Universitat de Girona
Escola Politècnica Superior

Projecte fi de grau

Estudi: Grau en Enginyeria Informàtica

Títol: Implemetació de tècniques de percepció NDT
i ajust d'aquestes al domini submarí

Document: Resum

Alumne: Miguel Malagón Pedrosa

Tutors: Pau Vial Serrat i Narcís Palomeras Rovira
Departament: Arquitectura i Tecnologia de Computadors
Àrea: Arquitectura i Tecnologia de Computadors

Convocatòria (mes/any): Juny 2021

Universitat de Girona
Escola Politècnica Superior

Projecte Fi de Grau

Implemetació de tècniques de percepció NDT i ajust d'aquestes al domini submarí

Autors:
Miguel Malagón Pedrosa
Ricard Segura Duran

Juny 2021

Grau en Enginyeria Informàtica

Tutors:
Pau Vial Serrat
Narcís Palomeras Rovira

1 Introducció

En aquest treball es vol aprofundir en la percepció i la construcció de mapes submarins, duta a terme per Vehicles Submarins Autònoms (AUV). Els AUV són plataformes robòtiques dotades de sistemes de percepció i control que operen sota l'aigua i que permeten efectuar diferents tasques com la localització, el mapeig i la planificació de trajectòries de manera autònoma. Els AUV utilitzen sensors acústics per percebre el seu entorn. Les propietats de l'aigua dificulten la utilització dels sensors basats en fenòmens electromagnètics que s'utilitzen habitualment en les aplicacions robòtiques terrestres o aèries, com ara els làsers o les càmeres òptiques. Per tant és habitual recórrer als sensors que apliquen fenòmens acústics, com els sonars.

El problema principal al que es vol adreçar aquest projecte és el registre per generar mapes d'ocupació, aquests es generen processant les dades de sensors de rang, on per evitar incoherències s'utilitzen tècniques d'associació de vistes, aquestes tècniques consisteixen en desplaçar la posició dels escanejos per tal de que siguin coincidents amb la resta, generant així millors mapes d'ocupació.

Existeixen diferents problemes a l'hora de treballar amb sensors acústics en l'entorn submarí. Els sonars, a diferència de la seva contrapart electromagnètica, són molt més lents i generen dades molt més sorolloses. En conseqüència, l'aplicació de les tècniques de registre de l'estat de l'art, orientades a sensors làser, a dades acústiques no donen resultats satisfactoris. Per solucionar aquesta problemàtica en aquest projecte es vol implementar una tècnica de registre específica per fer el registre de núvols de punts obtinguts amb un sonar en un entorn submarí. Per tant, en aquest projecte s'ha treballat en la implementació d'una llibreria en C++ basada en la tècnica Normal Distributions Transform (NDT) adaptada a les característiques de les dades acústiques. S'ha posat molt d'èmfasi en l'eficiència computacional de la implementació per assolir una execució en temps real i s'ha preparat per poder ser utilitzada tant en núvols de punts bidimensionals com tridimensionals.

2 Metodologia

Per al desenvolupament del projecte s'ha seguit una metodologia de desenvolupament de tipus àgil, on s'han definit sprints setmanals per realitzar les diferents tasques. S'ha realitzat d'una forma iterativa i incremental, on cada versió nova que es genera millora la versió anterior del projecte; refinant i millorant les diferents parts i on, alhora, s'implementen noves funcionalitats. La introducció a l'àmbit de treball i a la tècnica de registre s'ha fet de manera conjunta amb l'estudiant Ricard Segura. Això ens ha permès avançar més ràpidament amb els coneixements teòrics. Alhora, s'ha realitzat conjuntament el disseny de l'estructura de la llibreria i la implementació dels objectes bàsics. Aquest fet ens ha permès acordar l'estructura bàsica de la llibreria que després evolucionarem per separat.

3 Disseny i implementació base de la llibreria

El disseny de la llibreria s'ha basat en la definició de tres conceptes. En primer lloc, el Front-End s'encarrega de fitar un Gaussian Mixtures Model (GMM) a un núvol de punts. Un GMM és un model probabilístic definit per la suma de múltiples distribucions gaussianes

$$p(x|\theta) = \sum^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \qquad \text{amb} \qquad \sum^{K} \pi_k = 1,$$

Aquestes distribucions, anomenades components k, estan definides per una mitjana μ_k, una matriu de covariància Σ_k i un pes π_k; on la suma dels pesos de totes les components ha de ser 1. Per tant, el Front-End permet passar d'una representació discreta de la superfície percebuda a un model continu, a més de fer una compressió de les dades. En segon lloc, el mètode de registre defineix el problema de registre com a un problema d'optimització, on la variable de decisió és la transformada que permet registrar els dos núvols de punts. Es poden definir dues funcions de cost. Per una banda, el mètode Punt a Distribució (P2D) registra un scan parametritzat com a núvol de punts contra un scan parametritzat amb un GMM. Per tant, la funció de cost maximitza el likelihood del núvol de punts al model. Per altra

banda, el mètode Distribució a Distribució (D2D) registra dos scans parametritzats amb GMM. Per tant, la funció de cost minimitza la divergència entre les dues distribucions. En tercer lloc, el solucionador s'encarrega de resoldre aplicant mètodes numèrics el problema d'optimització definit pel mètode de registre.

Pel que fa a l'estructura de la llibreria, el seu element base és la classe GaussianMixtureModel que permet emmagatzemar un GMM i inclou mètodes de visualització. El Front-End s'implementa com una llibreria de funcions on a totes les funcions es passa un núvol de punts com a argument i es retorna un objecte de tipus Gaussian Mixture Model. S'estructura com a llibreria ja que existeixen diferents aproximacions a l'hora de fitar un GMM. Els ScanMatchingMethods s'encarreguen de donar una funció de cost a la relació entre dos registres, ja siguin núvols de punts o GMMs, on un dels dos s'ha desplaçat seguint una transformació t a l'espai. El mètode de registre s'ha implementat com una interfície, anomenada ScanMatchingMethod. Fer una interfície amb un seguit de subclasses ens permet utilitzar les funcions definides per l'interfície que están implementades en la subclasse sense haver de saber quina és la subclasse, això ens permetrà més endavant donar al solver un punter a un objecte del tipus de l'interfície i que aquest utilitzi les funcions implementades a les subclasses. Les funcions que necessitarà el solver són compute_score(), compute_score_and_gradient() i compute_score_gradient_and_hessian(). Aquesta interfície està templaritzada per poder definir subclasses tant en casos 2D com 3D.

Finalment, el solucionador també s'ha implementat com una interfície, anomenada Solver. El motius de l'interfície són els mateixos de poder establir un format a seguir en les subclasses i poder utilitzar les mateixes funcions sense dependre de la subclasse. Aquests implementen la funció compute_optimum(), que és l'encarregada de fer l'optimització seguint el mètode de Newton o alguna modificació d'aquest. Aquesta interfície també es troba templaritzada segons la dimensió, però a diferència de l'anterior, les subclasses també ho estan, ja que el mètode de Newton i les modificacions implementades segueixen la mateixa estructura independentment de la dimensió.

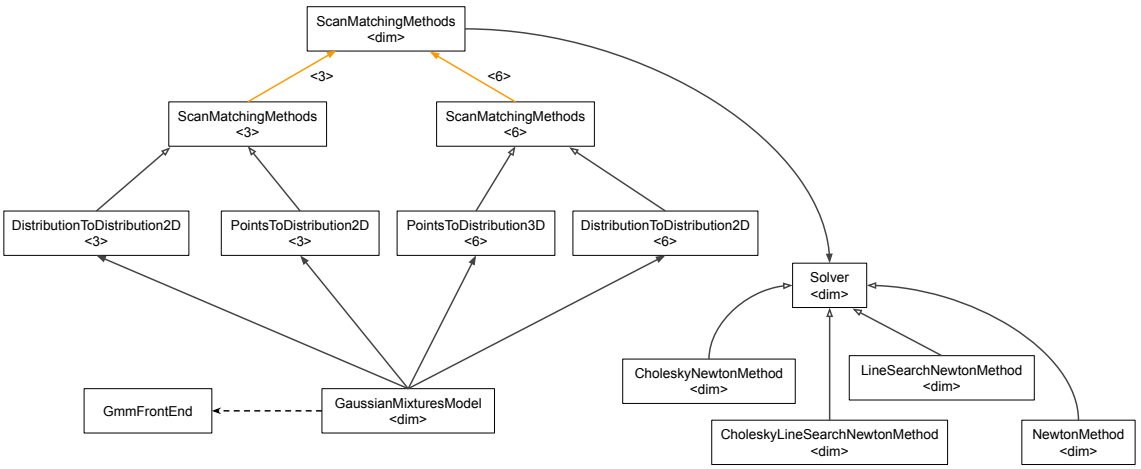

Figura 1: Diagrama de classes

A la Figura 1 es mostra l'estructura de classes de la llibreria. S'hi mostren les interfícies i les herències implementades. Pel que fa a la implementació de la llibreria de Front-Ends, en aquest projecte només s'ha implementat una funció basada en la tècnica NDT, anomenada ndt_constructor(). La tècnica NDT clusteritza un núvol de punts projectant una graella sobre el núvol. Per cada cel·la amb un mínim nombre de punts, genera una component al GMM fitant una distribució gaussiana.

De la interfície ScanMatchingMethods s'han implementat 3 subclasses. Les classes Points2Distrtibution2D i Points2Distrtibution3D, defineixen un registre P2D en dues i tres dimensions respectivament: mentre que la classe Distrtibution2Distrtibution2D, defineix un registre D2D bidimensional. Les implementa-

cions s'han separat completament ja que, tot i que la funció de cost només varia entre P2D i D2D, els càlculs de les derivades són completament diferents d'uns als altres. El codi és impossible de templaritzar. Destacar que la classe Distribution2Distribution3D encara no està implementada i forma part del treball futur del projecte.

Finalment, de la interfície Solver s'han implementat 4 subclasses. Totes tenen de base la implementació de la classe NewtonMethod, la qual aplica el mètode de Newton per resoldre un problema de minimització sobre la funció de cost del registre i obtenir la transformació òptima entre els scans. En aquest algorisme, segons les derivades primera i segona de la funció objectius definides a les implementacions de ScanMatchingMethod, a cada iteració es decideix una direcció i un pas per a l'optimització aplicant

$$x_{n+1} = x_n - \alpha \frac{f'(x_n)}{f''(x_n)}.$$

La classe LineSearchNewtonMethod afegeix al mètode de Newton un algorisme basat en les condicions de Wolfe que determina la mida òptima del pas α que defineix el mètode de Newton a cada iteració. La classe CholeskyNewtonMethod aplica la factorització de Cholesky modificada per factoritzar la matriu Hessiana i pertorbar-la si és indefinida per fer-la definida positiva i assegurar una direcció de minimització. Per acabar, la classe CholeskyLineSearchNewtonMethod combina les funcionalitats de LineSearchNewtonMethod i CholeskyLineSearchNewtonMethod per tenir un solucionador complet. Tot aquest codi fet per la llibreria es penjarà en un repositori públic junt amb la seva documentació generada amb doxygen.

En aquest punt al tenir la llibreria base feta, es van separar els projectes, en aquest s'ha posat èmfasi en la millora del temps d'execució d'un registre i la seva eficiència computacional. L'eficiència és un paràmetre de disseny molt important ja que la llibreria es vol aplicar en temps real per construir en línia un mapa d'ocupació de l'entorn del robot. Es proposen dues millores. En primer lloc, es detecta que les funcions amb més càlculs matricials i iteracions es troben a les subclasses de ScanMatchingMethods. No obstant, es veu que aquests càlculs es poden paral·lelitzar. Per fer-ho, es fa ús de l'API OpenMP i es reestructura el codi al no ser compatible amb variables de tipus de la llibreria d'Eigen, amb la que es defineixen les operacions algebraiques.

4 Millora de l'eficiència computacional

Per comprovar la millora de temps es van provar diferents números de fils pels càlculs de les funcions del mètode P2D i D2D, en la Figura 2 veiem com per exemple millora el temps de 5ms a 2ms en la cinquena i sisena columna en el cas P2D.

En segon lloc, es detecta que a la classe Points2Distrtibution2D es fan associacions punt-component que aporten molt poca informació al registre. Per tant, es proposa un filtre que permeti descartar de manera automàtica les associacions poc informatives. Per fer-ho s'implementa un test de Chi quadrat i només s'avaluarà el cost i les derivades d'aquelles associacions que passin el test. Es considera que el test s'ha passat si el punt es troba a certa distància de la distribució, aquesta distància s'anomena distància de Mahalanobis i es calcula de la següent manera

$$d_M = (p - \mu_k)^T \Sigma_k^{-1} (p - \mu_k),$$

on p és un punt, μ_k és la mitjana que defineix la distribució k i Σ_k és la matriu de covariància de la distribució.

Per comprovar el funcionament d'aquesta millora es va comprovar la diferència entre les transformacions amb i sense l'optimització i també es va avaluar qualitativament observant els gràfics com el de la Figura 3 que representa el procés de l'optimització pel registre de dos escanejos utilitzant el mètode P2D abans i després de l'optimització amb el test de chi quadrat. Es pot veure com tots dos aconsegueixen col·locar el núvol de punts coincident amb l'anterior.

Tal i com es veu a la dreta de la Figura 4 veiem que la millora si que genera resultats semblants a l'original retallant una quantitat de temps considerable com mostra a l'esquerra de la Figura 4.

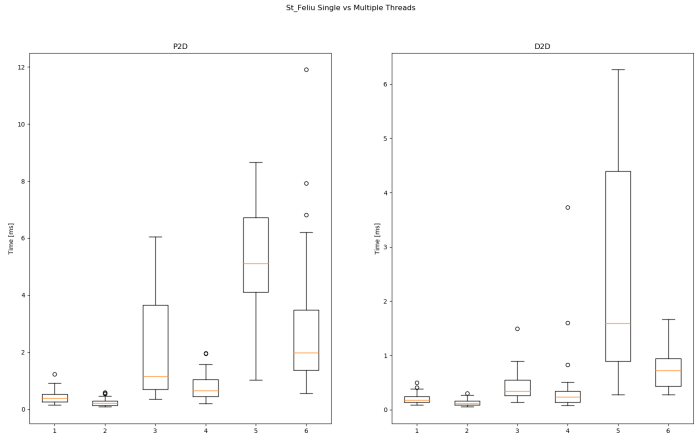

Figura 2: Temps emprat per fer els càlculs de score, vector jacobià i matriu hessiana, utilitzant 2 fils al paral·lelitzar, cada columna del boxplot representa un càlcul diferent, 1,2:score, 3,4:jacobià, 5,6:hessiana, on els parells están paral·lelitzats i els imparells no

5 Processat de dades tridimensionals

Finalment s'ha estés la llibreria al processat de núvols de punts tridimensionals.Es va començar per la creació de GMM en 3D. Per l'adició d'una nova dimensió en la funció ndt_constructor() es va haver de refer el control de la graella per no ocupar una quantitat de memòria innecessària.

En la Figura 5 es veu la representació en tres dimensions del GMM generat amb el núvol de punts de l'esquerra, on els el·lipsoides representen la covariància i els punts la mitja de cada component.

Amb els GMM definits en tres dimensions es va començar l'adaptació del solver i el mètode de registre, però les derivades de la funció de cost es van definir malament, el que no va permetre poder tenir acabada tota la tècnica per l'utilització de núvols de punts tridimensionals en aquest treball.

6 Conclusions

En conclusió, en aquest projecte s'ha implementat una llibreria en C++ per al registre de núvols de punts obtinguts amb sonar en un entorn submarí, on les dades que s'obtenen són molt sorolloses. També s'ha assegurat l'eficiència computacional del codi per poder ser utilitzat en temps real en un AUV. Alhora, es proporcionen diferents mètodes i solucionadors per resoldre el registre i s'ha començat a treballar en l'extensió tridimensional de la llibreria.

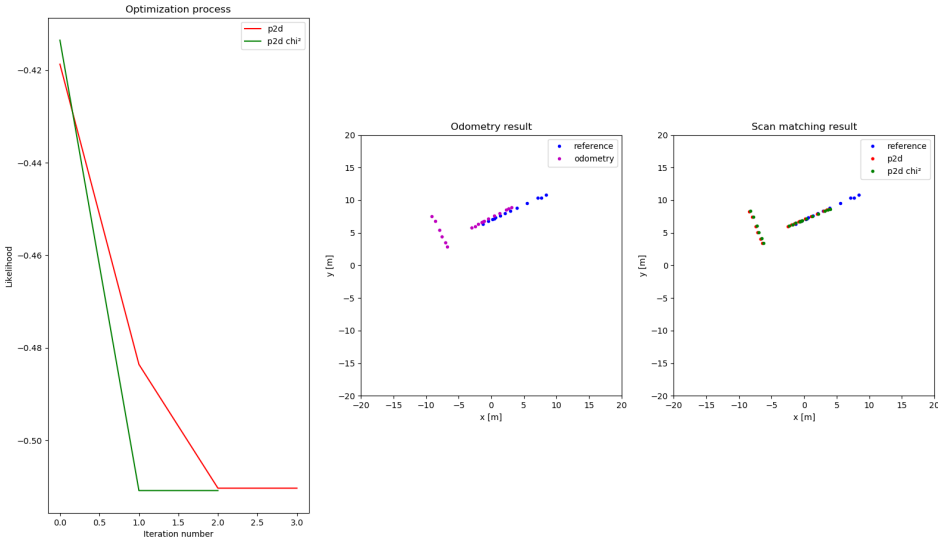

Figura 3: Exemple d'un registre particular. Esquerra: Evolució del score durant l'optimització. Centre: Associació d'escanejos provinent del sistema de navegació a la deriva. Dreta: Registre.

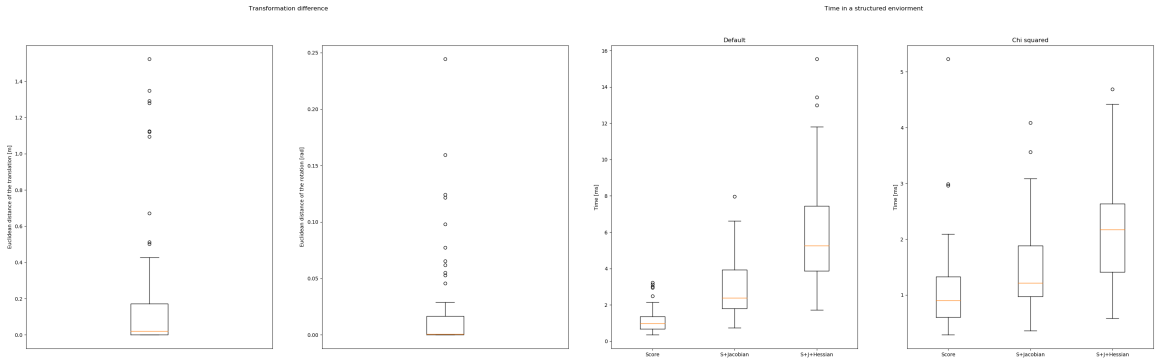

Figura 4: Dreta: Error en el registre en un entorn estructurat. Esquerra: Temps per computar score, jacobiana i hessiana en un entorn estructurat.

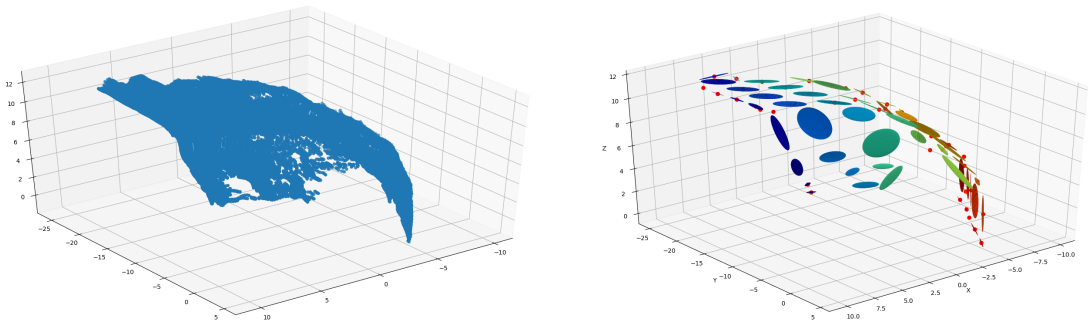

Figura 5: Gràfic amb el núvol de punts original i el GMM de 3 dimensions generat