

Integració d'un sistema d'autolocalització per a robots mòbils mitjançant visió per computador

Aniket Mavlankar

Juny 2018

*Treball Fi de Grau en Enginyeria Informàtica
Enginyeria de Computadors*



Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya
Director: Joan Climent Vilaró

Resum

Els sistemes de localització existents han donat lloc a molts altres sistemes més complexos, construïts al voltant de la seva tecnologia. Han obert un món de possibilitats, inexistents fins al moment. Han contribuït en gran mesura al procés de creixement tecnològic en el que vivim. Aquest projecte pretén explorar una alternativa, a les ja existents, que permeti cobrir les seves mancances i limitacions. El propòsit general del projecte és desenvolupar una tecnologia integrable en altres sistemes, que de forma similar a les tecnologies existents, aportin un valor afegit i permeti explorar noves funcionalitats.

La finalitat del treball serà la de desenvolupar i integrar un sistema d'autocalització basat en visió per computador per a robots mòbils. Aquest permetrà als autòmats tenir un major grau de percepció de l'entorn. Es tractarà d'un primer prototip que es posarà a prova en escenaris realistes on es pugui avaluar el seu rendiment i identificar els possibles aspectes a seguir desenvolupant.

Existing guidance systems have led to the creation of other complex systems, built using their technology. A whole new world of possibilities opened up thanks to their appearance. This project aims to explore an alternative to the existing ones, which overcomes their deficiencies and limitations. The purpose of the project is to develop a technology that can be integrated in other systems, adding value and offering new features to them.

The aim of this project is to develop and integrate a computer vision-based guidance system for mobile robots. This will give them a better perception of their surroundings. As a first prototype, the system will be tested in realistic scenarios to evaluate its performance and detect possible aspects to improve.

Dedicat a la meva germana, font d'inspiració i valors.

Índex

Índex de figures xi

Índex de taules xiii

1	Introducció	1
1.1	Formulació del problema	1
1.1.1	Objectius	1
1.2	Contextualització	3
1.2.1	Context	3
1.2.2	Ús de resultats previs	5
1.3	Abast i riscos	6
1.3.1	Obstacles i riscos del projecte	6
1.4	Metodologia	8
1.4.1	Mètode de desenvolupament	8
1.4.2	Eines de desenvolupament	9
1.4.3	Mètode de validació	9
2	Planificació temporal	11
2.1	Descripció de tasques	11
2.1.1	Estudi previ	11
2.1.2	Planificació del projecte	12
2.1.3	Configuració inicial de l'entorn de treball	12
2.1.4	Desenvolupament principal	12
2.1.5	Tasca final	13
2.2	Taula de temps	14
2.3	Diagrama de Gantt	14
2.4	Recursos utilitzats	15
2.4.1	Recursos hardware	15
2.4.2	Recursos software	15
2.5	Desviacions i pla d'actuació	16

3	Antecedents	17
3.1	Visió per computador	17
3.1.1	Algorismes de visió	17
3.1.2	Sistema de detecció	24
4	Detecció i selecció de punts d'interès	27
4.1	Selecció de l'algorisme	27
4.1.1	Selecció usant imatges estàtiques	28
4.2	Proves amb seqüències pregravades	32
4.2.1	Resultats en imatge i video	33
5	Disseny i arquitectura del sistema	39
5.1	Esquema general	39
5.2	Software robot	39
5.2.1	Robot Operating System	39
5.3	Aplicació Android	41
5.3.1	Requisits funcionals	42
5.3.2	Disseny d'interfície	45
5.4	Comunicació entre dispositius	47
6	Implementació	49
6.1	ROS	49
6.2	Android	52
6.3	Docker	54
7	Proves de camp	57
8	Lleis i regulacions	61
8.1	Drets d'imatge	61
8.2	Propietat intel·lectual	61
8.3	Llicències	62
9	Canvis de planificació	63
9.1	Desviacions en la planificació	63
9.2	Desviacions en els recursos utilitzats	65
10	Informe de sostenibilitat	67
10.1	Matriu de sostenibilitat	68
10.2	Dimensió econòmica	68
10.2.1	Identificació de costos	68
10.2.2	Estimació de costos	69
10.2.3	Control de gestió	73
10.2.4	Viabilitat econòmica	74

10.3 Dimensió ambiental	74
10.4 Dimensió social	75
11 Conclusions i treball futur	77
11.1 Conclusions	77
11.2 Treball futur	78
Glossari	81
Bibliografia	85

Índex de figures

1.1	Selecció d'una regió d'interès	2
1.2	Workflow sistema	3
1.3	Robots humanoides IRI	4
2.1	Gantt - Desenvolupament temporal de tasques	14
3.1	Inspecció de píxels veïns	18
3.2	Patró de cercles concèntrics	19
3.3	Regions de la imatge	20
3.4	Canvi d'escala en una regió	20
3.5	Diferència de gaussianes (DoG)	21
3.6	Descriptor de keypoint (SIFT)	22
3.7	Box filter	22
3.8	Patró de selecció FREAK	23
4.1	Imatges capturades amb el primer dispositiu	28
4.2	Imatges capturades amb el segon dispositiu	29
4.3	Fases del procés de detecció	29
4.4	Combinacions d'algorismes sense detecció	30
4.5	Combinacions d'algorismes amb falsos positius	30
4.6	Resultats de combinacions amb matchings	31
4.7	Comparació dels millors detectors	32
4.8	Comparació detectors Shi Tomasi i SIFT	32
4.9	Fotograma d'exemple les proves en vídeo	34
4.10	Comparació de matches de les proves en vídeo	35
4.11	Comparació d'algorismes en un frame de video	36
4.12	Frame/Matches	37
4.13	Frame/Time	37
4.14	Posició dels matches	38
5.1	ROS topics	40
5.2	Esquema ROS del robot	41
5.3	Diagrama de casos d'ús	42

5.4	Vista principal	45
5.5	Pantalla configuració	46
5.6	Selecció i enviament d'imatge	47
6.1	Imatge OpenCV a missatge ROS	50
6.2	Socket ROI Initializer	51
6.3	Esquema de l'aplicació FinderApp	53
6.4	Funcionament AsyncTask	54
7.1	Robot de proves Pioneer 3-AT	57
7.2	Escenari de l'experiment	58
7.3	Finestra de visualització de RViz	59
7.4	Segona prova de camp	60
7.5	Tercera prova de camp	60
9.1	Gantt origianl - Desenvolupament temporal de tasques	64
9.2	Gantt actualitzat - Desenvolupament temporal de tasques	65

Índex de taules

2.1	Tasques planificades	14
4.1	Resultats d'execucions de la figura 4.6	31
4.2	Estadístiques del l'execució de fotograma 4.9	35
4.3	Estadístiques promitjes de tots els frames de vídeo	36
5.1	Funcionalitat 1	43
5.2	Funcionalitat 2	43
5.3	Funcionalitat 3	44
5.4	Funcionalitat 4	44
5.5	Funcionalitat 5	45
10.1	Matriu de sostenibilitat	68
10.2	Recursos hardware	70
10.3	Recursos humans	70
10.4	Cost recursos humans	71
10.5	Costos indirectes	71
10.6	Incidències	72
10.7	Costos totals	73

1. Introducció

1.1 Formulació del problema

Dins dels sistemes de localització existents, el més conegut i estès és el Global Positioning System (GPS). El seu ús s'ha popularitzat fins al punt de ser imprescindible avui dia per a moltes persones i sistemes autònoms. Basat en tecnologies per satèl·lit, aquest permet estimar la ubicació amb coordenades geogràfiques mitjançant la triangulació de diverses senyals. Tot i ser prou precís, aquest sistema té certes limitacions, com poden ser espais tancats o entorns amb poca cobertura.

En situacions com aquesta, autòmats com els robots poden tenir problemes per ubicar-se. És per aquest motiu que en certs casos, pugui ser convenient disposar d'un sistema d'ubicació alternatiu al GPS, de forma que el complementi. Això permetrà tenir una referència de la posició tant en entorns tancats com en zones amb poca cobertura.

1.1.1 Objectius

L'objectiu principal del projecte és desenvolupar un sistema d'autolocalització per a robots mòbils mitjançant visió per computador. La idea és dissenyar una eina parcialment integrada en un dispositiu mòbil i en un robot, de manera que l'usuari pugui enviar ordres per mitjà d'una imatge i un punt d'interès al robot. Aquest processa les ordres rebudes i cerca el punt d'interès sol·licitat en el seu entorn.

Una ampliació de l'objectiu serà determinar la distància a la que es troba del destí fent ús de les dues càmeres del robot. D'aquesta manera es podran calcular les coordenades relatives que més endavant es convertiran en consignes de moviment.

Tal com s'ha citat anteriorment, el sistema constarà de dos dispositius (el dispositiu mòbil i el computador embarcat en el robot) amb arquitectures hardware

diferents i, addicionalment, un ordinador que servirà per fer les diverses proves.

El flux de treball del sistema comença en el primer dispositiu, el telèfon mòbil. Allà s'executarà una aplicació Android que permetrà comunicar-se amb el robot. Serà necessari un protocol de comunicació sense fils que servirà per establir la connexió inicial entre tots dos dispositius. El següent pas serà, a través de l'aplicació Android, seleccionar una imatge ja sigui capturada amb la càmera o des de la galeria. D'aquesta imatge l'usuari haurà de seleccionar un punt d'interès, que serà la regió a buscar i destí del robot.

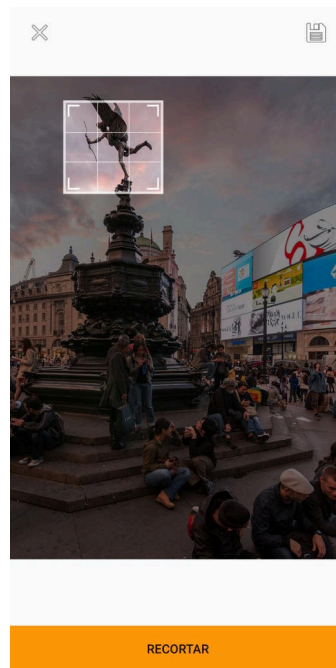


Figura 1.1: Selecció d'una regió d'interès

Tota aquesta informació s'enviarà al robot. Aquest calcularà els keypoints i descriptors de la regió enviada per l'usuari. A partir de la imatge en vídeo que captura a través de la seva càmera, el robot calcula en temps real els keypoints corresponents i les seves característiques, per buscar similituds amb els de la regió d'interès. L'algorisme del robot dona com a sortida, en cas positiu, les coordenades del centre de l'objecte trobat.

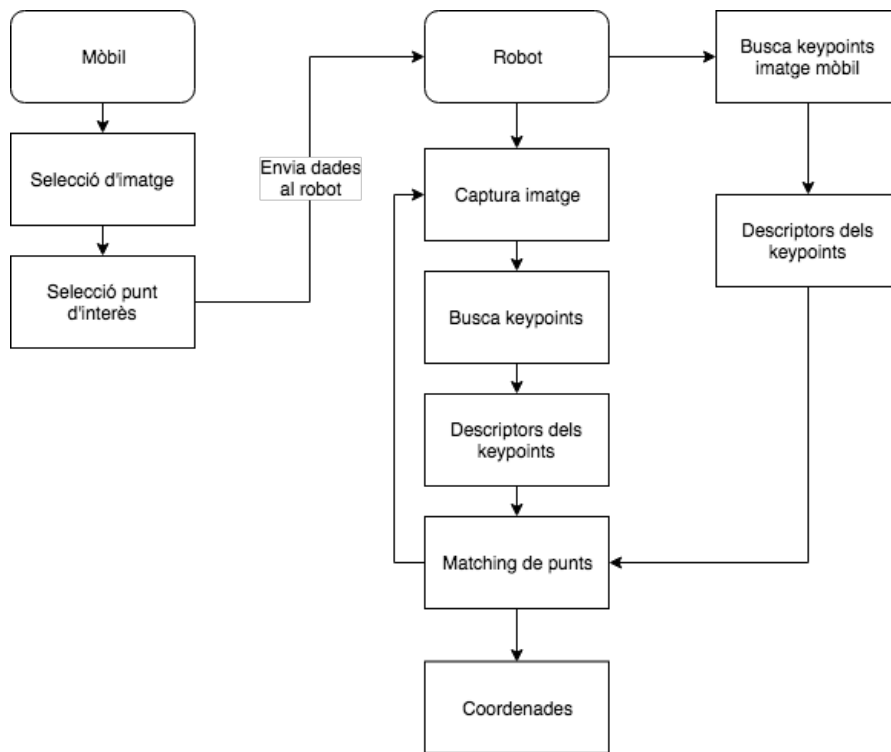


Figura 1.2: Workflow sistema

1.2 Contextualització

En aquesta secció es descriu breument quin és el context del projecte i la seva bibliografia. En la primera part es parlarà de quines àrees involucra i quins són els actors implicats. En la segona part es revisarà quin és l'estat de l'art d'aquestes àrees, i finalment, es discutirà si es farà ús d'alguna de les solucions presentades durant el projecte.

1.2.1 Context

L'impacte de la tecnologia en la societat els últims anys és innegable. Dispositius com els smartphones o tablets s'han popularitzat radicalment fins al punt de ser imprescindibles; han obert un món de possibilitats per resoldre tot tipus de problemes que puguem imaginar. Un altre exemple podria ser el del denominat Internet of Things (IoT), en el qual dispositius quotidians es troben connectats a la xarxa. També és molt present avui dia, per exemple, en televisors, rellotges o altaveus.

Ambdós exemples introduïts pertanyen a sectors tecnològics que han progressat molt i es troben en un avançat estat d'integració dins la societat. Però un dels sectors que encara no ho ha fet tant, i té gran potencial per fer-ho en un futur pròxim, és el de la robòtica. Si bé aquesta ha anat evolucionant a nivell d'investigació, i és molt utilitzada en processos industrials, no ha acabat de fer-se lloc dins la societat com ho han fet altres tecnologies, com ara els mòbils.

Aquest treball pretén donar un pas endavant en aquest sentit. Forma part d'un projecte dirigit a nivell nacional per l'Institut de Robòtica i Informàtica Industrial (IRI). Aquest té com a focus principal la integració de robots mòbils en entorns urbans, posant així a disposició dels ciutadans una eina amb un gran potencial i infinitat d'aplicacions.



Figura 1.3: Robots humanoides IRI

El que dotarà a aquests robots de percepció de l'entorn serà la visió per computador. El seu propòsit és fer ús dels algorismes i tècniques adequades a la situació, per tal de tenir la capacitat d'extreure informació a partir d'imatges obtingudes. D'aquesta manera es proporciona a màquines la capacitat de "veure" l'entorn, amb la intenció de replicar el comportament humà.

Aquest document pretén descriure detalladament el projecte, el qual es centrarà en el procés d'integració de tal sistema de visió. Consta de diversos apartats on s'hi detallen els objectius, el context, l'abast, la planificació i la gestió econòmica.

Àrees d'interès

- **Robòtica.** Existeixen múltiples definicions per definir la robòtica i no n'hi ha cap que sigui exacta i acceptada per tothom. Moltes coincideixen en dir que un robot és una màquina programable capaç de realitzar una sèrie de tasques interactuant amb l'entorn, ja sigui de forma autònoma o manual. En aquest cas concret, es tractarà d'un robot mòbil amb una morfologia similar a

la d'un vehicle. Disposa d'actuadors que el permeten desplaçar-se en l'espai i múltiples sensors per enregistrar diverses dades de l'entorn. D'aquests només es faran servir les càmeres per capturar imatges i els actuadors pel moviment.

- **Visió per computador.** Definim la visió per computador com la disciplina científica encarregada de crear mètodes per adquirir, processar, analitzar i entendre imatges del món real amb la finalitat de produir informació, tant numèrica com simbòlica, que pugui ser interpretable per una màquina [1].

Actors implicats

- **Project *developer*.** És l'única persona al càrrec del desenvolupament del projecte, en aquest cas es tractarà del propi autor. Conseqüentment, serà la persona responsable de realitzar la recerca, redactar tota la documentació i implementar tot el software que sigui necessari. Tot i treballar en coordinació amb el director, serà finalment l'encarregat de tota la planificació del projecte i el compliment dels deadlines.
- **Director de projecte.** La persona al càrrec d'aquest rol serà el Professor Juan Climent Vilaró, del departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAI). La seva funció serà de guia i suport al developer durant el transcurs del projecte en qualsevol obstacle que pugui sorgir.
- **Beneficiaris.** Qualsevol empresa o institució interessada en utilitzar el sistema i que puguin adaptar-lo a les seves necessitats, com pot ser el propi departament d'ESAI. També suposa una aportació a l'IRI, que investiga aquestes àrees amb l'objectiu d'arribar a l'integració de robots mòbils en entorns urbans.

1.2.2 Ús de resultats previs

Pel que fa a la part de visió per computador, es parteix d'un projecte final de grau previ, realitzat per l'alumne Joan Rodas Cusidó, en el que es fa un estudi dels diferents algorismes en l'estat de l'art de la visió per computador i es posa a prova la seva efectivitat. Com a punt de partida del nostre projecte, s'agafaran els resultats obtinguts i es farà una validació prèvia d'aquests per verificar empíricament la seva validesa i correctesa. Arribats a aquest punt, es procedirà a realitzar una integració d'aquests algorismes tant en el dispositiu mòbil com en el robot.

El robot per fer les proves serà proporcionat per l'IRI, de forma que no es podran realitzar canvis físics en la seva arquitectura. També és un dels components amb més pes en el pressupost. Per aquest motiu tot i existir millors models de robot en el mercat, s'optarà per utilitzar el que hi hagi disponible des de l'inici.

1.3 Abast i riscos

Aquesta secció pretén definir el punt de partida del projecte i en quin punt es donarà per acabat. També es descriuen els diferents obstacles que poden sorgir durant el seu desenvolupament.

Per tal d'afrontar el problema plantejat es dividirà el projecte en dues parts principals. Per una banda necessitem un programa que s'executi sobre l'arquitectura del robot, capaç de rebre consignes, processar-les juntament amb les imatges capturades per les seves càmeres i, finalment, fer una homografia entre la regió d'interès senyalada per l'usuari i la imatge observada en l'entorn. En el cas que es trobi alguna coincidència, s'indicarà la direcció en la que es troba.

Per l'altra banda, farà falta una interfície a disposició de l'usuari, perquè aquest pugui enviar les ordres del robot. S'utilitzarà un dispositiu mòbil, que anirà connectat en xarxa amb el robot i des d'on es podrà capturar una imatge i marcar-hi una regió d'interès a cercar. Per tant, en aquest cas l'objectiu serà crear una aplicació mòbil senzilla i intuïtiva que permeti capturar imatges de l'entorn, col·locar-hi un marcador per seleccionar la regió d'interès i, per últim, realitzar la comunicació amb el robot per enviar les dades.

S'observa d'aquesta forma, que tot i tenir dos blocs separats i diferenciables, aquests s'hauran de comunicar.

1.3.1 Obstacles i riscos del projecte

Durant la planificació i realització del projecte s'hauran de considerar els possibles obstacles que poden aparèixer. A continuació es descriu breument quins poden ser els problemes que es podrien afrontar.

Comunicació entre terminals

Un dels obstacles que podria presentar-se durant el transcurs del projecte és la dificultat d'integració del sistema en totes dues plataformes. El principal problema pot aparèixer durant la comunicació entre tots dos dispositius. S'haurà de trobar, per tant, una via de comunicació efectiva i eficient, compatible i que pugui estar disponible en les situacions on s'usarà el sistema.

Algorismes usats

Un altre dels obstacles que podria aparèixer és el de la precisió dels algorismes usats. Tot i que els tests previs a la integració cercaran trobar el millor algorisme pel nostre cas, podria passar que aquest tampoc sigui suficientment precís com per considerar que el sistema és funcional. En aquest cas s'hauran de buscar altres tècniques durant el preprocessat que puguin ajudar a millorar els resultats del sistema.

Capacitat de còmput

Tot i que els programes utilitzats sempre buscaran obtenir un bon compromís entre precisió i cost computacional, pot donar-se el cas que el robot no tingui la suficient potència de càlcul com per poder executar els algorismes dins d'uns certs marges de temps. Trobats en aquesta situació, una possible alternativa podria ser la d'executar el programa principal en el *cloud* i entregar directament els resultats al robot. Si fos el cas, probablement el *bottleneck* del sistema seria la comunicació entre tots dos dispositius i el servidor, i s'haurien d'explorar mecanismes per reduir la latència.

Eficiència energètica

Molt relacionat amb l'obstacle anterior, un possible risc a considerar seria el d'un consum energètic excessiu del programa en el robot. En aquest cas la solució podria similar a l'anterior, realitzant part dels càlculs en un servidor extern, o bé migrar la part d'execució dels algorismes al dispositiu mòbil per tal de reduir la càrrega de treball sobre el robot.

Planificació

Tots els problemes mencionats anteriorment, sumats amb una planificació inadequada, podrien fer que el projecte no finalitzés en la data esperada. Per aquest cas existirien dues úniques solucions: augmentar les hores de treball estimades pel període de temps restant o endarrerir el deadline del projecte.

1.4 Metodologia

1.4.1 Mètode de desenvolupament

A causa de la naturalesa del projecte, en el que hi ha implicades diverses plataformes les quals han de comunicar-se, s'ha considerat com a més oportuna utilitzar una metodologia àgil. Aquesta ens permetrà tenir cicles de treball curts, que s'aniran validant a cada fase. També podrem conèixer en tot moment, i de forma més precisa, quin és l'estat de desenvolupament del projecte respecte la planificació. Tot i tractar-se d'estratègies pensades per treballar en equip, els seus conceptes es poden també aplicar en projectes on hi ha un únic desenvolupador implicat.

Per poder dur a terme el projecte correctament i de forma efectiva, s'ha fet una planificació de com es desenvoluparà. Aquesta planificació s'explicarà en els pròxims paràgrafs.

Per començar, s'haurà de fer un estudi previ dels algorismes de visió per computador disponibles. El procés de detecció de punts d'interès consta de tres fases fonamentals: extracció de keypoints en cadascuna de les imatges, descripció de features d'aquests i finalment un matching dels punts considerats propers. Per cadascuna de les parts s'ha de cercar quins són els que ofereixen millors resultats sota les condicions on s'espera que treballi el robot. Aquests seran, per ara, entorns com el del Campus Nord. Aquestes proves es duran a terme en un entorn simulat i imatges ja donades amb la finalitat de centrar-nos en els algorismes.

La següent fase serà la de desenvolupar l'aplicació pel dispositiu mòbil i establir una connexió amb l'entorn de simulació per realitzar proves. Però en aquest cas, amb imatges reals capturades pel dispositiu.

Finalment, l'últim pas serà el de migrar el programa utilitzat de l'entorn de simulació al robot, fent les adaptacions que siguin necessàries. Arribats a aquest punt es faran proves de rendiment del programa, avaluant els resultats obtinguts i optimitzar-los en la mesura que sigui possible. En el cas que s'obtinguin resultats

favorables, es pot realitzar una ampliació del projecte i passar a fer proves en mode stereo (dues càmeres) per poder determinar la distància a la que es troba l'objecte.

1.4.2 Eines de desenvolupament

El programa principal es desenvoluparà en *Python* fent ús de la llibreria *OpenCV*. Aquesta és la que ens permetrà aplicar les tècniques de visió per computador, ja implementades, de forma més senzilla. Un altre dels seus avantatges és que es tracta d'una llibreria open-source, disponible en diverses plataformes i amb un ampli suport per part de la comunitat. Tot i això, certs algorismes addicionals que incorpora estan sotmesos a patents i s'haurà de valorar segons el cas les possibles decisions que es poden prendre.

Pel que fa l'entorn de simulació s'utilitzarà *Docker*. Es tracta d'una eina de virtualització basada en containers que ofereix un entorn senzill, àgil, portable i segur, tot permetent accedir de forma directa als recursos del sistema al contrari de les màquines virtuals convencionals. Per la part de l'aplicació Android, el desenvolupament es farà per mitjà de la plataforma *Android Studio*.

Pel que fa al control de versions dels diferents codis es farà servir *Git* i *Bit-Bucket*, mentre que per l'organització de tasques pendents s'utilitzarà *Trello*.

1.4.3 Mètode de validació

En cadascun dels blocs principals que comprèn el projecte, descrits en l'apartat de metodologia, s'aniran fent proves del sistema desenvolupat fins aquell punt amb diverses imatges d'exemple.

Es realitzaran reunions quinzenals amb el tutor per validar la feina realitzada fins al moment i per la resolució de problemes que hagin pogut sorgir. Aquestes reunions es realitzaran amb més freqüència si la situació ho requereix o bé quan s'hagi de visitar el laboratori per fer proves amb el robot.

Finalment quan estigui tot el sistema integrat es faran proves en casos reals amb el robot per comprovar-ne l'eficàcia.

2. Planificació temporal

En aquesta secció s'explicarà la planificació temporal que es durà a terme pel projecte, les tasques de les que constarà i el pla d'acció, tot amb el fi d'acabar el projecte dins del termini estipulat. Tot i això, aquesta planificació es podria veure alterada per possibles contratemps que puguin sorgir durant el desenvolupament.

La data d'inici d'aquest projecte és el 19 de febrer, i com a data de finalització esperada el 17 de juny.

2.1 Descripció de tasques

2.1.1 Estudi previ

Aquesta va ser la primera tasca a realitzar en el projecte. Donat que aquest projecte tracta de fer una optimització i integració, era necessari fer un estudi previ abans de començar, per tal de tenir una noció clara del treball ja realitzat fins al moment.

Això implica familiaritzar-se amb les llibreries i mètodes utilitzats. En aquest cas concret, es tracta de OpenCV, llibreria open-source utilitzada en la visió per computador. Tot i tenir coneixements de visió, com que l'experiència amb l'ús d'aquesta llibreria era nul·la, es va haver de dedicar un temps a estudiar i entendre-la.

Adicionalment, es va haver d'analitzar l'estructura i funcionament del programari ja disponible per poder desenvolupar el sistema.

2.1.2 Planificació del projecte

Aquesta és la tasca abarca majoritàriament tots els continguts tractats en el mòdul de Gestió de Projectes (GEP) i inclou les següents parts:

- Abast del projecte i contextualització
- Planificació temporal
- Gestió econòmica i sostenibilitat
- Presentació preliminar
- Plec de condicions

Té com a data d'inici el 19 de febrer i d'acabament el 9 d'abril, amb una presentació final realitzada el 16 d'abril i una càrrega de treball aproximada de 90 hores. Té com a dependència l'estudi previ, que serà necessari per tenir unes nocions bàsiques de l'abast del projecte i poder fer la seva correcta planificació.

2.1.3 Configuració inicial de l'entorn de treball

Per tal de realitzar les primeres proves i tests serà necessari tenir un entorn de treball virtual, amb totes les llibreries i dependències necessàries correctament instal·lades. Arribats a aquest punt es faran proves per a la familiarització amb la plataforma.

També serà necessari tenir el software corresponent per redactar i lliurar la documentació del projecte. A continuació es mostra una breu llista dels components principals a instal·lar:

- **Desenvolupament:** Python, OpenCV, Docker, Atom, BitBucket, XQuartz
- **Mòdul de GEP:** TeamGantt
- **Documentació:** Microsoft Word, Microsoft Excel, L^AT_EX

2.1.4 Desenvolupament principal

El desenvolupament principal del projecte constarà de diverses fases diferenciades però estretament relacionades entre si.

Recerca d'informació i proves

Es farà una investigació dels algorismes disponibles, es buscarà si n'hi ha de nous i es faran les proves corresponents amb aquests per valorar els resultats obtinguts. Aquestes proves es faran inicialment en un entorn virtual simulat, on es provarà l'eficàcia del sistema a l'hora de localitzar objectes amb imatges vistes des de diferents angles o il·luminacions. Més endavant es passarà a fer proves amb vídeo en viu, on es posaran a prova el moviment, rotació i canvis d'escala dels objectes a localitzar.

Paral·lelament, durant les proves amb vídeo, s'hauran de considerar els temps d'execució. S'ha de poder garantir la finalització de les execucions dins d'uns marges de temps, ja que aquest software finalment anirà a bord d'un sistema en temps real.

Primera fase d'integració

Amb el fi de modularitzar la fase d'integració del sistema, aquesta es farà primer en el dispositiu mòbil. Així tindrem a disposició una aplicació mòbil, desenvolupada en aquesta fase, que ens permetrà capturar imatges i enviar-les a l'entorn simulat per tractar de trobar el punt d'interès dins les imatges que li seran proporcionades.

Segona fase d'integració

Es migrarà el software desenvolupat en l'entorn de simulació al robot, adaptant-lo a la seva plataforma i possibles limitacions que aquest pugui tenir. S'haurà d'adaptar també la comunicació entre el dispositiu mòbil i el robot per mitjà d'algun canal compatible amb tots dos.

2.1.5 Tasca final

Com a tasca final es farà una comprovació del sistema complet i es realitzaran proves per verificar el seu correcte funcionament. Si aquesta tasca finalitza abans del temps estimat, es pot considerar realitzar alguna de les ampliacions previstes a l'inici del projecte.

També es prepararà l'entrega final del projecte, assegurant que tota la documentació sigui correcta per tal de poder preparar la defensa.

2.2 Taula de temps

Tasca	Inici	Finalització	Hores	Risc
Estudi previ	01/02/2018	19/02/2018	20h	Baix
Planificació del projecte	19/02/2018	16/04/2018	100h	Mitjà
Configuració de l'entorn	19/02/2018	02/03/2018	20h	Baix
Recerca i proves	05/03/2018	26/03/2018	60h	Baix
Primera fase integració	27/03/2018	20/04/2018	120h	Mitjà
Segona fase integració	23/04/2018	15/05/2018	140h	Alt
Tasca final	16/05/2018	27/06/2018	80h	Mitjà

Taula 2.1: Tasques planificades

2.3 Diagrama de Gantt

A continuació es mostra el diagrama de Gantt amb les tasques planificades.

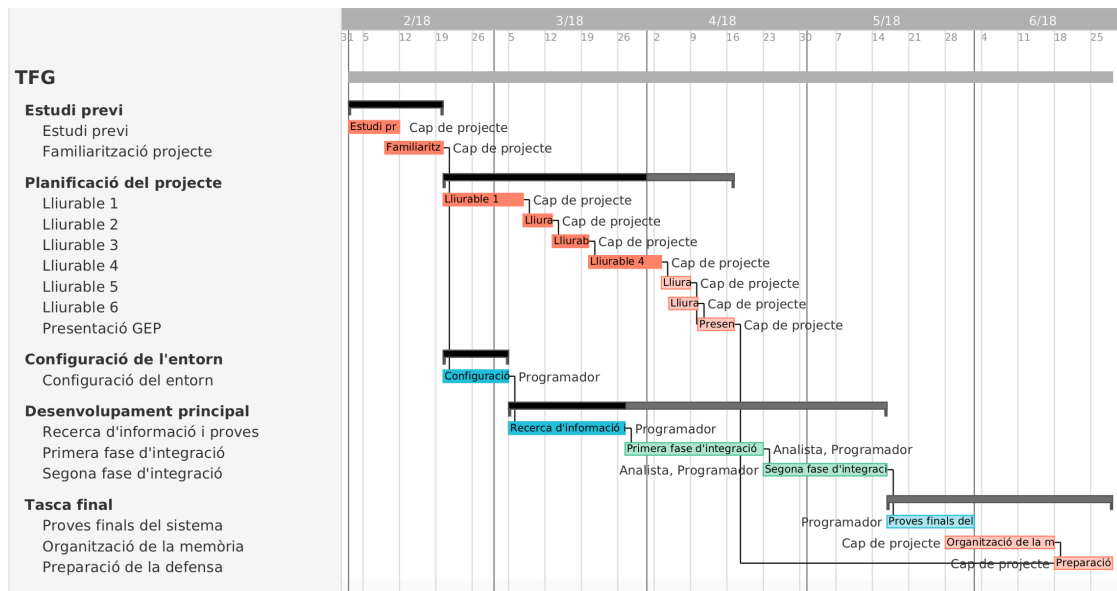


Figura 2.1: Gantt - Desenvolupament temporal de tasques

2.4 Recursos utilitzats

En aquesta secció detallarem els recursos utilitzats en aquest projecte en funció del seu tipus.

2.4.1 Recursos hardware

- Ordinador portàtil: Macbook Pro amb Intel Core i5 a 2,4 Ghz, 8GB de RAM i gràfica integrada Intel Iris. Utilitzat com a eina de desenvolupament per a totes les tasques.
- Dispositiu mòbil: Samsung Galaxy S9+. Utilitzat per una part de l'integració del software i la realització de les corresponents proves.
- Robot: utilitzat per l'altra part de l'integració i proves en l'entorn. El model base es tracta d'un Pioneer 3-AT [2]. A bord, porta instal·lat un Intense PC Pro Barebone [3].

2.4.2 Recursos software

- Sistema operatiu macOS: utilitzat per la realització de totes les tasques.
- Python: llenguatge de programació principal utilitzat durant el projecte.
- OpenCV: utilitzada pels algorismes de visió per computador aplicats.
- Android Studio: plataforma utilitzada durant el desenvolupament de l'aplicació.
- Microsoft Word: eina utilitzada per l'escriptura d'esbossos per les entregues.
- L^AT_EX: eina utilitzada per la presentació de les entregues.
- Microsoft Excel: eina utilitzada pel control d'hores de treball.
- Trello: utilitzat per la gestió i monitorització de tasques.
- Bitbucket: repositori pel control de versions del programari.
- Gantt Project: eina utilitzada pel desenvolupament del diagrama de Gantt.

2.5 Desviacions i pla d'actuació

En aquesta secció es discuteix com s'executarà la planificació explicada amb anterioritat.

La idea principal i l'objectiu serà dur a terme el projecte tal i com s'ha estipulat en la planificació, seguint l'ordre de tasques representat en el diagrama de Gantt.

Tot i això, poden sorgir imprevistos, com els comentats en la secció de possibles obstacles, que poden provocar un endarreriment de la planificació programada. Com que el temps disponible per a realitzar el projecte és limitat, en cas que ens trobem amb la possibilitat de no acabar dins del termini, s'haurà de fer una priorització de tasques per tenir la major part del sistema funcional.

Podria passar que les optimitzacions del sistema per millorar les deteccions no fossin les esperades i el temps dedicat s'allargués més de l'esperat. En aquest cas es prioritzaria tenir una versió bàsica del sistema integrada de forma completa, tot i que els resultats que ofereixi no siguin els òptims.

Pel que fa a la visió *stereo* del robot, per tal de determinar la distància, sembla un objectiu ambiciós i d'entrada no planificat. En el cas d'arribar a la fase de proves finals amb els resultats desitjats, seria interessant poder-ne fer una implementació.

Es farà una reunió amb el director cada dues setmanes, i en les fases on sigui necessari, setmanalment. Es farà una validació del treball fet fins al moment i una projecció de les tasques a completar abans de la següent reunió.

S'ha fet una estimació aproximada de les hores necessàries per realitzar aquest projecte. En total faran falta unes 540 hores, que repartides durant les 20 setmanes del quadrimestre, correspondran a 27 hores setmanals.

3. Antecedents

3.1 Visió per computador

La visió per computador [4] té com a finalitat dotar a les màquines de la capacitat de percebre i obtenir informació de l'entorn. Ho fa mitjançant l'extracció, anàlisi i processament de dades a partir d'imatges. Les seves aplicacions s'estan popularitzant cada vegada més a mesura que augmenta la potència del hardware que va a bord de sistemes encastats.

Avui dia s'utilitza en un ampli rang d'àrees, com per exemple les següents:

- Vehicles autònoms
- Realitat augmentada
- Reconeixement facial
- Restauració d'imatges
- Controls de qualitat
- Robòtica

El camp de la visió per computador que es treballarà serà el de la robòtica, on es buscarà dotar al robot de capacitat per percebre l'entorn i saber trobar-hi els objectes o regions d'interès indicades per l'usuari.

3.1.1 Algorismes de visió

Dins dels algorismes de detecció d'objectes existents [5], un dels més populars i reconeguts és el Scale-Invariant Feature Transform (SIFT) [6]. Tot i tenir moltes

variants, totes es centren en quatre fases essencials: detecció de màxims locals, localització de keypoints, orientació i descriptors dels keypoints. Un algorisme similar és el Speeded-up Robust Features (SURF) [7], que consisteix en una millora del SIFT. Essencialment té les mateixes fases, tot i que inclou lleugeres modificacions com, per exemple, la detecció de keypoints inicials.

Tots dos algorismes comentats amb anterioritat són clàssics i tenen un ampli suport per a diverses plataformes. Mentre el primer és invariant a la rotació, el segon dóna bons resultats en canvis d'il·luminació. Tots dos proporcionen robustesa a les oclusions, ja que treballen amb màxims locals, i tenen un cost computacional molt baix. Altres algorismes de visió més recents poden ser el BinBoost [8] o el Learned Arrangements of Three Patch Codes (LATCH) [9].

A continuació s'expliquen més en detall els algorismes de visió utilitzats durant el projecte, alguns mencionats prèviament. Podem dividir-los en dos grans grups: detectors de keypoints i descriptors de característiques. Però com veurem més endavant, alguns algorismes duen a terme totes dues funcions.

Features from Accelerated Segment Test (FAST)

Algorisme detector de keypoints publicat com a alternativa computacionalment eficient a les existents [10], enfocat a detectar keypoints de forma molt ràpida a canvi de sacrificar part del rendiment.

Comença seleccionant un *threshold* τ determinat. Per cada píxel de la imatge, calcula la seva intensitat I_p i busca entre els seus veïns un conjunt de n píxels amb una lluminositat major ($I_p + \tau$) o menor ($I_p - \tau$). Si es compleix la condició, aquell punt de la imatge és considerat un *keypoint*.

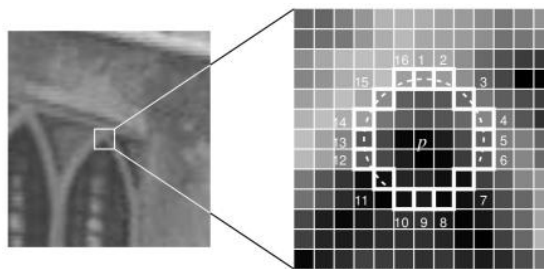


Figura 3.1: Inspecció de píxels veïns

L'algorisme disposa de mecanismes per optimitzar el procés, i no haver d'inspeccionar tots els píxels veïns.

Binary Robust Invariant Scalable Keypoints (BRISK)

El BRISK és un algorisme complet, que realitza tant la funció de detecció de keypoints com la descripció de característiques. Busca oferir un bon equilibri entre rendiment i cost computacional [11].

La part de detecció és molt similar a la del FAST. Per assolir invariància al canvi d'escala, s'aplica sobre una piràmide d'escala de la imatge, de la que s'interpolen els valors obtinguts en cada capa. A cada keypoint detectat se li aplica un mecanisme que determina la seva direcció, per poder normalitzar-ne la orientació. Així aconseguix robustesa a les rotacions.

Pel procés de descripció de característiques, es fa servir un patró de cercles concèntrics del que se n'agafen n punts. El descriptor BRISK es forma amb una cadena de caràcters binària, resultant de realitzar comparacions d'intensitat per cada parell de punts.

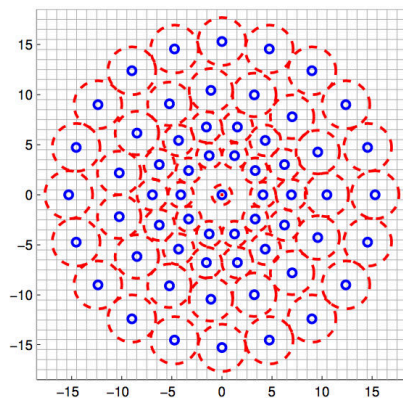


Figura 3.2: Patró de cercles concèntrics

Harris

Es tracta d'un detector de keypoints. La idea bàsica de l'algorisme és que detecta cantonades i arestes presents en la imatge. Per fer-ho recorre tota la imatge aplicant una finestra de convolució amb una distribució Gaussiana, mesurant els canvis d'intensitat en els dos components principals de la imatge (λ_1 i λ_2). Com el que mesura són les variacions entre píxels i no els valors absoluts, és invariant als canvis d'il·luminació.

Utilitza una funció (R) per calcular una puntuació per cada píxel. D'aquesta

manera determina de quin tipus de regió es tracta: cantonada, aresta o regió plana.

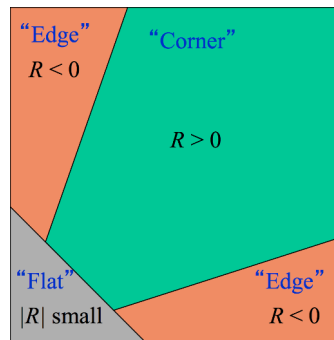


Figura 3.3: Regions de la imatge

Aquest detector, però, no funciona bé amb els canvis d'escala. El motiu és que a diferents escales una mateixa regió de la imatge pot ser considerada una cantonada o una aresta, en funció del grau de detall amb el que s'observi.

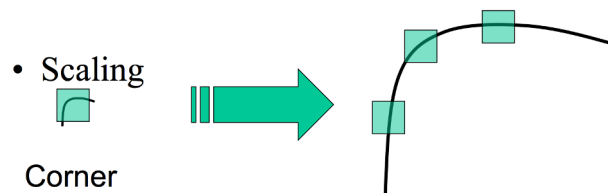


Figura 3.4: Canvi d'escala en una regió

Aquest fenomen el fa sensible a l'escala. Per donar solució al problema, es pot aplicar Harris en diverses escales, fent una piràmide de la imatge original [12].

Shi Tomasi

L'algorisme Shi Tomasi es tracta d'un detector de keypoints que està fortament basat en el de Harris, amb una petita diferència. Aquest difereix de Harris en la funció utilitzada per calcular les puntuacions de cada píxel. En el seu paper, Shi i Tomasi demostren experimentalment com el seu criteri de puntuació ofereix bons resultats [13].

Star

Detector de keypoints que deriva del CenSurE (Center Surround Extrema). Utilitza una aproximació de dos nivells del filtre Laplacian of Gaussians (LoG), per detectar arestes en la imatge. És invariant a la rotació [14].

SIFT

El SIFT és un algorisme que fa de detector de keypoints i descripció de característiques. Aplica un filtre gaussià (Laplace of Gaussians) a la imatge en diverses escales, contruint un espai d'escales de la imatge en el que es busquen màxims locals.

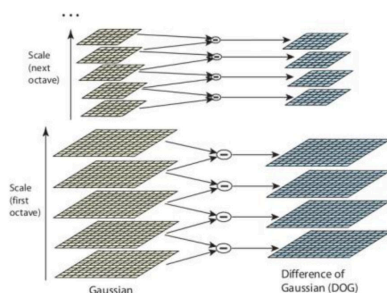


Figura 3.5: Diferència de gaussianes (DoG)

Treballar amb el filtre comentat és computacionalment costós de manera que el SIFT utilitza una aproximació del filtre, la diferència de gaussianes, per buscar màxims locals. Per trobar màxims cada píxel es compara amb els seus veïns, tant en l'escala en la que es troba com en les altres. Dels possibles keypoints candidats, se n'eliminen els menys robustos aplicant diversos mètodes. Finalment, amb cadascun dels keypoints trobats, es genera un histograma de gradients orientats, per obtenir la orientació del keypoint.

D'aquesta manera, per a cadascun d'ells, disposem de la seva ubicació, l'escala en la que es troba i la seva orientació. Aquesta informació permet al SIFT ser invariant a l'escala i la rotació. El pròxim pas és extreure característiques distintives dels keypoints que permetin identificar-los [15].

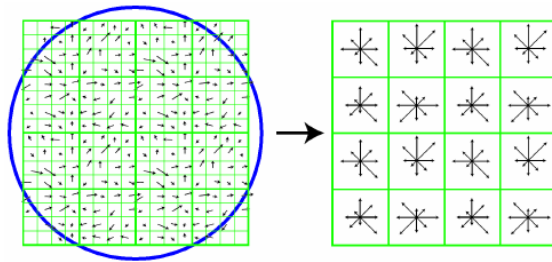


Figura 3.6: Descriptor de keypoint (SIFT)

La figura 3.6 representa gràficament un descriptor del SIFT. S'agafa una matriu de 16x16 píxels al voltant del keypoint i es calcula l'histograma d'orientacions, en 8 direccions, per cada sub-bloc 4x4 que formen la matriu. Obtenim com a resultat 128 valors, representats un vector de característiques.

SURF

Igual que el SIFT, el SURF és un descriptor i detector. Sorgeix de la necessitat de cobrir la major debilitat del SIFT: la seva alta complexitat computacional. En lloc de aproximar l'espai d'escala de la imatge amb la diferència de gaussianes ho fa mitjançant *box filters*, que es tracten de filtres de convolució lineals. Es poden calcular molt fàcilment amb l'ajuda de la integral de la imatge, i és possible aplicar-los en paral·lel per a cadascuna de les escales.

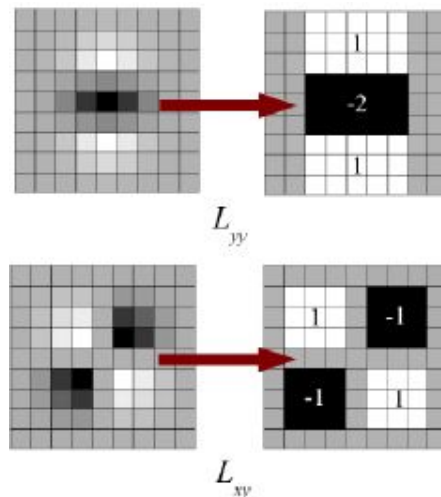


Figura 3.7: Box filter

Els keypoints són identificats calculant el determinant de la matriu Hessiana,

que expressa les variacions locals dins d'una regió. Per descriure els keypoints es calcula la seva orientació i s'hi aplica l'anomenat *Haar wavelet*, una seqüència de filtres lineals simples i fàcils d'aplicar.

Fast Retina Keypoint (FREAK)

El FREAK es tracta d'un descriptor de característiques binari, igual que el BRISK, inspirat en el sistema de visió humana. Té un funcionament molt similar al BRISK. Es comparen les intensitats dels punts d'un conjunt n per formar una cadena de caràcters binària. Es diferencia del BRISK en el patró que utilitza per seleccionar els punts del conjunt [16], on la densitat de punts al voltant del centre és major.

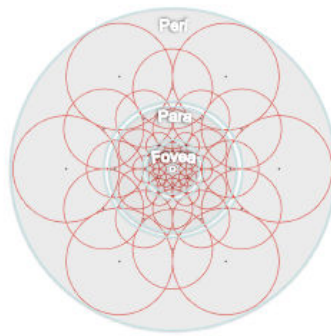


Figura 3.8: Patró de selecció FREAK

DAISY

Descriptor de característiques locals basat en histogrames d'orientació de gradients, igual que el SIFT. La reutilització dels histogrames calculats en un donat punt pels seus veïns fa que sigui molt eficient computacionalment [17].

LATCH

Es tracta d'un descriptor de característiques binari. Ofereix millors resultats que els altres descriptors de la mateixa categoria amb temps d'execució similars. A diferència del BRISK o del FREAK, el LATCH compara triplets de sub-porcions, o *patches*, de dimensió $k \times k$ de la imatge. L'algorisme evalua amb una funció binària la similitud de tots tres patches [18].

3.1.2 Sistema de detecció

El procés de matching entre totes dues imatges es pot dividir en quatre fases principals: preprocessat de les imatges, obtenció de keypoints de l'imatge, extracció de característiques i matching de punts.

Preprocessat de les imatges

El preprocessat digital de les imatges és un procés pel qual s'apliquen diverses transformacions amb l'objectiu de poder extreure informació més fàcilment. Permet reduir les distorsions i el soroll presents o bé destacar certes parts.

Entre les diverses tècniques de preprocessat que podem aplicar trobem:

- **Filtres.** L'aplicació de filtres de convolució a cadascun dels píxels redueix la variació entre ells per suavitzar la imatge i disminuir el soroll.
- **Escala de grisos.** Conversió de cadascun dels píxels de l'imatge a un rang de valors entre 0 i 256. Molts dels algorismes de visió treballen amb imatges en escala de grisos.
- **Equalització de l'histograma.** Per incrementar els contrast, es pot treballar sobre els histogrames de la imatge.

L'única tècnica de preprocessat aplicada en les imatges del projecte és l'escala de grisos. Segons les conclusions obtingudes en l'estudi previ, l'aplicació d'altres tècniques produïen iguals o pitjors resultats respecte de l'imatge resultant, tot afegint una càrrega de còmput addicional.

Obtenció de keypoints

A partir de les imatges preprocessades, s'aplica l'algorisme d'obtenció de keypoints. Aquest busca punts distintius en la imatge que permetin més endavant caracteritzar-la.

Extracció de característiques

Donats els punts detectats en l'anterior fase, l'extracció de característiques s'encarrega de descriure aquests keypoints de forma que aquests es puguin identificar i relacionar entre ells. L'algorisme utilitzat en el projecte per aquesta fase és el SIFT.

Matching de punts

El pas final és detectar similituds entre els punts, i consegüentment, trobar parts coincidents entre les imatges. S'agafen els descriptors dels keypoints de totes dues imatges i es compara cada parell mitjançant una funció de distància. Es consideren iguals els punts amb una distància per sota d'un llindar determinat.

En aquest projecte, s'utilitza com a llindar l'anomenat Lowe's Ratio, que permet eliminar aproximadament un 90% dels falsos positius.

4. Detecció i selecció de punts d'interès

En aquest capítol es detallen els experiments realitzats en el estudi previ del projecte i els resultats obtinguts.

4.1 Selecció de l'algorisme

S'han realitzat diversos experiments amb els algorismes de visió per computador disponibles, per escollir-ne el més adequat pel projecte. S'han provat tant algorismes de detecció de keypoints com descriptors de característiques, per trobar-ne el parell detector-descriptor més efectiu.

Els algorismes de detecció de keypoints posats a prova són els següents:

- BRISK
- Harris
- Shi Tomasi
- Star
- FAST
- SURF
- SIFT

Per altra banda, com a descriptors de característiques es va fer ús dels següents algorismes:

- BRISK
- FREAK
- SURF
- DAISY
- LATCH
- SIFT

Cal tenir en compte que alguns algorismes poden realitzar tant la part de detecció de keypoints com la dels descriptors. Totes les imatges i vídeos han estat capturats en entorns del Campus Nord.

4.1.1 Selecció usant imatges estàtiques

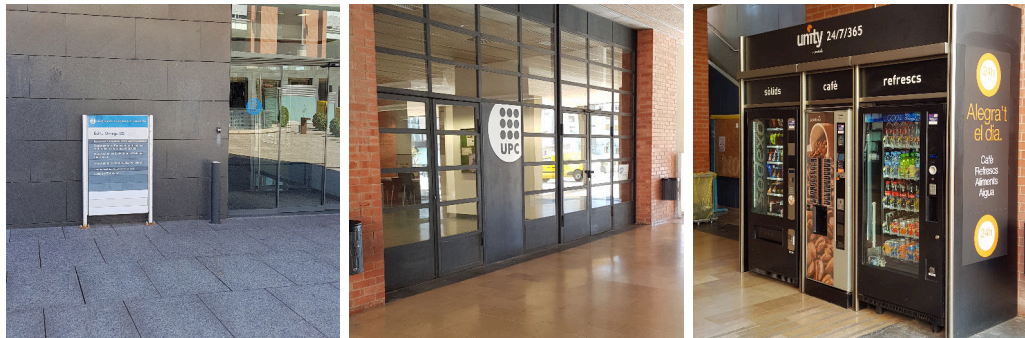
La primera part dels experiments s'ha realitzat amb dues imatges en el mateix escenari, simulant la imatge del mòbil i del robot. Es tracta d'una versió simplificada del problema per poder fer les primeres proves i analitzar de forma més precisa els algorismes. Donat que treballar amb imatges individuals té un cost computacional relativament baix, s'han utilitzat totes les combinacions d'algorismes possibles de les dues llistes mostrades amb anterioritat per fer les proves.

Entre les mètriques capturades per cada parell de algorismes, hi trobem el temps total, el nombre de matches, l'eficiència de cada match i el nombre total de keypoints de cada imatge.

Pel que fa la detecció dels matches correctes i falsos positius, s'han hagut de comprovar manualment ja que no existeix forma d'automatitzar el procés.

Les proves han estat realitzades amb imatges reals, dels entorns del Campus Nord. Per dotar de més rigorositat als experiments, les imatges s'han capturat des d'angles diferents i amb càmeres diferents, simulant la càmera del mòbil (Region of Interest (ROI)) i del robot.

A continuació mostrem algunes de les imatges de prova utilitzades, capturades amb el primer dispositiu.



(a) Cartell

(b) Sala d'estudis

(c) Màquina de vendent

Figura 4.1: Imatges capturades amb el primer dispositiu

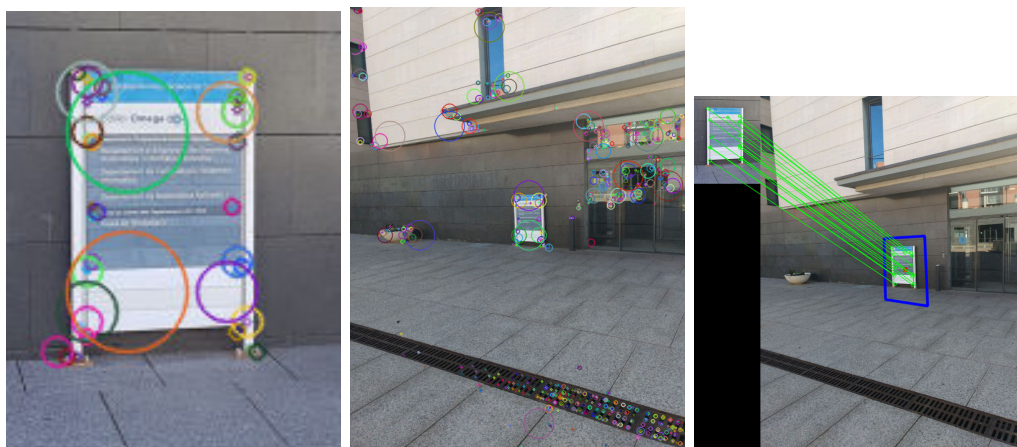
Aquestes són les corresponents imatges capturades des del segon dispositiu, amb angles i escales diferents.



(a) Cartell (b) Sala d'estudis (c) Màquina de vending

Figura 4.2: Imatges capturades amb el segon dispositiu

Per a cada parell d'imatges i combinació d'algorismes, s'han aplicat les fases de detecció de keypoints, descripció de característiques i matching, per trobar correlació i punts en comú entre ambdues.

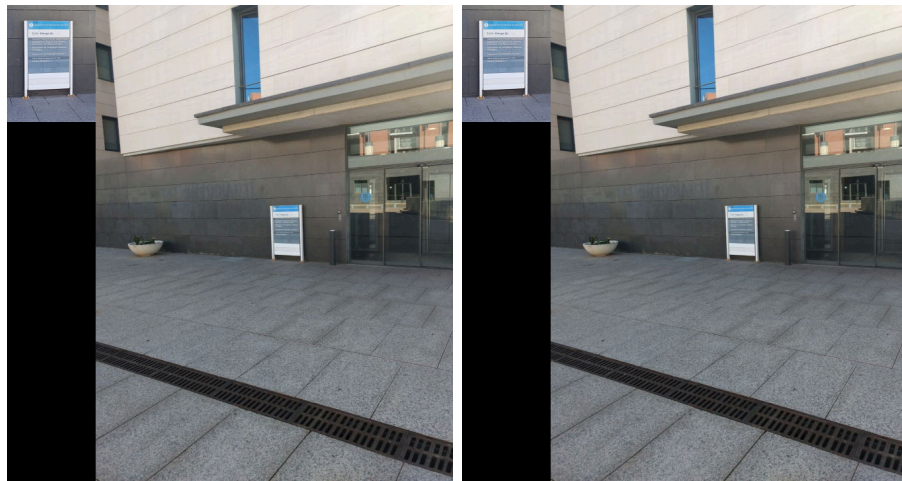


(a) Keypoints ROI (b) Keypoints ROB (c) Matching

Figura 4.3: Fases del procés de detecció

Moltes de les combinacions d'algorismes utilitzades en aquest experiment han estat descartades a causa de ser poc efectives a l'hora de realitzar el matching entre imatges i trobar punts en comú.

Algunes de les combinacions mencionades, com el BRISK + FREAK o el Star + LATCH, manquen a l'hora de detectar punts en comú.



(a) BRISK + FREAK

(b) Star + LATCH

Figura 4.4: Combinacions d'algorismes sense detecció

D'altres com, per exemple, Harris + DAISY o FAST + FREAK, troben alguns matchings entre imatges però aquests es tracten de falsos positius.



(a) Harris + DAISY

(b) FAST + FREAK

Figura 4.5: Combinacions d'algorismes amb falsos positius

Dels algorismes més efectius, destaquem els següents quatre per ser els que millors resultats proporcionaven de forma regular.

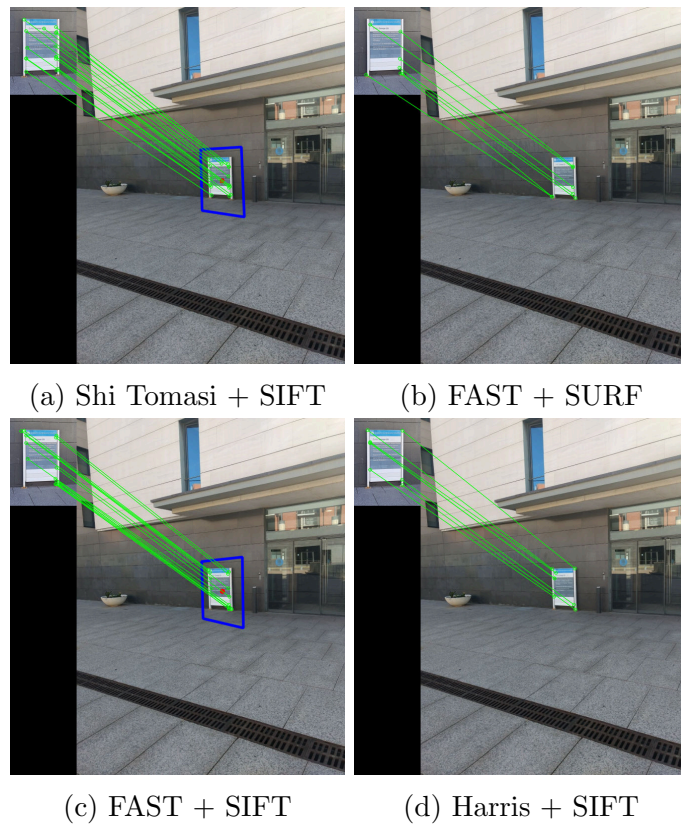


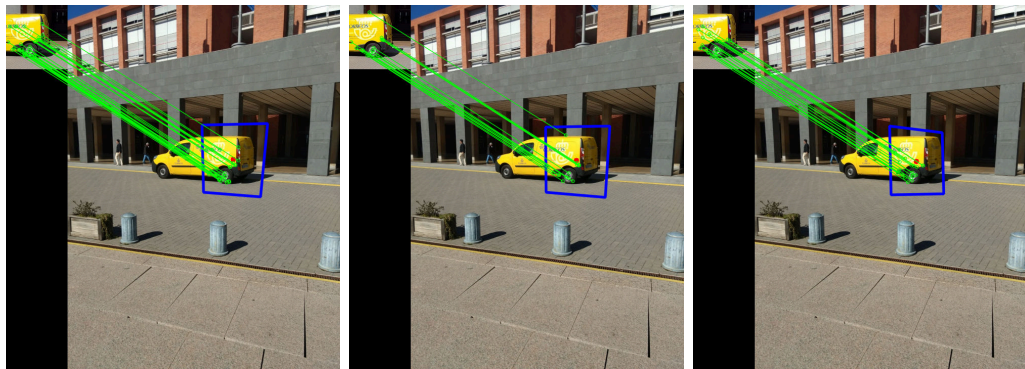
Figura 4.6: Resultats de combinacions amb matchings

A continuació es mostra una taula amb algunes estadístiques obtingudes de les imatges anteriors.

PS Alg	FE Alg	Time	Matches	ROI kp	ROB kp
Harris	SIFT	0.1057s	8	28	151
FAST	SIFT	0.0764s	17	32	387
FAST	SURF	0.016s	7	32	387
Shi Tomasi	SIFT	0.1649s	19	51	688

Taula 4.1: Resultats d'execucions de la figura 4.6

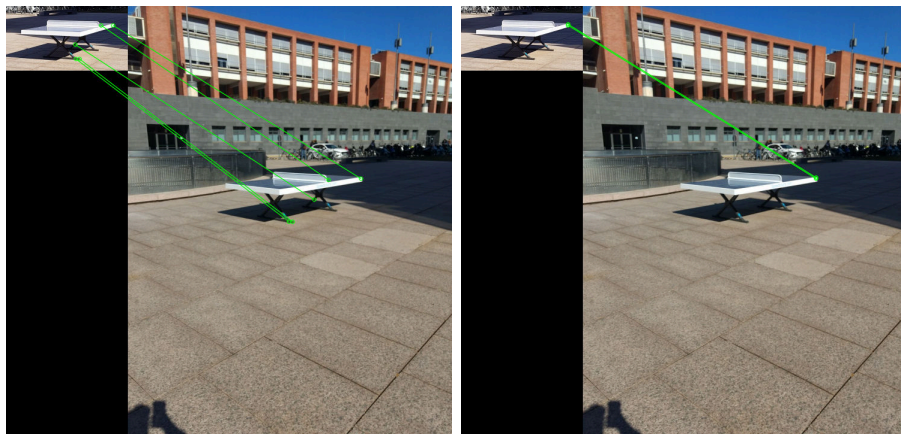
Com a conclusió obtinguda en les primeres proves, el millor algorisme d'extracció de característiques, amb diferència, és el SIFT. En determinats casos el SURF ofereix resultats similars. Els detectors de keypoints amb millor rendiment són Shi Tomasi i Harris, seguits del FAST.



(a) Shi Tomasi + SIFT (b) Harris + SIFT (c) FAST + SIFT

Figura 4.7: Comparació dels millors detectors

Els dos primers detectors obtenen resultats similars però, com podem observar en la taula 4.1, el nombre de keypoints trobats per l'algorisme Shi Tomasi és major. Això fa que tingui un temps d'execució lleugerament superior però, per contrapartida, té millors resultats en situacions on les imatges tenen pocs keypoints.



(a) Shi Tomasi + SIFT (b) Harris + SIFT

Figura 4.8: Comparació detectors Shi Tomasi i SIFT

4.2 Proves amb seqüències pregravades

La segona part es centra en escenaris més realistes, on per una banda tenim la imatge del dispositiu mòbil, i per l'altre tenim seqüències pregravades amb la càmera del robot. Els vídeos també han estat capturats en el Campus Nord, en les

mateixes ubicacions que les imatges, a més d'altres mostres addicionals obtingudes en els entorns del laboratori de l'IRI.

En aquesta part dels experiments ens centrem més en els temps d'execució i el seguiment d'objectes. Es realitzen canvis de perspectiva, escala i il·luminació per posar a prova els algorismes en casos extrems.

Per aquesta part dels experiments, s'han seleccionat els parells d'algorismes que millors resultats proporcionaven en l'anterior part. Aquests són:

- FAST + SIFT
- Harris + SIFT
- FAST + SURF
- Shi Tomasi + SIFT

Les mètriques de rendiment capturades són més detallades i variades en aquesta part dels experiments. Per a cada vídeo, s'ha generat una taula, que descriu les següents característiques per a cada fotograma: temps d'execució, nombre de matches, cost per match, nombre de keypoints detectats, coordenades de l'objecte relativa a la imatge, temps de la fase de detecció de keypoints, temps de la fase d'extracció de característiques i temps de la fase de matching.

En addició, s'emmagatzema un timeline amb tots els frames processats del vídeo, on apareixen els punts i matches detectats en aquest.

Finalment, amb totes les dades capturades es generen tres gràfiques per poder visualitzar millor la informació:

- **Matches.** Evolució dels matches en el transcurs del vídeo.
- **Time.** Acumulada del temps transcorregut fins a cada frame.
- **Position.** Gràfica de dispersió que indica les posicions de la imatge on s'ha trobat l'objecte.

Tots els scripts que generen aquestes mètriques i dades es poden trobar en el repositori del projecte.

4.2.1 Resultats en imatge i video

Partint dels resultats obtinguts en els experiments amb imatges, s'han seleccionat els quatre que millors resultats proporcionaven.

La finalitat d'aquest experiment és observar com funcionen els algorismes en seqüències de vídeo i com es comporten quan es produeixen canvis de perspectiva/escala, situacions que hauran de tractar quan el vídeo vingui donat per la càmera d'un robot en moviment.

A continuació es mostra una de les proves en vídeo realitzades durant l'experiment. S'ha seleccionat un fotograma arbitrari per poder comparar com es comporta cada algorisme en aquell instant de temps concret.

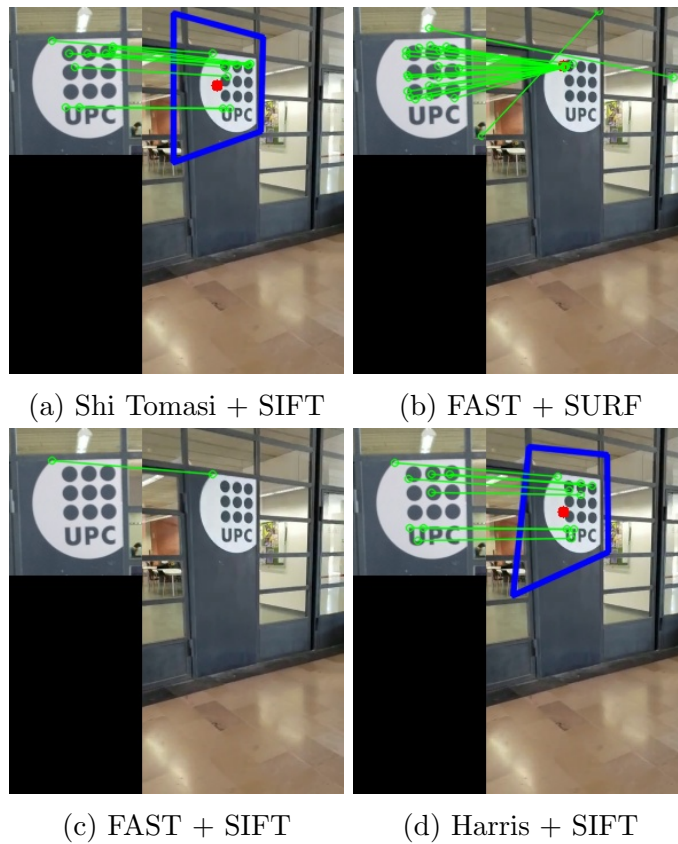


Figura 4.9: Fotograma d'exemple les proves en vídeo

El nombre de keypoints i matches detectats varia en cada cas. Tant la imatge 4.9a com la 4.9d troben correctament l'objecte en escena. En el cas de la figura 4.9b, troba un gran nombre de matches però molts no corresponen a punts iguals. La següent taula resumeix numèricament les estadístiques més rellevants d'aquestes quatre execucions.

Algorisme	Temps	Matches	Kp	PS time	FE time	M time
FAST+SIFT	0,0640s	1	492	0,0007s	0,0497s	0,0051s
FAST+SURF	0,0264s	9	522	0,0008s	0,0136s	0,0034s
Harris+SIFT	0,0376s	11	151	0,0028s	0,0254s	0,0010s
Shi+SIFT	0,0783s	13	280	0,0028s	0,0593s	0,0053s

Taula 4.2: Estadístiques del l'execució de fotograma 4.9

Les següents gràfiques fan referència a l'execució completa del vídeo, on l'eix d'abscisses representa la línia temporal dels frames. Mostren l'evolució dels matches trobats en la imatge del robot durant el vídeo. La línia horitzontal representa el mínim de matches establert per considerar que s'ha trobat l'objecte en escena.

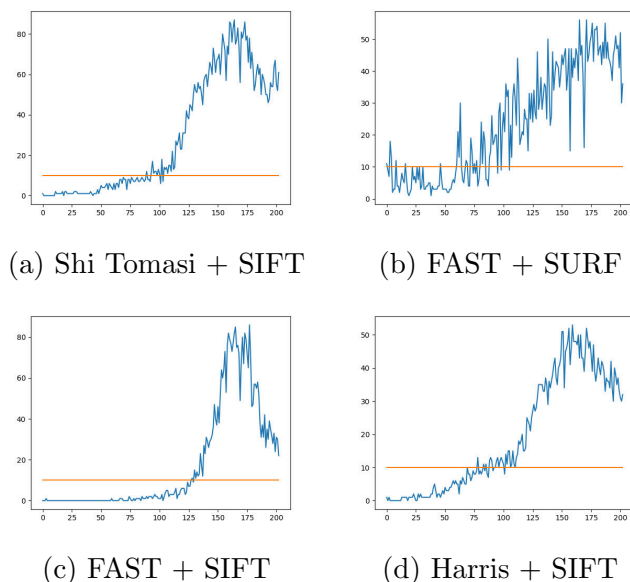


Figura 4.10: Comparació de matches de les proves en vídeo

Totes tres combinacions d'algorismes que utilitzen el SIFT com a descriptor de característiques presenten patrons molts similars i estables. Es diferencien entre ells pel nombre de matches trobats en cada cas i el moment en el que comencen a detectar l'objecte, factors que depenen del detector de keypoints. Pel contrari, l'algorisme SURF té un patró poc regular i amb molts falsos positius.

De forma anàloga a la taula 4.2, la següent mostra un resum de les quatre proves en vídeo amb els valors promitjos obtinguts en cada frame.

Algorisme	Temps	Matches	Kp	PS time	FE time	M time
FAST+SIFT	0,0775s	18	560	0,0005s	0,0522s	0,0062s
FAST+SURF	0,0394s	23	560	0,0005s	0,0103s	0,0043s
Harris+SIFT	0,0379s	19	154	0,0050s	0,0286s	0,0010s
Shi+SIFT	0,0591s	29	289	0,0055s	0,0498s	0,0034s

Taula 4.3: Estadístiques promitjes de tots els frames de vídeo

Després de múltiples proves, s'ha arribat a la conclusió que l'algorisme més adequat pel projecte és el detector Shi Tomasi, juntament amb el descriptor SIFT. Ha estat, amb diferència, el més regular, el més robust i el que menys falsos positius mostrava a l'hora de detectar matchings entre imatges. Tot i ser més costós a nivell de còmput, prioritzarem una correcta detecció envers el temps empleat.

A continuació es mostra un fotograma d'exemple, on l'algorisme seleccionat destaca per sobre d'altres provats.

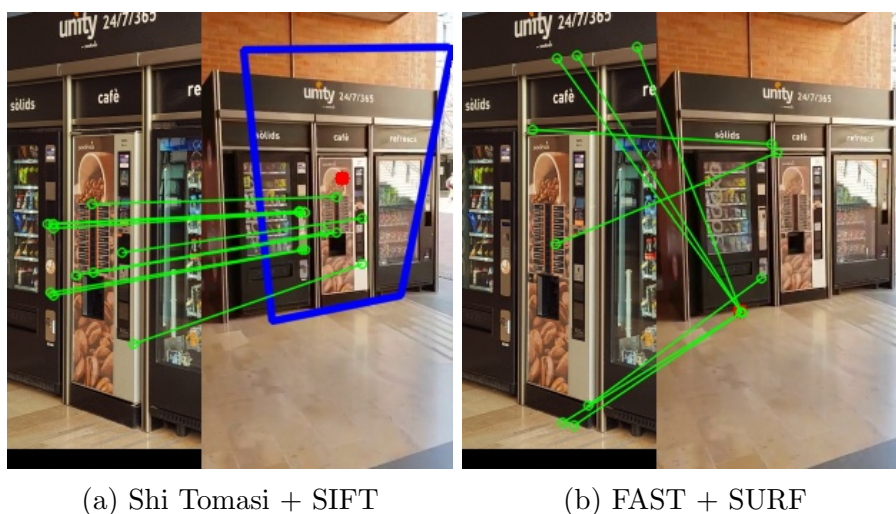


Figura 4.11: Comparació d'algorismes en un frame de vídeo

Com podem observar en la imatge 4.11a, l'objecte i el seu centre es detecten correctament. Per altra banda, en el cas 4.11b, no s'arriba a detectar la posició de l'objecte i molts dels matches indicats es tracten de falsos positius.

Les següents gràfiques són una representació de les dades registrades pel cas 4.11a. La primera ens indica com evoluciona el nombre de matches trobats durant el transcurs de la seqüència.

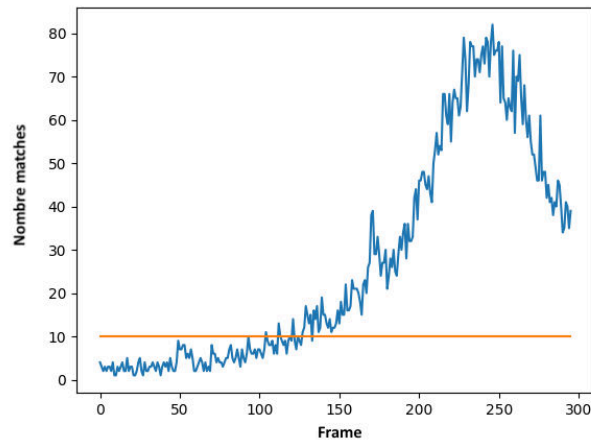


Figura 4.12: Frame/Matches

Observem que a partir del fotograma 125, el nombre de matches és el suficient com per considerar que l'objecte de la regió d'interès s'ha trobat en el fotograma de vídeo. Podem també destacar que un cop trobat l'objecte, no es perd el seu *tracking* fins arribar al final. Tot i això, quan el robot s'apropa molt a l'objecte el rati de matches baixa bruscament a causa del gran canvi d'escala i perspectiva que es produeix.

La segona gràfica representa el temps transcorregut en cada frame de vídeo. Aquesta s'ha generat amb la finalitat de trobar possibles relacions entre el nombre de matches detectats i el temps d'execució de l'algorisme.

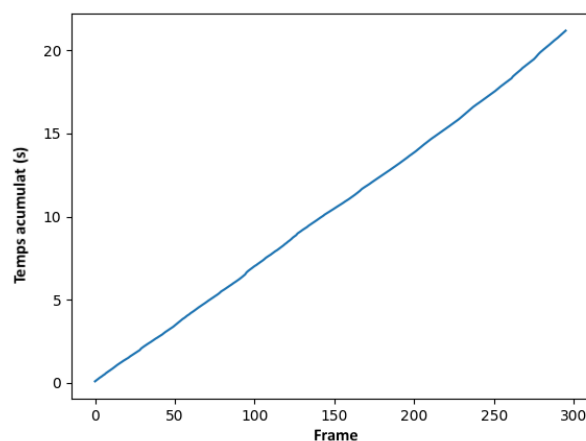


Figura 4.13: Frame/Time

Si comparem les dues figures anteriors, veiem que el temps d'execució té una alta linealitat, de manera que el nombre de matches trobat en cada cas hi té poca influència.

Finalment, la última gràfica representa en un pla de coordenades les posicions on s'ha detectat l'objecte durant tot el vídeo.

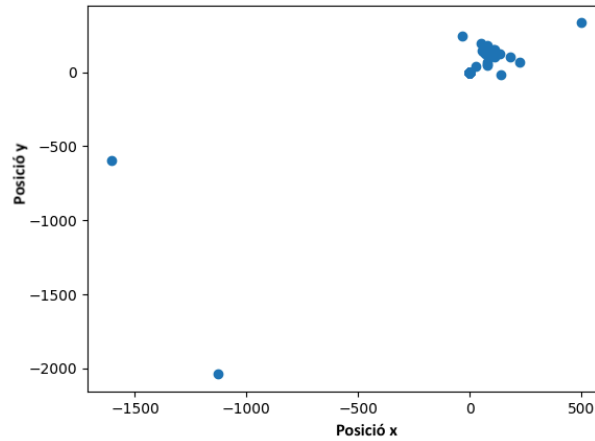


Figura 4.14: Posició dels matches

A excepció d'un parell de *outliers*, la resta de punts s'acumulen en una mateixa regió, donant un fort indicatiu que la imatge de referència es troba realment en aquella posició del vídeo.

5. Disseny i arquitectura del sistema

5.1 Esquema general

L'esquema de disseny general del sistema constarà de dos components principals: un dispositiu mòbil i un robot.

L'usuari es comunicarà amb el primer a través d'una aplicació Android creada específicament pel nostre sistema. L'aplicació rebrà com a *input* les indicacions donades per l'usuari, amb les que treurà com a *output* una imatge. Aquesta imatge de referència serà enviada al robot mitjançant un *socket* i el protocol Transmission Control Protocol (TCP).

La part del sistema integrada en el robot tindrà dos punts d'entrada de dades: la imatge de referència del mòbil i la seqüència gravada per la seva càmera. A partir d'aquestes dades el robot generarà com a sortida les coordenades d'imatge on es troba la referència dins de la seqüència que captura, sempre i quan aquesta es trobi present en l'escena. Les coordenades serviran més endavant per prendre decisions en quant al moviment del robot.

5.2 Software robot

5.2.1 Robot Operating System

Amb l'objectiu de fer el sistema portable i integrable en diversos robots, s'ha utilitzat el *framework* Robot Operating System (ROS) [19] per fer una abstracció al hardware. Aquest ens facilita el desenvolupament de sistemes complexos que

funcionen sobre robots en temps real i aporta robustesa.

El seu funcionament és similar a un pseudo-sistema operatiu organitzat en *packages*. En termes de ROS, un *package* és un mòdul que pot contenir diversos nodes ROS, dades, llibreries instal·lades i arxius de configuració entre altres. Permeten reutilitzar software de forma més senzilla i gestionar dependències dins d'un mateix projecte. Aquests es poden gestionar manualment o bé mitjançant l'ús d'eines com Catkin.

Un node ROS constitueix un procés que realitza un còmput determinat dins del sistema. L'ús de nodes té diversos beneficis pel sistema. Per una part ofereix robustesa a fallades, ja que els possibles errors queden aïllats a un node específic. Per altra banda, redueix la complexitat de codi en comparació a sistemes constituïts per un únic procés que engloba totes les funcionalitats.

Els diferents nodes en funcionament formen un graf i poden intercanviar informació entre ells mitjançant l'enviament de missatges, per uns canals determinats anomenats *topics*. Els *topics* són busos identificats per un nom en el que diferents nodes generen i consumeixen dades anònimament, de forma que aquests només coneixen el *topic* amb el que interactuen. Per poder generar dades un node realitza l'operació de *publish*, mentre que per llegir totes les dades escrites realitza un *subscribe*. Diversos nodes poden escriure i llegir dades en un *topic* de forma simultània.

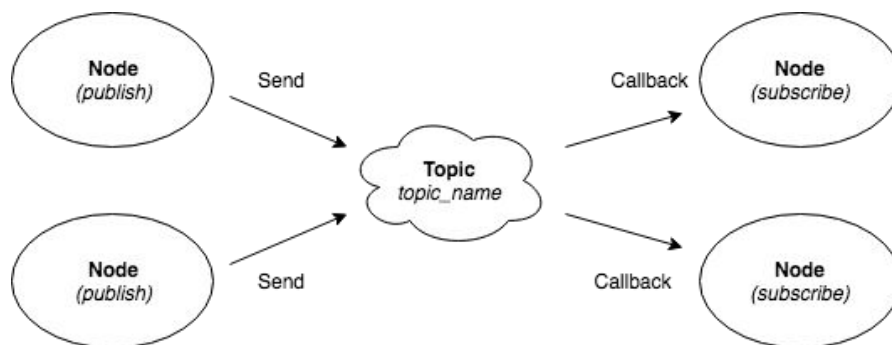


Figura 5.1: ROS topics

Altrament, la comunicació entre nodes es pot fer també per mitjà de *services*. En els *services*, un dels nodes té el rol de client (petició) i un segon node fa de servidor (resposta). El primer realitza peticions al servidor, que respon amb un altre missatge.

La diferència fonamental és que els *topics* estan pensats per a fluxos continus de dades mentre que els *services* per a l'enviament de missatges puntuals. En

el nostre cas, el nostre sistema ROS estarà constituït per cinc nodes. Cadascun s'encarregarà de realitzar una funció específica i diferenciable dins del conjunt.

A continuació es mostra una figura amb l'esquema de nodes, ressaltats en groc, que constitueixen el nostre projecte i els canals pels que es comuniquen.

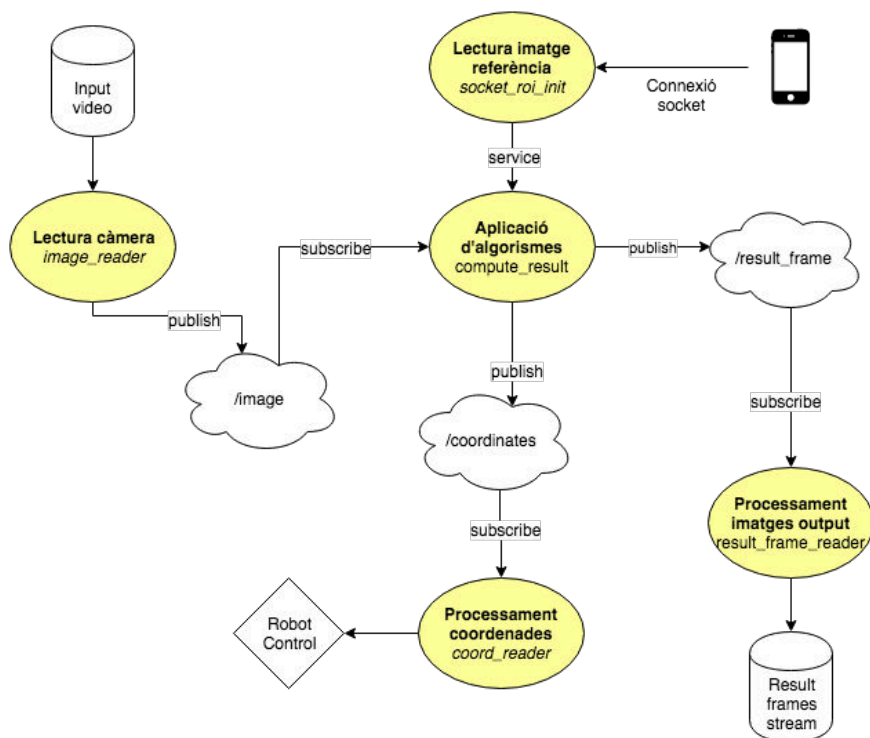


Figura 5.2: Esquema ROS del robot

S'ha optat per realitzar aquesta organització dels processos per tal de desacoplar els punts d'entrada i sortida de dades amb el nucli del sistema, de manera que sigui més fàcil realitzar modificacions en el cas que hi hagi canvis en la forma de llegir o mostrar les dades.

5.3 Aplicació Android

El rol de l'aplicació Android serà el de comunicar-se amb el robot, per tal de poder enviar noves imatges de referència a buscar en l'entorn de forma senzilla, ràpida i intuïtiva. A continuació s'especifiquen els requisits funcionals que aquesta ha de complir.

5.3.1 Requisits funcionals

Diagrama de casos d'ús

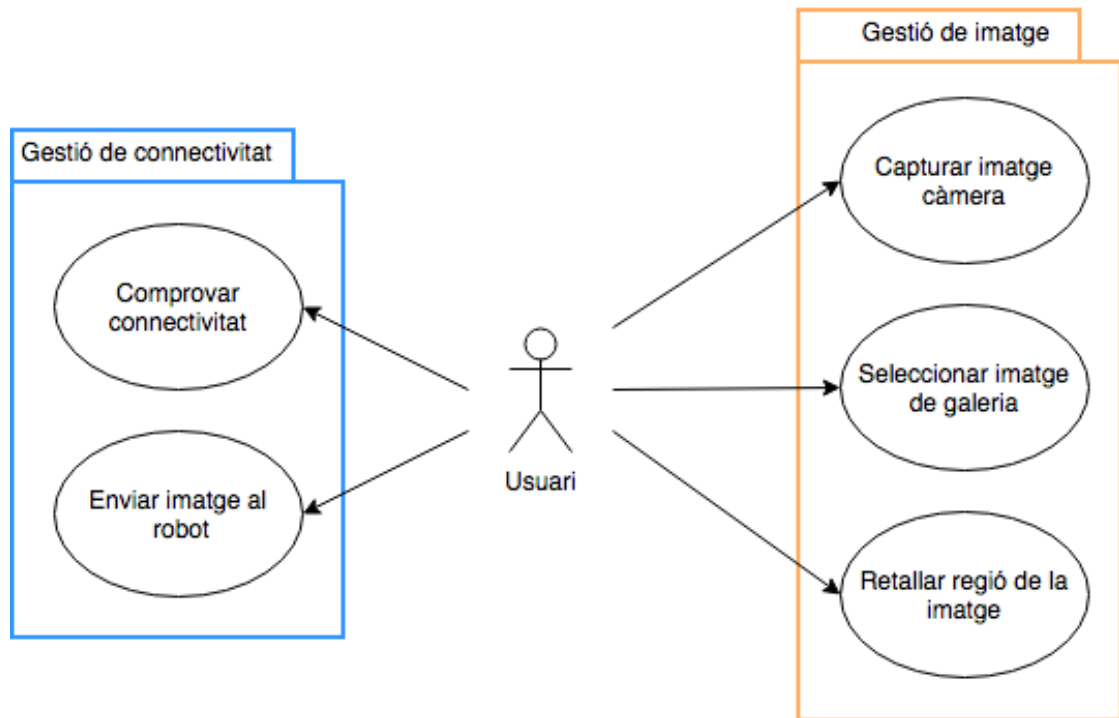


Figura 5.3: Diagrama de casos d'ús

Especificació completa de les funcionalitats

Cas d'ús:	1. Capturar imatge de la càmera	Actor principal:	Usuari
Precondició:	-		
Disparador:	L'usuari vol realitzar una fotografia, utilitzant la càmera del dispositiu.		
Escenari principal d'èxit:			
<ol style="list-style-type: none"> 1. El sistema mostra a la pantalla les imatges enregistrades per la càmera. 2. L'usuari enfoca l'objecte desitjat dins l'objectiu de la càmera. 3. L'usuari prem el botó de capturar per realitzar la fotografia. El sistema mostra a pantalla completa la imatge capturada per l'usuari. 			
Extensions:			
3.a. L'usuari vol tornar a capturar la imatge			
3.a.1. Es torna al punt 1			

Taula 5.1: Funcionalitat 1

Cas d'ús:	2. Seleccionar imatge de la galeria	Actor principal:	Usuari
Precondició:	-		
Disparador:	L'usuari vol seleccionar una imatge ja guardada en el dispositiu.		
Escenari principal d'èxit:			
<ol style="list-style-type: none"> 1. El sistema mostra a la pantalla una galeria amb les imatges guardades en el dispositiu 2. L'usuari selecciona una de les imatges mostrades. 3. El sistema mostra la imatge seleccionada a la pantalla. 			
Extensions:			
2.a. L'usuari vol tornar a seleccionar una imatge de la galeria			
2.a.1. Es torna al punt 1			

Taula 5.2: Funcionalitat 2

Cas d'ús:	3. Retallar regió de la imatge	Actor principal:	Usuari
Precondició:	L'usuari ha seleccionat prèviament una imatge, des la càmera o des de la galeria		
Disparador:	L'usuari vol seleccionar una regió de la imatge.		
Escenari principal d'èxit:			
<ol style="list-style-type: none"> 1. El sistema mostra a la pantalla la imatge en qüestió. 2. L'usuari selecciona la regió de la imatge que vol retallar. 3. El sistema mostra a la pantalla la regió seleccionada. 			
Extensions:			
2.a. L'usuari vol tornar a seleccionar una regió de la imatge			
2.a.1. Es torna al punt 1			

Taula 5.3: Funcionalitat 3

Cas d'ús:	4. Enviar imatge al robot	Actor principal:	Usuari
Precondició:	L'usuari ha seleccionat prèviament una imatge, des la càmera o des de la galeria, i aquesta ja es troba retallada.		
Disparador:	L'usuari vol enviar la imatge en qüestió al robot.		
Escenari principal d'èxit:			
<ol style="list-style-type: none"> 1. El sistema mostra a la pantalla la imatge en qüestió. 2. L'usuari dona l'ordre d'enviar la imatge al robot. 			
Extensions:			
2.a. La imatge no s'ha pogut enviar correctament al robot			
2.a.1. Es torna al punt 1			

Taula 5.4: Funcionalitat 4

Cas d'ús:	5. Comprovar connectivitat	Actor principal:	Usuari
Precondició:	L'usuari es troba a la pantalla de configuració.		
Disparador:	L'usuari vol comprovar l'estat de la connexió amb el robot.		
Escenari principal d'èxit:			
<ol style="list-style-type: none"> 1. Si l'usuari ja ha configurat la direcció IP prèviament, salta al punt 3. 2. L'usuari introdueix la direcció amb la que es vol comprovar la connexió. 3. El sistema comprova la connexió amb el robot i retorna el resultat a l'usuari. 			
Extensions:			
<ol style="list-style-type: none"> 2.a. La direcció IP introduïda per l'usuari és incorrecta <ol style="list-style-type: none"> 2.a.1. El sistema mostra un missatge d'error al usuari 			

Taula 5.5: Funcionalitat 5

5.3.2 Disseny d'interfície

L'interfície de l'aplicació s'ha de dissenyar de tal forma que sigui simple, elegant i usable. En base a les funcionalitats que havia de complir, s'ha dividit en tres blocs: galeria, càmera i configuració. L'usuari pot navegar per les diferents seccions utilitzant la barra de navegació inferior.



Figura 5.4: Vista principal

Les primeres dues permeten seleccionar la imatge desitjada i procedir amb l'enviament de les dades al robot. Pel que fa a la pantalla de configuració permet establir la direcció IP del dispositiu destí i realitzar una prova de connexió.

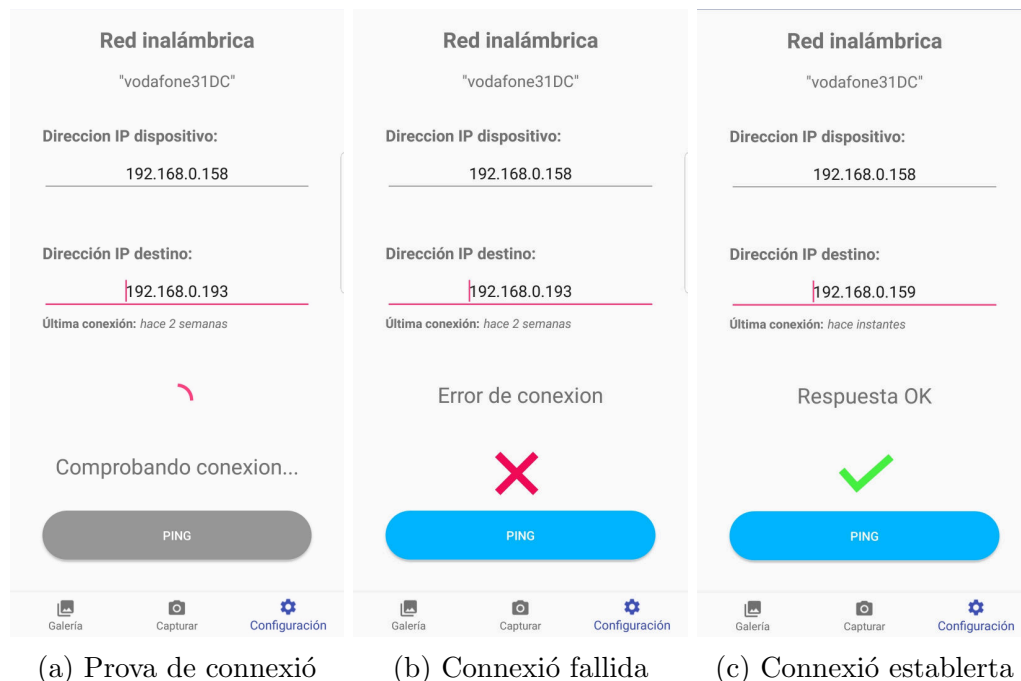


Figura 5.5: Pantalla configuració

Quant a l'enviament d'imatges, l'usuari té la opció de seleccionar-ne una de la galeria o capturar-la amb la càmera. Totes dues alternatives porten a la pantalla d'imatge prèvia. Allà es pot seleccionar una regió d'interès de la imatge i retallar-la. Finalment, es mostra el resultat final amb el botó d'enviar a disposició de l'usuari.

A continuació es mostren les interaccions a l'hora de seleccionar una imatge i enviar-la al robot.



Figura 5.6: Selecció i enviament d’imatge

A la cantonada superior dreta disposem de la icona de guardar, que permet guardar una imatge en la galeria del dispositiu per poder utilitzar-la més endavant. A la cantonada oposada tenim el botó de cancel·lar, pel cas que es vulgui canviar d’imatge. Es pot realitzar la mateixa acció a través del botó “enrere” natiu dels dispositius Android.

5.4 Comunicació entre dispositius

Tal com s’explica en l’esquema general del sistema, la comunicació entre tots dos dispositius es realitzarà mitjançant una xarxa WiFi amb una connexió *socket* TCP/IP. En aquesta secció s’exposaran les alternatives disponibles i el motiu d’aquesta elecció.

El requisit essencial del sistema de comunicació és que aquest sigui sense fils. L’escenari principal d’ús del sistema, un usuari enviant una imatge d’interès de l’entorn al robot, no pot ser satisfet de forma funcional per un mitjà de comunicació per cable. Altres requisits desitjables d’aquesta comunicació és que sigui ràpida, fiable i punt a punt.

Les dues tecnologies de comunicació disponibles en els dispositius, mòbil i robot, són el Bluetooth i WiFi. Tots dos sistemes de comunicació, basats en radiofreqüències i àmpliament estesos, presenten certs avantatges i inconvenients.

Per una part el Bluetooth es tracta d'una tecnologia de baix consum i senzilla, mentre que el Wifi és més complex i requereix de més hardware per poder funcionar. Això fa que el seu consum energètic sigui també més elevat. Aquest aspecte seria negatiu per la tecnologia Wifi, si no fos perquè aquesta ja es troba implantada en el robot pel seu ús en altres funcionalitats. D'aquesta manera, utilitzar-la per la comunicació en el sistema de visió no tindrà cap cost energètic addicional.

Per altra banda, els avantatges que ofereix el WiFi sobre el Bluetooth són diversos. Un d'aquests és el seu major radi de cobertura, que pot ser fins a 10 vegades major al del Bluetooth. Altres dos aspectes importants on el Wifi és superior són en el ample de banda i latència de la connexió.

Podem concloure que el mètode de comunicació més adequat pel projecte és la tecnologia WiFi. El router embarcat sobre el robot aixecarà una xarxa Wireless Local Area Network (WLAN), a la que es connectarà el dispositiu mòbil per poder enviar les dades, utilitzant el protocol de xarxa Internet Protocol (IP) pel que tots dos tindran una adreça associada. En concret, s'establirà una connexió socket utilitzant el protocol de transport TCP.

6. Implementació

6.1 ROS

Per l'implementació del software del robot en ROS, s'ha utilitzat la versió Python del framework. Les seves funcionalitats venen recollides en la llibreria *rospy*, que permet controlar i gestionar processos de ROS. L'altra principal llibreria utilitzada és OpenCV, que ens ofereix una sèrie de mètodes que ens faciliten el tractament i processat d'imatges.

Els packages de Python més utilitzats en el desenvolupament general del sistema són els següents: *numpy*, *tkFileDialog*, *CvBridge*, *sys*, *os*, *asyncore*, *socket*, *errno*.

A continuació es descriuran breument els diferents nodes implementats, la seva funció, l'ús que fan de les llibreries i com es comuniquen amb la resta de nodes.

Node **image-reader**

Es tracta del principal punt d'entrada de dades del sistema durant la fase de proves. Aquest node s'encarrega de llegir i processar l'informació de video per transmetre-la als altres nodes.

Per llegir l'informació s'utilitza la classe *VideoCapture*, pròpia de OpenCV. D'aquesta manera podem processar el video frame per frame i poder-los transmetre al següent node, en aquest cas, en forma de *topic*.

Per poder enviar informació en ROS, aquesta ha de tenir un format de missatge específic. El format de missatge es determina a l'hora de crear el vincle amb el *topic* adient, ja sigui en forma de *publisher* o *subscriber*. Existeix un format de missatge per defecte per les imatges.

Per convertir les imatges en format OpenCV al format missatge de ROS, tenim a la nostra disposició una classe de ROS: CvBridge.

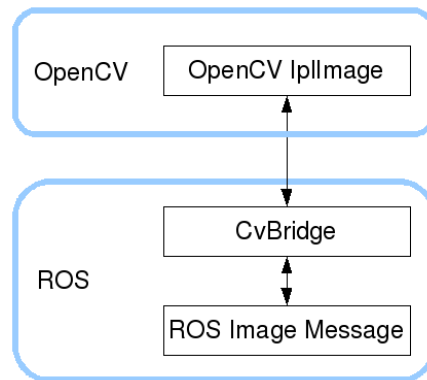


Figura 6.1: Imatge OpenCV a missatge ROS

Els missatges ROS generats mitjançant CvBridge són publicats en el *topic* “image”, a una freqüència de 10 Hz.

Node socket-roi-initializer

La principal funció d’aquest node és llegir noves imatges de referència enviades pel dispositiu mòbil i transmetre-les al node encarregat d’aplicar els algorismes de visió.

La comunicació amb el mòbil es fa a través de Socket TCP/IP, fent ús de les llibreries Python socket i asyncore. La direcció IP i el port per obrir la connexió es passen com a paràmetre en el node. Al començar la seva execució, s’estableix la connexió i el node queda estat idle, en espera de noves peticions.

Cada nova petició rebuda pel socket és tractada en una funció *handler*. Per conveni, les imatges les enviem codificades en forma de *string* en Base64. El node llegeix el *string* i el decodifica per poder tornar a disposar de la imatge. Aquest procés de decodificació el facilita la pròpia llibreria de OpenCV.

Com que el procés descrit anteriorment es fa en moments puntuals i no de forma periòdica, la transmissió de la imatge al següent node es fa utilitzant els *services* de ROS.

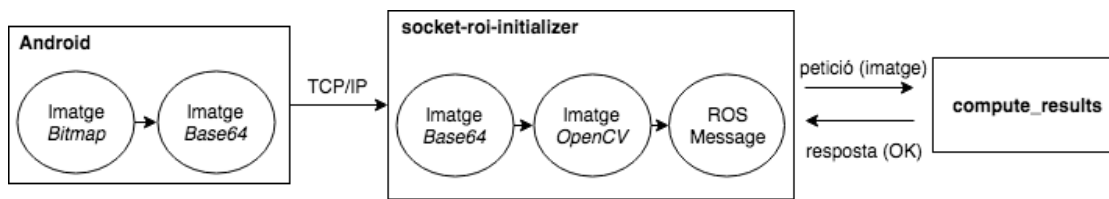


Figura 6.2: Socket ROI Initializer

A l'esquema mostrat veiem que el node en qüestió és el que fa de client, i realitza una petició amb la nova imatge rebuda. El receptor és el node “compute-results”, que fa de servidor. En aquest cas, com la petició del client no requereix de cap dada ni resposta, el servidor respon amb una confirmació. Novament, la imatge en format OpenCV haurà de ser convertida al format missatge de ROS amb CvBridge.

Node compute-result

És on es duen a terme la major part dels càlculs i l'encarregat de generar els resultats. En el graf de nodes de ROS és el que ocupa una posició central i, que per tant, es comunica amb tots ells. Com hem explicat amb anterioritat, rep dades del node “image-reader” per mitjà de *topic* i del node “socket-roi-initializer” per *service*.

Amb la imatge de referència rebuda, es realitza el procés explicat en la secció 3.1.2 i es guarda la informació dels keypoints i descriptors, ja que aquesta no variarà fins que es canviï d'imatge de referència. Per cada frame rebut s'apliquen els mateixos passos del procés de detecció.

El resultat és que per cada fotograma de video rebut, disposem dels seus keypoints i descriptors, a més dels de la imatge de referència que tenim pre-calculats. El procés de *matching* és el que, finalment, ens determina la correspondència entre ells. Del procés de *matching* obtenim:

- **Image resultant.** És una imatge generada on el col·loquen tant la imatge de referència com el frame de video. Es relaciona de forma visual cada punt trobat en la primera imatge que correspon amb algun punt de la segona. Quan el nombre de punts relacionats supera un llindar, podem dir amb certesa que l'objecte és present en la imatge i, per tant, es dibuixa també una *bounding box* a seu voltant.

- **Coordenades.** Quan el nombre de punts trobats és suficient, podem obtenir quines són les coordenades d'imatge de l'objecte. Aquestes s'obtenen aplicant un promig de tots els punts detectats. Per posteriors usos, les coordenades tenen com a origen el centre de la imatge.

Finalment aquests resultats s'envien a altres nodes, que els processen i realitzen les accions adients. Pel que fa a les imatges resultants es publiquen en el *topic* “result-frames”, mentre que les coordenades en el de “coordinates”. Per les coordenades, s'ha utilitzat un tipus de missatge personalitzat, que consta de dos enters per representar la “x” i la “y”.

Node coord-reader

S'encarrega de llegir les coordenades generades pel node anterior. És el que anirà connectat al hardware del robot per controlar-ne el moviment, de manera que aquest node pot canviar en funció del robot en el que es trobi.

Node result-frame-reader

Llegeix les imatges enviades pel node “compute-result” i amb la classe VideoWriter genera una seqüència en forma de video que guarda en disc.

6.2 Android

General

Pel desenvolupament de l'aplicació mòvil s'ha fet ús de la plataforma Android Studio. La SDK mínima fixada és la 22, que ofereix compatibilitat de l'aplicació amb el 62.6% [20] dels dispositius Android registrats i permet fer ús de funcionalitats del sistema operatiu relativament noves.

Vistes i processos

Per la vista principal de l'aplicació, s'ha optat per una implementació amb tres fragments: GalleryFragment, CameraFragment i ConfigurationFragment. Una

BottomNavigationView permet navegar entre les vistes.

Per les pantalles de tractament d'imatge, en canvi, s'ha optat per utilitzar activitats. L'esquema de transicions entre vistes seria el següent:

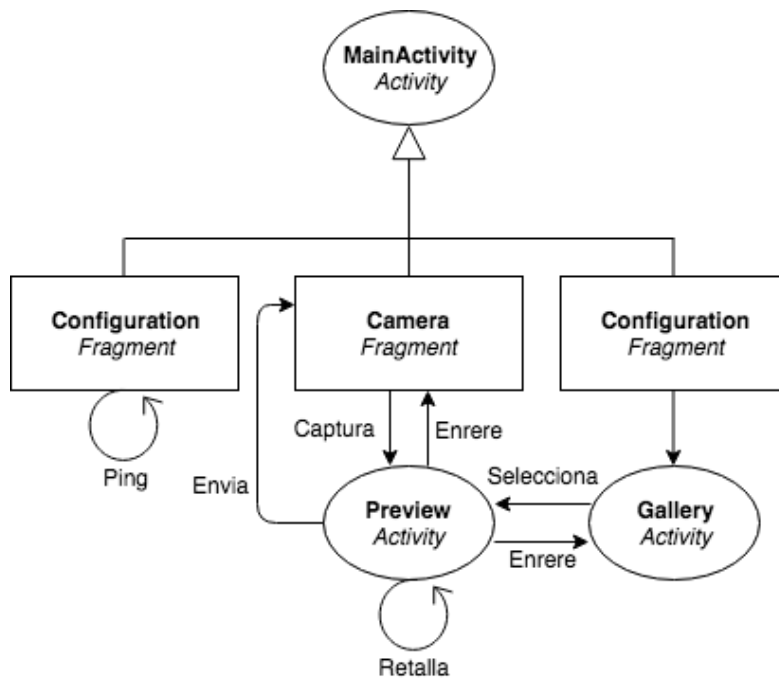


Figura 6.3: Esquema de l'aplicació FinderApp

Les tasques de enviar i *ping* són les que es comuniquen amb el robot, ja sigui per enviar dades o comprovar la connectivitat. Com que es tracten de tasques que es comuniquen amb un dispositiu extern, no podem garantir una durada determinada. Per a casos com aquest, Android ofereix una eina anomenada *AsyncTask*. Es tracta d'un procés asíncron, que s'executa en segon pla, evitant així que el procés principal encarregat de mostrar l'interfície quedi bloquejat.

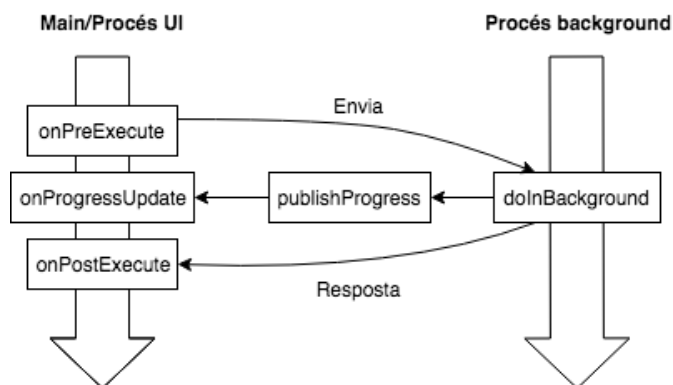


Figura 6.4: Funcionament AsyncTask

Permisos

La funcionalitat principal de l'aplicació és enviar imatges a un robot a través de la xarxa local sense fils. Per poder dur a terme tal tasca requereix connexió a Internet.

La funció de seleccionar una imatge de la galeria, necessita poder llegir el contingut de la memòria del dispositiu per realitzar els accessos adients.

Les imatges que es capturen des de la càmera poden ser guardades en el dispositiu, pel que novament faran falta permisos d'accés a l'emmagatzematge del dispositiu, en aquest cas per realitzar escriptures.

En resum, els tres permisos que farà falta demanar per oferir totes les funcionalitats completes seran els d'accés a Internet, lectura d'emmagatzematge intern i escriptura. D'aquests, l'únic imprescindible per oferir la funcionalitat principal és el primer.

6.3 Docker

Pel desenvolupament i proves del sistema, s'ha utilitzat Docker. Aquest ens permet crear un entorn de treball aïllat, de forma lleugera, modular i sense perdre l'accés directe als recursos del sistema. D'aquesta manera, podem anar construint una imatge personalitzada només amb les llibreries necessàries.

Les imatges de Docker es generen a partir d'un arxiu anomenat Dockerfile. Allà s'especifica la imatge base de Docker que es vol fer servir i tots els paquets i

llibries a instal·lar.

Pel nostre projecte, l'entorn de simulació té una imatge base amb Ubuntu 16.04. Aquestes són les principals paquets instal·lats:

- Python 2.7
- OpenCV 3.4
- ROS Kinematic

El Dockerfile amb les especificacions completes es troba disponible en el repositori del projecte.

7. Proves de camp

Les proves realitzades en la secció 4.2 han servit per poder simular escenaris on es trobarà el sistema. També han ajudat a determinar quins són els algorismes més adequats per a les tasques demandades. El següent pas serà el de realitzar els mateixos experiments amb la imatge en viu de robot, en lloc de les seqüències pregravades.

Aquestes proves, amb el sistema complet, s'han realitzat en el Barcelona Robot Lab, a l'IRI, on disposen dels diversos robots i equipament necessari. Tal com es va detallar durant la fase de planificació, el model utilitzat pels experiments ha estat un Pioneer 3-AT. Aquest porta a bord, un Intense PC Pro Barebone com a principal unitat de processament, on s'executarà el nostre sistema.

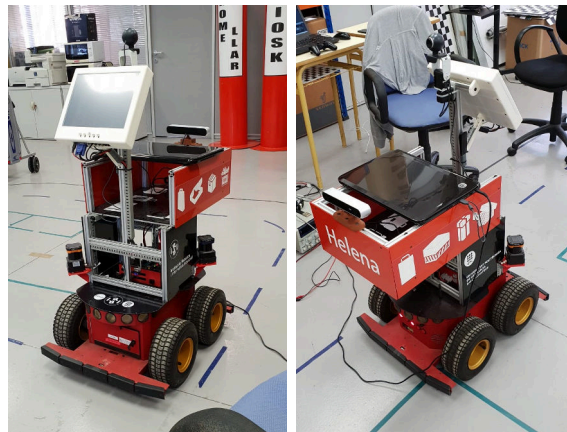


Figura 7.1: Robot de proves Pioneer 3-AT

Els tres factors clau del experiment han estat:

1. Inicialització del sistema amb la imatge enviada pel mòbil

2. Lectura i processament dels frames de la càmera
3. Interpretació dels resultats

Per tal de poder establir la comunicació entre dispositiu mòbil i robot, s'han connectat tots dos dispositius a la mateixa xarxa local sense fils, de manera que tenen visibilitat entre ells.

L'enregistrament d'imatges per part del robot es podia fer amb dues càmeres diferents, la primera una webcam i la segona la càmera ZED Stereo. Amb la webcam, de resolució mitjana-baixa, el sistema donava molts bons resultats i treballava amb fluïdesa. La ZED Stereo, en canvi, feia que el sistema funcionés lent i amb retard, a causa de la doble càmera amb major resolució d'imatge. S'ha optat per la webcam durant la resta de les proves, ja que l'únic avantatge significatiu que proporciona la càmera ZED (profunditat) no és, per ara, explotat pel sistema.

Com a conveni establert, el sistema de detecció dóna com a resultat les coordenades d'imatge de l'objecte buscat amb origen el seu centre. Aquestes faciliten l'addició de mobilitat al robot perquè les coordenades es trobaran centrades al seu camp de visió.

S'ha escrit un node que llegeix aquestes dades i n'extreu el desplaçament lateral de l'objecte respecte el centre del robot. D'aquesta manera podem convertir les coordenades d'imatge en velocitats angulars per les dues rodes. Com a resultat, el robot es desplaça cap a l'objecte si el detecta, centrant-lo en el seu punt de mira.



Figura 7.2: Escenari de l'experiment

Els experiments s'han realitzat amb diversos objectes, amb els que s'ha posat

a prova la detecció i posterior seguiment que feia el robot del seu objectiu. Totes aquestes dades s'han enregistrat amb una eina que ofereix ROS anomenada *rosvbag*. Ens permet seleccionar quins són els *topics* i canals de dades que es volen gravar. Tota aquesta informació és guardada en un fitxer anomenat *rosvbag*.

Més endavant, aquest fitxer es pot processar amb el software RViz, inclòs en l'instal·lació de ROS. Es tracta d'una eina gràfica que ofereix diverses visualitzacions i configuracions per mostrar les dades. D'aquesta forma es pot realitzar l'experiment un únic cop, per després efectuar el muntatge visual que més s'adeqüi a les nostres necessitats.

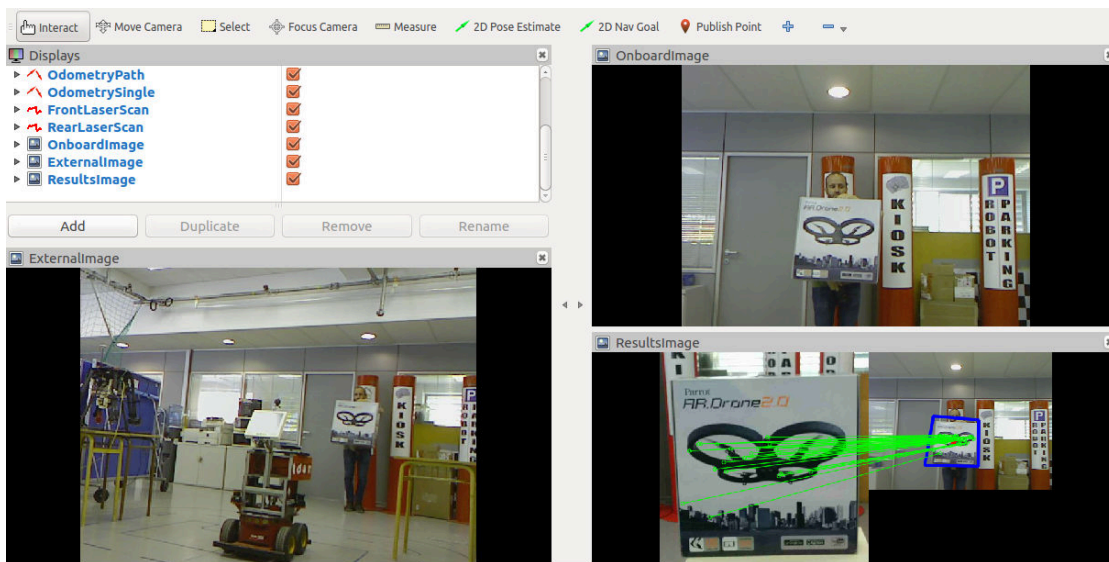


Figura 7.3: Finestra de visualització de RViz

En el muntatge de la figura 7.3 es mostren les dades de tres *topics*. A la finestra inferior esquerra s'observa la seqüència d'imatges capturada des d'una càmera externa al robot. Les dues finestres de la dreta són imatges des de la perspectiva del robot. La superior es tracta de la imatge sense processar provinent de la càmera. La inferior és la imatge de resultat que proporciona el sistema amb el frame de vídeo, la imatge de referència i els matches trobats.

La següent figura mostra un altre exemple de prova realitzada, en aquest cas, amb un objecte diferent a detectar.

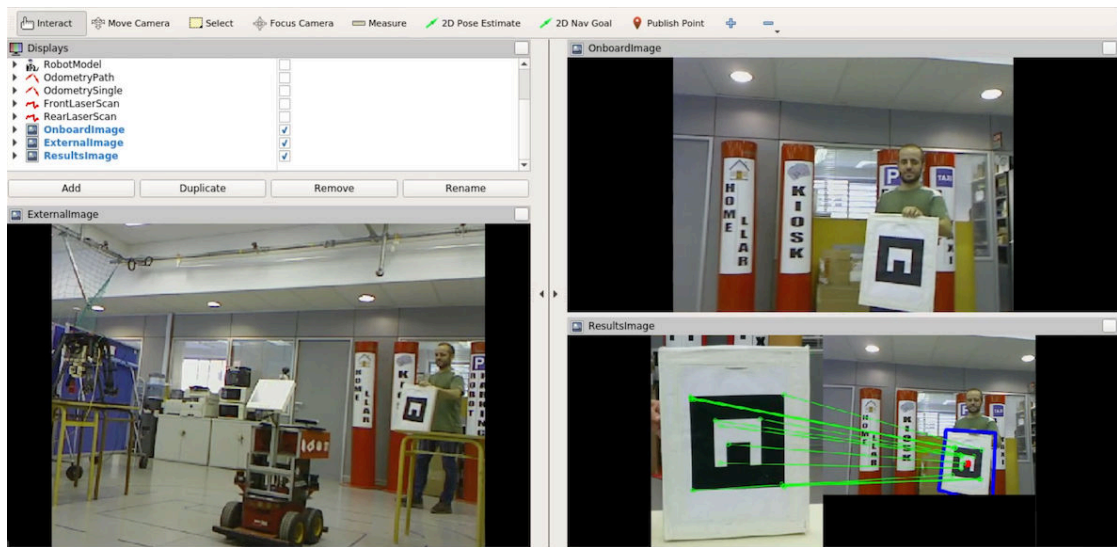


Figura 7.4: Segona prova de camp

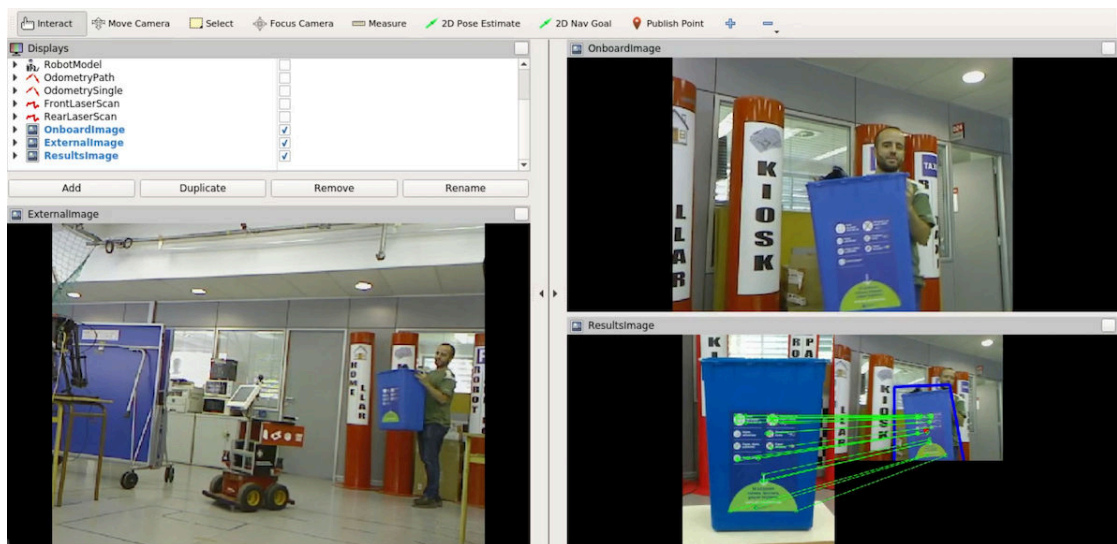


Figura 7.5: Tercera prova de camp

8. Lleis i regulacions

Pel que fa a lleis i regulacions, aquest projecte compleix la normativa vigent. Els dos àmbits principals que tracta són els drets d'imatge i les lleis de protecció de propietat intel·lectual.

8.1 Drets d'imatge

Totes les imatges de prova capturades durant el projecte han sigut preses en entorns del Campus Nord. Com que es tracta d'un projecte públic d'àmbit educatiu, la captura de tals imatges està permesa i no suposa un infringiment dels drets d'imatge.

Per altra banda, s'han restringit les imatges on puguin aparèixer terceres persones sense el seu consentiment i així acatar la *Llei Orgànica 1/1982* [21], que tracta sobre la protecció civil del dret al honor, a la intimitat personal i familiar i a la pròpia imatge.

8.2 Propietat intel·lectual

Quant al software i llibreries utilitzades per la realització del projecte, són *open-source* lliures de cap restricció d'ús. A excepció de l'algorisme de visió per computador SIFT, utilitzat en el sistema per la extracció de característiques. Es tracta d'un algorisme lligat a patent [22], propietat de David G. Lowe, que limita l'ús de l'algorisme a finalitats no-comercials. Aquesta restricció no afecta al projecte en qüestió, donat que té finalitats acadèmiques i d'investigació.

8.3 Llicències

Totes les llibreries de *Python* utilitzades en el projecte són *open-source* i, per tant, lliures d'ús.

Pel que fa a l'aplicació Android s'utilitzen les següents llibreries de tercers:

- **Camera Kit:** Utilitzat per integrar la càmera en una de les vistes. Disposa d'una llicència MIT, de manera que la llibreria pot ser integrada en el projecte sense cap restricció. [23]
- **Android Image Cropper:** Aquesta llibreria ofereix una interfície per poder editar i retallar imatges. La seva llicència és *Apache 2.0*. [24]

9. Canvis de planificació

Els transcurso del projecte pel que fa als primers blocs de tasques, s'han dut a terme tal com s'havia planificat inicialment. Tant els estudis previs, com el mòdul de GEP han finalitzat en el termini establert. A continuació es mostren els canvis realitzats en la planificació i el seu impacte en els recursos utilitzats.

9.1 Desviacions en la planificació

La primera tasca que es va estendre més temps de l'esperat va ser la configuració de l'entorn. El projecte requereix de l'ús de Python juntament amb la llibreria OpenCV. La correcta instal·lació d'aquesta va suposar un endarreriment aproximat d'una setmana.

La recerca d'informació i proves va ajudar a tenir una idea més clara dels requisits específics del projecte i la càrrega de treball que anava a suposar. Com que es tracta d'un projecte d'integració, és important disposar d'una primera versió del sistema funcionant. És per aquest motiu que es va prendre la decisió de prioritzar l'integració del sistema en el robot, ja que és la tasca que més impacte genera en el progrés general del projecte. D'aquesta forma, la tasca que anava a ser la segona fase d'integració va guanyar pes sobre les altres.

La integració del sistema en el robot tenia diverses implicacions. La primera i més important és la coordinació amb l'IRI, l'encarregada d'oferir els recursos materials i potencial beneficiari d'aquest projecte. Es va dur a terme una reunió, per visualitzar l'esquema general de la solució a la que es volia arribar i com es podia adaptar a les eines disponibles. Durant aquest període de comunicació es va aprofitar per progressar de forma paral·lela la integració del sistema en el dispositiu mòbil.

El projecte actualment es troba en un estat on l'implementació software del

robot està quasi finalitzada i a espera de poder fer proves reals en el robot ubicat a l'IRI. Pel que fa al dispositiu mòbil, l'aplicació té les funcionalitats bàsiques implementades.

A continuació es mostra el diagrama de Gantt original, dissenyat durant la planificació inicial del projecte:

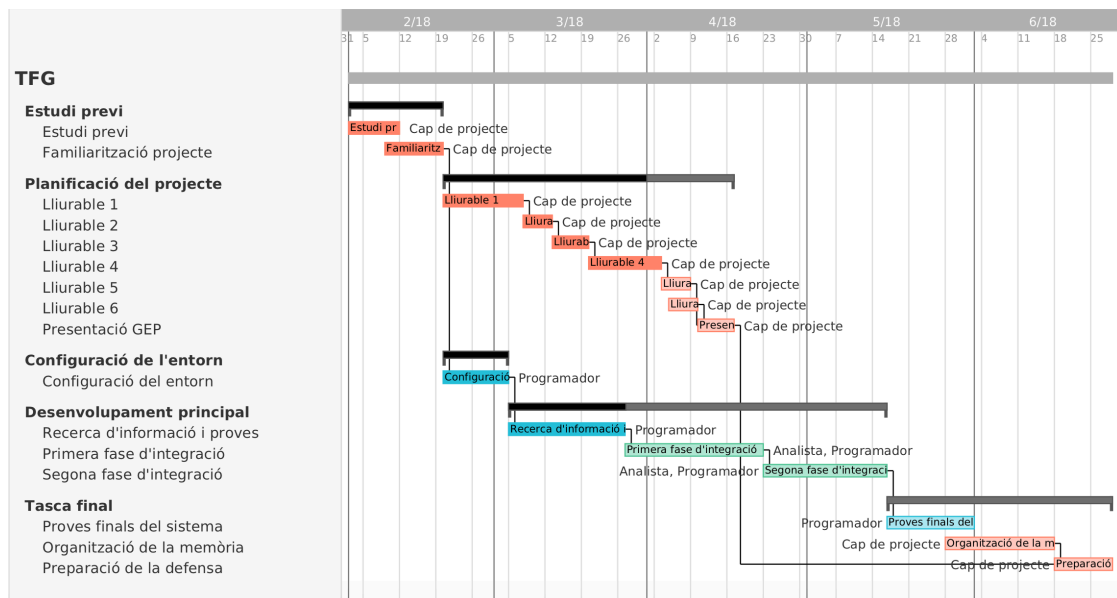


Figura 9.1: Gantt original - Desenvolupament temporal de tasques

El nou diagrama de Gantt amb els canvis de planificació mencionats amb anterioritat seria el següent:

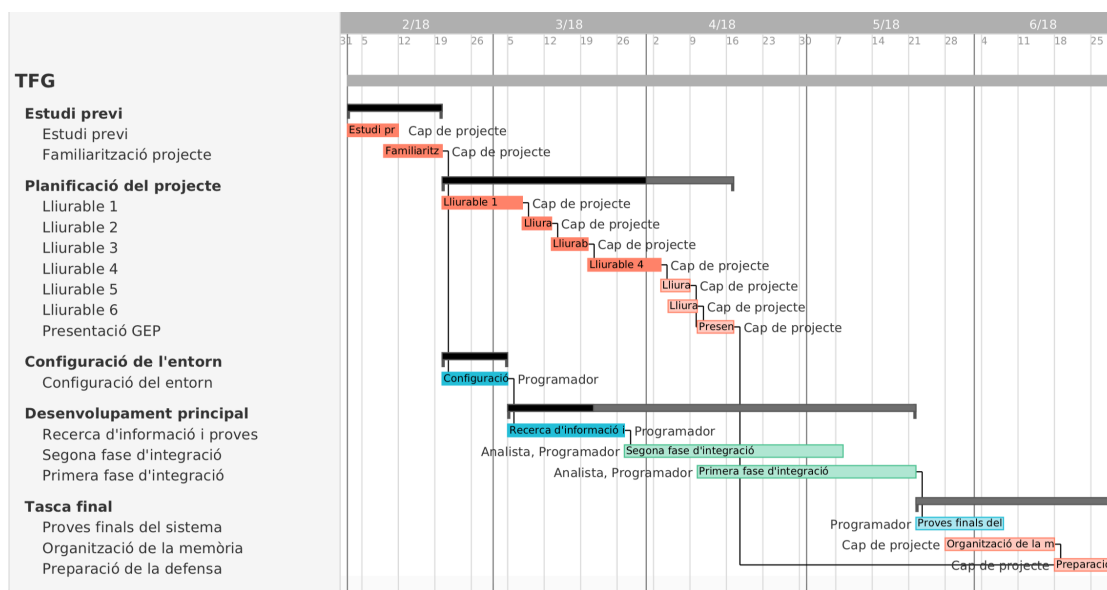


Figura 9.2: Gantt actualitzat - Desenvolupament temporal de tasques

9.2 Desviacions en els recursos utilitzats

Respecte a la planificació inicial, els canvis realitzats suposen només un canvi en l'ordre d'execució de les tasques. En cap cas implica un augment de les hores necessàries per finalitzar el projecte, de manera que el seu impacte en el cost de recursos humans es nul.

Per altra banda, en quant a recursos software addicionals, s'ha utilitzat la següent eina:

- **ROS:** es tracta d'un framework especialitzat per a facilitar el desenvolupament de sistemes complexos sobre sistemes de temps real com poden ser robots. Ofereix una abstracció respecte del hardware, cosa que fa el software portable i permet disposar d'una major robustesa.

Aquest framework disposa d'una llicència BSD, una llicència lliure que permet la reutilització de codi tant per finalitzats acadèmiques com comercials.

10. Informe de sostenibilitat

Des dels inicis de la industrialització, les empreses han buscat sempre dur a terme projectes que maximitzaven el benefici econòmic, sense considerar els perjudicis socials o ambientals que poguessin implicar. Això resultava en empreses buides de significat, amb l'únic objectiu de generar diners. Aquesta dinàmica ha provocat, durant molts anys, una poca integració de les empreses dins la societat i un greu deteriorament del medi ambient.

Però aquesta tendència que s'havia portat durant l'últim segle s'està invertint. Cada cop més, tant les empreses com les persones estan adquirint un major nivell de consciència social i ambiental, fins al punt de tenir el mateix pes que la vessant econòmica. El grau de consciència adquirit fa que durant la planificació d'un projecte s'estudii quin serà l'impacte que generarà. D'aquesta manera, es buscarà que un projecte aportí un valor afegit i benefici dins la societat, de forma que adquireix un propòsit més enllà de l'econòmic. Addicionalment, es té en molta més consideració la petjada ecològica que aquest pot generar, evitant així un major deteriorament del medi ambient.

Tot aquest procés es tracta d'una cadena, per tal que les empreses en un futur puguin dur a terme projectes sostenibles s'ha de conscienciar als estudiants sobre la importància d'aquests aspectes.

Avui dia existeixen competències transversals que tenen com a focus la sostenibilitat i on es treballa per aprofundir els coneixements de l'estudiant. Però encara queda un llarg camí a seguir i aspectes a millorar. Tot i informar als alumnes sobre quant d'important és que un projecte sigui sostenible en tots tres vessants (econòmic, social i ambiental), no hi ha un suficient nivell de conscienciació sobre la seva importància. Això fa baixar la implicació de l'estudiant en aquests temes.

Com a conclusió doncs, podem extreure que la sostenibilitat serà el gran repte al que s'hauran d'afrontar tots els projectes. La forma més òptima de millorar el compromís social i ambiental que es té és mitjançant una educació eficaç als estudiants. Hi ha molt marge de millora en la forma com es treballa avui dia,

ja que es tracta sempre com un tema separat en lloc d'integrar-lo dins del pla d'estudis.

10.1 Matriu de sostenibilitat

	PPP	Vida útil	Riscos
Ambiental	Consum de disseny	Emprempta ecològica	Ambientals
Econòmic	Factura	Pla de viabilitat	Econòmics
Social	Impacte personal	Impacte social	Socials

Taula 10.1: Matriu de sostenibilitat

Seguint la matriu de la taula 10.1, realitzarem l'anàlisi de la sostenibilitat del projecte des de tres punts de vista: l'econòmic, el social i l'ambiental. Per cada punt de vista, s'analitzarà l'impacte de la posada en producció del projecte, el cicle de vida que tindrà i els seus riscos inherents.

10.2 Dimensió econòmica

10.2.1 Identificació de costos

Distingirem quatre blocs principals a l'hora d'analitzar els costos del projecte per realitzar el pressupost.

El primer es tractarà dels costos directes, tots aquells costos en recursos utilitzats explícitament per la realització del projecte. Tindrem els recursos hardware i software utilitzats, a més de les despeses en recursos humans. Aquestes aniran directament relacionades amb les hores de treball especificades en el diagrama de Gantt. Cada tasca allà present suposarà la involucració de professionals amb rols diferents, cadascun amb el seu cost associat.

El segon es tractarà de costos indirectes, generats pels costos directes, com poden ser el consum energètic o l'accés a Internet. Seguidament es considerarà la

contingència, com a marge suplementari als costos directes i indirectes.

Finalment, tindrem en compte els costos addicionals que podrien sortir per imprevistos. El seu impacte en el total anirà lligat al risc que tenen aquests imprevistos a ocórrer.

10.2.2 Estimació de costos

En aquesta secció es realitza una estimació del pressupost necessari per tal de poder dur a terme aquest projecte. Es dividirà el pressupost en els quatre blocs principals descrits en l'apartat anterior, que combinats permetran obtenir el cost final. Es tracta d'una estimació inicial i, per tant, es podrà veure alterat durant el transcurs del projecte.

Costos directes - *Recursos hardware*

Per la realització del projecte seran necessaris els següents recursos hardware:

- **Ordinador portàtil.** S'utilitzarà per la realització de la major part de les tasques, tant de gestió, documentació, programació i proves.
- **Smartphone.** S'utilitzarà inicialment per capturar imatges de prova de l'entorn, i més endavant com a un dels components on s'integrarà el sistema per la seva utilització.
- **Robot.** Serà l'altre dispositiu on anirà integrat el sistema. El seu cost s'estimarà en funció dels preus disponibles en el mercat.

Per calcular l'amortització es farà ús de la següent fórmula:

$$A = \frac{\text{Hores en projecte} \times \text{Preu}}{\text{Vida total (hr)}}$$

Producte	Preu	Ús	Vida útil	Cost
Portàtil	1600 €	500h	6 anys	45,66 €
Smartphone	950 €	120h	4 anys	9,76 €
Robot	5224 €	10h	6 anys	0,99 €
Total				56,41 €

Taula 10.2: Recursos hardware

Costos directes - *Recursos software*

Pel que fa al software, l'utilitzat majoritàriament en aquest projecte és gratuït, de forma que no suposa cap impacte en el cost del projecte. L'única excepció és el programari Microsoft Office utilitzat en els esbossos de la documentació. En aquest cas, com que ja es disposava d'una llicència prèvia i aquesta és independent al nombre d'hores d'ús, s'ometrà també el seu impacte en el pressupost.

Costos directes - *Recursos humans*

El projecte el desenvoluparà una única persona que assumirà els diferents rols presents. Aquests seran el de cap de projecte, analista i programador. Seguint la taula de temps descrita en la secció 2.2, mostrem en la següent taula la distribució d'hores per a cada rol:

Tasca	Cap	Analista	Programador
Estudi previ	20h		
Planificació del projecte	100h		
Configuració de l'entorn	5h		15h
Recerca d'informació i proves			60h
Primera fase d'integració		20h	100h
Segona fase d'integració		30h	110h
Tasca final	80h		
Total	205h	50h	285h

Taula 10.3: Recursos humans

Rol	Hores	Preu / hora	Cost total
Cap de projecte	205h	50 /h	10250 €
Analista	50h	30 /h	1500 €
Programador	285h	20 /h	5700 €
Total			17450 €

Taula 10.4: Cost recursos humans

Pels costos dels diferents rols s'han agafat tarifes orientatives, competitives amb els preus del mercat.

Costos indirectes

Es consideraran com a costos indirectes el consum energètic d'electricitat i l'accés a Internet. Pel que fa al consum d'electricitat, es tindrà en compte el de l'ordinador, dispositiu mòbil i, més endavant, robot.

Tipus	Hores	Potència	Preu	Cost
Electricitat ordinador	500h	120W	0,11741 €/kWh	7,04 €
Electricitat smartphone	120h	5W	0,11741 €/kWh	0,07 €
Electricitat robot	N/A	N/A	0,11741 €/kWh	N/A
Accés Internet	450h		40 €/mes	25 €
Total				32,04 €

Taula 10.5: Costos indirectes

Per la tarifa de llum s'ha consultat els preus de la companyia Endesa [25], mentre que pel cost d'Internet s'ha fet una mitjana del preu de les diferents teleoperadores disponibles a Espanya [26]. Considerem negligible el consum energètic del robot a causa del nombre reduït d'hores d'ús que té en el projecte.

Imprevistos

Dins dels imprevistos inclourem els dos principals problemes observats en els obstacles del projecte. El primer serà la necessitat d'ampliació d'hores de treball, que es traduirà en un increment del cost pel que fa a recursos humans.

El segon imprevist es tractarà de l'insuficient capacitat de còmput en el robot durant l'execució dels algorismes i que no permeti garantir l'obtenció dels resultats dins d'una finestra de temps concreta. La solució per a aquest problema serà el de realitzar els càlculs en el *cloud* en un servidor extern, que augmentarà els costos hardware.

Incident	Solució	Risc	Impacte	Cost
Incompliment planificació prevista	Increment en 80h de treball de programador	10%	1600 €	160 €
Insuficient capacitat de còmput	Realització parcial dels càlculs en un servidor en el <i>cloud</i>	5%	20,83 € /mes	1,00 €/mes
Total				172,03 €

Taula 10.6: Incidències

Considerem, pel cas d'insuficient capacitat de còmput, l'ús de la plataforma *Google Cloud* [27] durant un període d'un any.

Contingència

Com a mesura de contingència, s'ha considerat suficient establir un marge del 5% sobre els costos directes i indirectes.

Costos totals

Tipus	Cost estimat
Costos directes	17.506,41 €
Costos indirectes	32,04 €
Imprevistos	172,03 €
Contingència (5%)	876,92 €
Total	18.587,40 €

Taula 10.7: Costos totals

10.2.3 Control de gestió

En la finalització de cada tasca es realitzarà una valoració del pressupost, i es sotmetrà a possibles canvis en el cas que el cost d'aquella fase no hagi estat l'esperat.

Per poder tenir un control acurat de les hores de treball involucrades en cadascun dels rols, es realitzarà un *daily report* amb un resum de les hores i tasques involucrades aquell dia. Això permetrà ajustar, si fos necessari, l'estimació d'hores en recursos humans a xifres més properes a la realitat.

Per calcular les desviacions que pugui haver es faran servir les següents fòrmules:

- **Desviació cost recursos no humans** = $(\text{Cost Estimat} - \text{Cost Real}) \times \text{Consum hores real}$
- **Desviació cost recursos humans** = $(\text{Cost Estimat} - \text{Cost Real}) \times \text{Consum hores reals}$
- **Desviació hores en recursos humans** = $(\text{Consum Estimat} - \text{Consum Real}) \times \text{Cost Real}$
- **Desviació hores en recursos humans** = Desviació preu + Desviació consum
- **Desviació total en recursos humans** = Cost estimat total - Cost real total

10.2.4 Viabilitat econòmica

Per poder considerar si el projecte és viable econòmicament s'ha realitzat una avaluació dels costos d'aquest, tant directes com indirectes. S'ha establert també un marge dins del pressupost pel que fa a possibles imprevistos i desviacions que puguin afectar al cost final. Tot això s'ha definit en els apartats anteriors del document, juntament amb el control de gestió que es realitzarà en cas que el pressupost es vegi modificat.

El cost d'aquest projecte és reduït i majoritàriament destinat en recursos humans. Això és a causa del reciclatge de la major part dels components. Per una part, tant l'ordinador portàtil com el smartphone tot i tenir un cost relativament elevat, són components adquirits prèviament a la realització del projecte i tenen una vida útil molt més enllà de la durada del projecte. Per altra banda, el robot, tot i ser el component més costós amb diferència, ha estat adquirit també prèviament i es farà servir de forma continuada per altres projectes.

Com que es tracta d'una petita part d'un projecte públic ja existent, els costos de la posada en producció seran nuls, perquè en cas satisfactori, el sistema es pot integrar directament amb els sistemes implementats en el robot.

A nivell econòmic, l'ús d'aquest projecte pot suposar una futura reducció de costos en diversos àmbits. L'addició d'un sistema d'autolocalització en robots com el desenvolupat en el projecte, complementat amb altres sistemes ja existents com el GPS, els proporciona una percepció més ampla i precisa de l'entorn. Això ofereix més facilitats a l'hora d'automatitzar certes tasques i reduir-ne el cost.

10.3 Dimensió ambiental

Tal com s'ha mencionat amb anterioritat, tots els components *hardware* utilitzats en aquest projecte han estat adquirits amb anterioritat i tenen una vida útil tant prèvia com posterior al projecte. Això fa que l'impacte ambiental directe que pugui tenir sigui pràcticament nul. Durant el desenvolupament no es generarà cap tipus de residu, l'únic recurs consumit serà l'energia elèctrica.

Les bateries integrades en el robot són de plom-àcid, molt comunes en motors d'arrencades o per traccionar vehicles elèctrics. Durant el reciclatge d'aquest tipus de bateria, un cop neutralitzat l'àcid sulfúric que tenen en l'interior, la major part del metall és reutilitzable per fabricar nous components. De fet, el plom és un dels metalls més fàcilment reciclables [28]. S'estima que a Occident l'índex de

recuperació d'aquestes bateries és del voltant del 90% i, a Espanya, els valors estan al voltant del 95-97%; pel que molt poca part del plom queda sense ser reciclada o reutilitzada [29].

Un cop finalitzat aquest projecte, tot el programari serà lliure i a la disposició d'altres usuaris per poder reutilitzar-lo en futurs projectes. L'ús o finalitat que li puguin donar tercers i l'impacte ambiental que aquest pugui tenir queden fora de l'abast del projecte.

Pot suposar una millora a nivell ambiental, en comparació a altres sistemes de navegació i localització existents. Aquest podria ser el cas del GPS, on són necessaris diversos satèl·lits i el sistema a bord del robot d'un alt consum energètic. Un sistema de localització alternatiu, basat en visió per computador, ofereix en certs escenaris una alternativa més efectiva i eficient energèticament.

10.4 Dimensió social

A nivell personal proporciona la perspectiva de com realitzar la correcta gestió general d'un projecte. Des de la formulació del problema, a la seva planificació i execució. També permet estudiar altres aspectes com les diferents metodologies a seguir, la realització d'una gestió econòmica efectiva i viable, o dur a terme un anàlisi de l'impacte que tindrà el projecte en les diferents dimensions.

Addicionalment suposa una ampliació dels coneixements en visió per computador i robòtica. L'experiència adquirida a nivell tècnic i acadèmic en aquest projecte servirà per poder desenvolupar amb èxit futurs projectes de característiques similars.

Aquest projecte pertany al sector anomenat robòtica social. Els robots socials són aquells que interactuen i es comuniquen amb altres agents de l'entorn, ja siguin humans o altres robots. Aquests segueixen un comportament, patrons i normes socials. La seva finalitat és contribuir i aportar un benefici a la societat, ja sigui en temes de salut, educació o assistència a persones [30]. Altres projectes de robòtica social són el reconeixement d'expressions facials o el tractament de pacients amb autisme [31].

La integració d'un sistema d'autolocalització mitjançant visió per computador, fa que aquests robots disposin d'una major percepció de l'entorn. L'impacte i beneficis que tal sistema pot generar venen donats de forma indirecta, pel seu potencial ús en projectes com els mencionats anteriorment, que aporten un gran valor a la societat.

Per altra banda, l'impacte social que pot tenir aquest projecte, ve donat de forma indirecta pels proveïdors utilitzats durant el seu desenvolupament. En aquest cas, el telèfon mòbil ha estat adquirit de l'empresa Samsung, mentre que l'ordinador portàtil per Apple. En un estudi realitzat sobre la sostenibilitat de les empreses del sector tecnològic [32], totes dues companyies suspenen en l'apartat de drets humans. Això és a causa de les condicions laborals aplicades en països on subcontracten la seva producció pels seus reduïts costos. Durant la compra de l'equipament, no es van tenir en consideració les conseqüències socials que implicava escollir un proveïdor o un altre de diferent. Durant la realització d'aquest treball, s'ha prèsc consciència de les conseqüències que poden tenir les decisions d'un enginyer en la sostenibilitat del seu projecte.

11. Conclusions i treball futur

Amb aquest apartat es conclourà la memòria tècnica del projecte. Primerament, s'extreuran les conclusions tècniques i personals obtingudes. Seguidament es detallarà el treball futur i possibles vies d'expansió del projecte.

11.1 Conclusions

L'objectiu principal del projecte era el d'integrar un sistema d'autolocalització en robots mòbils. Podem afirmar que aquest s'ha dut a terme tal i com s'esperava. El resultat final ha estat un robot capaç de detectar en el seu entorn objectes definits per l'usuari i indicar-ne la posició.

Les proves finals, que s'han realitzat en un entorn tancat, han donat bons resultats amb una taxa d'encert per sobre de l'esperada. El sistema ha respòs bé a les variacions d'escala i al moviment.

No obstant, s'ha observat que el sistema té certes limitacions. La principal ve donada pels keypoints detectats. En casos on l'objecte no es distingeix suficientment del seu entorn, pot passar que el sistema no detecti cap keypoint en la regió on es troba l'objecte buscat. Això porta a tenir falsos negatius. El mateix pot passar en situacions on hi hagi contrallum o reflexions.

Així doncs, podem dir que els resultats obtinguts han estat satisfactoris per tractar-se d'un primer model, però amb marge de millora per futures implementacions.

Per altra banda, aquest projecte ha sigut beneficiós tant a nivell personal com professional. Tot el procés de planificació inicial, gestió, control i organització ha estat d'ajut per poder tenir una clara perspectiva del desenvolupament d'un projecte, des de l'inici fins al seu tancament. El fet de ser una única persona,

assumint diferents rols en el projecte, ha exigint l'involucració en àrees diverses.

El desenvolupament del sistema de localització ha suposat un repte a nivell tècnic. S'han treballat conceptes i tecnologies noves, pel que ha sigut necessari un ràpid procés d'aprenentatge. Els coneixements previs adquirits durant el grau han permès una àgil adaptació a aquestes noves eines de treball que, combinades amb conceptes ja integrats per l'autor, han permès una correcta i exitosa progressió del projecte.

11.2 Treball futur

Aquest projecte és una primera versió d'integració del sistema d'autolocalització. Es tracta d'un primer pas, que pot donar lloc a una sèrie de nous projectes que presentin millores i noves funcionalitats addicionals.

- **Visió stèreo càmera.** Una de les ampliacions proposades a l'inici del projecte era l'utilització d'una càmera stèreo en el sistema. Aquesta ens proporciona una dada més de les imatges del robot: la profunditat. Complementada amb les coordenades d'imatge que calcula actualment el sistema, ens permetria determinar la distància a la que es troba l'objecte detectat.

Es va realitzar un prova pilot amb una càmera dual, però el sobrecost de processar el doble de fotogrames feia augmentar en excés els temps d'execució.

- **Actualitzar imatge de referència.** Un dels inconvenients que presenta el sistema actualment són els canvis d'escala. A mesura que el robot s'apropa a l'objecte destí, els keypoints que es troben van canviant a causa de la magnificació. En conseqüència, a certa proximitat el sistema perd l'objecte que seguia.

Una bona estratègia a seguir és la d'actualitzar la imatge de referència del robot amb noves a mesura que s'apropa al seu objectiu.

- **Control de moviment.** Per donar mobilitat al robot durant les proves, s'ha implementat un senzill controlador que converteix les coordenades d'imatge en velocitats angulars per les dues rodes. El resultat és que el robot es dirigeix cap a la direcció on detecta l'objecte.

Aquest sistema de control es pot desenvolupar perquè sigui més complex. No s'han considerat aspectes com la distància o els obstacles.

- **Aplicació mòbil per iOS.** L'aplicació mòbil s'ha desenvolupat únicament per la plataforma Android. Un possible treball futur seria el de crear la versió per iOS.

Glossari

bottleneck Fenomen produït quan la capacitat o rendiment d'un sistema es veuen limitats per un dels seus components. 7

cloud En referència a cloud computing, paradigma que permet oferir serveis computacionals a través de la xarxa. 7

deadline Data límit per la finalització d'una determinada tasca. 5, 8

developer Persona al càrrec del desenvolupament d'un projecte. 5

features Atributs o aspectes distintius d'una entitat. 8

framework Estructura conceptual i tecnològica de suport definit formada per mòduls software específics. 39, 65

handler Funció asíncrona que s'executa quan succeeix un esdeveniment concret del que s'esperen dades com a resultat. 50

homografia En geometria, transformació projectiva que determina la correspondència de punts i rectes entre dues figures geomètriques. 6, 81

keypoint En visió per computador, punt distintiu d'una imatge que ens permet caracteritzar-la. 2, 8

match Punts de diferents imatges que són coincidents. 28

matching Aplicació de l'homografia en dues figures. 8

open-source Enfocament per al disseny, desenvolupament i distribució que ofereix un accés pràctic al codi font del producte. 11, 62

outlier Valor atípic d'una mostra que és numèricament distant a la resta de dades. 38

ping Utilitat diagnòstica per comprovar l'estat de connexió amb altres dispositius remots connectats a la xarxa. 53

script Petita porció de codi amb una finalitat específica. 33

socket Connexió en xarxa entre dos dispositius, mitjançant el protocol TCP, pel que aquests intercanvien un fluxe de dades de manera fiable i ordenada. 39, 47, 48

string Seqüència de caràcters. 50

stèreo Estereoscopi. Captura de dues imatges del mateix objecte des de punts de vista lleugerament diferents, un al costat del altre. 16

timeline Representació gràfica d'un període de temps. 33

Acrònims

BRISK Binary Robust Invariant Scalable Keypoints. 19, 23

ESAI Enginyeria de Sistemes, Automàtica i Informàtica Industrial. 5

FAST Features from Accelerated Segment Test. 18, 19

FREAK Fast Retina Keypoint. 23

GEP Gestió de Projectes. 12

GPS Global Positioning System. 1

IoT Internet of Things. 3

IP Internet Protocol. 48, 50

IRI Institut de Robòtica i Informàtica Industrial. 4–6, 33, 57, 63, 64

LATCH Learned Arrangements of Three Patch Codes. 18, 23

ROI Region of Interest. 28

ROS Robot Operating System. 39, 40, 49, 50, 59

SIFT Scale-Invariant Feature Transform. 17, 21–23, 31, 33, 61

SURF Speeded-up Robust Features. 18, 22, 31

TCP Transmission Control Protocol. 39, 48

Bibliografia

- [1] David A Forsyth i Jean Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.
- [2] Cyberbotics Ltd. *Using the Pioneer 3-AT and Pioneer 3-DX robots*. 2018. URL: <https://www.cyberbotics.com/doc/guide/using-the-pioneer-3-at-and-pioneer-3-dx-robots?version=8.5> (cons. 30-09-2010).
- [3] Compulab. *Specifications (all models)*. 2017. URL: <http://www.fit-pc.com/web/products/specifications/?model\%5B\%5D=IPC-C3517V-WB-FM4U> (cons. 30-09-2010).
- [4] Richard Szeliski. *Computer Vision: Algorithms and Applications*. 2010.
- [5] Simon Achatz. “State of the Art of Object Recognition Techniques”. A: *Scientific Seminar, Neuroscientific System Theory* (2016).
- [6] David G Lowe. “Distinctive image features from scale-invariant keypoints”. A: *International journal of computer vision* 60.2 (2004), pàg. 91-110.
- [7] Herbert Bay et al. “Speeded-up robust features (SURF)”. A: *Computer vision and image understanding* 110.3 (2008), pàg. 346-359.
- [8] Tomasz Trzcinski et al. *Boosting binary image descriptors*. Inf. tèc. Technical report, EPFL, 2012.
- [9] Christopher Parker et al. “The CUDA LATCH binary descriptor: because sometimes faster means better”. A: *European Conference on Computer Vision*. Springer. 2016, pàg. 685-697.
- [10] Edward Rosten i Tom Drummond. “Machine learning for high-speed corner detection”. A: *European conference on computer vision*. Springer. 2006, pàg. 430-443.
- [11] Stefan Leutenegger, Margarita Chli i Roland Y Siegwart. “BRISK: Binary robust invariant scalable keypoints”. A: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pàg. 2548-2555.
- [12] Saad J Bedros. *Lecture 8: Interest Point Detection*. University of Minnesota. 2015.

-
- [13] Jianbo Shi et al. “Good features to track”. A: *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR’94., 1994 IEEE Computer Society Conference on*. IEEE. 1994, pàg. 593-600.
- [14] Akash Patel et al. “Performance analysis of various feature detector and descriptor for real-time video based face tracking”. A: *International Journal of Computer Applications* 93.1 (2014).
- [15] OpenCV. *Introduction to SIFT (Scale-Invariant Feature Transform)*. URL: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html.
- [16] *A tutorial on binary descriptors – part 5 – The FREAK descriptor*. Abr. de 2016. URL: <https://gilscvblog.com/2013/12/09/a-tutorial-on-binary-descriptors-part-5-the-freak-descriptor/>.
- [17] Engin Tola, Vincent Lepetit i Pascal Fua. “Daisy: An efficient dense descriptor applied to wide-baseline stereo”. A: *IEEE transactions on pattern analysis and machine intelligence* 32.5 (2010), pàg. 815-830.
- [18] *Learned Arrangements of Three Patch Codes*. Maig de 2016. URL: <https://www.youtube.com/watch?v=fvyd9aUJ09g>.
- [19] *ROS Wiki*. Open Source Robotics Foundation. 2018. URL: <http://wiki.ros.org/> (cons. 10-04-2018).
- [20] Android Developers. *Android 5.1 APIs*. 2018. URL: <https://developer.android.com/about/versions/android-5.1> (cons. 24-04-2018).
- [21] Ley Orgánica. “1/1982, de 5 de mayo, de Protección Civil del Derecho al Honor, a la Intimidación Personal y Familiar ya la Propia Imagen”. A: *Boletín Oficial del Estado* 115 (1982), pàg. 12546-12548.
- [22] David G Lowe. *Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image*. US Patent 6,711,293. Març de 2004.
- [23] *CameraKit/camerakit-android*. GitHub. Gen. de 2018. URL: <https://github.com/CameraKit/camerakit-android> (cons. 13-04-2018).
- [24] *ArthurHub/Android-Image-Cropper*. GitHub. Abr. de 2018. URL: <https://github.com/ArthurHub/Android-Image-Cropper> (cons. 13-04-2018).
- [25] *Tarifas Gas y Luz*. 2018. URL: <https://tarifasgasluz.com/faq/precio-kwh> (cons. 18-03-2018).
- [26] *ADSL Zone*. 2018. URL: <https://www.adslzone.net/2016/04/29/precio-internet-espana-los-10-paises-mas-caros-europa/> (cons. 18-03-2018).
- [27] Google. *Precios de Google Cloud Platform*. 2018. URL: <https://cloud.google.com/pricing/?hl=es> (cons. 30-03-2018).

- [28] María Gómez Gómez. *Posibilidades en el tratamiento de residuos de pilas y baterías*. Interempresas. Des. de 2011. URL: <http://www.interempresas.net/Reciclaje/Articulos/58972-Posibilidades-en-el-tratamiento-de-residuos-de-pilas-y-baterias.html> (cons. 13-06-2018).
- [29] Manuel Bañeres Sorinas. “Estudio de alternativas en el reciclaje de baterías de plomo fuera de uso”. A: *Universitat Politècnica de Catalunya* (2003).
- [30] Joost Broekens, Marcel Heerink, Henk Rosendal et al. “Assistive social robots in elderly care: a review”. A: *Gerontechnology* 8.2 (2009), pàg. 94-103.
- [31] Arvin Agah et al. “Sociorobotics”. A: *International Journal of Social Robotics* 10.2 (abr. de 2018), pàg. 177-178. ISSN: 1875-4805. DOI: 10.1007/s12369-018-0470-y. URL: <https://doi.org/10.1007/s12369-018-0470-y>.
- [32] *Ethical Comparison - Mobile Phones*. The Good Shopping Guide. URL: <http://www.thegoodshoppingguide.com/ethical-mobile-phones/>.