



Reconnaissance et mémorisation de commandes gestuelles graphiques semi-configurables

Peiyu Li, Eric Anquetil

► **To cite this version:**

Peiyu Li, Eric Anquetil. Reconnaissance et mémorisation de commandes gestuelles graphiques semi-configurables. Colloque International Francophone sur l'Écrit et le Document 2012 (CIFED), 2012, Bordeaux, France. hal-00741311

HAL Id: hal-00741311

<https://hal.inria.fr/hal-00741311>

Submitted on 12 Oct 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mémorisation de commandes gestuelles

Reconnaissance et mémorisation de commandes gestuelles graphiques semi-configurables

PeiYu LI — Eric Anquetil

IRISA — INSA de Rennes

Campus universitaire de Beaulieu

263 Avenue du Général Leclerc – Bât 12, 35042 Rennes cedex

{pei-yu.li, eric.anquetil}@irisa.fr

RÉSUMÉ. Les interfaces tactiles ou orientées stylo ouvrent aujourd'hui des possibilités d'interactions particulièrement intéressantes pour établir une communication Homme-Machine à la fois efficace et intuitive. Dans le contexte d'applications réelles, le jeu de commandes à utiliser peut devenir assez conséquent. L'utilisateur se retrouve alors confronté au problème de la mémorisation des gestes graphiques associés aux commandes. Par ailleurs, chaque utilisateur ayant ses habitudes de tracé, il semble intéressant de lui laisser définir ses propres gestes pour chaque commande. Il en découle, plusieurs questions autour de la définition, de la reconnaissance et de la mémorisation par l'utilisateur de commandes gestuelles graphiques : Peut-on imposer un jeu de commandes à un utilisateur, comme c'est le cas dans quasi tous les systèmes existants aujourd'hui ? Faut-il laisser l'utilisateur choisir ses propres gestes ? Pour répondre à ces questions, nous proposons un nouveau concept de commandes graphiques, les SCGC (Semi-Customizable Gesture Commands). Dans ce papier, nous montrons à travers plusieurs expérimentations la pertinence du concept des SCGC qui s'avère être la façon la plus efficace pour aider l'utilisateur à mémoriser et exploiter un jeu conséquent de commandes graphiques.

ABSTRACT. Pen-based interfaces give us possibility to interact in an especially interesting way to establish an effective and intuitive Computer-Human communication. In real world applications, the commands set can become consequent. The user faces to the learning problem of the gestures associated to commands set. Moreover, each user has his drawing habit, and it seems interesting to offer him the possibilities to define his gestures for each command. To answer this need, we propose a new gesture command's concept: Semi-Customizable Gesture Commands. In this paper, we will show you how this approach works to help the learning of gestures and its efficiency through some experimentation.

MOTS-CLÉS : commandes gestuelles graphiques, interaction orientée stylo, mémorisation.

KEYWORDS : gesture commands, pen-based interaction, learning.

1. Introduction

Les interfaces orientées stylo permettent aux utilisateurs de dessiner des tracés pour exécuter des commandes. Ces commandes gestuelles graphiques offrent une communication Homme-Machine à la fois efficace et intuitive, qui est de plus en plus recherchée de nos jours. Par exemple, on peut associer un geste de la forme « C » à la commande « Copier » et un geste de la forme « V » à la commande « Coller ». Dans le contexte d'applications réelles, le jeu de commandes à utiliser peut devenir conséquent. Aujourd'hui, les moteurs de reconnaissance de gestes graphiques deviennent de plus en plus performants ; certains permettent même de s'adapter et d'apprendre incrémentalement le style de l'utilisateur (Almaksour, 2011). Le problème apparaît alors plutôt du côté de l'utilisateur qui se trouve confronté à la difficulté de la mémorisation des gestes graphiques associés aux commandes. Il devient en effet difficile pour l'utilisateur de mémoriser un grand ensemble de gestes. Même si l'on essaie d'associer, quand cela est possible, des gestes « intuitifs » ou « signifiants » aux commandes, ou encore si on laisse la possibilité aux utilisateurs de définir leurs propres gestes, il s'avère fastidieux à l'utilisateur d'apprendre et de mémoriser un jeu de gestes graphiques trop conséquent. Pour pallier ce problème, Kurtenbach a proposé le concept des « Marking Menus » (Kurtenbach, 1993). L'idée est d'induire un apprentissage des commandes graphiques qui soit implicite à l'utilisateur via l'exploitation de menus dont l'ergonomie sous-tend les tracés graphiques des commandes. Deux modes de fonctionnement sont associés aux « Marking Menus » : le mode novice où l'utilisateur sera guidé avec les menus pour exécuter une commande et le mode expert où l'utilisateur a appris le geste final associé à une commande ; il effectue donc ce geste qui est interprété automatiquement par un reconnaiseur de gestes graphiques. La transition entre les deux modes peut être asservie par la vitesse d'exécution du geste.

A partir du concept des « Marking Menus », différentes variantes ont été proposées (Bau et al., 2008 ; Kurtenbach et al., 1993 ; Zhao et al., 2004 ; Zhao et al., 2006). Les « Hierarchical Marking Menus » permettent d'avoir plus d'un niveau sur les menus (Kurtenbach et al., 1993). On peut citer encore d'autres approches basées sur ce concept d'aide à l'apprentissage de gestes avec des menus : les « Zone and Polygon Menus » (Zhao et al., 2004), les « Flower Menus » (Bailly et al., 2007), etc. Toujours dans la même philosophie, Bau et al. ont proposé un concept voisin dénommé « Octopocus » (Bau et al., 2008). « Octopocus » ne travaille qu'avec un « Marking Menus » d'un niveau mais il propose un feedback continu pour guider l'utilisateur dans la réalisation de son geste en mode novice qui est vraiment très intéressant. L'autre grand avantage d'« Octopocus » est qu'il permet de s'appuyer sur des gestes graphiques qui peuvent être de formes très variés, contrairement aux autres approches à base de « Marking Menus » qui ne permettent d'engendrer qu'un jeu de gestes dont les formes graphiques se limitent souvent à des lignes polygonales qui par nature n'induisent pas un geste très fluide. Par contre, « Octopocus » ne permet pas aux utilisateurs de personnaliser leurs gestes, et il n'y a

Mémorisation de commandes gestuelles

aucune logique implicite ou explicite induite par les tracés des gestes par rapport à leur commande.

Dans nos travaux, pour faire face à ce problème d'aide à la mémorisation des gestes graphiques nous avons cherché à concevoir une approche qui puisse répondre à plusieurs propriétés :

- avoir une grande liberté d'expressivité dans le tracé des gestes qui doivent rester le plus fluide possible ;
- permettre à l'utilisateur de choisir certains gestes (les plus fréquents par exemple) ou certaine portion d'un geste ;
- pouvoir induire dans le tracé des gestes une sémantique ou une organisation hiérarchique qui va aider la mémorisation ;
- intégrer le concept des « Marking Menus » pour faciliter l'apprentissage des gestes par l'utilisateur.

Nous avons proposé dans un premier temps une approche pour tendre vers ces objectifs à travers le nouveau concept des Continuous Marking Menus (CMM) (Delaye et al., 2011 ; Li et al., 2011). Le retour des utilisateurs suite aux expérimentations que nous avons effectuées sur les CMM, nous ont convaincu de la nécessité de permettre un degré de personnalisation dans le jeu de gestes et de réduire au maximum les contraintes sur la forme des gestes possibles. C'est pourquoi nous proposons dans ce papier une nouvelle typologie de commandes graphiques hiérarchiques et semi-personnalisables : les « Semi-Customizable Gestural Commands (SCGC) ». Dans ce papier, après un bref rappel du concept des CMM et des principaux résultats obtenus, nous décrirons en détail le nouveau concept des SCGC. L'idée originale des SCGC est d'une part de considérer la fréquence d'utilisation des commandes pour faciliter la mémorisation de l'ensemble des commandes et d'autre part, de laisser la liberté aux utilisateurs de définir eux-mêmes les gestes associés à ces commandes prédominantes. Les gestes graphiques associés aux autres commandes seront alors générés automatiquement sur la base des gestes prédominants en fonction de leur catégorisation hiérarchique. Le but principal dans cette nouvelle approche est d'obtenir des gestes naturels, cursifs et faciles à retenir. Cette approche a été testée sur un panel d'utilisateurs afin de mesurer l'impact de l'exploitation des SCGC sur la courbe d'apprentissage d'un jeu de gestes graphiques par des utilisateurs. Autrement dit, nous cherchons à mesurer l'aide à la mémorisation des gestes graphiques apportée par les SCGC. Pour cela, nous nous sommes appuyés sur un protocole expérimental utilisé dans la communauté des usages et des IHM. L'ensemble de ces tests sont détaillés dans la dernière partie du papier.

2. « Continuous Marking Menus (CMM) »

2.1. Principes

Les CMM sont des « Marking Menus » hiérarchiques qui autorisent plusieurs niveaux de menus. Comme tous les Marking Menus, les CMM ont deux modes d'utilisation : le mode novice et le mode expert. En mode novice, l'utilisateur s'appuie sur les CMM pour se familiariser avec les gestes. A chaque niveau, les items de CMM s'ouvrent comme des pétales qui sont centrés sur le stylo. Pour mieux interagir avec l'utilisateur, chaque pétale est partitionnée en deux zones d'interaction : la zone de présélection et la zone d'inertie. Quand le stylo entre dans la zone de présélection d'un item, l'utilisateur voit par feedback continu l'item choisi qui devient de plus en plus foncé et alors que les autres items deviennent en parallèle de plus en plus clairs. Dans la zone d'inertie, le centre de sous-menu suit le mouvement de stylo. Quand l'utilisateur choisit une branche de sous-menu et quitte la zone d'inertie, un nouveau cycle commence. La Figure 1 illustre ce principe.

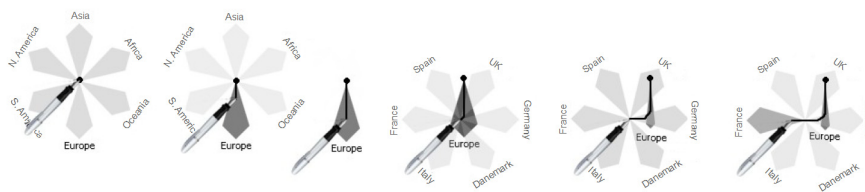


Figure 1. Etapes successives pour la sélection des menus.

En mode expert, les CMM utilisent un classifieur à base de distance élastique (Dynamic time warping (DTW)) pour reconnaître les gestes (Vuori, 2002 ; Niels, 2004). C'est un reconnaiseur relativement basique mais suffisant pour notre problème. Plus précisément, pour calculer la distance entre deux points, nous n'utilisons non seulement la distance euclidienne, mais nous considérons aussi la vitesse et l'accélération à ces points. De plus, on introduit une « pondération » entre ces différents aspects pour améliorer le taux de reconnaissance. Avec les tests effectués, on a obtenu un taux de reconnaissance à 99,2 %, ce qui est assez pertinent pour reconnaître les gestes. Cette approche est aussi utilisée par l'approche SCGC présentée plus loin dans le papier.

2.2. Expérimentations sur les CMM

Pour mesurer si l'approche CMM aide les utilisateurs à mieux retenir les gestes, nous avons conduit des expérimentations sur un ensemble de 12 personnes. Les 12 personnes ont été séparées en deux groupes. Le groupe A passait le premier test

Mémorisation de commandes gestuelles

(test1) sans utiliser les CMM et le groupe B passait le test1 en utilisant les CMM. Pour le second test (test2), nous échangeons pour les deux groupes leur condition, c'est à dire, le groupe B passait le test2 sans menu alors que le groupe A passait le test2 avec les CMM. Dans les tests servant de référence, c'est-à-dire sans utilisation des CMM, la liste des gestes associés aux commandes était affichée de manière standard, sous la forme d'un tableau d'aide récapitulant les commandes et les gestes associés.

Les résultats de ces tests sont présentés dans la Figure 2. On observe que l'utilisation des CMM améliore significativement l'apprentissage par les utilisateurs des commandes.

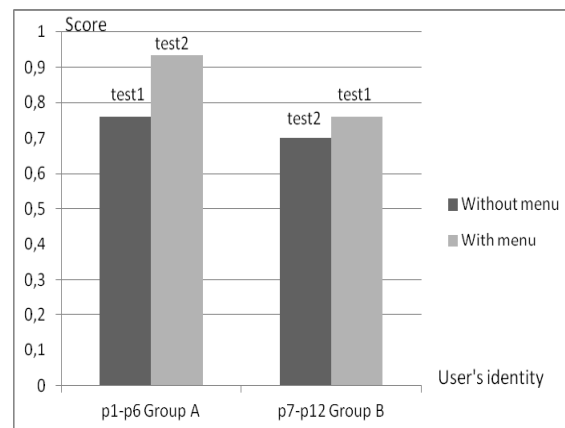


Figure 2. Taux de reconnaissance pendant les modes experts, avec ou sans l'aide des CMM.

3. Approche SCGC

Les tests effectués sur les CMM nous ont fait prendre conscience que ces menus introduisent trop de contraintes sur la forme des gestes. De plus, beaucoup d'utilisateurs ont indiqué leur souhait de pouvoir personnaliser certains gestes (notamment ceux associés à des commandes fréquentes) pour faciliter la mémorisation. En effet, dans un cadre applicatif réel, toutes les commandes gestuelles n'ont pas la même fréquence d'utilisation. Par exemple, on utilise plus souvent la commande « copier » que la commande « imprimer ». En général, l'utilisateur aimerait mémoriser en premier les commandes les plus souvent utilisées. Plus on utilise une commande, plus on l'apprend vite.

Une première idée directrice des SCGC est d'utiliser ces commandes que l'on qualifie de *prédominantes* comme un « point d'ancrage » pour mémoriser les autres commandes associées à la même famille par exemple. Plus précisément cette approche originale est basée sur les principes suivants :

- définir une structuration « sémantique » a priori du jeu de commandes en familles ;
- laisser l'utilisateur dessiner la commande prédominante de chaque famille ;
- synthétiser automatiquement toutes les variantes graphiques des autres commandes au sein d'une même famille, à partir de la commande prédominante définie par l'utilisateur.

Pour implémenter les SCGC et notamment gérer la génération automatique des alternatives aux commandes prédominantes, nous avons modélisé le tracé d'une commande graphique par une Spline. La génération des variantes d'une commande prédominante est obtenue par la déformation continue de la fin du tracé du prototype de la commande prédominante. Nous détaillons dans les paragraphes suivants les différentes étapes opérées pour implémenter les SCGC.

3.1. Première étape : approximation polygonale

La première étape consiste en une approximation polygonale du tracé de la commande gestuelle. On utilise un algorithme décrit par Perez (Perez et al., 1994). Ayant une collection de N points et une constante M, cet algorithme peut trouver les M points qui forme un polygone qui suit le mieux la courbe d'origine définie par les N points. La qualité de l'approximation polygonale est indiquée par un indice d'erreur. On peut trouver l'approximation polygonale optimale (défini par la valeur de M) selon cet indice. Dans notre cas, la valeur maximum de M est fixée à six. Par exemple, dans Figure 3 la courbe noire est le tracé original. L'étape 1 correspond à l'approximation polygonale. Les points bleus apparus sur le deuxième tracé sont les coins extraits par l'algorithme. Ici on a M=6.

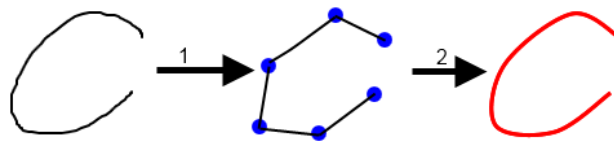


Figure 3. Création du prototype par approximation polygonale puis Spline.

3.2. Deuxième étape : synthèse par spline

Les M points extraits par l'approximation polygonale sont ensuite utilisés comme points de contrôle pour définir une spline canonique qui va approximer le tracé original.

Une spline canonique est une courbe polynomiale qui a pour but d'imiter la courbe originale en utilisant quelques points de contrôle pour régulariser le tracé. A la différence de courbes de Bézier, la spline canonique passe par tous ses points de contrôle. Dans notre cas, l'algorithme utilisé pour construire la spline canonique est celui de Petzold (Petzold, 2009). Dans la figure précédente, l'étape 2 correspond à la génération de la spline canonique à partir des points obtenus par l'approximation polygonale.

3.3. Troisième étape : génération des variantes de gestes pour les autres commandes de la même famille

Notons les M points de contrôle définis à l'étape précédente par : p_1, p_2, \dots, p_M . Pour les variantes de gestes de la même famille, nous gardons le même début du tracé, ie. p_1, p_2, \dots, p_{M-1} . Nous déplaçons simplement le dernier point de contrôle pour générer des nouveaux gestes. Grâce aux propriétés de la spline canonique, les nouvelles courbes générées vont rester fluides et cursives.

La Figure 4 illustre les principes de génération automatique des variantes de commandes de la même famille. Le tracé noir représente la spline canonique que l'on obtient à partir du tracé original de l'utilisateur. Pour une famille qui regroupe trois commandes, nous récupérons les coordonnées des deux derniers points de contrôle : p_{M-1} et p_M . Nous générons alors les coordonnées de point p_M' (nouveau dernier point de contrôle pour la première variante du geste) et p_M'' (nouveau dernier point de contrôle pour la seconde variante du geste), qui vont respecter les règles suivantes :

$$|p_{M-1} p_M| = |p_{M-1} p_M'| = |p_{M-1} p_M''| \text{ et}$$

$$\angle p_M p_{M-1} p_M' = \angle p_M p_{M-1} p_M'' = \angle p_M'' p_{M-1} p_M'.$$

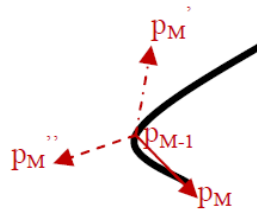


Figure 4. Génération des variantes pour la fin du tracé (p_M' et p_M'').

Les collections des points de contrôle pour les deux variantes du geste de la famille sont donc $(p_1, p_2, \dots, p_{M-1}, p_M')$ et $(p_1, p_2, \dots, p_{M-1}, p_M'')$ respectivement.

La Figure 5 illustre sur un exemple concret la génération de deux variantes de geste à partir du geste d'une commande prédominante : en étape 1, après avoir récupéré la courbe originale de l'utilisateur, une spline canonique est générée avec $M=5$. En étape 2, les deux variantes associées aux deux autres commandes gestuelles de la même famille sont générées automatiquement.

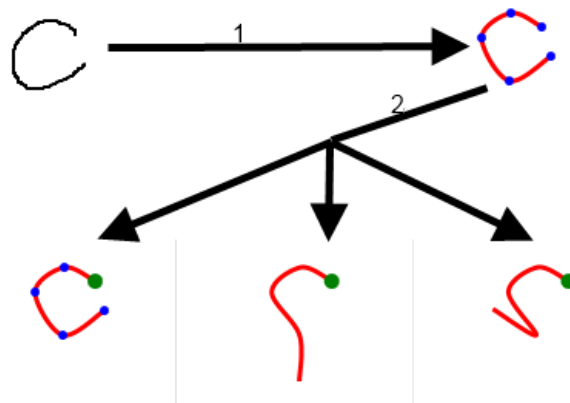


Figure 5. *Illustration de génération des prototypes pour les autres membres de la même famille.*

4. Expérimentations

4.1. Sujets et matériels

Pour évaluer l'impact sur la courbe d'apprentissage des utilisateurs du nouveau concept de commandes gestuelles SCGC, nous avons défini trois conditions de tests : le test 1 où tous les gestes sont prédéfinis, le test 2 où l'utilisateur peut choisir lui-même l'ensemble de ses gestes, en enfin le test 3 où l'on met en œuvre les SCGC, c'est-à-dire où l'utilisateur choisit le geste d'une commande prédominante par famille et toutes les variantes sont générés automatiquement. La quantification du niveau d'apprentissage de l'utilisateur est évaluée en divisant chaque test en 5 périodes successives, chacune constituée d'une phase d'apprentissage (le tableau d'aide est disponible) est d'une phase de test (l'aide n'est plus disponible).

Chaque utilisateur réalisera les trois tests. Les tests ont été faits dans le même ordre pour tous les utilisateurs. Même si il n'y a pas eu de contre balancement, pour éviter l'effet d'ordre et réduire l'influence mutuelle entre les tests, pour chaque

Mémorisation de commandes gestuelles

utilisateur les trois tests ont été espacés d'au moins 2 jours. Ceci garantit que l'utilisateur ait eu le temps d'oublier le jeu de gestes d'un test à l'autre.

Ces trois tests étaient présentés aux utilisateurs comme un jeu de mémoire. Pour chaque test, l'utilisateur doit mémoriser les gestes associés à 15 commandes différentes réparties dans 6 familles de geste. Par construction, les gestes exploités pour ces trois tests sont différents. Les commandes sont : copier, coller, couper (famille « éditer ») ; insérer une image, un tableau, du texte (famille « insérer ») ; sauvegarder au format .pdf, .doc, .docx (famille « sauvegarder ») ; afficher la barre d'outil, la règle, les commentaires (famille « afficher ») ; pivoter 90 degré, 270 degré (famille « pivoter ») et aide (famille « aide »).

4.2. Procédure

4.2.1. Quantification du niveau d'apprentissage de l'utilisateur

La quantification du niveau d'apprentissage de l'utilisateur pour chacun des tests s'appuie sur le séquençement de 5 périodes, chacune constituée d'une phase d'apprentissage appelée mode novice (le tableau d'aide est disponible) est d'une phase de test appelée mode expert (l'aide n'est plus disponible).

Chaque test est donc découpé en 5 périodes (phase novice → phase expert) :

- Période 1 : novice→expert
- Période 2 : novice→expert
- Période 3 : novice→expert
- Période 4 : novice→expert
- Période 5 : novice→ expert.

Pour chaque test, en mode novice, on demande à l'utilisateur de dessiner le geste d'une commande trois fois. Il peut dans ce mode consulter l'aide quand il le souhaite. En mode expert l'aide est cachée. On demande alors à l'utilisateur de réaliser une fois chaque commande dans un ordre aléatoire. A la fin du test, chaque utilisateur obtient cinq scores au total, un à l'issue de chaque période. Ces scores sont reportés dans les courbes d'apprentissages de la Tableau 1. Soulignons que chaque test dure environ 45 minutes par personne.

4.2.2. Test1 : test où tous les gestes sont prédéfinis

La Figure 6 présente l'interface en mode novice pour le test1. Elle est séparée en trois parties. L'utilisateur a le tableau d'aide en zone 1 de l'interface où il peut trouver les gestes de toutes les commandes prédéfinies regroupées par famille. Dans la zone 2, un champ texte en haut à gauche lui indique le label de la commande à dessiner. Il doit alors réaliser la commande dans la zone de saisie (zone 2). Quand

un geste est saisi, il est envoyé au reconnaisseur de gestes qui est basé sur un DTW. Le résultat de la reconnaissance est affiché sous le champ texte pour l'informer si son geste est bien reconnu. Dans la zone 3, le tracé du geste reconnu est affiché.

Dans le mode expert, seule la zone 2 est affichée. On demande à l'utilisateur de réaliser le geste d'une commande, son geste est envoyé au reconnaisseur de geste. On lui donne alors un feedback sur le résultat de la reconnaissance.

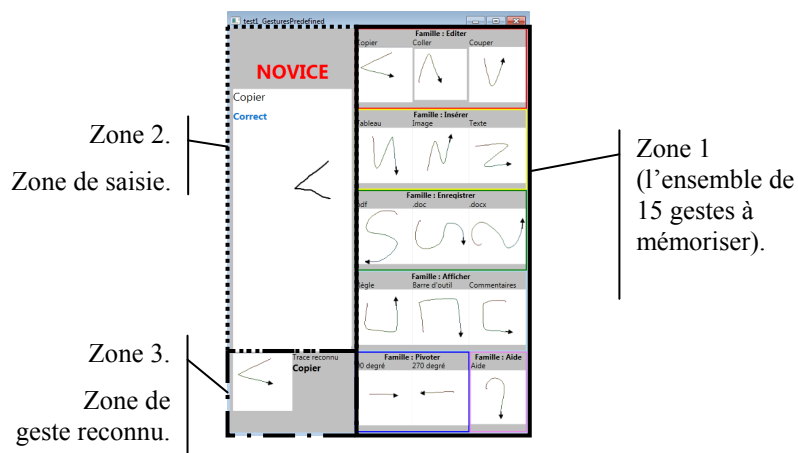


Figure 6. Interface du mode novice pour le test1 où tous les gestes sont prédéfinis.

4.2.3. Test2 : test où tous les gestes sont définis par l'utilisateur

L'interface définie pour le test 2 ressemble à celle du test 1. Pendant le test 2, l'utilisateur définit ses gestes pour chaque commande, on rend active la zone 1 qui permet alors à l'utilisateur de dessiner directement son geste dans la case correspondante. Une fois toutes les cases remplies et validées, l'ensemble des prototypes de référence est généré et sera utilisé ensuite par le reconnaisseur de geste. La suite du protocole est identique en tout point au test 1 (voir la Figure 7).

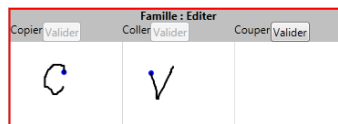


Figure 7. Une partie d'interface pour test2 où l'utilisateur choisit tous ses gestes.

4.2.4. Test3 : mise en œuvre des SCGC

Pour le test 3, l'utilisateur dessine selon son choix un geste (une case) par famille. Une fois qu'il a « validé » la case correspondante, le prototype est sauvegardé et les autres gestes de la famille sont construits automatiquement comme décrit précédemment. La suite du protocole est alors identique aux tests 1 et 2 (voir la Figure 8).

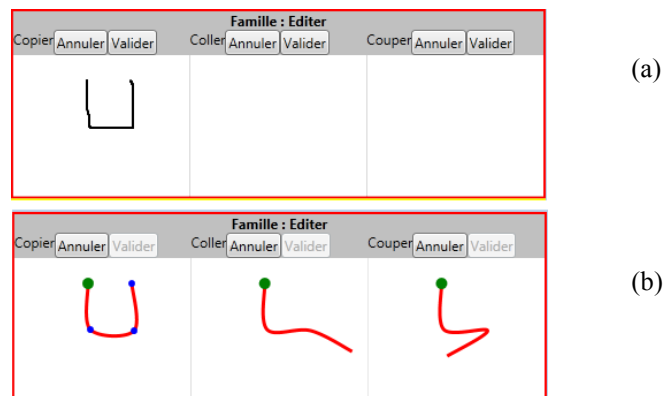


Figure 8. Une partie d'interface pour test3.

(a), l'utilisateur vient de réaliser son geste pour « copier » qui est la commande prédominante pour lui. (b), une fois le tracé validé, la spline canonique de « copier » est générée et les gestes pour les autres commandes de la famille « éditer », c'est-à-dire, « coller » et « couper », sont aussi générés automatiquement.

4.3. Résultats d'expérimentations

Les résultats de ces tests sont présentés en détail dans le Tableau 1 et synthétisés dans la Figure 9 qui donne la courbe d'apprentissage moyenne pour les neuf personnes pour chacun des trois tests.

Nous observons que tous les utilisateurs apprennent mieux et plus vite avec le test 3 (SCGC) qu'avec le test 1 (gestes prédéfinis). Sur les 9 utilisateurs, 8 d'entre eux obtiennent un bien meilleur résultat avec les commandes gestuelles graphiques semi-configurables (test3) qu'en définissant eux même la forme de tous les gestes (test2).

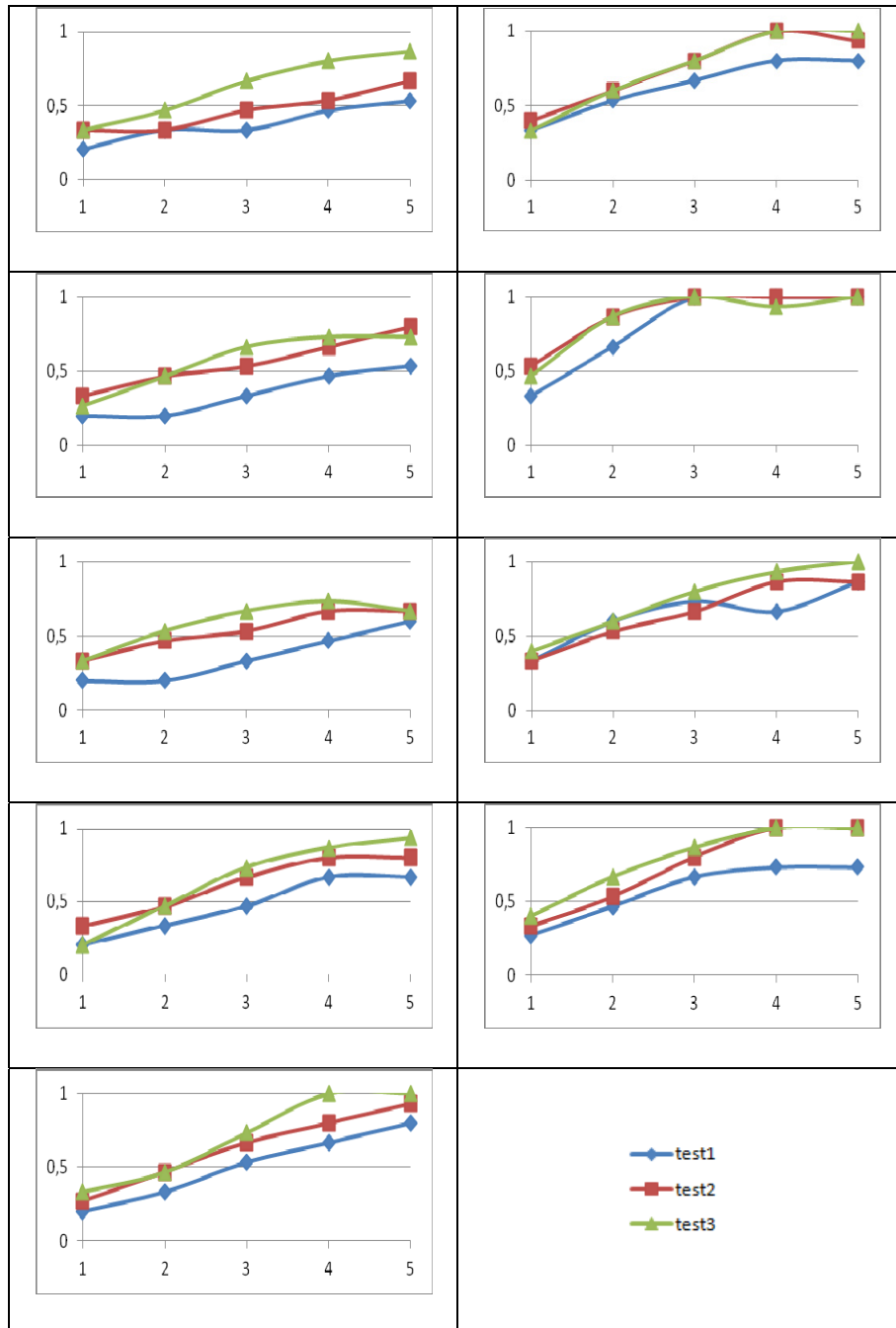


Tableau 1. Résultats des tests pour les neuf participants. L'axe x est la numérotation des cinq étapes; L'axe y est le taux d'apprentissage.

Mémorisation de commandes gestuelles

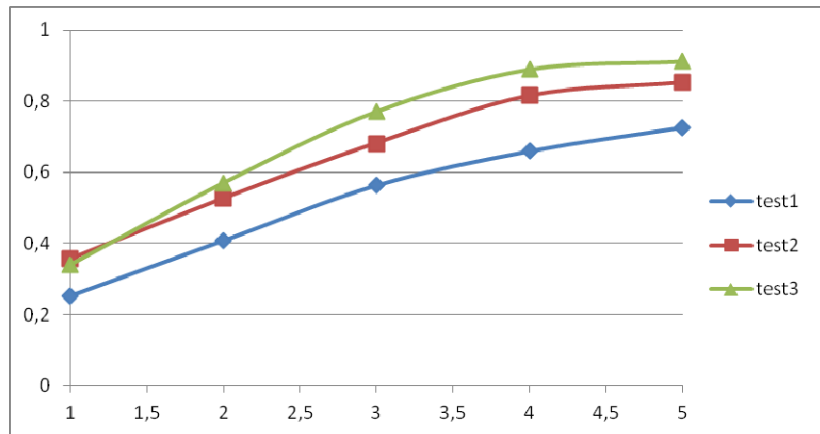


Figure 9. Courbe d'apprentissage en moyenne.

Pour approfondir ces résultats, nous avons opéré une évaluation de la significativité de ces résultats grâce au test ANOVA. Cette analyse est décrite dans le paragraphe suivant.

4.4. Test de significativité sur les résultats d'expérimentations

En analysant simplement les résultats que l'on a obtenus, on observe que les commandes gestuelles graphiques semi-configurables (SCGC) aide les utilisateurs à mémoriser les gestes. Pour quantifier plus finement ce résultat nous avons opéré une analyse de la variance via le test ANOVA.

Le test ANOVA est une analyse de la variance sur des données venant d'un même contexte. Ce test est souvent utilisé en psychologie expérimentale et en Interaction Homme Machine. Il permet de quantifier la significativité de notre hypothèse de travail : le test3 (gestes semi-configurables SCGC) est meilleur que le test2 (gestes choisis) et que le test1 (gestes prédéfinis).

Dans notre cas, nous avons deux facteurs de variabilité, c'est-à-dire deux variables indépendantes pour analyser tous les scores obtenus. Le facteur 1 est le numéro des différentes périodes : $i=1..5$. Le facteur 2 est le numéro du test effectué : $j=1..3$.

En utilisant l'analyse de variance à deux facteurs, on trouve un effet global très significatif :

- sur les périodes : $(F(4,32)=122, p<0.001, \text{partial } \eta^2=0.94)$
- et sur les tests : $(F(2,16)=39.5, p<0.001, \text{partial } \eta^2=0.83)$.

On a aussi trouvé une interaction significative des deux facteurs :

- Période*Test : ($F(8,64)=2.5$, $p=0.02$, $\text{partial } \eta^2=0.24$),

Ceci traduit que l'écart entre les scores obtenus dépend de la période considérée. Plus on avance dans l'apprentissage et plus les écarts sont significatifs. De plus, si l'on réalise une comparaison partielle (test1 versus test2) et (test2 versus Test3) on montre que globalement le test2 est meilleur que test1 et que le test3 est meilleur que le test2, en considérant toutes les données des périodes confondues.

5. Conclusion et perspectives

Dans ce papier, contrairement aux nombreux travaux de la communauté de l'écrit et du document qui portent sur l'évaluation des performances des systèmes de reconnaissance de tracés manuscrits, nous nous sommes intéressés au problème de la mémorisation et donc de l'apprentissage des gestes graphiques pour un utilisateur. En effet, l'utilisateur se retrouve souvent confronté au problème de la mémorisation des gestes graphiques associés aux commandes. Nous avons proposé une nouvelle famille de commandes gestuelles graphiques qui sont semi-configurables (SCGC). Les SCGC permettent de définir une structuration « sémantique » a priori d'un jeu de commandes en familles, tout en laissant le choix à l'utilisateur de définir ses repères de mémorisation par la définition du geste de la commande prédominante de chaque famille. Les premiers résultats expérimentaux sur la mémorisation des gestes de commandes reportés dans ce papier montrent que les SCGC sont plus efficaces que de laisser l'utilisateur définir tout seul tous ses gestes de commande ou encore que de lui imposer des gestes prédéfinis.

Dans nos prochains travaux, nous nous attacherons à fusionner le concept des « Marking Menus » explorés avec les CMM avec les commandes gestuelles semi-configurables (SCGC). L'objectif est d'améliorer encore le facteur d'apprentissage de l'utilisateur tout en augmentant l'ergonomie de l'exploitation des commandes gestuelles en mode novice pour éviter d'avoir besoin de consulter un tableau d'aide des commandes existantes.

6. Bibliographie

- A. Almaksour, Incremental Learning Of Evolving Fuzzy Inference Systems: Application To Handwritten Gesture Recognition, PhD Thesis, INSA de Rennes, (2011).
- Bailly, B., Lecolinet, E., Nigay, L. (2007). « Wave Menus: Improving the Novice Mode of Hierarchical Marking Menus », *INTERACT'07*. Springer. p. 475--488.
- Bau, O. and Mackay, W.E. « Octopocus: A Dynamique Guide for Learning Gesture-based Command Sets », *proc. UIST 2008*, ACM Press (2008), p. 37-46.

Mémorisation de commandes gestuelles

- Delays, A., Sekkal, R., and Anquetil, E. « Continuous Marking Menus for learning Cursive Pen-based Gestures », *proc. IUI 2011*, ACM Press(2011), p. 319-322.
- Kurtenbach, G. The design and evaluation of marking menus. PHDThesis (1993).
- Kurtenbach, G. and Buxton, W. « The limits of expert performance using hierarchic marking menus », *proc. INTERCHI (1993)*, IOS Press (1993), p. 482-487.
- Li, P., Delays, A. and Anquetil, E. « Evaluation of Continuous Marking Menus for Learning Cursive Pen-based Commands », *proc. IGS 2011*, p. 217-220.
- Niels, R. Dynamique Time Warping (An intuitive way of handwriting recognition). Master Thesis (2004).
- Perez, J. and Vidal, E. « Optimum polygonal approximation of digitized curves », *Pattern Recognition Letters 15* (1994), p. 743-750.
- Petzold, C. Canonical Splines in WPF and Silverlight.
<http://www.charlespetzold.com/blog/2009/01/Canonical-Splines-in-WPF-and-Silverlight.html>.
- Shier, R. Paired t-tests. <http://mlsc.lboro.ac.uk/resources/statistics/Pairedttest.pdf>.
- Vuori, V. Adaptive Methods for On-Line Recognition of Isolated Handwritten Characters. PHDThesis (2002).
- Zhao, S., Agrawala, M. and Hinckley, K. « Zone and Polygon Menus: Using Relative Position to Increase the Breadth of Multi-stroke Marking Menus », *proc. CHI 2006*, ACM Press (2006), p. 1077-1086.
- Zhao, S. and Balakrishnan, R. « Simple vs. Compound Mark Hierarchical Marking Menus », *proc. UIST 2004*, ACM Press (2004), p. 33-42.