

Calcul ensembliste pour la robotique: développement de robots marins, sous marins, terrestres ou aériens pour l'exploration

Rapport d'avancement T0+9 mois (convention MRIS 2010-2011)

F. Le Bars, A. Bethencourt, J. Sliwka, L. Jaulin

ENSTA Bretagne

15 Septembre 2011

Descriptif de l'étude

Dans le cadre de cette étude, nous cherchons à réaliser des plateformes robotiques marines, sous-marines, terrestres et aériennes dans le but de valider des concepts et algorithmes divers, notamment en utilisant des méthodes ensemblistes telles que le calcul par intervalles. Les thématiques liées à ce projet sont variées : réalisation d'une mécanique simple et robuste, télé-opération, autonomie, localisation et cartographie, détection et reconnaissance d'objets, collaboration entre robots...

Déroulement du concours SAUC-E 2011

Comme tous les ans, nous avons participé au concours SAUC-E (Student Autonomous Underwater Challenge - Europe), qui a eu lieu cette année du 4 au 10 Juillet à La Spezia en Italie. Nous avons terminé 4^{ème} sur 10 équipes, les 1^{ers} étant l'Université de Luebeck (Allemagne), les 2^{èmes} l'Université de Girona (Espagne) et les 3^{èmes} l'Université de Bremen (Allemagne). 10 équipes étaient présentes.



Figure 1 : L'équipe de l'ENSTA Bretagne pour SAUC'E 2011 avec les sous-marins SAUC'ISSE et SARDINE, ainsi que notre hovercraft (robot de surface)

Les épreuves se sont déroulées dans les mêmes conditions que l'année précédente : dans l'eau de mer d'une marina du NURC (NATO Undersea Research Centre).



Figure 2 : Zone de compétition de SAUC-E 2011

Les épreuves que les AUV (Autonomous Underwater Vehicle) devaient effectuer étaient les suivantes :

- Choisir le point de départ du sous-marin. 2 points de départs étaient possibles. Le premier dit « Start 1 » se trouvait sur le bord Nord ou Sud de la marina et l'autre dit « Start 2 » se trouvait au milieu.
- Passer ensuite par la porte de validation (nommée Validation Gate, aussi utilisée comme épreuve de qualification).
- Faire demi-tour et suivre un pipeline non rectiligne (en « S ») de couleur jaune à 50 cm de distance.
- Contourner puis libérer une bouée blanche se trouvant à proximité de la fin du pipeline en coupant le fil qui la maintenait.
- Suivre les murs formant l'un des coins de la marina à une distance supérieure à 2 m.
- Suivre le véhicule de surface autonome du NURC (par le dessous). Celui-ci était muni d'un pinger (générateur de signaux acoustiques) émettant toutes les secondes à 12 kHz.
- Faire surface sous le véhicule de surface.

Un fichier contenant les positions et actions du sous-marin en fonction du temps, ainsi que des preuves de détection (images du pipeline...) devaient être fournis aux organisateurs à la fin des épreuves.

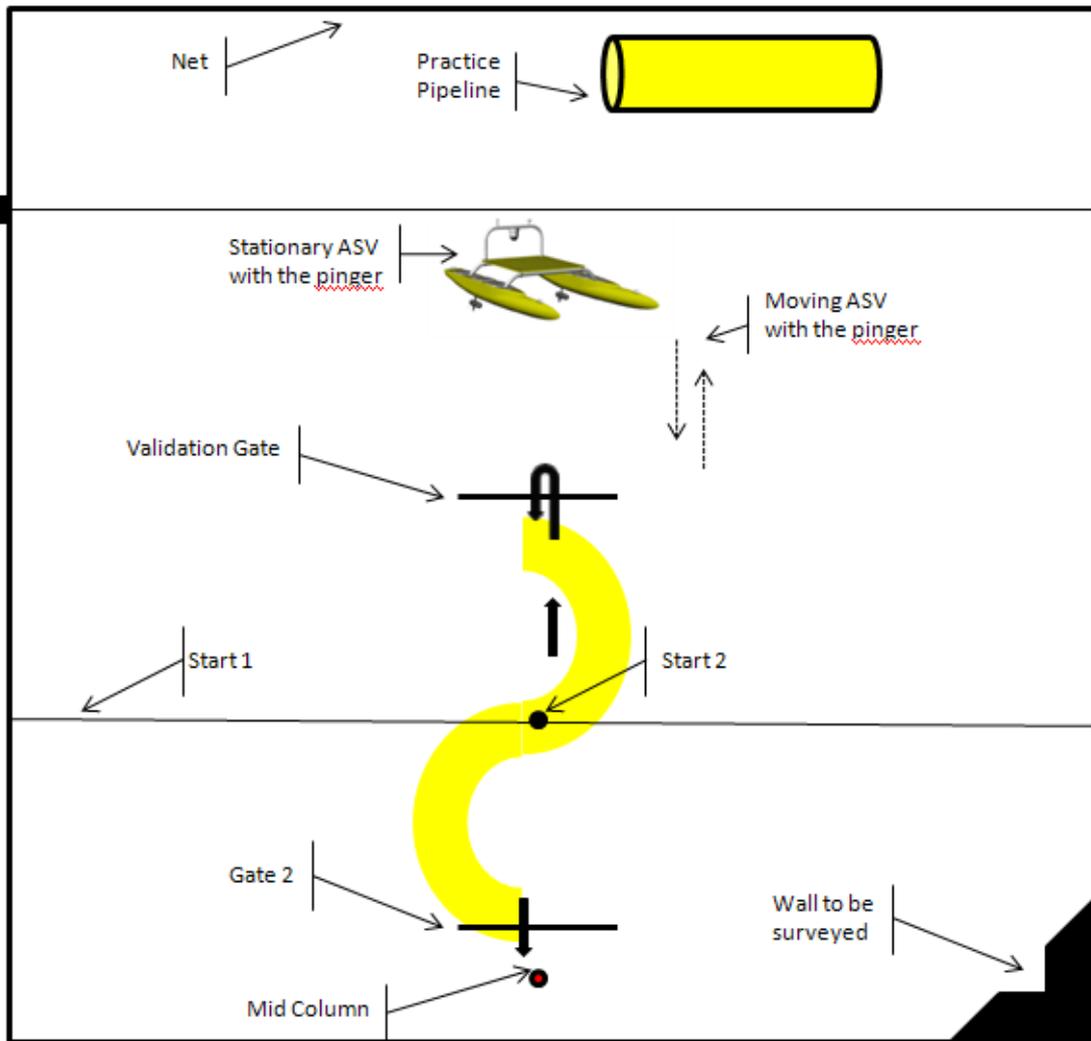


Figure 3 : Epreuves du concours SAUC-E 2011

Durant la compétition, nous avons eu plusieurs problèmes :

- L'alimentation des caméras de SAUC'ISSE a cessé de fonctionner de la même façon que l'année dernière, puis après remplacement les caméras ont elles-mêmes montré des signes de faiblesses. Nous avons dû les remplacer et modifier rapidement l'alimentation pour tout faire pour que le problème se reproduise. Cependant, nous n'avons pas encore clairement identifié sa cause.
- Lors d'un transport, l'un des connecteurs étanches de SAUC'ISSE a été endommagé. Il a été rapidement remplacé.
- Un fil (pour retenir le sous-marin) s'est coincé dans le propulseur vertical de SARDINE alors qu'il était en fonctionnement, ce qui l'a détruit. Il a fallu le remplacer aussi.
- Un brouilleur WIFI était présent sur la zone de compétition, sans que les organisateurs le sachent. Nous avons pu détecter sa présence et prévenir les organisateurs pour qu'ils le configurent pour autoriser nos réseaux WIFI. Toutes les équipes étaient affectées par ce problème.
- Un problème de gestion des temps de calculs a empêché de réussir correctement plusieurs missions lors de la finale.

Malgré ces problèmes, nous étions 3^{ème} aux qualifications, puis 2^{ème} après la demi-finale. Malheureusement, le dernier problème cité nous a fait perdre des places au classement final.



Figure 4 : SAUC'ISSE, le robot sous-marin principal



Figure 5 : SARDINE, le 2^{ème} robot sous-marin

Logiciel d'aide à la détection de mines sous-marines

Présentation

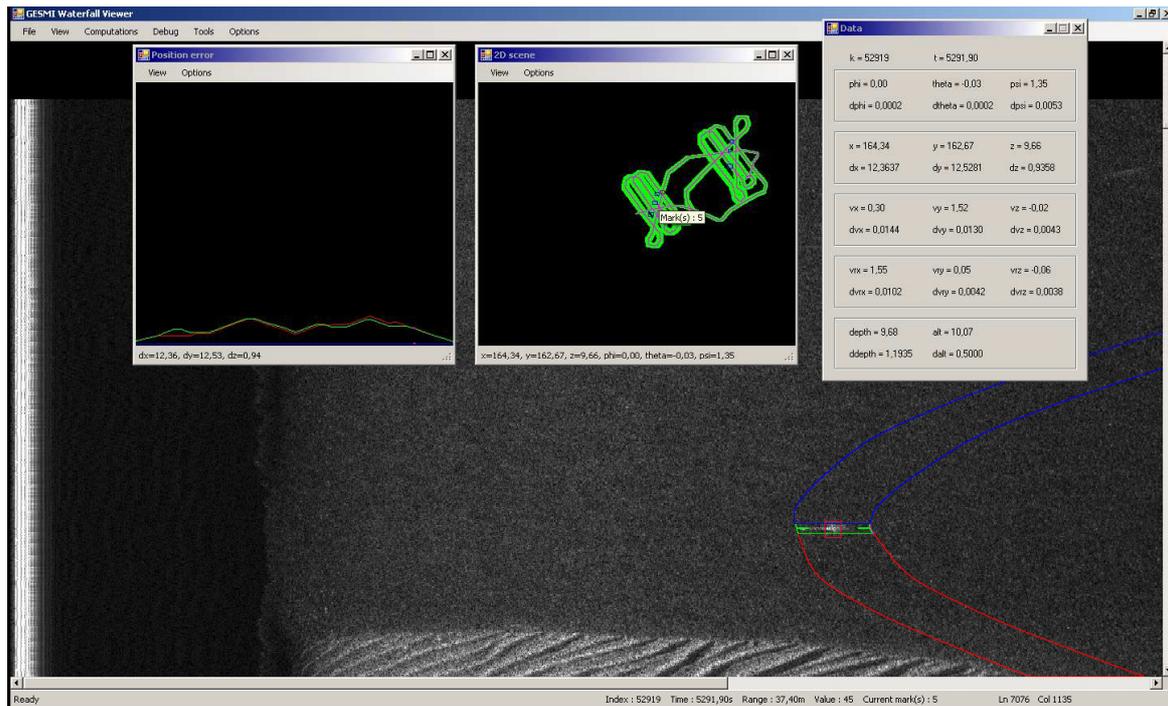


Figure 6 : Capture d'écran de GESMI, programme qui calcule et affiche la position des mines, l'enveloppe de la trajectoire, l'erreur en position d'un sous-marin lors d'une expérience de quadrillage de zone inconnue, et qui aide à identifier les mines parmi les différentes détections

Dans le cadre d'un contrat avec le GESMA sur le traitement des données de leur sous-marin Daurade, un logiciel d'aide à la détection de mines par SLAM (Simultaneous Localization And Mapping) par intervalles a été réalisé. Ce logiciel, nommé GESMI (Guaranteed Estimation of Sea Mines with Intervals) est basé sur celui déjà commencé en 2007, décrit notamment et qui traitait les données du Redermor. C'est un programme permettant d'estimer la trajectoire d'un robot sous-marin et la position d'objets particuliers (préalablement détectés au moins 1 fois par un opérateur humain en parcourant les données sonar) en utilisant le calcul par intervalles. Il permet en même temps de vérifier la cohérence entre les données qui lui sont fournies en entrée et d'aider à la détection des objets sur l'image sonar.

Les données dont il a besoin pour fonctionner sont les suivantes :

- Données de navigation du sous-marin (angles du sous-marin, altitude, profondeur, vitesses dans le repère mobile, quelques positions GPS connues)
- La distance sur l'image sonar où un amer a été détecté et s'il est sur tribord ou bâbord.
- Les temps et estimation d'erreur maximale pour chacune des données.
- La portée et l'erreur en distance du sonar.
- Les images sonar.

Les données qu'il produit ou affiche sont les suivantes :

- La trajectoire du sous-marin et la position des amers (sous forme d'enveloppe et centre de l'enveloppe ainsi que d'intervalles).
- L'erreur en position au cours du temps.
- La reconstitution de l'image sonar indiquant les distances sous-marin à amers, montrant les endroits où le sous-marin est passé à proximité des amers.

Le programme a été repris et amélioré notamment pour permettre son utilisation avec les données de la Daurade (rajout de la gestion de la partie bâbord des images sonar...), faciliter l'interprétation des résultats (affichage des erreurs en position...) et aider au maximum l'opérateur humain dans sa recherche de mines (amélioration des indications de positions estimées de mines directement sur l'image sonar...).

Concepts

GESMI utilise à la fois les données de navigation (vitesses, angles...) et les détections d'amers vus par un opérateur humain sur les images sonar pour estimer le plus précisément possible une enveloppe de la trajectoire du sous-marin et de la position réelle de ces amers. La cohérence entre les données de navigation et les détections est vérifiée en même temps.

Dans le programme, toutes les données sont représentées par des intervalles, pavés ou tubes. Par exemple, si à $t = 308,0$ s la centrale inertielle du sous-marin indique un angle de $-0,0258$ rad et que la documentation du constructeur de la centrale inertielle précise que son erreur maximale est de $0,0001$ rad, l'angle à $t = 308,0$ s sera représenté dans le programme par l'intervalle $[-0,0258-0,0001; -0,0258+0,0001] = [-0,0259; -0,0257]$. Tous les intervalles obtenus à partir des fichiers de données chargés par le programme sont ensuite utilisés dans des calculs mettant en jeu les équations d'état du sous-marin ainsi que d'autres relations géométriques. Ces équations et relations sont discrétisées (par rapport au temps) et mettent en jeu des additions, soustractions, cos, sin... d'intervalles de la même façon qu'avec des nombres réels, ainsi que des intersections et unions d'intervalles. Si un intervalle vide est trouvé lors d'un calcul, cela signifie en général qu'il y a une incohérence dans les données de navigation ou détections. Le plus souvent, elle est due à des données capteur moins précises que l'on croyait. Il faut donc augmenter l'erreur avec laquelle on considère ces données. Des détections imprécises d'objets sur l'image sonar peuvent aussi provoquer des incohérences. Dans ce cas il faut vérifier les détections et éventuellement éliminer celles qui sont trop approximatives pour pouvoir les corriger par la suite à l'aide de la reconstitution de l'image sonar.

Les 3 principales étapes de calcul pour la reconstitution de la trajectoire du sous-marin sont les suivantes :

- Propagation : les équations d'état du sous-marin donnent une relation entre les positions et données de navigation à t et $t + dt$. Des calculs sont donc faits en partant de $t = 0$ s vers la ...n. Par exemple, la position GPS du sous-marin à $t = 0$ s va permettre d'estimer ses positions suivantes avec une erreur qui augmente au cours du temps et qui dépend aussi de l'erreur avec laquelle on connaît ses différentes données de navigation (vitesses, angles...).
- Retro-propagation : les équations d'état du sous-marin peuvent donner aussi une relation entre les positions et données de navigation à $t + dt$ et t . Des calculs sont donc faits en partant de la fin vers $t = 0$ s. Par exemple, la position GPS finale du sous-marin va permettre d'améliorer l'estimation de ses positions précédentes.
- Mise en cohérence avec les objets particuliers détectés sur l'image sonar : des relations géométriques entre les positions du sous-marin et celles des objets à leurs temps de détection sur l'image sonar permettent d'une part d'estimer la position des objets dans le repère de navigation et d'améliorer l'estimation de la trajectoire du sous-marin d'autre part (principe du SLAM).

Ces 3 étapes doivent en général être réexécutées plusieurs fois pour atteindre la meilleure précision (on s'arrête lorsqu'on voit que l'estimation de l'erreur et de la position ne change plus).

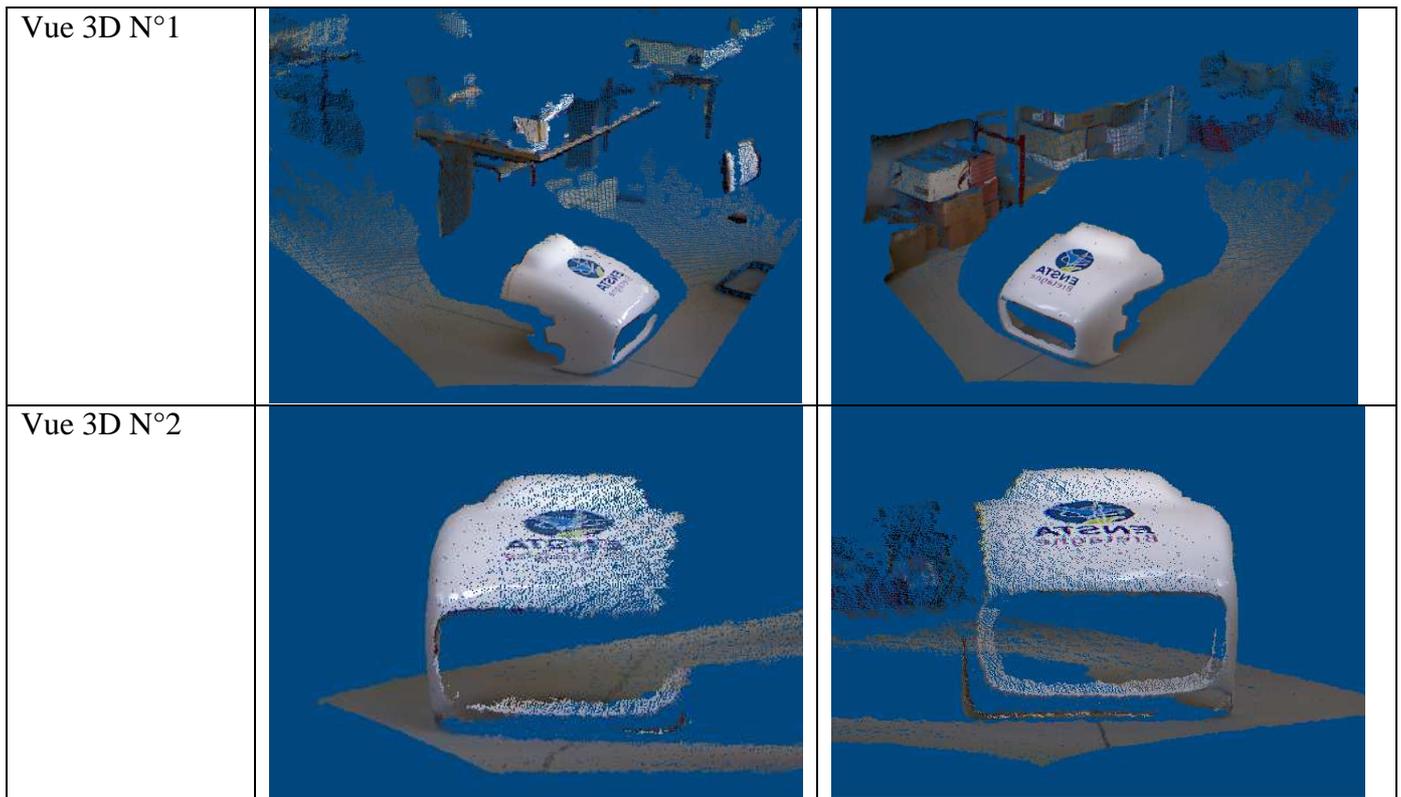
Une fois la trajectoire reconstituée avec la meilleure précision qu'on pouvait, GESMI peut indiquer à l'utilisateur les éventuelles détections d'objets sur l'image sonar qu'il aurait oubliées. En effet, connaissant la trajectoire du sous-marin et au moins une détection sur l'image sonar d'un objet, le logiciel peut indiquer les autres moments où le sous-marin a pu repasser devant l'objet et le voir sur le sonar. Ceci est affiché sous forme de reconstitution de l'image sonar qui affiche les enveloppes de la distance objet-sous-marin, superposées aux données sonar réelles.

Reconstruction 3D par méthodes ensemblistes avec une Kinect associée à une centrale inertielle

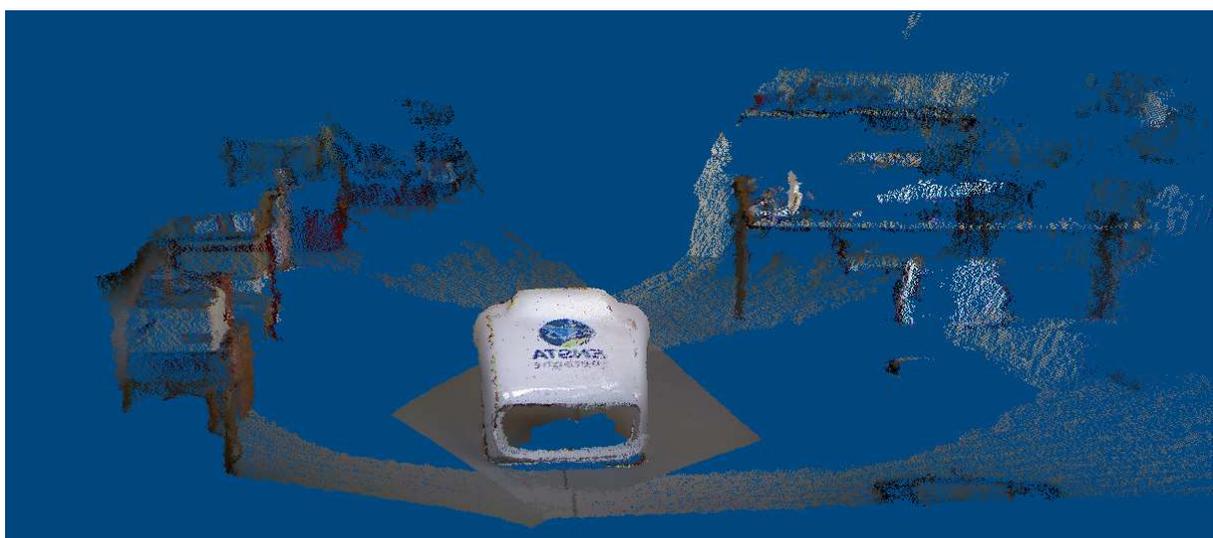
La disponibilité de caméras RGB-D à bas coût comme la Microsoft Kinect a récemment permis d'augmenter l'intérêt pour les applications de VSLAM (Visual Simultaneous Localization And Mapping) utilisées par exemple pour la modélisation ou reconstruction 3D d'environnements intérieurs. Pour estimer la transformation entre deux positions, la plupart des techniques existantes utilisent l'algorithme ICP (Interest Closest Point) combiné à RANSAC sur le nuage de points des deux vues. Une nouvelle approche basée sur le calcul par intervalles pour calculer de manière robuste les paramètres de transformation a été développée.

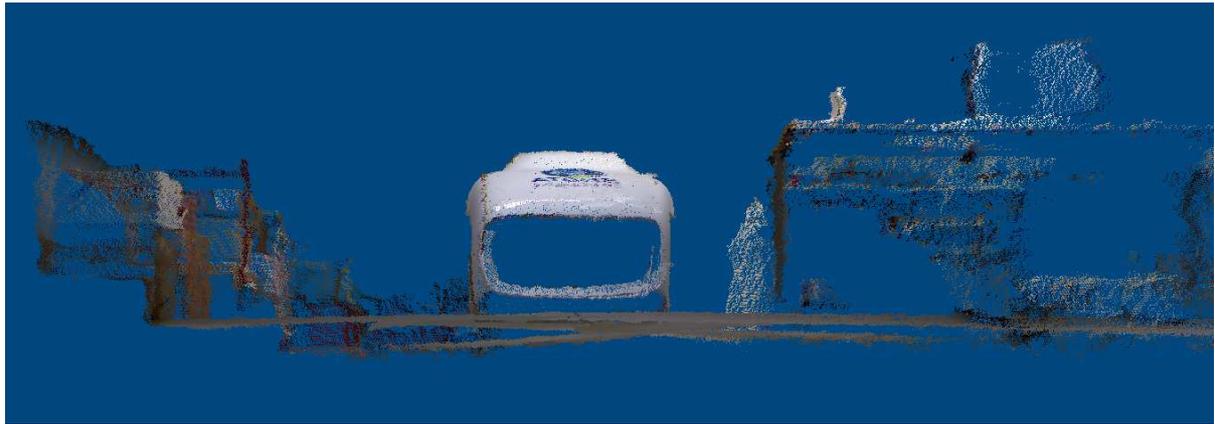
Pour cela nous avons tout d'abord pris deux photos 3D d'une structure mécanique avec la Kinect :

POSE N°	1	2
Vue 2D		

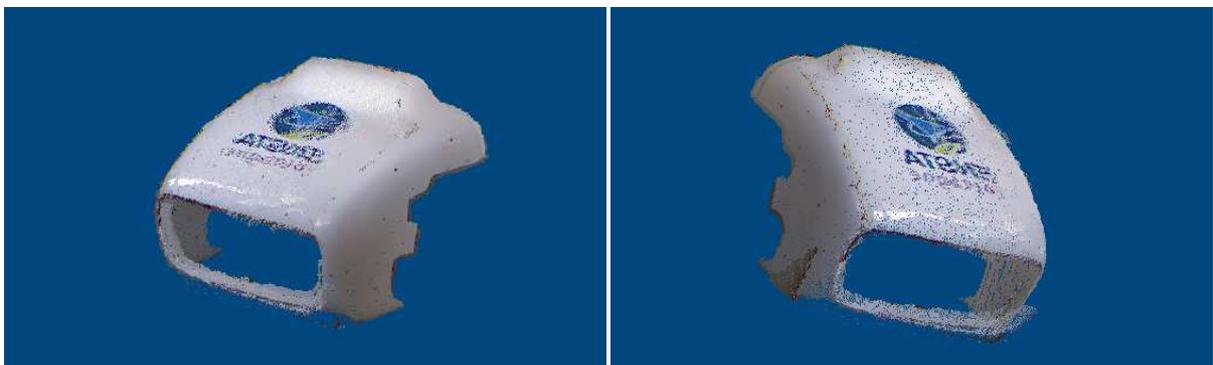


Nous avons utilisé SIFT, un algorithme de mise en correspondance de points d'intérêt dans les deux images 2D. Nous avons ensuite récupéré l'information de profondeur des points en question afin d'obtenir une liste de points 3D en correspondance dans les deux nuages de points. Enfin, nous avons écrit le système d'équation à résoudre afin d'obtenir les paramètres de transformation (rotation + translation) d'un nuage par rapport à l'autre. Pour résoudre ce système par méthodes ensemblistes, nous avons écrit un algorithme « forward-backward » contractant successivement les intervalles autour de la solution. Nous avons obtenu une largeur d'intervalle de 0.1rad pour les angles de rotation et de 0.05m pour les translations avec en moyenne un temps de calcul de 0.2ms. Voici le résultat obtenu sur notre exemple :





En supprimant les points inutiles, nous obtenons l'objet complet en 3D :



Enfin nous avons ajouté une centrale inertielle à notre système afin de connaître la rotation effectuée immédiatement, nous permettant ainsi de réduire le temps de calcul et d'améliorer la précision.



Figure 7 : Kinect montée avec une centrale inertielle