

Sistema d'autolocalització per a robots mòbils mitjançant tècniques de visió per computador

Treball final de grau en eng. informàtica

Tecnologies de la informació

Joan Rodas Cusidó

22 d'abril de 2017

Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya
Director: Joan Climent (ESAI)



1. Planificació
2. Recursos i costos
3. Lleis i sostenibilitat
4. Arquitectura del sistema
5. Tècniques de visió usades
6. Resultats
7. Conclusions

Objectiu

Dissenyar i desenvolupar un sistema d'autolocalització per a robots mòbils usant algorismes de visió.

1. Obtenció de *keypoints*
2. Extracció de característiques
3. *Matching* de dues imatges
4. Homografia

Planificació

Tasques (blocs)

Descripció	Metodologia	Hores
Preparació de l'entorn	-	5h
Curs de GEP	Cascada	75h
Desenvolupament del projecte	Àgil	355h
Preparació de la defensa	-	45h

Taula 1: *Blocs del projecte*

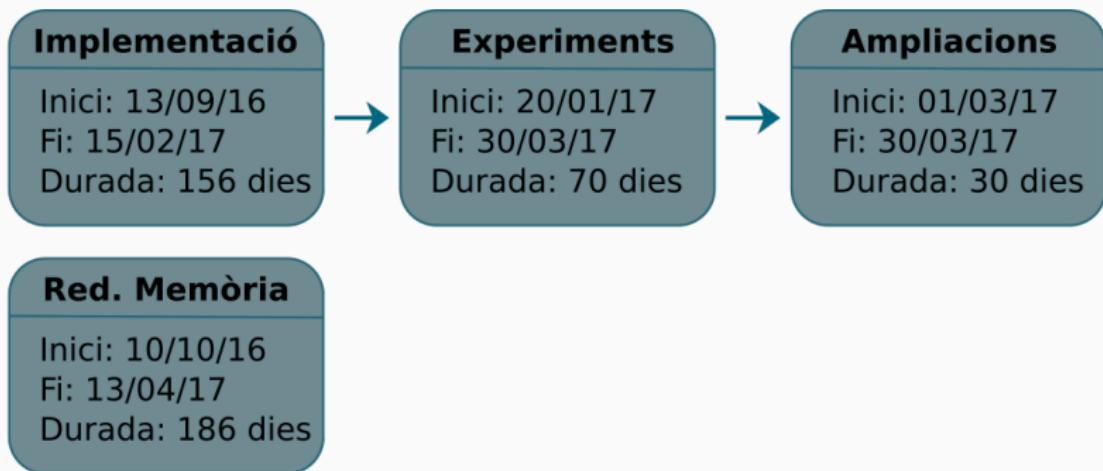


Figura 1: *Tasques desenvolupament*

Diagrama de Gantt

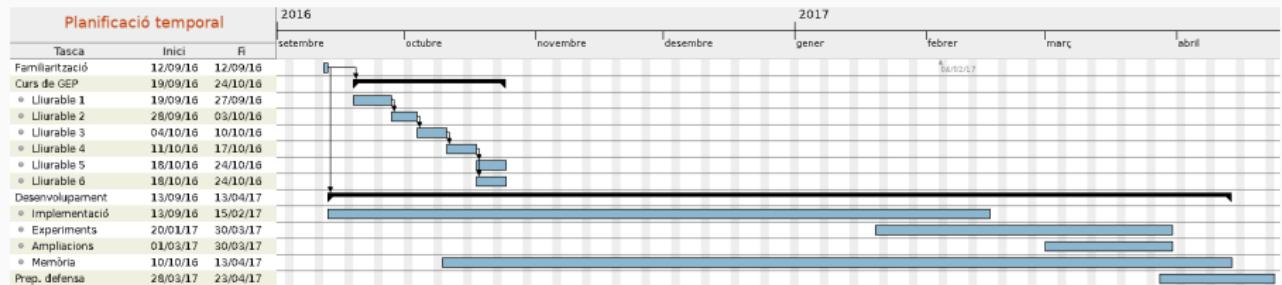


Figura 2: *Gantt del projecte*

Recursos i costos

Producte	Preu	Ús	Vida útil	Amortització
Ordinador	500€	7 mesos	5 anys	58,33€
Smartphone	39€	1 mes	3 anys	1,08€
Total				59,41€

Taula 2: Recursos de maquinari

Nom	Tipus	Ús
Arch Linux/Raspbian	Eina de desenvolupament	Execució del programari
Python + OpenCV	Eina de desenvolupament	Programació
Flask	Eina de desenvolupament	Micro-framework
uWSGI	Eina de desenvolupament	Servidor uwsgi
Nginx	Eina de desenvolupament	Servidor web/proxy
Geany/Atom	Eina de desenvolupament	Programació del codi
L <small>A</small> T <small>E</small> X	Documentació	Redacció de la memòria
Zathura	Documentació	Visualització de pdf
Gantt Project	Eina de gestió	Creació diagrames de Gantt
Git + GitHub	Desenvolupament i gestió	Control de versions

Taula 3: *Recursos de programari*

Recursos humans I

Tasca	Cap de projecte	Analista	Programador
Preparació de l'entorn	3h		2h
Curs de GEP	75h		
Implementació i proves		30h	195h
Experiments			40h
Ampliacions		10h	30h
Redacció memòria	50h		
Preparació defensa	45h		
Total	173h	40h	267h

Taula 4: Recursos humans (hores)

Rol	Hores	Cost/hora	Cost total
Cap de projecte	173h	25€/h	4325€
Analista	40h	20€/h	800€
Programador	267h	15€/h	4005€
Total			9130€

Taula 5: *Recursos humans (costos)*

Costos indirectes

Tipus	Temps	Cost	Cost total
Electricitat*	480h	0,028€/h	13,44€
Accès a Internet	480h	0,17€/h	81,6€
Total			95,04€

Taula 6: *Costos indirectes*

* Cost de l'electricitat = 0,141033€/kWh (considerem la potència 0,2kW)

Costos totals

Tipus	Cost estimat
Recursos humans	9.130€
Recursos de programari	0€
Recursos de maquinari	59,41€
Costos indirectes	95,04€
Imprevistos	600€
Contingència (5%)	494,22€
Total	10.378,67€

Taula 7: *Costos totals*

Lleis i sostenibilitat

Lleis i regulacions

Caldrà tenir en compte les lleis i regulacions, tant a l'hora de realitzar el projecte com a l'hora de publicar-lo o fer la documentació.

- Drets d'imatge
- Patents dels algorismes
- Drets d'autor

Sostenibilitat	PPP	Vida útil	Riscos
Ambiental	Consum del disseny 8 [0:10]	Petjada ecològica 15 [0:20]	Riscos ambientals 0 [-20:0]
Econòmica	Factura 7 [0:10]	Pla de viabilitat 10 [0:20]	Riscos econòmics 0 [-20:0]
Social	Impacte personal 8 [0:10]	Impacte social 5 [0:20]	Riscos socials 0 [-20:0]
Valoració total	53 [-60:90]		

Taula 8: Matriu de sostenibilitat

Arquitectura del sistema



Figura 3: Arquitectura del sistema

Font: Madebyoliver i Freepik

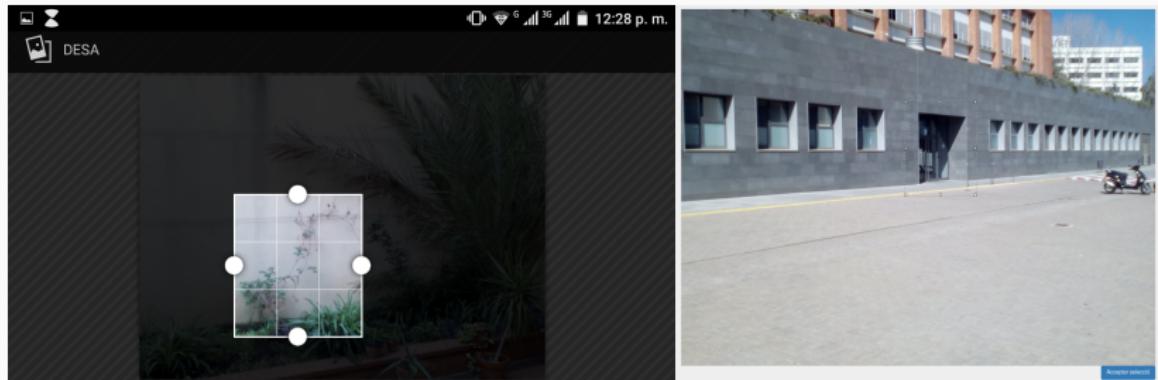


Figura 4: Selecció de la regió d'interès

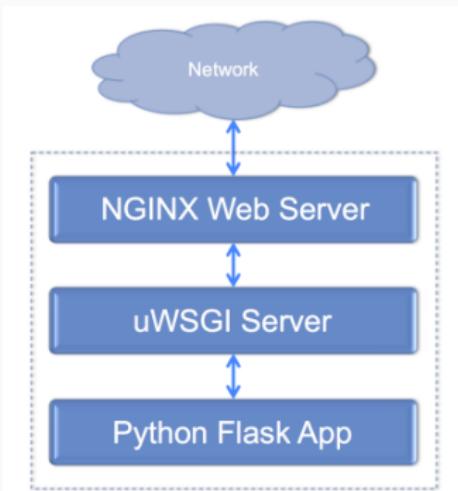


Figura 5: *Estructura del servidor*

Font: <https://iotbytes.wordpress.com>

1. Instal·lar sistema operatiu
2. Instal·lar Python + OpenCV
3. Instal·lar Flask
4. Instal·lar uWSGI i Nginx
5. Configuració bàsica

Tècniques de visió usades

- Preprocessat digital d'imatges
- Detecció de punts d'interès
- Extracció de característiques
- *Matching* de característiques
- Homografia

Què és?

Consisteix a obtenir punts de la imatge amb característiques distintives, que ens puguin ser útils més endavant.

Algorismes principals utilitzats:

- Harris[1]
- SIFT[2]
- ORB[3]

- Detector de cantonades
- Finestra NxM píxels
- Busca canvis d'intensitat
- No és invariant a l'escala

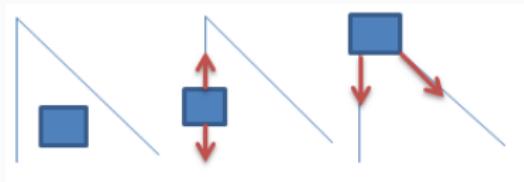


Figura 6: Flat, vora i cantonada

Font: Wikipedia

En el nostre cas, s'ha optat per aplicar Harris en diverses escales, fent una piràmide de la imatge original. A cada nivell, es redueix la imatge a la meitat.

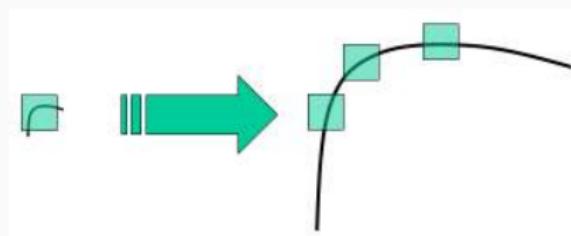


Figura 7: Vores a diferent escala

Font: OpenCV

1. Diferència de Gaussianes en diferents escales
2. Màxims i mínims locals en l'espai i l'escala
3. Es repeteixen els passos fent una piràmide

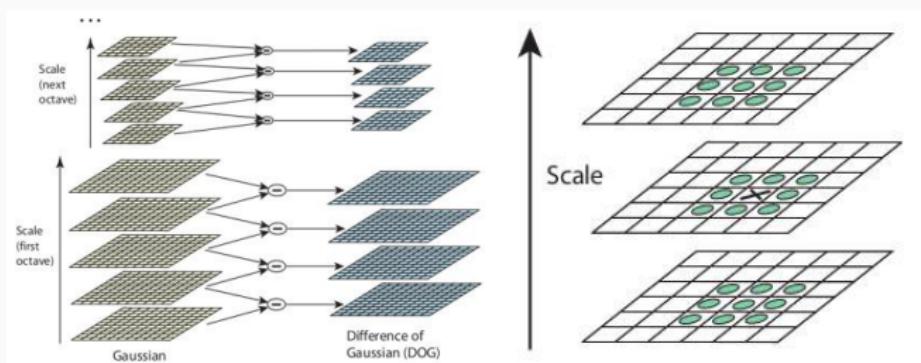


Figura 8: SIFT - DoG i local extrema

Font: OpenCV

4. S'eliminen punts amb intensitat menor a un cert llindar
5. S'eliminen vores
6. S'assigna una orientació als punts

- Detector FAST[4] amb modificacions
- Detecció de cantonades, molt ràpid
- Es compara la intensitat d'un píxel amb N veïns

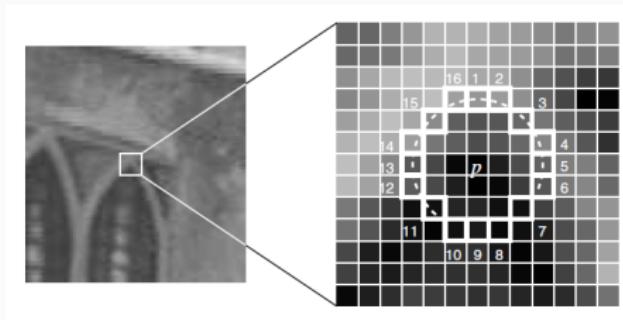


Figura 9: FAST, $N=16$

Font: <https://www.edwardrosten.com/work/fast.html>

FAST no és invariant a l'escala ni la rotació.

ORB aplica les següents millores:

- S'agafen els N millors punts després d'aplicar la mesura de Harris.
- Es fa una piràmide per fer multi-escala.
- S'utilitzen els moments per calcular l'orientació.

Què és?

Consisteix en descriure característiques de les imatges en els punts donats. Podrem comparar aquestes característiques amb les d'una altre imatge i veure si són similars.

Algorismes principals utilitzats:

- SIFT
- ORB
- BRISK[5]

SIFT - Descriptor

- Veïnatge de 16×16 píxels al voltant del punt
- Divisió en 16 blocs de 4×4
- Per cada sub-bloc es calcula histograma d'orientacions en 8 direccions
- Vector de dimensió 128

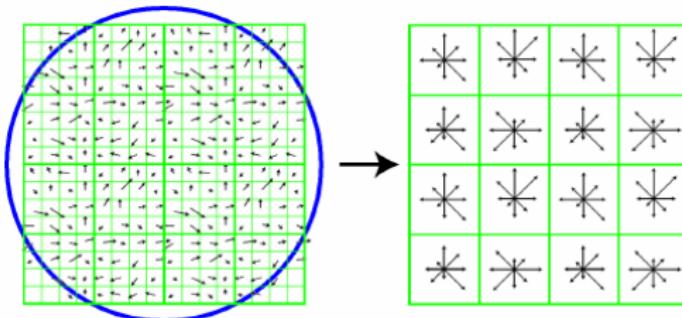


Figura 10: Descriptor SIFT

Font: <https://www.researchgate.net>

- Descriptor binari, senzill i ràpid
- Modificació BRIEF[6]
- N píxels veïns, s'agafen parells
- Per cada parell es compara la intensitat i es retorna 1 o 0 segons si és major la del primer o la del segon
- BRIEF no és invariable en la rotació → ORB “gira” el patró en funció de l'angle

- Descriptor binari
- Patró de cercles concèntrics
- Comparació cadenes binàries amb XOR

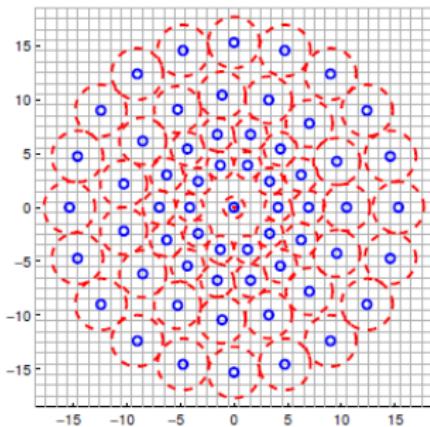


Figura 11: Patró BRISK

Font: <https://gilscvblog.com>

Matching

Què és?

Consisteix en trobar coincidències entre els punts de dues imatges, comparant les seves característiques.

Pels descriptors binaris utilitzem la distància de Hamming i pels vectorials l'euclidiana.



Figura 12: Matching

Què és?

Trobant la relació entre els píxels de les dues imatges podrem reprojectar el pla d'una imatge en l'altre i trobar el punt on volem dirigir el robot.

Aplicarem RANSAC (*Random Sample Consensus*)[7], un algorisme que ens permetrà eliminar *outliers*.

Resultats

Imatges similars I



Figura 13: Imatges motos i cotxes

Algorismes	Motos		Cotxes	
	Correctes	Erronis	Correctes	Erronis
Harris + SIFT	110	1	98	4

Taula 9: Matching - imatges similars



Figura 14: Imatges campus i jardins palau reial

Algorismes	Campus (moto)		Jardins (font)	
	Correctes	Erronis	Correctes	Erronis
Harris + SIFT	30	0	38	0
SIFT + SIFT	15	1	49	2
ORB + ORB	14	1	52	10
ORB + BRISK	19	0	71	6

Taula 10: *Matching - objectes*

Imatges diferents I



Figura 15: Imatges campus i jardins palau reial

Algorismes	Campus 1 (subimatge)				Campus 2 (subimatge)			
	Kp1	Kp2	Parells	t	Kp1	Kp2	Parells	t
Harris + SIFT	67	757	0	0.331s	304	417	2	0.839s
SIFT + SIFT	125	5148	1	0.987s	570	1342	5	1.113s
ORB + ORB	830	2500	2	0.840s	2405	2500	7	0.113s
ORB + BRISK	830	2500	1	0.942s	2405	2500	7	1.003s

Taula 11: Matching - imatges diferents

Resultats matching i homografia



Figura 16: Homografia - Resultats

Conclusions

Conclusions

L'objectiu principal s'ha complert.

- Harris + SIFT més robust
- ORB alternativa ràpida
- Marge de millora

- Comparació i anàlisi d'algorismes
- Diferents imatges
- Preprocessat
- Aplicació mòbil
- Entorn real + robot

Gràcies per la vostra
atenció

Referències I

-  Chris Harris i Mike Stephens. "A combined corner and edge detector".
A: *In Proc. of Fourth Alvey Vision Conference*. 1988, pàg. 147 - 151. URL:
www.bmva.org/bmvc/1988/avc-88-023.pdf.
-  David G. Lowe. "Object recognition from local scale-invariant features".
A: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Vol. 2. 1999, pàg. 1150 - 1157. DOI:
[10.1109/ICCV.1999.790410](https://doi.org/10.1109/ICCV.1999.790410). URL:
<http://www.cs.ubc.ca/~lowe/papers/iccv99.pdf>.
-  Ethan Rublee et al. "ORB: An Efficient Alternative to SIFT or SURF".
A: *Proceedings of the 2011 International Conference on Computer Vision. ICCV '11*. Washington, DC, USA: IEEE Computer Society, 2011,
pàg. 2564 - 2571. ISBN: 978-1-4577-1101-5. DOI:
[10.1109/ICCV.2011.6126544](https://doi.org/10.1109/ICCV.2011.6126544). URL:
<http://dx.doi.org/10.1109/ICCV.2011.6126544>.

Referències II

- 
- Edward Rosten i Tom Drummond. "Machine Learning for High-speed Corner Detection". A: *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*. ECCV'06. Graz, Austria: Springer-Verlag, 2006, pàg. 430 - 443. ISBN: 3-540-33832-2, 978-3-540-33832-1. DOI: 10.1007/11744023_34. URL: http://dx.doi.org/10.1007/11744023_34.
- 
- Stefan Leutenegger, Margarita Chli i Roland Y. Siegwart. "BRISK: Binary Robust Invariant Scalable Keypoints". A: *Proceedings of the 2011 International Conference on Computer Vision*. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pàg. 2548 - 2555. ISBN: 978-1-4577-1101-5. DOI: 10.1109/ICCV.2011.6126542. URL: <http://dx.doi.org/10.1109/ICCV.2011.6126542>.

Referències III

-  Michael Calonder et al. "BRIEF: Binary Robust Independent Elementary Features". A: *Proceedings of the 11th European Conference on Computer Vision: Part IV*. ECCV'10. Heraklion, Crete, Greece: Springer-Verlag, 2010, pàg. 778 - 792. ISBN: 3-642-15560-X, 978-3-642-15560-4. URL: <http://dl.acm.org/citation.cfm?id=1888089.1888148>.
-  Martin A. Fischler i Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". A: *Commun. ACM* 24.6 (juny de 1981), pàg. 381 - 395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: <http://doi.acm.org/10.1145/358669.358692>.