



EPS

Escola Politècnica
Superior

Projecte/Treball Fi de Carrera

Estudi: Eng. Tècn. Informàtica de Sistemes. Pla 2001

Títol: Disseny d'un comportament en ROS per a mantenir la posició d'un vehicle autònom submarí a partir de referències visuals

Document: Resum

Alumne: ÓSCAR SIMÓN CASTILLO

Director/Tutor: MARC CARRERAS PÉREZ / NARCÍS PALOMERAS ROVIRA

Departament: Arquitectura i Tecnologia de Computadors

Àrea: A.T.C

Convocatòria (mes/any): setembre / 2011

1. Introducció

En el Centre d'Investigació en Robòtica Submarina (CIRS) de la Universitat de Girona es disposa de diferents robots submarins els quals utilitzen una arquitectura software anomenada Component Oriented Layered-based Architecture for Autonomy (COLA2), la qual ha estat desenvolupada per estudiants i professors del mateix centre. Per tal de fer aquesta arquitectura més accessible per a professors i estudiant d'altres centres la COLA2 s'està adaptant al Robot Operative System (ROS) que és un framework genèric per al desenvolupament d'aplicacions amb robots.

Aquest projecte pretén dissenyar un comportament per al robot Girona500 que estigui desenvolupat dins la versió ROS de l'arquitectura COLA2. El comportament haurà de fer mantenir una determinada posició al robot amb informació visual de la càmera del robot i amb dades de navegació. La tasca de mantenir la posició es de vital importància per a poder realitzar intervencions submarines que requereixen de precisió i, precisament, el medi on es treballa no ajuda.

2. Objectius

L'objectiu del projecta consisteix en implementar un comportament per al robot Girona500 desenvolupat a la versió ROS de l'arquitectura COLA2 que ens permeti realitzar un control en posició del robot utilitzant informació visual d'una càmera o dades de navegació. Per aconseguir això s'haurà de:

- Desenvolupar un driver per la càmera del robot.
- Implementar el software que processi les imatges de la càmera per ubicar el robot en posició i determinar les correccions que s'hagin de transmetre als sistemes de navegació i control dels motors del robot per a realitzar la seva tasca.
- Desenvolupar un comportament que faci moure el robot en funció de l'error en posició detectat per el software de tractament d'imatges o procedents del mòdul de navegació.

3. Disseny del sistema

El disseny del sistema estarà estructura de la següent forma:

- Es tindrà un node que serà el driver de la càmera. Aquest node s'encarregarà de capturar les imatges de la càmera i publicar-les per el tòpic `imagesFromCamera`, de manera que el altres nodes podrà obtenir aquesta informació.

- Les imatges seran rebudes per el node de Processament d'Imatges. Aquest s'encarregarà de, fent servir una imatge com a referència, calcular els desplaçaments en píxels necessaris per tal que el template escollit com a referència es mantingui en una posició determinada. Per fer-ho podrà fer servir dues tècniques:
 - o Template Matching: Consisteix en agafar una imatge com a model. Després, l'algorisme anirà cercant-lo en les imatges que vagin arribant per tal d'ubicar la imatge de referència.
 - o SURF: Consisteix en agafar una imatge i extreure'n els seus punts d'interès com a referència. Després, l'algorisme anirà extraient els punts d'interès de les imatges que vagin arribant i buscarà coincidències amb els punt de referència per ubicar la imatge de referència.

Un cop s'ha ubicat el template, es calcularà el desplaçament que s'hauria d'aplicar per ubicar-lo allà on es vol i es publicaran aquestes correccions pel tòpic `pixel_correction`.

- Aquestes correccions en píxels es llegiran al node Keep Pose, que calcularà les consignes que s'hauria d'aplicar als motors per tal de que és mantingui la imatge ubicada allà on es vol.
- El sistema de navegació i control dels motors agafarà les consignes i les aplicarà als motors del robot.

En el següent diagrama es mostrarà el funcionament de tot el conjunt (Figura 1).

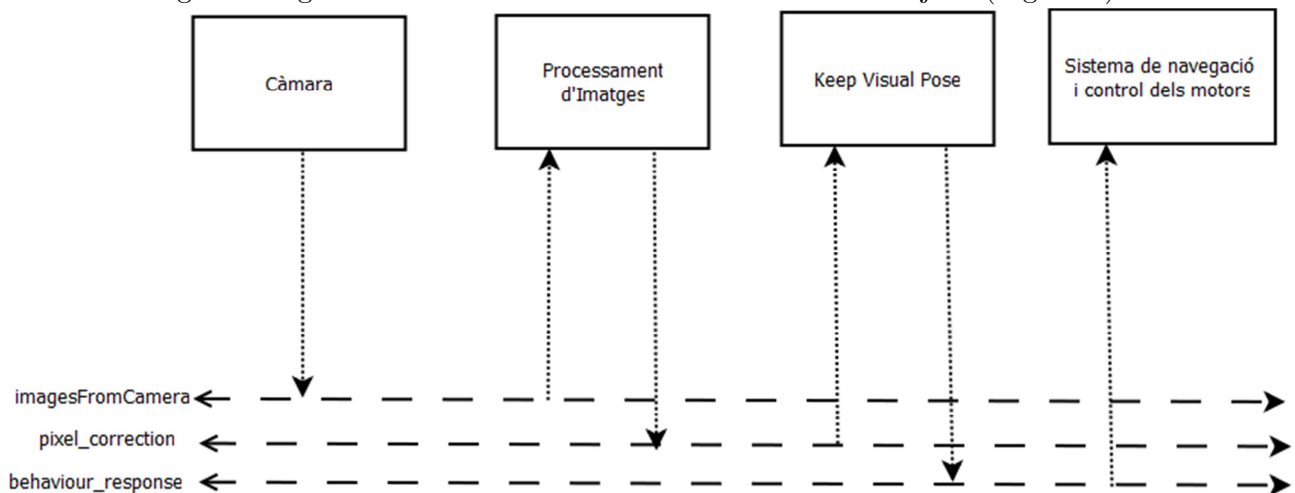


Figura 1: Diagrama funcionament del tot el conjunt

4. Implementació i Proves

4.1. Driver de la Càmera

Primer de tot la càmera captura una imatge de la framegrabber. Les OpenCV no estan pensades per capturar imatges amb framegrabber, per això apareixen uns marcs negres al voltant de la captura. Llavors es procedirà a retallar-los i després es calibrarà la imatge (Figura 2).



Figura 2: Exemple de Captura de la Càmera

Un cop la imatge està preparada, s'empaquetarà en un paquet d'imatges ROS i es publicarà pel tòpic imagesFromCamera.

4.2. Proves Template Matching

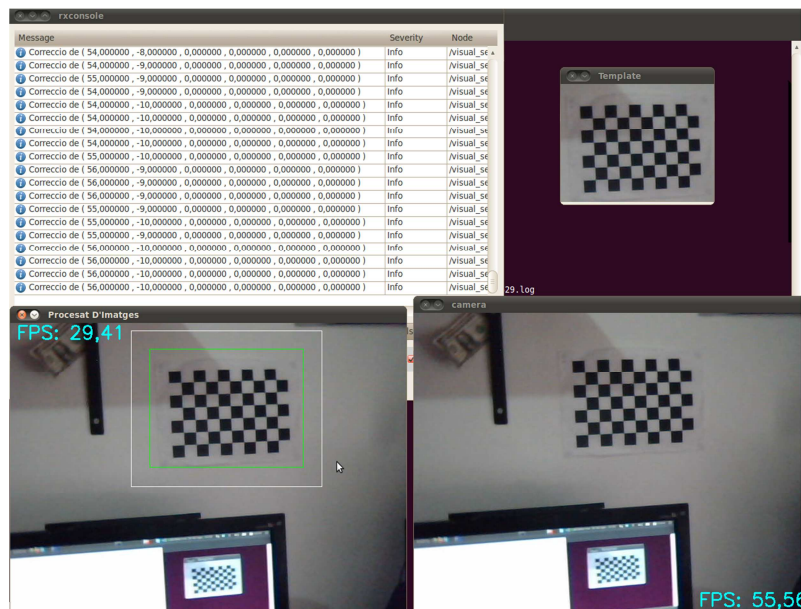


Figura 3: Exemple Template Matching en marxa

En el node de processat d'imatge amb la tècnica de Template Matching el que fa és, un cop es té el Template, buscar-lo a l'àrea de treball amb la resolució adequada (Figura 3, Finestra Processat d'Imatges, requadre blanc). Un cop s'ha localitzat i emmarcat el

template (Figura 3, Finestra Processat d'Imatges, requadre verd), es procedirà al càlcul de desplaçament per tal d'ubicar-lo allà on es vol i aquestes dades publicar-les pel tòpic `pixel_correction` (es poden veure les dades publicades a la Figura 3, finestra `rxconsole`).

4.3. Proves Surf

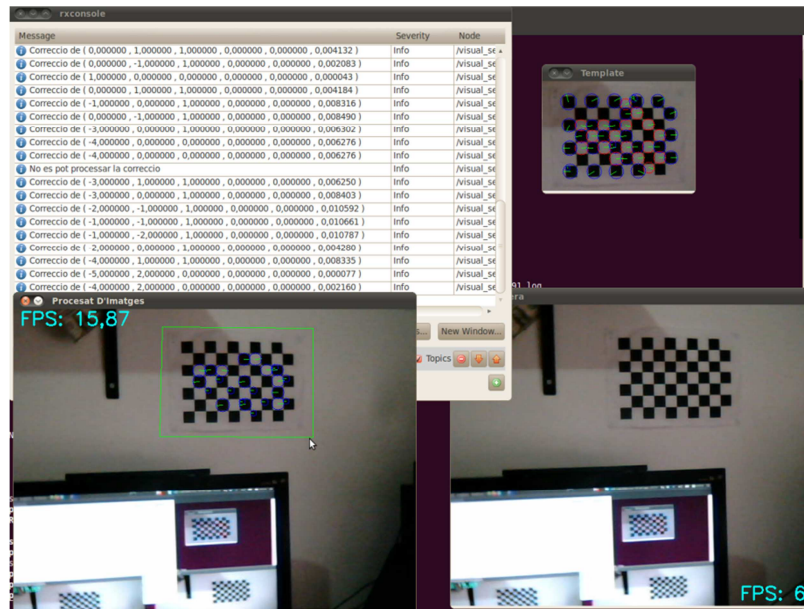


Figura 4: Exemple SURF en marxa

En el node de processat d'imatge amb la tècnica de SURF el que fa és, un cop es té el Template, calcular es seu punts d'interès (Figura 4, Finestra Template) buscar els punts d'interès de les imatges que van arribant (Figura 4, Finestra Processat d'Imatges) per tal de trobar coincidències. Un cop s'ha localitzat i emmarcat el template (Figura 4, Finestra Processat d'Imatges, requadre verd), es procedirà al càlcul de desplaçament per tal d'ubicar-lo allà on es vol i aquestes dades publicar-les pel tòpic `pixel_correction` (es poden veure les dades publicades a la Figura 4, finestra `rxconsole`).

4.4. Proves de Rendiment de node de Processat d'Imatges

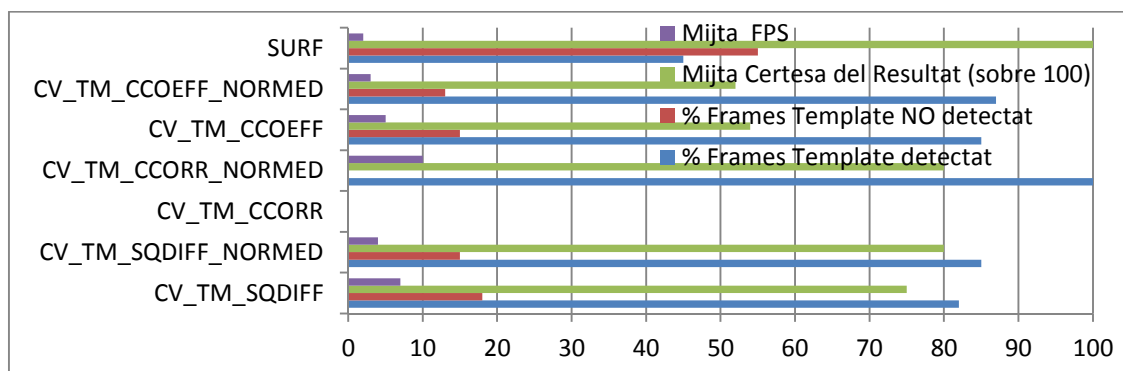


Figura 5: Resultat prova de rendiment amb tauler d'escacs i processador Atom

Per fer aquesta prova (Figura 5), s'ha fet servir un processador Intel Atom, el model de processador que porta incorporat el Girona500.

La primera cosa que s'ha de comentar es que els algorismes CCORR (Correlació creuada), donen resultats incongruents. Els resultats que donaven, comprovant-ho visualment amb la interfície d'usuari, eren incorrectes.

Els algorismes SQDIFF (Diferència de quadrats), detecten el template a més del 80% dels casos i amb una certesa del resultat d'un 75%, però quan perden el template es costa massa tornar-lo a identificar.

Els algorismes CCOEFF (Coeficient de correlació) són també molt bons, identifiquen el template sobre el 85% dels cops, però quan perden el template donen per bona l'àrea que més se li assembla al template i això són encerts que es compten i no s'haurien de tenir en compte, encara que quan el template torna a l'escena l'identifiquen al moment.

El SURF es poder el que té una taxa d'encerts més baixa, sobre el 45%, però sempre que identifica el template ho fa amb un 100% de certesa, cosa molt important a valorar. També s'ha de tenir en compte que es l'únic capaç de detectar els errors en z i yaw (encara que no ho es faci servir).

5. Conclusions

L'objectiu principal del projecte s'ha assolit satisfactòriament, s'ha implementat un comportament per a robot Girona500 en la versió ROS de l'arquitectura COLA2 que ens permet realitzar control en posició del robot utilitzant la informació tant de navegació com provinent d'una càmera.

Aquest objectiu, estava subdividit en tres parts.

La primera part del projecte era la desenvolupar un driver per a la càmera del robot, cosa que s'ha aconseguit, a més d'haver desenvolupat un mòdul extra que permet calibrar la càmera.

La segona part del projecte ha sigut implementar el software que processa les imatges de la càmera. Per fer-ho s'han fet servir dos mètodes:

- El Template Matching, que ha consistit en agafar una imatge com a referència i buscar on coincidia aquesta referència en les imatges que han anat arribant de la càmera per tal d'ubicar la imatge de referència.
- El SURF d'imatges, que ha consistit en agafar una imatge i obtenir els seus punts d'interès per a fer-los servir com a referència. Després s'ha anat buscant

els punts d'interès de referència a les imatges que han anat arribant de la càmera, per tal de trobar les coincidències i ubicar la imatge de referència.

Aquests dos mètodes, després de fer experiments i proves, s'han arribat a les següents conclusions:

- El Template Matching és molt útil en escenes en les que el robot no es veu molt afectat per les corrents d'aigua (això implicarà que hi no hagi molt de moviment) i dona molt bones taxes de frames per segon. Aquest algorisme te de dolent que, quan no troba correctament el template, sempre buscarà la coincidència més semblant i possiblement en alguns casos acabi per enganyar amb les correccions. També una cosa molt important que s'ha de tenir en compte es que aquest algorisme només podrà donar correccions els graus de llibertat x i y del robot, i per a corregir el de z i yaw fa falta ajuda d'altres mòduls externs al nostre, cosa que amb SURF no és necessari.
- El SURF es molt útil en escenes amb molts objectes i moviments bruscos, no dona tan bones taxes de frames per segon com el Template Matching (encara que en els casos en els que el Template Matching falla, el SURF pot doblar les taxes de FPS del Template Matching) però això es veu compensat per el fet de que mai falsos positius, o troba la referència o no la troba, a més dona correccions en tots els graus de llibertat del robot.

La tercera part i última, ha estat desenvolupar un comportament que faci moure el robot en funció de l'error en posició detectat per mòdul de tractament d'imatges o les dades de navegació. Amb les proves realitzades es pot comprovar que aquest mòdul tracta les dades de correccions i envia les consignes als motor d'acord amb l'error a corregir i ens els graus de llibertat que pertoca.

6. Treball futur

Com a treball futur es podrien realitzar proves en mar obert per estudiar el comportament del sistema en un entorn més complex. Un altre pas més en el projecte seria poder seguir objectes en moviment dintre de l'aigua. I finalment també es podria millorar el rendiment del sistema amb algorismes menys costosos a nivell computacional.