de seguiment

Aquest projecte es desenvolupa en l'entorn de *VisualStudio*. Aquest entorn no té un sistema de *backups* gaire sofisticat per anar mantenint registres de versions, de manera que tot això es porta de forma manual, i cada cert temps s'agafa el codi que està validat i s'arxiva com a versió antiga, del que se'n fan dues còpies per evitar possibles pèrdues.

2.3.4 Mètode de validació

Després de cada canvi rellevant, es fa un testeig general de les funcionalitats noves del programa. Això implica que no s'avançarà mai a una etapa sense tenir garanties que l'etapa prèvia funciona a la perfecció. D'aquesta forma, evitem perdre temps retrocedint per arreglar etapes del projecte que es consideraven tancades però estan donant nous problemes.

La eina de referència per fer aquestes comprovacions és *CloudCompare*, que permet obrir arxius de tipus *.pcl*. Aquests arxius contenen els núvols de punts tractats en el codi i se'n pot fer una visualització extensa per comprovar que el programa fa el que se n'espera.



2.3.5 Avaluació final

Un cop el projecte estigui en un estadi proper a la finalització, és a dir, que el sistema pugui fer execucions continuades de la seva funcionalitat des de lectura fins a representació visual, caldrà fer algunes proves de camp. Això implica sortir de l'entorn controlat del departament, on l'embornal és simulat amb un recinte tancat conegut, i provar el sistema en embornals reals del carrer per veure si el resultat té la qualitat desitjada.

De no ser així, caldria tornar al laboratori per fer les modificacions pertinents, i repetir el cicle fins que s'obtinguin resultats vàlids de forma consistent.

2.3.5 Revisió de la metodologia

Acabat el projecte, es pot concloure que el mètode de validació proposat ha estat de gran ajuda per complir la funcionalitat per la qual es va proposar. Gràcies a que s'ha fet nombroses proves de cada etapa abans de donar-la per tancada, s'han pogut corregir un gran nombre de problemes quan encara es trobaven en un entorn relativament controlat. Si aquests *bugs* del programa haguessin avançat d'etapa junt amb la resta de codi vàlid, a mesura que s'afegissin funcionalitats noves s'haurien fet cada cop més difícils de detectar enmig del creixent nombre de línies de codi sense verificar.

Pel que fa a la metodologia *Scrum*, aplicar-la ha estat una bona tria. Gràcies a aquesta metodologia, s'han pogut esquivar en certa mesura un bon nombre de dificultats tècniques que han aparegut durant el projecte. Si bé s'ha hagut d'ampliar el termini d'entrega del projecte igualment, sense aquesta metodologia de reunions constants per pactar rutes alternatives amb el tutor, aquests problemes haurien pogut agreujar-se en forma d'hores de treball potencials perdudes i en possibles retards addicionals o retallades de funcionalitats.



3. Solució final

La part més complexa del projecte és l'el·laboració del codi que processa les dades inicials (un núvol de punts) fins a un resultat numèric, en aquest cas un valor que ens indica quin percentatge del forat de connexió amb la cantonada està lliure.

El primer pas és definir el llenguatge de programació que es fa servir en aquesta implementació. Continuant amb el que s'havia fet servir en les parts del sistema que ja estaven fetes, el llenguatge a utilitzar és C++, no tan sols per la comoditat d'estar en sintonia amb la resta del projecte, sinó perquè es vol fer servir un llenguatge eficient i robust.

El primer pas del procediment final ha estat agrupar tots els punts del núvol en rectes. És a dir, adonar-se de quins punts segueixen un patró lineal, que indica que en aquell moment s'ha estat visualitzant una paret. L'objectiu d'aconseguir aquestes rectes és tenir un sistema que relacioni els punts entre sí, per poder fer càlculs a partir d'aquesta informació. A més, aquestes rectes es construeixen en un primer recorregut de tots els punts que també s'aprofita per afegir més informació als punts, la més important de les quals és si formen part d'una discontinuïtat o no (és a dir, si hi ha una separació molt gran entre un punt i el punt que s'ha llegit just a continuació).

Per detectar aquests patrons s'han provat un seguit de mètodes abans d'escollir el que donava millors resultats.

- Comparació punt a punt: observar les posicions relatives entre un punt i els seus veïns immediats. Ràpid en càlcul, però propens a l'error.
- Regressió lineal: aquest mètode dóna la línia recta que més aproxima un conjunt de punts a base de minimitzar la suma de distàncies de tots ells a la recta.
- Transformada de Hough: aquest mètode genera un seguit de rectes per cada punt, i per cadascuna d'aquestes rectes es genera una nova recta perpendicular que passa per l'origen de coordenades. La recta que més s'ajusti a tots els punts serà aquella que tingui la recta perpendicular més semblant en tots. Aquest mètode és molt precís perquè considera totes les opcions possibles, però en conseqüència és molt intensiu en càlcul i s'ha considerat que no és la solució òptima per aquest projecte, ja que el software està pensat per poder-se executar en un temps raonable en ordinadors de baixes prestacions.



Un cop conegudes les rectes, es pot extreure la posició de les cantonades intersecant aquestes rectes. Durant aquest pas, validarem cada recta contra les altres per comprovar que siguin coherents, de manera que es puguin eliminar les que s'han generat incorrectament per manca d'informació d'entrada o la incorrectesa d'aquesta. Aquest pas garantirà que les dades restants siguin consistents entre elles pels càlculs posteriors, i que les cantonades que es detecten s'han generat a partir de rectes vàlides.

A partir d'aquestes cantonades es pot tenir una referència clara de a quina paret de l'embornal pertany un punt en concret, i això es pot fer servir per classificar les discontinuïtats que s'han detectat en el primer escombrat del núvol de punts. La paret on hi hagi més discontinuïtats és on resulta més probable trobar el forat.

Quan s'ha establert en quina paret es troba el forat, es pot procedir a recuperar les dades d'aquest. Per això es fa servir un algorisme RANSAC (*RANdom SAmple Consensus*), que és capaç de trobar en un conjunt de punts certes formes geomètriques i descobrir-ne les seves propietats. En aquest cas, aplicarem l'algorisme a tots els punts on s'han trobat discontinuïtats per trobar el centre i el radi de la circumferència que conforma el forat.

Finalment, un cop coneguda l'àrea del forat, que hauria d'estar tota buida, es busquen els punts que hi són dins. A partir d'aquests punts, es pot fer una estimació de fins a quina alçada està obturat el forat, i calcular quina àrea queda lliura.



4. Planificació temporal

La durada prevista del projecte és d'uns 5 mesos, des del gener fins al juny de 2016. Es tracta d'un període força extens, i el millor és planificar amb antelació la realització de les diferents parts del projecte per garantir que hi ha temps de fer-ho tot. Tot i així, aquesta planificació no és ni de bon tros estàtica: durant la realització del projecte poden sortir tota una sèrie d'imprevistos que s'han de tenir en compte, així que també s'han de preveure aquestes desviacions en la planificació i es proposa un pla d'actuació.

4.1 Especificació de les tasques

En aquest apartat, es dona una explicació detallada de cadascuna de les tasques que es duran a terme durant el projecte en ordre cronològic. Concretament, s'explica en què consisteix la tasca, les hores aproximades de realització i els recursos utilitzats.

4.1.1 Gestió del projecte

La primera part del treball es du a terme a l'assignatura de GEP. En aquesta assignatura es desenvolupa una primera part de documentació del projecte, en concret tota la que fa referència a la seva gestió. Aquesta etapa comporta unes 75h (3 ECTS) de feina, distribuïda en diversos entregables:

- Abast del projecte i contextualització: 24,5 hores.
- Planificació temporal: 8,25 hores.
- Gestió econòmica i sostenibilitat: 9,25 hores.
- Presentació preliminar: 6,25 hores.
- Plec de condicions: 8,5 hores.
- Presentació oral i document final: 18,25 hores.

Durada prevista: 75 hores.

Recursos: un ordinador, *Microsoft Office (Word i Excel)*, Atenea de la UPC, Racó de la FIB, *Google Drive*, una càmera de vídeo i l'aplicació *Gantter*.



4.1.2 Incorporació i familiarització amb l'entorn de treball

Un cop iniciat el projecte i abans de començar a implementar res, és important fer un primer pas d'adaptació i anàlisi de la part del projecte ja desenvolupada. En aquesta primera part, l'objectiu és familiaritzar-se amb els programes que es faran servir al llarg del desenvolupament del projecte, especialment pel que fa a l'entorn de programació (*VisualStudio*) i al de proves (*CloudCompare*), així com també amb el *hardware* ja desenvolupat.

Per fer-ho, un bon mètode és fer un petit estudi del codi ja desenvolupat per entendre el funcionament del *hardware* per així acostumar-se alhora a l'entorn de programació. Finalment, executar unes quantes proves i mirar resultats ajudarà a entendre l'entorn de proves.

Durada prevista: 20 hores.

Recursos: un ordinador del Departament de Robòtica, *VisualStudio*, *CloudCompare* i el *hardware* del projecte.

4.1.3 Implementació del projecte

Aquesta tasca es dividirà en diverses subtasques, que s'han de fer de forma ordenada perquè totes elles depenen d'una correcta realització de l'anterior. No obstant, no cal que el resultat en aquest punt sigui perfecte, tan sols funcional. Més endavant es realitzarà una fase de prova i millora on ja sí que es perfeccionarà tot. El temps estimat de la suma de tasques és de 170 hores.

a) *Detecció de rectes i cantonades*

La primera tasca a realitzar amb el núvol de punts proporcionat és analitzar-lo, descompondre'l en rectes i trobar les cantonades de l'embornal que representa. L'objectiu de detectar les cantonades com a pas inicial es tenir un fort punt de referència en l'espai representat. Un cop estiguin detectades, es poden situar la resta de punts a partir d'elles

Aquesta part es pot començar sempre haver acabat la tasca anterior, però cal tenir en compte que tot el temps que no s'hagi pogut invertir en familiaritzar-se amb l'entorn implicarà allargar la durada d'aquesta tasca.

Durada prevista: 75 hores



b) Detecció de plans

Amb les cantonades ja detectades, el primer que es podria fer és interpolar entre elles per obtenir uns plans preliminars. Suposant que les parets fossin exactament planes, aquest mètode ja seria suficient per trobar les parets. No obstant, no es pot fer això sense haver abans comprovat que les dades llegides pel sensor realment són parets planes. Per tant, aquesta tasca té una petita part de feina consistent en fer aquesta comprovació, i en ajustar els resultats en cas d'observar diferències significants.

Durada prevista: 25 hores

c) Localització del forat i nivell d'obturació

Un cop detectades les parets de l'embornal, caldrà detectar quina d'ells té el forat. Un cop estigui localitzat, es pot calcular quina és la seva superfície aproximada. El següent que caldria fer és tornar a revisar la informació dels punts presents a la regió del forat i veure si representen obturació de la canonada o no. Amb aquestes dades, ja es pot proporcionar un *feedback* a l'usuari indicant-li el percentatge d'obturació i si seria convenient fer una inspecció visual per corroborar-ho, o una neteja.

Durada prevista: 40 hores.

d) Reconstrucció en 3D

En aquesta darrera part, l'objectiu és millorar el *feedback* que rep l'usuari. Fins a aquest punt, el programa haurà recollit una pila de dades, però l'únic que haurà rebut l'usuari és una xifra indicant el percentatge d'obturació del forat. Per tant, es vol proporcionar també una retroalimentació visual en forma d'un model 3D que representi l'interior de l'embornal.

Durada prevista: 30 hores.

Recursos al llarg de la implementació: un ordinador del Departament de Robòtica, *VisualStudio*, *CloudCompare*, el *hardware* del projecte i *OpenCL*.



4.1.4 Fase de validació

Un cop el desenvolupament del projecte estigui complert, es passarà a una fase de validació. Aquesta fase consta de dos passos, validació i millora, que es faran de forma iterativa tants cops com sigui necessari fins que el resultat sigui prou bo. En la fase de validació, es faran proves de camp en embornals reals per comprovar els resultats obtinguts i prendre nota dels aspectes a millorar, ja siguin de *hardware* o de *software*. Posteriorment, es dura a terme una fase de millora al laboratori, on s'aplicaran els canvis que s'hagin decidit.

Un cop complerta una iteració, es començarà la següent. En cas que els resultats siguin els esperats, es procedirà a la fase final del projecte. En cas contrari, es repetirà el cicle fins que s'arribi als resultats òptims.

Durada estimada: 135 hores.

Recursos: un ordinador del Departament de Robòtica, *VisualStudio*, *CloudCompare*, el *hardware* del projecte, un ordinador portàtil i la xarxa de clavegueram de la ciutat de Barcelona.

4.1.5 Fase final

Un cop s'hagin aconseguit els resultats desitjats, la fase final del projecte consistirà en validar-los en un nombre més ampli d'embornals per recollir dades definitives. En aquest procés, s'espera que el sistema respongui bé, però si fos necessari, es retrocediria al pas anterior per acabar d'optimitzar-lo.

Un cop el sistema sigui el definitiu, es procedirà a la redacció de la memòria final i a preparar la seva presentació.

Durada estimada: 50 hores.

Recursos: un ordinador del Departament de Robòtica, *VisualStudio*, *CloudCompare*, el *hardware* del projecte, un ordinador portàtil, la xarxa de clavegueram de la ciutat de Barcelona, *Microsoft Office* i *Google Drive*.

4.2 Temps per tasca

La següent taula exposa un resum de la planificació del projecte:

Tasca	Temps requerit (hores)
Gestió del projecte	75
Incorporació i familiarització	20
Detecció de rectes i cantonades	75
Detecció de plans	25
Localització del forat i nivell d'obturació	40
Reconstrucció 3D	30
Fase de validació	135
Fase final	50
TOTAL	450

Taula 1. Càlcul d'hores per cada fase i en total.

4.3 Dependències de tasques

La següent taula clarifica les dependències de tasques, així com el seu ordre cronològic:

Tasca actual	Tasca prerequisite
Gestió del projecte	-
Incorporació i familiarització	Gestió del projecte
Detecció de rectes i cantonades	Incorporació i familiarització (no necessàriament completa)
Detecció de plans	Detecció de rectes i cantonades
Localització del forat i nivell d'obturació	Detecció de plans
Reconstrucció 3D	Localització del forat i nivell d'obturació
Fase de validació	Reconstrucció 3D
Fase final	Fase de validació

Taula 2: Dependències entre tasques.



4.4 Alternatives i pla d'acció

És molt possible que aquesta planificació no es pugui complir al peu de la lletra. Les previsions d'hores són tan sols una estimació, i no és fins estar a punt d'acabar-les que es pot garantir quan es trigarà, perquè pot haver-hi o bé dificultats d'implementació o bé al revés, una sobreestimació de la feina que s'havia de fer.

En tot projecte informàtic, aquesta desviació de temps sol ser un problema freqüent, i aquest no n'és una excepció. Per no obviar cap d'aquestes desviacions, s'anirà portant un registre constant i metòdic de les hores que es van invertint en cada etapa. En funció d'aquest registre, es pot conèixer la desviació entre planificació i realitat i actuar en conseqüència.

A continuació es proposa com resoldre els dos tipus d'incidències:

- Falta de temps per una tasca: depèn de la tasca que s'estigui realitzant. Si es tracta de la gestió del projecte o de la fase final, cal dedicar-hi més hores per complir el termini d'entrega. En cas que sigui l'etapa de familiarització, es pot començar la tasca següent i acabar allà la familiarització. Si la tasca que s'endarrereix és alguna de la implementació, a no ser que la desviació en la planificació sigui escassa es passarà a la següent tasca, i les imperfeccions es corregiran a la fase de validació durant les millores. El temps estimat per la fase de validació és generós, però si arribés a faltar-ne caldria dedicar-n'hi més.
- Excés de temps per una tasca: no representa realment un problema, sempre es pot començar la següent o aprofitar el temps restant per col·laborar a millorar el *hardware*.

Com es pot veure aquí, la única solució viable als retards temporals és invertir més hores en el projecte. Com que és un projecte que ha començat amb més temps del que està previst, també és previsible que acabi abans si es segueix perfectament la planificació. En cas que no, hi ha prou temps de marge com per fer aquestes hores extres per poder acabar sense problemes. En el cas en que ni això fos suficient, es dedicarà el màxim de temps possible a les tasques més clau, com s'ha especificat abans.

La metodologia *Scrum* que es segueix en aquest projecte juga a favor d'una planificació dinàmica: les reunions constants permeten detectar desviacions en la planificació abans que arribin a produir-se realment, i es podrà ajustar la planificació adequadament.



5. Pressupost

En primer lloc, es fa un càlcul del cost directe de cada tasca. Aquest cost es calcula a partir de les unitats necessàries de cada producte i el seu cost unitari, d'on s'extreu el seu cost d'amortització a partir d'una estimació de l'ús que se'n farà durant la seva vida útil i les hores d'aquest ús pertinents al projecte. No es té en compte el preu del clavegueram de la ciutat, que és públic. A més, molt del *software* usat és lliure i per tant gratuït. Pel que fa al cost del *hardware* específic del projecte, el preu total s'estima al peu de la taula per evitar posar la llista de components cada cop i que la taula sigui més clara. En el tema de recursos humans, l'únic empleat és l'estudiant d'enginyeria informàtica especialitzat en enginyeria de computadors que realitza el projecte, i s'estima el preu de la seva feina en 25 euros/hora, durant les 450 hores de durada prevista i que es distribuïran de la forma en què s'indica al diagrama de Gantt.

En els imprevistos s'ha considerat el pitjor dels casos, un 15% de probabilitats que s'hagin tant de fer un 25% d'hores extra com que s'hagi de canviar el 25% del *hardware* del projecte. Només es tenen en compte a les parts més costoses que poden portar complicacions tant en terme d'hores per ser les més complicades com en *hardware*, ja sigui perquè es detecti un error o es trenqui alguna peça.

En els costos indirectes, es tenen en compte el que costaran les factures de la llum i connexió a internet durant els 5 mesos de vida del projecte.

Sobre el total de costos directes, indirectes i imprevistos s'aplica un percentatge de contingència per si encara hi hagués algun altre imprevist. Com que el càlcul d'imprevistos ja s'ha fet de forma molt generosa, aquest percentatge de contingència és baix, d'un 5%. Sobre aquest cost apliquem l'IVA del 21% per tenir el preu estimat definitiu del projecte.

A partir de la planificació detallada al capítol anterior i del que es menciona en aquest apartat, es pot extreure la següent taula d'estimacions de costos.

Despesa	Unitats	Preu (euros)	Vida útil (anys)	Amortització estimada (euros)	Preu final (euros)
Costos directes					
Gestió de projectes					
Ordinador (torre + Windows + monitor + teclat + ratolí)	1	1400	5	6	6
<i>Microsoft Office</i>	1	100	4	2	2
<i>Atenea</i>	1	0	-	0	0
<i>Racó de la FIB</i>	1	0	-	0	0
<i>Google Drive</i>	1	0	-	0	0
Càmera de vídeo	1	100	5	1	1
<i>Gantter</i>	1	0	-	0	0
Recursos humans (informàtic)	75 hores	25	-	-	1875
Incorporació i familiarització					
Ordinador del departament (torre + 2 monitors + ratolí + teclat + Windows)	1	1050	5	3.2	3.2
<i>VisualStudio</i>	1	0	-	0	0
<i>CloudCompare</i>	1	0	-	0	0
Hardware del projecte*	1	200	0.5	60	60
Recursos humans (informàtic)	20 hores	25	-	-	500
Implementació del projecte					
Ordinador del departament (torre + 2 monitors + ratolí + teclat + Windows)	1	1050	5	27.2	27.2
<i>VisualStudio</i>	1	0	-	0	0
<i>CloudCompare</i>	1	0	-	0	0
Hardware del projecte*	1	200	0.5	510	510



Recursos humans (informàtic)	170 hores	25	-	-	4250
Imprevistos	-	1196.75, risc 15%	-	-	179.5
Fase de validació					
Ordinador del departament (torre + 2 monitors + ratolí + teclat + Windows)	1	1050	5	21.6	21.6
<i>VisualStudio</i>	1	0	-	0	0
<i>CloudCompare</i>	1	0	-	0	0
Hardware del projecte*	1	200	0.5	405	405
Ordinador portàtil	1	500	4	21.6	21.6
Clavegueram de Barcelona**	1	-	-	-	0
Recursos humans (informàtic)	135 hores	25	-	-	3375
Imprevistos	-	955.8, risc 15%	-	-	143.4
Fase final					
Ordinador del departament (torre + 2 monitors + ratolí + teclat + Windows)	1	1050	5	8	8
<i>VisualStudio</i>	1	0	-	0	0
<i>CloudCompare</i>	1	0	-	0	0
Hardware del projecte*	1	200	0,5	150	150
Ordinador portàtil	1	500	4	8	8
Clavegueram de Barcelona**	1	-	-	-	0
<i>Microsoft Office</i>	1	100	4	1.4	1.4
<i>Google Drive</i>	1	0	-	0	0
Recursos humans (informàtic)	50 hores	25	-	-	1250

Costos indirectes					
Llum	5 mesos	40	-	-	200
ADSL	5 mesos	25	-	-	125
Total acumulat					13122.9
Contingència		5% del total acumulat			656.1
Total sense IVA					13779
Total amb IVA (21%)		21% del preu total			16672.6

Taula 3: Càlcul de pressupost

* El *hardware* del projecte està constituït per dues *Arduinos* (25*2 euros) + 4 LEDs d'alta potència (10 euros) + cablejat (aproximadament 50 euros) + servomotor (30 euros) + 2 càmeres (500*2 euros) + material de construcció (30 euros de resina d'impresora 3D més 30 en la resta). En total, el preu ascendeix a uns 1200 euros. El seu període de vida s'estima en mig any perquè és un prototip.

**El clavegueram de Barcelona és una estructura pública i, com a tal, no representa cap cost la seva implicació en el projecte.

5.1 Control de gestió

Totes les desviacions que es puguin produir durant el projecte seran tractades de la forma en que s'ha especificat a l'etapa de planificació. Aquestes desviacions poden no ser tan sols en la planificació com ja s'ha tractat anteriorment, sinó també en pressupost.

Per calcular les desviacions, al final de cada tasca susceptible de patir imprevistos es calcularà una desviació per aquella tasca, de manera que tindrem una planificació i un pressupost reals. Amb aquests costos reals a la ma, podem mirar-nos els previstos, comparar i, en cas que la diferència sigui molt gran, caldrà preguntar-se per què ha passat això. En cas que s'hagi tractat d'un fenomen en gran part evitable, s'haurà d'evitar a les següents etapes.

Tot i així, pel pressupost del projecte s'ha calculat un llindar bastant superior a l'esperat, considerant una probabilitat de risc més gran de la que hauria d'existir. Per tant, és d'esperar que la desviació amb la planificació no sigui gaire, o que fins i tot s'hagin sobreestimat els costos reals.



6.Desviacions

Com ja s'ha comentat durant la planificació de tasques i costos, tot projecte està subjecte a possibles desviacions, i aquest no n'ha estat pas l'excepció. Al llarg de la durada del projecte s'han produït dos grans imprevistos, els dos causats per problemes de *hardware* externs al software que s'està fent.

La planificació de la primera fase, Gestió del Projecte, es va a dur a terme tal i com estava previst, fent totes les entregues exigides a temps. Pel que fa a l'etapa d'implementació, la detecció de rectes va comportar més hores de les previstes, ja que es van provar un cert nombre de mètodes no trivials abans de decidir quin era el més apte per arribar a la solució desitjada. No obstant, el temps extra invertit en aquesta fase de la implementació es recupera en passos posteriors, la detecció del forat i el càlcul de la seva obturació han portat menys hores del previst, gràcies en bona part a la quantitat d'informació extra recollida en la detecció de rectes. A més, la detecció de parets no ha estat necessària com a tal, i s'ha reduït a fer algunes comprovacions de coherència entre les dades, i s'ha suprimit la representació visual en 3D per falta de temps.

6.1 Danys greus al capçal

El primer gran obstacle pel projecte va aparèixer la segona setmana de maig. A aquestes alçades, el projecte es trobava llest per avançar a l'etapa de reconstruir el forat i detectar la seva obturació. Per poder detectar el forat, però, feia falta una lectura de dades amb més qualitat que les dades amb les que s'havia treballat fins ara. El problema radica en un error a l'hora de tractar el servomotor anterior a l'inici del projecte (es trobava ja en el treball previ). El motor està programat de tal forma quan se li indica un nombre de graus a girar i una direcció realitza aquest moviment en un cert temps. En el codi, però, s'assumia que el motor té una capacitat de gir de 180°, i observacions posteriors han demostrat que no és així, sinó que el rang efectiu d'angles per on es pot bellugar es troba entre 10 i 175°. Fora d'aquest rang, el motor s'encalla i no és capaç de girar més, però el codi donava per suposat que el moviment s'havia completat, i la següent passada començava desfasada un cert angle, provocant que les lectures d'anada i tornada tinguessin angles diferents. Aquest desfasament fa impossible detectar la circumferència del forat amb el mètode proposat, perquè les discontinuïtats de les lectures es troben en angles diferents i no es visualitzen en forma de cercle com haurien.

Corregir l'error del motor hauria resultat una tasca senzilla, però des de l'inici del projecte el sistema *hardware* havia patit algunes modificacions importants en la seva estructura física. Això significa que la cal·ibració inicial era incorrecta, i que per poder prendre noves lectures calia no només corregir el problema d'angles amb el motor, sinó també recal·ibrar les càmeres per prendre les mesures correctament per la nova configuració del capçal.

Aquesta primera setmana de maig tocava doncs començar un apartat que no s'havia previst inicialment, la recal·libració del sistema, que ja es donava per feta. Abans de poder-la començar, però, un incident extern va destruir el capçal. Concretament, una exposició perllongada a altes temperatures va fer que l'estructura feta per impressora 3D es dilatés, deformés i trenqués finalment, deixant el capçal inutilitzable. Després de consultar-ho amb el tutor, es va decidir aprofitar aquesta aturada per començar amb la documentació, que era la única cosa que es podia fer en paral·lel, per no perdre completament els dies fins que el sistema tornés a estar operatiu. Els responsables del projecte van decidir aprofitar la reconstrucció del capçal per afegir-hi noves modificacions, cosa que va fer que el període de reparació s'allargués fins la primera setmana de juny.

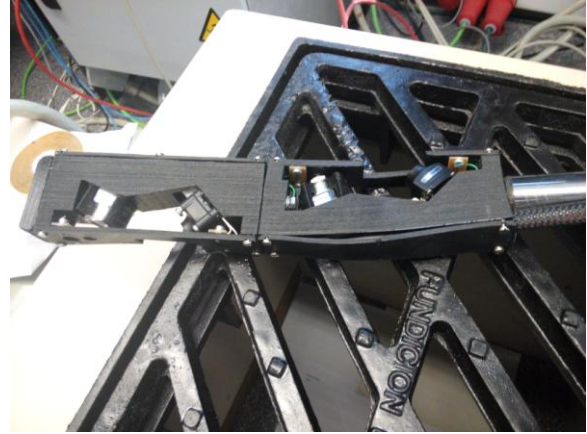


Figura 5. La deformació del material constructiu del capçal feia impossible el seu ús i va forçar un desviament de la planificació durant la seva reparació.

Arrel d'aquest incident, es va prendre la decisió d'allargar l'entrega d'aquest treball fins l'octubre del quadrimestre següent.

6.2 Cal·libració i generador de dades

El projecte es va reprendre la primera setmana de juny, havent fet una pausa per les últimes setmanes lectives del quadrimestre, ja que pel projecte s'havia demanat una ampliació. Durant el mes de juliol, es va estar fent la cal·libració de les càmeres, que va resultar ser una tasca molt més feixuga del que s'havia previst en un principi.

A principis d'agost, la cal·libració ja estava acabada i revisada, però va aprèixer un nou imprevist: l'*encoder* de la base havia deixat de funcionar correctament en algun moment, i tot i que les lectures estàtiques es podien fer correctament i amb un marge d'error bastant baix, en afegir la lectura de la posició vertical de l'*encoder* aquesta es feia de forma incorrecta i impedia la possibilitat de fer un escanejat.

Aquest nou imprevist tornava a fer impossible fer un escanejat per detectar el forat i la seva obturació, però en ser agost era previsible que trigaria tant o més temps en resoldre's que l'afer anterior. Era imperatiu acabar la part pràctica del projecte abans del dia 1 de setembre per motius de falta de temps a partir d'aquest dia, així que es va pactar amb el tutor una solució alternativa: generar dades artificials.

Inicialment, es va preveure la realització de les dades artificials com un pas per poder tornar ràpidament a la implementació principal, però aquest desviament es va allargar una mica per acabar creant un generador de dades parametrizable i el més fidel possible a la realitat.

La implementació es va reprendre a meitat d'agost, i es va acabar el dia 24.

La taula amb el temps per activitat revisada és la següent:

Tasca	Temps requerit (hores)
Gestió del projecte	75
Incorporació i familiarització	20
Detecció de rectes i cantonades	140
Fase final 1	25
Cal·libració	70
Generador de dades	30
Localització del forat i nivell d'obturació	40
Fase de validació	30
Fase final 2	35
TOTAL	465

Taula 4: Planificació temporal revisada.

S'han eliminat els apartats *Detecció de plans*, *Reconstrucció 3D* i *Fase de validació*, i en el seu lloc s'hi ha afegit Cal·libració del sistema i Generador de dades. També s'han modificat els temps de Detecció de rectes i Cantonades, Localització del forat i nivell d'obturació i la separació de la Fase final (redacció de la memòria) en dues parts.

En total, tot i que el projecte s'ha allargat 3 mesos, les hores de feina total han incrementat poc, ja que el retard va ser degut a una interrupció forçada i no a falta de temps per acabar la funcionalitat principal.

7. Sostenibilitat

Per tot projecte cal realitzar un estudi de sostenibilitat. Aquest estudi es fa seguint el mètode socràtic, és a dir, plantejant-se i responent un seguit de preguntes sobre el projecte. Aquestes preguntes valoren 3 vessants diferents del projecte: econòmica, social i ambiental. Cadascun d'aquests aspectes es valora de forma independent, i se li assigna una puntuació en funció de les respostes a les preguntes que hi fan referència. Aquestes puntuacions van de 0 (un projecte gens sostenible) a 10 (un projecte totalment sostenible).

Les puntuacions assignades a aquest treball després de respondre les preguntes són les següents:

Econòmica	Social	Ambiental
8	7.5	6.8

Taula 5: Matriu de sostenibilitat

7.1 Dimensió econòmica

- L'avaluació dels costos estimats s'ha fet de forma exhaustiva a l'apartat de *Pressupost* d'aquest document.
- Durant les etapes de validació i final especificades a la planificació del projecte, es pretén garantir-ne la robustesa i poder assegurar que funcionarà en qualsevol embornal. Per tant, no hauria de ser necessari fer-hi actualitzacions a no ser que es produís algun canvi molt significatiu en l'entorn de treball, com ara que els embornals es construïssin seguint uns patrons diferents. Serà necessari mantenir el *hardware* i possiblement reparar-lo, tot i que les parts crítiques (*Arduinos* i sensors) estan fortament protegides per evitar-los danys, així que en gran part serien reparacions superficials i poc costoses.
- El cost del projecte està força ajustat, i podria ajustar-se encara més si sorgís un competidor.
- Els recursos emprats són molt pocs: gran part del *software* és lliure, el clavegueram és públic i el cost del *hardware* és el mínim. On es podria retallar és en temps, si es contractés un enginyer amb més experiència que, encara que cobrés més per hora, li hagués de dedicar menys hores i garantís menys imprevistos.



- Com està detallat a la planificació, es dedica el gruix principal de feina a desenvolupar les tecnologies específiques en les que es basarà el projecte.
- El projecte ha estat sol·licitat específicament per l'empresa de FCC, que el faran servir en els seus manteniments rutinaris.

7.2 Dimensió social

- El país està desenvolupat, hi ha un estat del benestar, hi ha una democràcia i s'inverteix en els sectors d'investigació perquè són candidats a aportar noves fonts d'ingressos.
- El projecte no hauria d'afectar aquest equilibri, ni positivament ni negativament.
- Tenint en compte que ha estat demanat explícitament, podem concloure que n'existeix la necessitat.
- Si entenem com a consumidors la població que es beneficia del sistema de clavegueram, no veuran una millora directa en la seva qualitat de vida, però és altament probable que puguin observar que hi ha menys problemes amb el clavegueram (males olors, embornals inundats, etc.) degut a un sistema més eficient d'inspecció i prevenció.
- Farà la tasca dels operaris que s'encarreguin del manteniment molt més àgil, fàcil i descansada, i per tant es podran fer inspeccions més regularment.
- Existeix la possibilitat que FCC aprofiti el resultat del projecte per retallar personal i fer la mateixa feina amb menys gent. No obstant, aquesta seria una conducta poc ètica i molt criticada, amb la qual cosa és poc probable que es prengui aquest camí. A més, algú haurà d'encarregar-se de l'esmentada feina de mantenir els aparells, que podria ajudar a simplement reubicar operaris.

7.3 Dimensió ambiental

- Els recursos *hardware* tindran un cost associat de fabricació, que es pot pal·liar si es reutilitzen components. També hi ha associat un cost en llum durant la producció, que tindrà un efecte en el medi depenent de com hagi estat produïda.
- Actualment és una tasca manual, sense pràcticament gens de cost material ni energètic.
- Tot el material electrònic podria reaprofitar-se d'altres aplicacions.



- Més enllà del cost dels productes, l'únic cost durant el desenvolupament és l'energia consumida per l'ordinador on es desenvolupa el software i la que consumeixen les impressores 3D que fabriquen les peces de la carcassa del projecte.
- Tots els components es poden separar dels altres per reciclar-se o reutilitzar-se més fàcilment.
- Les plaques *Arduino*, el material manufacturat principal del projecte, es fan a Scarmagno, Itàlia, en bones condicions de treball. De la resta de components electrònics és més difícil saber-ne l'origen, tot i que s'intentarà tendir cap a productes creats de forma ètica. Pel que fa a les peces estructurals, es fan al propi departament en impressores 3D.
- La petjada ecològica augmentarà degut a un nou producte en fabricació, tot i que s'intenta minimitzar els costos que comporta.
- Tots els components electrònics que no estiguin danyats es poden reaprofitar per altres aplicacions.

7.4 Lleis i regulacions

A l'hora de realitzar el projecte, cal tenir en compte la legalitat vigent per assegurar-se que no es fa alguna cosa que entri en conflicte amb la llei. La principal preocupació seria que no estigués permès introduir al clavegueram elements que no hi estan destinats. Revisant el Decret 130/2003^[8] del Reglament de Serveis Públics de Sanejament, es pot veure que en tota l'extensió del document no es fa referència a una prohibició d'aquesta activitat. El que sí que hem de tenir en compte és el contingut de l'annex I del decret. En aquest annex, es llisten totes les substàncies que està prohibit abocar al clavegueram. En el cicle d'ús del sistema de reconeixement no hi hauria d'haver mai cap abocament intencional, però sempre pot haver-hi algun accident, pel qual cal assegurar-nos que no es fan servir productes nocius per la construcció del sistema, especialment del capçal (la part que podria ser més susceptible de desprendre's de l'aparell i caure al clavegueram).

8. Incorporació i familiarització

L'objectiu d'aquesta primera etapa era conèixer l'entorn de treball i tenir una primera visió del problema i de com resoldre'l. En el temps destinat aquesta etapa, he tingut un primer contacte amb les eines de treball (Microsoft Visual Studio, CloudCompare i el hardware del projecte), i he estudiat també el codi que ja estava fet a l'inici del projecte.

8.1 Microsoft Visual Studio

Microsoft Visual Studio és una entorn de desenvolupament integrat (*Integrated Development Environment*, o *IDE*) específic de les plataformes *Windows*, i pensat per programar aplicacions que corrin sobre aquest sistema operatiu. Dels molts llenguatges que permet fer servir, en aquest projecte es farà servir C++.

Aquest entorn inclou múltiples funcions per programar amb comoditat: té una vista de tots els arxius del projecte per navegar ràpidament, un editor que ajuda a completar codi automàticament basant (per exemple, autocompletant noms de funcions d'una classe), un dissenyador d'interfícies gràfiques molt intuïtiu i que permet generar codi automàticament i un *debugger* molt complet amb el que es pot controlar l'execució amb un gran nivell de detall i proporciona gran quantitat d'informació.

En aquest projecte, es fa servir la versió *Visual Studio 2013*.

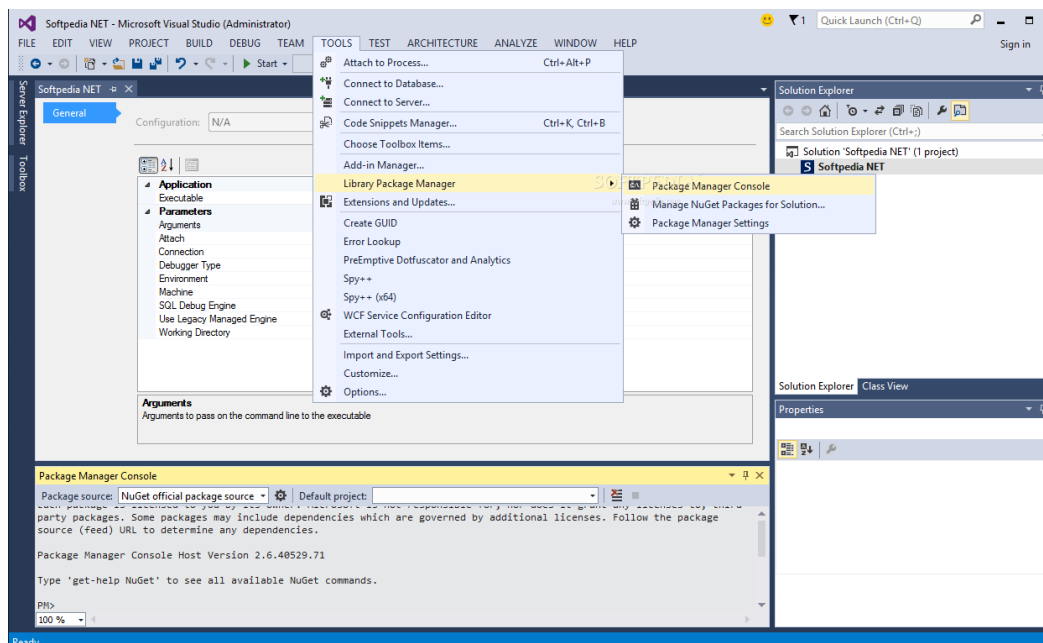


Figura 6. Vistes principals de Microsoft Visual Studio.

8.2 CloudCompare

Com ja s'ha mencionat anteriorment, una eina recurrent en aquest projecte serà la llibreria *Point Cloud Library*, que permet gestionar núvols de punts en l'espai. Aquest tipus d'estructures, anomenades *PointClouds*, es poden guardar mitjançant una funció de la llibreria en arxius amb l'extensió .pcd.

El software *CloudCompare* és una eina per processar punts en 3D que va néixer el 2003 com a doctorat de Daniel Girardeau-Montaut, a l'École Nationale Supérieure des Télécommunications, de París. Aquest projecte va començar com una eina per reconèixer fluctuacions en núvols de punts densos captats a través de potents escàners làser, i va anar evolucionant fins a convertir-se en una eina força genèrica per tractar núvols de punts. Permet carregar arxius de diversos formats, entre els que destaquen els arxius .pcd, els que a nosaltres ens interessien.

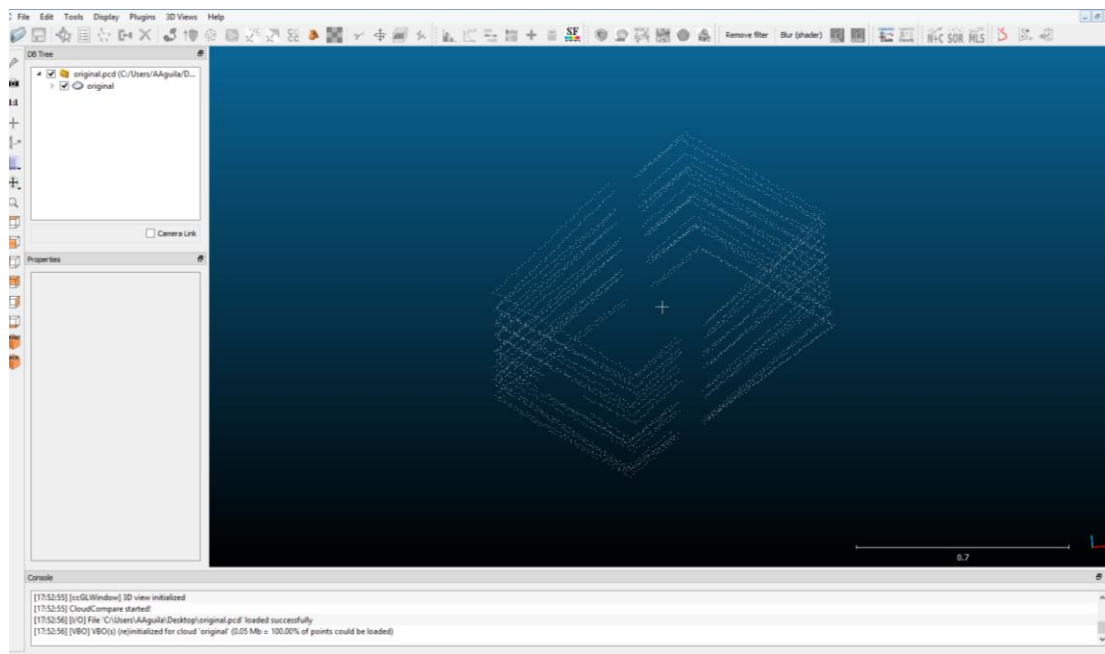


Figura 7. Entorn de CloudCompare.

Tot i que aquest software permet fer moltes coses a nivell de tractament de punts, en aquest projecte només es farà servir a nivell molt bàsic com a visualitzador interactiu d'aquests arxius .pcd per veure el progrés de les diverses etapes.

La versió utilitzada és la *CloudCompare* v2.6.1 de 64 bits.

8.3 El hardware

Per aquest projecte s'ha construït un *hardware* específic orientat a la tasca concreta que volem realitzar. Aquest *hardware* és una perxa pensada per poder introduir-se per les reixes de la tapa de l'embornal, permetent així fer l'exploració sense necessitat d'obrir aquesta tapa (procés sovint costós o impossible). Per facilitar-ne l'ús, compta també amb una base que es pot recolzar sobre la tapa. Aquesta base beneficia l'operari en proporcionar-li un punt de suport, i alhora millora la qualitat general de l'escanejat perquè evita molts errors induïts per moviments bruscos de la perxa, el pols de l'operari o qualsevol altre factor que pugui embrutar les dades amb lectures errònies.

Tan la base com la perxa compten cadascuna amb una sèrie de components *hardware* i un microcontrolador que els regula i que farà de pont entre els sensors i l'aplicació per l'ordinador. En aquest projecte, els microcontroladors utilitzats seran dues plaques *Arduino*. El microcontrolador de la base controla un *encoder* que llegeix el moviment vertical de la perxa. A mesura que l'usuari la mogui amunt i avall, aquest *encoder* anirà indicant l'alçada actual del capçal de la perxa, necessària per calcular la posició dels punts llegits.

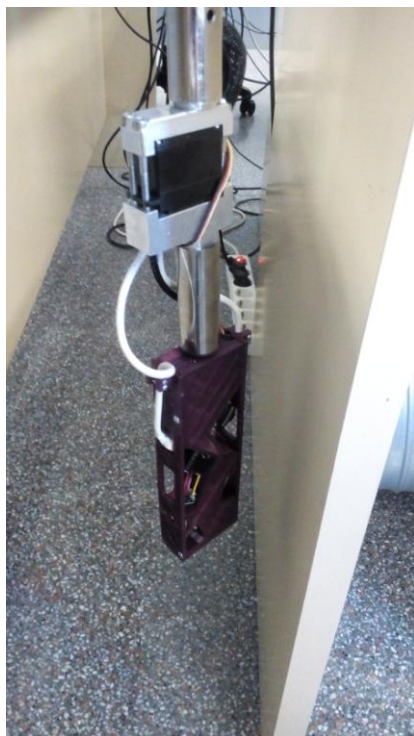


Figura 8. Aquí es poden veure el servomotor i el capçal del sistema.

La perxa és una barra d'alumini que té un segon microcontrolador instal·lat, encarregat de dues tasques vitals. La primera d'elles és transmetre al servomotor de la perxa les ordres de moviment enviades per l'aplicació. Aquest servomotor connecta la perxa amb el capçal on es troben els sensors, i permet que aquest roti en un angle de 180 graus. Com que la rotació permesa no arriba a 360 graus, el capçal té dues càmeres orientades en direccions oposades, de tal manera que en un gir complet del servomotor, cada càmera escombra 180 graus diferents als de l'altra, donant una lectura combinada de 360 graus. Cada càmera té també dos *LEDs* col·locats al costat, un per il·luminar l'interior de l'embornal en cas de voler-se explorar visualment i un d'alta potència que permet binaritzar fàcilment el que detecta la càmera per trobar un punt concret en la imatge. Fent una cal·libració experimental, es pot trobar una relació entre la posició d'aquest punt en la imatge i la seva posició en coordenades reals. El microcontrolador de la perxa també és responsable de controlar aquests *LEDs*, a diferència de les càmeres que van directament connectades a l'ordinador via *USB*.



Cal destacar també que tot el sistema s'ha intentat construir verticalment per permetre el seu pas pel major nombre d'embornals possibles, ja que si fos molt ample no entraria per les reixes més estretes. En conseqüència, les càmeres i els *LEDs* del capçal no estan col·locats de forma simètrica, sinó que una càmera està per sota de l'altra.

8.4 El codi

A l'iniciar aquest projecte, hi havia una certa quantitat de codi el·laborat. Aquest codi es manifestava en tres arxius diferents: el codi *Arduino* de la perxa, el codi *Arduino* de la base i un primer codi bàsic en C de l'aplicació per ordinador.

Els dos codis *Arduino* són força simples, i l'únic que fan és rebre comandes i retornar lectures. Les dues llegeixen comandes enviades pel port sèrie, activen o desactiven opcions en funció de la comanda rebuda, i retornen pel mateix port les lectures de posició de l'*encoder* i el servomotor.

Pel que fa al codi de l'aplicació, gran part de codi eren funcions per fer que en prémer un cert botó s'enviés la comanda apropiada a l'*Arduino*. Per exemple, en prémer el botó d'augmentar la velocitat del servomotor, s'envia la comanda apropiada per fer això.

La funció principal d'aquesta part ja feta és la lectura de les càmeres. Fent la binarització abans esmentada, es pot localitzar en tot moment la posició d'un punt lluminós en la imatge, i coneixent la posició del capçal es poden saber les coordenades d'aquell punt. Aquesta transformació es fa mitjançant un polinomi que va ser extret de forma experimental, a base de comprovar diverses distàncies, mirar la posició del punt i el capçal i relacionar totes les dades. Si en algun moment cal recalibrar les càmeres, aquest és el mètode que es faria servir.

9. Implementació

Gran part del gruix del projecte recau en aquesta etapa. L'objectiu final és, un cop acabada, disposar d'un codi que s'espera que pugui tractar qualsevol lectura d'un embornal real d'una manera prou precisa. Evidentment, l'entorn de proves del laboratori presenta molts menys obstacles que un embornal real. És per aquest motiu que un cop el resultat sigui acceptable, s'havia plantejat una etapa de proves de camp i millores per adaptar el dispositiu a un entorn real, problemes de la qual es tractaran en detall més endavant, però com ja s'ha explicat en l'apartat 6. *Desviacions*, ha estat impossible dur a terme aquestes proves per falta de temps i de preparació *hardware*.

9.1 Detecció de rectes

Si ens fixem en una lectura com en la de la figura, es pot veure que les lectures de cada càmera (separades pels espais en blanc) segueixen un patró similar a una mitja espiral: es comença a llegir enmig d'una paret, i cada càmera s'acosta en línia recta a una cantonada, ressegueix la nova paret que ha trobat fins la següent cantonada i continua per aquesta última paret fins que s'acaba el recorregut que permet fer el servomotor. Un cop arriba a aquest punt, el motor gira en direcció contrària i es repeteix el patró en sentit contrari.

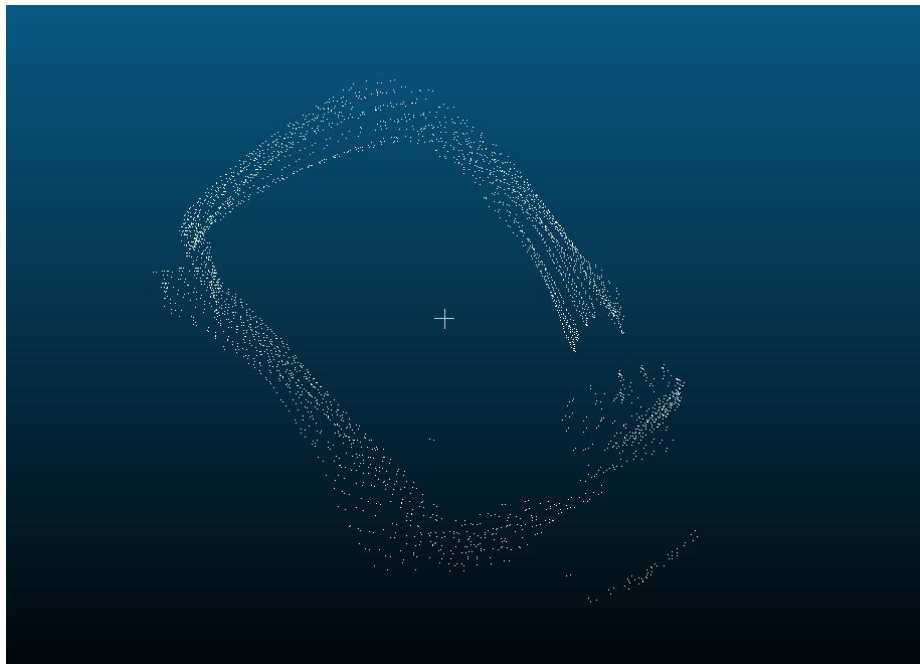


Figura 9. Lectura d'exemple on es poden veure la separació entre les lectures de les dues càmeres i el patró en mitja espiral.



El primer pas de la implementació serà classificar tots els punts en funció de la paret a la que pertanyin. Això implica que anirem dividint el conjunt de punts en rectes, entenent com a canvi de recta una cantonada entre dues parets. L'objectiu d'aquest procediment és facilitar la detecció del forat de la canonada: si es poden localitzar discontinuïtats notables entre punts, i sabem de quina paret formen part aquests punts, se sap a quina paret es troba la discontinuïtat. Repetint això per totes les discontinuïtats que es trobin, podrem saber on és el forat només amb buscar quina paret té més discontinuïtats en unes coordenades similars per totes les rectes.

Al llarg d'aquest procediment de detecció de rectes, cal destacar que obviarem la coordenada vertical z dels punts, per simplicitat. Per cada gir del motor, el capçal baixa molt poc, però en canvi es llegeixen punts que varien molt en coordenades horitzontals. Per tant, com que la z no aporta informació en aquesta primera part, no la tindrem en compte per estalviar temps de càlcul.

Per trobar aquestes rectes, s'ha anat provant l'eficàcia de diversos algorismes per veure quin donava el resultat més òptim. A l'estudi previ s'ha parlat de la transformada de Hough com a possible algorisme per resoldre aquest problema. Si bé és cert que la transformada dona el millor resultat, és un càlcul molt costós en temps i en memòria. L'aplicació que s'està desenvolupant s'executarà en portàtils especialitzats per treballar en entorns hostils, que són més cars que un portàtil estàndard amb un hardware equivalent perquè s'inverteix una quantitat notable de recursos en fer-los resistents a l'aigua, a la pols, als cops, etc. Per tant, és d'esperar que el pressupost que l'empresa destinarà a comprar portàtils per fer anar el sistema limiti la qualitat d'aquests a nivell *hardware*, i això fa que la transformada de Hough pugui convertir-se en un càlcul molt lent, tenint en compte les limitacions del portàtil i la gran quantitat de punts a tractar, i això entra en contraposició directa amb l'objectiu del projecte de tenir un sistema ràpid i eficient. És per aquest motiu que s'ha decidit explorar altres opcions que, si bé no s'espera que donin el resultat més bo possible, podrien fer una aproximació amb un error acceptable a canvi de trigar molt menys, efectivament intercanviant una mica de precisió a canvi de guanyar molt en velocitat.

9.1.1 Càlcul de distàncies mínimes

Previ

Per poder analitzar el nostre núvol de punts, la primera decisió que s'ha de prendre és com l'emmagatzemarem. La rutina que fa l'adquisició dels punts durant l'escaneig guarda un arxiu de text amb un format concret, que consisteix en posar 4 números per línia. De la segona a la quarta posició hi ha les coordenades x , y i z d'un punt, mentre que la primera posició té un valor enter de 0 a 3, segons si és una lectura de la càmera inferior (0 o 1) o superior (2 o 3), i de la direcció en la que es movia el servomotor quan s'ha fet la lectura.

Les dades del fitxer estan generalment intercalades entre càmeres (tot i que pot donar-se el cas de més d'una lectura seguida d'una sola càmera), però tenim la garantia que totes les lectures de la mateixa càmera estaran en ordre.

Partint d'aquest format concret, la forma més simple d'organitzar les dades pel seu anàlisi és crear dos conjunts diferents, un per les lectures de cada càmera, i anar-los assignant les lectures d'aquella càmera en l'ordre que apareixen, de manera que a cadascun tindrem la meitat de l'embornal en ordre d'escanejat.

Mètode

Una primera aproximació per trobar quins punts són cantonada és buscar els mínims de distància entre punts. La idea d'aquest mètode és aprofitar que el camí més curt entre dos punts és sempre una recta. Si per cada punt i calculem la seva distància respecte el punt $i + n$, tindrem una taula de distàncies entre punts. Pel teorema de Pitàgores, sabem que un punt que estigui a, per exemple, 5 cm en línia recta d'un altre, està més lluny que un altre punt que es trobi a 2 cm en línia recta i 3 cm en una altra direcció, perquè hi haurà una diagonal més curta que dugui fins a aquell punt directament. Aquest concepte és el que es vol aprofitar, intentant trobar els punts i on la distància amb el punt $i + n$ sigui mínima, cosa que ens indicaria que a mig camí entre els dos hi ha una cantonada. A partir de la cantonada, podríem classificar els punts que hi ha a banda i banda com a rectes diferents.

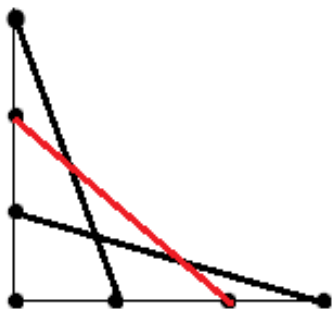


Figura 10. Esquema del mètode 1.

Per fer el mètode més visual, ens podem fixar en el següent esquema. Les rectes representen la distància entre un punt i i el que es troba a una n de 4 punts de distància. Suposant que tots els punts són equidistants i es troben a una distància d'1cm, les dues rectes negres tenen una longitud de $\sqrt{1^2 + 3^2} \approx 3.16$, mentre que la recta vermella té longitud $\sqrt{2^2 + 2^2} \approx 2.83$, i la cantonada és just el punt que es troba al mig de l'interval.

Resultat

El mètode té un funcionament molt dolent, per dos motius de pes. El primer és que no tots els punts es troben a la mateixa distància els uns dels altres, sinó que en lloc d'això, els més propers a les càmeres estan més junts degut a que el motor comença a frenar una mica abans de fer el canvi de sentit. Això fa que les distàncies entre aquests punts siguin similars a les distàncies entre els punts que tenen una cantonada entremig, i per tant es detecten com a tals. El segon és que com que el nombre de punts que es fan servir cada vegada és molt petit, el mètode és molt sensible al soroll, i si ampliem la mida de la finestra de manera que n sigui un nombre de punts molt gran, l'impacte de les cantonades en les distàncies es suavitzava massa i es fan gairebé impossibles de distingir. Aquest mètode ens retorna les cantonades amb una precisió acceptable, però la quantitat de falsos positius és inacceptable, per la qual cosa queda descartat.

9.1.2 Càlcul de desviacions

Mètode

El segon mètode és un derivat directe del mètode anterior, però en lloc de fixar-nos en les distàncies entre els punts que hi ha als extrems de la nostra finestra de mida n , estarem comparant angles entre rectes. El que s'espera veure ara és que els punts que estan en línia recta presentin una diferència d'angle mínima, mentre que quan hi hagi una cantonada, la recta que va en una direcció anirà canviant d'angle fins a estar rotada 90 graus, i intentarem detectar aquesta rotació mentre es produeix.

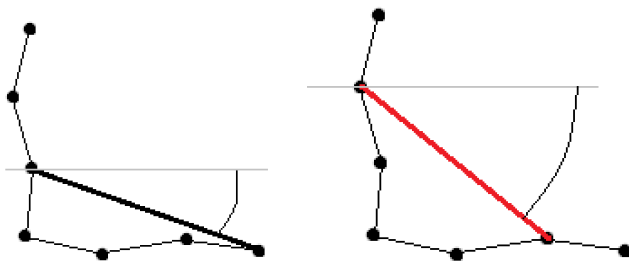


Figura 11. Esquema del mètode 2.

En la figura es representa un conjunt de punts llegit amb imprecisions. Amb la mateixa n que abans, en el cas de l'esquerra l'angle entre dos punts qualsevol és molt més petit que l'angle entre els dos punts que tenen la cantonada just enmig.

Resultat

El problema de la distància variable entre punts que teníem abans queda pal·liat, però seguim tenint un problema amb el soroll. Si n és un valor petit, petites desviacions donen grans errors, i si és un valor gran, la variació d'angle a detectar és tan petita que qualsevol variació deguda a un error també es detecta com a cantonada.

9.1.3 Vector de diferències

Previ

En lloc de guardar totes les lectures d'una càmera en un sol conjunt, formant una mena de mitja espiral amb tots els punts, és més interessant dividir aquests punts en plans. El conjunt de punts d'una càmera ara passarà a estar format per un nombre determinat de plans, que representen cadascun una recorregut complet de la càmera en un sentit d'extrem a extrem del seu recorregut. Quan canvia de direcció, donem el pla per acabat i comencem a generar el següent. D'aquesta manera, seguim tenint la possibilitat de fer servir les dades com abans si simplement ens mirem els plans en ordre, però s'obren noves portes com ara guardar dades addicionals per cada pla, tractar-los com si fossin conjunts en dues dimensions sense variació d'alçada, etc.



Les lectures de les càmeres presenten els punts ordenats per moment de lectura, així que estan consecutius, i amb cada punt arriba un camp que indica en quin sentit estava movent-se el motor quan s'ha llegit aquell punt. Per tant, aquesta classificació en plans és directa amb una sola lectura del conjunt de dades.

Mètode

Partint del càlcul de la diferència de pendents, que ens ha donat millors resultats que la primera versió, intentarem fer un algorisme més refinat que aprofiti aquesta idea. El mètode proposat genera dos vectors per cada pla, un de diferències en x i un altre en y . Aquests vectors contindran, en la seva posició i , la distància en x o y entre el punt i i el punt $i+n$. Considerant el cas genèric en el que l'escanejat ha començat en una paret i no en una cantonada, totes les diferències haurien de tenir el mateix signe fins arribar a una cantonada en la que aquest canvia, i això es compleix, idealment, per les dues cantonades, una per vector. Si l'escanejat ha començat en una cantonada, només una és visible en el pla, i per tant un dels vectors no tindrà canvi de signe.

Resultat

Aquest mètode pateix d'una sensibilitat a l'error comparable a la versió anterior. El podem fer més robust si considerem com a lectures errònies totes les posicions del vector amb un signe diferent de les seves posicions properes, però no acaba de funcionar a la perfecció. Si considerem només el signe de les posicions a distància 1, segueixen sense filtrar-se els errors que es propaguen a més d'un sol punt. Per altra banda, si aquesta distància s'amplia massa, podem passar per alt una cantonada que estigui molt propera al final del pla.

9.1.4 Algorisme final

Fins ara, tots els algorismes observaven la posició relativa entre dos punts, i després comparaven el resultat amb aquest mateix càlcul per uns altres punts. Aquesta doble comparació és el principal motiu pel qual tots aquests mètodes han estat tan sensibles a les dades errònies, i després de moltes millores successives, ni la última versió donava un bon resultat. S'hi ha acostat força, però tot i així hem seguit sense poder evitar errors. Per tant, després de reflexionar sobre els resultats que s'han anat veient, la conclusió és que una opció raonable seria provar alguna cosa radicalment diferent, que tot i que segurament serà més costosa en temps, eviti els problemes que hi ha hagut fins ara.

Mètode

Per evitar la vulnerabilitat a l'error que sorgeix d'analitzar per separat grups petits de dades, el nou algorisme pretèn tractar tots els punts d'un pla com una sola entitat. Cal tenir present que l'algorisme resultant serà més costós en temps perquè haurà de tractar un volum de dades molt més gran de cop, però cal implementar-lo per aconseguir una millora de precisió necessària.

La nova idea és localitzar a priori tots els punts candidats a formar part d'una recta, i aplicar sobre aquest conjunt una regressió lineal. Per tant, dividim l'algorisme en dues parts: criba de punts i regressió sobre aquests punts.

Per seleccionar els punts sobre els que farem la regressió, seguirem mantenint la idea dels pendents similars que hem fet servir en els altres mètodes. Aquests pendents els seguirem calculant en funció dels punts propers, però aquest cop els farem servir per construir un histograma. En aquest histograma, que ens mostrarà com estan distribuïts els pendents relatius entre punts, hauríem de veure-hi dues grans acumulacions de punts, que es correspondran a les dues rectes que estem buscant en cada pla. A més, sabem que aquests pendents hauran de ser clarament disjunts, perquè una cantonada marca un canvi de signe al pendent.

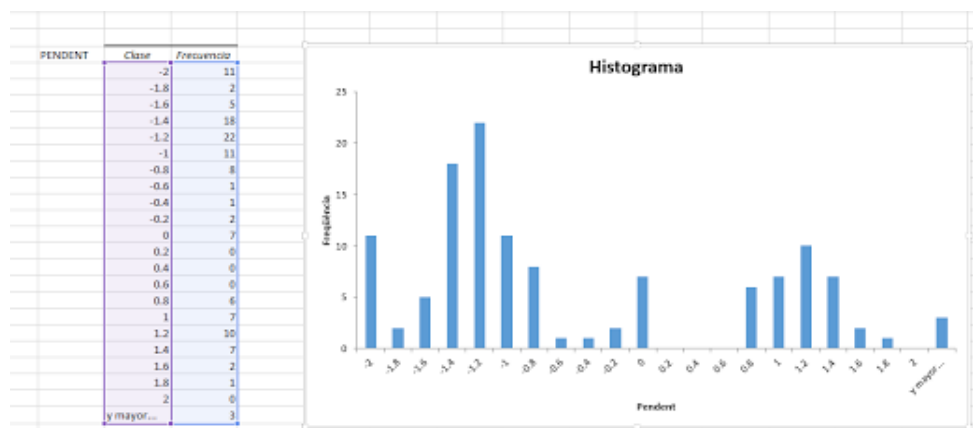


Figura 12. Gràfic resultant de classificar els pendents d'un pla en bins d'amplada 0,2. Es poden veure dues concentracions als pendents majoritaris.

A la figura es poden veure clarament dues zones on s'acumulen els pendents, una amb pendent negatiu i una amb pendent positiu. D'aquest histograma en podem treure la conclusió, per exemple, que el costat més llarg d'aquest pla té un pendent al voltant de -1,2. Si agafem els punts que tenen el pendent majoritari o els pendents immediatament superior i inferior, tindrem un conjunt de punts que en la seva gran majoria pertanyeran a la recta més llarga. Evidentment, no tindrem tots els punts de la recta, ni tots seran de la recta, però si la majoria, i ens donaran una bona idea del que és la recta.

Un cop descartats els punts amb el pendent majoritari i semblants, queda una segona regió amb menys punts, però pendent de signe oposat, que sabem que pertany a la segona recta del pla. Per tant, també ens quedem aquests punts per fer una segona regressió lineal. Seguint amb el cas de l'exemple, la segona recta té un pendent al voltant de 1,2.

Després d'aquest primer càlcul, tindrem dos conjunts de punts que tindran els punts amb pendents iguals que els de les rectes, però abans de processar-los, cal eliminar valors erronis. Com que amb aquests punts es farà una regressió lineal, un sol *outlier* no és un problema rellevant. El que volem eliminar aquí és una tercera recta que, si bé és cert que existeix, no volem tractar. Aquesta recta apareix perquè la majoria de lectures no es comencen a prendre en una cantonada, i els plans acostumen a estar formats per 3 rectes: la que està completa perquè es troba entre cantonades i dues rectes paral·leles incompletes, que apareixen perquè l'escanejat ha començat amb el capçal enfocat a una paret. En el moment en que es llegeixen els punts d'un pla, se n'està creant un altre de complementari i, per tant, aquestes rectes incompletes tenen la seva representació acabada en aquest pla complementari.

El primer que provarem per tractar aquestes rectes, doncs, és quedar-nos com ja hem dit amb la més llarga i obviar la curta. Quan es tracti el pla complementari, els papers s'invertiran i la recta que ara hem descartat és la que es tractarà. Per fer aquesta separació ens fixarem en les distàncies entre punts: si entre dos punts hi ha una distància molt gran, voldrà dir que hem canviat de recta, i es poden comptar els punts a banda i banda per descartar la recta de la que hi ha menys mostres.

Ara que ja disposem de dos conjunts de punts que sabem que representen rectes perpendiculars, podem generar rectes a partir d'ells. Això es podria fer amb el mètode de la transformada de Hough explicat a l'estudi previ, però abans intentarem explorar altres mètodes menys costosos, el primer dels quals serà la regressió lineal.

La regressió lineal és un mètode utilitzat per trobar la línia que més s'ajusta a un conjunt de punts determinat (anomenada també *best fitting line*). El criteri per determinar quina és aquesta línia és el que es coneix com minimitzar la suma de quadrats. que surt d'elevat al quadrat la distància vertical entre un punt (x,y) i la recta, i sumar aquests quadrats de tots els punts. En una recta perfecta, tots els punts estarien sobre la recta i per tant la suma de quadrats seria 0. A partir d'aquí, qualsevol desviament de la recta és erroni, i es busca la recta que minimitzi aquests desviaments.

Per trobar aquesta recta, aplicarem les fórmules següents:

$$(1) S_x = \sum_{i=0}^{N-1} (x_i - \bar{x})^2$$

$$(2) S_y = \sum_{i=0}^{N-1} (y_i - \bar{y})^2$$

$$(3) b = \frac{S_x * S_y}{S_x^2}$$



La fórmula (3) ens retorna el pendent b de la recta que millor s'ajusta al nostre conjunt de punts, i per calcular-la només necessitem saber la x i y mitjanes del conjunt, que ens permetran fer els dos sumatoris (1) i (2) per la desviació mitjana. Per trobar el punt de tall de l'equació de la recta $y = bx + a$, podem aïllar agafant les mitjanes que hem calculat com a punt x, y de la recta. Aplicant aquesta regressió lineal als dos conjunts de punts de què disposem, tindrem dues equacions de rectes diferents.

Finalment, podem fer servir aquestes equacions per trobar cantonades. Sabem que les dues rectes són perpendiculars, i si busquem la intersecció entre les dues trobarem una cantonada que s'ajusti al conjunt. Aplicant aquest procés per dos plans que siguin complementaris, sortiran dues cantonades oposades entre elles. Per trobar les dues que falten, podem interseccar les rectes no paral·leles dels plans entre elles.

Resultat

Aquest mètode dona bons resultats i presenta molts avantatges sobre els altres que s'han anat provant. En primer lloc, és robust als errors puntuals, perquè un pendent fora del normal s'ignora i no es té en compte en els càlculs posteriors. L'altre avantatge important és que aquest mètode simplifica molt el càlcul de les cantonades, ja que es redueix a calcular les interseccions entre rectes i obtenim exactament 4 punts, sense necessitat de processat addicional.

Fixant-nos ara en els resultats, apreciem que són suficientment bons. És evident que, com que estem fent servir un mètode d'aproximació, els punts obtinguts no coincidiran exactament amb punts del conjunt, però són molt propers en totes les ocasions al punt que per observació visual podríem determinar com a cantonada.

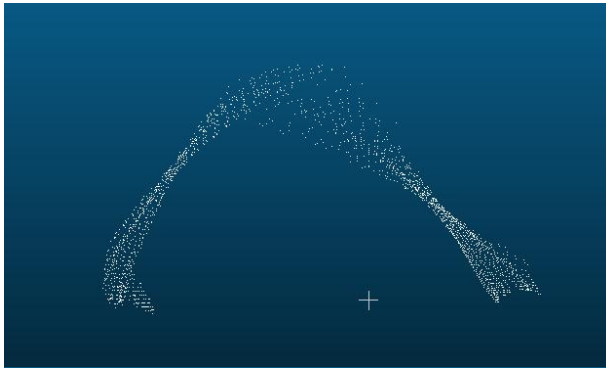


Figura 13. Per comprovar l'algorisme, el provarem sobre el conjunt de la prova de la figura, al que ja hem deixat només amb les lectures d'una càmera per claredat.

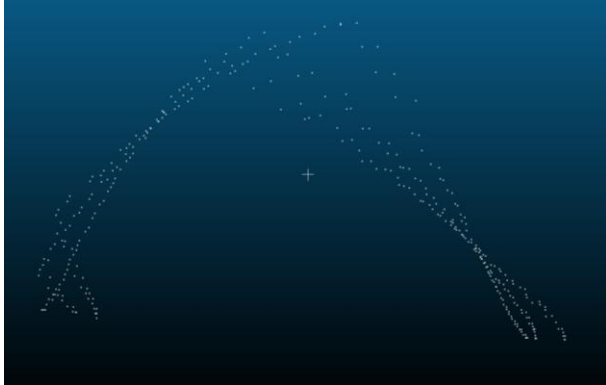


Figura 14. D'aquest conjunt se n'han separat els diversos plans que el conformen en entitats separades. En la figura es poden observar 3 d'aquests plans.

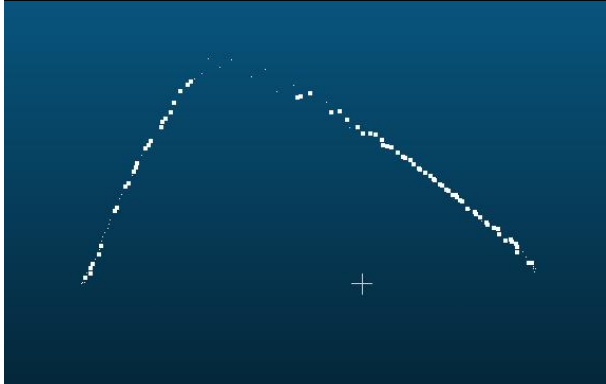


Figura 15. Centrant-nos en un sol pla, sobreposem els conjunts de punts detectats com a rectes i els ressaltem. Es pot apreciar que els conjunts seleccionats realment formen una recta bastant ben definida, especialment en el cas de la més llarga.

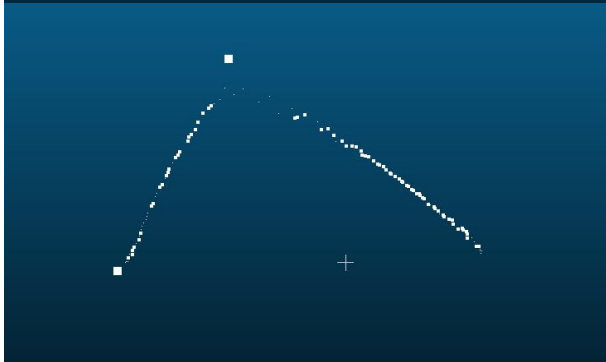


Figura 16. Finalment, amb els punts classificats en rectes es poden obtenir les equacions de les rectes, interseccar-les i trobar les cantonades. Es pot veure que la intersecció entre les dues rectes no és cap punt del pla, però es pot considerar que està rectificat la corbatura que es veu en les dades de la cantonada.

En vista dels resultats, es dona com a vàlid aquest mètode per aquesta primera etapa de detectar rectes i cantonades, i passarem a la següent, on les farem servir per determinar els plans que formen l'embornal.

9.2 Obturació de la canonada

Un cop determinades les rectes en les que està dividit l'embornal i les seves cantonades, podem fer servir aquesta informació per calcular el nivell d'obturació de la canonada, que era el nostre objectiu. Per fer-ho, primer cal detectar el perímetre del forat en qüestió

9.2.1 Detecció de la canonada

El que cal fer ara doncs és determinar quins punts formen part del perímetre del forat. Per fer-ho, farem ús de les cantonades que s'han detectat en l'etapa anterior. En primer lloc, necessitem afegir un càlcul addicional a la detecció de rectes. Aprofitant que fem un recorregut de tots els punts, per cadascun d'ells afegirem un càlcul de discontinuïtat, que no és altra cosa que mirar si un punt es troba a més d'una certa distància del següent. En cas afirmatiu, considerem que hi ha una discontinuïtat en ambdós punts. Un cop aïllades totes les discontinuïtats, ens fixem en quina paret es troben agafant les cantonades que hem trobat com a referència. Un cop classificades, és d'esperar que la paret on hi hagi el nombre més elevat de discontinuïtats sigui la paret on hi ha el forat, ja que cada recta que estigui en una alçada on hi ha forat té una alta probabilitat de tenir dues discontinuïtats (a no ser que estigui en una alçada on el forat està obturat). Per tant, la paret amb més discontinuïtats és la que s'analitzarà primer, i si no s'hi detecta forat, es provaran les parets restants per ordre de més a menys discontinuïtats.

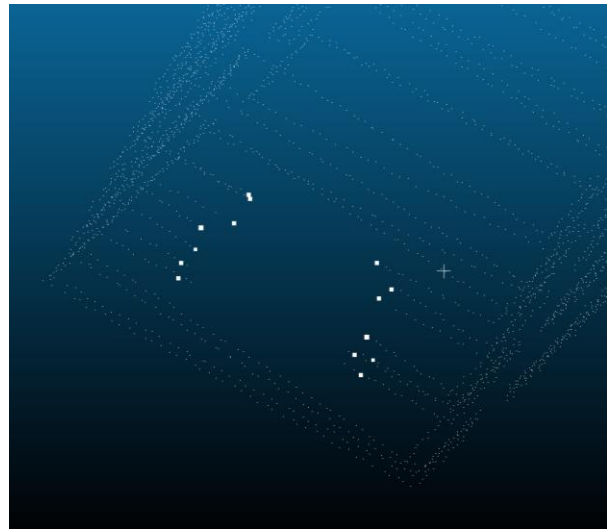


Figura 17. Detall de la paret on hi ha la connexió, es poden veure ressaltats els punts on hi ha discontinuïtats, que coincideixen amb la vora del forat.

9.2.2 Càlcul del nivell d'obturació

Un cop tenim el perímetre de la canonada trobat, el següent pas és determinar la seva mida i quin percentatge d'aquest està obturat. En cas d'haver-hi brutícia tapant-ne l'entrada, aquesta començarà a acumular-se a la base de la boca i anirà apilant-se. Per tant, podem començar suposant que el forat de la canonada es pot reduir a un cercle que pot estar parcialment tapat començant per baix. Sabent això, podem dividir aquesta part en dues etapes: en primer lloc, determinar la mida del forat, i tot seguit trobar fins a quina alçada arriba la brutícia.



Per trobar les dimensions del forat, ens podem ajudar d'una de les eines que ens proporciona la *PointCloudLibrary*, concretament la *SACsegmentation*, basada en un *RANSAC* que es fa servir per obtenir un model concret.

RANSAC (Random Sample Consensus) és un mètode que es fa servir per calcular els paràmetres d'un model matemàtic a base d'iteracions successives de prova i error. No és un algorisme determinista, sinó que fa aproximacions successives, i per tant se li ha de proporcionar una cota a partir de la qual considerarà el resultat suficientment bo com per aturar-se. Donat un conjunt d'entrada, el que fa és buscar un model concret a base de seleccionar uns quants punts aleatoris, construir el model seleccionat en base a aquests punts i provar la resta de punts contra aquest model, considerant com a bo un model on hi hagi gran part dels punts que s'han qualificat com a bons seguint aquell model.

Sabent que es disposa d'aquesta eina, l'aproximació escollida és triar els punts del núvol que formin part del perímetre del forat. A partir d'aquest subconjunt de punts, que sabem que formen un cercle, buscarem una aproximació a un cercle, que sabem que és la forma que hauria de tenir el conjunt. D'aquesta forma, el que farem serà calcular les propietats de l'objecte, i es podrà extreure d'elles el radi del forat.

Un cop trobat el cercle, la dimensió del forat és coneguda i podem calcular fins on arriba la brutícia. Si revisem les dades originals (sense les parts que no pertanyen a la recta eliminades), fixant-nos concretament en els valors dels punts que sabem que pertanyen a l'interior del forat, podem analitzar-los de baix cap amunt i veure fins a quina alçada veiem que els punts es distribueixen sobre la mateixa zona. És a dir, volem saber quina és l'alçada a la que encara s'estan veient punts en l'espai del forat que teòricament hauria d'estar desocupat. Un cop trobada la recta que es troba més amunt de les compreses dins el forat, podem extreure l'alçada mitjana dels seus punts compresos dins el forat i fer servir aquesta alçada com a alçada fins on arriba la brutícia (sota la premissa ja comentada que aquesta s'acumula uniformement). Aquesta alçada marca una divisió en el forat, separant-ne un arc superior d'espai lliure en cas que la brutícia ocupi més de mig forat, o un arc inferior de brutícia en cas que ocupi menys de mig forat. Obtenint l'àrea d'aquest arc, podem saber quina és l'àrea que queda lliure de forat (directament l'àrea de l'arc en el primer cas, o la diferència entre l'àrea del forat complet i la de l'arc en el segon).

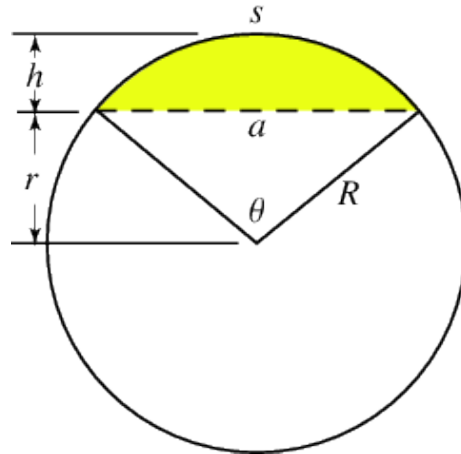


Figura 18. Esquema del càlcul de l'àrea de l'arc.

Veient l'esquema, es fa evident que l'àrea de l'arc (en groc) es pot obtenir a partir de l'àrea del sector circular determinat per θ menys l'àrea del triangle.

L'àrea del sector és:

$$(1) \quad A_s = \frac{1}{2} R^2 \theta$$

I la del triangle:

$$(2) \quad A_T = \frac{1}{2} r a$$

De l'esquema en podem extreure que:

$$(3) \quad R = r + h$$

Per trigonometria:

$$(4) \quad \theta = \frac{s}{R} = 2 \cos^{-1} \left(\frac{r}{R} \right)$$

$$(5) \quad r = R \cos \left(\frac{1}{2} \theta \right)$$

I finalment:

$$(6) \quad a = 2R \sin \left(\frac{1}{2} \theta \right) = 2r \tan \left(\frac{1}{2} \theta \right) = 2\sqrt{R^2 - r^2} = 2\sqrt{2Rh - h^2}$$

Finalment, substituint a (1) i (2) i fent la resta tenim:

$$(7) \quad A = R^2 \cos^{-1} \left(\frac{R-h}{R} \right) - (R-h) \sqrt{2Rh - h^2}$$

Que ens retorna l'àrea del segment, a partir de la qual obtenim l'àrea disponible en el forat com ja s'ha explicat.

Per més detalls, es pot consultar el lloc web^[9] adjunt a la bibliografia que explica aquest càlcul en detall.

9.3 Resultat final

Un cop calculada l'àrea lliure del forat de connexió, l'únic que resta és comprovar que el resultat que s'està obtenint és el que es desitjava. El procediment per fer això ha estat generar conjunts de dades amb el generador de dades (si el sistema no hagués estat incapacitat aquest pas s'hauria substituït per recollir dades reals) i provar l'algorisme sobre aquestes dades veient quin valor es retorna. L'avantatge de provar-ho sobre dades simulades en front de dades reals és que amb les dades reals hauríem de calcular de forma aproximada quin és el percentatge del forat que està obturat, i això ens podria dur a un cert error. En canvi, com que el generador de dades genera reproduccions amb uns paràmetres exactes, podem determinar exactament quin és el resultat que hauria de sortir i comparar-ne les diferències.

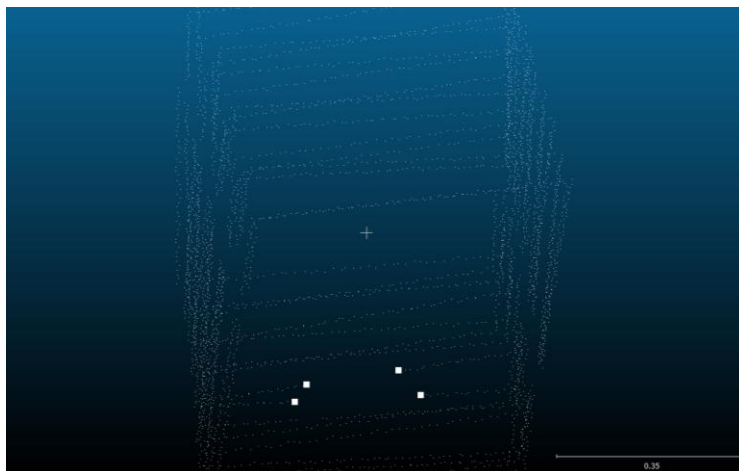


Figura 19. Embornal creat amb un 50% d'obturació.

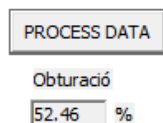


Figura 20. Resultat obtingut.

Resultat 1

L'embornal s'ha creat conservant tots els punts per sota del centre del forat, és a dir, amb una obturació del 50%.

Tenint en compte que no es pot generar amb exactitud aquest valor, ja que les línies tenen una lleugera inclinació, un error de 2,5% és més que acceptable.

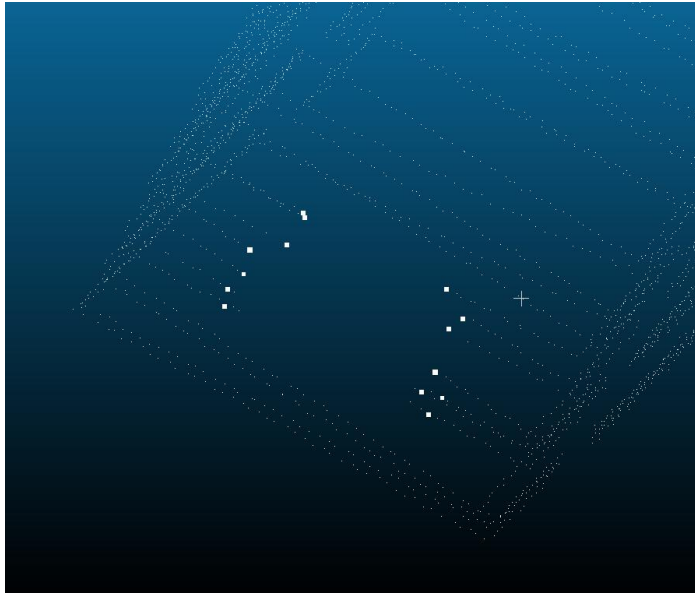


Figura 21. Embornal generat amb un 12,1% d'obturació.

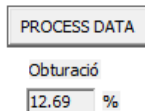


Figura 22. Resultat obtingut.

Alguns altres resultats sense imatges:

Resultat 3

Paràmetres: radi = 10 cm, h. centre = 15 cm, h. obturació = 10 cm, h = 5, r = 5

Àrea arc = 61.42 cm², Àrea cercle = 314.16 cm², Obturació = 61.42/314.16 = 0.1955

Obturació real: 19.55%

Obturació detectada: 18.37%

Els dos resultats següents són del tipus en què la obturació està per sobre del 50%, i el càlcul es fa lleugerament diferent (l'àrea ocupada és l'àrea total menys la de l'arc).

Resultat 4

Paràmetres: radi = 20 cm, h. centre = 15 cm, h. obturació = 25 cm, h = 10, r = 10

Àrea arc = 245.67 cm², Àrea cercle = 1256.64 cm², Obturació = (1256.64 - 245.67)/1256.64 = 0.8045

Obturació real: 80.45%

Obturació detectada: 81.09%

Resultat 2

Aquest segon embornal s'ha generat amb el centre del forat a una alçada de 24 cm i un radi de 14 cm, conservant tots els punts per sota de 15 cm d'alçada. Aplicant la fórmula abans vista per l'àrea d'aquest segment amb R = 14, r = 9 i h = 5, obtenim una àrea de 74.51 cm², mentre que la del cercle s'obté a partir de πR^2 i és 615.75 cm². Fent la relació entre els dos valors, vol dir que la obturació és de 74.51/615.75 = 0.121, un 12.1%, que resulta molt proper al 12.69% obtingut.



El següent resultat s'ha generat amb les mateixes dades pel cercle però duplicant l'error amb el qual es generen les dades.

Resultat 5

Obturació real: 80.45%

Obturació detectada: 82.21%

La precisió del resultat disminueix, però no tant com es podria creure, principalment perquè el mètode RANSAC que detecta la circumferència és molt robust, i al fer-se una mitjana de l'alçada dels punts del pla superior, es suavitzen bastant els errors puntuals.

Resultat 5

Paràmetres: **radi** = 20 cm, **h. centre** = 15 cm, **h. obturació** = 20 cm, **h** = 15, **r** = 5

Àrea arc = 430.42 cm², **Àrea cercle** = 1256.64 cm², **Obturació** = $(1256.64 - 430.42)/1256.64 = 0.6574$

Obturació real: 65.74%

Obturació detectada: 63.24%

Mantenint els paràmetres de l'anterior prova però augmentant el nombre de plans amb els que es generen les dades, el comportament és l'esperat i s'augmenta la precisió.

Resultat 6

Obturació real: 65.74%

Obturació detectada: 65.27%

Aquest comportament era el que s'esperava per un simple tema de nombre de mostres. Com que les dades estan discretitzades, com més plans hi hagi menys distància hi haurà entre ells, i per tant és molt més probable que el pla més proper al límit real hi sigui més a prop.

Podem concloure doncs que els resultats obtinguts són bastant satisfactoris. Cal tenir en compte que l'error màxim possible depèn de la distància entre plans: com més propers, més probable és que el pla superior que creua el forat coincideixi amb el marge superior de la obturació, i menor l'error que es cometrà. L'error màxim també és més gran com més propera sigui la obturació al 50%, ja que al centre del forat és quan el mateix en error en alçada comporta l'error més gran en àrea per tractar-se de la part més ampla.



10. Desviacions

Com ja s'ha mencionat en apartats previs del document, el projecte ha patit severes desviacions a causa de fallades del *hardware*. Aquests errors han fet, entre altres coses, que fos impossible fer les proves del hardware amb el software per comprovar que tot funcionés correctament. Les proves de laboratori no s'han pogut realitzar per problemes d'*encoder* cap al final del projecte, i les proves de camp presenten el problema addicional que el hardware encara no està preparat per suportar les condicions de la intempèrie (l'electrònica no està tancada hermèticament, les càmeres no tenen cap mena de protecció a l'objectiu, etc.).

No obstant, s'han dut a terme altres tasques per compensar aquestes desviacions i poder acabar el projecte, concretament la cal·libració del sistema i la creació d'un generador de dades, que s'expliquen a continuació.

10.1 Cal·libració del sistema

Durant el desenvolupament del codi del projecte, la part *hardware* del sistema ha patit un seguit de modificacions que han canviat les seves propietats físiques. Per tant, abans de fer la primera prova real ja hi ha un error conegut que podem arreglar, que és la forma en la que es tracten les dades llegides. Aquesta tasca es va dur a terme després de la primera fallada del *hardware*, sense possibilitat de saber que aquest tornaria a fallar més endavant. Per tant, no ha tingut impacte en el resultat final del projecte, ja que les últimes proves s'han fet amb dades generades, però el fet de tenir el procés de cal·libració fet pot servir en un futur si les propietats físiques del sistema no varien o, en el pitjor dels casos, tenir el procediment plantejat i documentat pot ajudar a fer-lo més ràpidament quan sigui necessari repetir aquest pas.

Quan es vol extreure un punt d'una lectura proporcionada per una càmera, en la que es veurà l'interior de l'embornal i la llum projectada pel *LED* vermell, el primer pas que es fa és generar una nova imatge a partir de la component vermella de la imatge original. En aquesta nova imatge, el punt on impactava la llum vermella serà molt lluminós, mentre que la resta de la imatge serà fosca. Aquesta imatge és molt simple de binaritzar, i permet saber amb exactitud la posició del punt, del qual se n'extreu el seu centre de gravetat. En funció d'aquest centre de gravetat, podrem saber les coordenades exactes del punt vermell i afegir-lo al conjunt de dades.

Per determinar a quina distància en horitzontal es troba un punt a partir del seu centre de gravetat, necessitem alguna equació que permeti arribar d'un valor a l'altre. Aquesta equació l'obtindrem de forma experimental. L'experiment en qüestió consistirà en el següent: amb el sistema fix, posarem un objecte a una distància coneguda i anotarem les coordenades del seu centre de gravetat.

El procediment en si no és complex, però cal garantir que les condicions en les que es realitza l'experiment són les apropiades i es documenten correctament els resultats. Per fer l'experiment, cal que el sistema es trobi en una posició estable i coneguda, perquè necessitem que la càmera es trobi a una alçada fixa que es farà servir de referència quan se n'extreguin conclusions. Aquesta posició fixa es fa més difícil de garantir conforme s'allarga l'experiment, per factors externs que no es poden impedir tals com simplement la gravetat que desplaça lentament el capçal.

En conseqüència, s'ha decidit fer unes quantes repeticions de cada experiment per treballar amb resultats promitjos, i fer cadascuna d'aquestes repeticions de principi a fi sense pausa (en sessions d'entre 2 i 3 hores).

Per cada experiment cal, en primer lloc, fixar el sistema com s'ha mencionat i anotar l'alçada del marge inferior de la càmera amb la qual s'estan prenent les lectures. Un cop determinada aquesta alçada, col·loquem un objecte pla, com ara una capsa, a una distància en horitzontal coneguda respecte la càmera. En el cas dels experiments fets en aquest projecte, aquesta distància inicial ha estat de 5 cm. El LED vermell projectarà un punt sobre aquesta superfície, del que anotarem a quina distància es troba de terra. Fetes aquestes mesures, correm una versió del programa modificada per fer una lectura estàtica (sense moviment del motor) i programada per guardar en un fitxer de dades no les coordenades del punt, sinó el seu centre de gravetat en la imatge captada per la càmera. Examinant aquest fitxer podem recuperar aquest centre de gravetat, i ajuntar-lo a les dades que ja teníem del punt.

Δz	Δy	cogx	cogy	Hcamera	Δy	Δz
39,2	5	878,38	256,46	45,5	5	6,3
38,7	7	777,47	251,85	45,5	7	6,8
37,8	10	633,33	245,35	45,5	10	7,7
36,4	15	485,69	238,78	45,5	15	9,1
34,9	20	395,1	233,84	45,5	20	10,6
33,5	25	328,49	230,55	45,5	25	12
31,8	30	279,73	228,72	45,5	30	13,7
30,5	35	248,66	226,28	45,5	35	15
28,9	40	219,18	225,78	45,5	40	16,6
27	45	196,49	223,92	45,5	45	18,5
25,7	50	178,52	223,22	45,5	50	19,8
24	55	163,66	222,3	45,5	55	21,5
22,3	60	150,08	221,5	45,5	60	23,2
20,6	65	139,78	221,63	45,5	65	24,9
19	70	129,91	220,51	45,5	70	26,5
17,5	75	121,39	220,5	45,5	75	28
15,8	80	114,39	220,52	45,5	80	29,7
14,3	85	107,36	220,44	45,5	85	31,2
12,8	90	101,3	219,24	45,5	90	32,7
10,9	95	95	219	45,5	95	34,6
9,1	100	90,29	218,79	45,5	100	36,4
7,7	105	88,03	219,19	45,5	105	37,8
6	110	82,15	218,48	45,5	110	39,5
4,3	115	78,23	218,18	45,5	115	41,2
2,8	120	75,45	218,88	45,5	120	42,7

Figura 23. En aquesta captura estreta del full de càlcul, es poden veure les dades obtingudes per la càmera superior. A l'esquerra, l'alçada a la qual estava el punter làser de terra, la distància fins la càmera i els valors x i y del centre de gravetat del punt. A la columna de la dreta, una conversió de l'alçada del punter làser a distància vertical fins la càmera. Es pot observar que el valor x del centre de gravetat pateix canvis molt més significatius a mesura que belluguem el punt que no pas la y, així que és el que farem servir per obtenir la fórmula.

Fet això, es pot desplaçar l'objecte una certa distància i repetir les lectures. En els experiments fets, la distància sempre s'ha augmentat a partir de 10cm en intervals de 5cm per mantenir la durada de l'experiment dins d'uns límits de temps raonables. Amb una inversió major de temps, es podria reduir aquest interval per tenir més lectures i més precisió. Les lectures més properes de 10cm s'han fet amb intervals més curts, perquè presenten molta més variabilitat.

Un cop preses totes les mesures que es vulguin, disposem de la següent informació de cada punt: l'alçada a la que es trobava de terra, la seva distància a la càmera (establerta manualment) i les coordenades x i y del seu centre de gravetat en la imatge de la càmera. A partir de la distància entre el punt i el terra i l'alçada de referència de la càmera que hem anotat prèviament, podem establir la distància vertical entre el punt i la càmera. A partir d'aquestes dades, ens ajudarem d'alguna eina de càlcul (com per exemple *Matlab*) per establir una relació entre el centre de gravetat del punt i la seva correspondència en coordenades relatives des de la posició de la càmera. Aquesta relació prendrà la forma d'un polinomi de grau n, en funció del que es vegi que dona una millor aproximació a la realitat. Gràcies a aquests polinomis, quan durant l'execució del programa es capti una imatge, es podrà fer servir el centre de gravetat del punt vermell per aproximar la seva distància respecte la posició de la càmera, que és coneguda en tot moment.

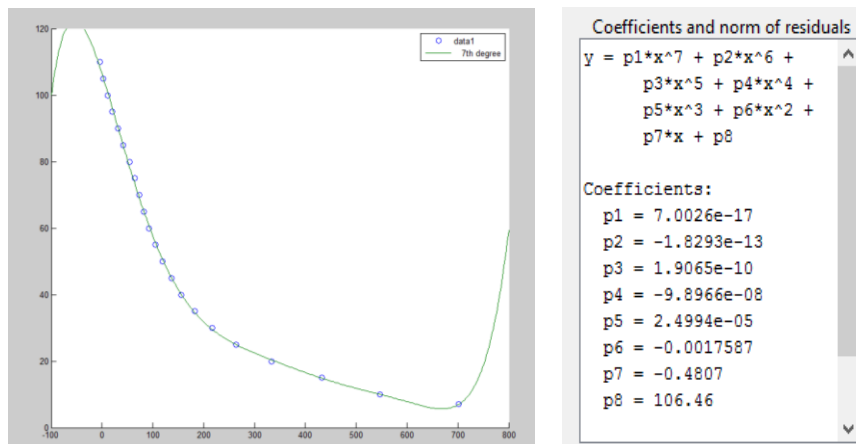


Figura 24, Figura 25. Polinomi que aproxima la distància de la càmera inferior fins el punt a partir de la x del centre de gravetat. Un polinomi de grau 7 era el que millors resultats donava. A la dreta, els coeficients del polinomi.

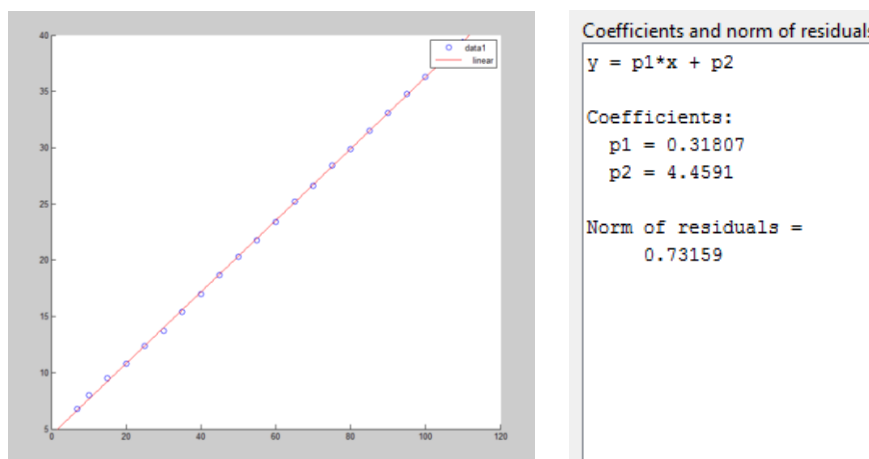


Figura 26, Figura 27. Recta que aproxima la distància vertical en funció de la distància horitzontal calculada amb el polinomi anterior. Tot i que es podria pensar que hi hauria menys error si es calculés directament des de la x del centre de gravetat, experimentalment els dos mètodes tenen un resultat comparable.

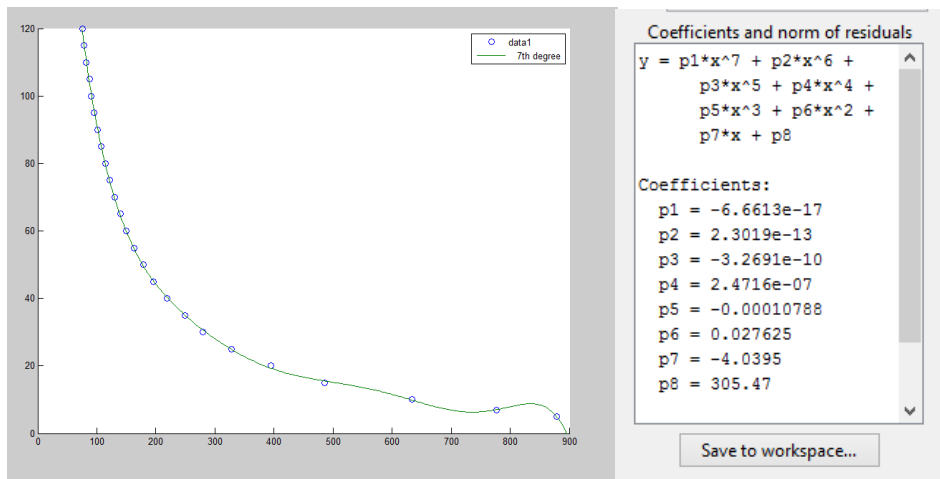


Figura 28, Figura 29. A diferència de la càmera inferior, per la superior el polinomi que ha donat una millor aproximació és de grau 6.

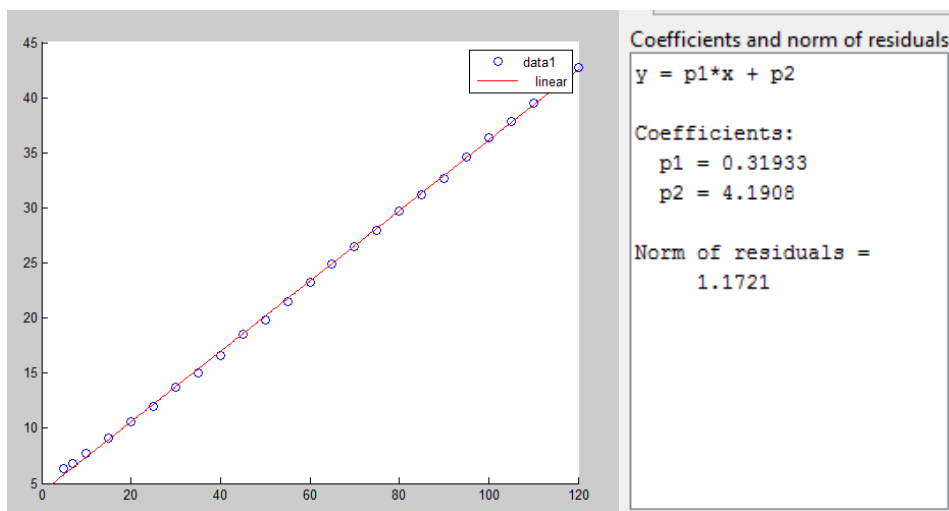


Figura 30, Figura 31. Recta per aproximar la distància vertical en funció de la x del centre de gravetat per la càmera superior i els seus coeficients.



10.2 Generador de dades

Per poder progressar en el desenvolupament del projecte, ha estat necessari fer un sistema de generació artificial de dades per suplir la falta de dades reals que no podia proporcionar el sistema *hardware*. La idea del generador és poder simular dades tan realistes com sigui possible, així que es prenen com a referència les lectures originals que hi havia prèvies a qualsevol modificació. El generador no té una interfície gràfica pròpia perquè no s'ha considerat necessària, però el codi està totalment parametritzat perquè els valors a partir dels quals es generen les dades siguin fàcilment modificables.

El primer pas ha estat fer un generador que sigui capaç de crear les parets d'un prisma simulant l'efecte d'anada i tornada que hi ha en les dades reals, i també el fet que la lectura no és una línia uniforme sinó un conjunt de mostres puntuals. Per crear-lo, es tria el nombre de plans que tindrà cadascuna de les meitats del prisma, l'alçada de cada paret, el nombre de punts per pla i les dimensions de cada paret. En base a aquestes dades, es determina l'alçada dels plans a partir del seu nombre i l'alçada de cada paret, i la distància entre punts a partir de la longitud sumada de les parets del pla i el seu nombre de punts.

Ara que ja s'han generat les parets d'un prisma perfecte, el següent a fer és simular un error humà que ha estat present en totes les lectures des del primer dia: el gir del capçal. Quan un escaneja, intenta baixar el capçal lentament seguint una trajectòria recta, però el que s'ha observat en repetides ocasions és que hom té tendència a fer més força amb una mà que amb l'altra i es gira progressivament el capçal de forma inconscient. A la llarga, la diferència d'angle entre el primer pla i l'últim ja és notable i s'ha de tenir en compte. En aquesta modificació del generador, s'estableix una diferència d'angle entre ambdós plans, i per cada pla intermig es fa un gir d'una fracció proporcional d'aquest angle, amb la qual cosa aquest angle passa a formar part de tots els càlculs de distàncies del generador.

Finalment, i per afegir sensació de realisme, es poden establir marges d'error tant per alguns dels paràmetres que fa servir el generador com de valors calculats en temps d'execució: una desviació màxima en nombre de punts per pla, en l'alçada i l'angle de gir que corresponen a cada pla (perquè no segueixin tots un interval uniforme), en la distància entre els diversos punts d'un mateix pla i en les pròpies coordenades x , y i z de cadascun dels punts, de manera que s'acaben generant unes dades que realment poden semblar captades manualment. Si bé en alguns aspectes es pot seguir notant que són artificials, s'han considerat suficients per provar les darreres fases del projecte.

A continuació es mostren alguns exemples de dades generades amb diversos paràmetres.

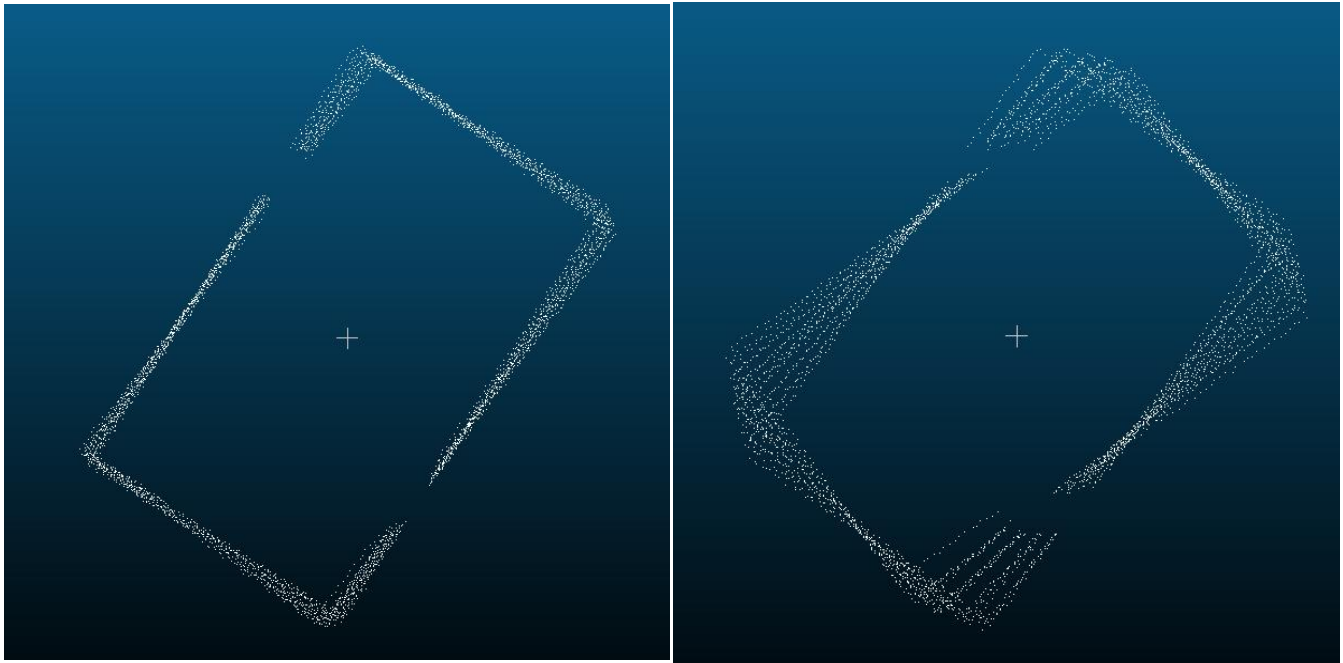


Figura 32, Figura 33. A l'esquerra, un conjunt de dades generat amb una diferència d'angle petita, de 10° entre el primer pla i l'últim. A la dreta, un conjunt generat amb els mateixos paràmetres exceptuant la diferència d'angle, que és de 40° .

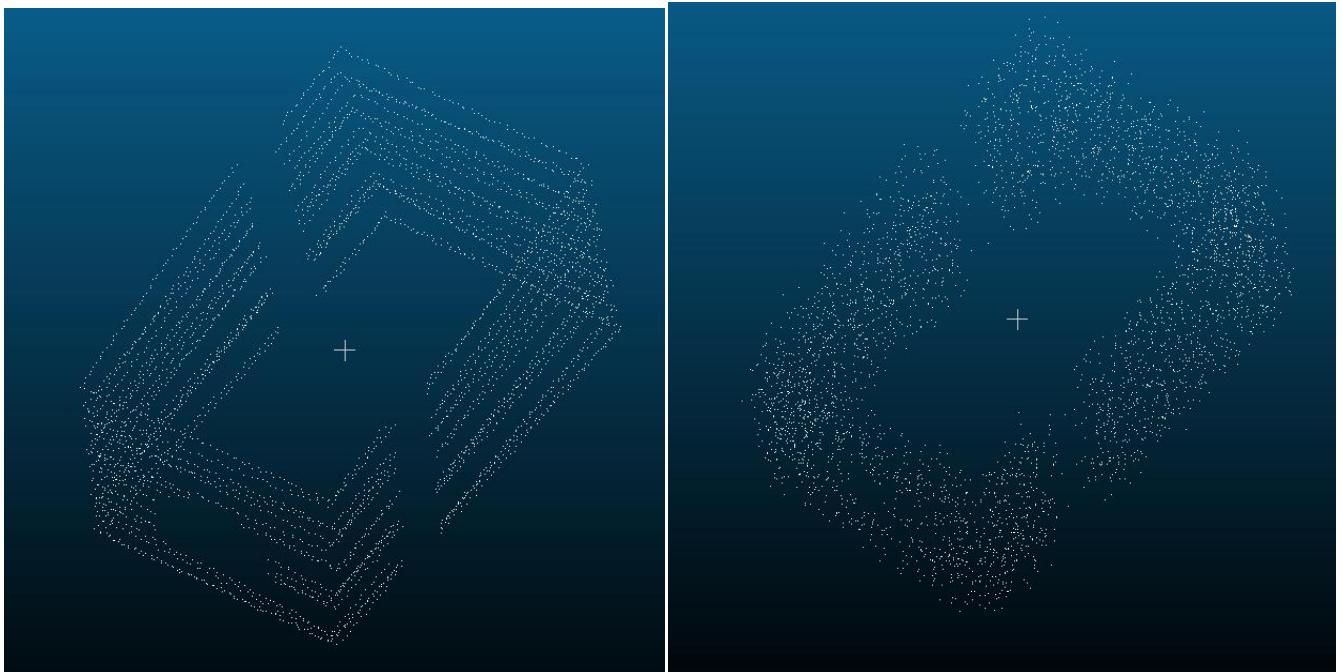


Figura 34, Figura 35. A l'esquerra, el mateix conjunt de la figura 32, però des d'una altra perspectiva. Aquest pla ha estat generat amb un error de precisió del 2%, és a dir, que cada cop que es crea un punt que ha d'estar a una distància x , y i z d'un altre, aquesta distància té un error del 2%. A la dreta, el conjunt generat amb 10 vegades més error, un 20%.



11. Conclusions i futura feina

Podem concloure que l'objectiu proposat ha estat complert, tot i els entrebancs que han sorgit pel camí, de forma satisfactòria. Partint d'un conjunt inicial de dades amb les característiques proposades, s'arriba a obtenir un resultat que quadra amb el que s'estava buscant.

Per arribar a aquests resultats, sempre s'ha buscat la solució més òptima que equilibri resultats i temps per obtenir-los, ja que recordem que l'algorisme desenvolupat aquí és només una part d'un projecte més gran que ha de córrer en ordinadors de poca potència i en un entorn on s'espera un resultat immediat, així que reduir els costos en temps ha estat necessari tant perquè l'algorisme en si trigui poc a executar-se com perquè no suposi una restricció per la resta del programa, en el sentit que aquesta resta hagi d'anar molt ràpid per compensar un anàlisi costós.

Dels canvis que ha calgut fer en la planificació inicial, el que més ha alterat el curs del projecte ha estat la impossibilitat de fer proves de camp deguda a tots els imprevistos que han impedit provar el sistema correctament. Tot i que s'han intentat emular el màxim nombre de defectes possibles quan es dissenyava el generador de dades, és evident que una lectura real sempre portarà una quantitat incomptable de petits errors inevitables. No obstant, tot i que la diferència entre les dades reals i les simulades que s'han vist al llarg del projecte és enorme, s'espera que amb els canvis proposats en quant a cal·libració i potser corregint alguns errors del servomotor, les dades que es puguin llegir siguin molt més semblants a les simulades.

A nivell de feina futura doncs, el més important seria recal·librar el sistema en quant es pugui considerar que el *hardware* és el definitiu i estigui comprovat per evitar errors. En aquesta recal·libració, s'hauria d'ajustar el sistema per aconseguir unes dades molt més uniformes, un intermig entre el que es tenia inicialment i les dades simulades. Hem pogut comprovar en les primeres iteracions del projecte que el sistema de detecció de rectes era molt robust, així que no cal que la millora en les lectures sigui inmensa per tenir els resultats desitjats.



12. Bibliografia

[1] <http://www.cplusplus.com/reference/>

[2] <http://pointclouds.org/documentation/>

[3] <http://www.danielgm.net/cc/documentation.html>

[4] https://www.cs.sfu.ca/~hamarneh/ecopy/compvis1999_hough.pdf

HOUGH, P. V. C. Method and means for recognizing complex patterns. U. S. Patent 3,069,654, December 18, 1962

[5] <http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm>

Yale University, material del curs Statistics 101-103. *Multiple Linear Regression*.

[6] <http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm>

Yale University, material del curs Statistics 101. *Linear Regression*.

[7] Muralikrishnan, B. i Raja, J. [2009]. *Computational Surface and Roundness Methodology*.

[8] www.gencat.cat/diari/3894/03127147.htm

Decret 130/2003, 13 de maig de 2003, Departament de Medi Ambient.

[9] <http://mathworld.wolfram.com/CircularSegment.html>



Índex de figures

<i>Figura 1. Exemple d'un embornal.....</i>	<i>7</i>
<i>Figura 2. Detall d'un costat del capçal</i>	<i>8</i>
<i>Figura 3. Exemple d'ús del sistema i detall de la base.</i>	<i>8</i>
<i>Figura 4. Sistema complet.....</i>	<i>8</i>
<i>Figura 5. Capçal Danyat</i>	<i>30</i>
<i>Figura 6. Vistes principals de Microsoft Visual Studio.</i>	<i>35</i>
<i>Figura 7. Entorn de CloudCompare.</i>	<i>36</i>
<i>Figura 8. Sservomotor i capçal del sistema.</i>	<i>37</i>
<i>Figura 9. Lectura d'exemple.</i>	<i>39</i>
<i>Figura 10. Esquema del mètode 1.....</i>	<i>41</i>
<i>Figura 11. Esquema del mètode 2.....</i>	<i>42</i>
<i>Figura 12. Gràfic resultant de classificar els pendents d'un pla.</i>	<i>44</i>
<i>Figura 13. Comprovació de l'algorisme 1</i>	<i>47</i>
<i>Figura 14. Comprovació de l'algorisme 2.</i>	<i>47</i>
<i>Figura 15. Comprovació de l'algorisme 3.</i>	<i>47</i>
<i>Figura 16. Comprovació de l'algorisme 4.</i>	<i>47</i>
<i>Figura 17. Detall de la paret on hi ha la connexió.....</i>	<i>48</i>
<i>Figura 18. Esquema del càlcul de l'àrea de l'arc.</i>	<i>50</i>
<i>Figura 19. Embornal creat amb un 50% d'obturació.</i>	<i>51</i>
<i>Figura 20. Resultat obtingut.</i>	<i>51</i>
<i>Figura 21. Embornal generat amb un 12,1% d'obturació.</i>	<i>52</i>
<i>Figura 22. Resultat obtingut.</i>	<i>52</i>
<i>Figura 23. Dades de cal·libració de la càmera superior.</i>	<i>55</i>
<i>Figura 24, Figura 25. Polinomi que aproxima la distància de la càmera inferior</i>	<i>56</i>
<i>Figura 26, Figura 27. Aproximació de la distància vertical de la càmera inferior</i>	<i>56</i>
<i>Figura 28, Figura 29. Polinomi que aproxima la distància de la càmera superior.</i>	<i>57</i>
<i>Figura 30, Figura 31. Aproximació de la distància vertical de la càmera superior.</i>	<i>57</i>
<i>Figura 32, Figura 33. Comparativa d'angles generats.</i>	<i>59</i>
<i>Figura 34, Figura 35. Comparació d'errors generats</i>	<i>59</i>

Índex de taules

<i>Taula 1. Càlcul d'hores per cada fase i en total.....</i>	<i>23</i>
<i>Taula 2: Dependències entre tasques.....</i>	<i>23</i>
<i>Taula 3: Càlcul de pressupost</i>	<i>28</i>
<i>Taula 4: Planificació temporal revisada.</i>	<i>31</i>
<i>Taula 5: Matriu de sostenibilitat</i>	<i>32</i>

Annex: Diagrames de Gantt

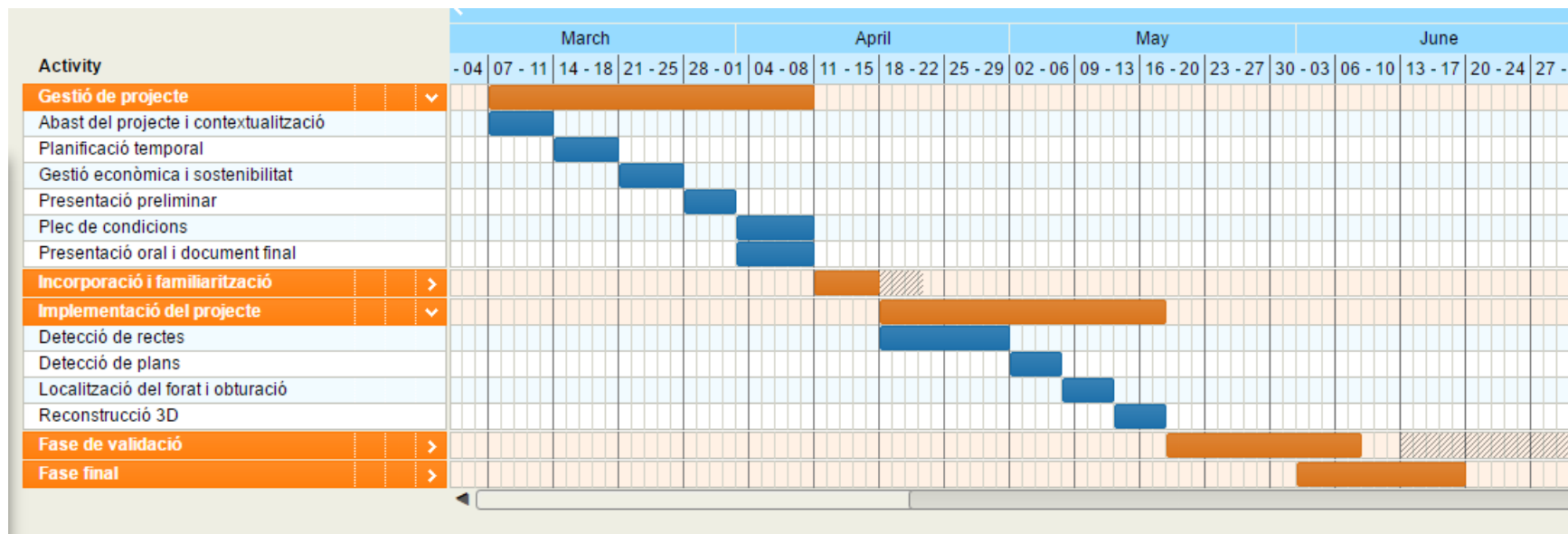


Diagrama 1. Planificació inicial

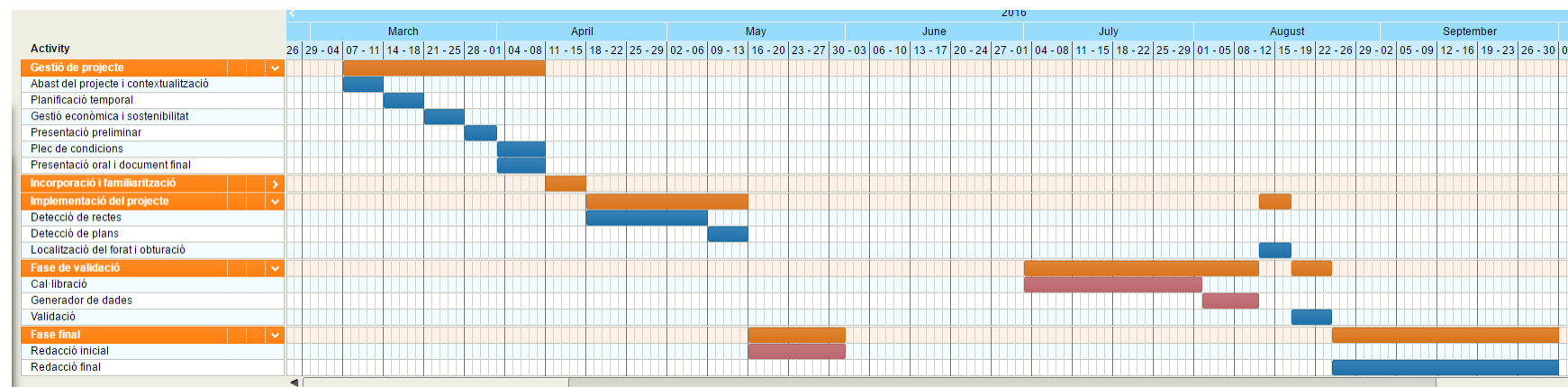


Diagrama 2. Planificació final.