Contrôle gestuel d'un robot en mobilité

Marc Dupont
IRISA
Université de Bretagne Sud,
Campus de Tohannic
Vannes, France
marc.dupont@irisa.fr

Pierre-François Marteau
IRISA
Université de Bretagne Sud,
Campus de Tohannic
Vannes, France
pierre-francois.marteau@univ-ubs.fr

IHM'15, October 27-30, 2015, Toulouse, France
ACM 978-1-4503-3844-8/15/10.
http://dx.doi.org/10.1145/2820619.2825022

Résumé
Les progrès actuels de la robotique offrent des opportunités considérables, pas seulement pour les particuliers mais également les professionnels en mission, tels que des pompiers ou militaires. Cependant, ce type d'opérateurs doit rester focalisé sur sa mission et attend donc un moyen de contrôle qui soit peu intrusif et très intuitif. De plus, il est indispensable de livrer un système qui prend en compte la mobilité intrinsèque à ce type d'activité (marche, course, saut).

Nous présentons ici un système de reconnaissance gestuelle à base de gant de données, à l'apprentissage instantané sur une large gamme de gestes possibles. Nous l'avons évalué sur sept gestes dédiés au contrôle d'un robot mobile, avec des résultats tout à fait satisfaisants.

Mots-clés
Reconnaissance gestuelle; robotique; IMU; distances élastiques; DTW

ACM Classification Keywords
H.5.2. [User Interfaces]: Input devices and strategies; I.5.2. [Pattern Recognition]: Design methodology; I.5.4. [Pattern Recognition]: Applications.

General Terms

Reconnaissance gestuelle ; robotique ; IMU ; distances élastiques ; DTW

Introduction

Les progrès technologiques actuels poussent le secteur de la robotique à devenir omniprésent dans notre société. Pour que le robot soit une réelle assistance à l'homme, il est nécessaire de viser l'efficacité et la compréhension dans les moyens de communication entre l'homme et la machine, afin que cette dernière soit en cohérence avec les désirs exprimés par l'humain qui la contrôle.

Notre cas d'utilisation est lié au développement d'un robot terrestre mobile, dont l'objectif est d'aider des opérateurs professionnels dans des missions qui peuvent parfois avoir de lourdes conséquences, tels que les secours (pompiers...), les forces de l'ordre, ou les forces militaires en opération.

L'objectif est le suivant : piloter un robot grâce au geste. Il peut s'agir de commandes de haut niveau ("viens ici") ou de plus bas niveau ("tourne à gauche").

Nous ne pouvons envisager la reconnaissance gestuelle vidéo en raison des contraintes de l'environnement variable auquel est confronté le robot et de son champ de vision. Ainsi, nous avons préféré mettre au point un système de reconnaissance gestuelle basé sur un *gant de données*, c'est-à-dire muni de capteurs. Les signaux de posture et de dynamique de la main sont captés directement sur le gant puis transmis au robot via une interface sans fil.

Nous avons réalisé nos essais avec le gant V-Hand 3.0 de DG5[1]. Celui-ci contient un capteur de flexion pour chaque doigt et une centrale inertielle (IMU, Inertial Measurement Unit).

1. http://www.dg-tech.it/vhand3/products.html

Travaux antérieurs

Des travaux antérieurs de reconnaissance gestuelles ont été menés sur des gants de données [2, 3, 7, 12], mais jusqu'ici aucun ne considère le problème de la mobilité. De plus, les gestes font souvent partie d'une gamme assez restreinte à cause des algorithmes sous-jacents ; nous cherchons au contraire à laisser le maximum de liberté à l'utilisateur dans le choix des gestes qu'il désire apprendre au système. Cela passe également par la prise en compte de l'orientation qui est bien souvent négligée alors qu'elle apporte une sémantique supplémentaire.

Certains systèmes se basent aussi sur des IMU, mais en utilisant un téléphone portable à la place du gant de données : [11] utilise DTW sur un geste unique de longueur fixe ; [1] impose 8 gestes détectés par seuillage sur les composantes dynamiques ; enfin, [8, 9] présentent deux techniques de *template matching* invariantes à la rotation.

Principe de fonctionnement

Le premier étage de *pré-traitement* analyse les données livrées par les capteurs et en extrait un faible nombre de caractéristiques :

- la *dynamique* en décrit sa variation : accélération, vitesse de rotation ;
- la *posture* décrit le positionnement des doigts ;
- l'*orientation* décrit par exemple si la main fait face au sol ou au ciel.

Puisque ces données arrivent à intervalles réguliers, on peut considérer que l'on a affaire à un flux de vecteurs multidimensionnels.

Le deuxième étage de *détection* a pour objectif d'analyser ce flux entrant et de déclencher un signal s'il reconnaît un des gestes appris. Dans le cas contraire, le système doit détecter le "vide", lorsque l'opérateur n'effectue aucun geste connu.

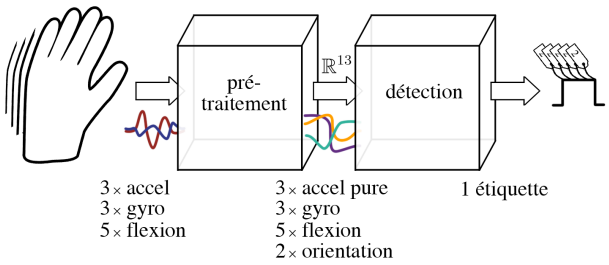

Figure 1 – L'étage de pré-traitement et l'étage de détection fonctionnent en *pipeline* pour la reconnaissance en temps réel.

Prétraitement

L'étage de prétraitement cherche à éliminer la composante dûe à la gravité dans les valeurs de l'accéléromètre, afin d'obtenir deux caractéristiques qui ne sont pas directement fournies par le capteur :

- l'accélération pure (sans gravité) ;
- l'orientation : scalaires indiquant la position de la main par rapport au sol, et au ciel.

Cela est réalisé grâce à un système estimation-correction similaire à [10]. Lors de la phase d'estimation, l'intégration continue des gyroscopes donne la rotation du repère capteur par rapport au repère terrestre. Lors de la phase de correction de type ZUPT (Zero-Velocity Update [6]), le repère estimé est recalé par rapport à la gravité.

Détection en flux

L'étage de détection est un algorithme d'apprentissage automatique qui s'apparente à un classifieur[2]. D'une part, le système est entraîné avec un flux d'exemples étiqueté ; d'autre part, il analyse un flux entrant et assigne une étiquette à chaque vecteur entrant.

Pendant l'apprentissage, on fournit au système des sous-séquences w étiquetées l. L'apprentissage se résume alors à stocker en mémoire chaque sous-séquence extraite et son étiquette : $(w,\, l)$.

Une fois l'apprentissage effectué, le système est soumis à un flux de vecteurs non étiqueté ; sa tâche est de replacer les étiquettes appropriées lorsqu'il détecte une sous-séquence apprise. Ceci doit être réalisé en streaming, ce qui se traduit par l'obligation d'étiqueter un vecteur dès qu'il a été émis, sans attendre de connaître la suite du flux. Cette contrainte est indispensable si l'on veut une détection temps réel.

Le système de reconnaissance est basé sur un principe simple : une étiquette est émise si la distance à sa classe est suffisament faible. Pour une sous-séquence w donnée, nous mettons à jour une fonction de dissimilarité (nous parlerons de distance par abus de langage) d_w qui est capable d'assigner à chaque instant un réel positif $d_w(t)$, signifiant dans quelle mesure le flux entrant est similaire à l'exemple w.

2. Il s'agit d'un problème plus complexe que la classification, puisque l'on travaille en flux, et que les vecteurs du flux ne sont pas indépendants les uns des autres.

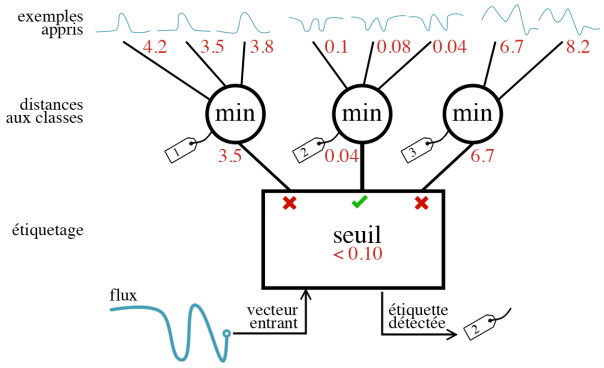

Figure 2 – Architecture de l'étage de détection sur un exemple de flux ; $d = 1$ par simplicité. Un vecteur entrant est obtenu et permet de mettre à jour la distance à chaque exemple, puis la distance aux classes. Une étiquette est délenchée si l'une des classes passe sous un seuil fixé.

Ainsi, tous les exemples sont regroupés par classe. Etant donné le nouveau vecteur entrant à un instant t, il suffit de mettre à jour les distances $d_w(t)$ pour chacun des exemples w. Si l'une des distances passe en dessous d'un certain seuil, c'est que l'utilisateur est en train d'effectuer un geste très similaire à un exemple déjà appris ; l'étiquette correspondante est alors renvoyée.

Comparaison de séries temporelles

Notre distance d_w mesurant la similitude entre un échantillon (une série temporelle) et le flux entrant est une version modifiée de DTW (Dynamic Time Warping [5]), conçue pour fonctionner en multidimensionnel [4], sur des séries temporelles de différentes longueurs, et surtout, en flux.

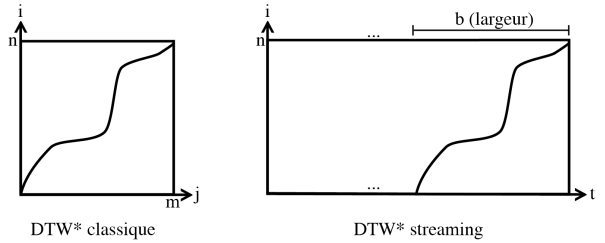

Figure 3 – DTW est modifié en une version acceptant des séries de longueurs différentes et adaptée pour une utilisation en flux.

Type de gestes envisageables

L'utilisateur est peu limité dans le choix des gestes qu'il désire apprendre au système. En effet, des gestes dynamiques (la main se déplace) comme statiques (la main reste en place) peuvent être appris ; de plus, l'orientation est prise en compte, ce qui donne la possibilité de discriminer par exemple deux gestes similaires mais qui s'effectueraient avec la paume vers le haut pour l'un, vers le bas pour l'autre (voir FASTER et SLOWER, plus bas). Enfin, l'utilisateur peut apprendre un geste très court mais également très long, de l'ordre de plusieurs secondes, ce qui sera pris en compte.

Actuellement, les limitations se font sutout ressentir lorsqu'un geste est inclus dans un autre, ce qui est le cas par exemple entre les gestes "cercle" et "huit".

Résultats

Afin de mettre à l'épreuve notre système, nous avons défini un dictionnaire de 7 gestes adaptés au contrôle d'un robot :

- GO* : main ouverte projetée vers l'avant
- STOP : poing fermé vers le haut
- LEFT : pouce+index en L, pouce vers la gauche
- RIGHT : pouce+index en L, pouce vers la droite
- FORWARD : pouce+index en L, pouce vers le haut
- FASTER* : paume face au ciel, projetée vers le haut
- SLOWER* : paume face au sol, projetée vers le bas

(* : gestes dynamiques)

Ces gestes ont été enregistrés en flux à quatre reprises, par le même opérateur. Le système de reconnaissance est soumis à l'épreuve via une procédure de cross-validation (quatre scénarios du type : 3 entraînement vs. 1 test). Les résultats sont présentés sous forme de courbe ROC à la figure 4.

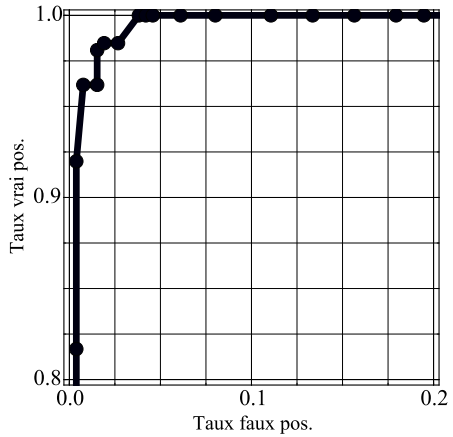

Figure 4 – Courbe ROC représentant la performance du système sur 7 gestes dédiés au contrôle d'un robot.

Les résultats présentés affichent des taux de reconnaissance satisfaisants, et ce sans tomber dans l'excès de faux positifs ou faux négatifs.

Pour l'instant, nous n'avons pas encore poussé le système dans ses retranchements au cours d'un scénario de mobilité forte, notamment où l'utilisateur serait amené à courir longtemps. Des essais informels réalisés en interne suggèrent que le système est déjà relativement robuste à la marche ainsi qu'à une course légère et brève. Toutefois, certains ajustements sur l'étage de prétraitement comme sur l'algorithme de détection seront sans doute nécessaires.

Nous envisageons également de comparer l'algorithme actuel, basé sur DTW, avec d'autres algorithmes de similitude pour séries temporelles.

Références

[1] Baglioni, M., Lecolinet, E., and Guiard, Y. Jerktilts : using accelerometers for eight-choice selection on mobile devices. Dans *Proceedings of the 13th international conference on multimodal interfaces*, ACM (2011), 121–128.

[2] Benbasat, A. Y., and Paradiso, J. A. Compact, configurable inertial gesture recognition. Dans *CHI'01 Extended Abstracts on Human Factors in Computing Systems*, ACM (2001), 183–184.

[3] Benoit, E., Allevard, T., Ukegawa, T., and Sawada, H. Fuzzy sensor for gesture recognition based on motion and shape recognition of hand. Dans *Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2003. VECIMS'03. 2003 IEEE International Symposium on*, IEEE (2003), 63–67.

[4] Gillian, N., Knapp, R. B., and O'Modhrain, S. Recognition of multivariate temporal musical

gestures using N-dimensional dynamic time warping. Dans *Proc of the 11th Int'l conference on New Interfaces for Musical Expression* (2011).

[5] Itakura, F. Minimum prediction residual principle applied to speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on 23*, 1 (1975), 67–72.

[6] Jimenez, A. R., Seco, F., Prieto, C., and Guevara, J. A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. Dans *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*, IEEE (2009), 37–42.

[7] Kamp, J.-F., Ménier, G., and Gibet, S. Une interface gestuelle pour l'apprentissage de la rythmique. Dans *RFIA 2012 (Reconnaissance des Formes et Intelligence Artificielle)* (2012), 978–2.

[8] Kratz, S., and Rohs, M. A $3 gesture recognizer : simple gesture recognition for devices equipped with 3d acceleration sensors. Dans *Proceedings of the 15th international conference on Intelligent user interfaces*, ACM (2010), 341–344.

[9] Kratz, S., and Rohs, M. Protractor3d : a closed-form solution to rotation-invariant 3d gestures. Dans *Proceedings of the 16th international conference on Intelligent user interfaces*, ACM (2011), 371–374.

[10] Madgwick, S. O., Harrison, A. J., and Vaidyanathan, R. Estimation of IMU and MARG orientation using a gradient descent algorithm. Dans *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, IEEE (2011), 1–7.

[11] Ruiz, J., and Li, Y. Doubleflip : a motion gesture delimiter for mobile interaction. Dans *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2011), 2717–2720.

[12] Tidwell, R., Akumalla, S., Karlaputi, S., Akl, R., Kavi, K., and Struble, D. Evaluating the feasibility of EMG and bend sensors for classifying hand gestures. Dans *Proceedings of the International Conference on Multimedia and Human Computer Interaction (63 : 1-8)* (2013).