Laroche-Ribert Damian

Mémoire technique

Techniques de calibration extrinsèque

22/08/2024

Table des matières

[Présentation de l’entreprise 3](#_Toc175832664)

[Ressources Humaines et Organisation 3](#_Toc175832665)

[Plateformes de Recherche et Transfert 3](#_Toc175832666)

[Domaines de Recherche 3](#_Toc175832667)

[Projets de Recherche 3](#_Toc175832668)

[Reconnaissance et Collaboration 3](#_Toc175832669)

[Résumé 4](#_Toc175832670)

[Présentation du Sujet de Mémoire Technique 4](#_Toc175832671)

[Introduction aux Méthodes de Calibration "Eye-in-Hand" et "Eye-to-Hand" 5](#_Toc175832672)

[Calibration "Eye-in-Hand" : 6](#_Toc175832673)

[Préparation : 7](#_Toc175832674)

[Expérimentation : 8](#_Toc175832675)

[Analyse des résultats : 9](#_Toc175832676)

[Calibration "Eye-to-Hand" : 9](#_Toc175832677)

[Préparation : 11](#_Toc175832678)

[Expérimentation : 11](#_Toc175832679)

[Création des datasets 12](#_Toc175832680)

[Extraction des données 12](#_Toc175832681)

[Calcul de la transformation entre la caméra et la base du robot : 14](#_Toc175832682)

[Pistes d’améliorations : 18](#_Toc175832683)

[Bilan : 19](#_Toc175832684)

[Conclusion 19](#_Toc175832685)

[Annexes 20](#_Toc175832686)

[Glossaire 20](#_Toc175832687)

# Présentation de l’entreprise

## Ressources Humaines et Organisation

CESI LINEACT, avec ses 149 membres, est composé d'un personnel diversifié et qualifié, incluant des enseignants-chercheurs, des ingénieurs de recherche, des techniciens, des doctorants, et du personnel de gestion. Il compte 7 Directeurs de Recherche, 1 Directeur de Recherche Émérite, 74 enseignants-chercheurs (dont 16 HDR), 46 doctorants (dont 3 internationaux co-encadrés), 12 ingénieurs de recherche, 1 technicien, 2 cadres administratifs et financiers, et 1 assistante. Cette diversité de compétences favorise un environnement de recherche riche et multidisciplinaire.

## Plateformes de Recherche et Transfert

Le laboratoire gère trois plateformes de recherche et de transfert : deux consacrées à l’Usine du Futur situées à Rouen et Nanterre, et une dédiée au Bâtiment du Futur à Nanterre. Ces plateformes jouent un rôle crucial dans le soutien des travaux de recherche et l'application pratique des découvertes et innovations.

## Domaines de Recherche

CESI LINEACT est structuré autour de deux équipes scientifiques principales : "Apprendre et Innover" et "Ingénierie et Outils Numériques". Ces équipes se concentrent sur deux domaines applicatifs principaux, l'« Industrie du Futur » et la « Ville du Futur », reflétant l'engagement du laboratoire à anticiper et accompagner les mutations technologiques dans ces secteurs.

## Projets de Recherche

Les projets de CESI LINEACT se répartissent en plusieurs catégories :

Projets d'Observation : Ces projets portent sur les métiers enseignés par CESI ou ses partenaires, comme la pédagogie de l'alternance.

Projets Scientifiques et Technologiques : Ces projets visent à produire des connaissances et savoirs dans les domaines scientifiques et technologiques des formations de CESI, incluant la conception de plateaux techniques.

Projets d'Intégration Technologique : Ces projets se concentrent sur l'intégration ou l'intensification de nouvelles technologies dans les programmes et diplômes de CESI.

## Reconnaissance et Collaboration

En 2018, le laboratoire a été intégré dans le collège doctoral HESAM et dans l'école doctorale ED 432 SMI, et en 2019, il a obtenu le label EA/UR 7527 suite à une évaluation positive du HCERES. CESI LINEACT a renforcé son positionnement en tant qu’acteur important du monde de l’enseignement supérieur et de la recherche. Il développe activement des partenariats au niveau international et consolide ses collaborations avec les partenaires industriels, notamment dans les filières et les pôles de compétitivité.

## Résumé

CESI représente un milieu dynamique et innovant, combinant recherche, enseignement, et application pratique. Son personnel diversifié, ses plateformes spécialisées, et ses projets de recherche variés en font un acteur essentiel dans les domaines de l'industrie et du BTP, ainsi que dans la formation des futurs professionnels de ces secteurs

Présentation du Sujet de Mémoire Technique  
Mon mémoire technique s'inscrit dans le cadre des activités du laboratoire CESI LINEACT et se concentre sur l'étude et la mise en œuvre des méthodes de calibration robotique, avec un accent particulier sur les calibrations "Eye-in-Hand" et "Eye-to-Hand". Ces techniques sont fondamentales pour le laboratoire, car elles sont cruciales pour optimiser la précision et l'efficacité des systèmes robotiques. Ces techniques sont applicables à de nombreux domaines notamment la robotique, le multi-point de vue et l’undustrie.

Au cours de ce projet, mes missions ont consisté à explorer en profondeur les méthodes de calibration utilisées en robotique, à développer et tester des algorithmes spécifiques pour améliorer la précision des calibrations, et à mettre en œuvre des solutions techniques permettant de réduire les erreurs de positionnement des robots. J'ai été amené à :

1. **Étudier les méthodes de calibration existantes** : J'ai commencé par une revue approfondie des méthodes de calibration robotique, en me concentrant sur les techniques "Eye-in-Hand" et "Eye-to-Hand", afin de comprendre leurs principes théoriques et leur application pratique.
2. **Développer un algorithme afin de calibre un système** : J'ai développé et implémenté des algorithmes de calibration, en utilisant notamment des techniques d'optimisation pour minimiser les erreurs de reprojection et les écarts 3D. Cette étape a impliqué une compréhension approfondie des transformations géométriques et des changements de repère, essentiels en robotique.
3. **Tester et valider les solutions** : J'ai réalisé des tests rigoureux pour évaluer l'efficacité des algorithmes développés, en les appliquant à des scénarios réels de calibration robotique. Ces tests m'ont permis d'affiner les méthodes utilisées et de proposer des améliorations basées sur les résultats obtenus.
4. **Optimiser les performances** : Pour répondre aux exigences de précision et de rapidité dans un contexte industriel, j'ai travaillé sur l'optimisation du code pour réduire les temps de traitement et d'exécution, ce qui est crucial pour une application en production.
5. **Proposer des pistes d'amélioration** : Enfin, j'ai réfléchi à des améliorations potentielles.

# **Introduction aux Méthodes de Calibration "Eye-in-Hand" et "Eye-to-Hand"**

La calibration extrinsèque d'un système robotique peut être réalisée selon deux méthodes principales : "Eye-in-Hand" et "Eye-to-Hand". La principale différence entre ces deux approches réside dans la position de la caméra par rapport au robot, et chacune est adaptée à des scénarios spécifiques en fonction de la nature des tâches à accomplir.

Dans le cas de la calibration **"Eye-in-Hand"**, la caméra est montée directement sur l'outil ou la pince du robot, ce qui lui permet de se déplacer en synchronisation avec les mouvements du robot. Cette configuration offre une interaction plus précise et dynamique avec les objets situés dans l'environnement proche du robot. La caméra, en étant mobile, peut ajuster sa position et son orientation en temps réel, ce qui est particulièrement utile pour les applications qui nécessitent une grande précision lors de mouvements complexes, comme l'assemblage, la manipulation délicate, ou l'inspection de proximité. Le robot peut ainsi suivre un objet avec exactitude, en tenant compte de ses changements de position ou de ses caractéristiques.

À l'inverse, la calibration **"Eye-to-Hand"** utilise une caméra fixe, souvent placée à proximité du robot mais indépendante de ses mouvements. Ici, le robot doit se déplacer dans l'espace en fonction de la vue statique fournie par la caméra. Ce type de calibration est plus adapté aux tâches qui nécessitent une vue d'ensemble de la zone de travail, comme la palettisation ou le tri d'objets dispersés sur une large surface. Le système "Eye-to-Hand" est particulièrement avantageux dans les environnements où le robot doit interagir avec plusieurs objets dans un espace étendu, car la caméra peut offrir une vision globale sans être limitée par les mouvements du robot.

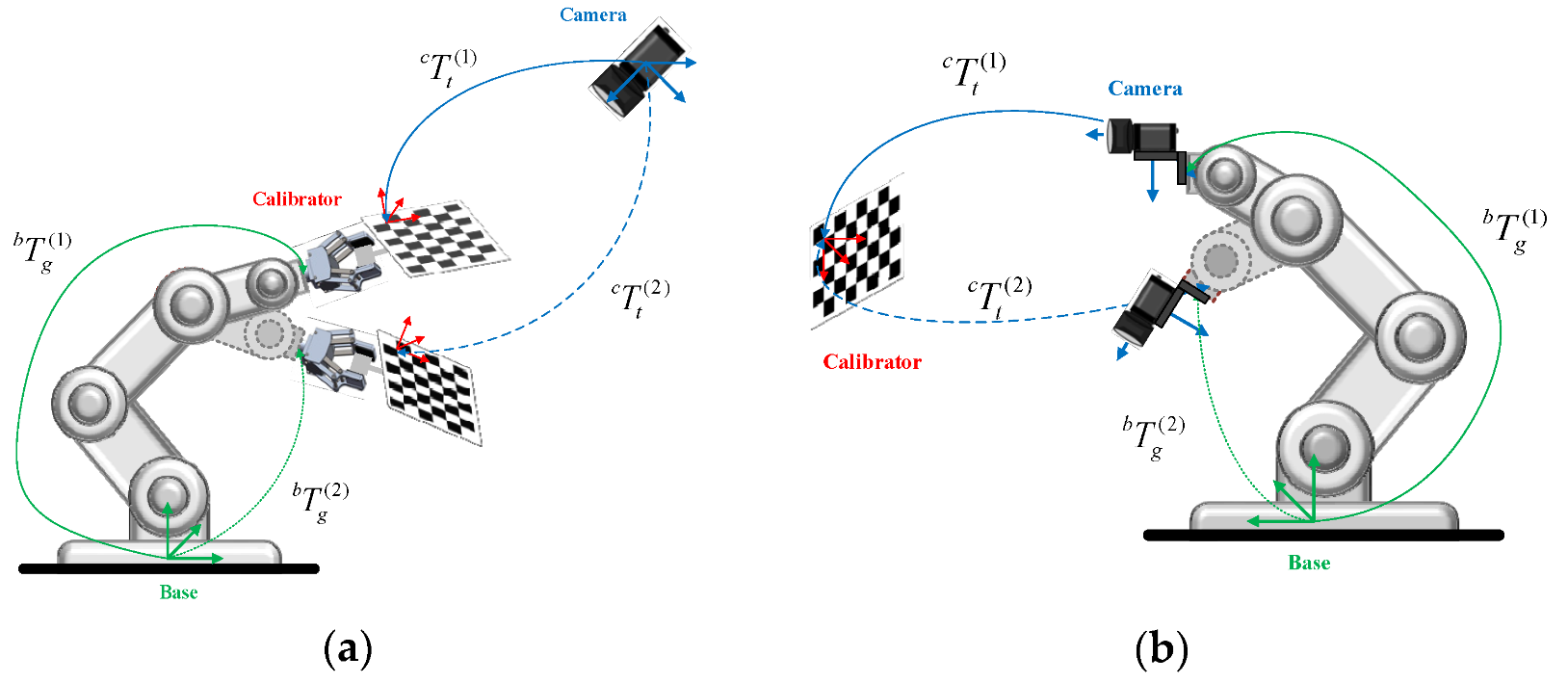
L'image ci-dessous illustre ces deux configurations de manière visuelle. La figure (b) représente un scénario "Eye-in-Hand", où la caméra est fixée sur la pince du robot, tandis que la figure (a) montre un exemple "Eye-to-Hand", avec une caméra fixe observant la scène depuis un point extérieur. Cette image permettra de mieux comprendre la configuration physique et les implications pratiques de chaque méthode.

Figure 1:Schéma explicatif de la différence Eye-in-hand et Eye-to-hand

Calibration "Eye-in-Hand" :  
La calibration "Eye-in-Hand" consiste à fixer la caméra directement sur le bras du robot, de sorte qu'elle se déplace avec l'outil. Ce type de calibration vise à établir la relation entre le repère de la caméra et celui du robot, en tenant compte de la position et de l'orientation de l'outil par rapport à l'environnement. L'objectif principal est d'obtenir une transformation précise entre les coordonnées de l'image capturée par la caméra et les coordonnées du robot. Pour ce faire, plusieurs poses du robot sont utilisées, capturant des images de cibles connues pour calculer la matrice de transformation. Cette méthode est particulièrement utile pour des tâches où la caméra doit suivre de près l'outil, comme dans les applications d'assemblage ou d'inspection de proximité.

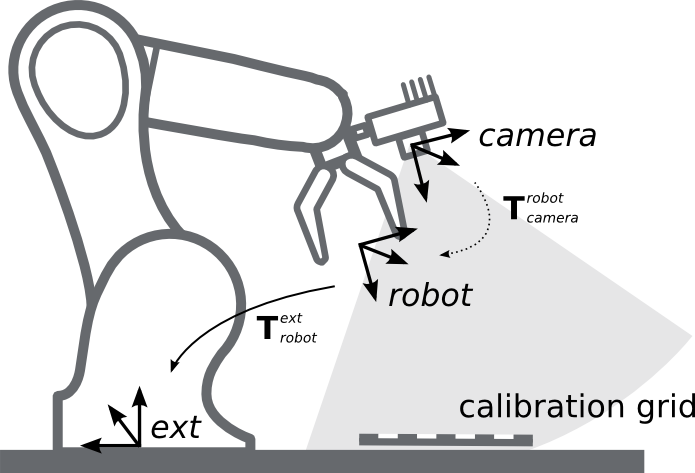


Figure 2 Schéma explicatif du Eye-in-hand

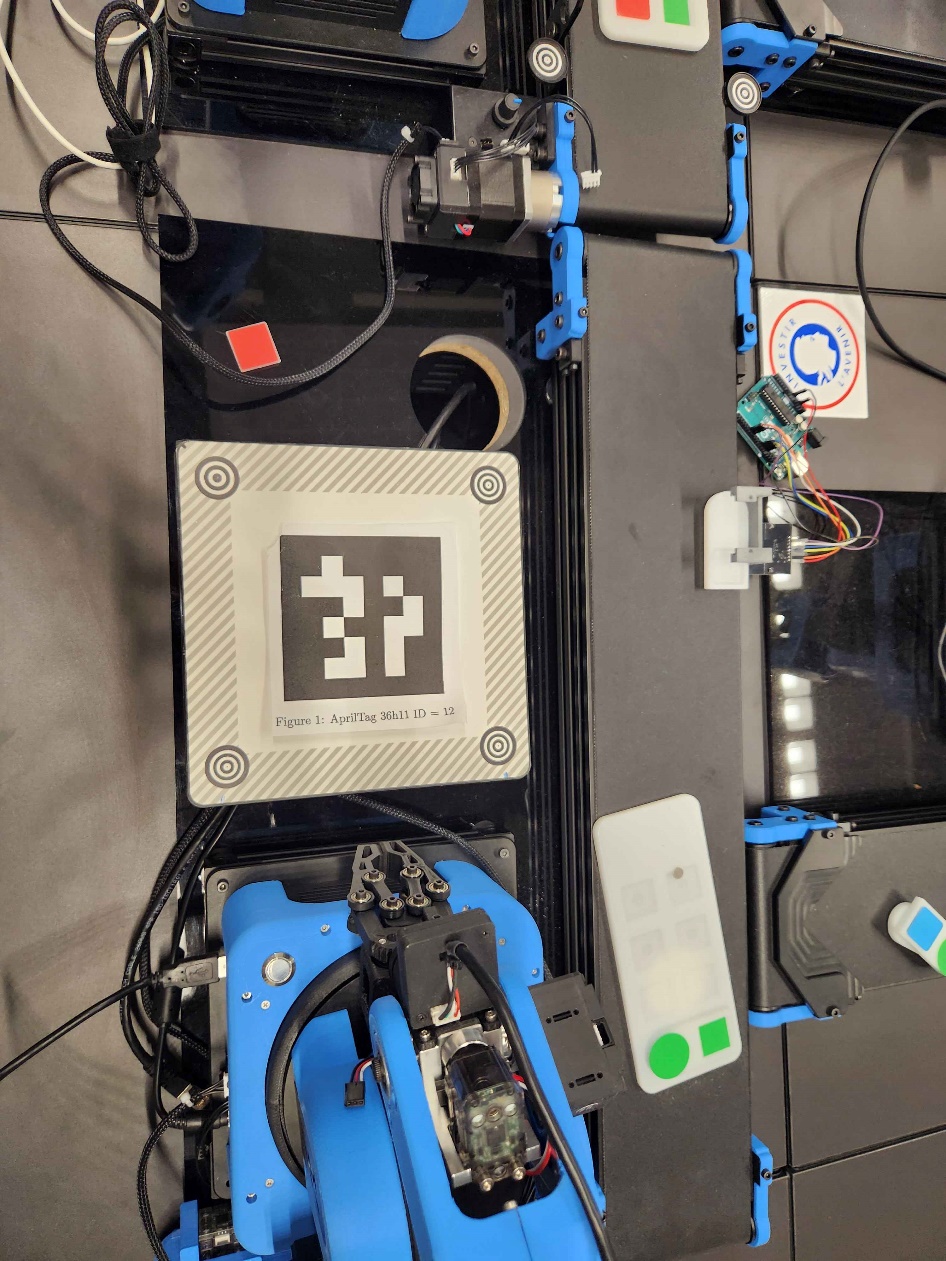
Préparation :  
Dans cette étape, j'ai commencé par équpier le robot Niryo Ned 1 sur lequelle était montée une caméra, j’ai ensuite ajoutée une pince au bout de celui-ci.

Figure 3 Robot Niryo Ned1

Figure 4 Vue aerienne du système

Pour la calibration de la caméra, j'ai utilisé la méthode de calibration extrinsèque à l’aide de damiers que j'avais déjà appliquée dans des projets antérieurs. J'ai fixé un AprilTag sur la table pour servir de point de référence durant l'expérience.

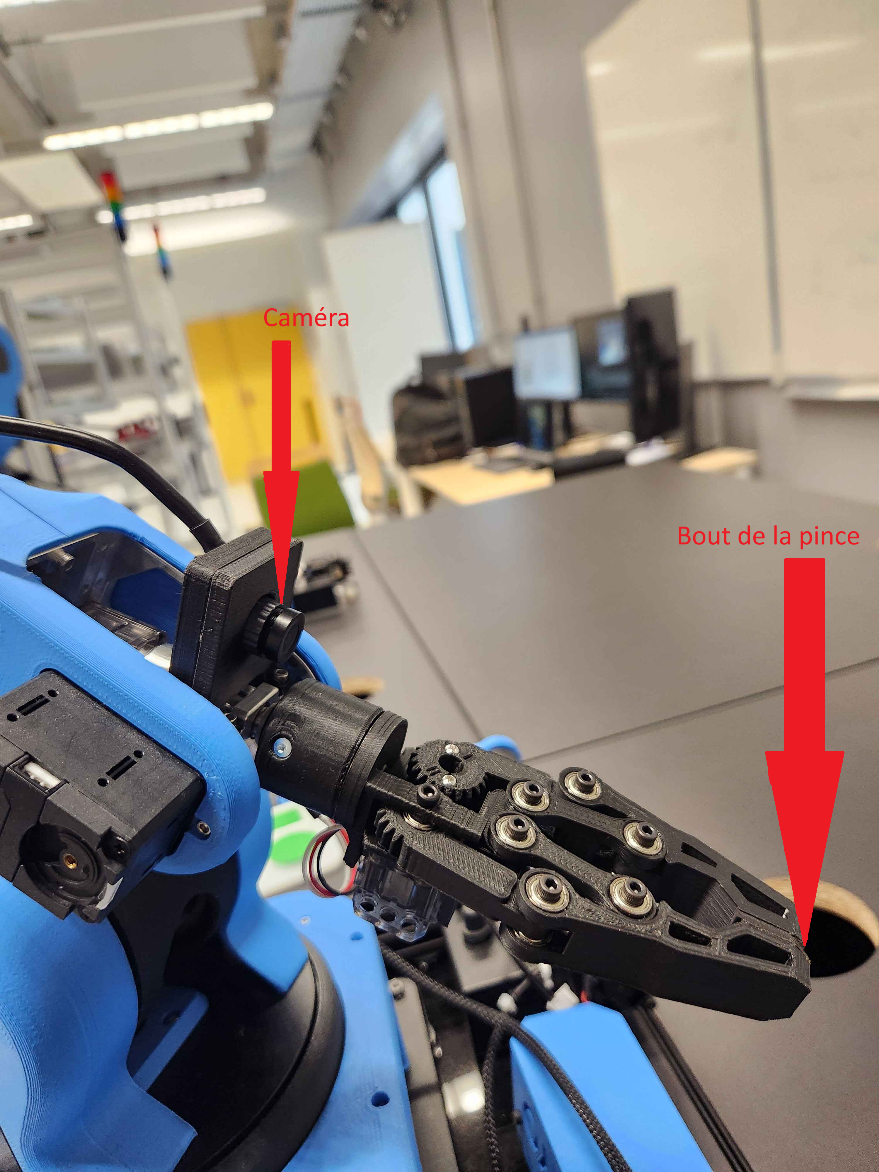
L'objectif était de calibrer le système "Eye-in-Hand", c'est-à-dire de déterminer la relation entre le repère de la caméra et celui de la pince du robot.

Figure 5 Vue rappochée de la pince et de la camera

Cette phase de préparation était cruciale pour assurer que tous les éléments nécessaires à l'expérience étaient bien positionnés et prêts pour la collecte de données.

Expérimentation :  
Une fois la phase de préparation achevée, j'ai débuté l'expérimentation en utilisant la caméra montée sur la pince du robot pour détecter l'AprilTag fixé sur la table. La détection de l'AprilTag s'est faite en capturant des images avec la caméra, ce qui a permis d'extraire les informations de position et d'orientation de l'AprilTag par rapport au repère de la caméra. Cette étape était cruciale pour obtenir les premières données de référence nécessaires à la calibration.

Après avoir établi cette première position, j'ai entrepris de réaliser une expérience durant laquelle l'AprilTag était fixé en un point, tandis que je déplaçais lentement le bras du robot autour de celui-ci. Le déplacement lent du bras était crucial pour garantir une capture précise des données, en minimisant les erreurs dues aux vibrations ou aux mouvements rapides, et en assurant que la caméra puisse toujours suivre l'AprilTag sans perte de vue. L'objectif était de capturer différents points de vue de l'AprilTag en faisant en sorte que le robot effectue plusieurs mouvements précis et variés, tout en conservant la visibilité de l'AprilTag à chaque instant. À chaque moment de cette séquence, j'ai enregistré deux ensembles de données : la position de l'AprilTag par rapport à la caméra (exprimée en coordonnées de translation [x, y, z] et en rotations, soit sous forme d'angles d'Euler, de matrices de rotation, ou de quaternions), ainsi que la position de la pince du robot par rapport à la base du robot (également en coordonnées de translation et en rotations).

Cette collecte de données a permis d'obtenir un ensemble de couples de données pour chaque frame de la vidéo. Chaque couple de données se compose, d'une part, de la position et de l'orientation de l'AprilTag par rapport à la caméra et, d'autre part, de la position et de l'orientation de la pince par rapport à la base du robot. Ce processus a permis d'accumuler suffisamment de données pour effectuer une calibration précise, en assurant une bonne couverture spatiale et angulaire autour de l'AprilTag. Les données recueillies sont essentielles pour calculer la relation géométrique entre la caméra (montée sur la pince) et la base du robot, ce qui est l'objectif principal de la calibration "Eye-in-Hand".

Analyse des résultats :  
Avec les données collectées, j'ai pu calculer le bras de levier, ou "Eye-in-Hand", entre la caméra et la pince. Une fois cette calibration effectuée, j'ai vérifié la validité des résultats en effectuant la procédure inverse : j'ai estimé la position de l'AprilTag par rapport à la base du robot en utilisant les données obtenues. Si la calibration "Eye-in-Hand" entre la caméra et la pince était correcte, la position estimée de l'AprilTag par rapport à la base du robot devait être précise. Cette vérification m'a permis de confirmer l'exactitude de la calibration et d'assurer que le système fonctionnait comme prévu.

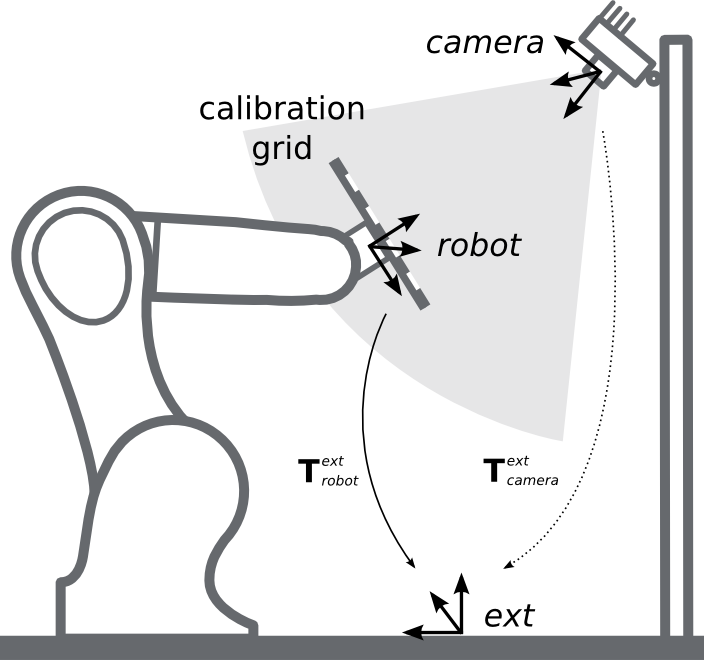
Calibration "Eye-to-Hand" :  
La calibration "Eye-to-Hand" est une méthode de calibration extrinsèque où la caméra reste fixe dans l'environnement, tandis que le robot se déplace pour interagir avec les objets. Contrairement à la calibration "Eye-in-Hand", dans "Eye-to-Hand", la caméra offre une vue d'ensemble de la scène, permettant au robot de se localiser par rapport à des objets dans son environnement.

Figure 6 Schéma explicatif du Eye-to-hand

Cette méthode est particulièrement utile dans des situations où le robot doit manipuler des objets dispersés sur une large surface ou dans des applications où une vue globale est nécessaire pour coordonner les mouvements du robot.

Dans le cadre de mon projet, j'ai choisi d'utiliser le robot Tiago pour réaliser la calibration "Eye-to-Hand". Ce choix a été motivé par plusieurs facteurs : tout d'abord, Tiago est équipé d'une pince dont la position peut être déterminée avec une grande précision, ce qui est essentiel pour des calibrations rigoureuses. De plus, sa facilité de manipulation en fait un outil idéal pour effectuer des expérimentations répétées et créer des datasets de manière efficace.

Figure 7 PAL Robotics TIAGo++

L'objectif était de relier avec précision les positions des pinces du robot aux coordonnées fournies par la caméra fixe. Pour ce faire, j'ai exploité les transformations (TF) du robot, qui fournissent des informations sur la position et l'orientation de chaque composant du robot dans son espace de travail. La caméra fixe capture les images de l'environnement, et à partir de ces images, le robot peut localiser des objets spécifiques, comme des marqueurs ArUco, et ajuster ses mouvements en conséquence.

Préparation :  
La phase de préparation a commencé par la fixation d'un ArUco sur une planche, laquelle était solidement attachée au gripper du robot Tiago.

Figure 8 Vue rapprochée de la pince et de l'aruco

L'ArUco sert de marqueur visuel pour la caméra, permettant de déterminer la position exacte de la pince du robot par rapport à la caméra. J'ai utilisé RViz, un outil de visualisation ROS, pour obtenir une représentation claire de l'orientation des pinces du robot. En RViz, j'ai pu identifier comment les axes X, Y et Z étaient orientés pour chaque pince, ce qui est crucial pour établir la relation entre le marqueur ArUco et la pince du robot.

Cette étape de préparation m'a permis de bien comprendre le cadre de référence de chaque pince du robot, y compris la direction des axes et l'origine des coordonnées. Cela a facilité la création de transformations précises entre le centre de l'ArUco et le point de référence de la pince dans les TF du robot. Cette compréhension initiale est essentielle pour garantir la précision des étapes ultérieures de l'expérimentation, car une mauvaise compréhension des orientations aurait pu entraîner des erreurs significatives dans la calibration.

Expérimentation :  
Avec la préparation terminée, j'ai entamé la phase d'expérimentation. Pour chaque bras du robot, j'ai déterminé la transformation entre le centre de l'ArUco et le point de la pince utilisé pour définir la position dans les TF.

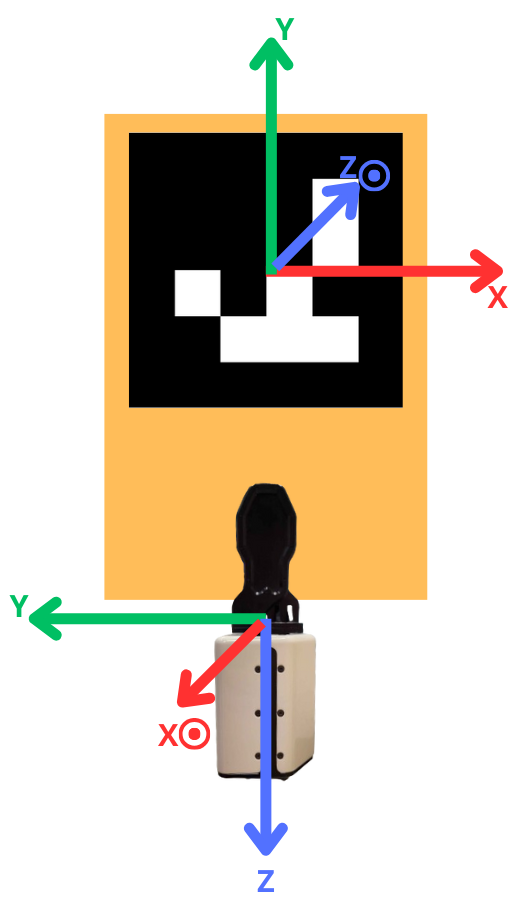
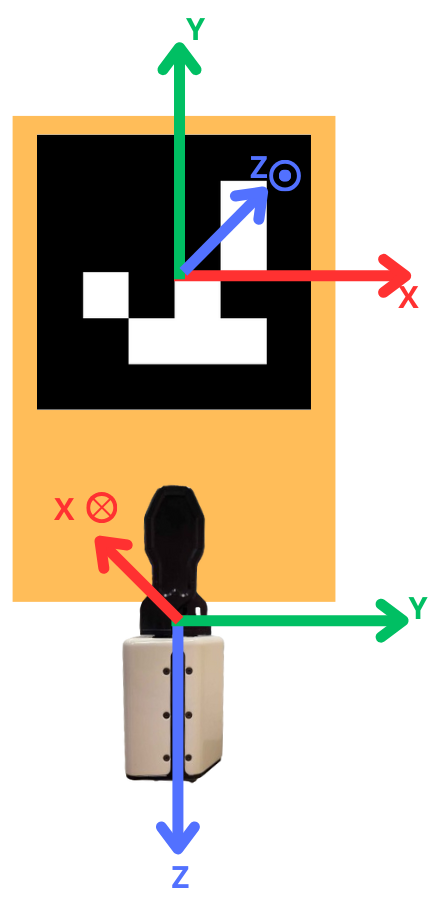


Figure 9 Schéma (avec repères) de la pince droite et de l'aruco

Figure 10 Schéma (avec repères) de la pince gauche et de l'aruco

Cette transformation consiste à relier les coordonnées détectées de l'ArUco par la caméra à la position réelle de la pince du robot.

Je parle

### Création des datasets

J’ai ensuite procédé à la création des datasets. Pour ce faire, j’ai créé des rosbags contenant les images de la caméra rgb de la tête, les paramètres de ladite caméra ainsi que les TF. Pendant la création des datasets de calibration il est crucial de faire varier les position et rotation du gripper sans quoi les résultats seront plus hétérogènes et donc moins précis. J’ai répété l’opération 2 fois par bras.

### Extraction des données

Une fois les rosbags créés, je les ai rejoués et ai lancé mon script Python d’extraction de données. Ce script commence par récupérer les paramètres de la caméra puis les stocke dans un fichier JSON. Ensuite, il enregistre dans un dossier toutes les images envoyées par le robot sur le topic dédié et, pour chaque image, en utilisant les TF, enregistre dans un fichier CSV la position du gripper par rapport à la base. Pour être plus précis, ce n’est pas la position du gripper qui est enregistrée mais celle de l'aruco. En réalité, j’applique la transformation entre l'aruco et le gripper de manière à avoir la même position (l'aruco) mais dans deux repères différents.Cette étape est cruciale car je ne pouvais physiquement pas placer le centre de l’aruco à l’origine du repère du gripper. Dans le reste du document je continurai de parler de la position du gripper pour plus de clareté mais gardez en tête qu’il s’agit de l’aruco.

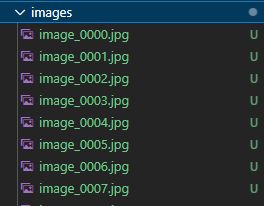


Figure 11 Screen du dossier où sont enregistrées les images

Figure 12 Screen du fichier CSV

J’ai ensuite procédé au calcule de la position de l’aruco dans le repère de la caméra.

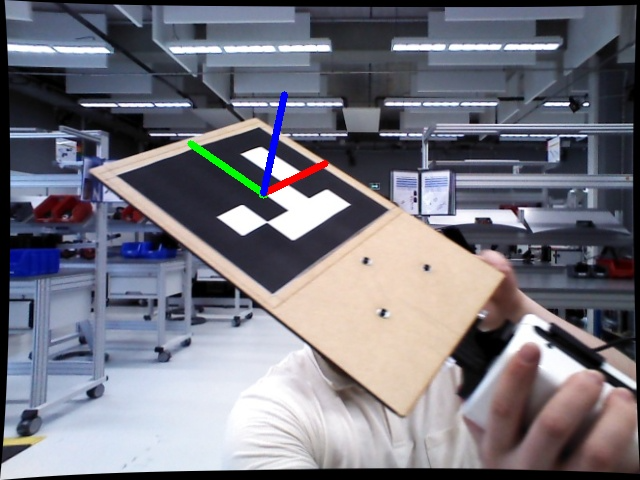


Figure 13 Image après détection de l'aruco

Toutes ces poses sont aussi enregistrées dans un fichier CSV.On peut voir par exemple ci dessous les données du premier aruco détécté.

Figure 14 Screen du fichier CSV

Je procède ensuite a une union des deux fichiers de poses de manière à n’avoir qu’un seul fichier et former ainsi ce qu’on appellera par la suite un « couple de pose » ou « couple de données ». Il faut cepandant faire attention car l’aruco n’est pas détécté dans toutes les images et il faut donc consciencieusement sélectionner les poses du gripper correspondantes.

Figure 15 Screen du fichier CSV

### Calcul de la transformation entre la caméra et la base du robot :

Pour chaque pose, une fois toutes les transformations récupérées, j'ai procédé au calcul de la transformation globale entre la caméra et la base du robot. Cette étape repose sur l'utilisation de l'algorithme de calibration de Tsai, une méthode couramment utilisée dans les systèmes de vision robotique pour résoudre des problèmes de transformation entre différentes bases de référence.

Mon objectif principal était de déterminer la matrice de rotation et le vecteur de translation qui décrivent précisément comment la caméra est positionnée et orientée par rapport à la base du robot. Pour garantir la précision de cette transformation, j'ai adopté une approche basée sur le tri des poses en fonction de leur erreur de reprojection.

#### Calcul de l'erreur de reprojection :

Pour chaque pose, j'ai calculé l'erreur de reprojection (en 2D ou en 3D). Cette erreur mesure la différence entre la position estimée des points projetés par la caméra et leur position réelle détectée sur l'image. J'ai réalisé cela en mesurant deux types d'erreurs :

Erreur de Reprojection 2D:  
L'erreur de reprojection est une mesure utilisée pour évaluer la qualité d'une calibration. Elle est calculée en projetant les points 3D estimés à partir de la caméra sur l'image 2D, puis en comparant ces projections avec les positions réelles observées des points dans l'image. En d'autres termes, elle mesure la distance entre les points observés et les points recalculés sur l'image après calibration.

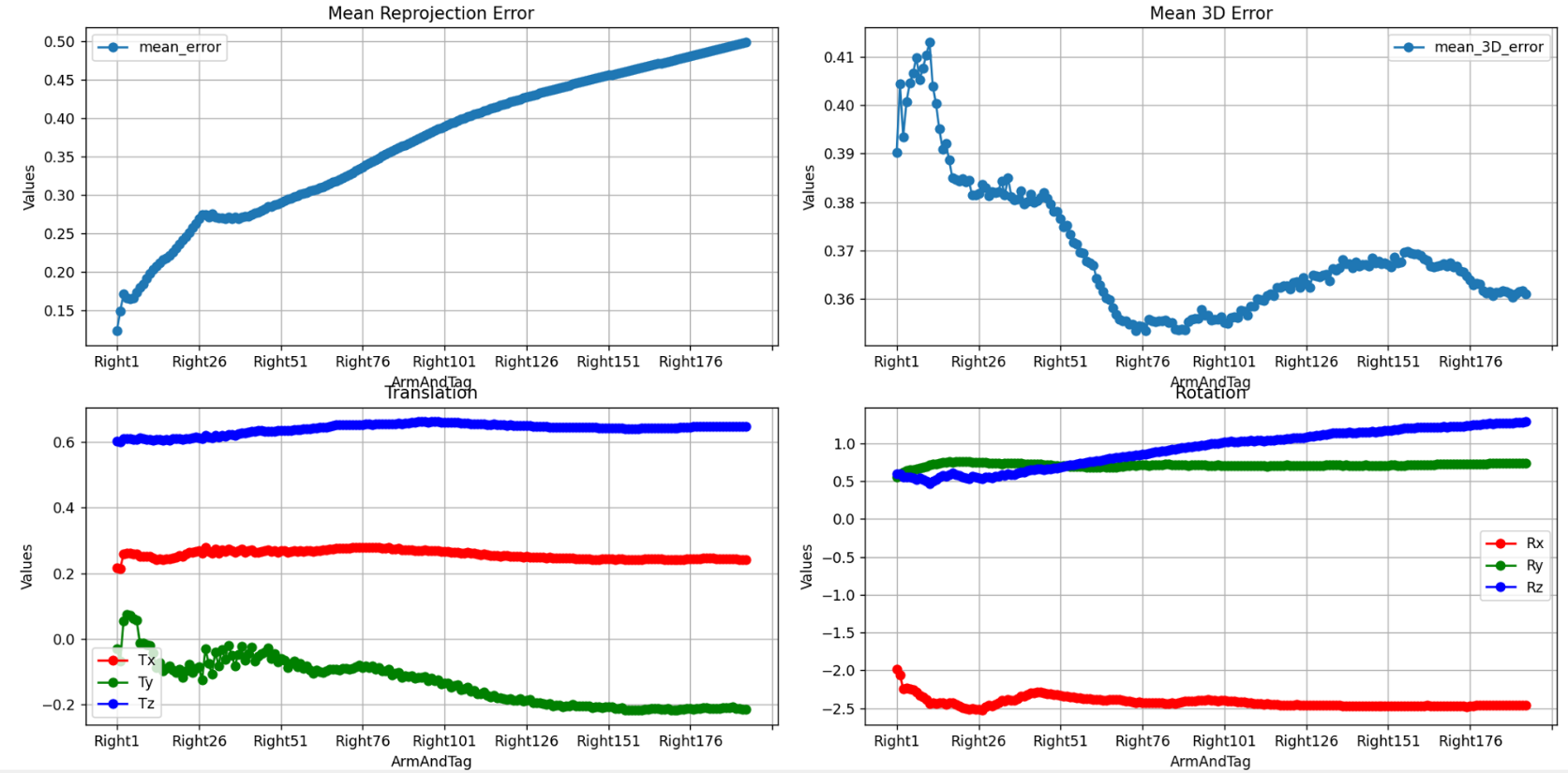
Pour chaque image de mon jeu de données, j'ai calculé cette erreur de reprojection pixelique (en 2D). Ensuite, j'ai classé ces images par ordre croissant d'erreur, afin de sélectionner les combinaisons d'images offrant les erreurs les plus faibles. Cette approche m'a permis d'identifier des ensembles d'images garantissant une erreur de reprojection minimale, ce qui améliore la précision de la calibration.

Figure 16 Schéma avec tanslation (m) et rotation (radians) après minimisation de l'erreur pixelique

**Point faible de cette méthode** : Bien que cette méthode de sélection par l'erreur de reprojection 2D soit efficace pour améliorer la précision dans le plan de l'image, elle présente certaines limitations. En se basant uniquement sur l'erreur 2D, on ignore les erreurs potentielles dans la profondeur (axe Z) ou dans les rotations de l'objet dans l'espace 3D, ce qui peut entraîner une calibration incorrecte. En d'autres termes, une faible erreur de reprojection 2D ne garantit pas nécessairement une précision optimale dans l'espace 3D réel.

C'est pourquoi il peut être pertinent de calculer également une **erreur 3D** qui prend en compte la précision de la position des points dans les trois dimensions de l'espace. Ces mesures plus complètes permettent d'obtenir une calibration plus robuste et précise.

Erreur de Reprojection 3D :

Malgré l’amélioration qu’a apporté le tri des images en fonction de l’erreur de reprojection, les résultats constatés restent encore torp hétérogènes et manque cruellement de précision.

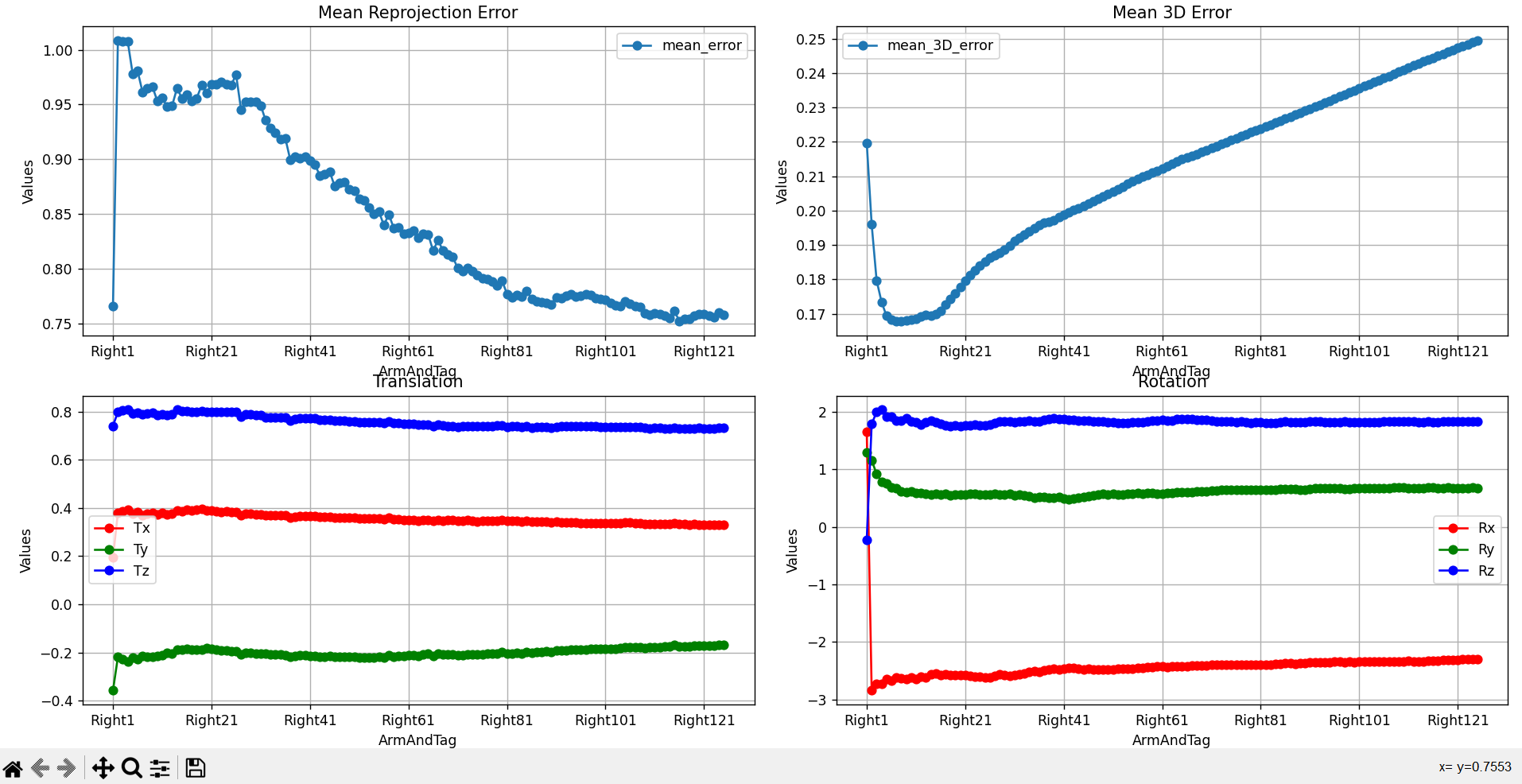
J'ai donc décidé d’appliqué une technique similaire en utilisant l'erreur 3D. Contrairement à l'erreur de reprojection, qui se concentre sur la précision des points projetés en 2D, l'erreur 3D évalue la différence entre les positions estimées et réelles des points dans l'espace tridimensionnel. Cette mesure est particulièrement utile pour s'assurer que les points calibrés sont cohérents étant donné que nous avons une troisième composante (profondeur).

Figure 17 Schéma avec tanslation (m) et rotation (radians) après minimisation de l'erreur 3D

On constate que l’erreur 3D minimale est légèrement inférieure à 0,17 m, ce qui est considérablement élevé et non cohérent avec la précision attendue du robot pour effectuer des tâches de pick-and-place, où une précision beaucoup plus fine est généralement requise. Cette erreur importante peut entraîner des imprécisions significatives dans les opérations du robot.

Néanmoins, on voir rapidement que les valeurs de rotation et de translation semblent être moins sujettes à des variations importantes, ce qui est un point positif.

#### Sélection progressive des images :

La précision de la calibration dépend grandement des images utilisées pour calculer la transformation. Pour garantir les meilleurs résultats, j'ai mis en place un processus de sélection progressive des images basées sur l'erreur de reprojection.

**Initialisation avec la meilleure image** : J'ai commencé par calculer l'erreur de reprojection pour chaque image individuellement. Cette erreur quantifie la différence entre la position réelle des points projetés sur l'image et leur position estimée. Parmi toutes les images, j'ai sélectionné celle avec la plus petite erreur de reprojection comme point de départ. Cette image sert de base initiale pour construire une combinaison optimale.

**Ajout itératif d'images** : Une fois la première image sélectionnée, j'ai procédé à un ajout itératif des images suivantes. À chaque étape, j'ai testé l'ajout d'une nouvelle image à l'ensemble des images déjà sélectionnées. Pour ce faire :

* + J'ai temporairement ajouté chaque image non encore sélectionnée à l'ensemble actuel.
  + J'ai recalculé l'erreur de reprojection globale en utilisant toutes les images de cet ensemble élargi.
  + J'ai comparé cette nouvelle erreur à celle de l'ensemble précédent.

**Sélection de l'image optimale** : Parmi toutes les images testées à cette itération, j'ai conservé celle qui a permis de minimiser l'erreur de reprojection globale. Ce processus a été répété jusqu'à ce que l'ajout de nouvelles images n'améliore plus significativement la précision, ou que l'ensemble de poses sélectionnées atteigne un seuil prédéfini de précision.

**Filtrage des images** : Si une image ajoutée augmentait l'erreur de reprojection au-delà d'un certain seuil ou n'améliorait pas significativement la précision, elle était exclue du processus. De cette manière, seules les images les plus pertinentes et les plus précises ont été retenues pour le calcul final.

#### Optimisation de la transformation :

Une fois les images optimales sélectionnées, j'ai utilisé ces données pour calculer la transformation globale entre la caméra et la base du robot en utilisant l'algorithme de calibration de Tsai.

**Application de l'algorithme de Tsai** : L'algorithme de Tsai est une méthode robuste et bien établie pour résoudre le problème de l'eye-to-hand calibration dans les systèmes de vision. À partir des positions et orientations sélectionnées, l'algorithme détermine la transformation optimale en termes de matrice de rotation (qui décrit l'orientation de la caméra par rapport à la base) et de vecteur de translation (qui décrit la position de la caméra par rapport à la base).

**Calcul de la matrice de rotation et du vecteur de translation** :

L'algorithme détermine simultanément la matrice de rotation et le vecteur de translation pour positionner et orienter la caméra par rapport à la base du robot. La matrice de rotation minimise les écarts angulaires entre les directions observées et prédictes des points d'intérêt, tandis que le vecteur de translation ajuste la position de la caméra en réduisant les différences entre les positions réelles et estimées. Ce processus garantit une transformation optimale entre la caméra et la base.

# Pistes d’améliorations :

**Utilisation avancée de RANSAC :**L'implémentation de RANSAC pour la sélection des inliers pourrait potentiellement améliorer la robustesse du calcul des transformations, notamment en évitant de tomber dans un minimum local, ce qui peut être le cas avec ma méthode actuelle. RANSAC pourrait offrir une solution plus optimale en filtrant les points qui introduisent des erreurs importantes. De plus, il serait intéressant d'explorer des variantes plus avancées de RANSAC, comme MLESAC (Maximum Likelihood Estimation Sample Consensus) ou PROSAC (Progressive Sample Consensus). Ces algorithmes pourraient offrir une meilleure précision dans la sélection des inliers, surtout dans des ensembles de données où le bruit est plus prononcé. Toutefois, cette hypothèse reste à tester dans le cadre de ce projet.

**Amélioration du calcul de l'erreur 3D :** Le calcul de l'erreur 3D pourrait être rendu plus robuste/précis.

**Optimisation du code pour une exécution plus rapide :** Le code actuel pourrait bénéficier de plusieurs optimisations pour réduire les temps de calcul. L'implémentation du parallélisme via des bibliothèques comme multiprocessing ou joblib permettrait d'accélérer les processus lourds, tels que le calcul des erreurs sur de grandes combinaisons de poses. De plus, une meilleure gestion de la mémoire, en évitant des copies inutiles de données ou en utilisant des structures de données plus efficaces, pourrait réduire l'empreinte mémoire et améliorer les performances globales.

**Compréhension approfondie des repères et des transformations :** Pour solidifier la compréhension des transformations entre repères, il serait bénéfique de travailler sur des exercices pratiques, tels que :

* Résoudre des problèmes d'application des matrices de rotation et des vecteurs de translation dans des contextes variés (robotique, réalité augmentée, etc.).
* Expérimenter avec des transformations successives dans des repères multiples, pour bien comprendre l'enchaînement des changements de repère.
* Créer et manipuler des simulations 3D où l'on doit calculer manuellement les transformations pour visualiser les changements de repères en temps réel.

**Exploration de nouvelles techniques :** Une piste intéressante pourrait être l'intégration de méthodes d'apprentissage automatique pour améliorer les processus de calibration. Par exemple, des réseaux de neurones pourraient être entraînés à prédire des transformations optimales à partir de grandes quantités de données calibrées, offrant une alternative ou un complément aux méthodes traditionnelles comme Tsai.

**Documentation et automatisation des tests :** Pour rendre le processus de calibration plus fiable et reproductible, il serait utile de développer une suite de tests automatisés qui vérifient systématiquement la précision des transformations calculées. Une documentation claire des différentes étapes et des choix algorithmiques permettrait aussi d'améliorer la maintenabilité du code et de faciliter son utilisation par d'autres.

# Bilan :

Ce projet m'a profondément enrichi tant sur le plan technique que personnel. L'une des principales leçons que j'en retire est une compréhension approfondie des concepts liés aux repères, en particulier les transformations entre repères et les changements de repère.

Travailler sur ce projet m'a également permis d'acquérir une expérience pratique précieuse. J'ai pu manipuler et appliquer des algorithmes complexes comme celui de Tsai pour la calibration de la caméra, ce qui a renforcé mes compétences en mathématiques appliquées et en programmation. Le projet m'a aussi familiarisé avec des techniques robustes comme RANSAC, que je commence à utiliser pour améliorer la précision de mes résultats. Ces connaissances sont transférables à d'autres projets où l'identification d'inliers et l'optimisation des modèles sont essentielles.

Ce projet m'a non seulement permis de mieux comprendre les concepts théoriques de la robotique, mais il m'a aussi doté d'outils et de méthodes pratiques que je pourrai réutiliser dans d'autres domaines techniques. L'expérience acquise ici sera sans aucun doute un atout pour mes futurs projets professionnels et académiques.

# Conclusion

Ce mémoire a traité de la calibration extrinsèque des systèmes robotiques en se concentrant sur deux méthodes principales : "Eye-in-Hand" et "Eye-to-Hand". Dans le cadre de ce projet, un algorithme de calibration a été développé, permettant de minimiser les erreurs de reprojection et d'améliorer la précision des transformations géométriques nécessaires pour l'alignement entre la caméra et le robot.

L'expérimentation a impliqué la collecte de données par le biais de pluisieurs séries de tests, suivie de l'analyse de ces données pour calculer les transformations entre la caméra et les différentes bases du robot. L'algorithme de Tsai a été utilisé pour déterminer les matrices de rotation et les vecteurs de translation, tandis que l'optimisation des images et des poses a permis d'améliorer la précision des calibrations. Des techniques robustes comme RANSAC ont été employées pour filtrer les données et renforcer la fiabilité des résultats.

Les résultats obtenus ont révélé des points forts et des points faibles dans les méthodes utilisées, avec des pistes d'amélioration identifiées. Ce travail a permis de proposer des solutions techniques pour améliorer la précision et l'efficacité des calibrations robotiques, essentielles pour leur application dans des environnements industriels complexes.

À l'avenir, ces premières avancées pourraient être approfondies et étendues, notamment par l'exploration de nouvelles techniques comme l'intégration de RANSAC ou d'autres algorithmes robustes. Ces perspectives offrent des opportunités prometteuses pour le développement de systèmes robotiques au sein du laboratoire, mais aussi dans les projets auxquels il participe.

# Annexes

## Glossaire

 **Calibration extrinsèque** : Processus de détermination de la relation spatiale (position et orientation) entre une caméra et un autre système de référence, comme un robot.

 **Eye-in-Hand** : Configuration où la caméra est montée sur le bras ou l'outil d'un robot, se déplaçant avec celui-ci.

 **Eye-to-Hand** : Configuration où la caméra est fixe et indépendante des mouvements du robot, offrant une vue globale de la scène.

 **Matrice de rotation** : Représentation mathématique utilisée pour décrire l'orientation d'un objet dans l'espace tridimensionnel.

 **Vecteur de translation** : Vecteur décrivant le déplacement d'un objet d'un point à un autre dans l'espace tridimensionnel.

 **Algorithme de Tsai** : Algorithme de calibration utilisé pour déterminer la position et l'orientation d'une caméra par rapport à un système de référence, basé sur les repères visuels.

 **RANSAC (Random Sample Consensus)** : Algorithme itératif utilisé pour estimer les paramètres d'un modèle mathématique à partir d'un ensemble de données contaminé par du bruit ou des outliers.

 **Outlier** : Donnée ou observation qui diffère significativement des autres données du jeu de données, souvent considérée comme une anomalie.

 **Transformation géométrique** : Opération mathématique qui modifie la position, l'orientation, ou la taille d'un objet dans l'espace tridimensionnel.

 **Erreur de reprojection** : Mesure de la différence entre la position d'un point observé et sa position projetée après calibration, utilisée pour évaluer la précision d'un modèle de calibration.

 **AprilTag** : Marqueur visuel utilisé en robotique pour la détection et la localisation dans les systèmes de vision par ordinateur.

 **Repère** : Système de coordonnées utilisé pour définir la position et l'orientation d'un objet dans l'espace.

 **Pose** : Position et orientation d'un objet dans l'espace, souvent définie par un ensemble de coordonnées et des angles de rotation.

 **ROS (Robot Operating System)** : Cadre de développement logiciel pour la robotique, facilitant la création de programmes modulaires et distribués pour les robots.

 **TF (Transformation Frame)** : Structure de données dans ROS qui gère les transformations entre différents repères dans un système robotique.