

# Aide haptique prédictive à la sélection de cibles en trois dimensions

Jonathan Wonner \*

ENS Cachan - Antenne de Bretagne  
LSIIT Université de Strasbourg

Jérôme Grosjean †

LSIIT Université de Strasbourg

Antonio Capobianco ‡

LSIIT Université de Strasbourg

Dominique Bechmann §

LSIIT Université de Strasbourg

## RÉSUMÉ

La sélection d'un objet dans un environnement virtuel en trois dimensions est une tâche complexe ; la planification du geste pouvant être faussée par une mauvaise perception de la profondeur de la scène, il peut être très difficile pour un utilisateur d'atteindre sa cible. L'utilisation d'aimants est une aide courante à la sélection dans un tel environnement. Cependant, la profusion d'aimants dans une scène 3D entraîne une baisse de confort pour l'utilisateur, et même une baisse de performances dans le cas d'une scène dense. Notre objectif est de produire une aide haptique plus efficace et plus confortable pour l'utilisateur. Dans cet article, nous nous intéressons à une méthode pour aider l'utilisateur à sélectionner une cible, en appliquant un aimant uniquement sur la cible visée. Cette méthode estime la position finale en se basant sur le pic de vitesse du mouvement effectué, et place un aimant sur la cible la plus proche de cette position. Les résultats de l'évaluation de cette première méthode nous conduisent à l'élaboration d'une seconde méthode pour estimer la cible de l'utilisateur, basée sur l'ensemble des données du geste. L'estimation du point d'arrivée du mouvement est obtenue en modélisant le profil de vitesse par une équation du second degré. Enfin, nous évaluons les performances de cette seconde technique.

**Index Terms:** H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities H.5.2 [Information Interfaces and Presentation]: User Interfaces—Haptic I/O H.5.m [Information Interfaces and Presentation]: Miscellaneous—Movement Prediction

## 1 INTRODUCTION

Dans le domaine de la réalité virtuelle, la sélection est une des principales tâches élémentaires d'interaction. Cette tâche est très simple et naturelle dans le monde réel, mais pose des problèmes, matériels et cognitifs, en environnement immersif. En effet, dans une scène 3D fixe sur un écran 2D d'une station de travail classique, il est difficile de percevoir la profondeur et la position relative des objets. Il n'est néanmoins pas impossible d'atteindre sa cible ; d'autres repères visuels peuvent aider à se repérer, comme la taille du curseur qui diminue lorsqu'il s'éloigne de l'utilisateur. Cependant, la perception incorrecte de la profondeur rend difficile la sélection d'une cible. L'utilisation de la stéréoscopie permet d'améliorer cette situation, en particulier dans le cadre d'un environnement immersif de type Workbench, mais les gains de performances restent peu satisfaisants sur une station de travail. L'utilisation de dispositifs à retour d'efforts, comme un Phantom

(voir figure 3), a souvent été présentée comme une solution potentielle à ce problème. La plupart des solutions proposées reposent sur l'idée d'attirer rapidement l'utilisateur vers sa cible. Un aimant est simulé sur chaque point d'intérêt de la scène. Lorsque le curseur entre dans la zone d'attraction d'un aimant, il est directement conduit sur celui-ci. Cette approche permet à l'utilisateur d'atteindre sa cible rapidement et précisément. Cependant, dans le cas d'une scène dense en points d'intérêt, et donc en aimants, le curseur peut être attiré par un aimant qui ne correspond pas à la cible de l'utilisateur. Celui-ci devra forcer pour se libérer de son attraction, avec le risque d'être immédiatement attiré par un autre aimant proche. Il en résulte une perte de confort pour l'utilisateur, et une perte de temps.

Pour palier cet inconvénient, nous souhaitons ne pas utiliser des aimants en permanence sur tous les points d'intérêt de la scène, mais seulement sur la cible à sélectionner. Ainsi, l'utilisateur garde un mouvement libre au début de son geste, et est attiré uniquement quand nous savons quel élément est sa cible. Cette approche suppose que nous soyons capables de prédire cette cible. Dans le cadre d'interfaces graphiques en deux dimensions, la prédiction de la cible d'un geste de désignation est un des moyens utilisés pour améliorer les performances de l'utilisateur. L'étude et la modélisation d'un mouvement de désignation, en particulier sa vitesse, permet de prévoir le comportement de l'utilisateur, et de déterminer la cible du mouvement.

Cet article se place dans ce contexte. Nous souhaitons deviner la cible d'un mouvement de sélection dans une scène en trois dimensions, par exemple un nuage de points 3D, et nous servir de cette prédiction pour attirer l'utilisateur sur la cible à l'aide d'un aimant. En section 2, nous présentons plusieurs aides haptiques existantes à la sélection de cibles, ainsi que des techniques de prédiction dans des scènes en dimension deux. Puis, en section 3, nous proposons une première méthode de prédiction en trois dimensions basée sur le pic de vitesse d'un mouvement balistique. Les résultats de l'évaluation de cette technique nous ont conduit à une seconde méthode, détaillée en section 4, et basée sur une étude de l'ensemble du profil de vitesse. Son évaluation étant en cours de réalisation, nous terminons en section 6 par un aperçu des perspectives engendrées par ce travail.

## 2 ÉTAT DE L'ART

Dans une tâche de modélisation en environnement immersif, un problème important est la modification des objets 3D, généralement en agissant directement sur un élément géométrique de ces objets, comme un sommet, une arête ou une face. Cette tâche n'est cependant possible que si cet élément a été sélectionné au préalable. *Picon*[11] propose que chaque élément géométrique soit associé à un aimant. Afin de favoriser la sélection des éléments de petites dimensions (les sommets), la zone d'attraction des sommets est prioritaire sur celle des arêtes, elle-même prioritaire sur la zone d'attraction des faces. Cette approche permet de parcourir rapidement la surface d'un objet 3D, et de sélectionner précisément les éléments à modifier. Cependant, elle n'est pas applicable dans le cas d'un nuage de points ; on aurait alors seulement des aimants sur les points, ce qui

\* e-mail : jonathan.wonner@ens-cachan.org

† e-mail : grosjean@unistra.fr

‡ e-mail : a.capobianco@unistra.fr

§ e-mail : bechmann@unistra.fr

peut, comme précisé en section 1, entraîner des pertes au niveau du confort et des performances. Dans le cas spécifique d'un nuage dont les points forment les sommets d'un polyèdre convexe (voir figure 1), Burel[2] propose une adaptation de la méthode de Picon : l'utilisateur est placé à l'intérieur du polyèdre convexe, et des murs haptiques non aimantés, sont placés sur les faces de celui-ci. L'utilisateur peut alors se servir des murs et des arêtes pour "glisser" jusqu'au point souhaité, la convexité permettant de s'y arrêter précisément.

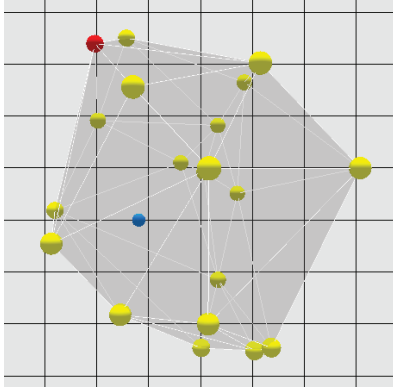


FIGURE 1: Le nuage de point forme un polyèdre convexe dans lequel est placé l'utilisateur (en bleu). Image tirée de [2].

Cette méthode permet de s'affranchir des aimants, améliorant ainsi le confort de l'utilisateur. Une application est proposée par Essert-Villard & Capobianco[5]. Les murs haptiques sont alors utilisés dans le cadre d'un menu 3D. Les items du menu sont les sommets d'un polyèdre, dont les bords sont infranchissables. Une série de test montre que les murs haptiques permettent d'être plus précis et rapide qu'avec des aimants sur les items du menu.

Dans cet article, nous couplons des effets haptiques à une prédiction de cibles. Dans le cadre d'une interface graphique en deux dimensions, une première approche pour prédire la cible de l'utilisateur est proposée par Guiard[6]. Dans une interface classique avec plusieurs icônes disséminées sur l'écran, l'espace vide entre les icônes n'est généralement pas un élément d'intérêt, et doit pourtant être parcouru par le curseur de l'utilisateur pour passer d'une icône à une autre. Il s'ensuit une perte de temps, et donc de performances. Guiard propose de considérer la direction du curseur au moment où celui-ci quitte un item, et de la faire "sauter" vers le prochain item dans cette direction. Les gains de temps sont significatifs dans le cas d'une scène peu dense, mais faibles voire inexistantes pour des densités plus élevées. Ceci s'explique par le fait que pour des fortes densités, l'espace entre les objets est réduit, et arriver à une cible lointaine implique de passer par tous les objets entre la position de départ du curseur et cette cible. Asano[1] présente alors un modèle basé sur la vitesse de l'utilisateur, pour déterminer plus précisément la cible visée. D'après ses observations, la vitesse maximale atteinte durant un mouvement de désignation dépend linéairement de la distance entre la cible et le point de départ de l'utilisateur. Cette relation linéaire dépend non seulement de l'utilisateur, mais aussi de la direction dans laquelle se dirige l'utilisateur. La méthode d'Asano consiste tout d'abord à déterminer ces relations entre vitesse maximale et distance par une phase de calibration. Ensuite, lors d'un mouvement de l'utilisateur, l'algorithme détermine la valeur de la vitesse maximale atteinte, et en déduit la distance entre le curseur au début du mouvement et la cible. Ceci permet, en suivant la direction du curseur, d'obtenir une position de

prédiction. Asano fait "sauter" le curseur sur cette position, dans le même but que Guiard. En section 3, nous expérimentons cette approche dans un environnement en trois dimensions.

Une approche différente de la prédiction de cibles est proposée par Lank[9]. Lors d'un mouvement de désignation, la courbe de vitesse de l'utilisateur en fonction de la distance parcourue, est une courbe en cloche, que Lank modélise par une équation du second degré. Les racines de cette équation, *ie.* les distances à laquelle la vitesse de l'utilisateur s'annule, donnent une estimation de la distance à laquelle se trouve le point visé. Lank calcule cette approximation du second degré à différents instants du mouvement, et estime la précision de la prédiction obtenue. Ces résultats montrent qu'à 80% du mouvement, la prédiction est la plus précise : 81% des positions prédites se situent sur ou à proximité de la cible. En section 4, nous reprenons cette approche dans une scène en trois dimensions.

### 3 MÉTHODE DE PRÉDICTION BASÉE SUR LE PIC DE VITESSE

Selon Woodworth[4], un geste de désignation peut se décomposer en deux phases : la phase de *planification* et la phase d'*ajustement*. Pour atteindre une cible, un utilisateur commence son mouvement par un geste rapide dans sa direction, puis l'ajuste plus lentement pour être plus précis. De cette décomposition résulte un pic de vitesse au cours du temps, comme l'illustre la figure 2.

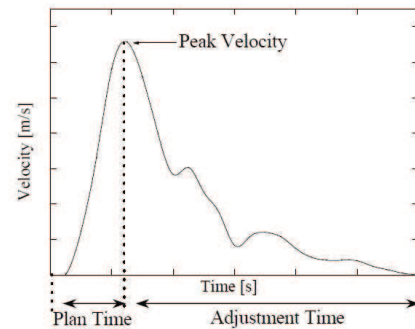


FIGURE 2: Courbe de vitesse en fonction du temps d'un mouvement de désignation en deux dimensions. Figure tirée de [1].

Asano[1] propose, dans le cadre d'une interface graphique en deux dimensions, une méthode de prédiction basée sur ce pic de vitesse. Ses évaluations montrent que l'amplitude du pic (*ie.* la vitesse maximale atteinte durant le mouvement de sélection) croît linéairement avec la distance entre la position de départ de l'utilisateur, et la cible. La distance que cherche à couvrir l'utilisateur est donnée par la relation :

$$D = a \cdot V + b, \quad (1)$$

où  $D$  est la distance entre le point de départ du curseur et la cible,  $V$  le pic de vitesse, et  $a$  et  $b$  deux constantes. Ces deux coefficients  $a$  et  $b$  dépendent de l'utilisateur et de la direction du mouvement. Une phase de calibration est nécessaire pour déterminer ces constantes pour chaque utilisateur et chaque direction. L'algorithme de prédiction est le suivant : lorsque un pic de vitesse, plus élevé qu'un seuil prédéterminé, est détecté, alors la relation (1) est utilisée pour déterminer la distance à laquelle se situe la cible. Le curseur est alors amené à cette distance, dans la direction courante du curseur. Un gain de temps et de précision significatif est observé pour la sélection de cibles lointaines. Nous souhaitons adapter cette méthode au cas d'une sélection en trois dimensions, et coupler les prédictions obtenues avec une aide haptique.

### 3.1 Profil de vitesse en trois dimensions

Notre idée est d'adapter la méthode de prédiction de *Asano* en trois dimensions. Lors d'un premier test, nous voulons valider les deux hypothèses suivantes :

- H1** le modèle de décomposition du mouvement en deux phases de *Woodworth* peut s'étendre en trois dimensions ;
- H2** l'amplitude du pic de vitesse croît linéairement avec la distance entre le point de départ du curseur et la cible.

Ce premier test consiste à effectuer une série de sélections en trois dimensions à l'aide d'un Phantom Omni (voir la figure 3). Ce périphérique nous permet de déplacer un curseur dans une scène en trois dimensions, et peut être utilisé pour simuler des aimants et des murs haptiques.



FIGURE 3: Un Phantom Omni, un dispositif de mouvement en trois dimensions à retour de forces.

Le protocole de ce test est le suivant : une cible s'affiche dans une scène en trois dimensions, représentée en figure 4. Nous demandons à l'utilisateur de sélectionner cette cible (*ie.* de cliquer lorsque le curseur est dessus). À ce moment, une autre cible s'affiche, et l'utilisateur doit répéter le processus. La consigne donnée est de concilier vitesse et précision. Dix personnes ont effectué ce test, en réalisant chacun une série de cinquante sélections.

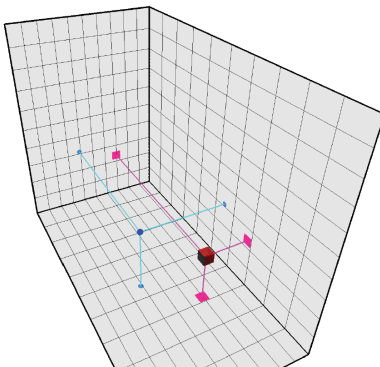


FIGURE 4: Scène 3D du test. Le curseur de l'utilisateur est représenté par une sphère bleue, la cible à atteindre par un cube rouge. Nous affichons sur trois plans l'ombre du curseur et de la cible pour apporter une information visuelle supplémentaire sur leurs positions.

Nous constatons tout d'abord que notre première hypothèse peut être validée. Nous observons, comme illustré par la figure 5, que le

mouvement peut être à nouveau décomposé en une phase de planification, avec son pic de vitesse caractéristique, et une phase de réajustement, où l'utilisateur affine plus lentement son geste pour atteindre la cible précisément. Nous voulons maintenant relier l'amplitude de ce pic de vitesse à la distance (dite *initiale*) entre le point de départ de l'utilisateur et la cible. La figure 6 représente la vitesse maximale  $V_{max}$  en fonction de la distance initiale  $D_{initial}$ .

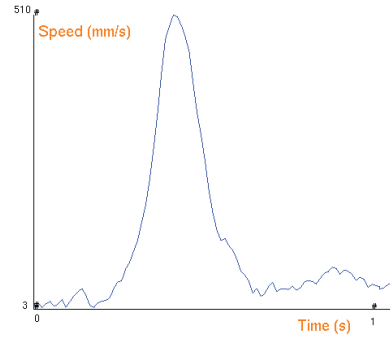


FIGURE 5: Courbe de vitesse en fonction du temps d'un mouvement de sélection en 3D, réalisé à l'aide d'un Phantom.

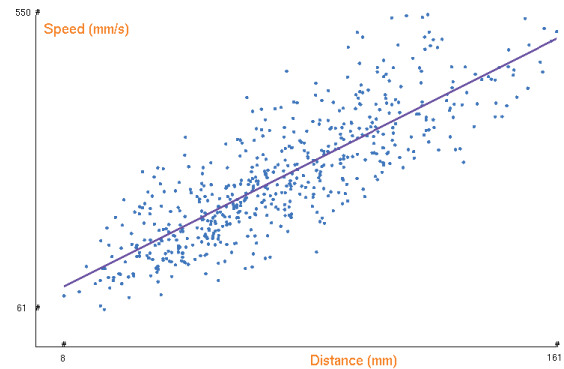


FIGURE 6: Nuage représentant l'amplitude du pic de vitesse  $V_{max}$  en fonction de la distance initiale  $D_{initial}$ . Chaque point correspond à une sélection. La droite en violet est la droite obtenue par régression linéaire sur ces données.

Un comportement linéaire se dégage, validant notre deuxième hypothèse. La méthode que nous proposons consiste à détecter le pic de vitesse d'un utilisateur lors d'un mouvement de désignation, et en déduire la distance à laquelle se trouve la cible. Pour déterminer le pic de vitesse, notre algorithme est le suivant (nous appelons  $t + 1$  l'instant auquel notre système acquiert la donnée juste après celle acquise à l'instant  $t$ ) : à l'instant  $t + 1$ , nous estimons que le pic de vitesse est intervenu au temps  $t$  si la vitesse  $v(t)$  au temps  $t$  est supérieure à un seuil prédéterminé, et si cette vitesse est supérieure aux vitesses aux temps  $t - 1$  et  $t + 1$ .

Nous souhaitons évaluer notre prédiction à travers un second test. Les coefficients  $a$  et  $b$  que nous utilisons dans ce test sont  $a = 5s^{-1}$  et  $b = 76mm.s^{-1}$ , qui correspondent à des coefficients moyens sur l'ensemble des participants du premier test.

### 3.2 Protocole de test

Nous voulons estimer la performance de notre méthode couplée à une aide haptique. Notre test consiste en une série de sélections de points dans une scène en trois dimensions. Comme l'illustre la figure 7, la scène est constituée de plusieurs points disposés en grille, la distance entre deux points adjacents étant de  $20\text{mm}$ . Lors d'une sélection, l'un des points change de couleurs. L'utilisateur doit alors se diriger vers lui et le désigner. Lors du déplacement, notre algorithme recherche un pic de vitesse plus rapide qu'un seuil prédéterminé. Lorsqu'un tel pic est détecté, la relation de linéarité est utilisée pour déterminer la distance à laquelle se trouve la cible. Une position 3D est obtenue en prenant en compte la direction courante du curseur ; cette position est la prédiction que nous cherchons à évaluer. Nous déterminons le point de la grille le plus proche de cette position. Un aimant, dont le rayon d'attraction est égal au tiers de la taille d'une arête de la grille, est simulé sur ce point. Nous voulons comparer cette aide haptique prédictive à une aide haptique permanente, dans laquelle des aimants sont simulés sur tous les points de la grille. Pour chacune de ses deux modalités (aide permanente ou prédictive), chaque utilisateur doit effectuer 32 sélections, ce qui correspond à différentes directions et distances : 8 sélections à  $17\text{mm}$ , ce qui correspond aux 8 sommets d'un cube dans la grille, et pour chacune des 6 directions principales, 2 sélections à  $33\text{mm}$  (cible à un cube de distance), et 2 sélections à  $52\text{mm}$  (cible à deux cubes de distance).

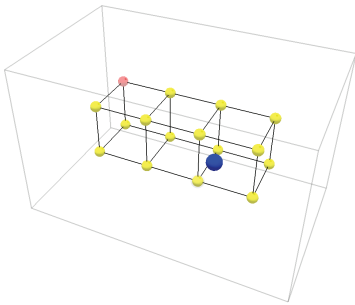


FIGURE 7: Scène 3D du test de sélection.

### 3.3 Résultats

Nous constatons tout d'abord que la prédiction est très imprécise. La distance moyenne, pour l'aide avec prédiction, entre le point de prédiction et la cible est de  $17.45\text{mm}$ , soit presque la taille d'une arête de la grille. Il en résulte que notre algorithme se trompe souvent de cible. Cela engendre un autre phénomène non prévu : la consigne donnée est de sélectionner les cibles en alliant rapidité et précision, sans tenir compte de la présence ou non d'un aimant sur la cible visée ; néanmoins, nous avons observé que les utilisateurs ont tendance à tourner autour de la cible jusqu'à ce que notre algorithme de prédiction place un aimant sur le point souhaité, pour pouvoir le sélectionner précisément. Ainsi, le temps moyen passé à proximité de la cible (ie. à une distance inférieure au rayon d'attraction d'un aimant) est presque deux fois plus long ( $1.08\text{s}$  contre  $0.57\text{s}$ ) pour l'aide prédictive que pour l'aide permanente. L'une de nos hypothèses est qu'au niveau du confort, l'aide avec prédiction serait préférée à l'aide permanente. Le questionnaire d'évaluation subjective que nous demandons aux participants de remplir à la fin du test, et demandant de noter les deux techniques selon plusieurs critères, révèle que l'aide avec prédiction est jugée plus agréable à utiliser que l'aide permanente (une moyenne de  $4.2$  contre  $3.6$ ). Cette expérience confirme que la présence de trop nombreux aimants rend le déplacement dans la scène peu

confortable. Ainsi, certains participants préfèrent ne pas avoir d'aide du tout, plutôt qu'une aide aussi contraignante.

Selon nous, le manque de précision de la prédiction provient de deux facteurs. Notre algorithme n'utilise qu'une seule relation linéaire pour tous les utilisateurs et pour toutes les directions. Une possibilité pour améliorer les performances de notre méthode consisterait à calibrer l'algorithme pour chaque utilisateur et pour chaque direction. Cependant, l'erreur commise sur les coefficients par une régression linéaire sur des profils de vitesse est trop élevée pour espérer obtenir une prédiction vraiment précise. En effet, pour une tâche de sélection donnée (même position de départ du curseur et même position de la cible), un utilisateur ne fait jamais deux fois exactement le même geste. Le mouvement peut être exécuté rapidement ou lentement. Cette situation est illustrée sur la figure 6. Une régression linéaire sur ces données donne la relation  $V_{\max} = a * D_{\text{initial}} + b$ , avec  $a = 2.7\text{s}^{-1}$  et  $b = 76\text{mm.s}^{-1}$ . L'erreur quadratique moyenne commise est  $EQ(a, b) = 102.2\text{mm.s}^{-1}$ . Nous estimons que cette erreur est trop importante pour que le pic de vitesse seul donne une information suffisamment précise pour prédire la cible visée.

De plus, au cours de ce test, nous demandons aux utilisateurs d'effectuer un mouvement en trois dimensions. L'absence de stéréoscopie et le nombre restreint de repères visuels dans la scène faussent la planification du geste. La direction dans laquelle se déplace l'utilisateur n'est alors pas précisément celle de la cible, ce qui engendre une erreur dans la position de la prédiction.

Dans la section 4, nous souhaitons améliorer les performances de notre prédiction en intervenant sur ces deux facteurs. Pour cela, nous considérons l'intégralité du profil de vitesse, et pas seulement le pic de vitesse, pour estimer plus précisément le point d'arrivée du mouvement. Nous évaluons cette nouvelle approche dans le cadre d'une scène simple, afin de ne pas fausser la planification.

## 4 MÉTHODE DE PRÉDICTION BASÉE SUR UNE MODÉLISATION DE LA VITESSE

En section 3, nous n'avons pas tiré de résultats satisfaisants d'un modèle basé uniquement sur le pic de vitesse d'un mouvement. Une approche différente consiste à considérer le geste dans son ensemble.

### 4.1 Modèle théorique de la vitesse

Hogan[8] considère une autre description d'un mouvement de désignation. Un tel mouvement se décompose en une phase balistique et une phase de correction. La phase balistique correspond à un geste rapide dans la direction de la cible, tandis que la phase de correction est plus lente mais plus précise pour atteindre la cible. Hogan montre que pour un geste balistique non contraint en deux dimensions, la vitesse varie de manière régulière (pas de brusques changements d'accélération). Hogan en déduit une équation reliant le temps  $t$  à la distance parcourue  $x$  :

$$x(t) = D \left[ 10 \left( \frac{t}{T} \right)^3 - 15 \left( \frac{t}{T} \right)^4 + 6 \left( \frac{t}{T} \right)^5 \right], \quad (2)$$

où  $D$  est la distance totale parcourue, et  $T$  le temps total du trajet. Lank[9] utilise ce modèle pour prédire la cible d'un utilisateur dans une scène en deux dimensions. En normalisant la distance et le temps (ie.  $D = 1$  et  $T = 1$ ), l'équation (2) devient

$$x(t) = 10t^3 - 15t^4 + 6t^5. \quad (3)$$

Une équation pour la vitesse  $v$  du mouvement est alors, par simple dérivation,

$$v(t) = 30t^2(t - 1)^2. \quad (4)$$



Afin d'éliminer tout bruit provenant de la durée du geste, *Lank* représente ces deux équations dans un graphe (*distance, vitesse*) (voir la figure 8). Il approche la courbe en cloche de ce modèle théorique par une équation du second degré. L'algorithme de prédiction de *Lank* consiste à évaluer, à un instant  $t$ , le pourcentage de distance qu'a déjà parcouru un utilisateur par rapport à la distance qu'il doit parcourir jusqu'à son point de destination, en comparant son profil de vitesse avec le modèle théorique. *Lank* en déduit alors la distance à laquelle se situe la cible. Sa position précise est ensuite obtenue en considérant la direction du curseur à l'instant  $t$ . Les évaluations de *Lank* montrent que cet algorithme obtient ses meilleurs résultats lorsque l'utilisateur a parcouru 80% de leur mouvement ; 81% des prédictions sont alors proches des points ciblés.

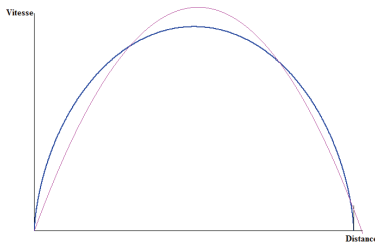


FIGURE 8: Courbe  $v = f(d)$  de vitesse en fonction de la distance parcourue. La courbe bleue représente le modèle théorique. La courbe rose en est une approximation du second degré. Cette approximation est obtenue par la méthode des moindres carrés en utilisant tous les points  $(d, v(d))$ .

Nous souhaitons adapter cette méthode de prédiction pour un geste balistique en trois dimensions. Nous supposons que le modèle théorique de *Hogan* est encore valide dans le cas d'un geste utilisant un Phantom.

## 4.2 Adaptation du modèle théorique

Plus le profil de vitesse est proche du modèle théorique, plus la prédiction sera précise. Nous avons observé les profils obtenus à l'aide du Phantom (courbes de vitesse en fonction de la distance parcourue). Or, comme l'illustre la figure 9, la phase d'accélération et la phase de décélération n'ont pas la même courbure. Plus précisément, la vitesse augmente plus rapidement dans la première phase, qu'elle ne diminue dans la seconde phase.

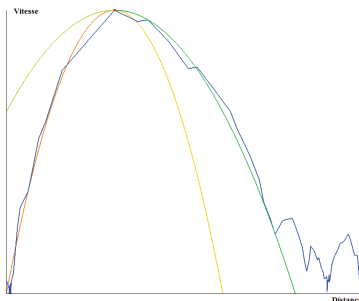


FIGURE 9: Profil de vitesse (en bleu) en fonction de la distance parcourue. Courbe acquise par un Phantom Omni dans une tâche de sélection de cible dans une scène 3D. La courbe verte représente une approximation du second degré du profil de vitesse après le pic de vitesse ; la courbe orange est une approximation avant le pic.

Nous en déduisons qu'avant le pic de vitesse, nous ne pouvons pas estimer l'allure du profil de la phase de décélération, et donc nous ne pouvons pas prédire le point d'arrivée du mouvement. Notre idée est de détecter le pic de vitesse d'après la méthode décrite en section 3.1, puis d'approcher le mouvement à partir de ce pic par une courbe du second degré. L'estimation de la distance à parcourir et du point de prédiction s'inspire ensuite de la méthode de *Lank*. Nous détaillons notre algorithme en section 4.3.

## 4.3 Méthode de prédiction

L'idée de notre algorithme est la suivante : nous considérons le mouvement de l'utilisateur à un certain moment après le pic de vitesse, nous approchons la phase de décélération par une équation du second degré, et nous en déduisons la distance totale du mouvement, puis le point 3D de prédiction. Cependant, l'instant auquel nous effectuons ces calculs influe sur la prédiction. Si nos calculs interviennent trop tôt après le pic, nous risquons de ne pas avoir assez de données pour approcher correctement le mouvement. S'ils sont effectués trop tard, l'utilisateur aura déjà commencé son mouvement d'ajustement au moment où nous prédisons la fin de son mouvement balistique. Le compromis que nous avons utilisé considère le ratio entre la distance prédite, et la distance déjà parcourue (depuis le pic de vitesse). Si ce ratio est plus élevé qu'un certain seuil  $\rho_0$ , alors nous estimons que la prédiction est satisfaisante. L'effet de ce paramètre sur la précision de la prédiction est étudié en 4.4.

Nous considérons dans notre algorithme la courbe de vitesse de l'utilisateur en fonction de la distance parcourue. Cet algorithme de prédiction en temps réel suit les étapes suivantes. La première étape consiste à détecter un pic de vitesse. Si à l'instant  $t_0$ , la vitesse  $v(t_0)$  est supérieure à un seuil  $V$ , et supérieure à  $v(t_0 - 1)$  et  $v(t_0 + 1)$ , alors le point  $(d_0, v_0) = (d(t_0), v(t_0))$  est un pic de vitesse. Nous considérons dans la seconde étape uniquement les données de la courbe à partir de  $t_0$ . Au temps  $t > t_0$ , deux cas peuvent se présenter.

- Si un nouveau pic de vitesse est détecté au temps  $t$ , avec une amplitude plus élevée que le pic en  $t_0$ , alors cela signifie qu'au temps  $t_0$ , nous n'avions pas encore atteint la vitesse maximale du mouvement balistique. Par la suite, l'algorithme considère les données à partir de ce nouveau pic, et non plus à partir de  $t_0$ .
- Si un nouveau pic de vitesse n'est pas détecté, alors notre méthode modélise les données entre les temps  $t_0$  et  $t$  par une équation  $v = f(d)$  du second degré, par la méthode des moindres carrés. Nous rajoutons deux contraintes :  $f(d_0) = v_0$  et  $f'(d_0) = 0$ . Autrement dit, la courbe du second degré a comme point extremum le pic de vitesse  $(d_0, v_0)$ .

Une équation  $v = f(d)$  du second degré calculée au temps  $t$  a deux racines  $d_{-1}$  et  $d_1$ , avec  $d_{-1} < d_0 < d_1$ . La distance  $d_1$  est une estimation de la distance qu'aura parcouru l'utilisateur à la fin de son mouvement balistique. Il s'agit d'une approximation de la distance entre la position de départ du curseur, et la cible. Nous calculons

alors le ratio  $\rho(t) = \frac{d(t) - d_{-1}}{d_1 - d_{-1}}$ . Ce ratio quantifie la distance par-

courue par rapport à la distance à parcourir. Si  $\rho(t) > \rho_0$ , alors l'algorithme s'arrête et renvoie  $d_1$  comme distance de prédiction. Dans le cas contraire, l'algorithme continue en considérant la donnée suivante  $(d(t+1), v(t+1))$ .

Lorsqu'une distance de prédiction est obtenue au temps  $t$ , nous considérons la direction moyenne de l'utilisateur entre  $t_0$  et  $t$ . L'association de cette direction et de la distance prédite nous donne une position 3D, qui est notre prédiction. La figure 10 montre les résultats de cet algorithme sur le profil de vitesse présenté en figure 9.

Nous souhaitons prédire la cible le plus tôt possible dans le mouvement, tout en gardant une précision satisfaisante. Dans la section

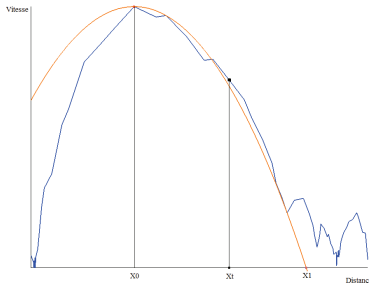


FIGURE 10: Profil de vitesse d'un mouvement de sélection 3D et résultat de l'algorithme présenté en section 4.3. Au point  $X_1$ , l'algorithme a déterminé le point d'arrivée comme étant  $X_1$ . On a  $\rho(t) = 0.75$ , qui est le seuil choisi pour cette figure.

4.4, nous déterminons l'effet du paramètre  $\rho_0$  sur les performances de cet algorithme.

#### 4.4 Précision de la prédiction en fonction de $\rho_0$

Nous supposons que le choix du paramètre  $\rho_0$  a un impact sur la précision de la prédiction. Intuitivement, si l'utilisateur parcourt plus de distance, alors l'algorithme dispose de plus de données, et l'estimation de la prédiction est plus précise. Plusieurs études ont montré en pratique que cette intuition est juste. Ainsi, Lank obtient les meilleurs résultats de précision pour une distance parcourue de 80%. De même, dans le cadre d'une interface graphique en deux dimensions, McGuffin[10] augmente la taille d'un item lorsque 90% de la distance entre le départ de l'utilisateur et l'item est parcourue, et obtient des gains de performance significatifs. Nous avons mis en place un protocole de test permettant d'étudier les effets du paramètre  $\rho_0$  sur la précision de la prédiction.

Ce test consiste à sélectionner une cible à l'aide du Phantom dans une scène 3D, présentée sur la figure 11. Afin d'éliminer les éventuels effets d'une mauvaise planification du mouvement, l'utilisateur doit avoir une bonne évaluation de la position de la cible, en termes de direction et de distance. Pour cela, nous considérons toujours la même scène : la cible est à la verticale du point de départ et un plan haptique infranchissable est situé au niveau de la cible. Ainsi, l'utilisateur peut rapidement, après quelques essais, avoir une estimation précise du geste à effectuer pour atteindre la cible.

Cinq volontaires ont participé à ce test, effectuant chacun cent sélections. Chaque sélection fournit une courbe de vitesse en fonction de la distance parcourue. En post-traitement, nous étudions ces profils de vitesse. Pour chacun de ces profils, et pour chaque entier  $n$  entre 51 et 99, nous obtenons un point de prédiction obtenu avec un seuil  $\rho_0 = \frac{n}{100}$ . Un seuil de 51% correspond à une prédiction juste après le pic de vitesse, tandis qu'un seuil de 99% implique un mouvement balistique presque terminé. Pour chacune des prédictions obtenues, nous estimons son erreur. Nous définissons l'erreur d'une prédiction comme le ratio  $\frac{d1}{d2}$ , où  $d1$  est la distance entre la prédiction et la cible, et  $d2$  la distance entre le point de départ et la cible. Si l'erreur vaut 0, alors la prédiction est confondue avec la cible. Cette situation est schématisée sur la figure 12.

Enfin, pour chaque valeur de seuil, nous calculons la moyenne des précisions obtenues. Le graphe de la figure 13 montre l'évolution de la précision en fonction de la valeur de seuil, entre 51% et 99%. Nous observons une décroissance rapide de l'erreur entre 51% et 56%, puis un palier entre 56% et 67%, un compor-

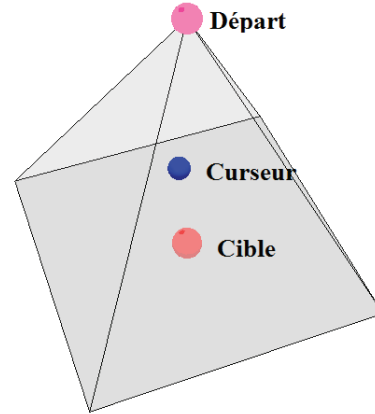


FIGURE 11: Scène 3D pour le test du paramètre  $\rho_0$ . Le curseur de l'utilisateur est en bleu. La position de départ est la sphère rose, et la cible à sélectionner, en rouge, est au centre de la base de la pyramide grise. Un plan haptique infranchissable est placé au niveau de la cible, et confondu avec la base de la pyramide.

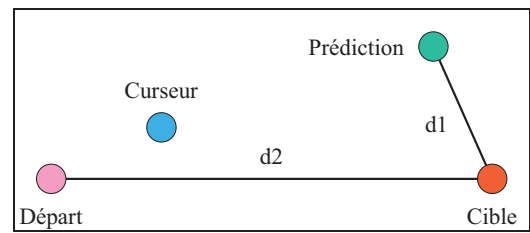


FIGURE 12: Situation au moment où la prédiction est estimée. L'erreur d'une prédiction est définie comme le ratio  $\frac{d1}{d2}$ .

tement linéaire entre 67% et 95%, et enfin l'erreur reste constante (elle vaut alors 9.2%) entre 95% et 99%. Ce comportement est observable chez la plupart des utilisateurs. Cependant, cinq personnes seulement ont réalisé ces tests ; l'allure de cette courbe reste donc encore à préciser. Nous estimons pour le moment qu'une valeur de  $\rho_0$  comprise entre 70% et 80% permet de prédire la cible tôt dans le mouvement, avec une précision satisfaisante (entre 17% et 22%). Nous avons ainsi une précision comparable aux techniques de prédiction développées en 2D.

## 5 CONCLUSION

Dans cet article, nous étudions deux méthodes de prédiction de cibles pour des mouvements de sélection dans une scène en trois dimensions. Notre objectif est de coupler ces méthodes à un aimant, afin de guider l'utilisateur vers sa cible. La première méthode de prédiction, détaillée en section 3, utilise le pic de vitesse du mouvement, et en déduit la distance à laquelle se situe la cible. Une position 3D est ensuite obtenue en considérant la direction du curseur. Un aimant est alors simulé sur le point d'intérêt de la scène le plus proche de cette position. Cette technique d'aide à la sélection est ensuite évaluée dans une scène constituée d'une grille de points. Il s'avère alors que l'algorithme se trompe souvent de cible. Nous expliquons ces erreurs par deux facteurs. La planification du mouvement est faussée par le manque de repères visuels. De plus, l'algorithme de prédiction utilise une relation linéaire entre la vitesse maximale d'un mouvement, et la distance entre le curseur et la cible. Or, pour une même cible, un même utilisateur n'effectuera

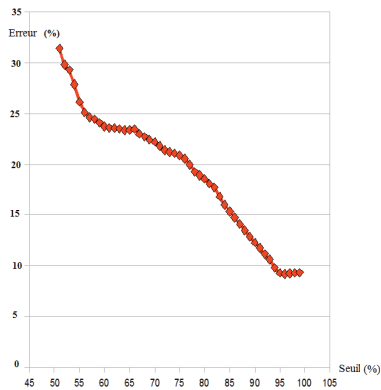


FIGURE 13: Courbe de l'erreur en fonction du seuil  $\rho_0$ . Globalement, plus la distance parcourue est grande, plus précise sera la prédiction.

jamais deux fois le même geste exactement. L'erreur commise par la relation linéaire est alors trop élevée pour obtenir une prédiction suffisamment précise. Cette observation nous a amenés à considérer une seconde méthode de prédiction, basée sur une modélisation du profil de vitesse d'un utilisateur ; la position de la prédiction dépend d'un facteur  $\rho_0$  quantifiant le pourcentage de distance parcourue par rapport à la distance à parcourir pour atteindre la cible. Afin d'atténuer les effets d'une planification faussée du geste, notre protocole de test se place dans le cadre d'une scène simple et répétitive. De plus, un plan haptique permet à l'utilisateur de calibrer son geste après quelques essais. Nous évaluons par ce test l'influence du paramètre  $\rho_0$  sur la précision de la prédiction. Les résultats de ces études semblent indiquer qu'une valeur pour  $\rho_0$  supérieure à 70% du mouvement donne une précision encourageante, comparable aux techniques de prédiction en deux dimensions.

## 6 TRAVAUX FUTURS

Dans un futur proche, nous aimerions compléter les tests présentés en section 4.4 et valider le profil observé jusqu'à présent, quelles que soient la distance et la direction dans lesquelles se trouve la cible. Dans un second temps, nous voulons tester la méthode de prédiction présentée en section 4 dans une application 3D réelle, avec plusieurs cibles potentielles, et couplée à une aide haptique. Ce test nous permettra d'estimer les performances de notre méthode d'aide haptique prédictive. Enfin, l'espace de travail du Phantom est relativement réduit, et nous aimerions tester nos techniques dans un environnement immersif, et utiliser des dispositifs de type SPIDAR (proposé par Hirata[7]), ou INCA 6D (proposé par Dominjon[3]).

## REMERCIEMENTS

Nous tenons à remercier les participants volontaires à tous les tests que nous avons effectué dans le cadre de ces travaux.

## RÉFÉRENCES

- [1] T. Asano, E. Sharlin, Y. Kitamura, K. Takashimai, and F. Kishino. Predictive interaction using the delphian desktop. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 133–141, 2005.
- [2] A. Burel. Aide haptique à la sélection de cibles dans un nuage de points en réalité virtuelle, 2009.
- [3] L. Dominjon, J. Perret, and A. Lecuyer. Novel devices and interaction techniques for human-scale haptics. *The Visual Computer : International Journal of Computer Graphics*, 23(4) :257–266, 2007.
- [4] 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.
- [5] C. Essert-Villard and A. Capobianco. Hardborders : a new haptic approach for selection tasks in 3d menus. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology 2009*, pages 243–244, 2009.
- [6] Y. Guiard, R. Blanch, and M. Beaudouin-Lafon. Object pointing : a complement to bitmap pointing in guis. In *Proceedings of Graphics Interface 2004*, pages 9–16, 2004.
- [7] Y. Hirata and M. Sato. 3-dimensional interface device for virtual work space. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 889–896, 1992.
- [8] N. Hogan. An organizing principle for a class of voluntary movements. *Journal of Neuroscience*, 4(11) :2745–2754, 1984.
- [9] E. Lank, Y.-C. N. Cheng, and J. Ruiz. Endpoint prediction using motion kinematics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 637–646, 2007.
- [10] M. McGuffin and R. Balakrishnan. Acquisition of expanding targets. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 57–64, 2002.
- [11] F. Picon, M. Ammi, and P. Bourdot. Case study of haptic methods for selection on cad models. In *Proceedings of the 2008 IEEE Conference 2008 on Virtual Reality*, pages 209–212, 2008.