



HAL
open science

Computer vision in the context of minimally invasive surgery | Internship report

Thibaud Piccinali

► **To cite this version:**

Thibaud Piccinali. Computer vision in the context of minimally invasive surgery | Internship report. Informatique [cs]. 2024. hal-04717346

HAL Id: hal-04717346

<https://inria.hal.science/hal-04717346v1>

Submitted on 1 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



Inria

DEFRO·ST...
DEFORMATION IN ROBOTICS

Rapport de Stage

LABORATOIRE INRIA
EQUIPE DEFROST
2 OCTOBRE 2023 - 1ER AVRIL 2024
HAUTE BORNE - VILLENEUVE-D'ASCQ

Thibaud PICCINALI

Tuteur entreprise

Paul CHAILLOU

Tuteur école

Benjamin KATRYNIOK

L'élève atteste que l'ensemble du travail présenté est original et indique en fin de ce rapport chaque source utilisée.

Ce rapport a été lu par l'ensemble des tuteurs de l'élève qui en valident le contenu.

Table des matières

Remerciements	3
Introduction	4
1 Contexte du stage	5
1.1 Le groupe INRIA	5
1.2 Le centre INRIA de l'université de Lille	5
1.3 L'équipe de recherche DEFROST	6
1.3.1 Histoire et Origines	6
1.3.2 Objectifs	6
1.3.3 Un exemple de réalisation	7
1.3.4 Organisation	7
2 Contexte de travail	9
2.1 Le contexte médical	9
2.1.1 La chirurgie oncologique	9
2.1.2 La chirurgie mini-invasive	9
2.2 Le travail de Monsieur CHAILLOU	10
2.3 Le sujet de stage	12
3 Mes missions	13
3.1 Logiciel d'utilisation du SPIDERMASS	13
3.1.1 Aide au développement	13
3.1.2 Publication	13
3.2 Réalité Augmentée	14
3.2.1 Objectifs	14
3.2.2 Méthode adoptée	14
3.2.3 Matériel utilisé	15
3.2.4 Présentation de la pipepline	17
3.2.5 Organisation du code	22
3.2.6 Résultats	23
3.2.7 Limites et perspectives d'amélioration	24
3.3 Asservissement optique	25
3.3.1 Objectifs	25
3.3.2 Principe	25
3.3.3 Résultats	26
4 Ressentis	28
4.1 Ressentis du travail proposé	28
4.2 Ressentis de la vie au laboratoire	28
5 Conclusion	29
6 Lexique	30

Remerciements

Je tenais tout d'abord à remercier l'ensemble du laboratoire INRIA, et plus particulièrement l'ensemble de l'équipe DEFROST pour l'accueil chaleureux qui m'a été fait, pour les bons moments passés à leur côtés, ainsi que pour tous les conseils qu'ils ont pu me partager.

Je souhaitais particulièrement remercier mon tuteur de stage : Monsieur PAUL CHAILLOU pour son excellent accompagnement (pour ses conseils, son encadrement ou sa disponibilité) durant ces six mois de stage. Monsieur CHAILLOU est une personne très inspirante, que ce soit sur le plan professionnel ou personnel et je suis très reconnaissant d'avoir pu apprendre de lui durant cette période.

Je voulais notamment remercier Monsieur DAMIEN MARCHAL et Monsieur JEREMIE DEQUIDT pour leurs expertises techniques qu'ils ont pu apporter au projet.

J'adresse également mes remerciements à Monsieur BENJAMIN KATRYNIOK pour son implication en tant que tuteur école, mais aussi à Monsieur ALEXANDRE KRUSZEWSKI pour son aide précieuse dans ma recherche de stage.

Je réserve aussi des remerciements pour toutes les personnes qui se sont impliquées de près ou de loin dans ce stage tant sur sa recherche que sur l'écriture du rapport.

Enfin je souhaitais remercier l'Ecole Centrale de Lille, pour avoir permis de rendre ce stage possible.

Introduction

Dans le cadre d'une année de césure, je souhaitais prendre du temps supplémentaire pour découvrir plus en détail le mode de fonctionnement d'une entreprise ou d'un laboratoire. Effectuer un stage est alors apparu pour moi comme le meilleur moyen de répondre à ce besoin. Effectivement, grâce à un stage d'une durée importante, il me serait possible d'intégrer totalement une équipe et de comprendre pleinement le contexte et l'impact des missions qui me seront confiées. Ce serait aussi pour moi l'occasion de développer des compétences nouvelles : en travaillant directement avec les acteurs qui côtoient au quotidien les problématiques d'un domaine, j'allais pouvoir proposer un travail complet et profond, qui se traduirait par l'acquisition de nouvelles connaissances.

Pour ma recherche de stage j'ai choisi de creuser une piste (que j'ai pu découvrir grâce à divers projets et enseignements) de ce que pourrait être mon futur professionnel : celle de l'informatique associée au domaine de la robotique. C'est avec cette idée en tête que je me suis mis à chercher un stage dans ce domaine, et c'est tout naturellement que je me suis retrouvé à candidater pour une offre de stage au sein du laboratoire INRIA. J'ai alors fait la connaissance de Monsieur PAUL CHAILLOU qui m'a fait part de son travail en robotique souple ainsi que de son projet. Après entretien, Monsieur CHAILLOU décide de m'accepter comme stagiaire sous sa tutelle pour des missions de computer vision.

Ainsi au travers de ce document, je souhaitais faire un rapport de l'ensemble des éléments que j'ai pu apprendre au cours de ce stage, mais également de l'ensemble des missions qui m'ont été confiées. Pour mener à bien cette idée j'adopterais le plan suivant. Dans un premier temps, je tacherai d'expliquer le cadre de ce stage en vous détaillant l'organisation du laboratoire INRIA ainsi que de l'équipe de recherche DEFROST à laquelle j'étais rattaché. Suite à cela, j'introduirai le contexte de recherche dans lequel s'inscrit l'ensemble du travail que j'ai proposé. Fort de cette introduction, je détaillerais les missions sur lesquelles j'ai travaillé. J'expliquerai notamment ma démarche ainsi que les différentes solutions obtenues. Enfin je clôturerai ce rapport en donnant mon ressenti personnel sur ce stage.

1 Contexte du stage

1.1 Le groupe Inria

L'INRIA (Institut National de Recherche en Informatique et en Automatique) [1] est un centre de recherche français spécialisé en mathématiques et sciences du numérique. Sous la tutelle du ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation et du ministère de l'Économie et des Finances, cet institut fournit un travail de rang mondial (à travers différents projets internationaux) et souvent en collaboration avec des partenaires industriels.

Créée en janvier 1967, l'Inria compte aujourd'hui 220 équipes-projets regroupant plus de 3900 chercheurs et ingénieurs. Depuis 1984 le groupe est également à l'origine de la création de plus de 230 start-ups.

On compte aujourd'hui 10 centres de recherche INRIA répartis à travers la France dans de grands campus universitaires, mais également à l'étranger.

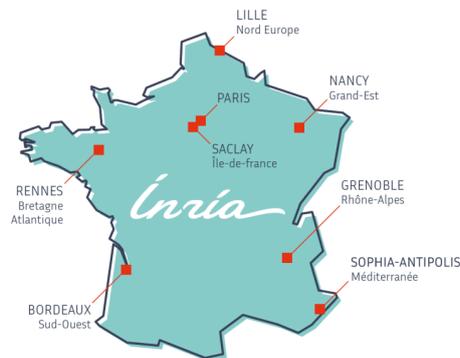


FIGURE 1 – Répartition des sites INRIA en France

Dans le cadre de ce stage j'étais rattaché au centre de l'université de Lille dont je vais désormais détailler sommairement l'organisation.

1.2 Le centre Inria de l'université de Lille

Le centre Inria de l'Université de Lille, créé en 2008, est implanté sur deux sites : à la Haute-Borne, au cœur du campus de l'université de Lille, et à EuraTechnologies, au sein de l'écosystème entrepreneurial. L'équipe projet dont je faisais partie durant ce stage était basée sur le site de la Haute Borne.



FIGURE 2 – Site Inria Haute Borne Lille

Ce centre lillois comporte 15 équipes-projets qui sont pour la plupart rattachées à des laboratoires et établissements d'enseignement supérieur locaux : l'Université de Lille, Centrale Lille, CNRS, Polytech Lille... Ces équipes regroupent plus de 365 scientifiques de nationalités variées travaillant sur des missions de sciences des données, d'intelligence artificielle, de génie logiciel, de santé numérique...

1.3 L'équipe de recherche Defrost

Au sein du site de Lille, je faisais partie de l'équipe DEFROST (DEFormable RObotic SofTware) [2]. Je souhaite ainsi faire un point un peu plus détaillé sur cette entité.

1.3.1 Histoire et Origines

Cette équipe de recherche est née de deux parents : le centre de recherche de L'INRIA de Lille et le centre de recherche CRISTAL. Elle est ainsi dépendante de plusieurs établissements de recherche mais aussi de ministères différents [3].

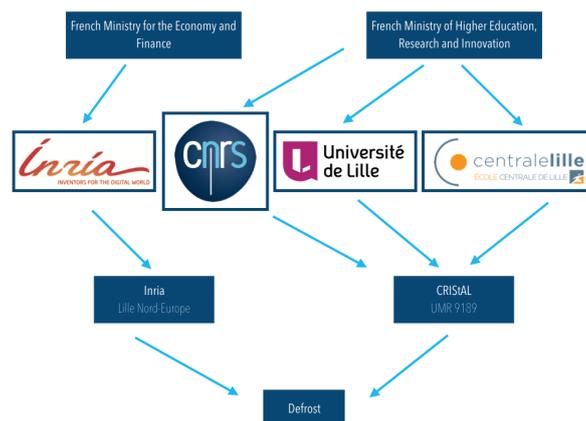


FIGURE 3 – Dépendances de l'équipe DEFROST

1.3.2 Objectifs

La principale caractéristique de cette équipe est qu'elle regroupe des personnes travaillant sur le design, le contrôle et la fabrication de robots souples.

La robotique déformable consiste à utiliser des matériaux souples pour réaliser des systèmes qui s'adaptent mieux à leur environnement, manipulent plus facilement des objets fragiles, interagissent de manière plus sûre avec les humains ou naviguent dans des espaces restreints. Ce concept trouve notamment son inspiration dans la nature, où les organismes vivants ont naturellement évolué pour adapter leur forme et leur structure à leur environnement. La conception et le contrôle de ce genre de robots représentent l'un des grands défis de ce domaine, qui demeurent largement inexplorés à ce jour. En effet, toute la difficulté réside dans le fait que le mouvement de ce type de robot est basé sur ses déformations qui sont plus complexes à calculer et à prévoir comparé à des mouvements de robots rigides. Ainsi la modélisation, qui sert notamment de base au contrôle, est plus difficile : les méthodes doivent être repensées et adaptées au cas des robots souples. De par leurs avantages comparé à des robots rigides, ce nouveau domaine

de la robotique pourrait être à l'origine de nombreuses révolutions dans les années à venir : il pourrait par exemple permettre d'amener la robotique dans des environnements plus sensibles (comme l'intérieur du corps humain) et ainsi proposer de nouvelles solutions à des problèmes existants.

1.3.3 Un exemple de réalisation

Je souhaitais également partager un exemple de projet réalisé par l'équipe DEFROST. En effet, je pense que c'est un bon moyen de comprendre le travail et les moyens mis en place par l'équipe.

Le projet, en collaboration avec l'institut national de recherche en santé l'INSERM, permet de simuler l'insertion d'un implant cochléaire [4]. Ce type d'opération est destinée à des individus sourds ou malentendants qui, grâce à cet implant, peuvent espérer retrouver une partie (ou la totalité) de leurs capacités auditives. Cependant due à la forme en spirale de la cochlée l'insertion de cet implant est souvent difficile et imprécis, ce qui peut amener à des complications lors de l'opération. C'est dans ce contexte que s'inscrit le projet : grâce à cette simulation il est désormais possible de déterminer avec plus de précision l'angle d'insertion optimal ou éventuellement un nouveau design de l'implant. Ces changements vont se traduire par de meilleurs résultats pour le patient : moins de séquelles opératoires, une audition mieux ajustée ou de meilleures connexions nerveuses avec l'implant.

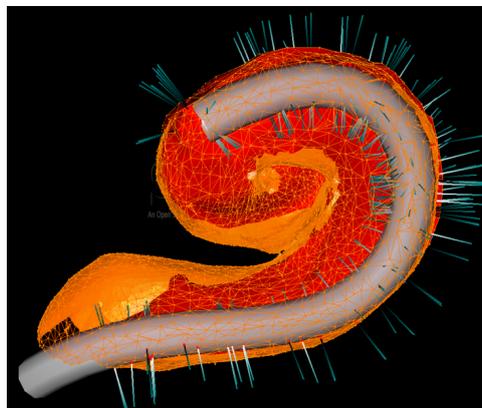


FIGURE 4 – Exemple d'une simulation d'une insertion réussie

1.3.4 Organisation

L'équipe est composée d'une vingtaine de personnes répartis entre doctorants, post-doctorants, chercheurs, et ingénieurs. On constate assez rapidement que l'organisation globale de l'équipe diffère d'une organisation plus classique, que l'on pourrait trouver par exemple en entreprise. En effet, même si on remarque la présence d'un leader (en la personne de Monsieur CHRISTIAN DURIEZ) il n'y a pas de forme de hiérarchie au sein du groupe. Les doctorants travaillent principalement seuls sur leurs projets (même s'ils échangent et s'entraident régulièrement entre eux) et seule une minorité de l'équipe travaille ensemble au quotidien. Cependant, même si ces derniers sont plutôt autonomes sur leurs projets, leurs travaux s'inscrivent toujours dans une démarche collective (projet européen, collaboration avec d'autres laboratoires ...) selon un contrat établi en amont.

Deux temps importants sont néanmoins à noter en ce qui concerne la vie de l'équipe. Le premier est celui du *Team meeting* le mardi après-midi. Au cours de ce rendez vous hebdomadaire une personne de l'équipe (définie selon une planification préalable) présente les avancées de son travail, ou tout autre sujet scientifique qu'il/elle juge bénéfique à l'équipe. Ces *Team meeting* sont de bons moyens pour chaque membre de l'équipe de se tenir au courant des travaux en cours mais également des avancées de la recherche à l'extérieur au laboratoire. Le deuxième temps est celui de la *Coding session* le jeudi après midi. Ces sessions sont l'occasion de se focaliser sur la partie programmation en recherche : amélioration de la qualité des logiciels, rédaction de documentation, apprentissage de nouvelles notions, résolutions de bugs... Elles sont dirigées par les membres les plus expérimentés en programmation de l'équipe.

2 Contexte de travail

Dans cette partie, je souhaitais détailler le contexte dans lequel s'inscrit l'intégralité des missions qui m'ont été confiées au cours de ce stage. En effet, l'ensemble de mes missions avaient comme point commun le domaine médical, et plus particulièrement le domaine de la chirurgie.

Puisque mon stage se place dans le même cadre que la thèse de Monsieur CHAILLOU je vais dans un premier temps détailler le contexte médical de son travail ainsi que son sujet de thèse avant de développer mon sujet de stage.

2.1 Le contexte médical

2.1.1 La chirurgie oncologique

Aujourd'hui le traitement d'un cancer dans l'abdomen (foie, estomac, intestin...) présente des difficultés majeures.

Habituellement, on traite ce type de cancer par chirurgie oncologique [5] : après ouverture de la paroi du péritoine du patient, le chirurgien vient retirer localement la ou les tumeurs ainsi qu'une partie des tissus qui l'entourent (exérèse). Ce type d'opération intervient généralement après d'autres traitements (chimiothérapie ou radiothérapie par exemple). Pourtant, même si ce genre d'opération représente le pilier majeur du traitement de cancer aujourd'hui, il n'en reste pas moins très impactant et incertain pour le patient. D'abord parce qu'il s'agit d'une opération longue qui nécessite un long temps de rétablissement (entre deux et une dizaine de jours) marqué par des effets secondaires potentiels (douleurs, saignements, gonflements...), mais aussi parce que ce type de traitement est plutôt imprécis.

En effet, lors de l'opération, le chirurgien retire la tumeur ainsi que la zone péri-tumorale (zone avec des cellules saines et cancéreuses) pour éviter que des cellules cancéreuses restantes ne puissent provoquer une rechute ou des métastases. Cependant, l'extraction de cette zone péri-tumorale peut être compliquée : d'abord, parce que la délimitation de cette zone est imprécise (identification faite à l'œil par le chirurgien) mais aussi parce qu'il est nécessaire d'analyser les tissus retirés. En effet, en réalisant des tests immunohistochimiques sur les parois des tissus prélevés il est possible de contrôler la qualité des marges qui ont été choisies. Cette étape, bien que obligatoire, se déroule en plein milieu de l'opération, elle est lente (entre 30 et 40 minutes), imprécise et donc impactante pour le patient.

Au final, de part les difficultés présentées précédemment, il est difficile d'estimer la position des cellules cancéreuses. De fait, pour éviter les rechutes et les métastases, le chirurgien se retrouve souvent contraint à prendre des marges de sécurité importantes ce qui peut être inutilement impactant pour le patient. Il est cependant toujours possible que ces marges se retrouvent à être quand même insuffisantes pour retirer totalement les cellules cancéreuses.

2.1.2 La chirurgie mini-invasive

La chirurgie mini-invasive (ou coelioscopie / laparoscopie) est un type d'opération qui consiste à effectuer des petites incisions sur la paroi abdominale (de l'ordre du centimètre) pour permettre le passage d'instruments médicaux et ainsi accéder au site chirurgical. En

gonflant l'abdomen avec du gaz (CO_2), le chirurgien dispose de l'espace nécessaire pour procéder à l'opération.

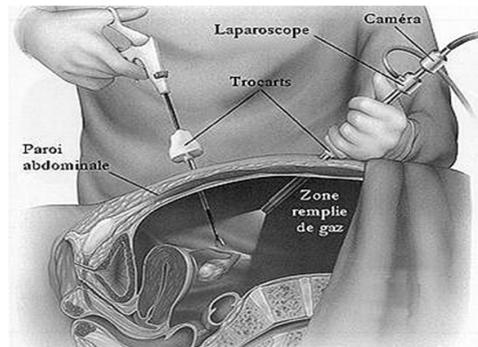


FIGURE 5 – Schema d'une opération de chirurgie mini-invasive

De par ses nombreux avantages (sécurité, rétablissement rapide...) ce type de chirurgie représente une alternative intéressante vis à vis d'une chirurgie traditionnelle notamment en ce qui concerne le traitement de cancer. Elle permet notamment aux patients de rentrer chez eux le jour même, alors qu'en chirurgie ouverte il faut compter trois ou quatre jours d'hospitalisation minimum.

On peut cependant noter une prise en main et un apprentissage difficile pour maîtriser ce type d'opération.

2.2 Le travail de Monsieur Chaillou

La thèse de mon tuteur s'inscrit donc dans ce contexte chirurgical. Cette dernière repose sur la technologie du SPIDERMASS. Le SPIDERMASS est une technique de spectrométrie de masse qui permet d'analyser la composition d'un tissu. Pour se faire, on utilise un laser qui lors de l'impact sur une surface permet de vaporiser quelques molécules qui composent cette dernière. Ces molécules sont ensuite aspirées et amenées au spectromètre qui, couplé à un algorithme de machine learning, est capable d'analyser en temps réel la nature de la surface étudiée (saine, cancéreuse, nécrosée...).

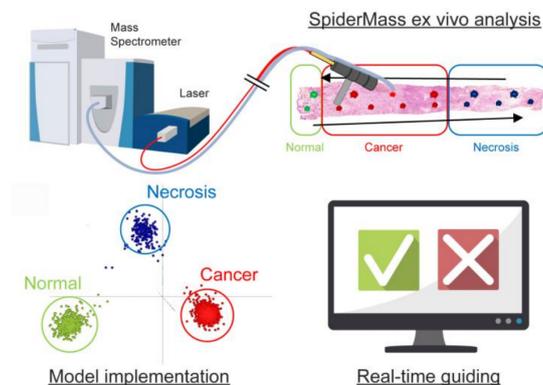


FIGURE 6 – Principe de fonctionnement du SPIDERMASS

L'idée serait donc d'utiliser cette technologie lors d'opération de chirurgie oncologique. On pourrait de cette façon améliorer grandement le déroulement de ce type de traitement,

notamment en ce qui concerne la délimitation de la zone péri-tumorale et de sa marge d'exérèse. L'objectif est donc de porter la technologie proposée par le SPIDERMASS en in-vivo. A cette fin, Monsieur CHAILLOU a décidé de travailler avec un nouveau type de robot adapté à la réalisation de cet objectif [6].

Un robot souple apparaît comme une piste de départ intéressante. En effet, sa souplesse permettrait des contacts intrinsèquement sûrs et son mouvement induit par les déformations lui permettrait une grande dextérité et une capacité à se contorsionner pour atteindre des zones difficiles d'accès. Ainsi, ce type de robot pourrait opérer efficacement et en limitant au maximum les risques dans l'abdomen d'un patient.

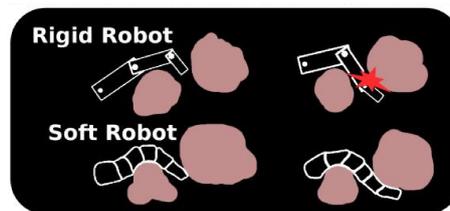


FIGURE 7 – Illustration des avantages d'un robot souple par rapport à un robot rigide

Pour répondre à ces exigences (et aux contraintes du problème) mon tuteur a donc décidé d'utiliser un robot souple. Il s'est alors tourné vers un des collaborateurs de l'INRIA, UCL (University College London) qui proposait un modèle prometteur [7].

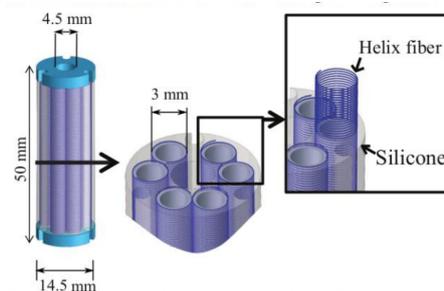


FIGURE 8 – Schéma du robot souple utilisé

Ce robot, actionné par air comprimé est composé de fibres circulaires qui en contraignant les déformations latérales permettent au robot de se déformer pour réaliser une courbure ou une elongation. Grâce à une simulation informatique il est possible de le contrôler en position [8].

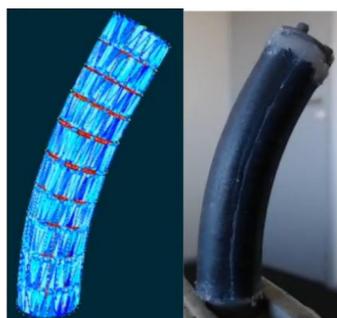


FIGURE 9 – Exemple d'un contrôle en position

L'objectif final serait donc d'utiliser ce genre de robot pour réaliser une analyse complète de l'abdomen d'un patient. Le robot, accompagné des outils pour le spectromètre (laser et tube aspirant), viendrait balayer l'ensemble de la zone d'opération en y effectuant un tir laser et en collectant les particules s'y échappant. Ensuite, grâce à la technologie de discrimination du SPIDERMASS il serait possible de déterminer avec précision l'état des différentes molécules de la surface étudiée. Une carte 3D, présentant les localisations des identifications, pourrait ainsi être générée et utilisée par le chirurgien pour réaliser une opération de meilleure qualité (notamment en ce qui concerne la délimitation de la zone péritumorale). Également, grâce à ses faibles dimensions, ce genre d'opération pourrait être effectué en coelioscopie ce qui réduirait les séquelles post-opératoires.

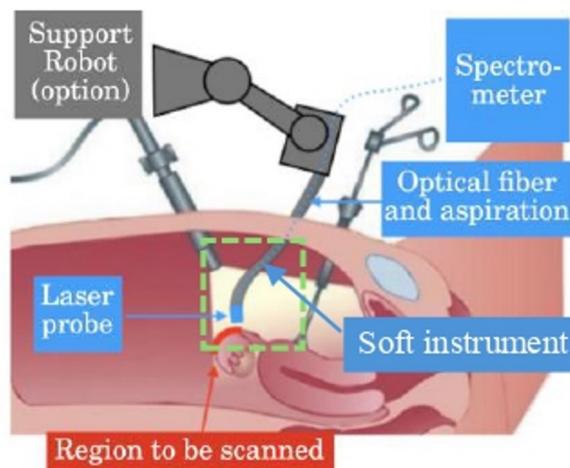


FIGURE 10 – Illustration du processus envisagé

2.3 Le sujet de stage

Mon sujet de stage s'inscrit donc dans ce contexte médical. Plus particulièrement, dans la mise en place de cette nouvelle technologie de diagnostic, il se situe dans la partie d'imagerie médicale. En effet, la mission principale qui m'a été confiée était la réalisation d'un algorithme de réalité augmentée (RA) qui permettait de superposer un modèle 3D à la vue d'une caméra. Ce programme avait pour objectif de projeter la carte 3D générée par le SPIDERMASS sur la vue d'une caméra présente lors de l'opération et de faire des retours en temps réel pour que le chirurgien puisse prendre des décisions capitales : celles des définitions des marges d'exérèse.

A noter que d'autres missions m'ont également été attribuées, mais ces dernières représentent qu'une minorité du travail fourni au cours de ce stage. Elles seront cependant présentées et expliquées dans la partie suivante.

3 Mes missions

Au cours de ce stage j'ai pu participer à plusieurs missions qui s'inscrivent toutes dans le cadre de la thèse de Monsieur CHAILLOU. J'ai travaillé sur trois sujets principaux, que je vais désormais vous détailler dans cette partie.

3.1 Logiciel d'utilisation du SpiderMass

La première mission dont je vais vous parler ne représentait qu'une faible partie de mon stage, je vais donc vous la présenter de manière brève. Il s'agit de ma contribution au développement du logiciel d'utilisation du SPIDERMASS.

Pour utiliser le SPIDERMASS et réaliser les opérations que j'ai pu décrire dans la partie précédente, Monsieur CHAILLOU a développé un logiciel permettant de synchroniser l'ensemble des outils qui sont utilisés lors de la création d'images moléculaires 3D (bras robotique, capteur de profondeur, tube aspirant...). Cependant, même si le programme était fonctionnel, son utilisation commençait à devenir limitée. En effet, ayant été développé il y a plusieurs années et avec de mauvaises pratiques de programmation, il devenait difficile de corriger le code en cas de dysfonctionnement et même, pour un nouvel utilisateur, de s'en servir. Face à ce problème, mais aussi face à la volonté de partager son travail avec d'autres laboratoires de recherches, mon tuteur de stage a donc pris la décision de consacrer une partie de son temps de thèse à corriger, améliorer et nettoyer le code qu'il avait proposé. C'est ainsi que Monsieur CHAILLOU me proposa de se joindre à lui dans cette tâche.

3.1.1 Aide au développement

J'ai ainsi participé à l'amélioration du logiciel, sur des missions de correction de bugs visuels, d'export de données ou d'amélioration de performances. Au vu de la complexité technique et de l'intérêt que je trouve limité pour un rapport de stage, je ne présenterais pas davantage les correctifs que j'ai pu mettre en place pour ce logiciel. J'aimerais cependant vous détailler la méthode que j'ai adoptée.

En effet, de part la complexité du logiciel (complexité double entre l'organisation du code et le contexte scientifique), mes connaissances limitées dans le langage de développement (MatLab) et le peu de temps que je pouvais consacrer au projet (puisqu'il ne représentait qu'une minorité du stage), j'ai dû adopter une méthodologie particulière. J'ai choisi de travailler en duo avec mon maître de stage : ce dernier m'expliquait à l'oral le fonctionnement global du logiciel (ses composantes, leurs rôles, l'organisation du code ...), ensuite je tachais de trouver une solution théorique, algorithmique aux problèmes relevés pour enfin aborder plus efficacement la partie programmation que je développais avec des tests et ajustements progressifs. Cette méthode m'a permis de rapidement mettre en place des solutions efficaces et optimisées.

3.1.2 Publication

Comme je l'expliquais dans la partie précédente, cette amélioration du logiciel était également motivée par l'envie de partager le programme à d'autres laboratoires. Pour rendre son utilisation plus intuitive à ses futurs usagers, l'UX du logiciel a donc été retravaillé. C'est un point où je me suis également impliqué : ayant assez peu utilisé le

logiciel (comparé à Monsieur CHAILLOU ou à d'autres usagers du laboratoire), j'ai pu proposer un regard neutre sur l'organisation de l'interface et y suggérer des améliorations en ce qui concerne son utilisation. Le logiciel est désormais beaucoup plus facilement compréhensible pour un nouvel usager, notamment en ce qui concerne sa prise en main.

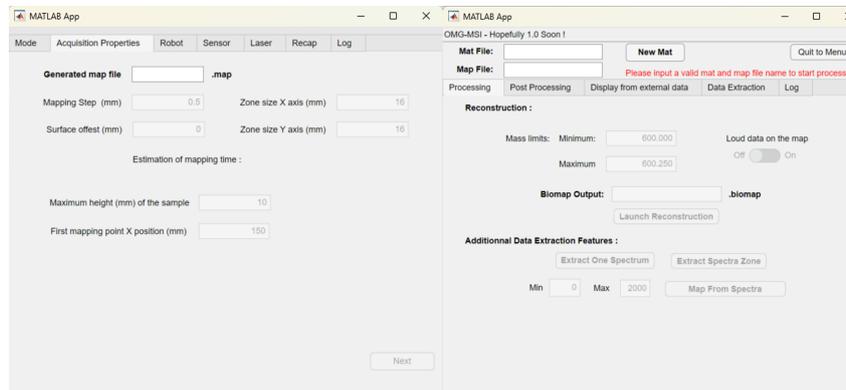


FIGURE 11 – Exemples d'écrans de l'interface du logiciel

Pour accompagner le partage de ce logiciel, la publication d'un papier scientifique a également été prévue. De par mon implication dans le développement du programme et dans son utilisation j'ai pu participer à l'écriture de ce dernier. J'ai pu faire des retours concernant l'organisation du document et j'ai notamment pu rédiger une partie entière (sur l'algorithme de réalité augmentée et son lien avec le SPIDERMASS). Ce papier scientifique n'est pas encore totalement finalisé au moment où j'écris ces lignes, mais il devrait être soumis dans les prochains mois.

3.2 Réalité Augmentée

La deuxième mission que je vais vous présenter concerne le développement d'un algorithme de réalité augmentée. Comme expliqué précédemment, cette mission représente le cœur du stage.

3.2.1 Objectifs

L'objectif que l'on souhaitait atteindre était de proposer un programme capable de superposer un modèle 3D, qui pouvait être généré par le SPIDERMASS, à la vue d'une caméra. L'idée était de développer un programme capable de fonctionner sur deux cas : un premier avec uniquement des images et un second qui fonctionnerait en temps réel sur flux vidéo. A court terme, l'objectif était de proposer un algorithme capable de s'intégrer dans un laboratoire pour y faire des démonstrations et preuves de concept. Enfin, il était également question de proposer un code propre et facilement réutilisable dans l'optique d'une participation et d'un partage à la communauté Open Source, qui fournit une grande partie des outils sur lesquels nous nous basons.

3.2.2 Méthode adoptée

Pour mener à bien ces objectifs j'ai procédé en plusieurs étapes. Je me suis d'abord concentré sur le traitement d'images.

Dans un premier temps, j'ai repris le travail de Madame Tinhinane SMAIL [9]. Madame SMAIL était la précédente stagiaire qui travaillait avant moi sur le projet, et même si son programme ne fonctionnait pas dans sa totalité il représentait une base solide de travail. J'ai donc passé la première partie de mon stage à comprendre ce qui avait été fait et à me familiariser avec les contraintes, les outils et méthodes utilisés. Notamment, pour réussir à prendre en main le code et les différents formats de données, j'ai d'abord effectué des tâches simples comme de la correction de bugs ou de l'implémentation de fonctionnalités mineures.

Fort de cette première prise en main, j'ai pu par la suite commencer à développer des fonctionnalités plus complexes. En procédant par cycles itératifs de type méthode agile (idée, implémentation, test, correction) et soutenu par mon maître de stage, j'ai pu arriver à une première version du programme. Cette dernière était encore très éloignée d'une version finale, notamment parce qu'elle contenait encore beaucoup de dysfonctionnements et de limites d'utilisation, mais en répétant ces étapes plusieurs fois j'ai pu me rapprocher peu à peu d'une version qui correspondait à nos exigences.

Une fois cette dernière développée, j'ai consacré un temps important à la mise en forme du code. L'idée était de faire en sorte qu'il soit plus facilement réutilisable : cela comprenait des étapes de debug, de découpage de fonctions, d'écriture de documentations... Ce travail, bien qu'assez chronophage, m'a permis d'être beaucoup plus efficace dans la suite du stage.

La dernière étape, une fois le développement du programme pour le traitement d'images réalisé, était d'étendre l'application de l'algorithme à des flux vidéos. Pour ce faire, l'ensemble des fonctions développées ont dû être retravaillées pour minimiser leur temps d'exécution et ainsi pouvoir prétendre à des résultats en temps réel. Avec une méthode similaire à celle suivie lors du développement de la version pour les images, j'ai pu proposer une solution fonctionnelle dans ce cas ci.

3.2.3 Matériel utilisé

Au cours du développement, différents éléments ont été utilisés : il me paraît important de vous les présenter avant de rentrer dans le détail du fonctionnement des algorithmes.

Les premiers étaient des modèles et objets 3D de tests. Ces derniers, bien que n'ayant aucun lien avec le contexte médical du projet font office de preuve de concept, et m'ont permis de réaliser les différents essais du programme : c'est grâce à ces derniers que j'ai pu développer le code, et réaliser les démonstrations que je vous présenterais dans cette partie. J'ai ainsi travaillé avec deux modèles principaux : un représentant une fleur de lys et un autre représentant un estomac. Pour chacun d'entre eux un modèle avec des couleurs a également été créé afin de pouvoir estimer la qualité des superpositions.

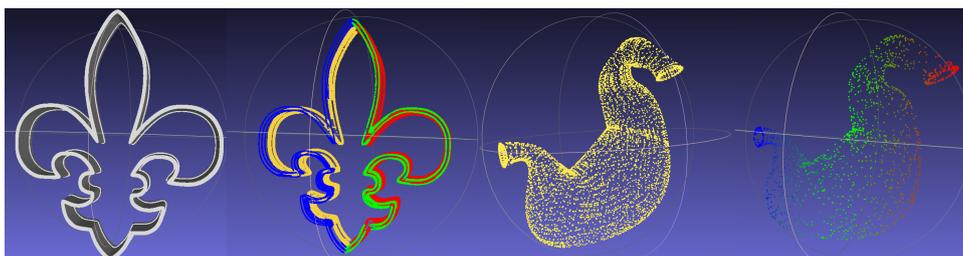


FIGURE 12 – Modèles 3D utilisés

Ces modèles ont ainsi été imprimés en 3D grâce aux imprimantes 3D du laboratoire.



FIGURE 13 – Objets 3D utilisés

J'ai également pu travailler avec des échantillons de foie de bœuf et de chair de poulet. Ces derniers m'ont permis de faire des tests qui se rapprochaient beaucoup plus du contexte médical du projet.

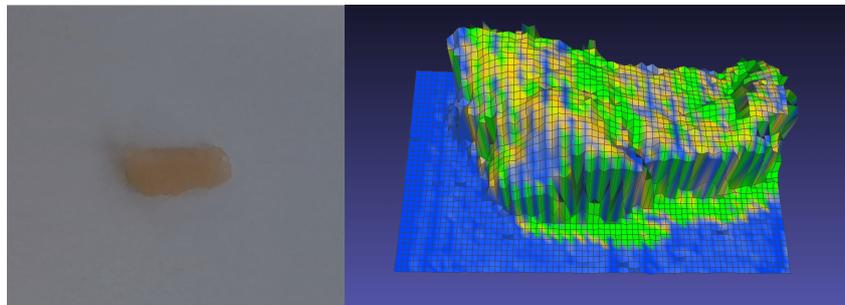


FIGURE 14 – Chair de poulet et son modèle 3D (les couleurs représentent des informations biologiques)

Les modèles 3D de ces échantillons ont été réalisés grâce au SPIDERMASS, selon le principe que j'ai pu décrire dans la partie précédente.

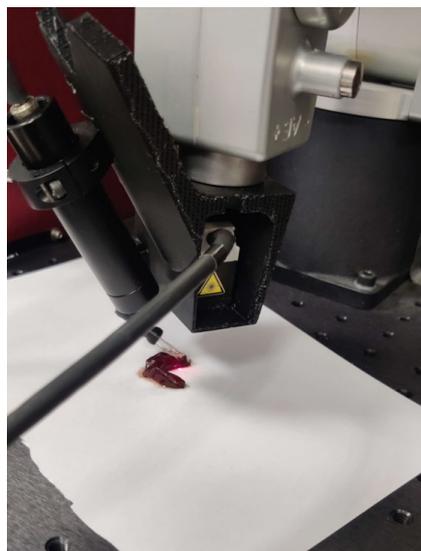


FIGURE 15 – SPIDERMASS qui réalise une cartographie d'un échantillon de foie de boeuf

Le dernier outil qui a été utilisé au cours de ce stage est la caméra qui a servi pour faire l'acquisition des images 3D. Cette dernière est une caméra de profondeur de chez INTEL : la Realsense D405 [10].



FIGURE 16 – Realsense D405

Ce type de caméras permettent de capturer un nuage 3D d'une scène, et donc en plus de fournir une image 2D elles peuvent également fournir son image 3D associée. Une des caractéristiques majeures de ce type de caméra est qu'elle fournit un lien entre ces deux types d'image (2D et 3D) : il est possible d'alterner entre vue 2D et vue 3D facilement. Plus particulièrement, chaque point du nuage correspond à un pixel unique dans l'image et vice versa.

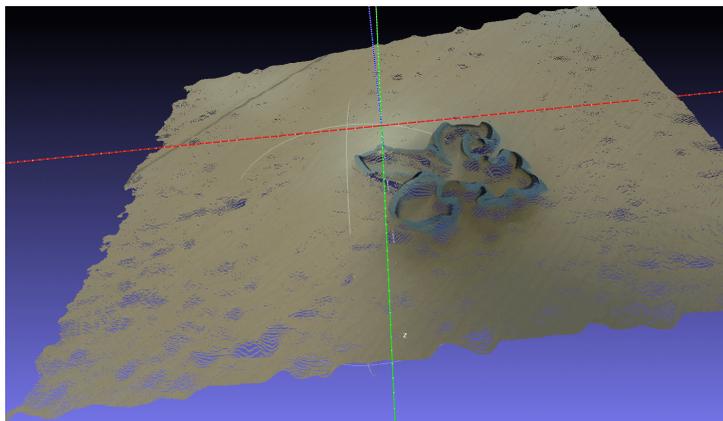


FIGURE 17 – Exemple d'acquisition avec la caméra Realsense D405

La capacité à capturer des nuages de points 3D représente le cœur de notre programme qui se distingue des algorithmes de réalité augmentée existants (qui se basent sur l'identification et le suivi de points d'intérêts dans une image 2D) en proposant un version qui se base sur la correspondances de formes 3D.

3.2.4 Présentation de la pipeline

L'algorithme développé fonctionne comme une succession de différentes étapes de traitement qui permettent d'arriver à un résultat final. Je vais ainsi dans cette partie vous présenter ces étapes.

Acquisition : La première est celle de l'acquisition du nuage de points grâce à la caméra Realsense. Comme introduite dans la partie précédente, cette caméra permet de récupérer les points 3D d'une scène ainsi que ses couleurs associées.



FIGURE 18 – Exemple d’acquisition

Extraction de zone : Suite à cette première étape, on procède à une sélection de la zone d’intérêt grâce à une interface. Cette dernière permet de conserver dans le nuage de point uniquement les points de la zone sélectionnée.

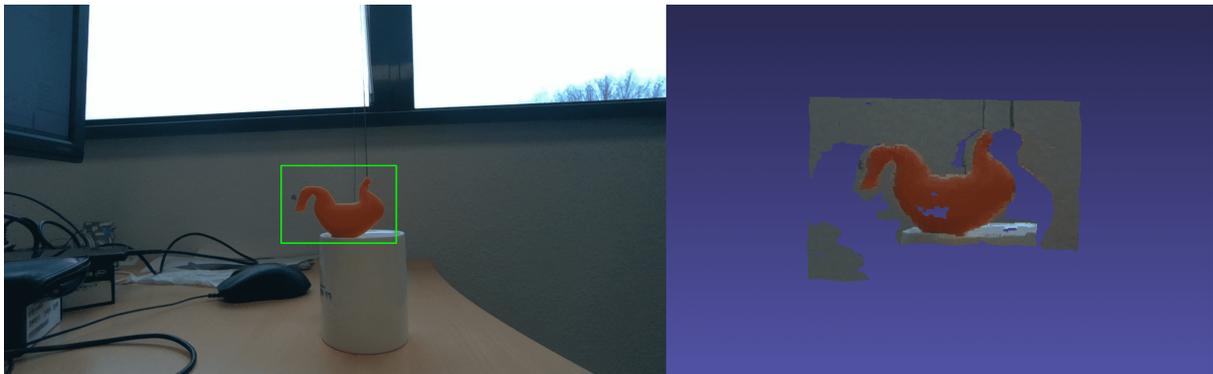


FIGURE 19 – Selection de la zone d’intérêt

Filtrage HSV : L’étape suivante est une étape de filtrage par masque HSV. Un masque HSV (Hue, Saturation, Value) permet de créer un filtre qui sélectionne les pixels d’une image en fonction de leur teinte, saturation et valeur. Ce type de masque représente une bonne alternative aux masques RGB puisque ces derniers ne se basent pas uniquement sur les couleurs : un masque HSV peut ainsi permettre de sélectionner un objet précis même dans une scène colorée.



FIGURE 20 – Modèle 3D filtré avec un masque HSV

Filtrage bruits : Pour compléter cette étape de filtrage, une seconde, basée sur la distance des points par rapport au centre de masse du nuage, est réalisée. Cette dernière permet de supprimer les éventuels bruits parasites pour ne garder uniquement les points correspondants à l'objet.

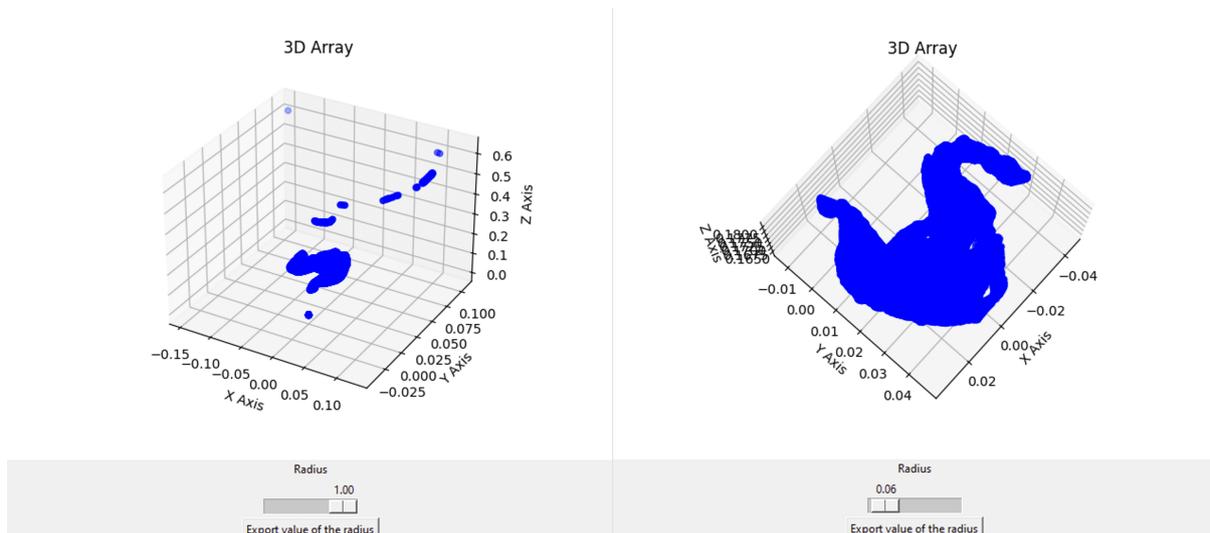


FIGURE 21 – Interface qui présente la détection et la correction des bruits parasites

Redimensionnement : Ensuite, on redimensionne le modèle 3D pour que sa taille corresponde avec celle de l'objet capturé par la caméra. Au début de la conception de l'algorithme cette étape s'effectuait avec un seuil d'agrandissement fixe mais au vu de la qualité moyenne des résultats, elle s'effectue désormais de manière automatique en calculant les extrêmes de chaque ensemble de points et en y déduisant un facteur d'agrandissement.

Calcul de la matrice de translation : Une fois les deux formes à la même taille, on les superpose en translation grâce un algorithme qui calcule le point milieu de chaque ensemble de points.

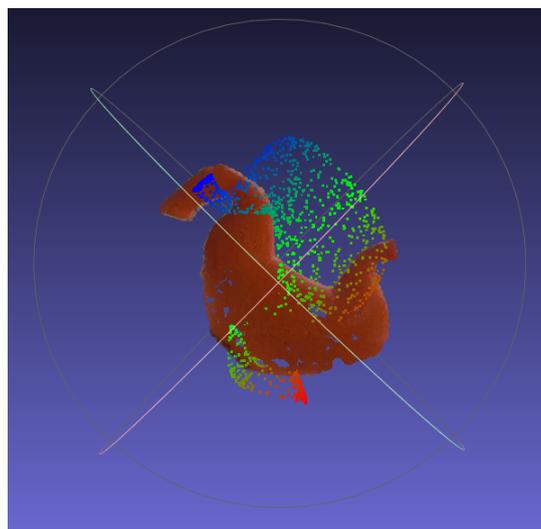


FIGURE 22 – Modèle 3D après recalage en translation

Calcul de la matrice de rotation : On passe ensuite aux calculs pour déterminer la

rotation qui permet de superposer au mieux la forme à son modèle. Pour cela, on procède en deux temps : un premier qui permet de calculer une approximation de la valeur de rotation, et un deuxième qui permet d'affiner cette valeur. La première étape d'approximation s'effectue en parcourant une liste d'angles pré-définis et en calculant quelle valeur dans cette liste permet de superposer au mieux les deux fichiers 3D. L'affinage de cette valeur se fait ensuite à l'aide d'un algorithme ICP (Iterative Closest Points) [11]. L'algorithme d'ICP permet de trouver la rotation qui minimise la distance entre deux ensembles de points de manière itérative en calculant un coût pour chaque configuration.

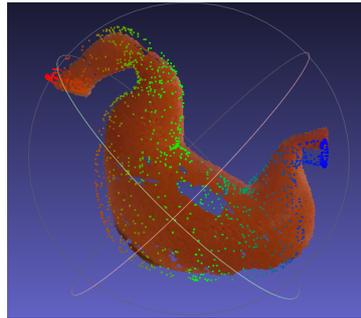


FIGURE 23 – Modèle 3D recalage en rotation

Projection et affichage : L'étape finale est celle de l'affichage de notre modèle 3D sur la vue 2D de la caméra. Différentes méthodes d'affichage ont été développées mais toutes se basent sur le calcul point par point de la nouvelle position du modèle 3D : c'est ce que l'on appelle faire une estimation de pose. Pour effectuer ce traitement, on utilise le fait que l'ensemble des résultats des calculs de transformation (translation et rotations) peuvent se représenter sous forme matricielle. En effet, une matrice de translation et une matrice de rotation peuvent s'écrire sous la forme suivante :

$$M_t = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ avec } t_i \text{ la valeur de translation sur l'axe } i$$

$$M_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \left\{ \begin{array}{l} \text{avec } \theta \text{ la valeur de rotation autour de l'axe } x \\ \text{(forme similaire pour les autres axes)} \end{array} \right.$$

Grâce à ce genre de représentation matricielle on peut facilement calculer la nouvelle position d'un point après une transformation :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \left\{ \begin{array}{l} \text{avec } M \in M_4(\mathbb{R}) \text{ une matrice de transformation} \\ \text{avec } (x, y, z) \text{ les coordonnées du point avant transformation} \\ \text{avec } (x', y', z') \text{ les coordonnées du point après transformation} \end{array} \right.$$

Ainsi en suivant ce principe, on peut calculer la position du modèle 3D. Comme je l'expliquais, trois modes d'affichage ont été développés.

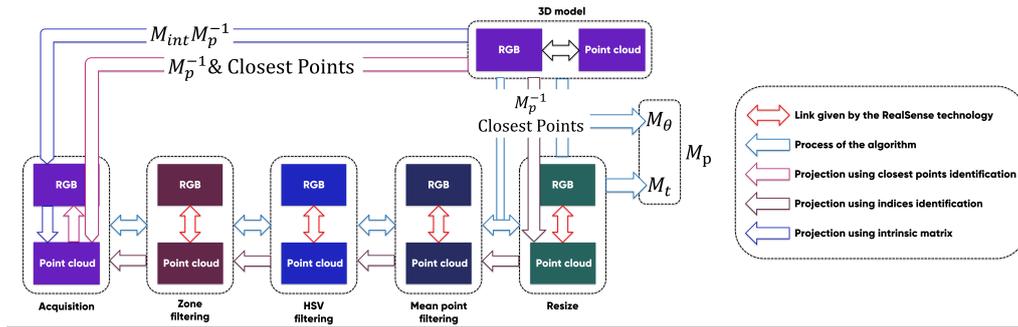


FIGURE 24 – Schéma illustrant les différents modes de projections

Le premier (représenté par les flèches roses sur la figure 24) consiste à calculer la nouvelle position de chaque point du modèle 3D et à chercher (à l'aide d'un algorithme de Closest Points) dans le nuage initial (celui capturé par la caméra) le point le plus proche correspondant. Le deuxième (représenté par les flèches bordeaux) reprend le même principe sauf qu'au lieu de chercher la correspondance de points dans le nuage initial on la cherche dans l'ensemble de points qui constitue la forme 3D puis on fait l'affichage dans la vue originale de la caméra. On prend pour cela, bien soin de conserver en mémoire la position de cet ensemble dans le nuage de point initial (doubles flèches bleues claires). Enfin, le dernier mode d'affichage (représenté par les flèches bleues foncées) utilise la matrice intrinsèque de la caméra. Ce genre de matrice contient l'ensemble des propriétés intrinsèques, telles que la focale, le décalage du centre de l'image et la distorsion optique. Elle s'écrit de la manière suivante :

$$M_{int} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{cases} \text{avec } f_x \text{ et } f_y \text{ les focales en pixels.} \\ \text{avec } (c_x, c_y) \text{ le décalage du centre de l'image en pixels.} \end{cases}$$

Avec cette matrice on peut alors directement calculer la projection du modèle 3D dans la vue 2D de la caméra :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M_{int} * M_p^{-1} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \begin{cases} \text{avec } M_{int} \text{ la matrice intrinsèque de la caméra} \\ \text{avec } M_p \text{ la matrice d'estimation de pose} \\ \text{avec } (x, y, z) \text{ les coordonnées du point avant transformation} \\ \text{avec } (x', y', z') \text{ les coordonnées du point après transformation} \end{cases}$$

Portage sur flux vidéo : Enfin, en ce qui concerne le fonctionnement de l'algorithme sous flux vidéo, on utilise le processus précédemment décrit pour traiter chaque image qui compose notre flux. Quelques légères modifications ont cependant été apportées au programme notamment pour améliorer la rapidité d'exécution. En effet, pour l'étape de détermination des rotations du modèle 3D, on effectue tout d'abord un calcul complet pour la première image du flux et, pour toutes les suivantes, on considère que l'objet n'a pas pu bouger de plus que de quelques degrés : cette considération limite ainsi le nombre de calculs à effectuer et permet donc d'augmenter la fréquence de traitement des images.

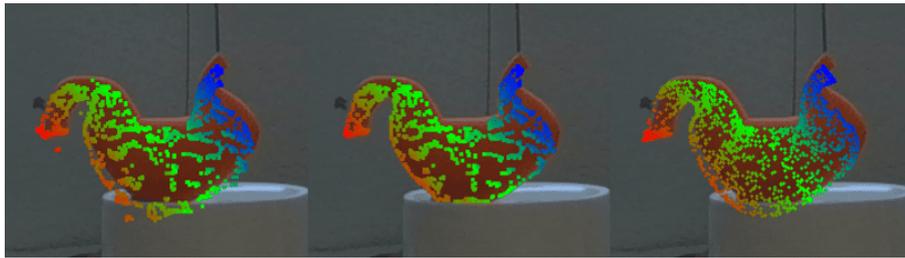


FIGURE 25 – Les trois modes de projection de l'exemple (zoomé)

3.2.5 Organisation du code

Comme expliqué en introduction de cette partie, l'un des objectifs de ce projet, en plus de proposer une solution fonctionnelle, était que cette dernière soit développée selon des bonnes pratiques de programmation, bien documentée et facilement réutilisable. Ainsi un temps important a été consacré au partage du code.

Nous avons choisi de partager le logiciel comme logiciel Open Source : il est accessible librement depuis le dépôt GITHUB de l'équipe DEFROST. Plus précisément nous l'avons publié avec une licence LGPL qui permet à qui que ce soit de l'utiliser et de le modifier librement. Ainsi n'importe quel futur usager (notamment les membres de l'équipe) pourront facilement reprendre le travail fourni.

Nous avons également implémenté plusieurs éléments pour que ce logiciel soit facilement utilisable par de nouveaux usagers. En effet, sous les conseils de Monsieur DAMIEN MARCHAL (expert en développement logiciel de l'équipe), nous avons choisi d'orienter le développement vers un code modulaire, c'est à dire un code où chaque fonction peut être prise indépendamment des autres pour éventuellement être utilisée seule ou remplacée. Ce type de programmation permet également de faciliter le développement de nouvelles fonctionnalités qui peuvent être ensuite partagées à toute la communauté du projet. Enfin, pour que la prise en main du logiciel soit la plus facile, le code suit la norme PEP8 et possède de nombreux éléments de documentation (type hints, docstrings, ...) ainsi que des fichiers de test.

Je vais désormais, pour conclure cette partie, vous présenter rapidement la structure du code. Ce dernier est organisé selon deux parties principales.



FIGURE 26 – L'organisation du code de la réalité augmenté

La première, comprend l'ensemble des fonctions de traitement des nuages de points et de pilotage des caméras RealSense, et la seconde contient les fonctions propres à l'algorithme

de réalité augmentée (fonctions d’affichage, de calcul matriciel,...). Il est important de noter que la première partie représente un module à part entière : elle a été construite dans l’optique de pouvoir être utilisée de manière indépendante aux codes relatifs à la réalité augmentée. Ce module est lui-même organisé en sous fichiers relatifs aux types de données utilisées (par exemple au traitement de nuage de points, d’images, de tableaux,...). Cette structure permet de favoriser la modularité du code et donc la possibilité de remplacer ou d’utiliser individuellement les fonctions qui ont été développées.

Ainsi ce n’est pas un, mais deux dépôts GITHUB qui ont été publiés : le premier pour les fonctions de traitements des nuages de points¹, et un second pour l’algorithme de réalité augmentée².

3.2.6 Résultats

Nous sommes globalement très satisfaits des résultats obtenus. En effet, comme vous avez pu le voir sur la figure 25 les modèles sont correctement projetés.

Nous avons également pu effectuer d’autres tests concluants avec d’autres modèles. Notamment, nous avons pu réaliser des tests satisfaisants sur des échantillons de chair de poulet.

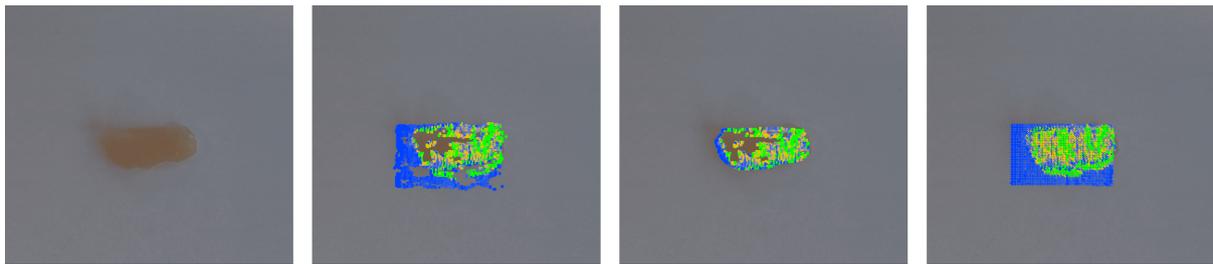


FIGURE 27 – Projection 2D sur un échantillon de chair de poulet

Ces projections, il est aussi possible de les observer en vue 3D grâce au lien (fourni par la caméra) entre image 2D et 3D que j’évoquais précédemment. Ainsi des objets 3D ont été produits.

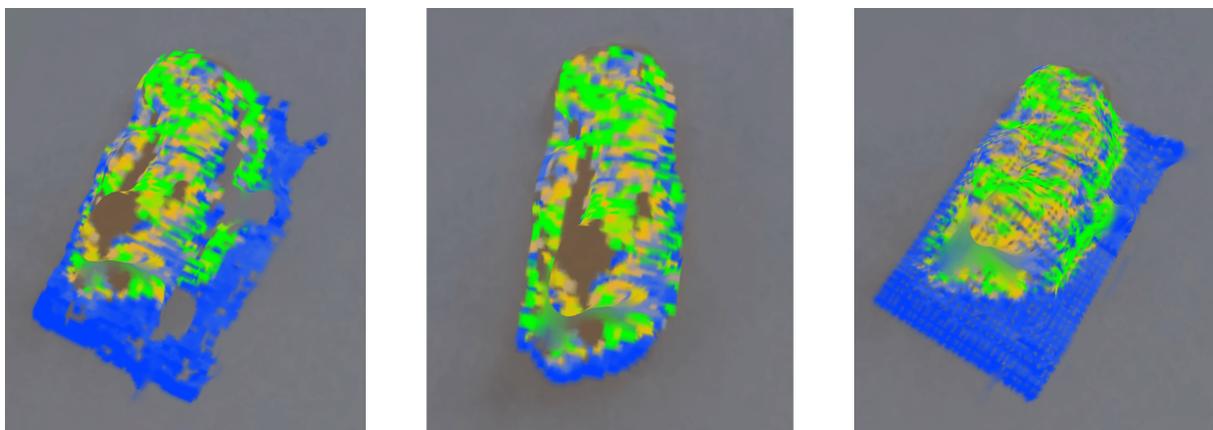


FIGURE 28 – Projection 3D sur un échantillon de chair de poulet

1. https://github.com/SofaDefrost/Python_3D_Toolbox_for_Realsense

2. https://github.com/SofaDefrost/Shape-Based_Augmented_Reality

Les tests effectués sur des échantillons de chair de poulet constituent une preuve de concept de la capacité de notre solution à être appliquée à des images biologiques similaires à celles qui seraient utilisées en chirurgie. Ce type d'objets 3D pourraient être directement utiles pour le chirurgien pour avoir une meilleure compréhension de la position et du type de chaque cellule.

Des résultats en flux vidéo ont également été produits. A ce sujet, une vidéo de démonstration mettant en scène plusieurs configurations a été publiée sur la chaîne Youtube de l'équipe³.

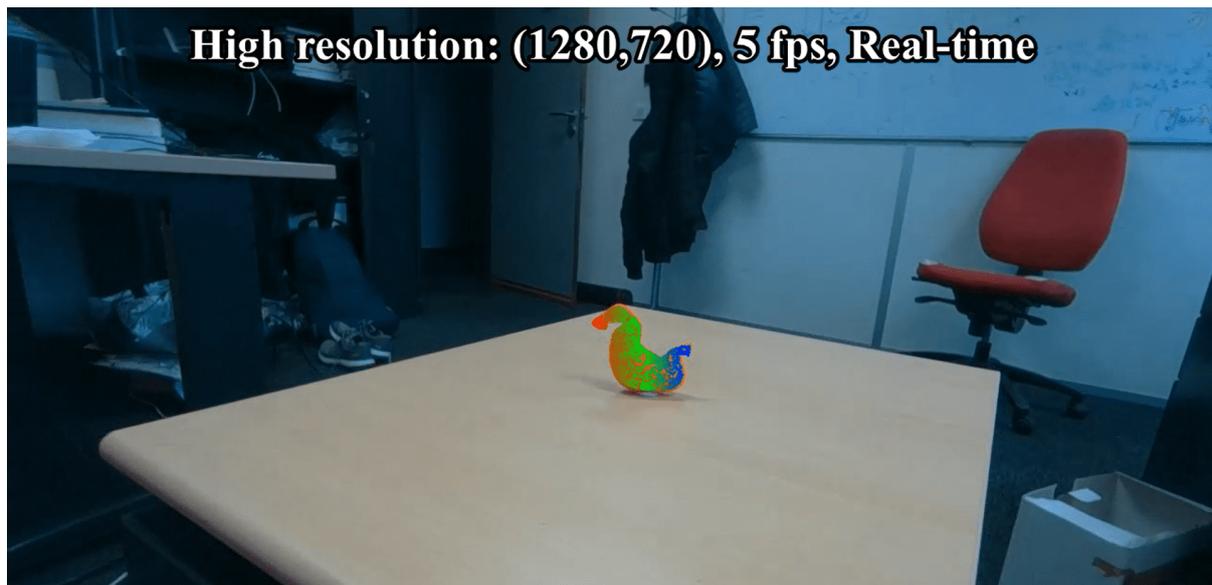


FIGURE 29 – Extrait de la vidéo de démonstration disponible sur la chaîne youtube de DEFROST

Sur cette vidéo vous pouvez constater de l'optimisation du traitement de données qui permet d'avoir des résultats en haute qualité et en temps réel (5 fps). De plus, en fin de vidéo, vous pouvez assister à une démonstration en haute définition et à haute fréquence (30 fps). Cette démonstration finale a été réalisée en post production : chaque image (2D et 3D) qui compose la vidéo a été enregistrée en amont puis traitée dans un deuxième temps. Pour être capable de stocker plusieurs nuages de points en même temps, et parce qu'aucune solution n'existait pour répondre à ce besoin, nous avons décidé de créer notre propre format de données (le *.mply*).

3.2.7 Limites et perspectives d'amélioration

Même si nous sommes globalement très satisfaits des résultats de l'algorithme, il est important de noter ses limites :

- le programme peut être imprécis sur la valeur de rotation à effectuer, ce qui peut mener à de légers décalages.
- l'étape de filtrage par masque HSV est assez sensible aux variations de luminosité, ce qui peut induire des difficultés pour détecter l'objet (notamment dans le cas vidéo).

3. https://www.youtube.com/watch?v=ntkH_wG3x9o

- les vidéos réalisées en post-production peuvent prendre un temps important à être générées (30 secondes de vidéo peuvent prendre une heure à être générées).

Pour palier à ces problèmes, plusieurs pistes d'améliorations peuvent être envisagées :

- Développer une autre implémentation de l'algorithme d'ICP, ce qui permettrait d'améliorer la précision du programme pour les calculs d'orientations.
- Améliorer le procédé de détection d'objet dans le nuage de point (modèle de deep learning, algorithme de segmentation...). Une piste qui avait été envisagé était d'utiliser le modèle SAM (Segmentation Anything Model) de Meta pour réaliser la détection d'objet, mais devant les résultats limités (notamment en ce qui concerne le temps de calcul) vis à vis de notre problème cette piste a été abandonnée [12].

3.3 Asservissement optique

La dernière mission que je vais vous présenter concerne l'asservissement en position du robot. Cette mission représente la dernière charge de travail majeure qui m'a été confiée lors de ce stage.

3.3.1 Objectifs

L'objectif était de pouvoir asservir en position le robot de Monsieur CHAILLOU et ainsi réaliser une preuve de concept de la capacité à contrôler ce dernier pour atteindre n'importe quelle position. L'idée était d'avoir deux versions : une première qui permettrait au robot de se positionner sur une cible fixe et une seconde sur une cible mouvante.

3.3.2 Principe

Nous avons utilisé un laser (positionné sur la tête du robot) dont la projection représentait la position réelle du robot, une caméra RealSense (positionné de manière fixe) qui permettait de déterminer la position du laser dans l'espace et donc de connaître la position mesurée (théorique) du robot, ainsi qu'une simulation mathématique qui contrôlait la pression des valves pour déplacer le robot. L'idée était donc d'utiliser un régulateur PID [8] pour contrôler la position du robot mesurée à l'aide de la caméra.

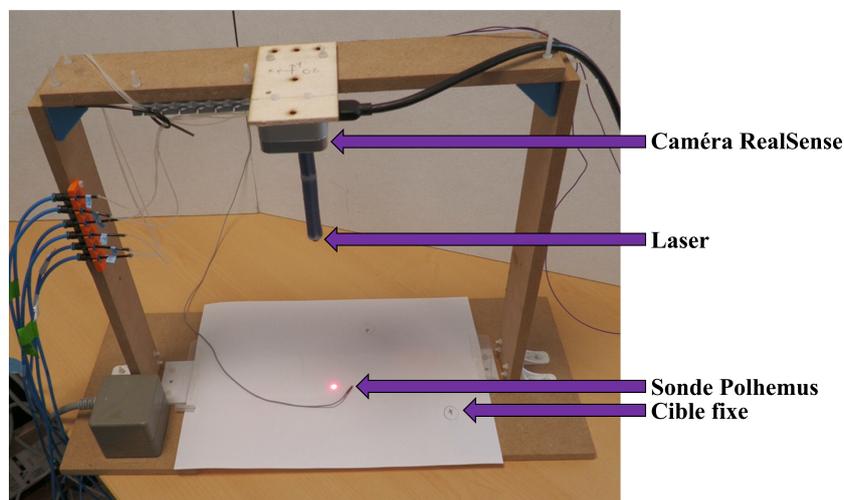


FIGURE 30 – Schéma de l'organisation du prototype

Dans ce processus j'étais chargé de détecter le laser. Pour ce faire j'ai pu ré-utiliser les fonctions que j'avais développées, notamment celles permettant d'appliquer un masque HSV à une capture. En effet, puisque la luminosité de la scène ne changeait pas, il était possible de détecter de manière précise et régulière la position de ce point laser à l'aide de ce type de masque. Enfin, même si j'ai pu participer à l'optimisation des paramètres du régulateur, le reste de l'asservissement (calcul de l'erreur, calculs des consignes...) à été réalisé par mon tuteur de stage.

3.3.3 Résultats

Les résultats obtenus sont plutôt concluants. En effet, on arrive avec cette méthode à asservir le robot selon une cible fixe.

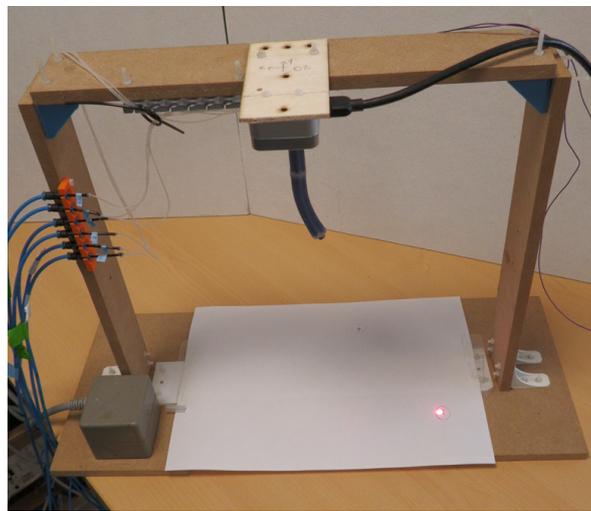


FIGURE 31 – Asservissement du robot avec cible fixe

On constate notamment que même en présence de perturbations extérieures (ici représenté par un câble gênant) le robot parvient à corriger ces dernières et à atteindre la cible.

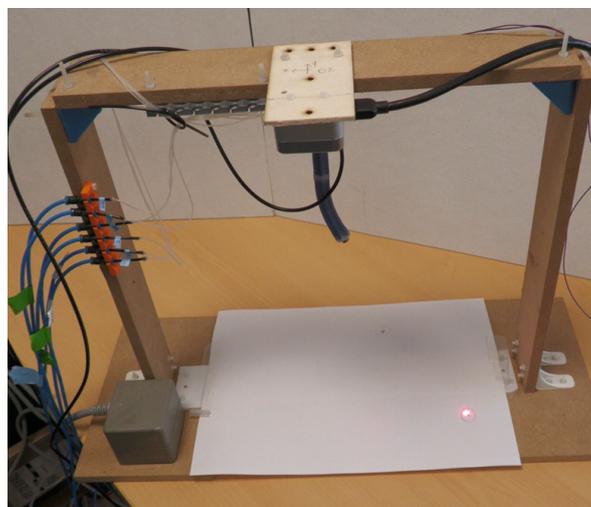


FIGURE 32 – Asservissement du robot avec cible fixe et perturbations

Enfin pour une cible mouvante, représentée par la sonde du Polhemus (système qui utilise des capteurs magnétiques pour suivre avec précision la position et l'orientation d'objets en temps réel), grâce à de légères modifications de code on parvient à proposer une solution fonctionnelle :

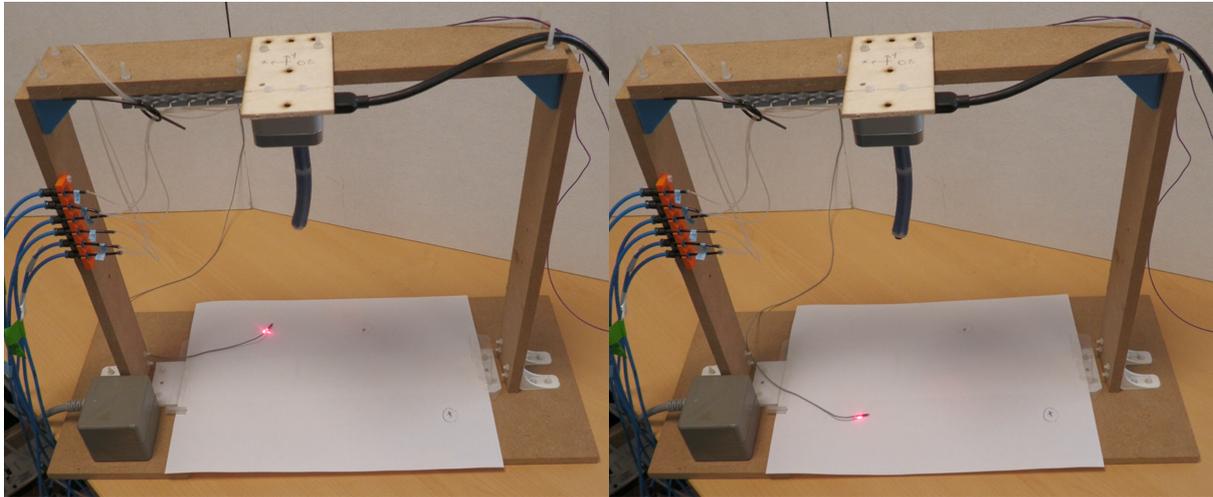


FIGURE 33 – Asservissement du robot avec cible mouvante

4 Ressentis

4.1 Ressentis du travail proposé

Je suis globalement très satisfait du travail que j'ai proposé : les solutions développées fonctionnent bien et sont, je pense, à l'image de l'attitude que j'ai adoptée durant ce stage. Je suis notamment fier de la qualité des résultats de démonstration (vidéo et papier scientifique) qui mettent bien en valeur le travail. Je suis d'autant plus satisfait que des personnes de l'équipe commencent déjà à utiliser mon travail dans leur projet et à faire des retours !

J'ai énormément apprécié travailler sur ces thématiques. En effet le contexte médical dans lequel s'inscrit l'ensemble des missions rend le travail très stimulant : on a envie de s'impliquer et de proposer une solution de qualité. Également, de part la multitudes des tâches il est possible d'organiser son travail librement ce qui permet notamment d'alterner entre les différents lots de travail et ainsi éviter une forme de redondance.

4.2 Ressentis de la vie au laboratoire

J'ai, de part plusieurs aspects, vraiment apprécié mes six mois de stage au laboratoire. Premièrement parce que j'ai trouvé très stimulant de pouvoir travailler avec des personnes talentueuses et reconnues dans leur domaine. J'ai notamment pu profiter d'excellents conseils, que ce soit sur mon travail ou mes idées de projets professionnels. Deuxièmement parce que j'ai aimé l'ambiance globale : l'équipe est dynamique, bienveillante et il y règne une bonne ambiance de groupe. Dernièrement, parce que le laboratoire possède plusieurs avantages dans la vie quotidienne (ajustements des horaires de travail, grande liberté sur les congés...) et personnelle (nombreux avantages et réductions pour des sorties culturelles et sportives...).

5 Conclusion

Ce stage a été très enrichissant pour moi et je pense qu'il tiendra un rôle majeur dans ma prise de décision de ce que sera mon avenir professionnel. J'ai notamment pu répondre à plusieurs questions que je me posais sur le monde de la recherche et de son fonctionnement. Plusieurs aspects m'ont séduit et ont fait naître en moi une envie de poursuivre dans ce domaine.

Je suis convaincu que cette expérience sera un réel atout dans ma future carrière professionnelle, parce qu'elle m'a permis de développer plusieurs de compétences (programmation, computer vision, robotique...) et méthodologies précieuses.

Fort de ces réponses et ces compétences acquises je considère ce stage comme une excellente première partie de césure, et je suis désormais confiant pour aborder la deuxième partie de cette année.

6 Lexique

Bug : défaut de conception d'un programme informatique à l'origine d'un dysfonctionnement.

Computer Vision : branche de l'informatique dont le but est d'analyser et de traiter une ou plusieurs images.

Deep Learning : sous-catégorie de l'apprentissage automatique (Machine Learning) qui implique l'utilisation de réseaux de neurones artificiels composés de plusieurs couches (d'où le terme *profond* ou *deep* en anglais).

Docstring : documentation rédigée par le développeur pour détailler le fonctionnement d'un segment de code.

Exérèse : terme médical qui se réfère à l'action de retirer chirurgicalement une partie ou la totalité d'un tissu, d'un organe ou d'une lésion anormale du corps.

FPS : acronyme de *Frames Per Second* (*Images Par Seconde* en français) est une mesure qui indique le nombre d'images individuelles affichées par un dispositif d'affichage (comme un écran ou un moniteur) chaque seconde.

Git/Github/GitLab : système de contrôle de version, permettant de suivre les modifications apportées à un code source au fil du temps. GitHub et GitLab sont des plateformes web qui utilisent Git pour l'hébergement de projets logiciels.

LGPL : acronyme de *Lesser General Public License*, est une licence de logiciel libre. Elle est utilisée pour distribuer des logiciels et permet généralement à quiconque de voir, modifier et distribuer le code source du logiciel librement.

MatLab : environnement de programmation et langage de programmation utilisé principalement dans le domaine du calcul numérique et de l'analyse technique.

Metastase : cellule cancéreuse qui s'est propagée d'un site primaire (où le cancer a commencé) vers un autre site dans le corps.

Open Source : terme qui désigne les logiciels dont le code source est disponible pour le grand public. Cela signifie que quiconque peut consulter, modifier et redistribuer le code source du logiciel sans avoir besoin de demander la permission du créateur ou de payer des frais de licence.

PEP8 : acronyme de *Python Enhancement Proposal*, est un ensemble de directives de style de code pour le langage de programmation Python. Il propose notamment des conventions pour écrire du code de manière claire, lisible et cohérente.

PID (régulateur) : dispositif de contrôle automatique utilisé dans de nombreux systèmes pour maintenir une variable à une valeur cible en ajustant un processus de sortie. PID signifie Proportionnel-Intégral-Dérivé, qui sont les trois termes fondamentaux utilisés dans le calcul du contrôle PID.

Pipeline : terme qui désigne une séquence d'opérations ou de processus interconnectés, dans lesquels la sortie d'une étape devient l'entrée de la suivante. Chaque étape du pipeline traite ou transforme les données d'une manière spécifique, contribuant ainsi à l'ensemble du processus.

Python : langage de programmation populaire et polyvalent, largement utilisé dans divers domaines tels que l'analyse de données, l'intelligence artificielle, l'automatisation de tâches, le développement de jeux, et bien d'autres encore.

RGB : acronyme de *Red Green Blue*, est un système de représentation des couleurs qui est largement utilisé dans la conception graphique, la photographie, la télévision, les écrans d'ordinateur et de nombreux autres domaines.

Typehints : fonctionnalité Python qui permet aux développeurs de spécifier le type des variables, des arguments de fonction et des valeurs de retour de fonction dans leur code. Cela ne modifie pas le comportement du programme, mais fournit des informations supplémentaires sur les types de données que les fonctions attendent et renvoient.

UX : acronyme de *User eXperience*, désigne la manière dont les utilisateurs interagissent avec un produit ou un service, ainsi que leur ressenti lors de cette interaction. Il s'agit d'un domaine important dans la conception de produits numériques tels que les sites web, les applications mobiles, les logiciels, etc.

Références

- [1] Accueil | Inria.
- [2] DEFROST – Deformable Robotic Software.
- [3] DEFROST / Welcome To Defrost · GitLab, December 2023.
- [4] Olivier Goury, Yann Nguyen, Renato Torres, Jeremie Dequidt, and Christian Duriez. Numerical Simulation of Cochlear-Implant Surgery : Towards Patient-Specific Planning. volume 9900, page pp 500, October 2016.
- [5] M. V. Studio. Traitements d’un cancer.
- [6] Paul Chaillou, Nina Ogrinc, Jialei Shi, Helge Wurdemann, Christian Duriez, and Isabelle Fournier. *Poster JRDA (Journée Régionale Doctorant en Automatique) 2022*.
- [7] STIFFness controllable Flexible and Learn-able manipulator for surgical OPERations | STIFF-FLOP Project | Fact Sheet | FP7.
- [8] Paul Chaillou, Jialei Shi, Alexandre Kruszewski, Isabelle Fournier, Helge A. Wurdemann, and Christian Duriez. Reduced finite element modelling and closed-loop control of pneumatic-driven soft continuum robots. In *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 1–8, April 2023. ISSN : 2769-4534.
- [9] Tinhinane SMAIL. *Prototypage d’un système de vision dans le contexte de chirurgie mini-invasive*.
- [10] Depth Camera D405.
- [11] Daniel Lopez, Carl Haas, and Sriram Narasimhan. Specific object finding in point clouds based on semantic segmentation and iterative closest point. *Automation in Construction*, 156 :105116, December 2023.
- [12] Peng Zhang and Yaping Wang. Segment Anything Model for Brain Tumor Segmentation, September 2023. arXiv :2309.08434 [cs, eess].