Commande visuelle et orale d'un robot mobile

Bruno Buttice, Daniel Gepner, Noëlle Carbonell, Francis Lepage

Commande visuelle et orale d'un robot mobile.

Bruno Buttice, Daniel Gepner, Noëlle Carbonell, Francis Lepage
Université Henri Poincaré, Nancy 1,
Centre de Recherche en Automatique de Nancy (CRAN), CNRS UMR 7039
Laboratoire Lorrain de Recherche en Informatique et ses Applications(LORIA), CNRS UMR 7503
Bruno.Buttice@cran.uhp-nancy.fr, Daniel.Gepner@loria.fr
Noelle.Carbonell@loria.fr, Francis.Lepage@cran.uhp-nancy.fr

Mots-clés : *robot, téléopération, Interface Homme Machine, oculométrie.*

Résumé

Le but de ce papier est de présenter une interface de commande multimodale d'un robot mobile distant, équipé d'une caméra orientable en direction et azimut. Les modalités utilisées sont le regard associé à la parole ou au geste. Le champ d'applications que permet d'envisager le démonstrateur réalisé couvre des domaines variés, télé-surveillance, handicap moteur, exploration de sites hostiles ou inaccessibles. Cet article décrit l'architecture matérielle, l'architecture logicielle et les différents protocoles de communication mis en oeuvre sur une plateforme démonstrative réalisée dans le cadre d'une collaboration régionale entre des équipes du CRAN et du LORIA.

1 Introduction

Les oculomètres (eyes trackers) offrent de nouvelles possibilités en interaction Homme-Machine. Avec un tel dispositif, il est envisageable de commander un système à partir d'une consigne donnée par le regard d'un opérateur. Ceci est particulièrement intéressant lorsque l'opérateur ne peut utiliser ses mains, soit parce qu'elles sont occupées à d'autre tâches, soit parce qu'il est handicapé. Si la commande de ce système s'effectue à travers un réseau de communication, le découplage spatial ouvre de nouvelles perspectives d'applications, notamment aux systèmes mobiles. L'exploration visuelle à distance d'environnements naturels intervient dans de nombreuses activités de télé-robotique : télé-surveillance, diagnostic et maintenance des centrales nucléaires, exploration spatiale, etc. Dans ces situations, l'opérateur contrôle un robot mobile doté d'une ou plusieurs caméras vidéo, en vue d'explorer un espace distant, familier ou non. Or, malgré les remarquables progrès scientifiques et techniques réalisés au cours des dernières années dans ces domaines, l'écart fonctionnel entre les capacités du système robotique et les mécanismes de contrôle sensori-moteurs humains demeure important. Pour réduire cet écart et accroître la synergie entre l'opérateur et le système de télé-robotique, deux approches sont possibles, à notre connaissance. L'une tente de doter ce type de système d'un comportement d'anticipation visuo-motrice analogue au comportement humain de contrôle des déplacements ; voir, par exemple, l'approche " bionique " de la coopération homme-machine mise en oeuvre par les auteurs de [1]. L'autre vise à enrichir les modalités d'interaction entre l'opérateur et le système, en vue de donner à l'opérateur humain la possibilité d'exercer ses facultés d'anticipation visuo-motrices dans la conduite du robot et le contrôle de l'orientation de la (ou des) caméra(s). Nous avons choisi cette seconde approche.

La commande à distance d'un robot à l'aide d'un oculomètre est inédite à notre connaissance. A l'université de technologie de Brno en République Tchèque a été développée une application de commande à distance sur un robot pour effectuer des missions de sauvetage. Cette application met en oeuvre deux types de commandes, la commande joystick pour piloter le robot et l'utilisation d'un *Head-Tracker* pour piloter la caméra. Le *Head-Tracker* capte les mouvements de la tête et commande la caméra en fonction de l'orientation de ceux-ci tandis que l'opérateur récupère en 3D le flux vidéo sur l'écran de l'Interface Homme-Machine[2] [3]. Notre application met en oeuvre d'autres technologies pour commander le robot et la caméra : la commande orale (émulée par une commande manuelle actuellement) pour commander la direction du robot et la commande visuelle avec un *Eye-Tracker* pour commander la rotation de la caméra. Le but de ces deux commandes est de visualiser le mieux possible ce que souhaite voir l'opérateur comme dans la chirurgie laparoscopique [4] [5]. Cette application peut servir de support, en terme d'expérimentation pour l'étude de la commande robuste via un système à qualité de service non contrôlée.
La figure 1 représente la boucle cybernetique de notre application avec les deux commandes de notre

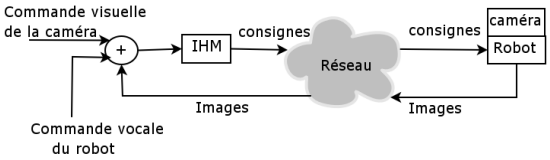

FIG. 1: Boucle automatique de notre système

système.

2 Application à vocation de recherche

Dans un système téléopéré via un réseau comme Internet on ne peut garantir précisément ni le temps de transmission de la commande ni celui du retour d'état. En effet Internet est un réseau à qualité de service non contrôlée. On ne peut pas connaître à l'avance le chemin que les paquets de commande ou d'informations vont emprunter, le matériel réseau traversé par les paquets, le trafic concurrent qui va charger notre réseau et donc ralentir notre système. Dans un système téléopéré via Internet on peut avoir de la perte d'information (due à la congestion d'un routeur) et un temps de transmission pour chaque paquet de commande différent. Les temps de transmission des paquets de commande ou d'information seront différents à chaque cycle. Le problème est qu'actuellement les modèles automatiques employés considèrent la plupart du temps un retard τ constant entre deux bornes τ_{min} et τ_{max} pour les problèmes cités auparavant. Ces modèles ne peuvent pas être utilisés dans le contexte d'un système contrôlé par un réseau à qualité de service non contrôlée. Cette application pourra servir de support à l'étude des systèmes à retard variable.

3 Description du robot mobile

Le robot mobile utilisé est un modèle ActivMédia Pioneer. Comme le montre la figure 2, le robot possède uniquement deux roues motrices et mesure 44 cm de longueur, 38 cm de largeur et 22 cm de hauteur. Le diamètre des roues motrices est de 16,5 cm. La commande embarquée est organisée en deux niveaux :

- Un niveau bas constitué d'une commande des moteurs du robot réalisée par le constructeur et d'une commande de rotation de la caméra.

- Un niveau haut qui envoie des consignes à l'une ou l'autre des commandes de niveau bas.

Un PC portable sous Linux (redhat 7.2) qui est la *station robot* est posé sur le robot figure 4, relié à celui-ci par une liaison câblée série. La communication entre

FIG. 2: Vue du robot Pioneer de l'application

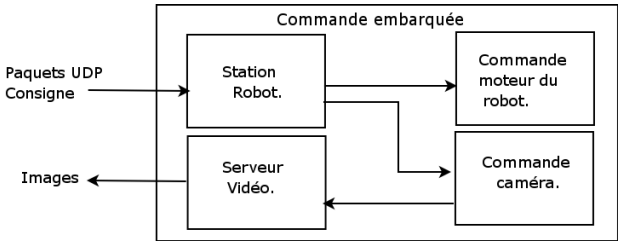

FIG. 3: Commande embarquée sur le robot

l'opérateur situé au LORIA et le serveur sur le robot situé au CRAN, est établie via le réseau Stannet et une partie de réseau Ethernet sans fil (802.11) qui permet au robot d'avoir une totale liberté de mouvement.

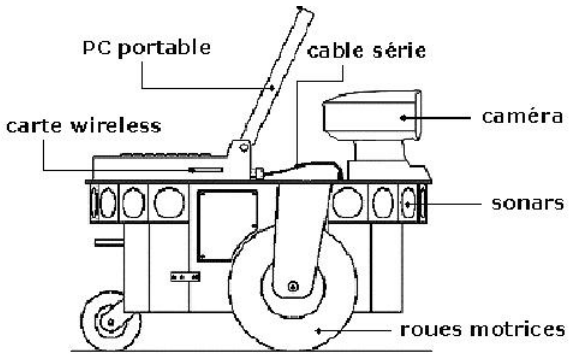

FIG. 4: Les équipements du robot

Une caméra de type *Pioneer PTZ Robotic Camera System* est couplée au robot et sert de support à une caméra de type *Logitech QuickCam Messenger*. En effet, pour des raisons techniques, la caméra couplée au robot sert uniquement à piloter la Quickcam qui elle est branchée à un serveur vidéo.
Le robot est équipé de sonars permettant de mesurer la distance le séparant d'obstacles en face de lui sur un angle de 180 degrés ; ceux-ci servent pour le dispositif anti-collision.

4 Présentation de l'oculomètre

La capture des mouvements oculaires est effectuée à 60 Hz par un oculomètre casque (ASL, modèle 501, voir figure 5) qui permet d'avoir une bonne précision tout en laissant à l'utilisateur sa liberté de mouvement. Un algorithme robuste de calcul en temps réel des fixations oculaires a été développé sous Linux.

FIG. 5: Oculomètre ASL modéle 501

5 Commandes du système

Pour éviter toute ambiguité entre la commande du robot et celle de la caméra, la commande du système met en oeuvre deux modalités différentes :

– la commande visuelle pour la caméra,

– la commande manuelle pour le robot.

5.1 Commande robot

Dans un futur proche le déplacement du robot sera commandé vocalement par l'opérateur. Cette modalité n'est pas encore intégrée dans le démonstrateur. Le robot est actuellement commandé par le clavier.

5.2 Commande caméra

Concernant le pilotage de la caméra vidéo par le regard, deux familles de solutions sont envisageables :
i) Créer des boutons que l'utilisateur sélectionne par le regard lorsqu'il souhaite changer son point de vue sur l'environnement distant. Cette approche, qui s'inspire des interfaces actuelles d'exploration des cartes disponibles sur Internet (cartes routières, plans de ville etc.), peut mettre à profit les résultats des nombreuses recherches menées sur la réalisation d'interfaces adaptées aux handicapés moteurs ; voir, par exemple, la synthèse des travaux sur les claviers virtuels présentée dans [6]. Mais, cette utilisation du regard est contraignante et risque d'être jugée peu naturelle par les utilisateurs. Son seul avantage par rapport à l'usage de la souris est de laisser une entière liberté manuelle à l'utilisateur.
ii) Asservir l'angle de prise de vue de la caméra au regard de l'utilisateur. Cette approche s'inscrit dans la continuité des recherches menées sur les affichages dont la résolution est asservie au regard (ou 'gaze-contingent displays') ; voir, pour une revue de l'état de l'art, le chapitre 14 de [7], et pour une étude ergonomique de ce type d'affichage interactif [8]. Au cas particulier, l'objectif serait de faire évoluer l'orientation de la caméra de façon à maintenir alignés en permanence (i.e., avec un retard minimum, idéalement imperceptible) l'axe optique de la caméra et la direction du regard de l'utilisateur.
Nous avons choisi la seconde approche sur la base de l'hypothèse suivante. Dans la mesure où elle respecte davantage le comportement visuo-moteur humain que la première, elle devrait être perçue par l'utilisateur comme plus naturelle et s'avérer d'utilisation plus confortable et plus efficace. Néanmoins, sa mise en œuvre présente des difficultés majeures. D'abord, on ne peut modifier l'angle de prise de vue à la suite de chaque fixation. Compte tenu de la rapidité des mouvements oculaires, le rythme des changements de point de vue serait beaucoup trop rapide pour assurer un confort visuel suffisant à l'utilisateur. Mais surtout, celui-ci ne se sentirait pas maître des mouvements de la caméra. En effet, la majorité des mouvements oculaires impliqués dans l'exploration d'une scène visuelle sont involontaires. Pour résoudre ce problème, nous filtrons les fixations oculaires en fonction de leur position dans l'affichage. Sont retenues pour agir sur l'orientation de la caméra uniquement celles qui se situent en périphérie de l'affichage de l'environnement distant. Ce filtrage spatial est assorti d'un filtrage temporel qui permet d'éliminer les variations trop rapides ou trop brutales de l'angle de prise de vue qui pourraient nuire au confort visuel de l'opérateur et à l'efficacité de l'exploration visuelle du site distant.
La caméra est pilotée grâce aux coordonnées oculaires de l'opérateur. Une coordonnée oculaire possède une abscisse et une ordonnée comprises entre 0 et 100 et suivant ces valeurs la caméra va pivoter vers la gauche, la droite, le haut ou le bas... Le principe de l'asservissement est d'offrir à l'opérateur une image vidéo centrée sur l'objet ou la zone de l'objet que celui-ci observe. Ainsi, lorsqu'il regarde vers la gauche de l'écran, l'interface va élaborer une commande de pivotement de la caméra vers la gauche, et lorsqu'il

regarde vers la droite de l'écran, l'interface va élaborer une commande de pivotement vers la droite, etc... Pour cela, l'écran est affecté d'un système de coordonnées 100*100.
Comme pour le robot, l'opérateur ferme la boucle de commande, mais il le fait dans ce cas directement avec son regard.

6 Architecture matérielle

6.1 Organisation générale

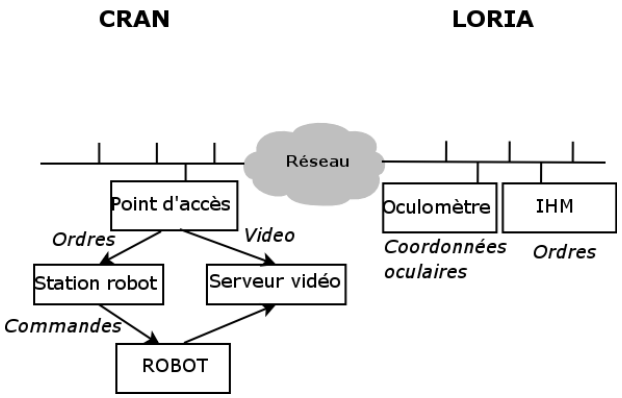

FIG. 6: Architecture réseau de l'application

L'application est distribuée sur deux réseaux différents, interconnectés, voir figure 6 :

– le réseau du CRAN,
– le réseau du LORIA.

L'opérateur de cette application se situe sur le site du LORIA et commande à distance le robot qui se situe au CRAN. Sur le réseau circulent uniquement les commandes de déplacement du robot et d'orientation de la caméra, ainsi que le flux d'images jpeg.
L'application est composée de 3 stations principales :
Station au LORIA : sur cette station est installée la Java Virtual Machine 1.5 qui permet d'exécuter le programme de pilotage du robot à distance et de recevoir le retour vidéo capturé par la caméra.
Station robot : c'est une station Linux (Redhat 7.2). En fait, cette station reçoit les commandes à destination du robot, en provenance de la station au LORIA, les interprète et actionne le robot et la caméra par le biais d'un programme développé en C++ qui utilise les fonctionnalités du kit de développement ARIA fourni par le constructeur du robot.
Station serveur Vidéo : il s'agit d'un ordinateur portable sous Windows XP. La seule utilité de cette station est d'héberger notre serveur vidéo *JMStudio*. Ce serveur vidéo émet à destination de l'opérateur situé au LORIA les images capturées par la caméra au format jpeg. Le flux vidéo est diffusé en multicast.

6.2 Rôle de chaque station

6.2.1 Serveur Vidéo

Sur la station serveur vidéo est installé l'outil *JMStudio* développé par Sun.
Cet outil répond aux exigences de l'application car il permet d'envoyer sur le réseau un flux d'images jpeg en multicast. L'avantage de cet outil c'est d'une part qu'il est aisément paramétrable, et d'autre part qu'il, prend en compte la caméra *QuickCam Messenger de chez Logitec* et qu'il est gratuit. D'après la figure 7

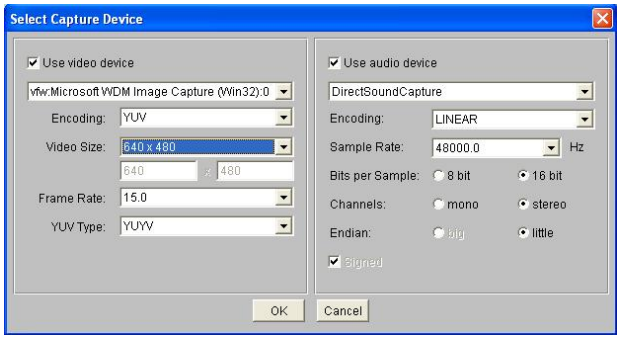

FIG. 7: Paramètrage de l'outil JMStudio

on peut remarquer que l'on peut capturer l'audio et le diffuser en multicast mais cette fonctionnalité n'a pas été utilisée dans l'application.

La taille de la Vidéo est paramétrable. La taille choisie par notre application est *640*800*, et l'interface Homme-Machine s'adapte automatiquement à cette interface.
On peut aussi interagir sur le nombre d'images que l'on veut envoyer sur le réseau par unité de temps. A partir de 10 images par seconde, l'opérateur commence à ressentir une bonne qualité vidéo. Un inconvénient de la solution retenue est le multicast. En effet, sans l'autorisation des administrateurs réseau le multicast ne peut pas passer.

6.2.2 Station robot

La station robot est embarquée sur le robot tout comme le serveur vidéo. Elle permet de piloter localement le robot grâce aux ordres envoyés par la station du LORIA. Un programme en C++ a été développé sur cette station et permet d'actionner la caméra et le robot.

6.2.3 Station cliente

La station cliente est le PC qui est installé au LORIA. Le rôle de cette station est d'héberger l'interface Homme-Machine qui est un programme réalisé en Java. L'interface Homme-Machine récupère les coordonnées délivrées par l'oculomètre, elle traite ces coordonnées et envoie les consignes de rotation de

la caméra du robot mobile. Cette interface Homme-Machine récupère le flux vidéo fourni par la caméra embarquée sur le robot.

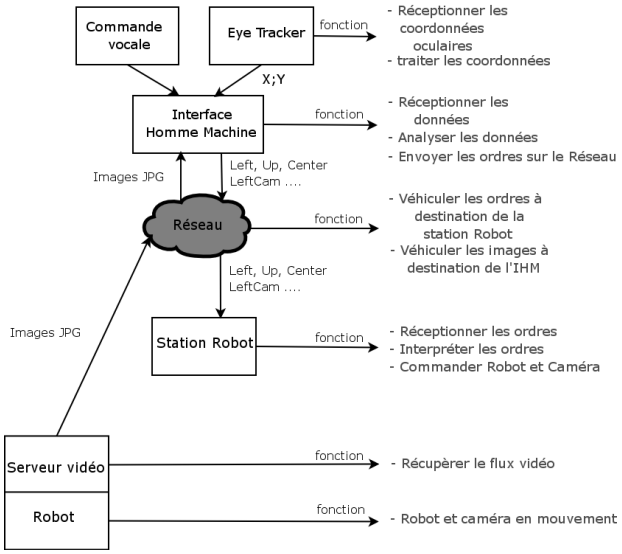

FIG. 8: Architecture de communication

La figure 8 présente l'architecture de communication de l'application. Les communications inter-stations utilisent le protocole UDP non connecté. Les ordres sont des chaînes de caractères encapsulées dans des paquets UDP.

7 Filtrage des consignes

Les données fournies par l'oculomètre doivent être filtrées pour élaborer une consigne valide. En effet, le regard a un double rôle dans cette application : explorer visuellement la scène distante et piloter la caméra. Il s'agit donc de distinguer l'une de l'autre ces deux types d'activité oculaire. Pour limiter le moins possible l'activité d'exploration visuelle de l'opérateur, nous avons choisi, dans un premier temps, d'interpréter comme des commandes d'orientation de la caméra uniquement les fixations oculaires qui se situent à la périphérie de l'écran où s'affiche le flux vidéo. Une étude expérimentale ergonomique permettra de raffiner cette stratégie grâce à l'analyse du comportement visuel spontané d'utilisateurs.

7.1 Elaboration de la consigne

Dans l'application, les coordonnées des points de l'écran cibles de fixations oculaires interprétées comme des commandes (précision de 1° d'angle visuel) sont envoyées à l'IHM. Ensuite, l'IHM envoie les ordres correspondants à la station robot qui interprète ces ordres et pilote le mouvement de la caméra par paliers angulaires.
Les coordonnées oculaires prélevées par l'oculomètre ont des abscisses et des ordonnées comprises entre 0 et 100. Par exemple

si l'échantillon prélevé par l'oculomètre a une abscisse entre 0 et 10 et une ordonnée entre 10 et 90 la commande sera une rotation de la caméra à gauche
Si l'échantillon prélevé par l'oculomètre a une abscisse entre 10 et 90 et une ordonnée comprise entre 10 et 90 il n'y aura pas de commande. Si l'échantillon prélevé par l'oculomètre a une abscisse entre 0 et 10 et une ordonnée comprise entre 0 et 10 on va avoir deux commandes successives : caméra vers le haut, caméra vers la gauche.

Ici on a deux problèmes à résoudre : comment régler la période d'émission des consignes, et quel angle de rotation devra parcourir la caméra à l'exécution de ces consignes ?

7.2 Performances de la caméra

Pour identifier avec précision les performances, en terme d'orientation, de la caméra, on a effectué des mesures du temps d'exécution d'une consigne. Le seul paramètre sur lequel on peut agir, au niveau de la commande de la caméra, est l'angle panoramique parcouru à chaque exécution d'une commande. On a donc mesuré le temps d'exécution de chaque consigne suivant l'angle panoramique choisi.

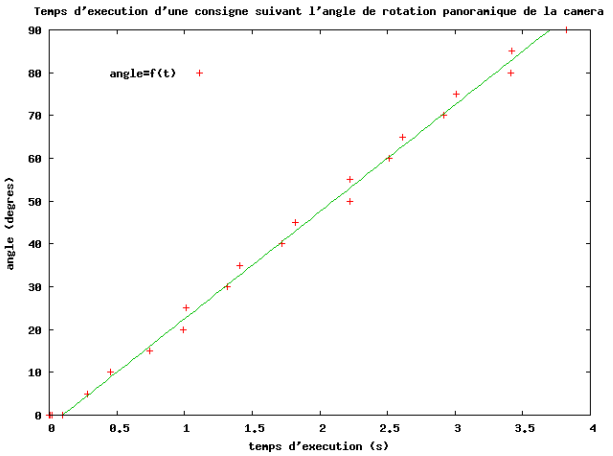

FIG. 9: Temps d'exécution d'une consigne suivant l'angle panoramique

La courbe 9 représente le temps d'exécution d'une consigne suivant l'angle panoramique d'exécution. Ce temps d'exécution comprend principalement le temps du parcours réseau de la consigne visuelle à destination de la station robot, le temps d'analyse de cet ordre par la station robot et le temps de traitement du robot pour effectuer cet ordre. L'analyse de cette courbe a permis de déduire comment paramétrer la fréquence à laquelle les consignes vont être émises sur le réseau par rapport à l'angle panoramique.

7.2.1 Analyse des résultats expérimentaux

Les résultats, visualisés sur la figure 9, montrent que le temps d'exécution d'une consigne est proportionnel à l'angle panoramique choisi. Grâce à ces résultats expérimentaux, il est possible d'adapter la fréquence d'émission des consignes par rapport à l'angle panoramique choisi. La courbe représente une limite de fonctionnement de l'application, il est clair que si la fréquence d'émission des consignes est mal réglée par rapport à l'angle panoramique, l'asservissement en position ne se fera pas. Voici un exemple expérimental : pour qu'à l'exécution d'une consigne la caméra parcourt un angle de 5 degrés, il faut attendre 200 ms avant d'envoyer une nouvelle consigne. Si cette règle n'est pas respectée, il n'y a pas d'asservissement visuel, la caméra ne répond pas aux commandes. L'augmentation de l'angle panoramique de rotation de la caméra entraîne l'augmentation du délai à respecter avant l'envoi d'une nouvelle consigne. Plus l'angle panoramique est grand, plus il faut envoyer les commandes lentement.
La droite a été obtenue par régression linéaire des points de mesure expérimentaux, et le coefficient directeur de cette droite représente la vitesse angulaire de la caméra. En effet, comme la caméra est un outil léger, l'inertie de la caméra est négligeable, la vitesse peut donc être considérée constante. Par conséquent, le temps d'exécution d'une consigne est proportionnel à l'angle panoramique choisi.

7.2.2 Formalisme mathématique de l'analyse des résultats expérimentaux

Il est possible de formaliser les résultats expérimentaux. Pour cela, il faut définir 4 variables :
- le temps d'exécution de la consigne :T_Θ
- le temps de traitement et de parcours réseau : T_Φ
- l'angle panoramique : θ
- la vitesse angulaire : Ω
- la période de la fréquence d'émission des consignes : $P_{T,\Theta}$

$$T_\Theta = T_\Phi + \theta/\Omega \tag{1}$$

D'après l'équation (1), le temps d'exécution d'une consigne correspond à la somme des temps de traitement et au quotient de l'angle panoramique et de la vitesse angulaire de la caméra.

$$P_{T,\Theta} > T_\Phi + \theta/\Omega \tag{2}$$

L'équation (2) formalise le bon fonctionnement de l'application.

$$P_{T,\Theta} < T_\Phi + \theta/\Omega \tag{3}$$

Si les paramètres de l'application suivent l'inégalité (3), on observera un fonctionnement altéré. En effet, si la période de la fréquence d'émission est trop petite par rapport à l'angle panoramique parcouru par la caméra à l'exécution de cette consigne, il n'y aura pas d'asservissement en position. Si la caméra reçoit une consigne alors qu'elle est déjà en train d'en exécuter une, elle se bloquera ou elle ne tiendra pas compte de cette nouvelle consigne.

$$P_{T,\Theta} = T_\Phi + \theta/\Omega + \epsilon \tag{4}$$

Dans l'équation (4), ϵ est un réel strictement positif. Si les paramètres de l'application respectent l'équation (4), on obtient un fonctionnement optimal de l'application. ϵ est négligeable mais il est nécessaire. En effet, la présence de cet epsilon garantit que la période de la fréquence d'émission des consignes sera très légèrement supérieure à l'égalité stricte : $P_{T,\Theta} = T_\Phi + \theta/\Omega$, ce qui permet à l'opérateur de gagner en confort d'interaction Homme-Machine car l'asservissement en position se fait de façon optimale.

$$P_{T,\Theta} < T_\Phi \tag{5}$$

Si les paramètres de l'application vérifient l'inégalité (5), la caméra n'exécutera aucune consigne même si l'angle panoramique choisi est de l'ordre de 1°.
Si l'on fixe l'angle panoramique que la caméra doit parcourir pour exécuter une consigne, on peut calculer la période de la fréquence d'émission optimale des consignes délivrées par l'IHM en utilisant la formule (6).

$$P_{T,\Theta} = T_\Phi + \theta/\Omega \tag{6}$$

Et inversement, si l'on fixe la période de la fréquence d'émission des consignes, on peut calculer l'angle panoramique optimal que la caméra doit parcourir à l'exécution de la consigne pour asservir au mieux le système en utilisant la formule (7).

$$\theta = \Omega.(T_\Theta - T_\Phi) \tag{7}$$

8 Conclusion

Le démonstrateur présenté est opérationnel. Nous comptons effectuer prochainement des tests d'utilisabilité de cette nouvelle forme d'interaction homme-machine multimodale qui associe le regard à la commande manuelle pour contrôler un robot mobile équipé d'une caméra orientable. Cette étude nous permettra d'ajuster les paramètres du prototype actuel en vue d'améliorer le confort visuel de l'utilisateur et d'accroître l'efficacité de l'interaction. Nous aborderons ensuite le contrôle du zoom de la caméra qui, nous semble-t-il, s'exprime plus " naturellement " par le geste que par le regard : on se rapproche/s'éloigne de l'objet d'intérêt, ou bien l'objet effectue ces déplacements sous le contrôle de l'utilisateur. La miniaturisation des équipements qui ne cesse de progresser permet d'envisager à moyen terme des dispositifs oculométriques portables qui présentent la

même robustesse et la même fiabilité que les oculomètres, actuels mais dont le port n'impose pas plus de contraintes que celui d'une paire de lunettes.

Références

[1] Y. Rybaczyk, D. Mestre, P. Hoppenot, and E. Colle, "Implémentation de mécanismes d'anticipation visuo-motrice en téléopération. le travail humain, 67(3), 209 :234," 2004.

[2] L. Zalud, "Integration of 3d proximity scanner to orpheus robotic system," 2005, 2005 IFAC.

[3] L. Zalud, "Universal autonomous and telepresence mobile robot navigation," in *International Symposyum on Robotics*, Seoul, Korea, 2001, pp. 1010–1015, 2001 IFAC.

[4] A. Krupa, C. Doignon, J. Gangoloff, M. de Mathelin, and G. Morel, "A vision system for automatic 3d positioning of surgical instruments for laparoscopic surgery with robot," Bourges, France, juin 2002, 12$^{\text{th}}$ International Symposium of Measurement and control Robotics.

[5] A. Krupa, C. Doignon, J. Gangoloff, M. de Mathelin, and G. Morel, "Autonomous retrieval and positioning of a surgical tool in robotized laparoscopic surgery using visual servoing," Wachington D.C., USA, mai 2002, ICRA 2002, Video Proceedings.

[6] P. Majaranta and K.J. Räihä, "Twenty years of typing : Systems and design issues," in *ACM Symposium on Eye Tracking Research and Applications*, A. Duchowski, R. Vertegaal, and J.W. Senders, Eds., New Orleans, LA, March 2002, pp. 15–22, New York : ACM Press.

[7] A. T. Duchowski, *Eye Tracking Methodology : Theory and Practice*, Springer, 2003.

[8] L.C. Loschky and G.W. McConKie, "User performance with gaze contingent multiresolutional displays," in *ACM Symposium on Eye Tracking Research and Applications*, A. Duchowski, Ed., Palm Beach Gardens, FL, November 2000, pp. 97–103, New York : ACM Press.