

Effet du paramétrage

Le mécanisme peut être paramétré avec 13 variables, ou l'intuition humaine peut être utilisée initialement pour réduire la dimension à optimiser. Les résultats suivants expliquent à quel point l'intuition humaine est proche du résultat optimisé pour certaines stratégies de récompense. Lorsque nous optimisons le mécanisme en récompensant la performance du mécanisme inversement proportionnelle à la distance du point (α, β) de $(0, 0)$, le paramétrage général conduit à un résultat optimisé avec des paramètres très proches des 4 paramétrages choisis par l'homme.

Les résultats optimisés avec les 4 paramètres sont :

$$[0.357, 0, 0, 0.438, 0, 0.105, 0.357, 1.57, 0, 0.438, 1.57, 0.1, 3.72]$$

avec une plage d'actionneurs de $[3.411, 4.154]$. Les résultats optimisés avec les 13 paramètres sont :

$$[0.458, -0.443, -0.084, 0.742, -0.409, 0.018, 0.631, 1.588, -0.03, 0.724, 1.157, 0.012, 3.6]$$

avec une plage d'actionneurs de $[2.986, 4.358]$

5.3.10.5 Temps de calcul

Pour comprendre l'importance de réduire la dimension d'optimisation ainsi que l'effet des différentes contraintes sur le temps de calcul, le tableau 5.2 présente le nombre d'itérations et le temps pour chaque itération en optimisant avec 4 paramètres et 13 paramètres. De plus, comme la discrétisation et les critères de recherche sont plus stricts dans la recherche fine, ces recherches plus fines sont très coûteuses en calcul. Ceci corrobore notre proposition d'une recherche grossière nous permettant de choisir quelques simplexes valides afin d'initier une recherche beaucoup plus approfondie d'un meilleur point optimal.

Ces résultats ont été calculés sur un ordinateur portable possédant un processeur i5-6200 avec 8 Gb de mémoire sous Matlab 2020.

Paramètres	Type de recherche	Itérations (min-max)	Temps moyen (sec/iteration)
4-paramètres	grossière	12 - 42	64
4-paramètres	fine	25-98	84
13-paramètres	grossière	34-138	286
13-paramètres	fine	58-152	410

TABLE 5.2 – Temps de calcul, suivant le nombre de paramètres et le type de recherche.

5.3.10.6 Résultats de l'optimisation de la variation 2UPS-U

Dans la variation de l'architecture 2UPS-U, le cardan de la troisième jambe définit les deux axes de rotation ainsi que le centre de rotation, qui peut être déporté par le parallélogramme, comme le montre la Figure 5.38. Comme vu précédemment, ce mécanisme peut être paramétré selon 13 paramètres à optimiser, résumées dans le tableau 5.3.

Nous présentons ici les résultats obtenus en optimisant le nombre de conditionnement global et en récompensant un point valide dans l'espace de travail comme 1 et 0. Le temps nécessaire pour évaluer une instance, c'est-à-dire un ensemble donné de paramètres, a été enregistré en même temps que le temps moyen pour un seul départ, c'est-à-dire l'opération complète jusqu'à ce que l'algorithme s'arrête pour renvoyer les paramètres optimisés localement. Il a ensuite été analysé pour noter l'impact des différents choix d'objectifs sur le temps total d'optimisation. Les résultats sont présentés dans le tableau 5.4 et le temps de calcul est enregistré sur le même système et ne doit être utilisé qu'à des fins de comparaison. La Figure 5.44 présente le tracé schématique pour le mécanisme optimisé pour un indice de performance global maximal, accompagné de la heatmap pour l'évaluation de cet indice avec les paramètres optimisés. De la même manière, la Figure 5.45 illustre le schéma et la heatmap de la performance reliée au facteur d'amplification de vitesse pour les paramètres optimisés correspondants. Il est intéressant de constater à partir des schémas présents sur les deux figures que les paramètres optimisés tendent vers une architecture telle que les jambes actionnées soient écartés de $\frac{\pi}{2}$ radians, et alignés le long des axes du cardan de la troisième jambe. Cette observation suggère également que nous pouvons utiliser l'intuition et l'expérience humaines pour réduire la dimension de l'espace d'optimisation.

Paramètres	Valeurs	Paramètres	Valeurs
Dimension d'optimisation	13	Amplitude de a_i	[0.25, 1.5]
Amplitude de b_i	[0.25, 2]	Amplitude de ϕ_i and ψ_i	[-1.745, 1.745]
Amplitude de h_i	[-0.1, 0.1]	Amplitude de t	[1, 4]
Nombre de départs	200	Nombre d'itérations	10 and 20
Fonctions objectif	Espace de travail, Indice de performance global, Facteur amplification de vitesse	Plage d'amplification de vitesse	[0.3, 3]
Amplitude de b_i	[0.25, 2]	Amplitude de ϕ_i et ψ_i	[-1.745, 1.745]
Espace de travail (en roulis et en tangage)	cercle de rayon 1	rapport de course	1.5
Limites sur les rotules	$\pm\pi/6$ radians	Contraintes de collision	pris en considération

TABLE 5.3 – Les paramètres définis pour l'optimisation de la variation 2UPS-U du mécanisme.

Paramètres	Indice de performance global	Facteur d'amplification de vitesse
Temps pour 1 évaluation grossière	14 seconds	18.3 seconds
Temps pour un démarrage grossier	291 seconds	347.5 seconds
Temps pour une évaluation fine	50.5 seconds	51 seconds
Temps pour un démarrage fin	1072 seconds	1077 seconds
Meilleur point [$a_1, \phi_1, h_1, b_1, \psi_1, h_2, a_2, \phi_2, h_3, b_2, \psi_2, h_4, t$] (voir Figure 5.38)	[1.13, -1.02, -0.06, 1.47, -1.01, -0.05, 0.72, 0.44, -0.02, 1.52, 0.54, 0.02, 3.04]	[0.68, -0.25, 0.08, 1.03, 0.1, 0.04, 0.25, -1, 0.01, 1.1, -1.45, 0.17, 2.4]
Meilleure gamme d'actionneurs	[2.54, 3.8]	[2, 3]

TABLE 5.4 – Résultats de l'optimisation pour la variation 2UPS-U du mécanisme.

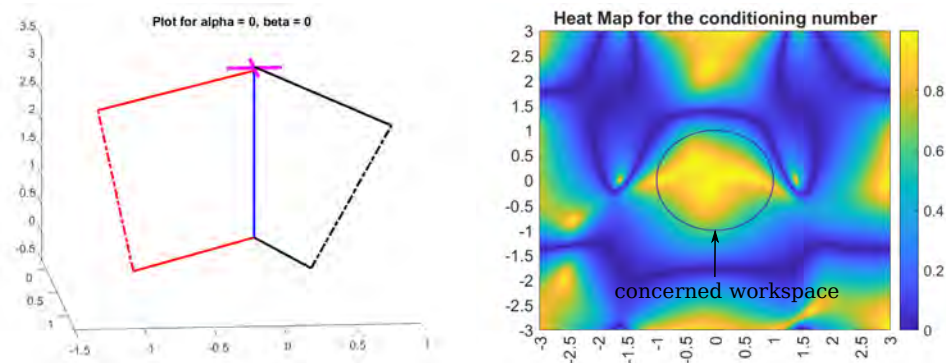


FIGURE 5.44 – Tracé schématique pour le mécanisme optimisé pour l'indice de performance global, et heatmap de l'évaluation.

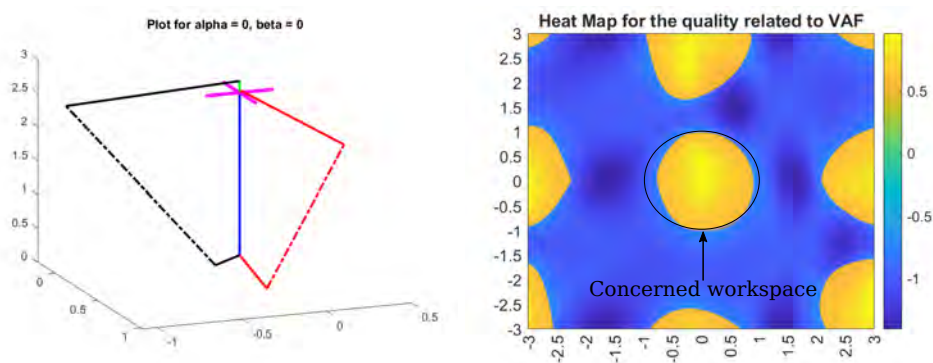


FIGURE 5.45 – Tracé schématique pour le mécanisme optimisé pour le facteur d'amplification de vitesse, et heatmap de l'évaluation.

5.4 Stratégie de commande par la vision

Le contrôle du robot par suivi d'image est un élément important, car il permettrait au porte endoscope de suivre un outil du chirurgien, et libérerait celui-ci de la contrainte de la commande par joystick.

Dès l'analyse fonctionnelle 4.3, nous avons identifié cet axe de travail, pour développer ce mode de contrôle pour le robot. La possibilité de libérer les mains du chirurgien grâce à ce type de suivi est également un point d'attente fort exprimé par les chirurgiens lors de l'étude de marché 4.6.

Ce travail a été réalisé avec Aurèle DALIBOT et André D'ELBREIL, étudiants à Centrale Nantes, de septembre à décembre 2019, puis Thomas GREBAUT, étudiant à l'ESEO, de septembre à novembre 2020.

5.4.1 Découpage des fonctionnalités du programme

Dans un premier temps, nous avons étudié les étapes à suivre pour permettre le suivi de l'outil. Nous avons alors défini un cycle de fonctionnement basique, représenté sur la Figure 5.46 :

- calibrage du suivi de l'outil au démarrage,
- traitement de la vidéo : maskage,
- tracking de l'outil,
- recentrage de la caméra,
- détection de la perte du tracking,
- arrêt du tracking.

5.4.2 Tracking de l'instrument

Pour pouvoir effectuer au mieux le tracking de l'objet, nous avons choisi d'effectuer un traitement de la source vidéo. Par exemple, si l'outil choisi est la canule d'aspiration métallique, il semble judicieux d'utiliser un masquage pour faire ressortir la couleur grise de l'instrument du fond de la vidéo, majoritairement rouge (tissus, sang, ...). Nous avons choisi la fonction `mask` de la bibliothèque OpenCV, qui va isoler une teinte de couleur en se basant sur un modèle TSV (Teinte Saturation Valeur, HSV en anglais) des couleurs.

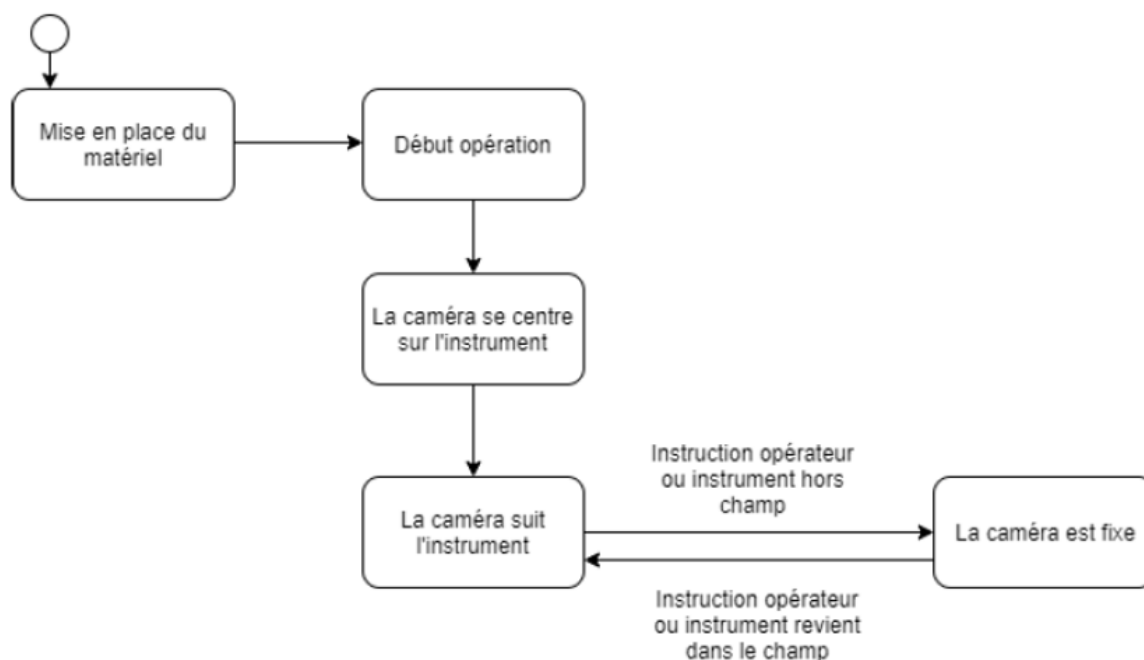


FIGURE 5.46 – Cycle de fonctionnement du programme de suivi d'image.

Il existe cependant un problème : l'outil est en métal et reflète les parois environnantes. Il prend donc parfois la même couleur que l'environnement. Cela, en addition d'une potentielle mauvaise qualité de vidéo, nous amène à un tracking peu robuste. Nous risquons dans certains cas de perdre la position de l'outil.

Nous avons donc eu l'idée d'ajouter une couleur sur l'outil qui ne soit pas présente dans le reste de l'image. Nous avons initialement choisi la teinte bleue d'une bande collante usuellement présente dans les blocs opératoires. En effet, le bleu n'est pas une couleur présente en anatomie, ce qui limite le risque de fausse reconnaissance lors de l'utilisation du logiciel. Le masquage sera donc concentré sur la couleur du strap du bout de l'outil et permettra de le faire ressortir aisément et de limiter les pertes de tracking (Figure 5.47).

Cependant, la réflexion de la lumière sur la bande collante bleue lui donnait, dans certaines circonstances, une couleur plutôt blanche, ce qui rendait l'algorithme inefficace.

Nous avons alors choisi d'utiliser un scotch bicolore pour augmenter la robustesse du suivi. Les couleurs choisies étaient le jaune et le vert, deux autres couleurs bien différentes de la couleur rosée souvent visualisée lors d'une intervention endoscopique. Cependant, un autre choix de couleurs est bien sûr possible, et le programme est adapté pour convenir à différentes combinaisons de couleurs. La démonstration porte sur l'intérêt de combiner

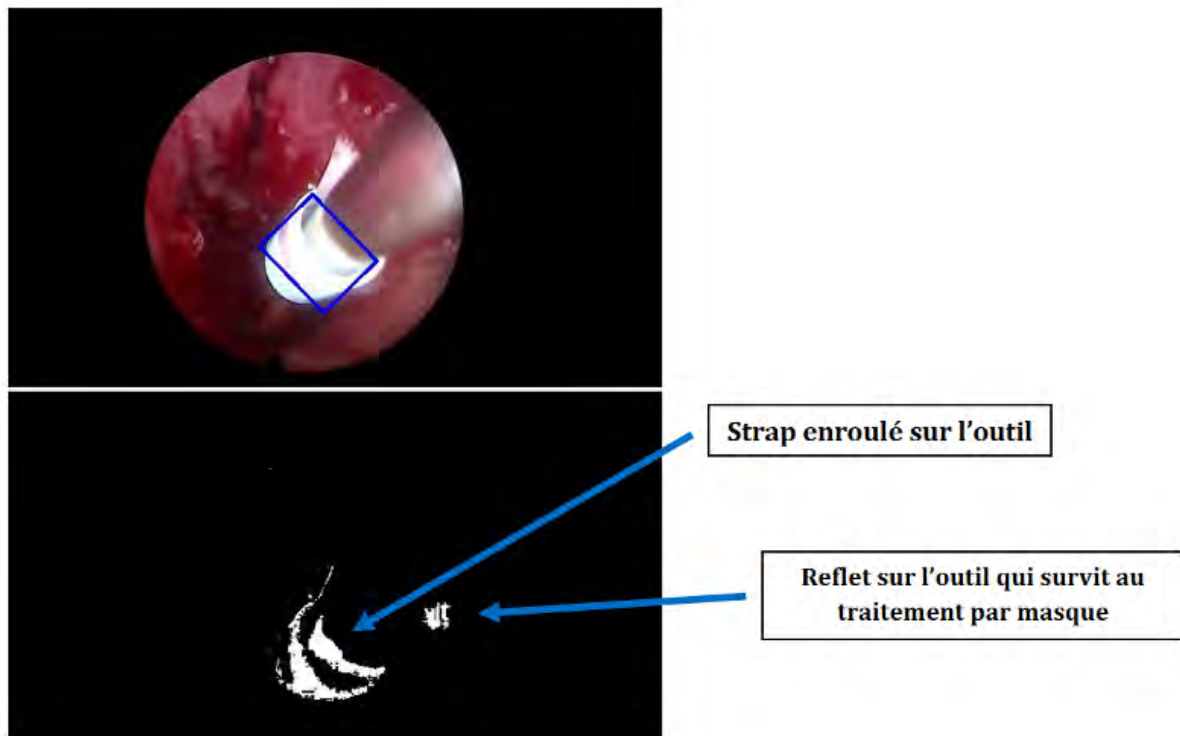


FIGURE 5.47 – Image per opératoire d’une canule d’aspiration avec strap, sous endoscopie, sans masquage (en haut) puis avec masquage (en bas).

deux couleurs pour augmenter la fiabilité du suivi.

Une fois les couleurs choisies, elles sont converties en teintes HSV. La conversion des couleurs en cette teinte permet de séparer une couleur en trois parties : la teinte, la saturation et la luminosité (en comparaison, le modèle de teinte RGB sépare une couleur en 3 composantes, le rouge, le vert et le bleu). Contrairement aux modèle RGB où chaque composante est échelonnée sur 256 bits (255-0-0 pour le rouge par exemple, ou encore 255-255-255 pour le blanc), la teinte HSV se décompose comme suit : 0-179 pour la teinte, 0-255 pour la saturation et 0-255 pour la luminosité. En effet, la saturation et la luminosité sont traduites par un pourcentage tandis que la teinte est trouvée à partir d’un disque chromatique. Après conversion en HSV, un intervalle de confiance est déterminé, avec les valeurs HSV minimales (vert/jaune clair très lumineux et peu saturé) et maximales (vert/jaune foncé très sombre et saturé), pour que l’algorithme identifie correctement les teintes quelque soient les variations de luminosité. Pour rendre l’interface simple pour l’utilisateur, nous avons étudié un système de détection de la couleur à une localisation particulière dans l’image. Pour cela, nous avons créé une fonction “appareil

photo”, permettant de prendre une photo de l'outil en ayant une région précise de l'image à analyser afin de récupérer la couleur majoritaire sur cette image. Lors du lancement du programme, l'utilisateur est invité à placer la première couleur de son scotch dans le carré vert. Le programme prend alors une photo, isole la région du carré vert et fait la moyenne des pixels dans cette zone. Il compare ensuite cette valeur à une liste d'intervalles définis au préalable pour savoir quelle est la première couleur du scotch. Ensuite, le programme refait ces mêmes étapes pour la seconde couleur, puis commence le suivi avec comme paramètres les deux couleurs détectées. Si la détection des deux couleurs venait à échouer, l'utilisateur redémarre le programme en pressant une touche. Sinon, aucune intervention de l'opérateur n'est nécessaire.

Une fois ces teintes établies l'image est convertie en couleurs HSV, puis un mask de l'image est créé, c'est-à-dire une image binaire où seuls les pixels porteurs de la couleur de l'intervalle sont blancs, et tout le reste est noir (Figure 5.48).

Afin d'éliminer tout bruit qui pourrait venir perturber le suivi, nous avons souhaité appliquer un filtre au mask. Une première possibilité était le filtre de Kalman, utilisant une série de mesures bruitées afin d'estimer les mesures à venir. Mais pour extraire une estimation des mesures bruitées, il faut établir un modèle de ces mesures ; dans notre cas, la modélisation de la position de l'objet n'était pas concevable, car les mouvements du chirurgien lors d'une opération sont à vitesse trop aléatoire. Une seconde possibilité était l'utilisation d'un filtre médian. Ce filtre parcourt chaque élément de l'image et remplace chaque pixel par le médian de ses pixels voisins. Ce procédé permet le lissage de l'image et l'élimination du bruit (Figure 5.49).

Après ces premières étapes de détection de couleurs et de mask, les algorithmes de suivi d'image peuvent alors être utilisés.

5.4.3 Suivi de l'outil en cours de mouvement

Nous avons évoqué deux possibilités d'algorithmes de suivi d'image : le Mean Shift et le CamShift.

La fonction Mean Shift, fournie par la librairie OpenCV, permet de suivre le déplacement de l'objet uniquement dans un rectangle de taille fixe. Il peut trouver un objet dans une image à partir d'une couleur donnée, en prenant un ensemble de points dans une certaine zone (par exemple un cercle) et en déplaçant le centre de ce cercle vers la région du cercle où la densité de pixels de la couleur donnée est la plus grande. Ensuite, on réitère l'expérience jusqu'à trouver la zone où la densité de pixels est maximale. Par exemple,

5.4. Stratégie de commande par la vision

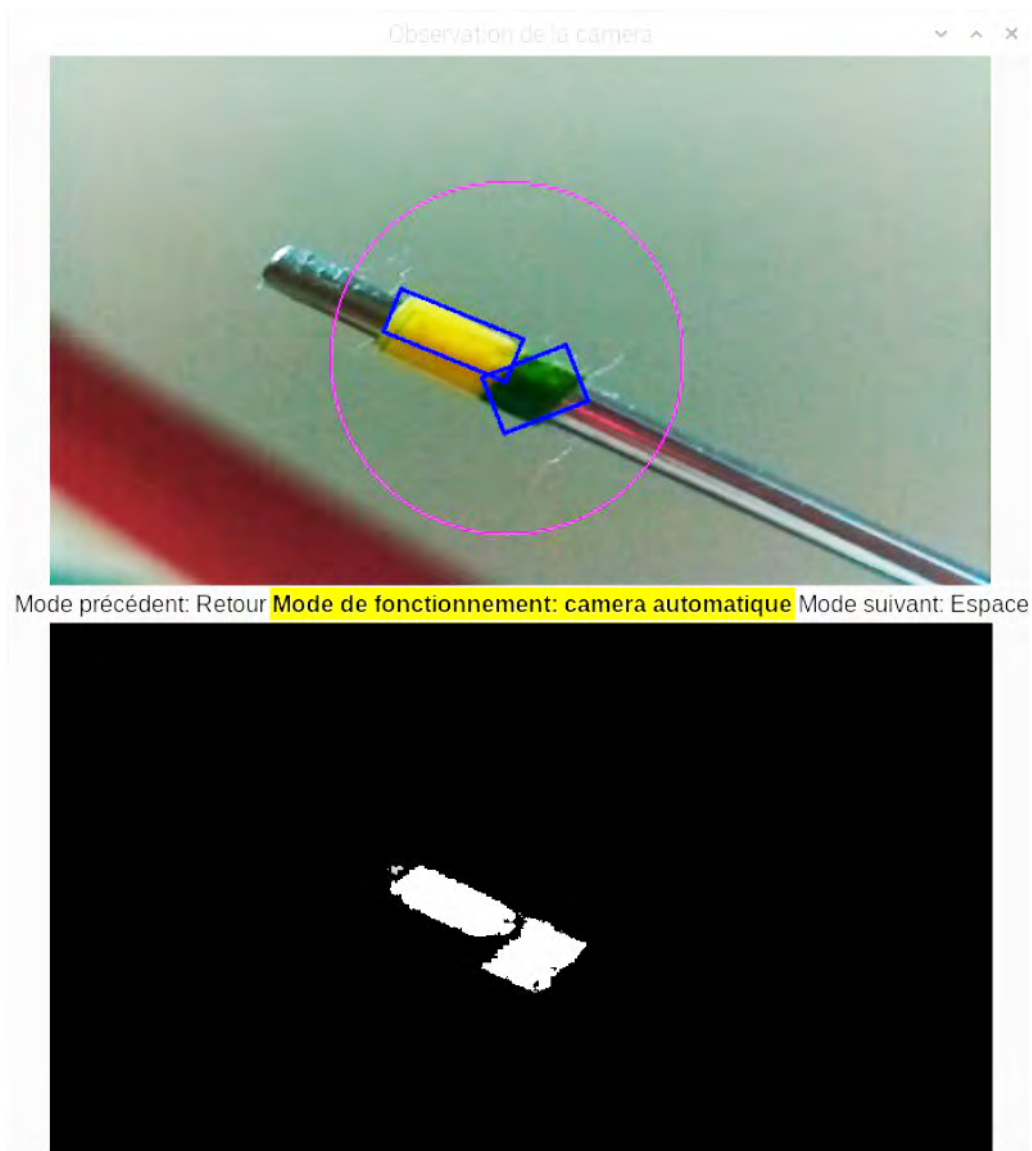


FIGURE 5.48 – Interface de suivi avec les deux masks pour chaque couleur.



FIGURE 5.49 – Comparaison du mask sans (à gauche) et avec le filtre médian (à droite).

dans la Figure 5.50, on prend un cercle $C1$ de centre $C1_o$ dans lequel on recherche le maximum de pixels rouges. On déplace le centre du cercle $C1$ vers l’isobarycentre des pixels rouges de $C1$. Le nouveau centre de $C1$ est maintenant $C1_r$. On répète cette opération jusqu’à trouver l’endroit où la densité de pixels rouges dans le cercle est maximale. Cet évènement est atteint avec le cercle $C2$.

Cependant, puisqu’il suit un objet dans un rectangle fixe uniquement, ce tracking est très peu précis lorsque l’objet se rapproche ou s’éloigne de l’objectif (Figure 5.51).

L’algorithme du CamShift (Continuously Adaptive Mean Shift), encapsule celui du Mean-Shift dans une boucle faisant varier la taille de la fenêtre jusqu’à convergence. À chaque itération, le Mean Shift est appliqué avec une fenêtre de taille donnée. Ainsi l’algorithme du CamShift permet de faire varier la taille du rectangle de tracking au cours du temps. Si l’objet se rapproche de l’objectif, le rectangle va s’agrandir et s’il s’éloigne sa taille va diminuer (Figure 5.51). Le suivi est alors beaucoup plus fin et on peut savoir précisément où est le bout de l’outil.

Des tests de suivis ont été réalisés sur des vidéos d’explorations endoscopiques de fosses nasales (<https://youtu.be/XIsDSWH9fsA>) et de conduit auditif externe (<https://youtu.be/Tfrb6Z0Y73k>). Le suivi était fonctionnel lors de ces tests, y compris lorsqu’une des deux couleurs disparaissait. Un essai a également été réalisé en contexte chirurgical avec présence de sang (<https://youtu.be/s9XzBQ0NGdI>); cependant, il faut noter qu’il ne s’agissait que d’une phase d’observation lors d’une chirurgie, et que des essais plus approfondis en conditions réelles demeurent indispensables.

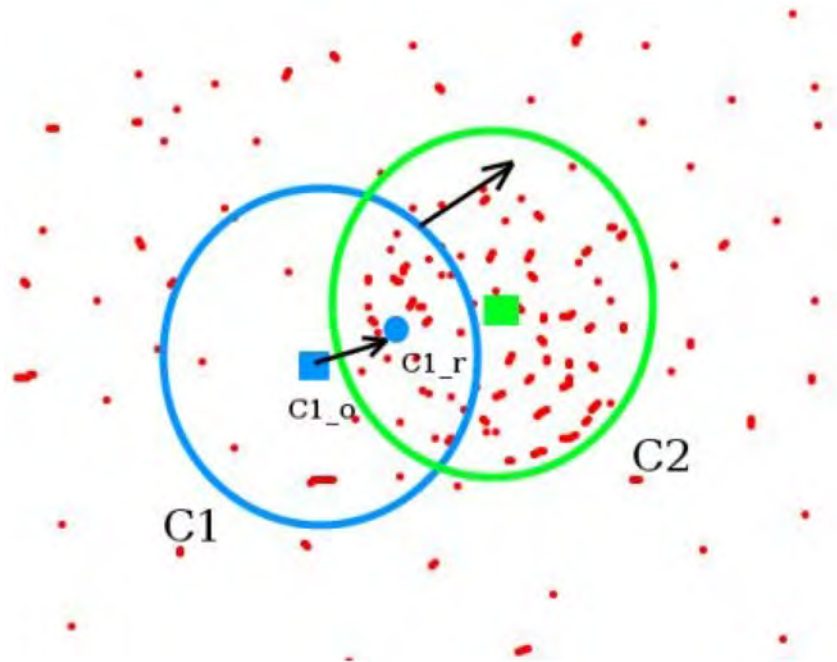


FIGURE 5.50 – Illustration du fonctionnement de l'algorithme du Mean Shift. *Source : OpenCV.*

5.4.4 Recentrage de la caméra sur l'outil

Nous nous sommes posé la question sur le choix d'un mouvement continu de l'endoscope, ou d'un recadrage lors que l'outil sort d'une zone prédéfinie. Il nous est apparu non nécessaire, voire gênant, que le robot soit toujours en mouvement, en miroir des mouvements de l'outil. En effet, en observant la gestuelle du chirurgien lors d'une intervention sous endoscopie, nous avons constaté que l'endoscope bouge moins que l'outil, et se fixe sur certaines positions, souvent pour avoir un champ de vision large sur la zone opérée.

Nous avons donc choisi de créer un cadre factice, un peu moins grand que la taille de l'image, qui représente la zone de travail du chirurgien. Tant que le barycentre de l'outil est dans ce cadre, il n'est pas nécessaire de déplacer la caméra. Cependant si le barycentre de l'outil dépasse un bord du cadre, il est nécessaire de déplacer la caméra (Figure 5.52). Suivant les coordonnées du barycentre que nous récupérons il est aisé de choisir dans quelle direction déplacer la caméra car nous savons exactement quelle frontière il a franchi.

Lorsque l'outil sort de la zone prédéfinie, la caméra se déplace de sorte à ce que l'outil reste sur la frontière. Par exemple si jamais l'outil va dépasser la frontière droite de

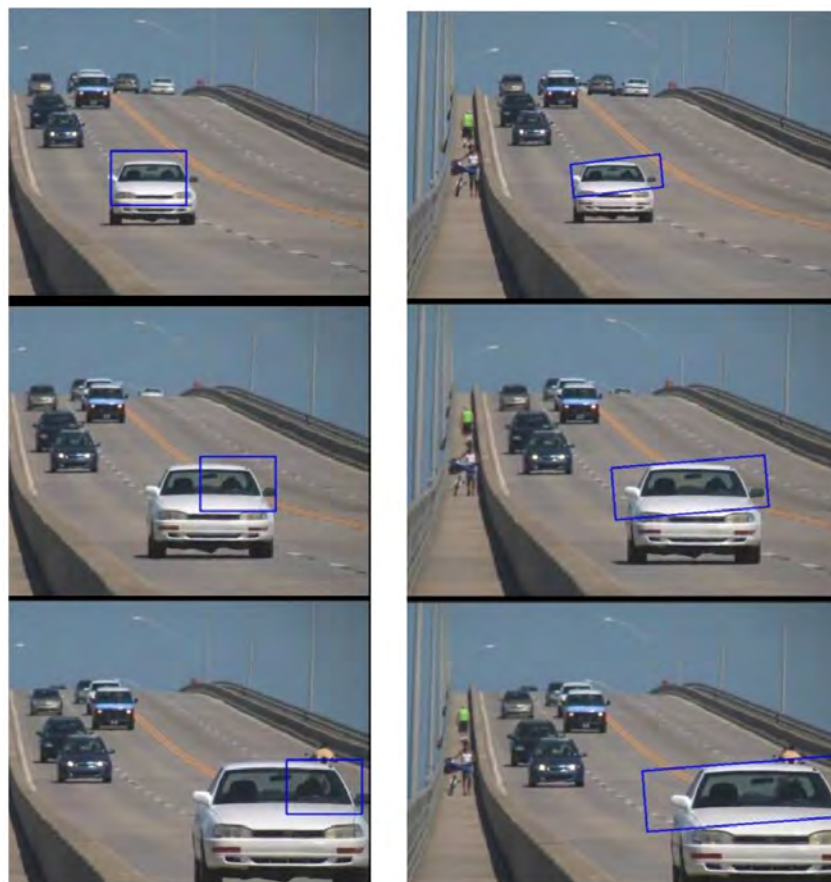


FIGURE 5.51 – Tracking d'une voiture avec la fonction Mean Shift (à gauche) et CamShift (à droite).

l'image, on déplace la caméra de sorte à ce qu'il ne la dépasse pas mais qu'il reste juste dessus. Ainsi l'outil du chirurgien lui permet de cadrer l'image comme il veut par des petits à-coups sur les frontières de la zone de travail. Cela évite, dans un milieu étroit, que la caméra touche une paroi si elle devait se recentrer systématiquement.

Nous avons pu définir le suivi du robot en fonction de sa position angulaire. Cependant, il reste à définir le mécanisme de suivi lors d'un déplacement en profondeur, selon l'axe z de l'effecteur. Une solution serait de permettre à l'opérateur de se mouvoir selon cet axe à l'aide d'une pédale ou d'un joystick. Cependant, cela limiterait alors l'intérêt du suivi d'image, visant à libérer complètement les mains du chirurgien et à l'affranchir de la commande du robot pour plus de simplicité. Une autre alternative, qui paraît plus adaptée, serait de définir une mesure entre l'aspiration, ou plus précisément la bande de scotch bicolore qui y est fixée, et l'endoscope. Lors de la validation initiale des couleurs,

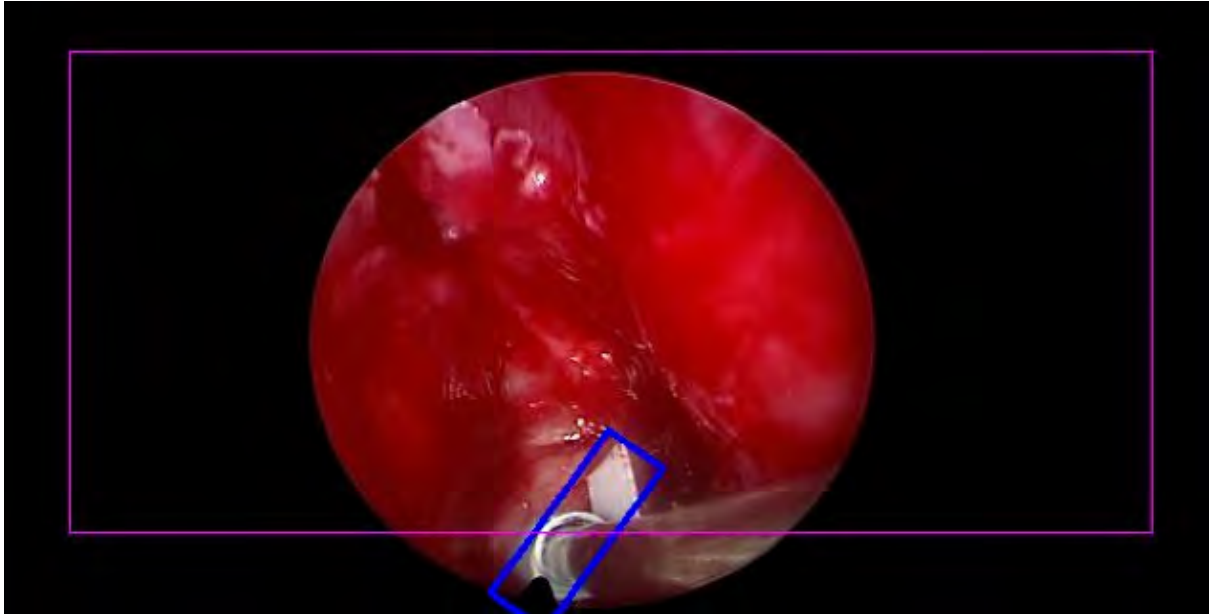


FIGURE 5.52 – Chirurgie endoscopique, avec cadre factice (en violet) et cadre de suivi CamShift (en bleu).

cette distance est alors fixée, et le robot se déplacera alors par la suite pour maintenir cette distance fixe. Pour modifier cette distance, un nouvel enregistrement de distance serait alors réalisé. Pour valider cette fonction, des essais avec un prototype doivent être réalisés.

5.4.5 Commande du robot

Les instructions à suivre pour déplacer la caméra doivent être envoyées au robot, en fonction de la position par rapport au cadre prédéfini. Par exemple, si l’outil sort du cadre en haut, il faudra envoyer l’instruction “Déplacement vers le haut”. Désormais, il faut convertir ces instructions en commande pour le robot.

5.4.5.1 Position de l’objet par rapport à la caméra

Une première étape est de connaître la position de l’objet par rapport à la caméra. L’étude des angles d’Euler peut permettre de réaliser cette étape. Pour obtenir ces angles, une matrice de rotation est obtenue en calibrant la caméra, c’est-à-dire en obtenant les paramètres internes et externes de cette dernière. Les paramètres internes sont la distance focale, le centre optique et les coefficients de distorsions. La distance focale (f_x, f_y) et

le centre optique (cx, cy) forment une matrice appelée “camera matrix” et les coefficients de distorsions forment un vecteur.

Une fonction sous OpenCV permet de renvoyer ces paramètres et calibrer la caméra : en utilisant un échiquier en papier dont on connaît les dimensions, on peut établir une liste de points du système de coordonnées 3D appelée Object Points. Il faut ensuite trouver les quatre coins de l'échiquier dans le champ de la caméra, et ces points du système de coordonnées 2D en pixels sont les Image Points.

En photographiant l'échiquier sous divers angles avec la caméra utilisée, et à partir des deux listes de points, un calibrage était alors possible avec l'obtention de ses paramètres internes. Cependant, le calcul est long et coûteux pour le programme, et sa réalisation à chaque lancement du programme serait inutile.

Pour résoudre le problème de perspective des points, l'utilisation d'une fonction spécifique de la librairie OpenCV permettait de récupérer un vecteur de rotation et un vecteur de translation. En convertissant ce vecteur en une matrice de rotation, grâce à la formule de rotation de Rodrigues, il était possible de déduire les angles d'Euler.

5.4.5.2 Programmation des déplacements du robot

Pour programmer les déplacements du robot, nous avons utilisé deux moteurs pas à pas, une carte Makeblock MeOrion pour lier les moteurs à un processeur Raspberry Pi 3, et un prototype en impression 3D. Une fois les moteurs prêts à l'utilisation, les fonctions régissant les mouvements du bras ont été écrites dans le programme. Avec les angles d'Euler, nous utilisons les deux premiers angles pour commander les moteurs pas à pas, en traduisant les angles en nombres de pas, en fonction du type de rotation et de la valeur en degrés d'un pas du moteur. La synchronisation des moteurs a été réalisée, afin que l'envoi des commandes aux moteurs entraîne un mouvement de même durée de chaque moteur (pour que le mouvement commence et se termine en même temps).

L'assemblage avec le robot sera une étape ultérieure pour vérifier le bon fonctionnement de l'algorithme de suivi avec le robot.

5.4.5.3 Conclusion du suivi d'image

Un algorithme de suivi d'image basé sur le CamShift a été présenté dans cette partie. Il permet un suivi d'un instrument, comme la micro-aspiration, grâce au tracking de deux scotchs de couleurs différentes placées à son extrémité.

5.4. Stratégie de commande par la vision

Selon la position de l'outil suivi, le robot est soit fixe (outil en position centrale), soit mobile dans la direction de l'outil (lorsqu'il sort du cadre central). Cela permet d'éviter un mouvement continu du robot qui pourrait être gênant pour l'opérateur. Une simplification de la reconnaissance des couleurs a été menée pour ne pas que le chirurgien soit obligé de passer par plusieurs écrans de validation. En cas de perte de reconnaissance de l'outil ou de sortie de l'écran, le robot passe en mode transparent.

5.5 Conclusion

Dans ce chapitre, nous avons analysé différents choix d'architecture, pour tenter de définir laquelle serait la plus adaptée au besoin chirurgical et à l'espace de travail considéré.

L'étude d'un mécanisme à centre de rotation déporté, couplant un mécanisme sphérique à 2 DDL avec un double parallélogramme, est présenté, ainsi que ses variations.

Afin de répondre à la question de l'optimisation de cette architecture pour une utilisation dans la chirurgie de l'oreille et des sinus, un nouvel algorithme basé sur l'algorithme de Nelder Mead est présenté. Cet algorithme permet d'optimiser un mécanisme parallèle, en réduisant les temps de calcul grâce à une technique d'analyse grossière puis fine.

Enfin, pour mieux définir le contrôle de ce robot et permettre au chirurgien de garder ses deux mains libres, nous avons étudié un algorithme de suivi d'image basé sur la technique du CamShift et la reconnaissance de deux couleurs fixées sur l'outil d'aspiration.