

# SPEED : Prédiction de cibles

Jonathan Wonner  
ENS Cachan-Bretagne  
Campus de Ker Lann  
35170, Bruz, France  
jonathan.wonner@ens-  
cachan.org

Jérôme Grosjean  
LSIIT UMR 7005  
Bd Sébastien Brant  
67412, Illkirch, France  
grosjean@unistra.fr

Antonio Capobianco  
LSIIT UMR 7005  
Bd Sébastien Brant  
67412, Illkirch, France  
a.capobianco@unistra.fr

Dominique Bechmann  
LSIIT UMR 7005  
Bd Sébastien Brant  
67412, Illkirch, France  
bechmann@unistra.fr

## RESUME

Nous présentons la méthode de prédiction de cibles SPEED, basée sur l'analyse des caractéristiques cinétiques du geste de pointage. Notre modèle repose sur une séparation du geste de sélection en une phase d'accélération et une phase de décélération, afin d'estimer précisément la cible visée. L'analyse de la première phase nous permet d'identifier un pic de vitesse qui marque le début de la seconde phase. Cette dernière est alors approchée par un modèle quadratique afin de prédire le point d'arrivée du geste. Une première évaluation expérimentale montre que SPEED permet de prédire une cible plus précisément que les autres méthodes existantes, dans le cadre de sélections 1D sans distracteurs.

## MOTS CLES

Prédiction de cible, geste de sélection, loi de Fitts, cinématique

## ABSTRACT

We present the SPEED method to predict endpoints, based on analysis of the kinetic characteristics of the pointing gesture. Our model splits the gesture into an acceleration phase and a deceleration phase to precisely detect target. The first phase allows us to identify a velocity peak that marks the beginning of the second phase. This phase is approached with a quadratic model to predict gesture endpoint. A pilot study shows that SPEED predicts a target more precisely than other existing methods, for 1D tasks without distractors.

## Categories and Subject Descriptors

H.5.2 [Information interfaces and presentation]: User Interfaces—*Interaction styles, Input devices and strategies*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IHM'11, October 24-27, 2011, Sophia Antipolis, France

Copyright © 2011 ACM 978-1-4503-0822-9/11/10 ...\$10.00.

## General Terms

Algorithms, Human factors

## Keywords

Target prediction, selection gesture, Fitt's law, kinematics

## 1. INTRODUCTION

Dans une interface graphique, il est souvent nécessaire de sélectionner une icône, une option dans un menu, une cellule dans une feuille de calcul... Plusieurs techniques ont été proposées pour faciliter ces sélections, et ainsi améliorer les performances de l'utilisateur. La loi de Fitts [8] suggère que cette amélioration peut s'effectuer par le biais d'une réduction de l'indice de difficulté  $ID$  de la sélection. Dans le cas d'une tâche 1D, cet indice est donné par la formule :

$$ID = \log_2 \left( \frac{A}{W} + 1 \right), \quad (1)$$

où  $W$  est la largeur de la cible, et  $A$  la distance entre la cible et l'utilisateur. Cette formulation a ensuite été étendue pour des tâches 2D et 3D [10, 16, 13, 1, 2].

En s'appuyant sur ce modèle, des solutions ont été proposées pour augmenter les performances des utilisateurs. La méthode du drag-and-pop (Baudisch et al. [4]) fait apparaître des copies des icônes lointaines près du curseur, afin de réduire la distance entre le curseur et les cibles potentielles. McGuffin et al. [17] agrandit la taille des icônes se trouvant à proximité du curseur, facilitant leur sélection. Guiard et al. [11] utilise un curseur qui se déplace de cible en cible sans passer par le vide entre elles. Cependant, l'efficacité de ces aides est fortement liée à la distance et à la densité des cibles. Pour McGuffin et al. [17], il est nécessaire de prédire la cible de l'utilisateur pour proposer une aide efficace.

Plusieurs travaux ont ainsi étudié des algorithmes de prédiction de cibles [3, 15, 17, 18, 19]. Ces algorithmes reposent principalement sur la trajectoire et la vitesse du curseur pour prédire la cible de l'utilisateur au cours de son mouvement. Nous proposons ici un nouvel algorithme de prédiction, SPEED (*Speed Profile sEparation for Endpoint Divination*), basé sur une modélisation du profil de vitesse du curseur lors d'un mouvement de sélection. Détecter le pic de vitesse dans le profil de vitesse du geste permet de ne considérer que

la phase de décélération, pour estimer au mieux la cible de l'utilisateur.

Nous présentons en section *État de l'art* deux techniques de prédiction basées sur la vitesse de l'utilisateur. Nous décrivons ensuite l'algorithme SPEED, qui est alors évalué dans une expérience détaillée en section *Expérimentation*. Cette expérience représente le cas simple d'une tâche de sélection monodirectionnelle sans distracteurs.

## 2. ÉTAT DE L'ART

De nombreuses recherches ont été menées afin de proposer des techniques améliorant le temps de sélection d'un élément dans une interface graphique. Un grand nombre de ces travaux note une perte de performances lorsque la densité des cibles augmente [6, 11]. Pour proposer une aide efficace, il semble alors nécessaire d'identifier la cible de l'utilisateur, ou au moins son voisinage, en prédisant le point d'arrivée de son mouvement [17].

Cependant, assez peu d'études ont été menées sur cette prédiction de point d'arrivée. Asano et al. [3] a observé que lors d'un geste de sélection, l'amplitude du pic de vitesse dépend linéairement de la distance de la cible. L'algorithme consiste alors à détecter un pic de vitesse, et en déduire la distance à laquelle se trouve la cible. La direction courante du mouvement permet ensuite d'identifier la position de la cible. Le curseur saute alors vers cette position. Dans le cas de cibles lointaines, cette méthode permet de sélectionner sa cible plus rapidement. Néanmoins, une perte de performance est observée pour des cibles proches ou denses.

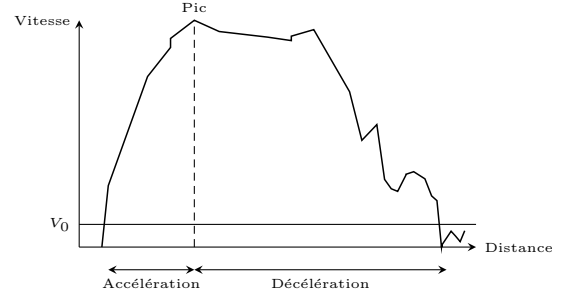
Lank et al. [15] propose de se baser sur le profil de vitesse en entier. À chaque instant, la méthode extrapole une courbe quadratique du profil de vitesse, et en déduit la distance de fin de mouvement. Ruiz et al. [18] donne une version plus robuste de cet algorithme, KEP (*Kinematic Endpoint Prediction*), dans laquelle la prédiction doit se stabiliser avant d'être validée. KEP est actuellement la méthode de prédiction la plus précise, obtenant un taux de réussite moyen de 51.0%, pour un calcul de prédiction réalisé à 85% du mouvement. Dans un cadre applicatif, les auteurs ont montré qu'il est possible d'améliorer les performances de l'utilisateur en couplant la prédiction avec une technique de type "extension de cibles"[17], même si la prédiction intervient vers la fin du mouvement[19].

Ce modèle prend en compte l'intégralité du geste de désignation. Cependant, des études ont montré qu'un geste de sélection peut se décomposer en une phase balistique rapide, et une phase de contrôle plus lente mais plus précise ([12, 7]). Nous émettons l'hypothèse que décomposer le mouvement en une phase d'accélération et une phase de décélération permet d'estimer la fin du mouvement plus précisément. Notre algorithme SPEED, détaillé dans la section suivante, consiste à modéliser exclusivement la phase de décélération pour en déduire la cible visée.

## 3. ALGORITHME SPEED

Notre méthode repose sur l'idée qu'un geste de sélection peut se décomposer en une phase d'accélération, terminée par un pic de vitesse, et une phase de décélération, comme illustrée sur la figure 1.

À partir de la phase de décélération d'un geste (*i.e.* lorsqu'un pic de vitesse a été détecté), nous approchons le profil de vitesse à chaque instant par une courbe quadratique. Nous



**Figure 1:** Décomposition d'un profil de vitesse obtenu pendant le test détaillé en section *Expérimentation*.

obtenons alors la distance qu'il reste à parcourir avant la fin du mouvement. Si cette distance nous semble fiable, nous en déduisons la position de la cible visée.

Afin de distinguer deux gestes de sélection successifs, nous considérons qu'un geste commence lorsque la vitesse de l'utilisateur dépasse un certain seuil, et se termine lorsque la vitesse redescend en-dessous de ce même seuil. Ce seuil de vitesse, que nous notons  $V_0$ , est une constante dont la valeur est fixée empiriquement.

Lorsqu'un nouveau geste est détecté, l'algorithme est réinitialisé (toute mémoire des gestes passés est perdue), et suit ces étapes :

1. *Détection du pic* : La détection d'un pic de vitesse distingue la phase d'accélération et la phase de décélération.
2. *Prédiction de la distance* : La distance à laquelle se trouve la cible visée est prédite pendant la phase de décélération.
3. *Prédiction de la cible* : La cible visée est déduite de la distance prédite et de la direction du mouvement.

Ces étapes sont détaillées dans les parties suivantes.

### 3.1 Détection de pics

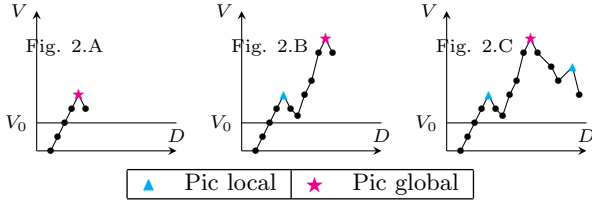
Dans la suite, nous notons  $(d_t, v_t)$  la donnée du profil de vitesse acquise au temps  $t$ , avec  $d_t$  la distance parcourue, et  $v_t$  la vitesse. Nous notons également  $(d_{\text{start}}, v_{\text{start}})$  la première donnée acquise du geste (on a donc  $v_{\text{start}} \geq V_0$ ). Nous appelons *pic local* une donnée  $(d_{t_{\text{pic}}}, v_{t_{\text{pic}}})$  du profil de vitesse tel que la vitesse en ce point soit supérieure aux vitesses acquises juste avant et juste après ce point, soit :

$$v_{t_{\text{pic}}} \geq \max(v_{t_{\text{pic}}+1}, v_{t_{\text{pic}}-1}). \quad (2)$$

Un pic local au temps  $t_{\text{pic}}$  ne pourra alors être détecté qu'au temps  $t_{\text{pic}} + 1$ . Nous appelons aussi *pic global* le pic local, parmi ceux détectés, dont la vitesse est la plus élevée. Nous le notons  $(d_{t_{\text{picg}}}, v_{t_{\text{picg}}})$ . La figure 2 illustre cette détection de pics.

### 3.2 Prédiction de la distance

À chaque instant  $t$ , si un pic global a déjà été détecté, nous considérons tous les points du profil de vitesse entre  $t_{\text{picg}}$  et  $t$ . Nous approchons ces données par une courbe quadratique. Cette courbe a deux racines,  $d_{r_1}$  et  $d_{r_2}$ , avec  $d_{r_1} < d_{\text{picg}} < d_{r_2}$ . Cette deuxième racine  $d_{r_2}$  correspond à la distance à laquelle nous estimons que la vitesse de

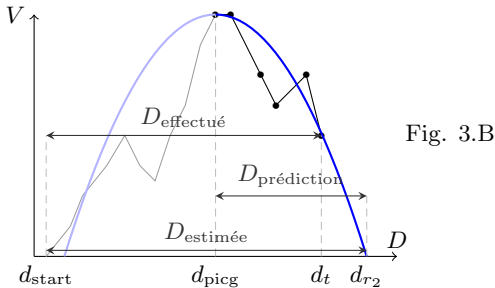


**Figure 2: Détection de pics.** Fig. 2.A : Premier pic local et global. Fig. 2.B : Nouveau pic local. Sa vitesse est supérieure à celle du pic global actuel, qui est donc mis à jour. Fig. 2.C : Nouveau pic local. Sa vitesse est inférieure à celle du pic global actuel, qui reste donc inchangé.

l'utilisateur va s'annuler, donc à la fin de son geste. La distance  $D_{\text{estimée}} = d_{r_2} - d_{\text{start}}$  est alors la distance estimée entre le début et la fin du geste. Le calcul de cette estimation est un procédé sensible au bruit. Avec trop peu de données, le calcul peut donner des résultats aberrants. À l'opposé, si nous attendons trop longtemps, nous perdons tout le bénéfice d'une prédiction. Nous posons :

$$\rho(t) = \frac{D_{\text{effectué}}}{D_{\text{estimée}}}, \quad (3)$$

où  $D_{\text{effectué}} = d_t - d_{\text{start}}$  est la distance effectuée depuis le début du geste. Cette quantité  $\rho(t)$  (représentée sur la figure 3) est la portion de distance réellement effectuée, par rapport à la distance totale estimée du geste. Une estimation à l'instant  $t$  est considérée comme fiable si la condition  $\rho(t) \geq \rho_0$  est vérifiée, avec  $\rho_0$  une constante dont la valeur est fixée empiriquement. Nous appelons alors *distance de la prédiction* la quantité  $D_{\text{prédiction}} = d_{r_2} - d_{\text{picg}}$ , et nous posons  $P_0$  le point de la scène pour lequel cette estimation fiable a été calculée.



**Figure 3: Calcul de la distance de la prédiction.**

### 3.3 Prédiction de la cible

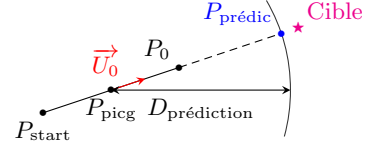
Au moment où nous calculons la distance de la prédiction  $D_{\text{prédiction}}$ , nous définissons la *direction de la prédiction* comme étant le vecteur unitaire :

$$\vec{U}_0 = \frac{\overrightarrow{P_{\text{picg}}P_0}}{\|\overrightarrow{P_{\text{picg}}P_0}\|}. \quad (4)$$

Notre prédiction  $P_{\text{prédic}}$  est alors définie par la relation :

$$P_{\text{prédic}} = P_{\text{picg}} + D_{\text{prédiction}} \cdot \vec{U}_0. \quad (5)$$

La figure 4 illustre la création de ce point.



**Figure 4: Construction de la prédiction**

### 3.4 Valeurs des paramètres

SPEED dépend de deux facteurs,  $V_0$  et  $\rho_0$ .  $V_0$  est le seuil de vitesse délimitant le début et la fin d'un geste.  $\rho_0$  représente le ratio de distances que l'utilisateur doit effectuer pour que la prédiction soit fiable.

Asano et al. [3] montre que la vitesse maximale atteinte durant un geste de sélection dépend de la distance de la cible. Une valeur trop élevée pour  $V_0$  ne permettrait alors pas de détecter des gestes de sélections pour des cibles proches. Lors de pré-tests, nous avons empiriquement fixé  $V_0$  à 8 mm/s. Nous souhaitons comparer nos résultats avec la méthode actuellement la plus précise, KEP [18], qui prédit une cible à 85% du mouvement. Nous avons alors fixé  $\rho_0$  à 0.85. La section suivante détaille une expérimentation dans laquelle nous évaluons les performances de SPEED.

## 4. EXPÉRIMENTATION

Nous présentons ici le protocole et les résultats de l'expérience que nous avons menée afin d'étudier la précision de notre algorithme de prédiction.

### 4.1 Matériel utilisé

L'expérience a été réalisée sur une station de travail classique avec un écran LCD 22" de résolution  $1680 \times 1050$ . Le ratio contrôle-affichage de la souris est constant et égal à 0.125. Ses mouvements sont capturés à une fréquence moyenne de 100Hz.

### 4.2 Protocole du test

L'objectif de cette expérience est de quantifier la précision de SPEED, et de la comparer avec l'algorithme KEP. La tâche à effectuer est une tâche de sélection monodirectionnelle. Les sélections se font toujours dans la même direction droite-gauche. Les cibles ont une taille de 16, 32, 64 et 128 pixels. Les distances entre la cible de départ et la cible d'arrivée sont 512, 1024, et 1536 pixels. Les 12 combinaisons Taille/Distance fournissent une gamme d'Indice de Difficulté (voir l'équation 1) entre 2.32 et 6.60. Lors de cette expérience, nous avons fixé les constantes  $V_0 = 8 \text{ mm/s}$  et  $\rho_0 = 0.85$ . L'expérience est composée d'une phase d'entraînement et une phase de test lors desquelles chaque combinaison Taille/Distance est présentée respectivement 2 et 10 fois à l'utilisateur. L'ordre des tailles de cibles présentées aux utilisateurs est établi selon un carré latin de taille 4. L'expérience dure environ 15 minutes.

L'environnement présenté à l'utilisateur est une grille de cases. L'une d'elles, en rouge, représente la case de départ. Une autre, en vert, représente la cible à sélectionner. Une sélection consiste en un clic droit sur la case de départ, puis un clic gauche sur la cible. L'alternance clic droit/clic gauche

permet d'éviter des double-clics involontaires. L'utilisateur n'est pas informé des calculs effectués par SPEED. Dix sujets ont participé à cette expérience.

### 4.3 Résultats

Une prédiction à l'intérieur de la cible visée est *Réussie*. Une prédiction dans une case adjacente à celle visée est *Proche*. La table 1 détaille le taux de réussite pour chaque combinaison Taille/Distance. Ce taux de réussite varie de 26.2% pour l'indice de difficulté maximal 6.60, à 74.3% pour l'indice de difficulté minimal 2.32. Nous avons également implémenté KEP [18, 19] afin de comparer nos résultats. Le calcul des prédictions pour SPEED et KEP est effectué en post-traitement sur les mêmes données. Nous obtenons un gain moyen (différence entre le taux de réussite moyen de SPEED et celui de KEP) de respectivement 27.3%, 31.4%, 27.1% et 17.3% sur les cibles de 16, 32, 64 et 128 pixels. Ces résultats montrent que baser la prédiction uniquement sur la phase de décélération permet d'être plus précis que considérer le profil de vitesse dans son intégralité.

Taille	Distance		
	512	1024	1536
16	32.1 (17.6) / 5.0	30.8 (14.2) / 2.0	26.2 (13.1) / 0.0
32	43.0 (19.4) / 17.0	38.1 (9.9) / 3.0	35.1 (15.2) / 2.0
64	50.5 (26.6) / 35.0	45.4 (17.3) / 16.0	46.5 (10.1) / 10.0
128	74.3 (22.3) / 76.0	60.9 (14.5) / 33.0	47.9 (14.2) / 22.0

**Table 1: Performances de SPEED et KEP pour chaque combinaison Taille/Distance. Nous indiquons le taux de réussite (pourcentage de prédictions réussies en %), le taux de prédictions proches entre parenthèses, et le taux de réussite de KEP en bleu.**

## 5. CONCLUSION

Nous avons présenté un algorithme de prédiction de fin de mouvement de sélection, basée sur une modélisation de la phase de décélération par une courbe quadratique. Pour une prédiction obtenue à 85% du mouvement, nous obtenons un meilleur taux de réussite que KEP. Néanmoins, l'expérience menée ici ne traitait qu'une unique direction, en l'absence de distracteurs ; de nouvelles expériences seront nécessaires pour étudier les performances de SPEED dans des applications réelles.

Nous espérons aussi que notre algorithme est robuste à un changement d'échelle et de dimension. En particulier, nous souhaitons étudier les conditions sous lesquelles notre méthode pourrait s'étendre à des sélections 3D dans des environnements de réalité virtuelle. La trajectoire d'un pointage 3D n'étant pas rectiligne [5, 9, 14], une adaptation de la direction de la prédiction sera nécessaire.

## 6. REFERENCES

- [1] J. Accot and S. Zhai. Refining fitts' law models for bivariate pointing. In *Proc. ACM SIGCHI*, pages 193–200, 2003.
- [2] C. Appert, O. Chapuis, and M. Beaudoin-Lafon. Evaluation of Pointing Performance on Screen Edges. In *Proc. ACM AVI*, Napoli, Italy, May 2008.
- [3] T. Asano, E. Sharlin, Y. Kitamura, K. Takashimai, and F. Kishino. Predictive Interaction using the Delphian Desktop. In *Proc. ACM UIST*, pages 133–141, 2005.
- [4] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-Pop and Drag-and-Pick : techniques for accessing remote screen content on touch- and pen-operated systems. In *Proc. of Interact 2003*, pages 57–64, 2003.
- [5] E. Bizzi, N. Accornero, W. Chapple, and N. Hogan. Posture Control and Trajectory Formation during Arm Movement. *Journal of Neuroscience*, 4(11):2738–2744, 1984.
- [6] R. Blanch and M. Ortega. Benchmarking Pointing Techniques with Distractors: Adding a Density Factors to Fitt's Pointing Paradigm. In *Proc. ACM CHI*, Vancouver, Canada, May 2011.
- [7] D. Elliott, W. F. Helsen, and R. Chua. A century later : Woodworth's (1899) two component-model of goal-directed aiming. *Psychological bulletin*, 127(3):342–357, 2001.
- [8] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.
- [9] J. R. Flanagan and D. J. Ostry. Trajectories of Human Multi-Joint Arm Movements: Evidence of Joint Level Planning, 1989.
- [10] T. Grossman and R. Balakrishnan. Pointing at Trivariate Targets in 3d Environments. In *Proc. ACM SIGCHI*, pages 447–454, 2004.
- [11] Y. Guiard, R. Blanch, and M. Beaudoin-Lafon. Object Pointing: A Complement to Bitmap Pointing in GUIs. In *Proc. Graphic Interfaces '04*, pages 9–16, May 2004.
- [12] N. Hogan. An organizing principle for a class of voluntary movements. *Journal of Neuroscience*, 4(11):2745–2754, 1984.
- [13] H. Iwase and A. Murata. Modelling of human's three-dimensional movement - extending fitts' model to three-dimensional pointing task-. In *Proc. IEEE International Workshop on Robot and Human Interactive Communication*, pages 594–599, 2001.
- [14] T. Kang, J. He, and S. I. H. Tillery. Determining natural arm configuration along a reaching trajectory. In *Proc. IEEE Engineering in Medicine and Biology Society*, 2003.
- [15] E. Lank, Y.-C. N. Cheng, and J. Ruiz. Endpoint Predictions Using Motion Kinematics. In *Proc. ACM SIGCHI*, pages 637–646, 2007.
- [16] I. S. MacKenzie and W. Buxton. Extending Fitts' law to two-dimensional tasks. In *Proc. ACM SIGCHI*, pages 219–226, 1992.
- [17] M. J. McGuffin and R. Balakrishnan. Fitts Law and Expanding Targets: Experimental Studies and Designs for User Interfaces. *ACM ToCHI*, 12(4):338–422, 2005.
- [18] J. Ruiz and E. Lank. Effects of Target Size and Distance on Kinematics Endpoint Prediction. Technical report, Univ. of Waterloo, 2009.
- [19] J. Ruiz and E. Lank. Speeding Pointing in Tiled Widgets: Understanding the Effects of Target Expansion and Misprediction. In *Proc. ACM IUI*, Hong Kong, China, 2010.