Leap Motion pour la capture de mouvement 3D par spline L_1. Application à la robotique

F. Hernoux[1], R. Béarée[1], L. Gajny[1], E. Nyiri[1], J. Bancalin[3], O. Gibaru[1,2]

[1] Arts et Métiers ParisTech, LSIS - UMR CNRS 7296, 8 Boulevard Louis XIV, 59046 Lille, France
[2] INRIA Lille-Nord-Europe, Equipe NON-A, 40 avenue Halley 59650 Villeneuve d'Ascq, France,
[3] Plateforme Technologique Usine Agile http ://www.usine-agile.fr

Résumé

Afin d'accroître leur compétitivité les entreprises ont recours de plus en plus à des systèmes robotisés pour réaliser différentes tâches complexes. Ces robots sont très attractifs par leur coût mais nécessitent d'introduire des capteurs externes afin de garantir une plus grande précision de mouvement. Un enjeu majeur concerne la co-activité avec l'homme. Fort de l'acquisition très récente du système de vision low-cost, Leap Motion, qui présente des caractéristiques de précision inégalées à ce coût, nous proposons un premier travail d'apprentissage, par un système robotisé, de la gestuelle d'un opérateur. L'objectif est de reproduire des tâches complexes en 3D sans contraintes pour l'opérateur. Cette interaction permet d'engendrer un nuage de points et de directions très précises. Afin de garantir une bonne répétabilité du mouvement sur notre robot UR10, nous réalisons une interpolation de ces données par des splines polynomiales minimisant la norme L_1. Ce formalisme développé récemment présente une complexité de calcul linéaire avec les données et permet de conserver la forme des données même lorsque le pas de discrétisation n'est pas uniforme en espace.

1. Introduction

L'accroissement de la compétitivité des entreprises passe par l'adaptation des lignes de production au phénomène de *Mass Customization*, qui peut se traduire par une nécessaire agilité ou flexibilité de la ligne. Dans ce contexte, la demande en faveur de l'exploitation de robots industriels pour des tâches complexes est en pleine croissance. Les robots sont capables de réaliser des opérations spécifiques avec ou à la place d'humains. Une nouvelle génération de robots est apparue ces dernières années, comme ceux de la société Danoise *Universal Robots* ou encore les robots Baxter de la société américaine *Rethink Robotics*. Ces derniers se distinguent radicalement de leurs aînés sous plusieurs aspects. Ils sont intrinsèquement adaptés à une collaboration homme-machine. En effet, ils peuvent limiter les efforts qui s'exercent sur leurs axes. Ainsi, il devient possible de supprimer les grilles de protection habituelles. Leur programmation est intuitive et ergonomique. Enfin, ils admettent des prix relativement bas. Ces robots collaboratifs sont de fait très attractifs, notamment afin de mettre la robotisation à la portée des PME, mais également d'effectuer des tâches impossibles à réaliser auparavant.

Même si ces robots peuvent désormais travailler à proximité des humains et ce en toute sécurité, leur exploitation dans le cadre de tâches complexes est généralement limitée par leur précision. La gestion efficace de l'*humain dans la boucle* nécessite alors l'exploitation de capteurs externes précis et rapides. Dans ce papier, nous proposons d'exploiter un capteur, low-cost, d'un tout nouveau genre et encore en développement, le **Leap Motion**. Ce périphérique, qui sera commercialisé prochainement, permet de capturer les mouvements des doigts de l'utilisateur à un taux de rafraichissement pouvant aller jusqu'à 200 frames par seconde et une précision affichée inférieure au millimètre. Ce périphérique est donc, *a priori*, un outil idéal pour retranscrire en temps réel le mouvement humain sur un robot.

Nous proposons ici un premier travail d'apprentissage par un système robotisé de la gestuelle d'un opérateur permettant de reproduire des tâches complexes en 3D. Le robot peut ainsi suivre la position et l'orientation d'un doigt de l'utilisateur. Afin de garantir une bonne répétabilité du mouvement sur notre robot de type UR10, nous réalisons une interpolation de ces données (points et tangentes) par des splines polynomiales minimisant la norme L_1 de leur dérivées se-

conde. Ces splines ont été introduites dans [Lav00] pour la première fois.

Dans la première partie, nous présentons les techniques utilisées pour reproduire le mouvement humain, ainsi que les caractéristiques de mesure du nouveau périphérique Leap Motion. Dans la seconde partie, nous présentons les données fournies par le Leap Motion. Puis, nous exposons notre méthode d'interpolation L_1 des données issues des mouvements des doigts. Enfin, nous exposons les premiers résultats obtenus dans le cadre de l'utilisation de ces données avec un robot collaboratif.

2. Moyen de mesure 3D du mouvement des mains

2.1. Etat de l'art Technologique

De nombreux périphériques permettent de capturer en 3D les mouvements des mains avec plus ou moins de précision. Ces systèmes reposent sur diverses technologies comme les exosquelettes mécaniques [Stu92], pneumatiques [BPBB02], à câbles [HTH*06], les gants de données à fibre optique [SZ94], à capteurs de flexion [KHW95], à accéléromètres [LTD11], les trackers magnétiques [MMJ*11], à ultrasons [SZ94] ou encore les systèmes optiques comme les marqueurs actifs ou passifs [GP09].

Si tous ces périphériques permettent généralement plus de 6 degrés de libertés, en admettant l'utilisation d'un nombre suffisant de capteurs, ils nécessitent cependant tous l'usage de matériel à fixer sur la main de l'utilisateur [Her11]. On peut donc les qualifier de périphériques *non-transparents* car ils nécessitent le port de capteurs afin de déterminer les mouvements effectués. Ce matériel lourd et gênant limite les mouvements de l'utilisateur. De plus, ils sont sensibles à de nombreuses sources de bruit : métal, bruit, lumière et poussière. Ils présentent une précision absolue de l'ordre du millimètre voire du centimètre pour certains. Des systèmes optiques permettent, par exemple, de réaliser une capture des mouvements très précise mais leur prix peut varier de 15 000$ à plus de 250 000$.

Des solutions low-cost de capture des mouvements ont commencé à apparaître avec l'avènement de la Kinect de Microsoft© en novembre 2010. Cette technologie repose sur l'usage d'une caméra 3D par stéréoscopie et triangulation active [MPS07]. Elle permet d'atteindre une précision absolue de l'ordre de 1 à 5 centimètres selon la distance. Elle présente aussi le très grand avantage d'être non intrusive, de travailler dans le spectre infrarouge et d'être, par conséquent, totalement transparente pour l'utilisateur. De plus, bien que reposant sur de la vision par ordinateur, le système peut également opérer de nuit grâce à l'infrarouge. La Kinect est ainsi un périphérique permettant de capturer les mouvements de l'utilisateur en 3D pour un moindre coût, environ 250$. Cependant, elle souffre de plusieurs inconvénients tels qu'un faible taux de rafraichissement de 30 images par seconde et une sensibilité aux objets réfléchissants. De plus, elle ne permet de capturer que les mouvements du corps. Par exemple, les mouvements des doigts de la main ne sont pas gérés par le SDK actuel.

2.2. Présentation du capteur Leap Motion

Le système que nous nous proposons d'exploiter repose, quant à lui, sur l'utilisation d'un périphérique d'un tout nouveau genre : le Leap Motion © 2012 présenté à la figure 1. Les caractéristiques de ce petit périphérique de 8cm x 3cm x 1cm en font un appareil très intéressant pour la capture précise des mouvements des doigts. A ce stade des développements, ce périphérique est capable de mesurer à la fois la position et l'orientation dans l'espace des doigts ou de tout objet long et fin comme des bâtons ou des stylos. Il est même possible de différentier un objet d'un doigt. La précision affichée du positionnement est de l'ordre de $1/10$ de millimètre selon les 3 axes dans un volume cubique d'environ 60cm de côté. Nous verrons par la suite que cela correspond à la répétabilité du capteur et non à sa précision absolue.

Le taux de rafraichissement du système peut aller de 30 à 200 images par seconde selon la puissance de l'ordinateur utilisé. La distance de détection est comprise entre 2cm et 70cm avec un champ de vision de 110° actuellement. Ces caractéristiques doivent passer prochainement à 1m et 140°. Le système peut fonctionner à travers des surfaces comme le verre ou le plexiglass. Il est possible de chainer plusieurs de ces périphériques afin d'accroître la zone de détection. Le Leap Motion se connecte à un ordinateur par un câble USB. Il est ainsi alimenté en courant. Il peut envoyer les informations 3D à l'ordinateur. Le SDK fourni avec le Leap Motion permet aux développeurs de récupérer la position et l'orientation de la main et de chaque doigt ou de tout objet de type pointeur. Il capture même les mouvements de chaque doigt de façon indépendante. Il peut suivre ainsi jusqu'à une quinzaine de doigts. Enfin, le prix de vente sera de l'ordre de 70$. Ce qui en fait un périphérique à la fois précis, peu onéreux et innovant.

On notera que ce périphérique ne permet de détecter un objet que si ce dernier entre dans son champ de mesure en passant par le plan XY (Cf. figure 3).

Figure 1: *Carte électronique du Leap Motion (à gauche) et dans son boîtier (à droite).*

Dans la suite, nous avons restreint notre étude aux données fournies par le Leap Motion lors de la réalisation d'un

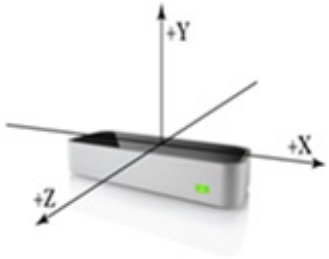

Figure 2: *Le repère associé à la Leap Motion.*

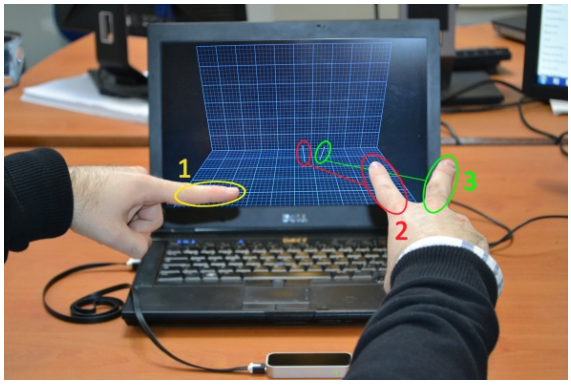

Figure 3: *Un objet entrant par le plan YZ (doigt 1) n'est pas détecté contrairement aux doigts 2 et 3.*

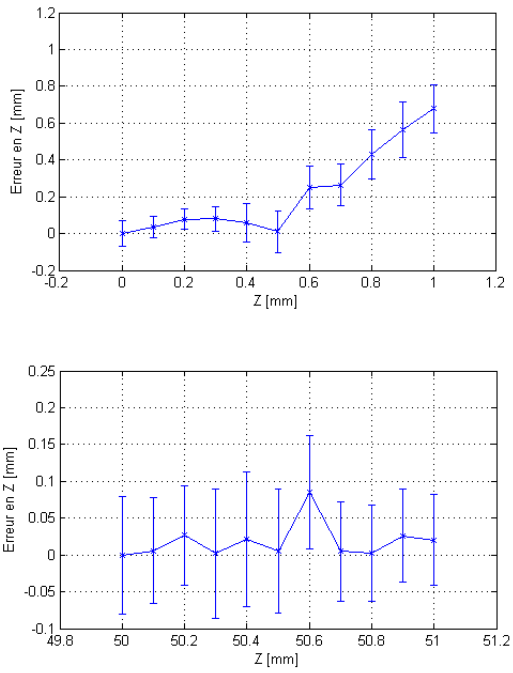

Figure 4: *Mesure de l'erreur suivant l'axe Z ($X = 0$,$Y = 100mm$) et écarts types associés.*

mouvement d'un unique doigt. Nous avons donc développé en C++, un module qui permet de détecter si un doigt est présent dans le champ de vision de l'appareil et de récupérer sa position et son orientation dans le repère du Leap Motion (Cf. figure 2). L'origine du repère est définie comme le centre de l'appareil.

Nous avons réalisé une campagne de mesures afin de caractériser la précision de positionnement atteignable sur le Leap Motion. Les mesures ont été effectuées dans le plan XZ pour différentes hauteurs Y. La figure 4 présente l'erreur obtenue pour un déplacement selon l'axe Z situé à $100mm$ au-dessus du dispositif. L'écart type en chaque valeur de Z est estimé à partir de 100 mesures.

Ces résultats montrent une précision absolue de $0,7mm$ pour des points mesurés à une distance inférieure au *cm* par rapport au centre du repère du capteur. Ce constat est généralisable dans les autres directions. Ces premières analyses conduisent à une répétabilité à un σ de $0,125mm$. La répétabilité du capteur semble cohérente avec les données constructeur. Elle est aussi en accord avec notre approche de la reproduction de la gestuelle humaine. Une analyse métrologique de ce capteur est en cours.

3. Reproduction du mouvement des doigts

3.1. Mouvement des doigts : données brutes issues du Leap Motion

La figure 5 illustre un mouvement d'écriture simple mesuré en 3D par le Leap motion. La figure 6 présente le déplacement d'un doigt dans l'espace ainsi que son orientation en chacun des points enregistrés. Pour ce mouvement, un changement d'orientation a volontairement été introduit afin de vérifier la détection correcte de la direction du doigt (zoom sur la figure 6).

Le protocole d'acquisition actuel ne permet pas d'assurer un échantillonnage temporel constant. Ainsi, même à vitesse quasi-constante, certaines zones sont dépourvues de points de mesure. Afin de permettre la reproduction du mouvement par un système industriel de type robot anthropomorphe, il est indispensable d'interpoler les données mesurées en garantissant la meilleure préservation possible de la trajectoire initialement décrite. Dans la suite, nous proposons d'exploiter des splines minimisant la norme L_1 de leur dérivée seconde pour effectuer cette interpolation. Ces splines présentent en effet la propriété de bien préserver la forme des données.

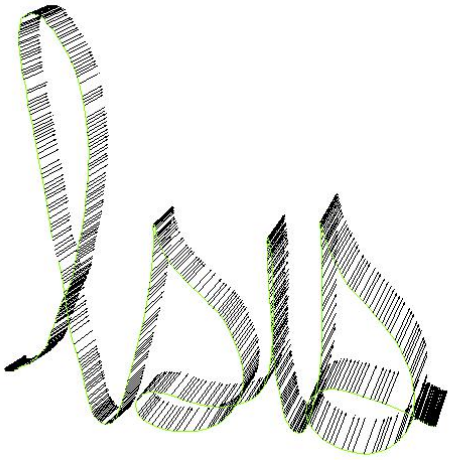

Figure 5: *Données mesurées pour un mouvement d'écriture à vitesse normale.*

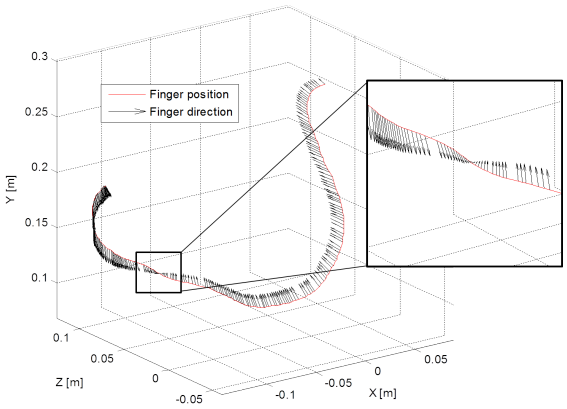

Figure 6: *Position et direction du doigt lors d'un mouvement 3D.*

3.2. Interpolation $L_1 - C^2$ du mouvement des doigts

Nous considérons d'abord le cadre des splines cubiques paramètriques [Auq07]. Les résultats théoriques obtenus pour ces splines cubiques serviront de base pour définir des splines quintiques admettant une régularité de raccord C^2 et des bonnes propriétés de préservation de la forme des données. Nous nous appuyons ici essentiellement sur les résultats de [NGA11]. Ce niveau de régularité C^2 est nécessaire afin de minimiser les sollicitations d'ordre dynamique de la structure du robot lors du mouvement [OBGD10].

Définition 1 Soit $u = \{u_i\}_{i=1,\dots,n}$ $(n \in \mathbb{N})$ une suite arbitraire monotone réalisant une partition de $[0,1]$. Nous appelons spline cubique paramètrique C^1 toute application γ satisfaisant :

i) γ est une application polynomiale de degré inférieur ou égal à 3 sur chaque $[u_i, u_{i+1}[\to \mathbb{R}^d \left(d \in \mathbb{N}^*\right)$

ii) γ est de classe C^1 sur $[0,1]$.

Nous notons $\mathcal{S}_{3,u}$ cet espace vectoriel.

Les points 3D $Q = (q_i)_{i=1,\dots,n}$ donnés par le Leap Motion permettent de définir l'ensemble suivant

$$\mathcal{I}_Q, u = \left\{\gamma \in \mathcal{S}_{3,u} \mid \gamma(u_i) = q_i \text{ pour } i = 1,\dots,n\right\}.$$

Afin de limiter le nombre de degré de liberté de cet ensemble, nous fixons la séquence u selon une partition chordale prenant en compte les distances relatives entre les points. Des partitions plus spécifiques, prenant en compte aussi les angles relatifs entre les points peuvent être aussi utilisées. Il reste ainsi comme degrés de liberté les valeurs des vecteurs des dérivés aux u_i. Cet ensemble $\mathcal{I}_Q, u$ admet donc une infinité d'éléments. Classiquement, il est proposé de minimiser un critère en norme L_2 afin de sélectionner une solution *via* la résolution d'un système linéaire. J. Lavery en 2000 a proposé dans [Lav00] de minimiser un critère basé sur la norme L_1 défini comme suit

$$\inf_{\gamma \in \mathcal{I}_Q, u} \int_0^1 \left\| \frac{\mathrm{d}\gamma^2}{\mathrm{d}u^2}(u) \right\|_1 \mathrm{d}u. \qquad (1)$$

Cette minimisation conduit à un problème non linéaire issu de l'utilisation des valeurs absolues. Cependant, une résolution numérique par algorithme de minimisation de type point-intérieur (Cf. [Van89]) permet de montrer que ce type de fonctionnelle donne des résultats numériques intéressant concernant la préservation de la forme et le respect des alignements des données. Il a été montré dans [AGN07] que le phénomène de sur-oscillation ou phénomène de Gibbs est évité lorsque les données présentent des variations brusques des écarts relatifs.

En considérant le formalisme spline cubique, nous montrons que le problème de minimisation (1) peut s'écrire sous la forme

$$\min_{b_i \in \mathbb{R}^d} \sum_{i=1}^{n-1} \int_{-\frac{1}{2}}^{\frac{1}{2}} |\Delta b_i + 6t(b_i + b_{i+1} - \frac{2}{\Delta u_i}\Delta q_i)| \, \mathrm{d}t, \qquad (2)$$

où les b_i sont les vecteurs des dérivées inconnus en chaque point donné par la Leap Motion. L'opérateur Δ désigne l'opérateur des différences progressives i.e. $\Delta q_i = q_{i+1} - q_i$. Comme la fonctionnelle n'est pas strictement convexe, il n'y a pas nécessairement unicité de la solution. La résolution numérique de ce problème peut conduire aussi à des problèmes de robustesse pour un ensemble de points Q important.

Dans [NGA11], nous avons proposé une nouvelle méthode de résolution basée sur une minimisation locale sur une fenêtre de cinq points. Le principal avantage de cette méthodologie est que nous avons été capable de résoudre formellement la solution de (1) sur cinq points. Par conséquent, la complexité de calcul est devenue linéaire avec les données. Cette méthologie locale s'est avérée intéressante puisqu'elle permet de conserver les bonnes propriétés de conservation des alignements, de non oscillation et d'invariance par rotations des splines obtenues par une résolution dite globale. En appliquant itérativement cet algorithme sur les dérivées premières b_i calculées par cette méthode, il est possible de calculer avec une complexité linéaire des dérivées secondes aux points d'interpolation. Nous appliquons la même méthodologie afin d'interpoler l'ensemble des orientations du doigt. Finalement, nous calculons une spline quintique d'interpolation des points présentant de bonnes propriétés de design géométrique et une spline cubique d'interpolation des orientations.

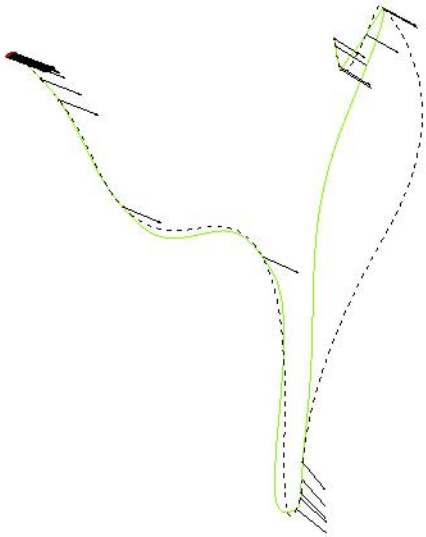

Figure 7: *Comparaison entre une spline d'interpolation L_2 en pointillé et une spline L_1 en continue.*

La figure 7 montre que la spline d'interpolation obtenue par minimisation L_1 présente moins d'oscillations que la spline $L2$ dans ce cas où les données varient avec de fortes amplitudes au centre. Ces fortes variations sont dues essentiellement à la fréquence de traitement des données issues du Leap Motion qui était trop basse par rapport à la vitesse de déplacement de l'extrémité du doigt dans cette zone.

4. Reproduction du mouvement par un robot 6 axes collaboratif

Pour reproduire la gestuelle, nous avons choisi d'utiliser un robot UR10 vendu par *Universal Robot* et présenté à la figure 8. Ce système est à la fois peu onéreux (ce qui reste dans l'esprit low-cost du système de capture des mouvements) et en même temps ergonomique et sécurisé. Bien que peu dispendieux, ce bras robotique reste toutefois flexible et hautement spécialisé. Il peut être utilisé dans quasiment n'importe quelle industrie où les robots traditionnels sont trop grands, chers, bruyants ou pas assez flexibles. Il ne nécessite pas d'installation ou de configuration lourde. L'UR10 est un robot industriel anthropomorphe à 6 axes de $29kg$. Il permet de soulever des charges allant jusqu'à $10kg$. Il a été spécialement développé pour les petites et moyennes entreprises qui ont besoin d'une automatisation flexible, efficace et rapidement rentable. Ces robots sont faciles à déplacer et ils ne nécessitent pas une base particulièrement solide étant donné leur faible poids. L'UR10 est un bras robotisé qui peut, conformément à la réglementation en vigueur, être en service sans protection particulière. Ainsi, il est parfaitement adapté à la notion de collaboration homme/machine. Les actionneurs d'axe du robot sont dimensionnés afin de limiter tout impact sur l'environnement ou sur un homme à la valeur maximale de $150N$.

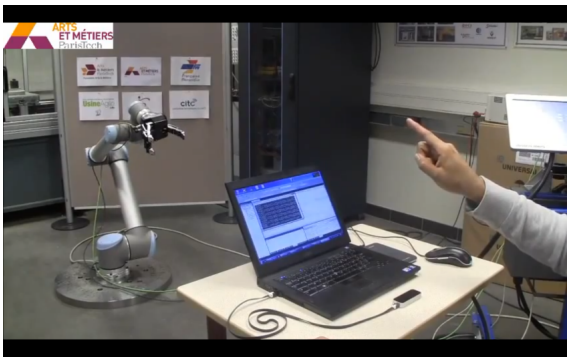

Figure 8: *Robot UR10 et son interface de programmation.*

Il est possible de piloter le robot soit à l'aide de son interface utilisateur graphique, soit au travers de scripts (sur l'UC du robot lui-même), ou encore grâce à une API en C à partir d'un ordinateur délocalisé au moyen d'une connexion TCP / IP classique. Le langage de programmation du robot est l'URScript. L'URScript contient un certain nombre de variables intégrées et des fonctions assurant la surveillance, le contrôle des entrées / sorties ainsi que les mouvements du robot. Les programmes URScript sont exécutés en temps réel dans le RuntimeMachine de l'URControl (RTMachine). Le Runtime Machine communique avec le robot avec une fréquence de $125Hz$.

Les trajectoires du robot sont générées en ligne en appelant des fonctions de déplacement (movej, movel) et

des fonctions de vitesse (speedl, speedj). Les positions et les vitesses des articulations sont représentées directement comme des listes de six nombres flottants : un réel pour chaque articulation du robot. Il suffit alors d'envoyer au robot des vecteurs composés de 6 valeurs : 3 pour la position et 3 pour l'orientation de l'effecteur terminal.

Grâce au Leap Motion, nous pouvons *via* un programme simple, récupérer les positions et les orientations d'un doigt ou d'un outil. Nous composons alors un vecteur à 6 valeurs que nous envoyons par connexion TCP / IP au robot afin que l'effecteur reproduise le plus fidélement possible les mouvements du doigt de l'utilisateur. L'interpolation spline L_1 est calculée actuellement sur le PC après enregistrement de l'ensemble des positions en hors-ligne. La méthodologie de minimisation par fenêtre glissante permettra dans un travail futur une implémentation temps réel où la spline sera regénérée au fur et à mesure de l'obtention des données. Des calculs parallélisés sur carte GPU de l'algorithme d'interpolation L_1 présentés dans [NGA12] démontrent cette possibilité.

5. Conclusion

Nous avons utilisé le nouveau capteur Leap Motion afin de réaliser le design de trajectoires par un opérateur dans le but de copier la cinématique de son mouvement par robot. Nous avons montré dans le cas où la vitesse de traitement des informations de mouvement provenant du Leap Motion est trop faible que l'utilisation de spline L_1 peut être intéressante pour générer des trajectoires robot plus "tendues".

Références

[AGN07] AUQUIERT P., GIBARU O., NYIRI E. : On the cubic l1 spline interpolant to the heaviside function. *Numerical Algorithms. Vol. 46(4)* (2007), 321–332.

[Auq07] AUQUIERT P. : *Interpolation de points par des splines L^1 régulières*. PhD thesis, Université de Valenciennes et du Hainaut-Cambrésis, 2007.

[BPBB02] BOUZIT M., POPESCU G., BURDEA G., BOIAN R. : The rutgers master ii-nd force feedback glove. In *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* (2002).

[GP09] GOEBL W., PALMER C. : Finger motion in piano performance : Touch and tempo. In *International Symposium on Performance Science* (2009).

[Her11] HERNOUX F. : *Conception et évaluation d'un système transparent de capture de mouvements des mains pour l'interaction 3D temps réel en environnements virtuels*. PhD thesis, Arts et Métiers ParisTech, 2011.

[HTH*06] HASEGAWA S., TOSHIAKI I., HASHIMOTO N., SALVATI M., MITAKE H., KOIKE Y. E. A. : Human-scale haptic interaction with a reactive virtual human in a real-time physics simulator. *Comput. Entertain.. Vol. 4(3)* (2006).

[KHW95] KESSLER G. D., HODGES L. F., WALKER N. : Evaluation of the cyberglove as a whole-hand input device. *ACM Trans. Comput.-Hum. Interact.. Vol. 2(4)* (1995), 263–283.

[Lav00] LAVERY J. E. : Univariate cubic Lp splines and shape-preserving, multiscale interpolation by univariate cubic L1 splines. *Computer Aided Geometric Design. Vol. 17*, Num. 4 (avril 2000), 319–336.

[LTD11] LOBO J., TRINDADE P., DIAS J. : Observing hand grasp type and contact points using hand distributed accelerometers and instrumented objects. In *IEEE ICRA 2011 : Workshop on Autonomous Grasping, Shanghai, China* (2011).

[MMJ*11] MA Y., MAO Z.-H., JIA W., LI C., YANG J., SUN M. : Magnetic hand tracking for human-computer interface. *Transactions on Magnetics. Vol. 47(5)* (2011), 970–973.

[MPS07] MAY S., PERVOELZ K., SURMANN H. : 3d cameras : 3d computer vision of wide scope. *International Journal of Advanced Robotic Systems. Vol. 4* (2007), 181–202.

[NGA11] NYIRI E., GIBARU O., AUQUIERT P. : Fast $L_1^k C^k$ polynomial spline interpolation algorithm with shape-preserving properties. *Computer Aided Geometric Design. Vol. 28*, Num. 1 (janvier 2011), 65–74.

[NGA12] NYIRI E., GIBARU O., AUQUIERT P. : Nonlinear ${l_1 c}^1$ interpolation : Application to images. *Lecture Notes in Computer Science. Vol. 6920* (2012), 515–526.

[OBGD10] OLABI A., BÉARÉ R., GIBARU O., DAMAK M. : Feedrate planning for machining with industrial six-axis robots. *Control Engineering Practice. Vol. 18(5)* (2010), 471–482.

[Stu92] STURMAN D. J. : *Whole-hand input*. PhD thesis, Massachusetts Institute of Technology, 1992.

[SZ94] STURMAN D. J., ZELTZER D. : A survey of glove-based input. *IEEE Computer Graphics and Applications. Vol. 14(1)* (1994), 30–39.

[Van89] VANDERBEI R. J. : Affine-scaling for linear programs with free variables. *Math. Program.. Vol. 43*, Num. 1 (janvier 1989), 31–44.

Remerciements

Les auteurs remercient Xavier Helle pour son aide lors de l'intégration sur le robot UR10.