

**INSTITUT NATIONAL DES SCIENCES
APPLIQUEES DE STRASBOURG**

Section Mécanique

Spécialité Mécatronique



Power and productivity
for a better world™

**PROGRAMMATION OFFLINE D'UNE
CELLULE ROBOTISEE A L'AIDE DE
TECHNIQUES INNOVANTES**

Rapport de Projet de Fin d'Etudes de :

GIANCOLA Silvio

Septembre 2012

Condensé (Abstract) :

L'utilisation de robots anthropomorphes à des fins de décoration d'objets par peinture doit permettre d'allier la répétabilité et les rythmes de production soutenus à la reproduction de mouvements précis et artistiques. Afin de programmer la dynamique du robot, l'apprentissage des mouvements doit pouvoir se faire de manière interactive, en s'affranchissant d'une programmation point par point conventionnelle. Dans ce projet, c'est une utilisation intelligente de trois caméras, calibrées et couplées en stéréoscopie qui nous permet d'enregistrer des mouvements de pinceaux. Lors d'un enregistrement, les pinceaux utilisés par le décorateur sont instrumentés à l'aide de sphères réfléchissantes, dont la position dans l'espace est tracée avec la géométrie épipolaire, concept de base de la stéréoscopie. Grâce aux positions de ces sphères et à un algorithme de « Rigid Registration », il est possible de repositionner la géométrie complète du pinceau dans l'espace, et donc de retrouver position et orientation de la pointe de l'outil. Un filtrage de ces données et l'enregistrement de ces séquences de trajectoires comme modèles pour le robot permet de dupliquer un procédé de peinture, avec une précision dans les tracés de l'ordre du millimètre.

The use of anthropomorphic robots for tiles decoration, painting them, links the repeatability and the high rhythm of production with a precise and artistic movement. For programming such a robot dynamic, learning movements must be done in an interactive way, by releasing the user from a conventional point-by-point programming. In this project, a smart use of three cameras, calibrated and coupled in stereoscopy, allows the programmer to register the paintbrush movements. During a registration, the paintbrush used by the designer is sensorized with reflector spheres, of which the positions in space are known and traced using epipolar geometry, basic concept of stereoscopy. Through the spheres positions and a « Rigid Registration » algorithm, we are able to find the paintbrush geometry in space and therefore to find the position and the orientation of brush tip. Filtering data and registering painting sequences like models for the robot allows it to replicate the painting process, with a precision of about a millimeter.

Remerciements :

Je remercie les ingénieurs Paolo Conca et Remo Sala qui m'ont donné la possibilité de réaliser mon Projet de Fin d'Etudes respectivement dans les locaux de « ABB » à Sesto San Giovanni, et dans le « VBLab (Vision Bricks Laboratory) » du Politecnico di Milano, et grâce à qui j'ai eu l'occasion de mûrir tant sur le plan professionnel que sur le plan personnel. Je remercie de même mon tuteur INSAsien, le professeur Pierre Renaud, qui a su me guider et apporter un point de vue externe au projet.

Je remercie aussi tous les collègues m'ayant accompagné et ayant participé à ce projet, à commencer par mon binôme Davide Chiarion, qui outre sa science a su partager sa culture italienne. Je remercie de même « ABB » pour son appui dans la partie robotique, avec une mention spéciale aux techniciens Gianluigi Moro, Paolo Bassetti et Paola Gambini, sans oublier les collègues du « VBLab » Giacomo Mainetti, Ambra Vandone, Alessandro Basso et Mario Galimberti, pour la transmission de leur savoir pour la partie vision.

Je remercie particulièrement la décoratrice et restauratrice d'œuvre d'art Isabella Pirola, grâce à qui nous avons pu tester le développement de notre système, et valider la faisabilité de ce projet.

Je remercie aussi mes parents, à qui je dédie ce travail, sans qui je n'aurais jamais pu réaliser un tel parcours universitaire, parsemé d'obstacle, mais si enrichissant.

Enfin, un dernier remerciement va à toutes les personnes m'ayant entouré lors de ce projet et qui ont su rendre ce dernier vivant et inoubliable.

Sommaire

CHAPITRE 1	INTRODUCTION	8
CHAPITRE 2	SYSTEME D'ACQUISITION	9
2.1	MODELISATION D'UNE CAMERA.....	9
2.2	GEOMETRIE EPIPOLAIRE ET STEREOSCOPIE	15
2.3	CALIBRATION	19
CHAPITRE 3	METHODOLOGIE DE MESURES	21
3.1	PRESENTATION DU MATERIEL	21
3.2	PRESENTATION DES LOGICIELS	25
CHAPITRE 4	FILTRAGE ET INTERPOLATION.....	31
4.1	INTRODUCTION	31
4.2	TECHNIQUE D'INTERPOLATION ET DE FILTRAGE	33
4.3	CONFRONTATION DES TECHNIQUES	36
4.4	CONCLUSIONS.....	40
CHAPITRE 5	EXPERIMENTATIONS	43
5.1	PRESENTATION DE LA CELLULE ROBOTIQUE	43
5.2	ESSAI AVEC DECORATRICE.....	45
CHAPITRE 6	ANALYSE METROLOGIQUE	47
6.1	PERFORMANCES CONFORMES A LA NORME ISO 9283	47
6.2	PRECISION DU SYSTEME COMPLET	48
6.3	REPETABILITE DU SYSTEME COMPLET	49
6.4	BILAN SUR LES PERFORMANCES DU SYSTEME COMPLET	50
CHAPITRE 7	DEVELOPPEMENTS FUTURS ET CONCLUSIONS.....	51
7.1	DEVELOPPEMENTS FUTURS.....	51
7.2	CONCLUSIONS GENERALES	52

Tables des figures

Figure 2.1-a : Principe de base du modèle de la « pinhole camera »	9
Figure 2.1-b : Optique diaphragme obturateur et capteur CCD	9
Figure 2.1-c : Modélisation géométrique	10
Figure 2.1-d : Déplacement du repère de l'image	10
Figure 2.1-e : Image sujette à distorsion et image non sujette à distorsion	12
Figure 2.1-f : Image sans distorsion, avec barillet et coussinet	13
Figure 2.1-g : Modélisation de la distorsion radiale (droite) et tangentielle (gauche) .	14
Figure 2.2-a : Modèle de stéréoscopie	15
Figure 2.2-b : Recherche d'un point grâce au plan épipolaire.....	16
Figure 2.2-c : Stéréoscopie à deux caméras	16
Figure 2.2-d : Stéréoscopie à trois caméras	18
Figure 2.2-e : Récupération de 49 points par stéréoscopie	18
Figure 2.3-a : Exemple de calibrateur utilisé par la méthode de Zhang	19
Figure 2.3-b : Calibrateur HALCON.....	20
Figure 3.1-a : Caméra uEye avec objectif	21
Figure 3.1-b : Triplette de caméra	22
Figure 3.1-c : Position idéale des caméras selon les 3 axes X, Y et Z	22
Figure 3.1-d : Illuminateur circulaire et filtre	23
Figure 3.1-e : Photo du calibrateur réfléchissant (avec et sans illumination).....	23
Figure 3.1-f : Pinceau instrumenté	23
Figure 3.1-g : Blobs de tailles différentes testés.....	24
Figure 3.1-h : Ecart type en fonction du diamètre du blob	24
Figure 3.2-a : Schéma de transfert de données	25
Figure 3.2-b : Présentation du logiciel de récupération des blobs dans l'espace	26
Figure 3.2-c : Exemple d'homothétie : rotation, translation et facteur d'échelle.....	29
Figure 4.1-a : Position du pinceau (bleu) et trajectoire (rouge).....	31
Figure 4.1-b : Position et orientation de la pointe du pinceau	32
Figure 4.2-a : Exemple d'interpolation cubique.....	33
Figure 4.2-b : Interpolation Spline	34
Figure 4.2-c : Tracé d'une courbe de Bézier (en bleu, $u=0.5$)	34
Figure 4.2-d : Interpolation par courbe de Bézier (dans l'espace)	35
Figure 4.2-e : Interpolation de Bézier d'un cercle avec 20 points et 50 points.....	35
Figure 4.3-a : Mouvement sinusoïdal	37
Figure 4.3-b : Combinaison linéaire de mouvements sinusoïdaux.....	38
Figure 4.3-c : Signal constant bruité par un bruit blanc	39
Figure 4.3-d : Mouvement sinusoïdal avec bruit blanc	40
Figure 4.4-a : Mouvement interpolé par B-Spline.....	41

Figure 4.4-b : Composante du mouvement interpolé par B-Spline	41
Figure 4.4-c : Mouvement filtré par un passe-bas	42
Figure 4.4-d : Composante du mouvement filtré par un passe-bas.....	42
Figure 5.1-a : Architecture du robot IRB 120	43
Figure 5.1-b : Cellule robotisée implémentée	44
Figure 5.1-c : Ilot de travail imaginé par ABB	44
Figure 5.2-a : Effet recherché avec une peinture aquarelle et sur carrelage	45
Figure 5.2-b : Confrontation des résultats des essais (numéroté de 40 à 49).....	46
Figure 6.1-a : Définition de précision et répétabilité par la norme ISO 9283.....	47
Figure 6.2-a : Trajectoire suivant les axes du référentiel commun	48
Figure 6.3-a : Gaussiennes des positions suivant X, Y et Z.....	50
Figure 7.1-a : Exemple de pince pneumatique à prise cubique	51
Figure 7.2-a : Remerciement de la part du robot lui-même	52

Chapitre 1 Introduction

L'utilisation de la robotique dans des environnements productifs ou liés à la recherche est de plus en plus commune. Que ce soit pour des tâches répétitives, difficiles, délicates ou dangereuses pour l'opérateur, les robots sont des solutions alternatives à la main d'œuvre humaine. Certaines tâches sont cependant difficiles à programmer à travers une méthode conventionnelle, la richesse et la précision des mouvements humains étant longs et fastidieux à reproduire, surtout si ces derniers sont travaillés et élaborés de manière spéciale ou artistique.

L'utilisation de robots anthropomorphes à des buts décoratifs de peinture d'objets ne fait pas exception à cet aspect artistico-productif, alliant la répétabilité et les rythmes de production soutenus avec des mouvements précis et artistiques. C'est ainsi que des techniques innovantes et alternatives peuvent être trouvées, utilisant par exemple des systèmes de caméras pour tracer et enregistrer ces mouvements, pour ensuite être reproduit par un robot.

Dans ce mémoire, nous présenterons une technique basée sur la stéréoscopie utilisant un ensemble de trois caméras pour enregistrer, traiter et dupliquer un quelconque mouvement. Une telle méthode sera appliquée à la décoration de carrelage, qu'il est actuellement possible à réaliser uniquement par des décorateurs qualifiés. Une méthode productive et disponible pour décorer ces carreaux est l'impression, mais cette dernière ne rend pas le relief tel que la peinture peut le faire.

Le mémoire est composé de quatre grandes parties, la première étant une présentation des modèles de caméra et de la technique de stéréoscopie ; la seconde présentera les méthodes utilisées pour acquérir et traiter les mouvements de pinceau réalisés par un décorateur ; la troisième partie présentera les essais réalisés servant à justifier la faisabilité d'un tel processus, et finalement une dernière partie sera consacrée à l'analyse des performances du système complet.

Chapitre 2 Système d'acquisition

2.1 Modélisation d'une caméra

Une caméra est un appareil permettant de visualiser un environnement tel qu'il est visible à partir d'un point de vue, et d'en extraire une projection dans un plan. Une référence peut être faite au modèle de « pinhole camera » [1] [2] représenté en **figure 2.1-a**, où sur une simple boîte est réalisé un perçage. La lumière de l'extérieur entrante par ce trou vient se projeter sur le côté opposé de la boîte et il apparaît une image, celle de la scène extérieure vu à travers ce trou, correspondant pour une caméra à son centre optique.

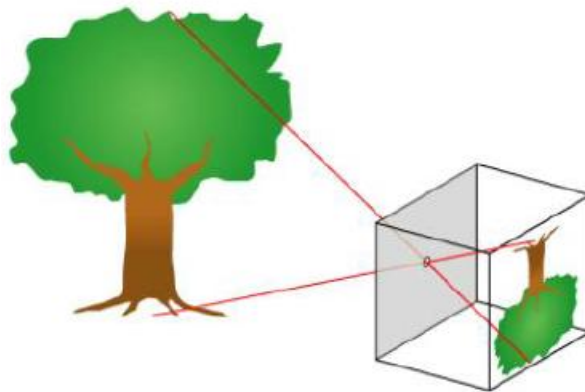


Figure 2.1-a : Principe de base du modèle de la « pinhole camera »

Ce modèle est repris par toutes les caméras numériques qu'il est possible de trouver dans le commerce, où un système de lentilles minces, diaphragme et obturateur remplace le perçage, autorisant ainsi un temps d'exposition variable sur un capteur CCD et enregistrant l'image projetée (voir **figure 2.1-b**).

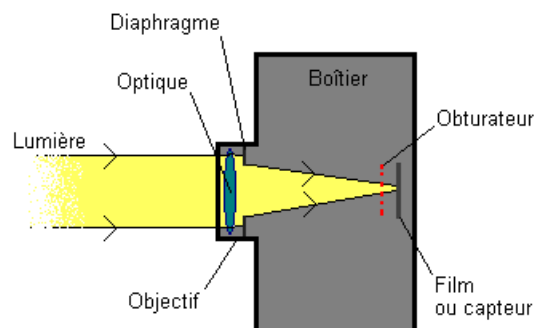


Figure 2.1-b : Optique diaphragme obturateur et capteur CCD

Ce même modèle permet de comprendre comment la projection est réalisée géométriquement. La **figure 2.1-c** illustre différemment ce modèle, où la fonction mathématique mise en avant transforme un point P de coordonnées (X, Y, Z) de la scène en un point p de coordonnées $\left(\frac{fX}{Z}, \frac{fY}{Z}, f\right)$ sur le plan image de la caméra, f étant la distance du centre optique au capteur CCD, appelé focale.

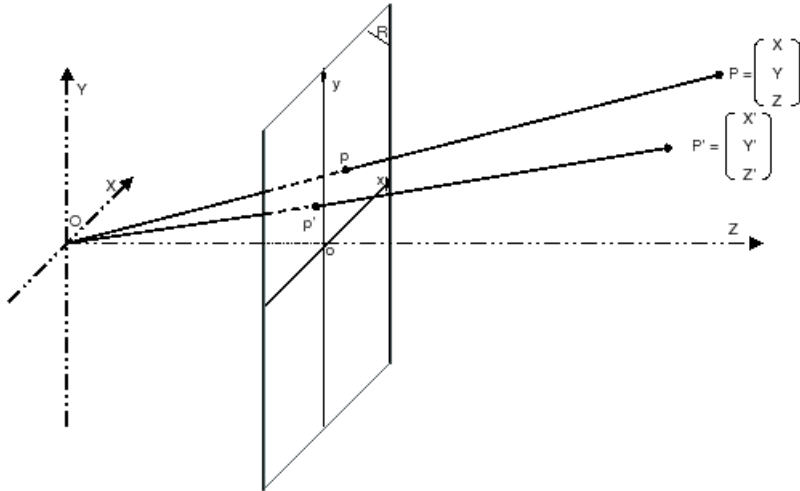


Figure 2.1-c : Modélisation géométrique

Les coordonnées généralisées sont communément utilisées dans la modélisation, pour simplifier les équations et écrire le modèle de la caméra par matrice. Ainsi :

$$\begin{pmatrix} f \cdot X \\ f \cdot Y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P * \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

La matrice P obtenue est appelé *matrice de projection*, ou *matrice de la caméra*. Elle sera complétée par d'autres paramètres en améliorant le modèle. En déplaçant premièrement le centre du repère dans l'angle de la photo et non au centre, nous ajoutons deux paramètres au modèle, x_0 et y_0 , correspondant à la translation du repère dans l'angle inférieur gauche de l'image (voir **figure 2.1-d**), et prenant en compte une possible excentration entre l'optique et le capteur.

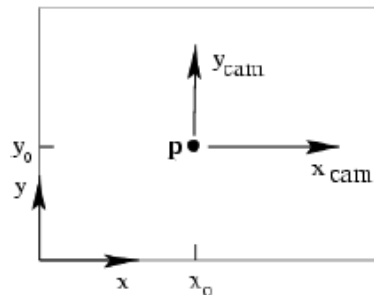


Figure 2.1-d : Déplacement du repère de l'image

La matrice de projection devient alors :

$$\begin{pmatrix} f \cdot X + Z \cdot x_0 \\ f \cdot Y + Z \cdot y_0 \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & x_0 & 0 \\ 0 & f & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Où x_0 et y_0 sont les coordonnées du point principal p . En introduisant la *matrice de calibration* K suivante :

$$K = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Il est possible d'écrire une forme compacte suivante :

$$x = P * X_{cam} = K * [I|0] * X_{cam}$$

Dans le cas où le système de référence sur lequel on souhaite se baser ne correspond pas au système de référence de la caméra, les deux systèmes sont reliés par une rotation et une translation. En posant R la matrice de rotation et C les coordonnées du centre de la caméra dans le référentiel galiléen, l'équation suivante est obtenue :

$$x = K * R * [I|-C] * X_{monde}$$

Ou encore :

$$x = K * [R|t] * X_{monde}$$

où $t = -RC$ représente la translation directe entre coordonnées cartésiennes et coordonnées dans l'image. Un total de 9 paramètres a déjà pu être défini : trois dus à la matrice de calibration K (f , x_0 , y_0), trois à la matrice de rotation R (rotation sur 3 axes), et trois à C (3 coordonnées spatial du centre de la caméra)

Pour ce projet sont utilisées uniquement des caméras digitales, dont le capteur n'est autre qu'une matrice de semi-conducteurs appelé *pixels*. Chacun de ces derniers convertit les photons captés par l'objectif de la caméra en lumière incidente dans son secteur. La géométrie d'un pixel n'est pas toujours carrée, et comme les images sont par la suite mesurées en pixels, les facteurs horizontaux et verticaux transformant pixel en mesure sont différents. Pour tenir compte de ce phénomène, la matrice de calibration K devient :

$$K = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Avec $f_x = f * m_x$ et $f_y = f * m_y$, où m_x et m_y représente le nombre de pixel par unité de mesure respectivement selon x et y .

Il est nécessaire d'introduire un paramètre supplémentaire pour décrire entièrement le modèle de caméra, il s'agit du « skew ». Il décrit la possibilité que les pixels aient des cotés non perpendiculaires entre eux, et est exprimée par l'expression suivante :

$$s = \cos(\alpha) * f$$

Où α est l'angle formé par deux côtés d'un pixel. Ce paramètre est nul pour une majorité de caméra. Au final, la matrice de calibration est exprimée comme suit :

$$K = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ce modèle contient ainsi 11 degrés de liberté, deux de plus par rapport au précédent, avec l'ajout du paramètre de « skew » et la duplication du paramètre de focale, prenant en compte les pixels rectangulaires.

Le principal avantage du modèle « pinhole camera » est qu'il décrit la caméra en utilisant uniquement des relations linéaires entre les paramètres. Dans de nombreuses applications ce modèle offre une bonne approximation de la réalité. Cependant, il n'est pas suffisamment précis lorsque d'importantes distorsions optiques sont présentes, causées par la présence d'une lentille mince. Ces distorsions font de la caméra un système non-linéaire. Le modèle linéaire vient donc être modifié ultérieurement afin de s'approcher au maximum de la réalité. Les effets de la distorsion sont visibles en figure 2.1-e, où à gauche est présentée une image sujette à des distorsions importantes et à droite la même image mais non sujette à ces distorsions. Les lignes rouges tracées sur le plafond dans l'image illustrent la non-conservation de la linéarité dans la projection de l'environnement.



Figure 2.1-e : Image sujette à distorsion et image non sujette à distorsion

Deux types de distorsions peuvent se produire. Des distorsions appelées *radiales* peuvent apparaître avec barillet ou avec coussinet (**figure 2.1-f**) en fonction des lentilles minces utilisées, et surtout lorsque la distance focale est faible.

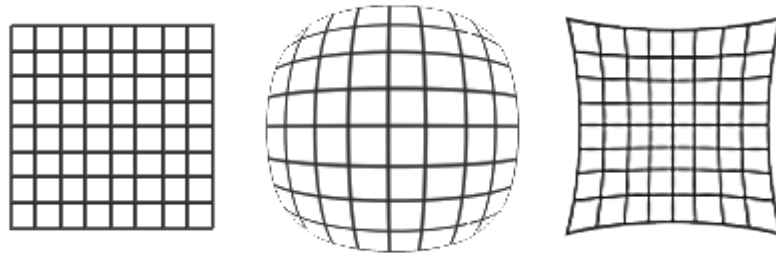


Figure 2.1-f : Image sans distorsion, avec barillet et coussinet

Une fois corrigées ces distorsions, la formation d'images peut à nouveau être considéré comme linéaire. Le processus pour cette correction est présenté par la résolution des équations suivantes :

$$\begin{aligned}\hat{x} &= x_0 + (x - x_0)(k_1 r^2 + k_2 r^4 + \dots) \\ \hat{y} &= y_0 + (y - y_0)(k_1 r^2 + k_2 r^4 + \dots)\end{aligned}$$

Avec :

$$r = (x - x_0)^2 + (y - y_0)^2$$

Et où (x, y) sont les coordonnées image mesurées, (\hat{x}, \hat{y}) sont les coordonnées corrigées, tandis que (x_0, y_0) est le centre de la distorsion radiale. Le facteur $(k_1 r^2 + k_2 r^4 + \dots)$ n'est autre que le développement limité d'une fonction de Taylor, où la connaissance des paramètres de distorsion k_1, k_2, k_3, \dots suffit à la reconstruction de l'image.

Des distorsions tangentielles peuvent également se produire. Un meilleur modèle pour traiter les aberrations optiques a été introduit par Brown en 1966 [3], appelé modèle « Plumb Bob ». Le déplacement du centre des distorsions est traduit comme une distorsion tangentielle, dû principalement à une erreur de montage de la lentille ou à un défaut de fabrication de cette dernière. En nommant x_n les coordonnées normalisé de la caméra (x, y) , la correction devient :

$$x_d = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) * x_n + dx$$

Avec dx décrivant le déplacement introduit par les distorsions tangentielles :

$$dx = \begin{bmatrix} 2p_1 xy + p_2 (r^2 + 2x^2) \\ 2p_2 xy + p_1 (r^2 + 2y^2) \end{bmatrix}$$

La **figure 2.1-g** ci-dessous représente les distorsions radiales et tangentielles sous forme de champs vectoriels [4].

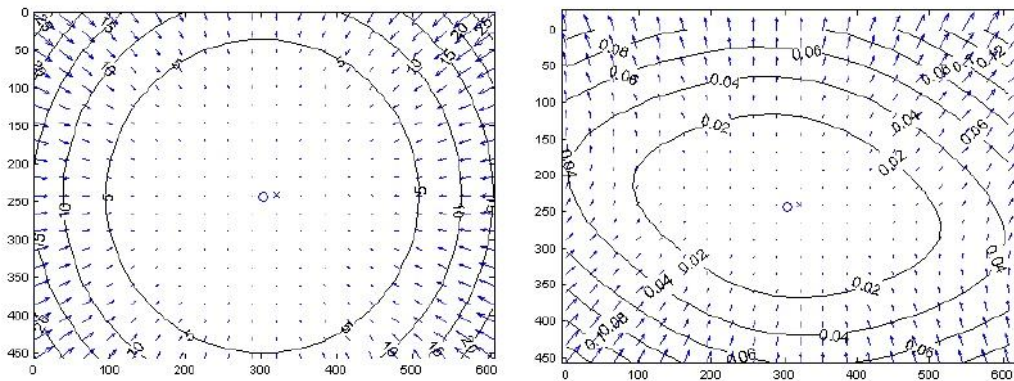


Figure 2.1-g : Modélisation de la distorsion radiale (droite) et tangentielle (gauche)

Finalement, le modèle est complet pour décrire une quelconque caméra. Celle-ci est caractérisé par 16 paramètres, séparables en diverses catégories : intrinsèques, extrinsèques et de distorsion. La calibration est l'ensemble des opérations permettant de déterminer ces paramètres, dont les descriptions complètes sont les suivantes :

- 5 paramètres intrinsèques :
 - f_1 et f_2 , la distance focale exprimée dans les deux directions du plan image,
 - x_0 et y_0 , les deux coordonnées du centre optique,
 - s , le « skew ».
- 6 paramètres extrinsèques :
 - r_1, r_2 et r_3 , les trois paramètres de rotation exprimés en angle d'Euler,
 - t_1, t_2 et t_3 , les trois paramètres de translation.
- 5 paramètres de distorsion :
 - k_1, k_2 et k_3 , les trois paramètres de distorsions radiales,
 - p_1 et p_2 , les deux paramètres de distorsions tangentielles.

Il n'est pas toujours nécessaire d'utiliser le modèle complet, la complexité introduite dans ce modèle peut être excessive pour certaines applications. En général, les distorsions peuvent être omises, en particulier la distorsion tangentielle et les deux derniers paramètres de distorsion radiale. Pour des raisons liées à la convergence des méthodes de minimisation, considérer un modèle trop large avec des paramètres qui sont absents dans la réalité conduit souvent à des calibrations erronées.

2.2 Géométrie épipolaire et stéréoscopie

2.2.1 Introduction

La *stéréoscopie* est la capacité perceptive qui permet d'unir des images provenant de deux points de vue différents (par exemple les yeux dans le cas des êtres humains), et qui grâce à leur disposition structurale, présentent un déplacement latéral. Cette disparité est utilisée par le cerveau pour récupérer des informations sur la profondeur et sur la position spatiale d'un objet ciblé. Par conséquent, la stéréoscopie permet de générer une vision tridimensionnelle. Cette méthode naturelle a été étudiée et a été appliquée à la vision artificielle, donnant vie à la *stéréoscopie*, correspondant à l'ensemble des techniques permettant de retrouver des informations sur une structure tridimensionnelle, grâce à au moins un couple d'images acquises de manière synchrone et à partir de positions différentes. Un modèle est présenté ci-dessous en figure 2.2-a.

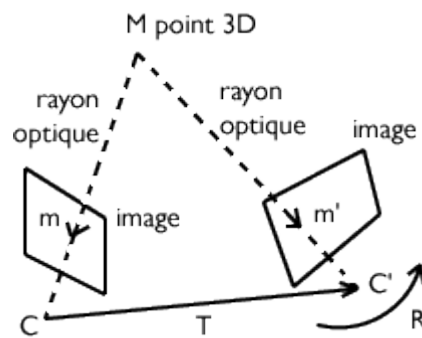


Figure 2.2-a : Modèle de stéréoscopie

L'utilisation de cette méthode peut être décomposée en 2 parties, la 1^{ère} correspondant à la recherche de points homologues à partir des différents points de vue, et la 2^{nde} à l'accouplement de ces points pour en retrouver la position dans une scène.

La première partie se résout simplement en prenant des points présentant des géométries notables, tel que des sphères. Ces dernières ont la particularité de se projeter dans n'importe quel plan de l'espace en un disque. Par conséquent, ces disques peuvent être vus de manière égale par toutes les caméras, et le centre de ces dernières peut être considéré comme le centre de la sphère.

En revanche, la deuxième partie est plus délicate, pour accoupler deux points dans deux caméras distantes, il faut exploiter la géométrie épipolaire, expliquée ci-dessous.

2.2.2 Géométrie épipolaire avec deux caméras

La géométrie épipolaire de base s'appuie sur un couple de deux caméras calibrées, récupérant par une quelconque méthode un même point sur les deux images. La **figure 2.2-b** représente schématiquement les deux caméras positionnées dans l'espace ainsi que leur « plan épipolaire ». Ce dernier, comme représenté sur la partie (a), est le plan passant par les deux centres optiques C et C' des caméras et par le point X à reconstruire. Sur la partie (b), la caméra de gauche ne voit que la projection de l'inconnue X dans son plan image, et donc ne peut pas déterminer sa position dans l'espace, mais déjà réduire l'ensemble des solutions à une droite, appelée « droite épipolaire ». La caméra de droite, ayant un angle de vue différent, peut retracer cette droite appartenant au plan épipolaire, et ce grâce aux paramètres extrinsèques de la calibration que sont la position et l'orientation relatives des caméras. Sur cette 2^{nde} caméra, il est alors possible de retracer la projection de la droite épipolaire, et donc de trouver une solution pour la position de X dans l'espace.

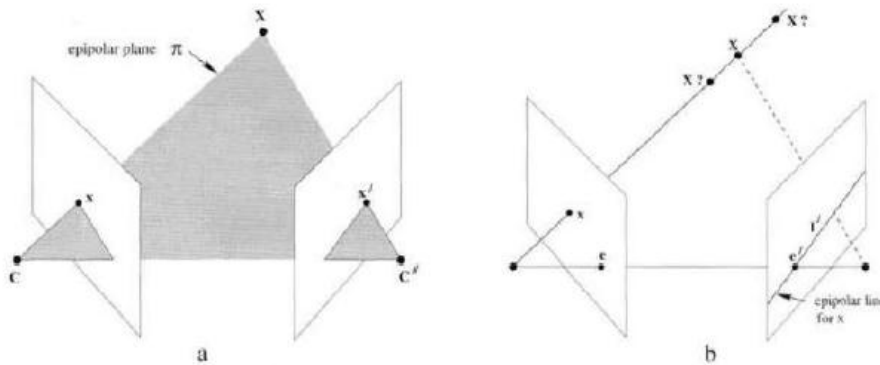


Figure 2.2-b : Recherche d'un point grâce au plan épipolaire

La **figure 2.2-c** illustre un exemple de vue des deux caméras, avec un point notable qu'il est possible de retrouver sur les deux vues. Les lignes rouges sont les droites épipolaires projetées sur chacune des images.

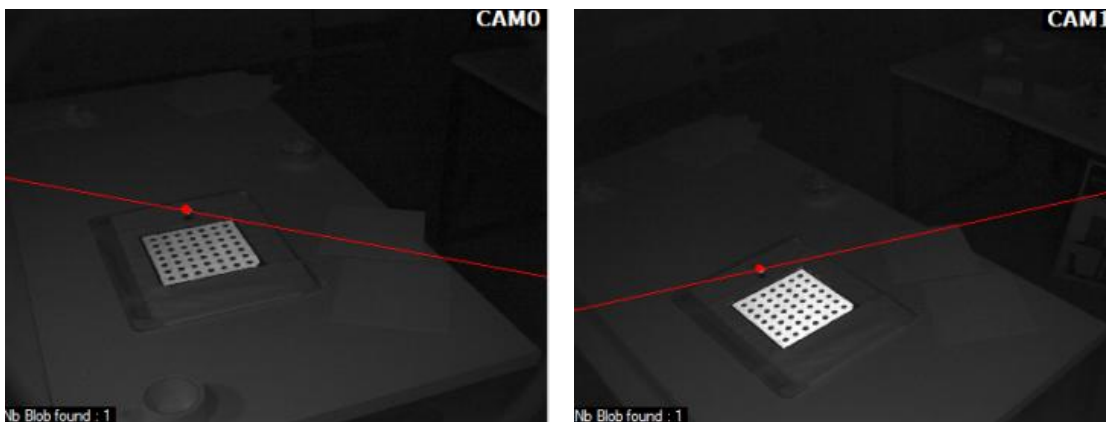


Figure 2.2-c : Stéréoscopie à deux caméras

Remarquons qu'un nombre n de points peut être retrouvé, à partir du moment où les droites épipolaires ne se confondent pas. Dans ce cas spécial, les informations de position dans l'espace peuvent être erronées. Cette erreur est présente surtout lorsque les points à retrouver possèdent une dynamique importante, dont héritent les plans épipolaires. En se mouvant, ces plans peuvent se rapprocher et se chevaucher pendant un court instant, créant ainsi des incertitudes sur l'intégrité des mesures. Les données sont alors inexploitables.

2.2.3 Géométrie épipolaire avec trois caméras

La stéréoscopie avec trois caméras peut se faire de différentes méthodes, en réalisant trois accouplements de caméras deux à deux ou en modifiant l'algorithme de base.

Une 1^{ère} méthode consiste à réaliser simplement trois stéréoscopies indépendantes, en couplant deux à deux les caméras. En moyennant intelligemment les résultats des trois stéréoscopies, on réduit ainsi les erreurs et on a accès à des informations privilégiées grâce à certaines stéréoscopies sans pour autant utiliser toutes les caméras à disposition. En clair, l'absence d'information sur une caméra, si l'objet n'est pas visible de son point de vue, peut être résolue en utilisant les informations de la stéréoscopie formée par les deux autres caméras. Cela permet d'avoir un système plus robuste, avec une meilleure capacité à reconstruire l'environnement. C'est une méthode généralisable à m caméras, utilisant alors $\sum_{k=1}^m (k-1)$ accouplement stéréoscopiques. Avec une augmentation du nombre de caméras, le nombre de points de vue accroît, ainsi que la robustesse et la précision, mais aussi le temps de traitement informatique des données.

Malheureusement, cette méthode conserve le défaut principal de la stéréoscopie à deux caméras, qui est la possibilité d'avoir des résultats erronés lorsque n points sont à retrouver, comme expliqué précédemment. De plus, avec un plus grand nombre de stéréoscopie, la possibilité d'avoir des résultats erronés est aussi plus grande. Finalement, la dynamique si critique est présente par définition de notre application, vu qu'un « mouvement » doit être mesuré avec certainement plus d'un point notable.

Une 2^{nde} méthode consiste à adapter l'algorithme de la stéréoscopie avec deux caméras à trois caméras, en utilisant les informations disponibles avec la 3^{ème} caméra pour enlever le doute possible sur la véracité des résultats. Pour ce faire, la géométrie épipolaire est utilisée trois fois, afin d'avoir deux plans épipolaires reliés à chaque caméras. Ainsi, comme illustré dans l'exemple en *figure 2.2-d*, chaque caméra peut voir les deux droites épipolaires qui lui sont rattachées, et plutôt que de chercher le point le plus proche à une seule droite, l'information de deux droites est disponible. Le point de l'image correspondant est donc à une distance minimale des deux droites en même temps, c'est-à-dire de leur intersection, et donc réduit à un point.

Algorithmiquement, on cherche à accoupler les points trouvés sur une caméra à la suivante, en commençant par la distance point-droite la plus faible, et en cherchant à boucler un cycle complet. En partant d'un point en caméra 0, on projette son plan épipolaire en caméra 1, où l'on cherche le point à plus faible distance de la droite. A partir de ce nouveau point en caméra 1, on fait de même pour trouver son homologue en caméra 2. Finalement, on vérifie que ce point en caméra 2 correspond au point initial en caméra 0. Si c'est le cas, la boucle est bouclée et le point peut être reconstruit. Si en revanche, il ne correspond pas au point initial, on revient en arrière dans l'arborescence de l'algorithme et le prochain point à distance minimale de la droite est alors recherché, jusqu'à ce que la boucle soit bouclée. Il s'agit d'un algorithme « Depth First » utilisant un « Back Tracking » pour rechercher les autres solutions. Il s'avère efficace car les solutions parcourues sont classées par distance croissante aux droites épipolaires, et donc les premières solutions sont les plus probables.

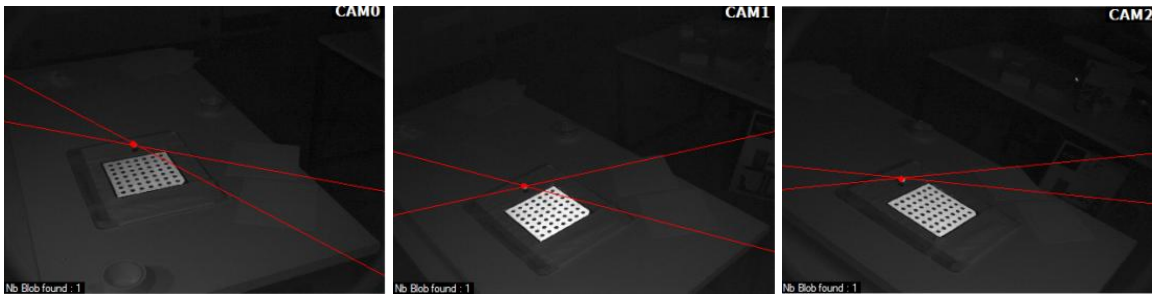


Figure 2.2-d : Stéréoscopie à trois caméras

Sur la *figure 2.2-e*, un ensemble de 49 points a pu être retrouvé. Ces mêmes points n'avaient pas pu être retrouvés avec une stéréoscopie à deux caméras, car les droites épipolaires étaient trop proches. Avec la modification de l'algorithme de base comme présentée précédemment, la robustesse est accrue, et ce même en cas de fortes dynamiques des points.

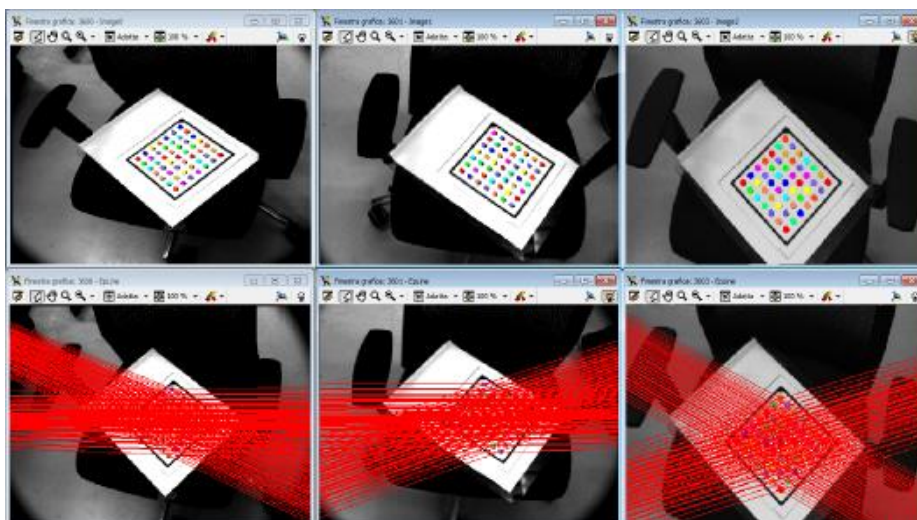


Figure 2.2-e : Récupération de 49 points par stéréoscopie

2.3 Calibration

Le terme de *calibration* d'une caméra correspond au processus de détermination des paramètres qui définissent le modèle de la caméra décrit précédemment. La calibration est un processus nécessaire pour pouvoir retrouver des informations métriques sur une image et pouvoir ainsi utiliser la caméra comme instrument de mesure. Il n'existe pas de norme qui impose une méthode de calibration particulière, c'est en fonction de l'application qu'un choix peut se faire.

Les techniques existantes peuvent se diviser en deux principales catégories : les méthodes basées sur l'utilisation d'un calibrateur (*pattern*), et les techniques d'auto-étalonnage (*self-calibration*) [5].

Les premières consistent en l'utilisation de *calibrateurs*, c'est-à-dire un objet dont la disposition dans l'espace de certains points sont connus avec une précision suffisamment élevée. La reconnaissance de ces points par une caméra permet d'individualiser les paramètres du modèle de cette dernière, à travers la projection de la scène dans l'image acquise. Parmi ces méthodes, il est possible de citer la « *Direct Line Transformation Method* » (DLT), l'« *optimisation non linéaire* » ou encore le « *Tsai two-step method* » et la « *méthode de Zhang* » [5]. Ci-dessous en figure 2.3-a, un exemple de calibrateur utilisé par la méthode de Zhang est présenté.

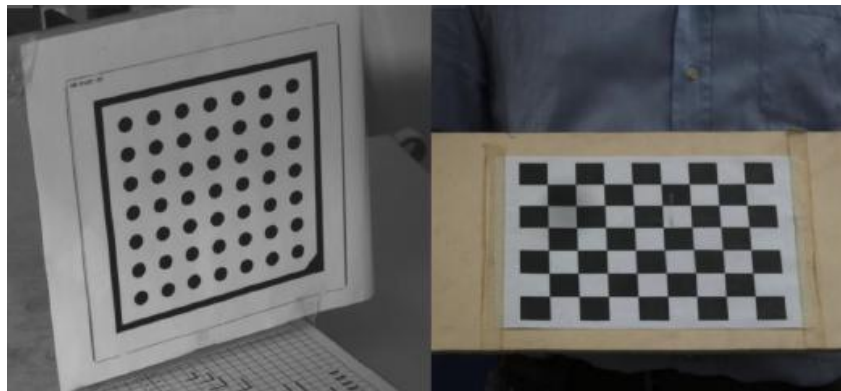


Figure 2.3-a : Exemple de calibrateur utilisé par la méthode de Zhang

Les secondes méthodes pouvant être utilisées sont connues sous le nom de techniques d'« auto-étalonnage » (*self-calibration*). Ces dernières ne nécessitent pas de calibrateur particulier, mais à travers un mouvement de la caméra dans une scène statique, il est possible de retrouver les relations qui lient les paramètres de calibration. Ces méthodes sont utiles lorsque les paramètres de la caméra ne peuvent être maintenus fixes pour l'application en question. Parmi ces dernières, il est possible de citer divers algorithmes génétiques tels que les méthodes de « *Yang* » et d'« ajustement des faisceaux » (*bundle adjustment*) [5].

Des outils commerciaux ont été développés par Matlab ou encore MVTEC (HALCON) [6] basés sur ces méthodes théoriques de calibration. La technique de MVTEC utilisée dans notre application est développée en langage HDevelop, basé lui-même sur le langage C#. Cette technique se base sur un calibrateur présenté en *figure 2.3-b* qui, pris en photo dans différentes positions, permet de retrouver les paramètres du modèle de la caméra.

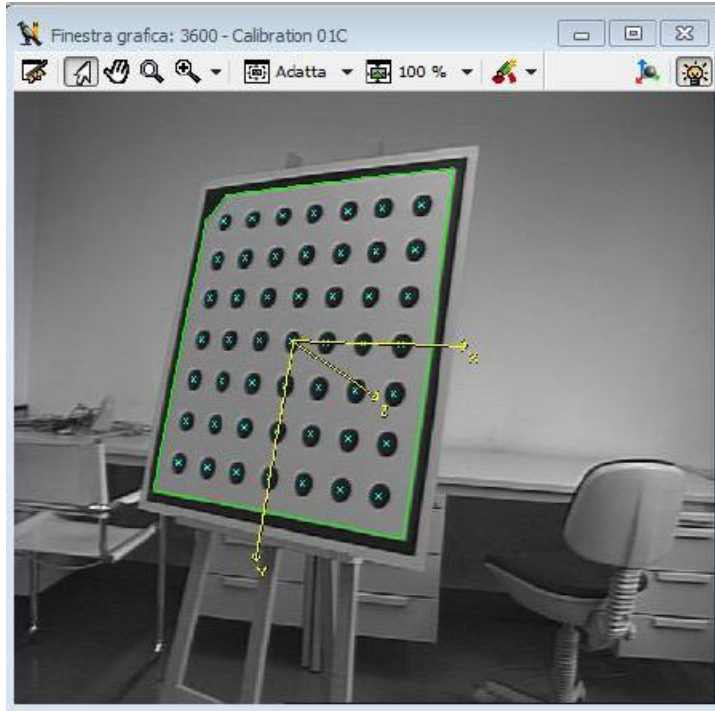


Figure 2.3-b : Calibrateur HALCON

Différents gabarits de calibrateur existent, mais ce dernier doit être choisi pour couvrir environ un quart de l'image, et entre 15 à 20 images doivent être prises pour avoir une bonne calibration, représentant le calibrateur dans différentes poses.

Pour la calibration d'une stéréoscopie, qui résulte en la connaissance de la position d'une caméra par rapport à une autre, les images du calibrateur doivent être prises de manière synchrone par toutes les caméras, afin de faire correspondre sur chacune des prises les différents repères, et retrouver la translation et la rotation des caméras entre elles. Une des prises peut aussi servir de référentiel pour l'application, de manière à placer ce dernier au choix dans la scène.

Chapitre 3 Méthodologie de mesures

3.1 Présentation du matériel

Au début du stage, un système de stéréoscopie à trois caméras avait déjà été installé au VBLab. Par conséquent, un modèle de base était déjà disponible. Celui-ci a fait l'objet d'une précédente étude ayant pour sujet principal les méthodes de calibration [5]. De ce fait, le choix du matériel avait déjà été en grande partie fait.

3.1.1 Les caméras

Les caméras sont au nombre de trois, référencé *UI-5240CP* de la marque « *uEye* », et sont montrées sur la [figure 3.1-a](#). Elle possède une résolution de 1280 par 1080 pixels, et chacun de ces pixels sont de forme carrée de côté $5,3 \mu\text{m}$. Il s'agit uniquement d'un capteur CCD, auquel vient s'ajouter un objectif, permettant de régler la focale, l'ouverture du diaphragme et le temps d'exposition. Le capteur permet d'acquérir des images à une fréquence maximale de 50 fps (frame per second), mais nous la limiterons physiquement par un générateur de signal carré fonctionnant à 30 Hz, afin de réduire le nombre de données acquises.



Figure 3.1-a : Caméra uEye avec objectif

Pour ce qui est du positionnement des caméras, celles-ci sont installées sur un support comme présenté sur la [figure 3.1-b](#). Cette position n'est pas celle idéale en ce qui concerne la précision d'acquisition, mais peut être convenable (voir Chapitre 6 : Analyse métrologique). En effet, une caméra ne connaît que la projection de l'environnement, et donc n'a aucune information par rapport à son axe sortant. La raison d'un tel emplacement est la facilité que l'on a à placer un calibrateur visible par toutes les caméras, afin de calibrer en même temps les paramètres du modèle des caméras et la stéréoscopie.

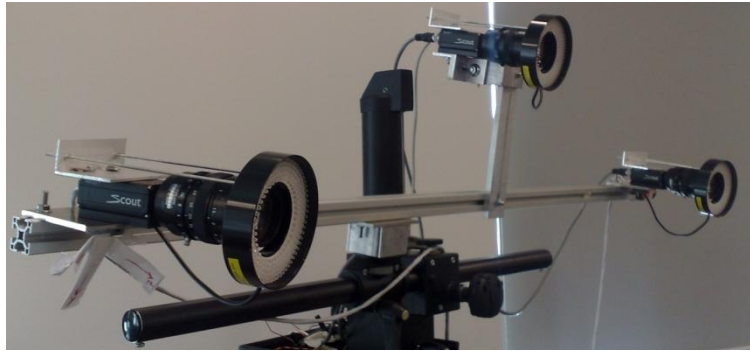


Figure 3.1-b : Triplette de caméra

Une position idéale pour les caméras serait d'aligner les trois axes optiques de celles-ci avec les 3 axes du référentiel de l'objet à peindre, comme schématisé en *figure 3.1-c*. Ainsi, les erreurs seraient homogénéisées sur les trois axes de l'espace. Cependant, la calibration à l'aide d'un calibrateur simple est difficile, car ce dernier doit être visible par toutes les caméras dans diverses positions, et dans ce cas selon les trois dimensions de l'espace représenté par les trois axes.

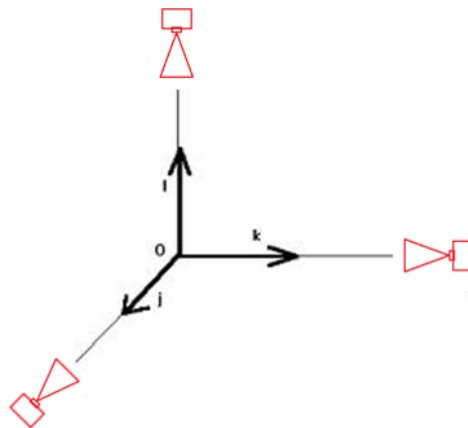


Figure 3.1-c : Position idéale des caméras selon les 3 axes X, Y et Z

3.1.2 Système d'illuminateur et tissu réfléchissant

Un des principaux problèmes d'un système de vision est la lumière qu'il est nécessaire de gérer pour pouvoir acquérir une image qui soit par la suite exploitable.

Pour résoudre ce problème, trois illuminateurs circulaires ont été montés sur les objectifs afin de fournir la lumière nécessaire à la scène. Les illuminateurs choisis émettent à une longueur d'onde de 850 nm, correspondant aux infrarouges et donc invisible à l'œil humain. Les caméras précédemment choisies captent cette longueur d'onde, et l'ajout d'un filtre passe bande autour du 850 nm permet de ne capter que cette lumière. Avec un tel système, il est possible d'obtenir des informations sur une scène, même dans la pénombre. La *figure 3.1-d* montre le système d'illuminateur circulaire et le filtre passe bande.

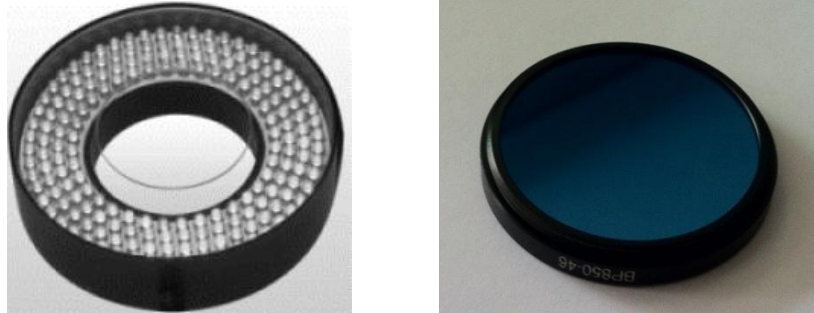


Figure 3.1-d : Illuminateur circulaire et filtre

En accentuant le contraste sur les images acquises, les informations peuvent être mises en avant de manière artificielle. Dans notre cas, l'utilisation de tissu réfléchissant permet de faire réfléchir la lumière émise par les illuminateurs, qui est observée par les caméras à travers le filtre passe-bande. Ci-dessous en [figure 3.1-e](#), est mis en avant l'effet de ce tissu réflecteur sur un calibrateur modifié. La partie blanche est remplacée par ce tissu, qui est captée par la caméra comme étant surexposée, et donc d'un blanc pur. En revanche les points noirs peuvent être soit imprimés, soit ajoutés par-dessus en collant des gommettes noires. Ainsi uniquement le calibrateur est visible, et le bruit créé par l'environnement autour de la scène ne vient pas perturber la calibration.

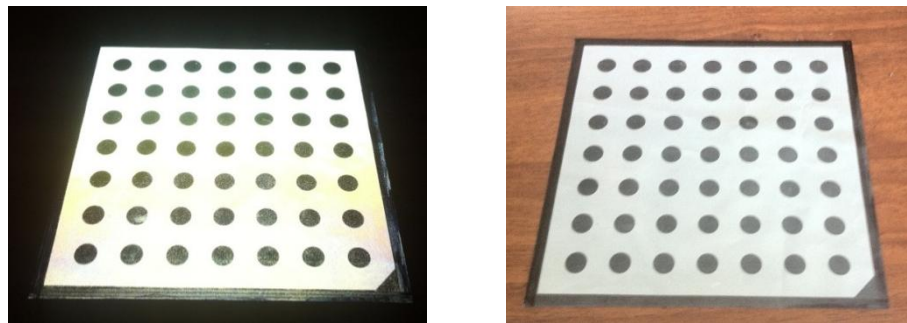


Figure 3.1-e : Photo du calibrateur réfléchissant (avec et sans illumination)

En ce qui concerne le pinceau à retrouver dans l'espace, des points identifiables sont ajoutés à ce dernier, comme en [figure 3.1-f](#). Des sphères, communément appelés « *blobs* » et enrobées d'un même tissu réfléchissant que le précédent calibrateur, permettent de retrouver la position et l'orientation du pinceau.



Figure 3.1-f : Pinceau instrumenté

3.1.3 Choix des blobs

Les « blobs » ajoutés au pinceau existent en différentes tailles. Ces derniers sont des sphères, qui ont la particularité d'être projetées dans n'importe quel plan comme un disque. En cherchant le centre de ce disque sur l'image, on obtient la projection du centre de la sphère dans le plan image de la caméra. Le processus est précis, et la stéréoscopie peut commencer.

Ci-dessous, une étude de cette précision par notre système a été réalisée, en utilisant différentes tailles de blobs (voir *figure 3.1-g*). Positionnant ces derniers de manière statique sur la scène, des séries de 1000 points ont été enregistrées pour chacun des blobs. Ainsi, une étude statistique a été réalisée, mettant en avant la précision, via l'écart-type des données, et ce en fonction du diamètre du blob.



Figure 3.1-g : Blobs de tailles différentes testés

Les résultats sont visibles en *figure 3.1-h*, où l'on observe que l'erreur de mesure semble croître avec l'augmentation du diamètre du blobs. Ceci pourrait être expliqué par le fait que retrouver un plus petit blob et donc le centre d'un plus petit disque sur l'image soit plus précis. Néanmoins un disque trop petit, composé de trop peu de pixels risque de ne pas être reconnu comme tel par l'algorithme. Par conséquent, au vue des résultats, nous avons considéré qu'un blob de diamètre 10 mm serait préférable pour pouvoir être reconnu par notre système.

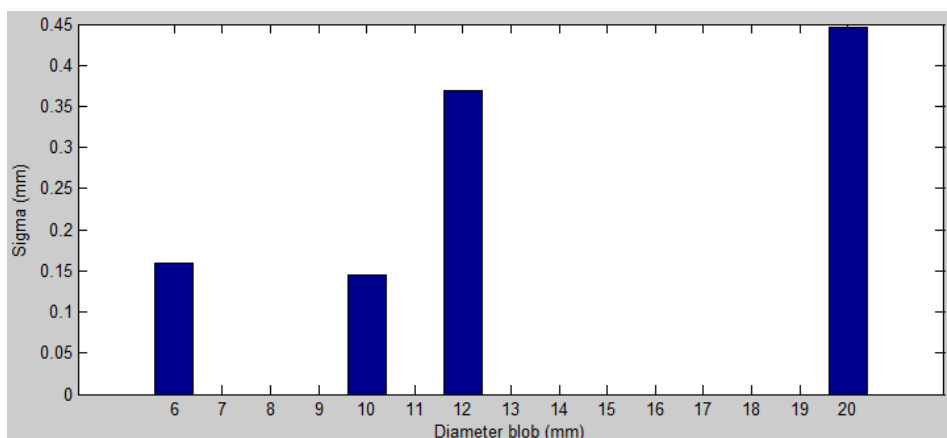


Figure 3.1-h : Ecart type en fonction du diamètre du blob

3.2 Présentation des logiciels

Avant de pouvoir dupliquer les mouvements d'un décorateur, plusieurs étapes et plusieurs logiciels sont à utiliser. La première étape consiste à acquérir les mouvements du pinceau, ou plutôt le mouvement des blobs entourant ce dernier. Une fois ces données enregistrées, ils sont utilisés pour retrouver la position et l'orientation de la pointe du pinceau. Grâce à ces nouvelles données, traitées en cas de besoin, il est possible de les faire lire par le robot, et ainsi de recopier un quelconque mouvement.

Les différentes étapes de transfert des données sont présentées sur le schéma de la figure 3.2-a ci-dessous. De plus, une vidéo d'explication du processus est jointe en annexe.



Figure 3.2-a : Schéma de transfert de données

3.2.1 Acquisition en langage C#

Une ossature de logiciel était déjà disponible pour le laboratoire, écrit en langage « C# », langage dérivé du C, et utilisant des bibliothèques utiles à la géométrie 2D et 3D. Ce logiciel servait uniquement pour la vérification de diverses calibrations, en retrouvant des blobs dans l'espace. Il a été modifié pour intégrer une calibration utilisant l'outil de calibration de HALCON, et utilisé pour comparer la stéréoscopie à deux caméras avec celle à trois caméras. Une capture d'écran du logiciel est disponible en figure 3.2-b.

Lors de la première utilisation du logiciel, alors que les caméras viennent d'être montées, une calibration du système complet est nécessaire. Pour ce faire, une série de 100 images prises simultanément par les trois caméras est enregistrée, présentant le calibrateur dans diverses positions, de manière à recouvrir entièrement toute la surface du plan image de chacune des caméras. Ceci étant fait, le programme se charge d'extraire un fichier contenant tous les paramètres de la calibration du modèle, ainsi que les positions relatives des caméras entre elle. La première triplette d'image acquise impose le repère de référence pour l'acquisition et l'enregistrement des positions des blobs. Une seconde calibration est réalisable prenant uniquement la première image, permettant ainsi de redéfinir ce référentiel de la scène. Une fois le système calibré, il est possible d'acquérir les données de

position des blobs selon ce référentiel, en visualisant les images des trois caméras, les droites épipolaires liées à chacune d'entre elles, et la position trouvée des blobs sur les plans images. Une reconstruction et une visualisation 3D est de même disponible en temps réel. Ainsi, des séquences complètes de mouvement des blobs sont enregistrées.

Pour ce qui est du programme en lui-même, il fonctionne à base de trois threads différents. Le premier consiste en la récupération des informations des blobs projetées sur les plans images, plus exactement des cercles blancs. Le second thread s'occupe de la visualisation des lignes épipolaires et du calcul en temps réel des derniers points acquis, à une fréquence inférieure à 30Hz car couteux en opérations informatiques. Le troisième thread s'occupe des calculs et de l'enregistrement de tous les points les uns après les autres. La décomposition des tâches entre le second et le troisième thread s'explique par le temps de calcul, trop long pour être affiché en temps réel à 30Hz. Le second thread sert donc uniquement à afficher les informations calculées de manière synchrone avec la vidéo.

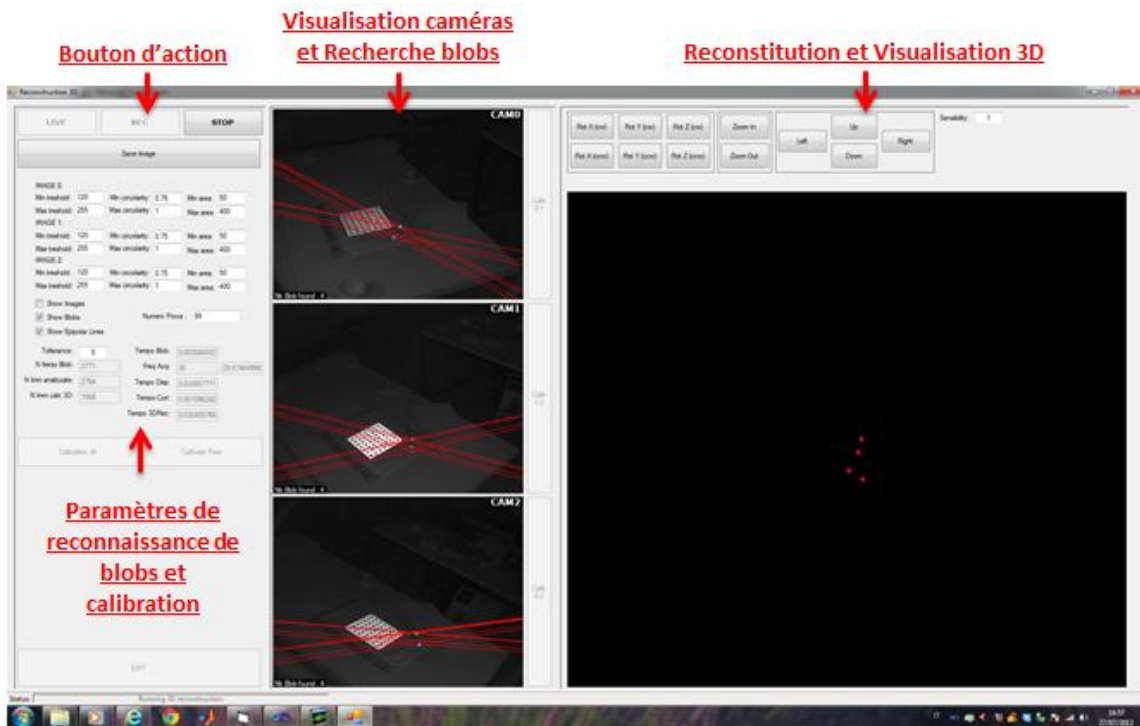


Figure 3.2-b : Présentation du logiciel de récupération des blobs dans l'espace

Au final, un fichier avec toutes positions des blobs est créé, et est utilisé pour le traitement a posteriori des données.

3.2.2 Traitement des données

Une fois les données acquises, il s'agit de les utiliser pour obtenir les informations du mouvement du pinceau. Ce traitement se décompose en trois étapes principales qui sont la réordination des blobs, la recherche de la pointe par recalage rigide (« *Rigid Registration* ») et le filtrage de ces mouvements.

La première étude est celle de la réordination des blobs. Ces derniers sont enregistrés dans un ordre aléatoire, en fonction de leur position dans l'espace. Il faut donc leur donner un ordre précis afin de pouvoir les traiter par la suite. Pour ce faire, on se sert de l'unique information disponible sur ces blobs, leurs distances respectives des uns aux autres. En créant une matrice de distance et en utilisant les informations que l'on peut en tirer, il est possible d'imposer un ordre entre chacun de ces blobs.

Cette matrice est la suivante, généralisée pour l'étude de n blobs fixes les uns aux autres :

$$\begin{bmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & d_{33} & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \cdots & d_{nn} \end{bmatrix} \quad \text{avec : } d_{ij} = \text{distance entre les blobs } i \text{ et } j$$

Une hypothèse, non des moindres, vient s'ajouter au problème, il s'agit de pouvoir réordonner n blobs, et ce même si un nombre m de blobs ne soient visibles, cachés par un quelconque objet. Il faut cependant que trois blobs soient toujours présents pour retrouver le pinceau avec l'algorithme de « *Rigid Registration* », d'où l'hypothèse que $n - m \geq 3$.

Ex: $\begin{bmatrix} 0 & d_{12} & X & d_{14} \\ d_{21} & 0 & X & d_{24} \\ X & X & 0 & X \\ d_{41} & d_{42} & X & 0 \end{bmatrix}$ au lieu de $\begin{bmatrix} 0 & d_{12} & d_{13} & d_{14} \\ d_{21} & 0 & d_{23} & d_{24} \\ d_{31} & d_{32} & 0 & d_{34} \\ d_{41} & d_{42} & d_{43} & 0 \end{bmatrix}$ signifie que le blob 3 est manquant

L'algorithme implémenté se base alors sur une approche expérimentale d'une inversion de 4 blobs dont un des blobs a été supprimé. Il s'agit bien sûr d'exploiter la matrice des distances, en étudiant toutes les informations qu'on puisse obtenir, en comparant chaque élément de la matrice de l'outil avec la matrice réellement obtenue.

L'algorithme implémenté est le suivant, illustré dans la suite par un exemple : Pour chacun des éléments de la matrice à permuter, on recherche sa position réelle dans la matrice de l'outil en omettant la diagonale nulle, afin de créer un vecteur « indice » pour chaque ligne, où sera implémenté le numéro de la ligne et de la colonne de cette distance. S'en suit un traitement de ces indices par ligne permettant de retrouver l'inversion des blobs.

Ex: $\begin{bmatrix} 0 & d13 & d12 & d14 \\ d31 & 0 & d32 & d34 \\ d21 & d23 & 0 & d24 \\ d41 & d43 & d42 & 0 \end{bmatrix}$ au lieu de $\begin{bmatrix} 0 & d12 & d13 & d14 \\ d21 & 0 & d23 & d24 \\ d31 & d32 & 0 & d34 \\ d41 & d42 & d43 & 0 \end{bmatrix}$ signifie que le blob 2 et 3

ont été inversés. Ainsi, en parcourant la 1^{ère} ligne de la matrice, le 1^{er} élément est 0, donc n'est pas pris en compte et retourne [0,0] en indice. Le 2^{ème} élément est d13, se situant en position [1,3] de la matrice de l'outil. Le 3^{ème} élément est d12, retournant [1,2] pour l'indice, et le dernier élément est à la bonne place et retourne donc [1,4]. Le vecteur « indice » obtenu pour la 1^{ère} ligne est alors [0,0,1,3,1,2,1,4]. En réitérant le processus pour les lignes 1 à 4, nous obtenons les 4 vecteurs indices suivants :

indice ligne 1 : [0,0,1,3,1,2,1,4] ; indice ligne 2 : [1,3,0,0,2,3,3,4] ;
 indice ligne 3 : [1,2,2,3,0,0,2,4] ; indice ligne 4 : [1,4,3,4,2,4,0,0] ;

En recherchant dans chaque vecteur l'indice le plus présent, il est possible de retrouver les inversions, pour la ligne 1, l'indice le plus représenté est 1, donc le blob 1 est correctement positionné. Pour la ligne 2, l'indice le plus représenté est le 3, donc le blob 3 est en fait le blob 2. Pour la ligne 3, l'indice le plus représenté est le 2, donc le blob 2 est en fait le blob 3. Finalement, le blob 4 reste inchangé. Ainsi les inversions sont retrouvées.

Cet exemple est généralisable pour n'importe quelle inversion entre n blobs. L'avantage de cette méthode est qu'il est possible de l'adapter lorsque plusieurs blobs sont manquants. Lorsque cela arrive, le vecteur indice de la ligne ne retourne que des 0. Le souci est que si toutes ces lignes et colonnes sont inversées, le vecteur ligne du blob manquant ne retourne aucune information. Cependant, avec la méthode précédemment décrite, on ne fait que réordonner les informations que l'on possède déjà sur les blobs dans leurs bonnes lignes et colonnes, et donc les blobs manquants resteront vides, sans information.

Cette méthode a été utilisée dans notre étude avec des acquisitions à l'aide de trois blobs désordonnés puis avec des acquisitions de quatre blobs avec possibilité d'avoir un blob manquant. Les résultats obtenus sont positifs, un minimum de trois blobs a pu être réordonné, nécessaire par la suite pour paramétrer l'outil dans l'espace.

La seconde étape du traitement des données est la récupération de la position et de l'orientation du pinceau avec les données des positions des blobs. Pour ce faire, il a fallu raisonner sur le principe de la « Rigid Registration » [7]. Cet outil est un processus s'appuyant sur les similitudes, des transformations d'objets géométriques composées d'une rotation, d'une translation et d'un facteur d'échelle. Définissant les objets géométriques par des nuages de points, la « Rigid Registration » s'attaque à la recherche des paramètres de la similitude. La figure 3.2-c illustre le problème, un modèle (2D) et une similitude de ce modèle y sont représentés, et le but est de retrouver ces paramètres en ayant pour seule connaissance les deux nuages de points numéroté de 1 à 5.

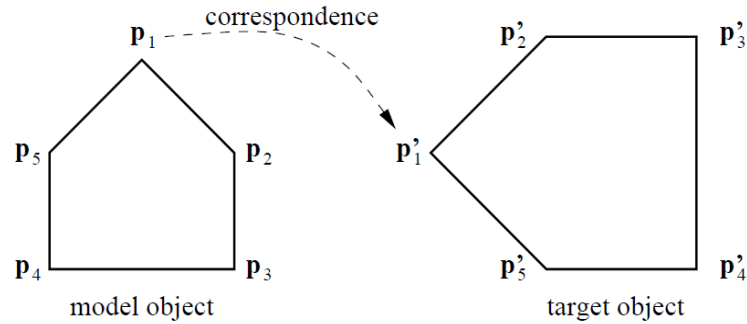


Figure 3.2-c : Exemple d'homothétie : rotation, translation et facteur d'échelle

La question est de savoir comment faire pour retrouver ces paramètres, et c'est là que la méthode de décomposition en valeurs singulières (abrégé « SVD » pour « Singular Value Decomposition ») intervient. La « SVD » est un outil mathématique permettant de décomposer une matrice M quelconque en produit de trois matrices U , S et V , tel que $M = U.S.V$, et avec U et V des matrices contenant ses vecteurs de base orthonormés de M ($U.U^T = I$ et $V.V^T = I$) et S correspond aux valeurs singulières de M . Cet outil, utilisé pour la « Rigid Registration » permet de retrouver position, orientation et facteur d'échelle, suivant l'algorithme relativement simple suivant :

1^{ère} étape : nommons les points du modèle p_i , et les points de la similitude p'_i , avec i allant de 1 à n . Calculons les barycentres \hat{p} et \hat{p}' de chacun des deux nuages de points, modèle et similitude. Calculons ensuite les écarts r_i et r'_i entre chacun des points et leur barycentre.

2^{ème} étape : soit s le facteur d'échelle, il est obtenu par la formule suivante :

$$s = \frac{\sum_{i=1}^n \|r'_i\|^2}{\sum_{i=1}^n \|r_i\|^2}$$

3^{ème} étape : soit M une matrice définie par $M = \sum_{i=1}^n r'_i * r_i^T$. La décomposition SVD de cette matrice M retourne les trois matrices U , S , V , tel que $M = U.S.V$ et la matrice de rotation R de la similitude est obtenue grâce à la formule :

$$R = U \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \text{sign}(\det(U.V^T)) \end{bmatrix} \cdot V^T$$

4^{ème} étape : soit T le vecteur de translation, ce dernier est obtenu avec les valeurs précédentes de s et de R grâce à la formule suivante :

$$T = \hat{p}' - s.R.\hat{p}$$

Il est démontré mathématiquement que les solutions s , R et T obtenues minimisent l'erreur de « fitting » avec la formule des moindres carrés. Dans notre cas, seule la rotation et la translation seront calculées, étant donné que le facteur d'échelle n'entre pas en compte. Il faut savoir que des méthodes de « Non-Rigid Registration » permettent de retrouver des formes en ajoutant la notion de transformation affine et projective d'objet, mais ce thème regroupe une toute autre théorie, bien plus complexe, appliqué par exemple à l'imagerie médicale.

Illustré en *figure 3.1-f*, le pinceau est numérisé à l'aide un ensemble de quatre blobs représentant un nuage de quatre points dans l'espace. Ces blobs ont été placés à des distances différentes les unes des autres, tant pour pouvoir reconnaître leur ordre que pour faciliter le calcul du SVD et ainsi éviter des positions ambiguës. Le nuage de points doit être composé d'un minimum de trois points, et une augmentation de ce nombre améliore la précision. L'unique critère à remplir pour ajouter un blob est de trouver une position pour ce dernier pour qu'il soit à une distance différente de ceux déjà implémenté, afin de ne pas interférer pour retrouver l'ordre des blobs.

Avec cette méthode, nous obtenons le vecteur de translation et la matrice de rotation du pinceau. De cette matrice sont extraits les angles d'Euler pour ensuite être filtrées et transformées en quaternions utilisés par le robot. Le souci lors de ce filtrage est le fait que les angles d'Euler sont modulo 2π , donc non continus lors du dépassement des valeurs au-delà de 2π , rendant ainsi un filtrage incohérent. En revanche, en passant directement par les quaternions, qui ont la propriété de ne pas présenter de discontinuité comme les angles d'Euler, un traitement successif est possible. De plus, le robot ABB utilise les quaternions pour transmettre l'orientation, ce qui est plus commode.

L'ultime étape réalisée avec Matlab est le traitement du mouvement du pinceau, afin de lisser les données et d'avoir un mouvement lisse et épuré. Ce traitement fait l'objet du chapitre suivant, et permet d'obtenir des données de position et d'orientation. La vitesse est aussi calculée grâce aux positions et à la fréquence d'acquisition, et toutes ces données sont enregistrées dans un ultime fichier. Ce dernier est ensuite transféré dans la mémoire interne du robot, relié en « FTP » à un ordinateur, pour être ouvert par un programme implémenté en langage spécifique pour les robots ABB, le langage RAPID. Ce dernier sera décrit au paragraphe 5.1.

Chapitre 4 Filtrage et Interpolation

4.1 Introduction

Grâce au système de vision et au traitement des données associé, il est possible de récupérer dans un fichier les données des coordonnées X, Y et Z de la pointe du pinceau, à intervalle de temps régulier, dans le référentiel du calibrateur. Ce référentiel se retrouve tant sur le système de vision que sur l'aire de travail du robot : il est le repère liant enregistrement et reproduction des mouvements. Les données acquises peuvent donc être exploitées par le robot.

Plusieurs acquisitions ont pu être enregistrées, traitées puis reproduites, et quelques observations ont pu être faites. La première observation a été que le bras robotisé peut reproduire précisément un mouvement enregistré sans problème de rapidité de calcul, à une fréquence de 30Hz, correspondant à la fréquence d'acquisition de la triplette. La deuxième observation d'importance est l'influence du bruit des données qui se traduit par de fortes vibrations au niveau de l'outil du robot. Il a été identifié que ce bruit pourrait provenir à la fois du manque de précision de la triplette et du tremblement de la main humaine dû à des hésitations lors d'un tracé. Ces vibrations sont indésirables et il est souhaitable de les supprimer, mais sans altérer la qualité du mouvement recopié.

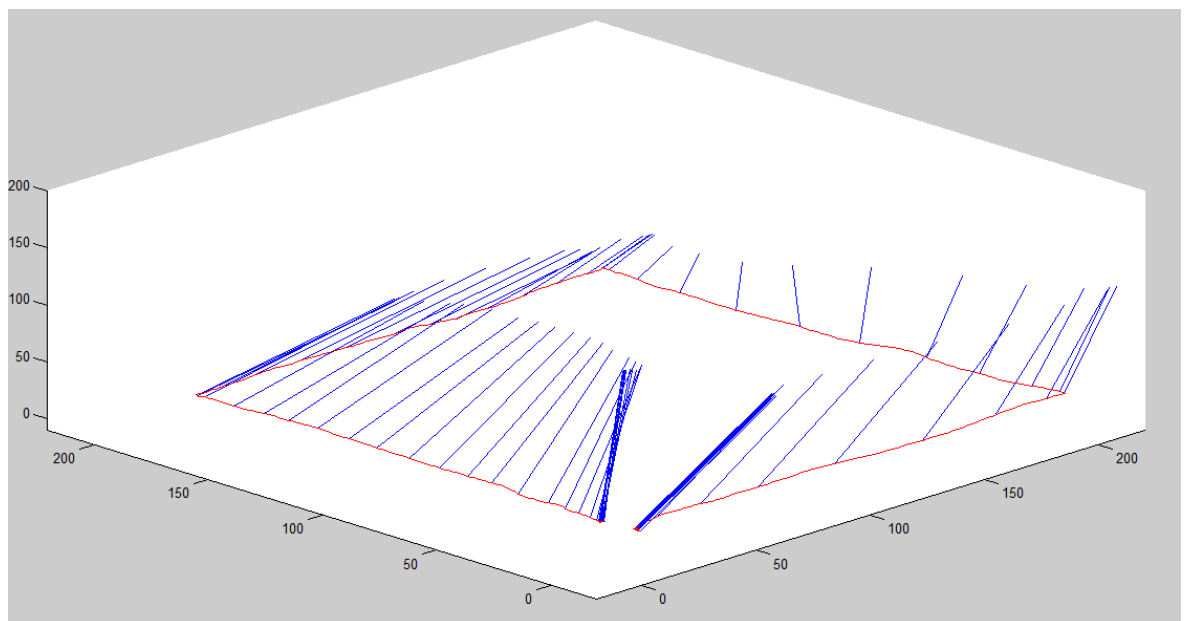


Figure 4.1-a : Position du pinceau (bleu) et trajectoire (rouge)

Ci-dessus en figure 4.1-a, un exemple d'acquisition est illustré. Il s'agit du tracé du contour du référentiel. D'une durée de plus de 20 secondes, se sont presque de 700 points qui ont été enregistré et joint bout à bout. Nous apercevons sur cette figure la trajectoire 3D en rouge, et en bleu l'orientation du pinceau représenté à une fréquence de 3 Hz uniquement pour des soucis de visualisation.

La figure 4.1-b ci-dessous décompose le mouvement du pinceau en position selon les axes X, Y et Z du calibrateur (à gauche), et en orientation selon les quaternions Q1, Q2, Q3 et Q4 (à droite). C'est grâce à la visualisation de la position selon Z que l'on s'aperçoit du bruit présent dans ces données. Ces dernières varient de quelques millimètres, à une fréquence élevée de 30 Hz, ce qui provoque des vibrations indésirables au robot et des accélérations critiques pour les moteurs de ce dernier.

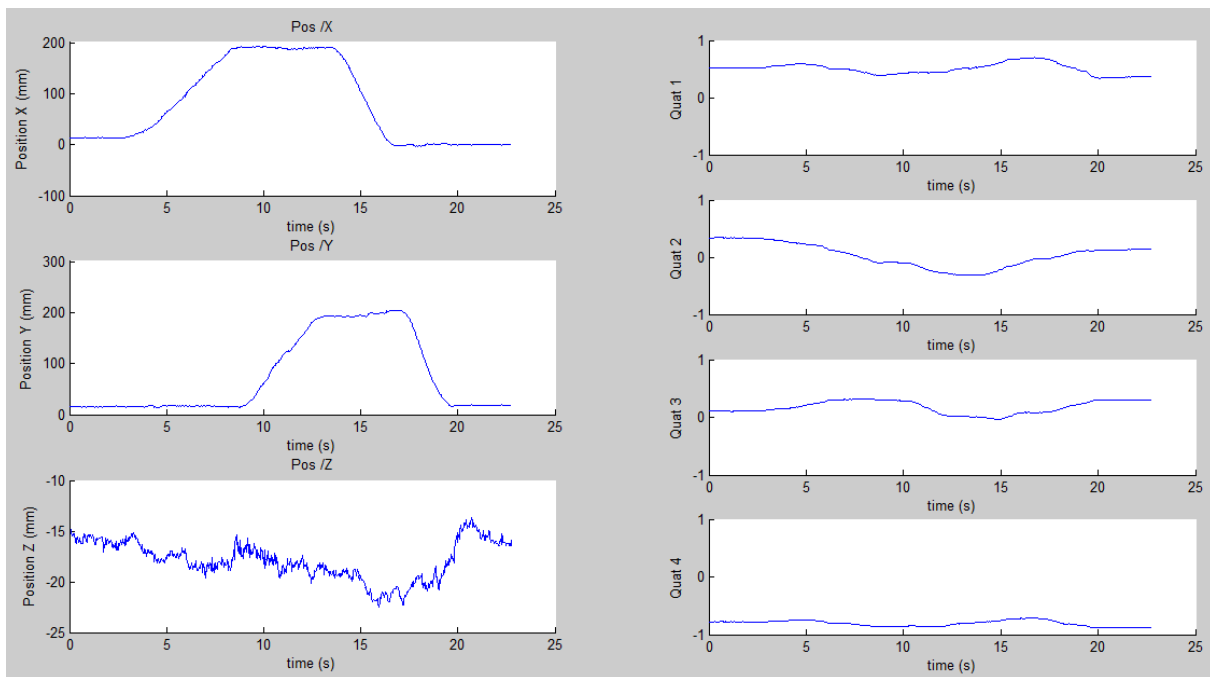


Figure 4.1-b : Position et orientation de la pointe du pinceau

Grâce à cet exemple, l'importance de traiter les données acquises a été mise en avant. S'offre alors à nous deux grandes possibilités pour résoudre ce problème :

- L'étude des filtres
- L'étude d'interpolation de point

4.2 Technique d'interpolation et de filtrage

Les interpolations de points sont peu utilisées pour lisser des données : dans la majorité des cas les filtrages sont préférés. Cependant, ces derniers peuvent s'avérer un outil efficace pour notre étude, car en s'affranchissant d'une étude fréquentielle, nous pourrions obtenir des résultats équivalents. Evidemment, une étude des filtrages donnera une comparaison sur les performances des deux méthodes de traitement de données.

Les techniques d'interpolation utilisent des points de l'espace pour contrôler la courbe traitée. Ces dernières peuvent être divisées en deux familles principales, la première utilise ces points de contrôle comme des points appartenant aux données traitées (*interpolation cubique, Spline*) [8], et la seconde les utilise pour s'approcher au mieux de ces données sans pour autant les atteindre (*courbes de Bézier, NURBS*) [9].

Pour ce qui est des techniques de filtrage, il sera identifié que le bruit présente des composantes à hautes fréquences, tandis que les mouvements de peinture présente plutôt des composantes à faibles fréquences, d'où l'étude des *filtres passe-bas*.

4.2.1 Interpolation cubique

L'interpolation cubique joint n points deux à deux par des polynômes du 3ème degré. Un polynôme du 3ème degré est défini par quatre paramètres, et au total $n-1$ polynômes sont implémentés, qui correspondent donc à $4*(n-1)$ inconnues. Pour chacun de ces polynômes, les positions et les vitesses obtenues par la vision compensent les quatre inconnues des polynômes, soit un total de $4*(n-1)$ équations. En ce qui concerne les vitesses initiale et finale, celles-ci sont choisies au cas par cas. Dans le nôtre elles seront nulles. Avec une telle méthode nous obtenons des courbes comme celles présentés en figure 4.2-a, avec des positions et des vitesses continues, mais des accélérations discontinues, présentant de forts pics.

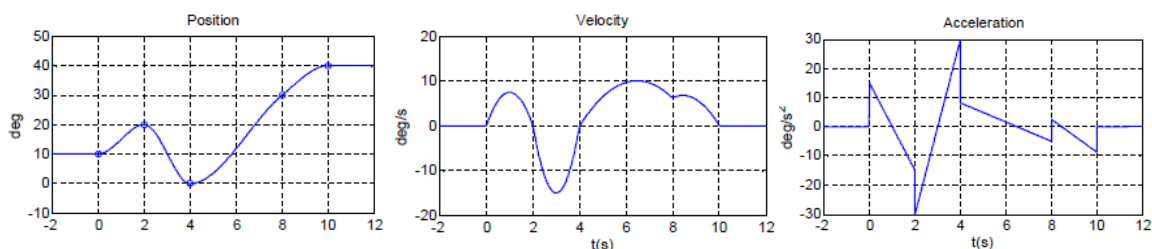


Figure 4.2-a : Exemple d'interpolation cubique

4.2.2 Interpolation par Spline

Le principe de l'interpolation par Spline est simple puisqu'il s'agit d'utiliser des polynômes du 3ème degré entre chaque point, où seule la position est définie par la vision et où l'on tente de réaliser une continuité pour les vitesses et les accélérations. En faisant les comptes, nous avons $4 \cdot (n-1)$ paramètres, $n-2$ conditions de continuité des vitesses, et $n-2$ conditions de continuité des accélérations. Il nous reste donc deux degrés de liberté, utilisés généralement pour attribuer une vitesse initiale et une vitesse finale, dans notre cas nulles.

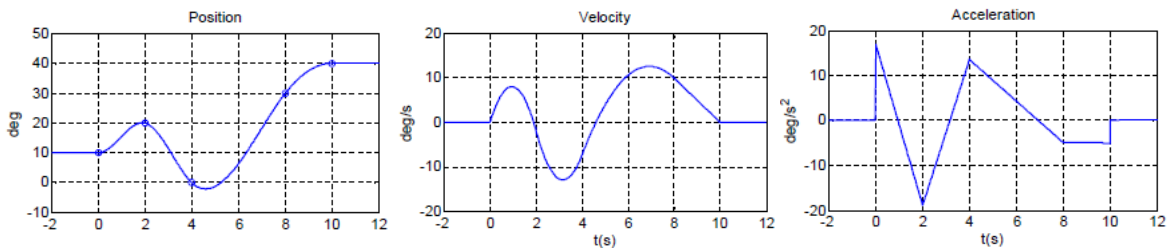


Figure 4.2-b : Interpolation Spline

4.2.3 Interpolation par courbe de Bézier

Le problème avec les interpolations précédentes est qu'elles n'atténuent pas le bruit, mais le reconstruisent tel quel, au mieux à une fréquence différente. L'idée dans ces cas, est d'utiliser des courbes spéciales telles que les courbes de Bézier. Ces courbes ont une méthode plutôt originale de se créer. L'algorithme permettant de tracer une telle courbe est simple et répétitif. Entre chacun des points à interpoler, un segment est créé. De nouveaux points sont créés sur ce segment, à la fraction de la longueur u . Ainsi on diminue de un le nombre de points et on peut recommencer la routine. Au final on obtient un unique point de la courbe, et on réalise la même opération pour une valeur de u différente. Ainsi parcourant u de 0 à 1, on obtient une série de points correspondants à la courbe de Bézier. Ci-dessous, la figure 4.2-d illustre la formation d'une telle courbe.

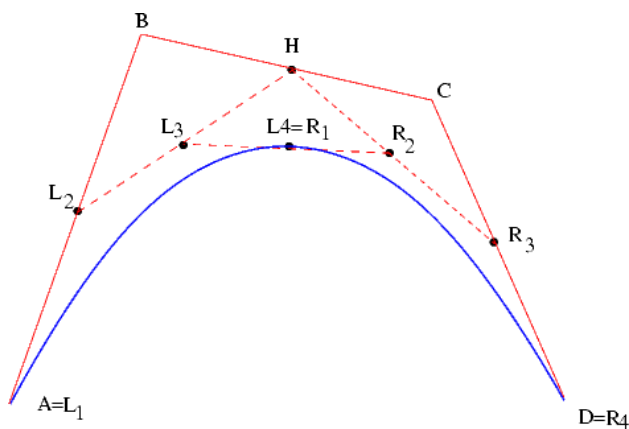


Figure 4.2-c : Tracé d'une courbe de Bézier (en bleu, $u=0.5$)

Cette méthode est difficilement utilisable sur un ordinateur, car peu efficace informatiquement (trop long). C'est pourquoi il existe l'« algorithme de Casteljaou », qui est plus efficace et plus rapide, et qui permet de tracer l'interpolation par courbe de Bézier. L'implémentation de cet algorithme nous a permis de retrouver la courbe de Bézier, à une fréquence souhaitée.

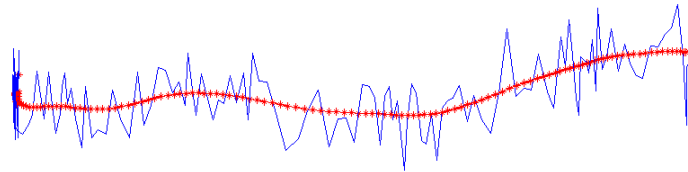


Figure 4.2-d : Interpolation par courbe de Bézier (dans l'espace)

On note directement que la courbe du mouvement est lissée, avec absence d'oscillation entre les points de contrôle. Cette méthode a été testée directement sur le robot et on s'aperçoit que le robot ne tremble plus du tout, ce qui est en accord avec notre objectif. Cependant, avec un grand nombre de points il s'avère que les courbes ne suivent plus les tracés initialement prévus, comme il est possible de le voir en [figure 4.2-e](#).

L'inconvénient principal des courbes de Bézier est qu'elles prennent en compte tous les points de contrôle pour tracer chaque point de la courbe. Or certains points de contrôle ne devraient pas intervenir de la même manière que d'autres, par exemple un point éloigné dans le temps ne devrait pas intervenir autant qu'un point plus proche.

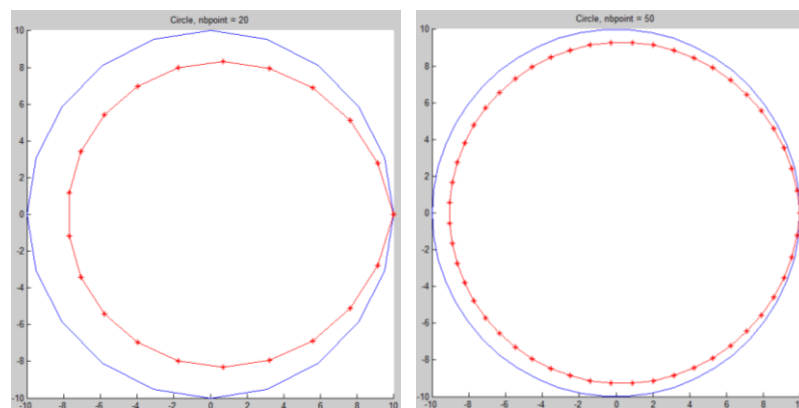


Figure 4.2-e : Interpolation de Bézier d'un cercle avec 20 points et 50 points

4.2.4 Interpolation NURBS

C'est ainsi qu'entre en jeu la famille des NURBS, abréviation de « Non Uniform Rational B-Spline », et qui correspond à une généralisation des courbes de Bézier. Avec ces dernières, il est possible de jouer sur deux paramètres, le poids de chacun des points de contrôle, et une fenêtre de balayage dans le temps qui ne prend en compte que les points dans un certain intervalle mobile plutôt que tous les points à chaque fois.

En jouant sur le poids des points, on parle de « courbes de Bézier rationnelles » (Rational Bézier Curves). Ces courbes permettent le tracé d'arc de cercle parfait, mais dans notre cas cela n'est pas intéressant, car aucun paramètre ne nous permet de dire qu'un point doit avoir plus de poids qu'un autre. Tous les points doivent être répartis de façon égale.

Il nous reste alors à jouer sur la fenêtre de balayage afin de limiter à un certain nombre de points de contrôle voisins d'intervenir sur le calcul des points de la courbe. Dans ce cas on parle de « B-Spline ». L'unique inconvénient de cette méthode est que l'échantillonnage obtenu n'est plus réalisé à période régulière. Ce changement de fréquence d'échantillonnage apparaît au début et à la fin de l'interpolation, sur n points, n étant le degré de la B-Spline.

On réussit alors à avoir des interpolations filtrant le bruit et suivant les mouvements de manière suffisamment proche, et avec lesquelles il est possible de jouer sur un paramètre qui est le degré de la B-Spline.

4.2.5 Filtre Passe-Bas

Finalement, une étude fréquentielle permet de déterminer les caractéristiques d'un mouvement de main humaine, plus précisément du mouvement que le pinceau peut effectuer. Les mouvements retenus pour une peinture sont des mouvements lents, présentant peu d'oscillations rapides, mais plutôt des composantes à fréquence basse. Un filtre passe-bas peut donc être utilisé, pour ne garder que les informations souhaitées. Une fréquence de coupure aux alentours des 3 Hz devrait supprimer le bruit indésirable.

4.3 Confrontation des techniques

Différentes interpolations et filtrage sont maintenant à notre disposition, il est intéressant de les comparer pour déterminer lesquelles seront les plus utiles pour notre application. Pour cela, des tests standards ont été réalisés avec des données se rapprochant au plus de la réalité. En effet, il ne faut pas oublier que les signaux obtenus proviennent d'un mouvement de peinture, et que ceux-ci sont limités par le mouvement humain. Un signal échelon ne sera donc pas réaliste, mais des signaux de types sinusoïdaux, de bruitages ou une combinaison de ces derniers seront plus adaptés. Les interpolations étudiées se limiteront aux cas suivants :

- Spline, la meilleure interpolation passant par les points de contrôle,
- Courbes de Bézier, qui sont en réalité un cas particulier des NURBS prenant en compte toutes les données sans pondération,
- B-Spline de degré 50, soit une NURBS utilisant une fenêtre de balayage de 50 échantillons,
- Filtre passe bas du 1^{er} ordre avec une fréquence de coupure à 3 Hz.

Les interpolations cubiques n'ont pas été prises en compte car n'apportant rien de plus par rapport à une interpolation par Spline. De plus, les filtrages ont été limités à l'étude d'un passe bas du 1^{er} ordre, un 2nd ordre n'apportant rien de plus. Enfin, chaque essai a été réalisé à une fréquence de 30 Hz, soit la fréquence d'acquisition du système de vision, et est transmis à cette même fréquence au robot. La courbe bleue définit la référence à approcher, tandis que la courbe rouge correspond aux données traitées.

4.3.1 Test d'un mouvement sinusoïdal ($T = \pi$)

Le mouvement sinusoïdal est l'élément de base d'un mouvement de pinceau. Une hypothèse est émise, considérant que les mouvements de peinture peuvent être décomposés en fonctions trigonométriques de basses fréquences.

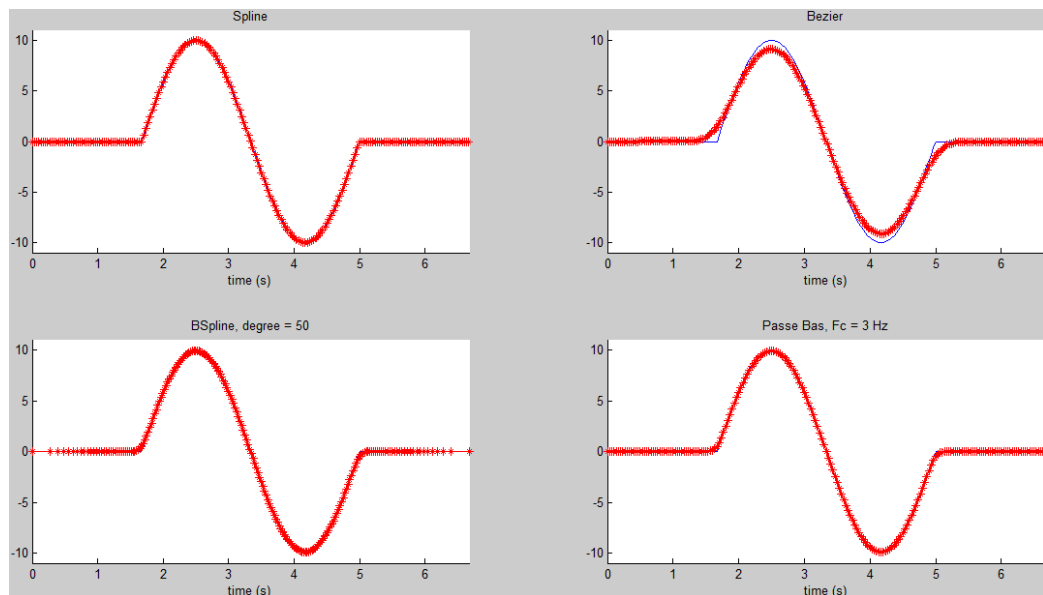


Figure 4.3-a : Mouvement sinusoïdal

On remarque donc sur la figure 4.3-a que les filtrages répondent bien à ce signal de base, à l'exception de la courbe de Bézier, qui réduit son amplitude. En ce qui concerne la B-Spline et le filtre passe-bas, ces deux courbes approchent la courbe de référence de manière suffisamment précise pour ne pas qu'ils soient remarqués de différences. En revanche la Spline est l'unique interpolation passant exactement par les points de contrôle, et donc la plus précise.

4.3.2 Test d'une combinaison de mouvement sinusoïdal ($T1 = \pi$, $T2 = \pi/2$, $T3 = \pi/3$)

La figure 4.3-b permet de compléter l'étude des mouvements sinusoïdaux puisqu'encore une fois la courbe de Bézier s'écarte complètement de la courbe de référence. La Spline quant à elle passe exactement pas les points de contrôle donc permet une reproduction parfaite, tandis que la B-Spline et le filtre passe-bas permet une reproduction suffisamment fidèle du signal de référence.

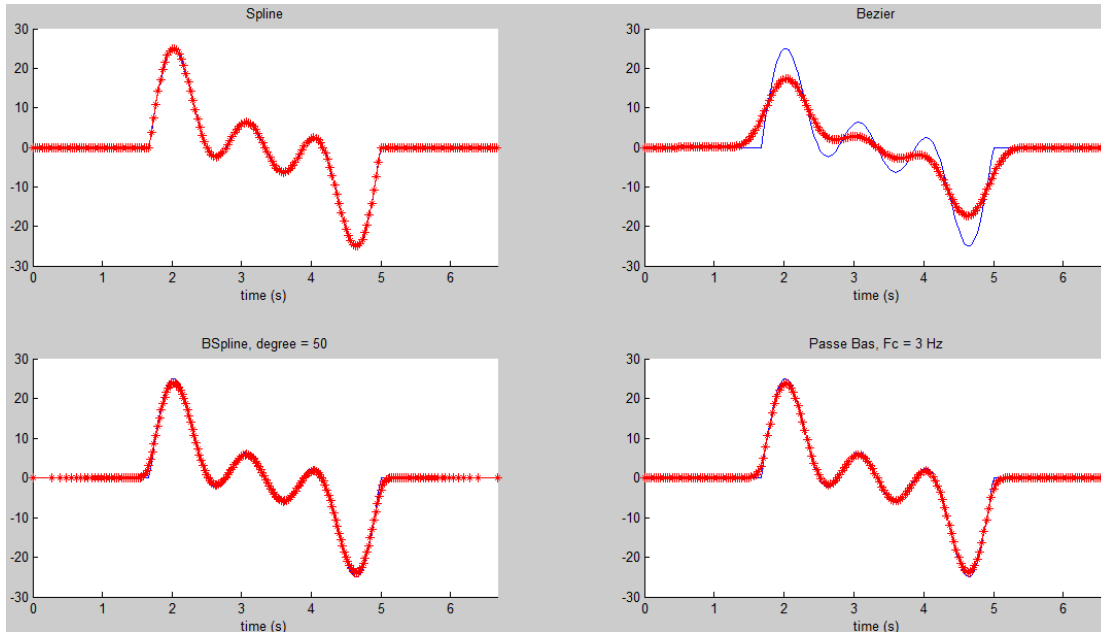


Figure 4.3-b : Combinaison linéaire de mouvements sinusoïdaux

4.3.3 Test d'un bruit blanc

L'étude du bruit blanc est différente de celui de la reproduction d'un mouvement sinusoïdal. En effet, un bruit est rarement le bienvenu et dans notre cas pose un problème de vibration du robot. Le choix du filtrage des données doit donc permettre de le supprimer.

La figure 4.3-c illustre l'incapacité d'une interpolation par Spline à supprimer un bruit. En effet, en passant par chacun des points, il est impossible de lisser un bruit. En revanche, et c'est ici que la courbe de Bézier possède une caractéristique particulière qui élimine le bruit. La courbe de Bézier est une B-Spline de degré égal au nombre de points de contrôle. En réduisant ces points de contrôle à 50 (il y a 200 points dans nos essais) on obtient une courbe moins lisse mais tout aussi convenable. Finalement, l'utilisation d'un filtre passe-bas avec une fréquence de coupure à 3 Hz permet d'obtenir un résultat quasi similaire à la B-Spline.

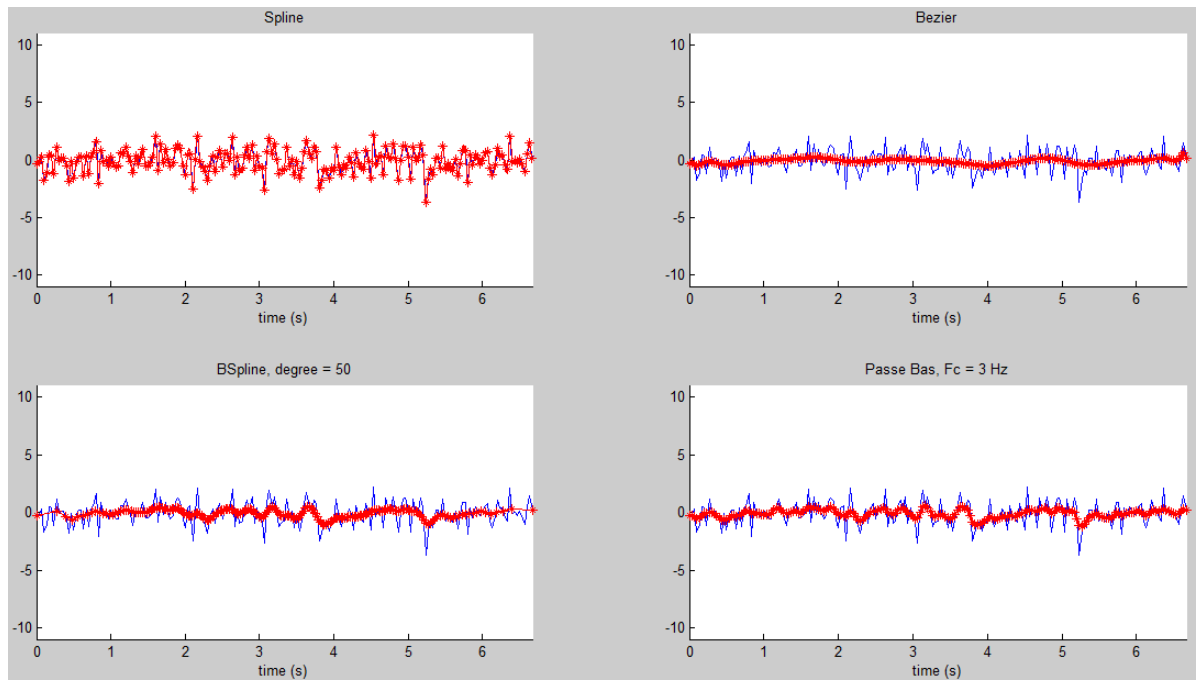


Figure 4.3-c : Signal constant bruité par un bruit blanc

4.3.4 Test d'une combinaison d'un mouvement sinusoïdale ($T = \pi$) avec un bruit blanc

Ce dernier test, se rapprochant le plus à une acquisition réelle de données, nous permet de faire des commentaires généraux concluant ces essais. Tout d'abord l'utilisation d'une Spline simple n'est pas du tout conseillée pour notre application. Elle ne filtre aucun bruit, ne supprime pas les vibrations. Ensuite pour ce qui est de l'utilisation de la courbe de Bézier, elle serait parfaite pour des mouvements continus ou courts, car élimine parfaitement le bruit, cependant, nous avons à faire avec des mouvements longs et variables dans le temps. Finalement, l'utilisation de B-Spline et de filtre passe-bas permet un compromis entre élimination du bruit et suivi d'une trajectoire. De plus, ces dernières sont paramétrables, l'une par un degré et l'autre par une fréquence de coupure. Une préférence va cependant au filtre passe bas, qui possède un échantillonnage dans le temps qui est constant. Il est cependant possible de combler cette différence qu'à la B-Spline en ajoutant artificiellement des points initiaux et finaux comblant l'irrégularité d'échantillonnage à ces zones critiques.

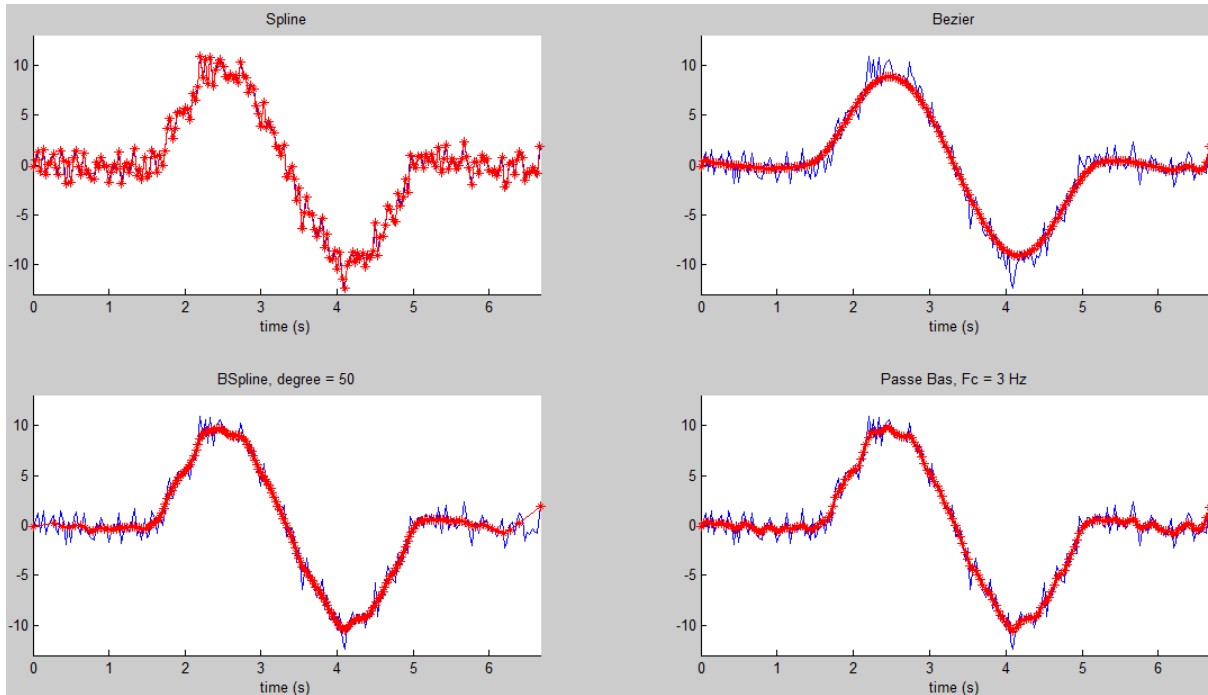


Figure 4.3-d : Mouvement sinusoïdal avec bruit blanc

4.4 Conclusions

Cette étude des traitements de données réalisés sur des mouvements-types a pu mettre en avant une interpolation et un filtrage convenables pour notre étude. Il s'agit de la B-Spline de degré 50 et du filtre passe-bas à 3 Hz. Ces deux derniers ont été testés sur un essai en condition réel, et les résultats sont visibles dans les sous-sections suivantes.

Sur les figures 4.4-a et 4.4-c, en vert sont représentées les trajectoires brutes non-traitées et en rouge les trajectoires traitées. En bleu sont représentées les orientations du pinceau à divers instants. Les trajectoires traitées sont relativement écartées de leur trajectoires originelles, surtout lorsque des vitesses importantes entrent en jeu, pour recharger le pinceau en peinture par exemple. Cependant, lors du mouvement du pinceau sur la feuille, pour peindre, les mouvements sont bien plus lents et sont quasi confondus avec le mouvement acquis.

Sur les figures 4.4-b et 4.4-d, les mouvements sont décomposés en position et en orientation, pour être traités séparément. Le filtrage et l'interpolation effectuent de très bons traitements, traduits par une absence de vibrations sur le robot. On notera cependant que pour l'interpolation par B-Spline, les fréquences d'échantillonnages initial et final ne sont pas constantes, tandis que le filtre passe-bas conserve la fréquence de 30 Hz sur toute la durée. C'est uniquement pour cette raison que pour la suite du projet nous utiliserons des filtres passe-bas pour le traitement des données.

4.4.1 B-Spline de degré 50

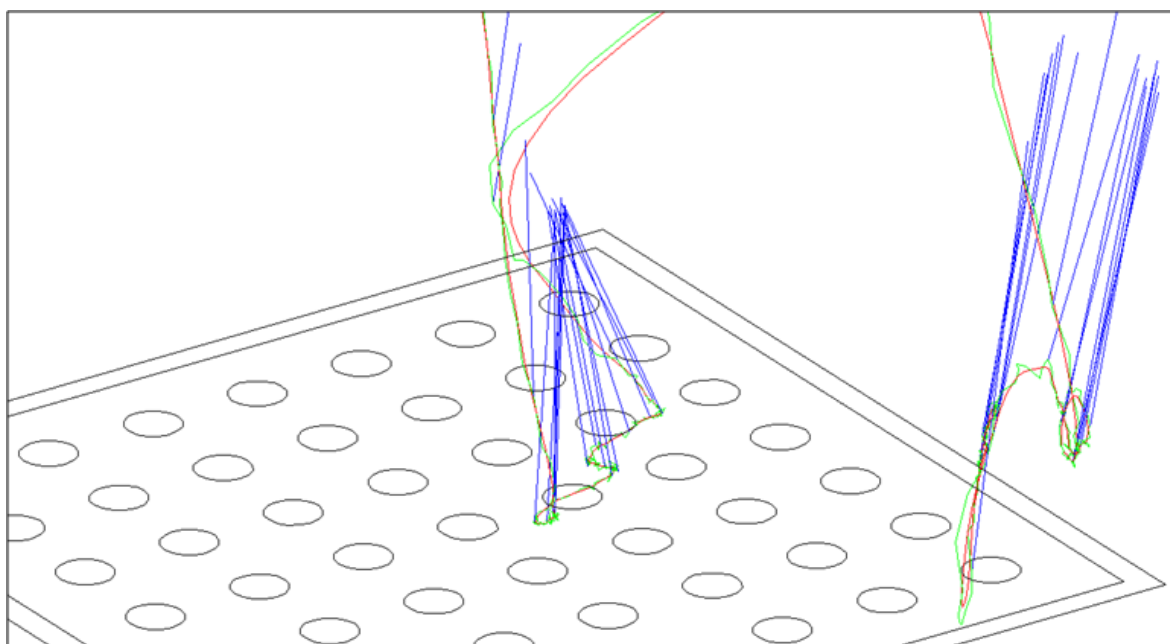


Figure 4.4-a : Mouvement interpolé par B-Spline

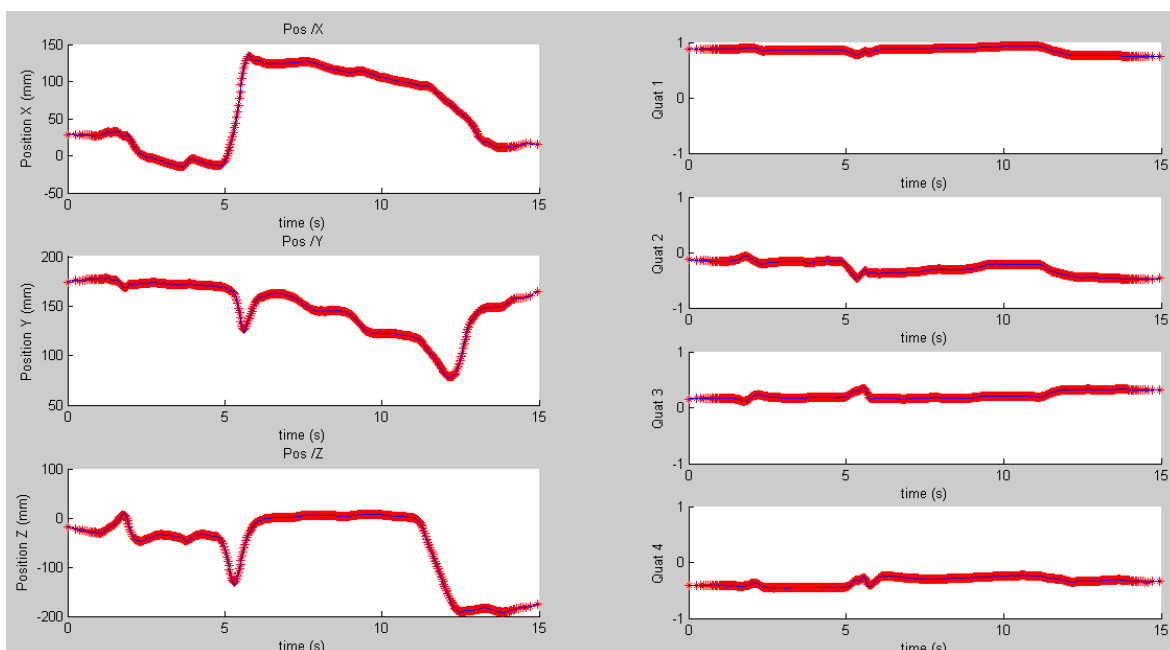


Figure 4.4-b : Composante du mouvement interpolé par B-Spline

4.4.2 Filtre passe-bas 3Hz

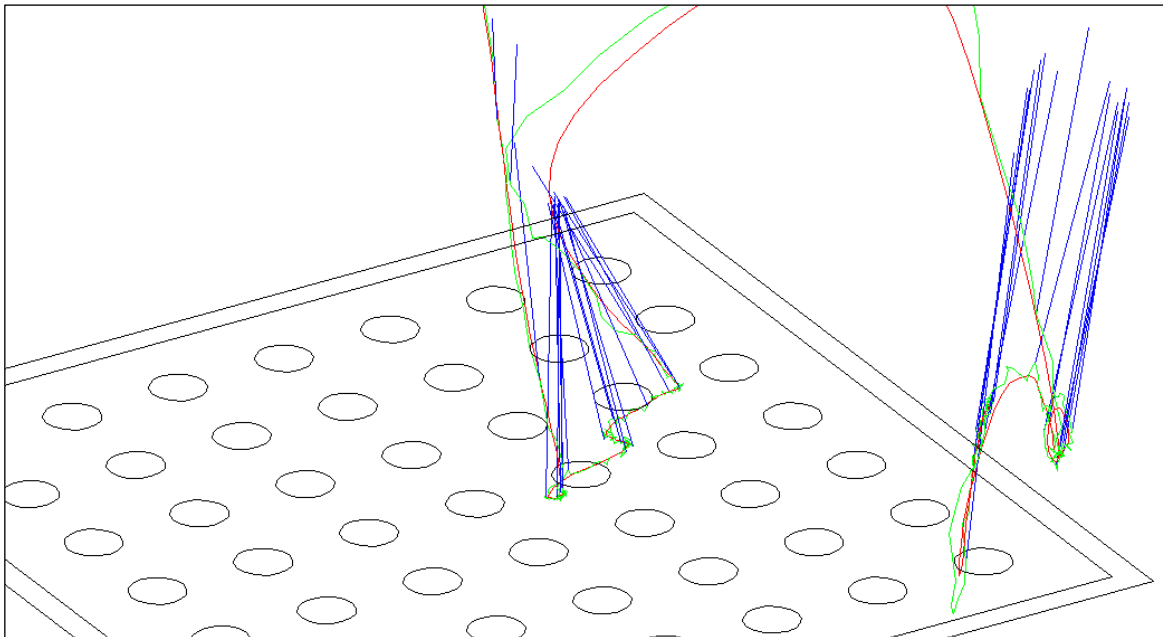


Figure 4.4-c : Mouvement filtré par un passe-bas

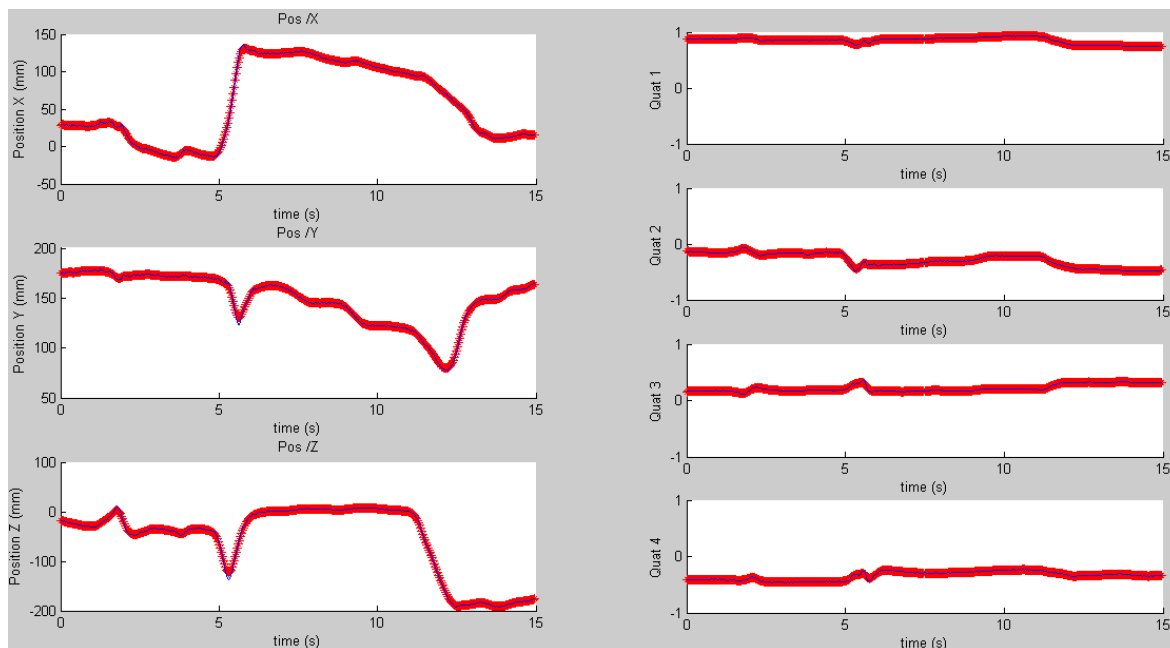
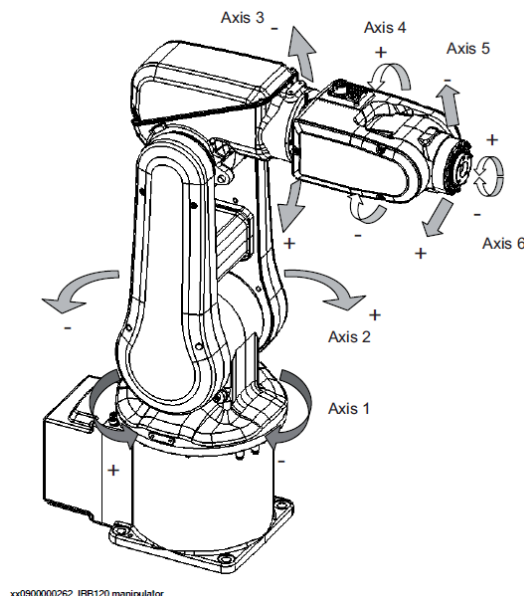


Figure 4.4-d : Composante du mouvement filtré par un passe-bas

Chapitre 5 Experimentations

5.1 Présentation de la cellule robotique

Le robot utilisé pour le projet est un « IRB 120 ». Il est le dernier arrivé de la 4^{ème} génération de robot ABB, et fait de lui le plus petit robot des gammes disponibles, avec un bras de 580 mm et une charge maximale supporté de 3 kg. Il est idéalement utilisé pour la manipulation et l'assemblage de composant, au vue de sa compacité et de son agilité. Un contrôleur a spécialement été conçu pour ce dernier, le « IRC5 Compact », qui comme son nom l'indique, est plus compact qu'un contrôleur normal. L'architecture du robot est présentée ci-dessous en figure 5.1-a.



Amplitude angulaire des axes :

Axe 1 : de -180° à 180°

Axe 2 : de -90° à 150°

Axe 3 : de -180° à 75°

Axe 4 : de -400° à 400°

Axe 5 : de -125° à 120°

Axe 6 : de -400° à 400°

Figure 5.1-a : Architecture du robot IRB 120

Ce robot est installé dans une cellule de dimension 1200x1400mm, où une protection en plexiglass l'isole de l'utilisateur. Au bout de ce robot peut être fixé différents types de pinceau, interchangeable pour une plus grande possibilité d'essais et défini comme outil pour le robot. Le support de travail est rehaussé par rapport à la base du robot, pour ainsi éviter des positions singulières, en particulier l'alignement des axes 4 et 6. Le même calibrateur utilisé pour le système de vision y est installé, mais recouvert sur la figure 5.1-b par un support papier canson de dimension 200x200mm, modélisant le carreau. Un récipient contenant la peinture y est aussi installé, ainsi qu'un tissu permettant d'absorber une charge trop importante de peinture sur la pointe du pinceau.



Figure 5.1-b : Cellule robotisée implémentée

Un travail avait déjà été réalisé par ABB, lors du lancement du projet, imaginant une installation possible de l'îlot de travail (voir [figure 5.1-c](#)). Bien évidemment, celle-ci viendra modifiée ou adaptée selon les demandes de l'industrie.

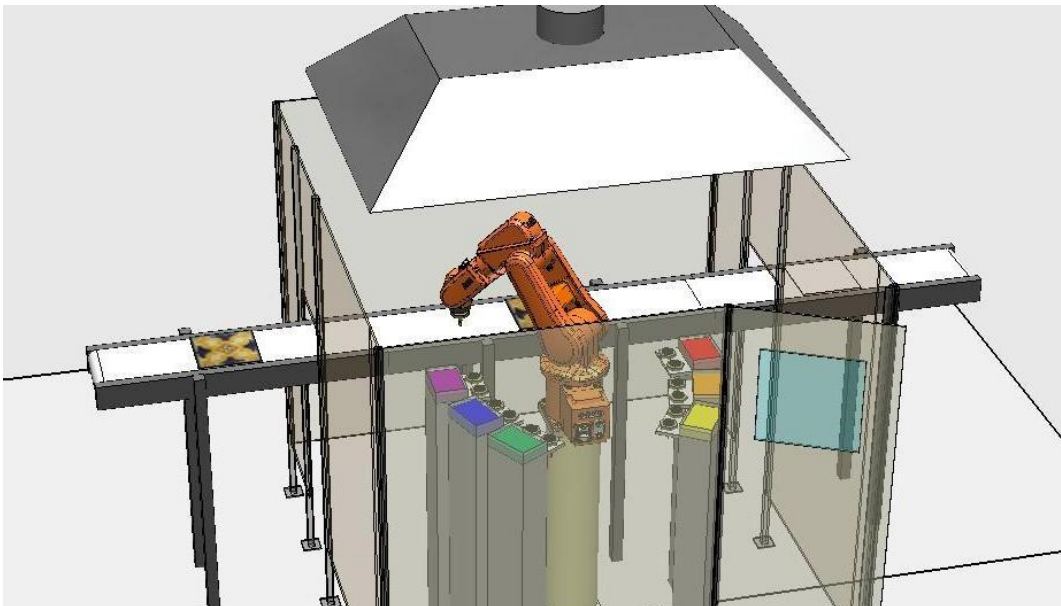


Figure 5.1-c : Îlot de travail imaginé par ABB

La programmation du robot est faite en langage RAPID, un langage propre aux robots ABB, s'assimilant au langage des machines-outils à commande numérique. Les données des mouvements étant précédemment enregistrées dans la mémoire interne du robot, le programme du robot se charge d'ouvrir ces derniers, en choisissant la référence de l'essai et le filtrage souhaité. Ainsi le robot a accès à une bibliothèque complète de mouvements, qu'il est possible de concaténer les uns aux autres. Parmi ces données figurent bien évidemment la position, l'orientation et la vitesse des trajectoires de l'outil et le contrôleur se charge de traiter en interne la cinématique inverse du robot.

5.2 Essai avec décoratrice

Une rencontre avec une restauratrice d'œuvre d'art et peintre dans son temps libre a pu être réalisée. Cette personne a pu nous enseigner les techniques utilisées pour la peinture sur céramique, en nous montrant des exemples sur papier canson et en utilisant de l'aquarelle. Cette technique semble être quasi similaire à celle utilisée sur céramique, ou en tout cas les effets utilisés le sont.

Un élément exploité par cette technique est la quantité de peinture présente sur la pointe. Comme pour l'aquarelle, la quantité de peinture appliquée sur le support absorbant conditionnera la clarté ou l'obscurité d'un trait. Pour ce contrôle de nuance de couleur, il a fallu se fier à la répétabilité d'un tel mouvement, en mettant à disposition le même environnement de travail au robot qu'à la décoratrice, c'est-à-dire un niveau de peinture équivalent dans les deux barquettes de peinture, mais aussi de quoi essuyer un trop-plein de peinture à l'aide d'un tissu absorbant, positionné au même endroit.

Il s'agit aussi d'exploiter le niveau d'écrasement de la pointe du pinceau sur le support, élément lié bien évidemment à la force appliquée. Pour cela, ce n'est pas la pression qui est contrôlée, mais bien la position. Etant donné que la pointe du pinceau est souple, celle-ci se déforme en fonction de son écrasement sur le support, et comme le robot peut fournir jusqu'à 30N de force, le mouvement ne sera pas altéré.

Ci-dessous en ***figure 5.2-a*** sont présentés les différents effets recherchés par une telle peinture.

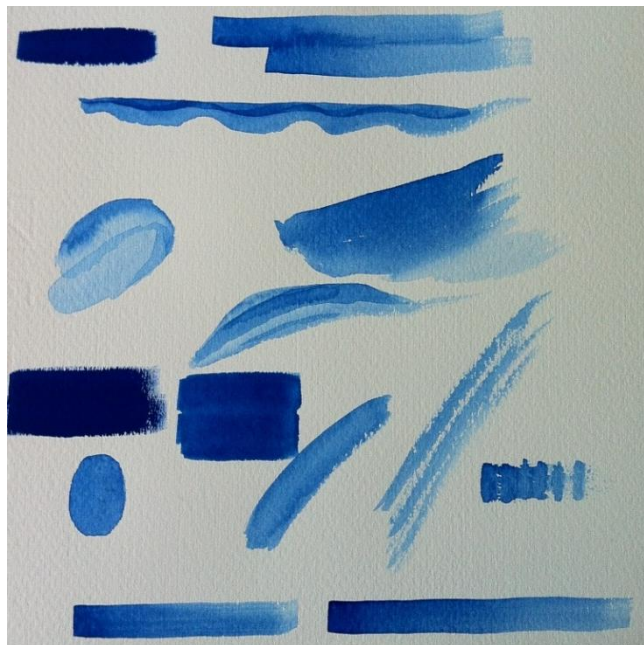


Figure 5.2-a : Effet recherché avec une peinture aquarelle et sur carrelage

Plutôt que de raconter comment se sont déroulés les essais, deux vidéos sont disponibles en annexe, expliquant le processus pas à pas, et résumant les essais réalisés. Ces deux vidéos ont été réalisées pour ABB Robotics Italy dans le but de présenter le projet à une personne extérieure à celui-ci.

La **figure 5.2-b** présente les résultats obtenus par la décoratrice (à gauche) et ceux obtenus par le robot (à droite)

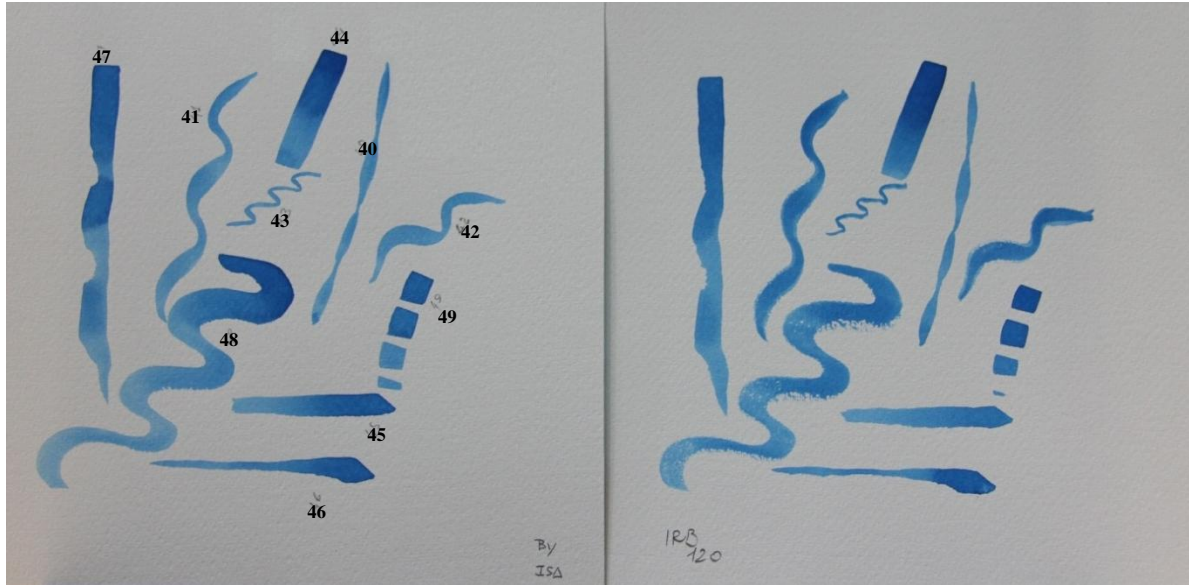


Figure 5.2-b : Confrontation des résultats des essais (numéroté de 40 à 49)

Ces essais ont été réalisés avec 2 types de pinceau, le 1^{er} à pointe ronde et fine (essais 40 à 43) et le 2nd à pointe plate (essais 44 à 49). Divers mouvements ont été testés, s'assimilant le plus généralement à des coups de pinceau utilisés lors de décorations.

Pour les essais avec la pointe fine, nous pouvons voir que les écrasements du pinceau ont été respectés, créant ainsi un grossissement du tracé (essai n°40-41-42). La précision en position est visible sur l'essai 43, où seul le bout de pointe vient toucher le support.

Concernant les essais avec la pointe plate, nous avons pu tester la nuance de couleur (essais 44 et 49). La pointe plate est plus délicate à traiter, car l'orientation du pinceau entre en jeu. Les essais 45 à 47 illustrent la précision des rotations de pinceau, élargissant ou rétrécissant le trait.

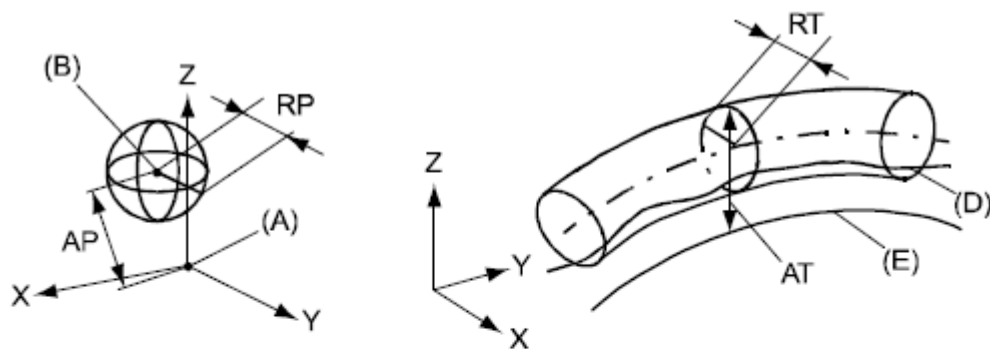
L'essai 48 en revanche mérite une explication sur la qualité reproduite. Etant donné qu'il est l'un des derniers essais réalisés, le niveau de peinture dans la barquette avait diminué, et donc la quantité de peinture sur le pinceau n'était plus suffisante pour une reproduction fidèle. Cet essai met en avant le problème du contrôle du niveau de peinture.

Au vu des résultats plus que satisfaisants des essais, nous pouvons conclure positivement sur la faisabilité du processus, c'est-à-dire de l'utilisation de la stéréoscopie et d'un marquage du pinceau à l'aide de sphères réfléchissantes, pour la reproduction de mouvements de décoration par peinture.

Chapitre 6 Analyse métrologique

6.1 Performances conformes à la norme ISO 9283

Afin de caractériser les performances de notre système de reproduction de mouvement, la norme ISO 9283 a été utilisée. Cette norme est utilisée pour caractériser toute la gamme de robot anthropomorphe ABB, définissant les concepts de précision et de répétabilité d'une position et d'une trajectoire (voir figure 6.1-a).



xx0800000424

Rep	Description	Rep	Description
A	Position programmée	E	Trajectoire programmée
B	Correspond au positionnement durant l'exécution du programme	D	Trajectoire réelle à l'exécution du programme
AP	Correspond à la distance de la position programmée	AT	Déviation max. par rapport à E
RP	Tolérance de la position B au positionnement répété	RT	Tolérance de la trajectoire à l'exécution répétée du programme

Figure 6.1-a : Définition de précision et de répétabilité par la norme ISO 9283

ABB nous garantit les prestations suivantes pour son robot IRB 120, données en datasheet :

	Position (*P)	Trajectoire (*T)
Répétabilité (R*)	0.01 mm	0.07 – 0.16 mm
Précision (A*)	0.02 mm	0.21 – 0.38 mm

Les spécifications du robot semblent suffisantes pour reproduire une trajectoire, et que l'écart entre position programmée et position réelle ne puisse être remarquée à l'œil nu.

Seulement, à cette incertitude il faut ajouter celle de la vision stéréoscopique, qui ne devrait pas être aussi performante que celle du robot. Les performances du système de vision n'étant pas données, nous avons donc décidé de la calculer par nous-même en assemblant le robot à la vision, et en définissant un système de référence commun. Ainsi nous pouvons avoir la précision et la répétabilité de la position récupérée par les caméras, qui viendront s'ajouter à celle du robot.

6.2 Précision du système complet

La précision du système de vision est mesurée grâce à l'emploi de la SVD, appliquée à des points connus et imposés par le robot. Comme expliqué précédemment, le système de vision stéréoscopique vient mesurer directement les mouvements de l'outil du robot, par rapport à son référentiel, mis en commun entre les deux systèmes. Un simple blob est installé comme outil pour le robot, pouvant être mesuré par le système de vision.

Le robot effectue trois mouvements de translation partant de l'origine repère, et en dessinant ses trois axes. Pour rappel l'axe Z est l'axe normal au calibrateur, dirigé vers le bas. Quatre positions viennent alors être mesurées, pendant deux secondes, soit une acquisition de 60 points par position. Ces dernières sont les positions $(0, 0, 0)$, $(200, 0, 0)$, $(0, 200, 0)$ et $(0, 0, -200)$ par rapport au référentiel commun. La trajectoire effectuée est visible ci-dessous en [figure 6.1-b](#).

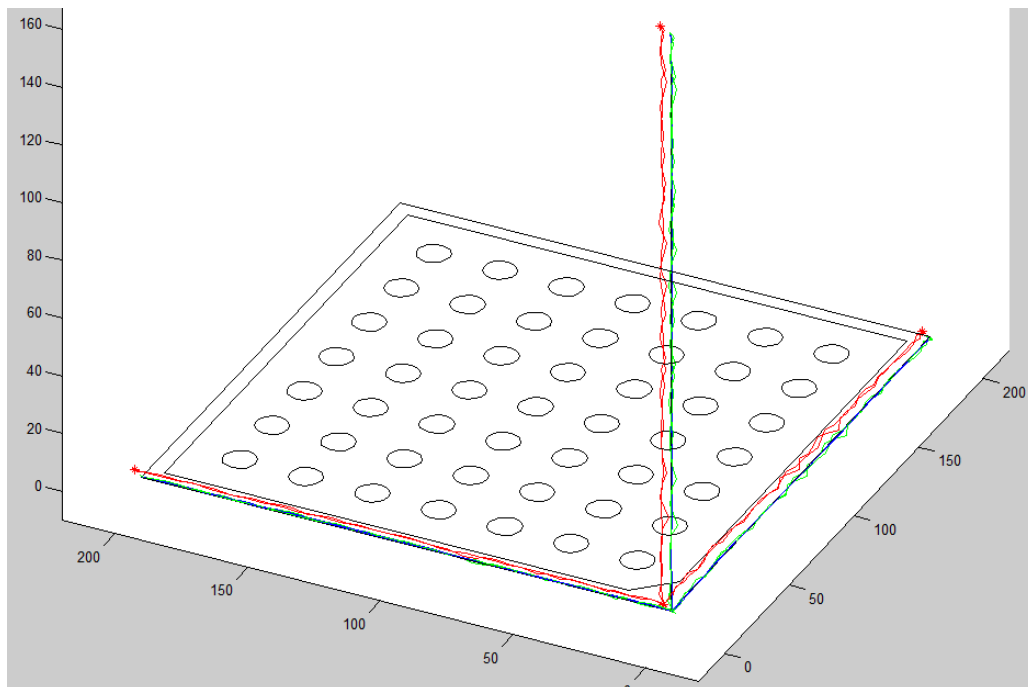


Figure 6.2-a : Trajectoire suivant les axes du référentiel commun

En rouge est représentée la trajectoire mesurée, avec le bruit qu'elle présente, vu qu'elle n'a pas encore été filtrée. On remarque que cette dernière présente un offset en position, puisque l'origine mesurée ne correspond pas exactement à l'origine du calibrateur. Cet offset est considéré comme la précision de l'ensemble du système de vision et du robot. En utilisant l'algorithme de « Rigid Registration » appliqué aux quatre points précédemment présentés, il est possible de retrouver cet offset en position et en orientation. Les erreurs de positions calculées sont alors de $3,7 \text{ mm}$ selon l'axe X, $0,7 \text{ mm}$ selon l'axe Y, et $1,2 \text{ mm}$ selon l'axe Z. Les erreurs d'orientations sont quant à elles négligeables, de l'ordre du dixième de degré en utilisant les angles d'Euler. L'erreur d'offset générale est donnée par la formule suivante :

$$Err = \sqrt{ErrX^2 + ErrY^2 + ErrZ^2}$$

Qui donne une précision égale à $3,95 \text{ mm}$.

Tandis que les précisions selon les axes Y et Z sont encore acceptables, la précision si faible selon l'axe X mérite une explication. C'est la position de la triplette par rapport au calibrateur qui donne de tels résultats. En effet, la triplette est positionnée de manière quasi-normale au plan (O, \vec{y}, \vec{z}) , et il a été défini que l'incertitude maximale du système stéréoscopique dans sa configuration actuelle serait présente suivant les axes optiques des caméras, correspondant à l'axe X du calibrateur.

Une nouvelle configuration des caméras alignant les axes optiques avec les axes X, Y et Z du calibrateur donnerait des meilleurs résultats, cependant l'absence d'une structure permettant l'installation d'une telle configuration et sa calibration ont fait défaut. Néanmoins, il s'avère que cet offset est constant une fois l'installation fixe, il est donc possible de le supprimer a posteriori.

Une fois cet offset supprimé, nous obtenons la trajectoire verte de la [figure 6.2-a](#), qui une fois traitée par un filtre passe-bas à 3 Hz, donne la trajectoire bleue, semblant être relativement bonne. Pour justifier la validité de cette trajectoire filtrée, un test de répétabilité a été réalisé.

6.3 Répétabilité du système complet

La répétabilité du système de vision est calculée en plaçant le même outil précédemment présenté à l'origine du référentiel commun. Evidemment, le même offset précédemment observé est présent, mais il est aussi possible d'étudier la répétabilité de la mesure, en en réalisant un grand nombre.

Pour cette dernière étude, 10 000 points ont été mesurés et filtrés, desquelles il a été possible d'extraire les écarts-types, permettant de justifier la répétabilité du système complet.

En **figure 6.3-a** sont illustrées les courbes des gaussiennes obtenues par le traitement statistique.

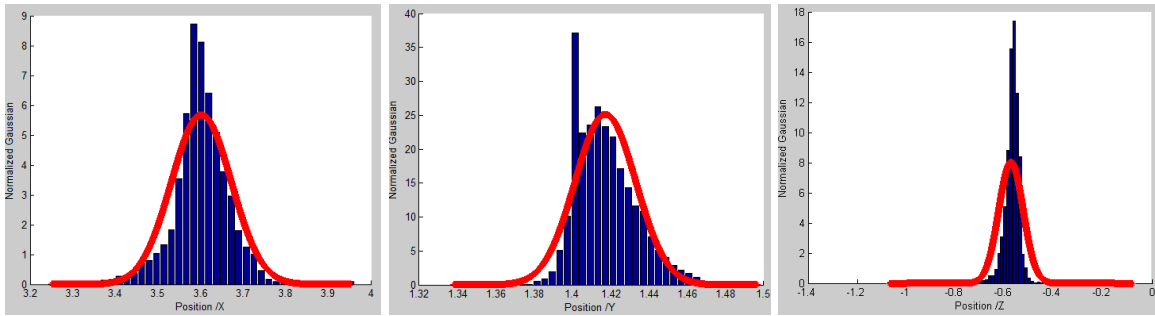


Figure 6.3-a : Gaussiennes des positions suivant X, Y et Z

Pour passer de l'analyse statistique à l'incertitude et donc la répétabilité dans notre cas de figure, il suffit de multiplier l'écart-type par un facteur de couverture (*coverage factor*). En prenant un facteur de 2, nous pouvons affirmer qu'un intervalle de confiance à 95% couvrira notre estimation de l'erreur. En poussant ce facteur à 3, cet intervalle passera à 99%. En couvrant l'incertitude avec ce facteur de 3, la répétabilité suivant les différents axes deviennent 0.21 mm selon X, 0.05 mm selon Y et 0.15 mm selon Z. En combinant les trois composantes de l'erreur, nous arrivons à une répétabilité générale de 0.26 mm , qui est tout à fait correcte.

6.4 Bilan sur les performances du système complet

Les deux précédents tests ont mis en avant la répétabilité et la précision du système complet de vision et du robot. En comparant avec les données du robot seul, il semble effectivement que la vision conditionne les performances globales. Ci-dessous est présenté le tableau des différentes performances.

	Robot	Vision + Robot
Répétabilité (RP)	0.01 mm	0.26 mm
Précision (AP)	0.02 mm	3.95 mm

Bien que la répétabilité soit tout à fait acceptable, de l'ordre du quart de millimètre, la précision pose un sérieux problème. A moins d'identifier cet offset lors de l'installation des caméras et de l'enlever lors des acquisitions, seule une installation comme présentée en 3.3.1 permettrait d'uniformiser et de minimiser cette écart.

Lors des précédents essais, cet offset avait pu être supprimé en décalant le référentiel du robot de manière opportune.

Chapitre 7 Développements futurs et conclusions

7.1 Développements futurs

Au vu des résultats obtenus plus que satisfaisants, illustrés à la fois par les essais et par les analyses métrologiques, il nous est permis de définir l'utilisation de la vision stéréoscopique comme un moyen robuste pour la programmation de cellules robotisées.

L'application de cette technique au monde industriel doit cependant être adaptée et peut encore être améliorée. Pour ce qui est de l'amélioration, l'utilisation de plus de trois caméras ou la configuration de ces dernières peuvent être des solutions à nos problèmes, permettant d'exploiter au maximum les informations d'une scène et de rendre le système encore plus robuste. De plus, l'utilisation de plus de quatre blocs permettrait une plus grande liberté à la personne effectuant les mouvements à enregistrer, ainsi qu'une meilleure précision et une plus grande robustesse dans le traçage de la dynamique du pinceau.

Enfin, pour ce qui est de l'adaptation au monde industriel, des systèmes de préhension peuvent être installés sur le robot, afin d'augmenter sa liberté, son autonomie et la richesse des dessins reproduits. Ci-dessous est présenté un exemple de pince à prise cubique, pouvant être utilisée pour le changement de pinceau, préalablement préparé. Idéalement, ce serait tout l'îlot utilisé par le peintre, présentant les peintures, les pinceaux, le matériel de nettoyage, et le carreau à peindre, qui pourraient être adaptés pour une utilisation par le robot.

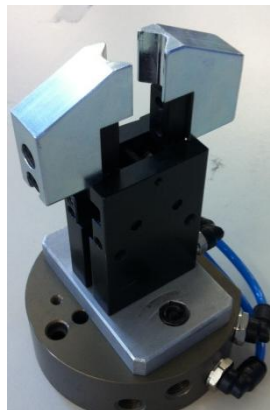


Figure 7.1-a : Exemple de pince pneumatique à prise cubique

7.2 Conclusions générales

L'utilisation de la vision stéréoscopique pour l'enregistrement de mouvements à faire reproduire à un robot anthropomorphe est une technique innovante qui n'est encore que très peu utilisée dans le monde de l'industrie et de la recherche. Son implémentation dans un projet de peinture présente un aspect interactif et visuel et est un premier pas pour son développement futur. Au vu des résultats obtenus plus que satisfaisants, il a été possible de valider cette technique, et ainsi espérer son élargissement dans d'autres domaines. On peut imaginer de nombreuses applications pour ce dernier, comme la médecine ou l'industrie.

Durant ce projet de fin d'étude, j'ai pu acquérir des compétences de recherche et de développement de systèmes à un niveau satisfaisant. J'ai pu y découvrir la mise en place d'un projet complet : de la recherche de l'état de l'art, en passant par le développement de solutions concrètes au problème, pour arriver au final à un produit répondant à des demandes précises. De plus, j'ai pu connaître une entreprise de grande envergure et importantes dans le monde de la robotique, et de connaître la richesse des applications que l'on peut trouver à la robotique.

Grâce à ce projet de fin d'étude, j'ai eu l'occasion de mûrir personnellement et professionnellement, mais surtout de m'éveiller et d'attiser ma curiosité sur des sujets tels que la vision artificielle, l'intelligence artificielle et la robotique en général. Pour l'avenir, j'espère pouvoir continuer dans la voie de l'innovation et de la conception de solutions, et espérer un jour pouvoir contribuer moi aussi au développement de la recherche scientifique.

Pour conclure je tiens encore une fois à remercier tous les acteurs de ce projet par une vidéo disponible en annexe, présentant des remerciements de la part du robot en personne.



Figure 7.2-a : Remerciement de la part du robot lui-même

Bibliographie

- [1] A. Zisserman et R. I. Hartley, «Multiple View Geometry in Computer Vision. Second edition,» *Cambridge University Press*, 2003.
- [2] A. Fusiello, «Visione Computazionale,» *Università degli Studi di Verona, Italy*, 2008.
- [3] D. C. Brown, «Decentering Distortion of Lenses,» *Photometric Engineering*, vol. 32, pp. 444-462, 2004.
- [4] «Toolbox de calibration de Matlab».
- [5] G. Mainetti, «Calibrazione di telecamere per sistemi di visione stereoscopica: Confronto tra algoritmi genetici e tecniche tradizionali,» *Politecnico di Milano*, 2011.
- [6] «www.mvtec.com/halcon/,» [En ligne].
- [7] L. W. Kheng, «Rigid Registration,» *National University of Singapore*.
- [8] P. Rocco, «Control of industrial robot - Motion Planning,» *Politecnico di Milano*, 2011.
- [9] A. Les Piegl et W. Tiller, *The Nurbs Book*, 1997.