



HAL
open science

Aide haptique à la sélection de cibles dans un nuage de points en réalité virtuelle

Jonathan Wonner

► **To cite this version:**

Jonathan Wonner. Aide haptique à la sélection de cibles dans un nuage de points en réalité virtuelle. Synthèse d'image et réalité virtuelle [cs.GR]. 2010. dumas-00530791

HAL Id: dumas-00530791

<https://dumas.ccsd.cnrs.fr/dumas-00530791v1>

Submitted on 29 Oct 2010

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.

Université de Rennes 1
ENS Cachan - Antenne de Bretagne

Stage Master 2 Recherche en Informatique
Équipe IGG
LSIIT, Strasbourg
Vendredi 4 juin 2010

Aide haptique à la sélection de cibles dans un nuage de points en réalité virtuelle

JONATHAN WONNER

encadré par
JÉRÔME GROSJEAN, ANTONIO CAPOBIANCO et
DOMINIQUE BECHMANN



Table des matières

Introduction	4
1 Contexte	5
1.1 Une tâche élémentaire mais complexe	5
1.1.1 En deux dimensions	5
1.1.2 En trois dimensions	6
1.2 L'aide haptique	7
1.3 Périphériques à retour de force	8
1.4 Une domination des aimants	9
1.5 La constriction : naissance des murs haptiques	11
1.6 Cas réel : idée des multi-cellules	13
2 Étude du mouvement	14
2.1 Un premier test	14
2.2 Un profil de vitesse récurrent	15
2.3 Un nuage de points intéressant	16
3 Analyse et réflexion	16
3.1 Idée d'une méthode de prédiction	17
3.2 Réglage des paramètres de prédictions	18
3.3 Évaluation par un protocole	19
4 Protocole de test	20
4.1 Description du test	20
4.1.1 Objectif	20
4.1.2 L'environnement	20
4.1.3 Conditions du test	22
4.1.3.1 Modalités haptiques	22
4.1.3.2 Prédications	24
4.1.3.3 Valeurs numériques	24
4.1.4 Déroulement du test	24
4.1.5 Données enregistrées	26
4.2 Questionnaire	26
4.2.1 Questions	26
4.2.2 Création du questionnaire	27
4.2.3 Fichiers de résultats	27
5 Résultats	28
5.1 Conditions du test	28
5.2 Résultats expérimentaux	29
5.2.1 Temps	30
5.2.2 Distance parcourue	30
5.2.3 Erreur absolue	30
5.2.4 Erreur de prédiction absolue	31
5.2.5 Temps à proximité	32
5.3 Résultats de l'évaluation subjective	32
5.4 Commentaires	33
Conclusions	35

Perspectives	36
Références	38
A Fichier de profile	39
B Analyse statistique	42
B.1 Temps	42
B.2 Distance parcourue	42
B.3 Erreur absolue	43
B.4 Erreur absolue de prédiction	44
B.5 Temps passé proche de la cible	45

Table des figures

1	Le Workbench du LSIIT.	4
2	Un CAVE à 6 murs.	4
3	Un PHANToM Omni.	4
4	Un SPIDAR.	4
5	Expérience de mouvement « va-et-vient ».	6
6	Affichage volumétrique.	7
7	Illustration du problème de la perception de la profondeur.	8
8	Un gant de données.	8
9	Un <i>Flystick</i>	8
10	Force exercée par l'aimant.	10
11	Zones d'influence des aimants triples.	10
12	Réductions successives de la dimensionnalité.	11
13	Un polyèdre convexe à 20 sommets.	12
14	Coupe d'un polyèdre concave.	12
15	Problème du cas multi-cellules.	13
16	L'Univers du premier test.	14
17	Le début d'un fichier de résultats du premier test.	15
18	Un profil de vitesse souvent observé.	16
19	Nuage représentant V_{\max} en fonction de D_{initial}	17
20	La grille avec toutes les cibles visibles.	21
21	Image typique de l'application.	22
22	Activation de 8 cubes. Selon leur position, nous n'affichons pas les mêmes arêtes.	23
23	Le curseur est bloqué en mode Pause	25
24	Une question typique du questionnaire.	27
25	Les questions de comparaison de séquences.	28
26	Notes moyennes extraites du questionnaire.	33

Liste des tableaux

1	Les différentes activations des effets haptiques.	24
2	Le temps moyen par tâche pour chaque modalité (en secondes).	30
3	La distance parcourue moyenne par tâche pour chaque modalité (en millimètres).	30
4	L'erreur absolue par tâche pour chaque modalité (en millimètres).	31

5	L'erreur absolue moyenne d'une prédiction pour chaque modalité (en millimètres).	31
6	La moyenne de temps passé près de la cible par tâche pour chaque modalité (en secondes).	32
7	Les comparaisons par paires des modalités pour le temps passé par tâche.	42
8	Classification pour le temps passé par tâche.	42
9	Les comparaisons par paires des modalités pour la distance parcourue par tâche.	43
10	Classification pour la distance parcourue par tâche.	43
11	Les comparaisons par paires des modalités pour l'erreur absolue par tâche.	43
12	Classification pour l'erreur absolue par tâche.	44
13	Les comparaisons par paires des modalités pour l'erreur absolue par prédiction.	44
14	Classification pour l'erreur absolue par prédiction.	44
15	Les comparaisons par paires des modalités pour le temps passé à proximité de la cible par tâche.	45
16	Classification pour le temps passé à proximité par tâche.	45

Liste des équations

1	Loi de Fitts modifiée.	5
2	Régression linéaire sur le nuage de points.	17

Introduction

La *réalité virtuelle* a pour objectif, pour citer [AFT03], de simuler dans un monde virtuel le comportement d'entités *3D*, qui sont en interaction en temps réel entre elles et avec un ou des utilisateurs en immersion dans ce monde. Si l'immersion d'un tel utilisateur n'est jamais parfaite, l'apparition récente des Workbench (voir la figure 1), ou des CAVE (figure 2), qui permettent de visualiser le monde virtuel en trois dimensions (contrairement à un écran classique), a rendu l'expérience plus crédible. La vue est le sens humain le plus exploité pour donner à l'utilisateur une impression de réalisme du monde virtuel, au moyen de divers périphériques et techniques. De plus, il n'existe, à notre connaissance, pas d'application de réalité virtuelle sans le moindre visuel. L'ouïe est elle-aussi largement employée, dans le sens où des effets sonores sont souvent utilisés dans un environnement immersif.

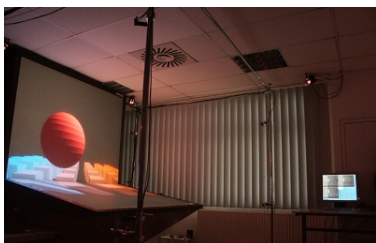


FIGURE 1 – Le Workbench du LSIIIT.

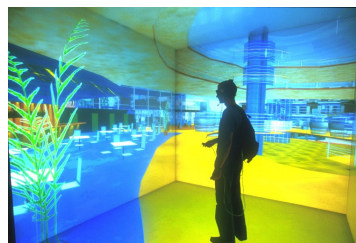


FIGURE 2 – Un CAVE à 6 murs.

En ce qui concerne le toucher, l'intérêt d'un retour haptique est qu'il est plus naturel d'interagir avec des objets *3D* si l'utilisateur peut directement les toucher et les manipuler, en sentant les forces qu'ils renvoient. La création de différents périphériques permettant de se déplacer dans une scène *3D* tout en bénéficiant d'un retour de forces, comme le PHANTOM (figure 3) ou le SPIDAR (figure 4), a ainsi permis l'exploration du sens du toucher.



FIGURE 3 – Un PHANTOM Omni.

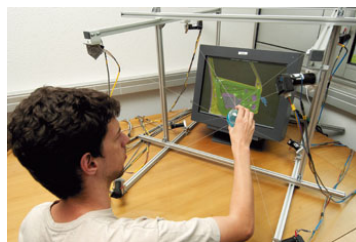


FIGURE 4 – Un SPIDAR.

Dans une application de réalité virtuelle, quel que soit l'objectif de l'utilisateur, l'ensemble de ses actions réalisables est composé de quelques tâches élémentaires. L'une des plus communes est la tâche de sélection. En effet, pour interagir avec des entités du monde, il faut pouvoir être capable, au préalable, de les sélectionner. Cette tâche, très simple dans un espace en deux dimensions, devient complexe en trois dimensions. L'objectif du stage est d'élaborer une technique d'aide à la sélection de cibles dans un espace en trois dimensions, puis de l'implémenter et de la tester. Ce travail prolonge les résultats d'un précédent

travail de stage réalisé au sein de l'équipe **IGG**, effectué par Alexandre Burel en 2009.

Nous commencerons en section 1 par présenter notre problématique et les outils que nous souhaitons utiliser pour la résoudre. Nous donnerons ensuite un aperçu des différentes solutions généralement utilisées. Enfin, nous résumerons les résultats obtenus en 2009 par Alexandre Burel, avant de poser les différentes questions que cela a engendrées. La section 2 présentera le début de mes travaux et de mes tests, commençant par l'étude du mouvement humain lors d'une tâche de sélection, tandis que la section 3 décrira les enseignements que nous avons tirés de cette étude, et la méthode que nous avons développée à partir de ces conclusions. Puis, en section 4, nous présenterons le protocole que nous avons élaboré afin de tester notre méthode. Enfin, en section 5, nous commenterons les différents résultats obtenus pendant ce test.

1 Contexte

Créer un monde virtuel n'a d'intérêt que si un utilisateur immergé dans ce monde peut interagir avec lui, principalement en agissant sur les objets y appartenant. Mais avant de pouvoir interagir avec un de ces objets, il faut pouvoir le sélectionner.

1.1 Une tâche élémentaire mais complexe

La tâche de sélection, si elle est très simple et naturelle dans le monde réel, n'est pas aussi évidente dans un monde virtuel $3D$. Pour en comprendre les raisons, intéressons-nous à cette tâche, en commençant par l'étudier dans un cas simple.

1.1.1 En deux dimensions

En 1954, Paul Fitts ([Fit54]) propose une loi permettant de prédire le temps de désignation pour une tâche de pointage en deux dimensions. Cette loi donne le temps nécessaire pour sélectionner une cible de largeur W située à une distance D :

$$T = a + b \times \log_2\left(\frac{2D}{W}\right)$$

où T est exprimé en secondes, D est l'amplitude du mouvement à effectuer, W est la largeur de la cible à atteindre, et a et b sont des constantes déterminées empiriquement. Dans les expériences de Fitts, le support et la cible à atteindre sont physiques (voir figure 5). En 1978, Card ([CEB78]) montre que le pointage d'une cible virtuelle via un dispositif de pointage (typiquement une souris) suivait aussi la loi de Fitts.

Après plusieurs essais pour améliorer la loi de Fitts, reconnue comme n'étant pas optimale et robuste, celle-ci est légèrement modifiée, pour parvenir à la formulation qui est maintenant la plus utilisée :

$$T = a + b \times \log_2\left(\frac{D}{W} + 1\right) \quad (1)$$

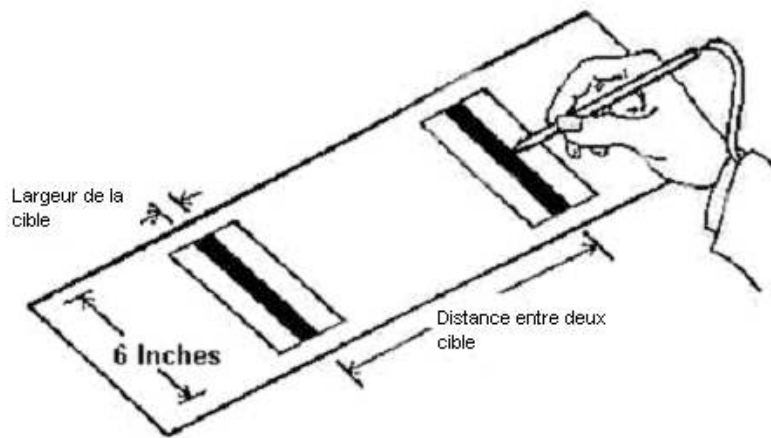


FIGURE 5 – Expérience de mouvement « va-et-vient ».

De manière générale, le cas de deux dimensions ne pose pas de problèmes en termes de perception. Tous les objets étant situés sur le même plan de profondeur, il est simple de sélectionner un objet particulier avec une souris (par exemple). Le cas $3D$ est plus complexe.

1.1.2 En trois dimensions

En 2004, [GR04] propose différents modèles pour étendre la loi de Fitts en trois dimensions. Ces modèles sont comparés au travers de tests utilisant un dispositif d’affichage volumétrique, développé par Actuality Systems¹ (voir la figure 6). Néanmoins, nous nous intéresserons ici uniquement à des scènes $3D$ sur un écran $2D$ classique.

Le problème qui se présente immédiatement lorsqu’on regarde une scène $3D$ fixe sur un écran $2D$ est celui de la distance relative entre les objets. L’utilisateur perçoit difficilement la profondeur et la taille des objets, car même s’il s’agit d’une scène $3D$, il n’en voit qu’une projection en $2D$. Un exemple simple mais néanmoins flagrant est celui présenté en figure 7. Dans cet exemple, nous pouvons émettre plusieurs hypothèses :

- Les trois cubes ont la même taille, mais sont à des distances différentes par rapport à nous.
- Les trois cubes sont de tailles différentes, mais sont à la même distance.
- Le cube du bas est le plus petit, mais le plus proche.
- ...

En regardant simplement l’image, il est impossible de valider l’une ou l’autre des hypothèses. Néanmoins, cela n’empêche pas d’atteindre sa cible. L’utilisateur peut s’aider d’autres repères visuels : le plus souvent, le curseur devient plus petit lorsqu’il s’éloigne, et est caché lorsqu’il passe derrière un objet de la scène. D’autres aides sont possibles selon les applications, comme afficher l’ombre du

1. Actuality Systems : <http://www.actuality-systems.com>

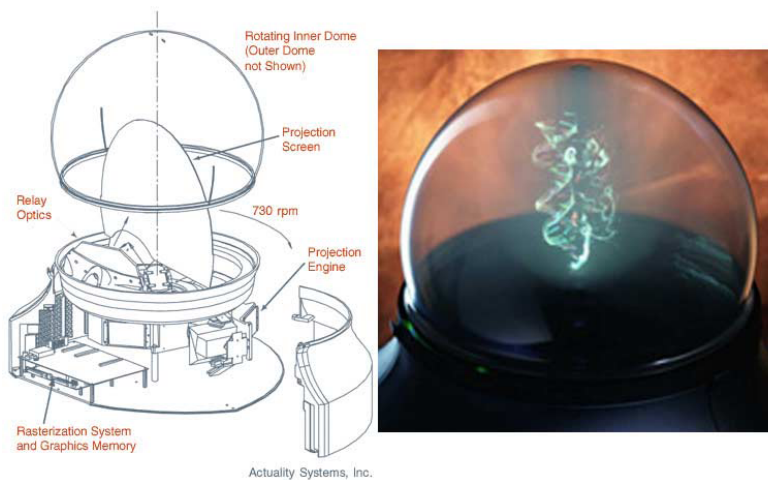


FIGURE 6 – Affichage volumétrique.

curseur sur des plans, comme nous le verrons en section 2.1.

Dans une scène $3D$, la perception incorrecte de la profondeur pose donc des problèmes pour sélectionner les objets. L'utilisation de la stéréoscopie permet d'améliorer la situation. Cette technique permet de reproduire une perception en trois dimensions de la scène affichée sur un écran $2D$. Cela fonctionne en affichant alternativement et à très haute fréquence² la scène du point de vue de l'œil droit, puis du point de vue de l'œil gauche. Des lunettes spéciales permettent d'obtenir chaque œil successivement pour que chacun ne voit que les images qui lui sont destinées. Des capteurs de position et d'orientation peuvent être attachés à ces lunettes pour que le point de vue s'adapte à la position de la tête. L'utilisateur a alors l'impression de pouvoir tourner autour des objets flottant dans l'espace $3D$. Cette technique a néanmoins quelques inconvénients : pour un bon résultat visuel, il faut régler les différents paramètres pour chaque utilisateur, en particulier l'écart interoculaire ; de plus, dans certains cas, une utilisation prolongée peut provoquer une fatigue visuelle et des maux de tête (l'origine de ces troubles est l'objet de plusieurs travaux, par exemple [LWW09]).

Ici, nous recherchons une solution autre que visuelle pour nous aider à sélectionner des cibles. Nous avons décidé de nous tourner du côté de l'haptique.

1.2 L'aide haptique

L'haptique est la science du toucher, comparable à l'optique pour la vue ou l'acoustique pour l'ouïe. L'haptique inclut à la fois le toucher et la proprioception. La proprioception correspond à la perception du corps dans son environnement via des récepteurs et des centres nerveux impliqués dans la perception. On rattache à l'haptique le fait de toucher des surfaces, mais également de res-

². Pour un confort d'utilisation, un minimum de 60Hz pour chaque œil est requis, donc une fréquence globale de 120Hz.



FIGURE 7 – Illustration du problème de la perception de la profondeur.

sentir les textures , la température, *etc.* On distingue en général deux types de perception :

- la perception haptique active, ou tactilo-kinesthésique, qui correspond à la stimulation de la peau par des mouvements actifs d’exploration, en rentrant en contact avec des objets, et
- la perception cutanée, ou passive, qui résulte de la stimulation de la peau par des éléments mobiles, alors qu’on est passif, comme une mouche se posant sur le bras d’une personne.

Durant ce stage, nous nous occuperons seulement de la perception active, car elle est facilement simulée par les différents périphériques à retour de force (voir la section 1.3), et car nous nous intéresserons uniquement à des scènes fixes.

1.3 Périphériques à retour de force

Il existe de nombreux périphériques permettant au programme de repérer la position de l’utilisateur dans l’espace, et à celui-ci de sélectionner des objets virtuels, comme par exemple les gants de données (figure 8) et la manette positionnelle (*flystick*) (figure 9). Munis de différents capteurs, ils permettent de positionner un curseur dans le monde 3D afin de sélectionner des objets et interagir avec eux.



FIGURE 8 – Un gant de données.



FIGURE 9 – Un *Flystick*.

Parmi ces périphériques, certains permettent d'enregistrer les forces fournies par l'utilisateur, et surtout de retourner des forces, afin de le bloquer, de l'attirer vers un endroit précis, voire même de simuler des textures. Parmi les plus courants d'entre eux se trouvent le PHANTOM et le SPIDAR (figures 3 et 4). Le SPIDAR (*Space Interface Device for Artificial Reality*), développé par la société Cyverse Corp³, est composé d'une sphère retenue par huit câbles tendus, chaque câble étant relié à un moteur, permettant ainsi de déplacer la sphère. Le PHANTOM (*Personal HAptic iNterface Mechanism*), développé par la société SensAble Technologies⁴, consiste en un stylet attaché à un bras mécanique. Le stylet permet de déplacer un curseur dans une scène en trois dimensions. Le bras offre six degrés de liberté (trois en translation, trois en rotation) et un retour de force. Le stylet comporte deux boutons (appelés dans la suite « principal » et « secondaire ») afin de permettre plusieurs types d'interactions. À l'aide de la librairie **OpenHaptics**, il est très simple et rapide de créer des effets haptiques complexes avec le PHANTOM. De plus, celui-ci est plus robuste et plus adapté à la sélection que le SPIDAR. C'est pourquoi ce périphérique a été privilégié dans le cadre de ce stage.

1.4 Une domination des aimants

L'apparition des périphériques haptiques date de la fin des années 80 et est donc relativement récente. Dans les années 90, la profusion d'expériences utilisant le PHANTOM ([GM99, SS97]) laisse penser que la création de la société SensAble Technologies, en 1993, a eu un impact très important sur l'étude de l'haptique et de ses applications.

L'une des applications les plus étudiées est l'aide à la sélection, et simuler des aimants est une des premières solutions imaginées, mais aussi l'une des plus efficaces. Cette méthode consiste à considérer les points d'intérêt de la scène comme des aimants : lorsque le curseur de l'utilisateur est suffisamment proche d'un aimant, il est attiré par celui-ci. Cette méthode est largement utilisée dans le cadre de la sélection de points ([PAB08, YTOH02]). Un aimant possède deux paramètres : une distance et une force d'attraction. La distance d'attraction détermine la zone d'influence de l'aimant. Lorsque le curseur pénètre dans cette zone, il est attiré vers l'aimant. La force d'attraction varie en fonction de la distance du curseur à l'aimant (voir figure 10). La force appliquée croît linéairement jusqu'à une certaine distance P , puis redescend linéairement jusqu'à devenir nulle au point aimanté. Cette distance P est définie afin d'améliorer la stabilité du curseur. En effet, si on applique une force maximale au niveau du point aimanté, alors la moindre modification de la position du curseur modifie la direction de la force exercée par l'aimant, et on obtient une vibration du curseur, non désirable. Si les zones d'influence de deux aimants se chevauchent, alors la priorité est mise sur l'aimant le plus proche du curseur⁵.

Une autre solution, pour aider l'utilisateur à sélectionner un point, consiste à aimanter non seulement les points, mais aussi les arêtes les reliant, et les faces

3. Cyverse Corp : <http://www.cyverse.co.jp/eng/index.html>

4. SensAble Technologies : <http://www.sensable.com>

5. **Du côté d'OpenHaptics** : Lorsque nous voulons placer un aimant avec la librairie **OpenHaptics**, il nous suffit d'indiquer la position de l'aimant, la force maximale d'attraction et la distance d'influence. Le calcul de la distance P est laissé à la charge de la librairie.

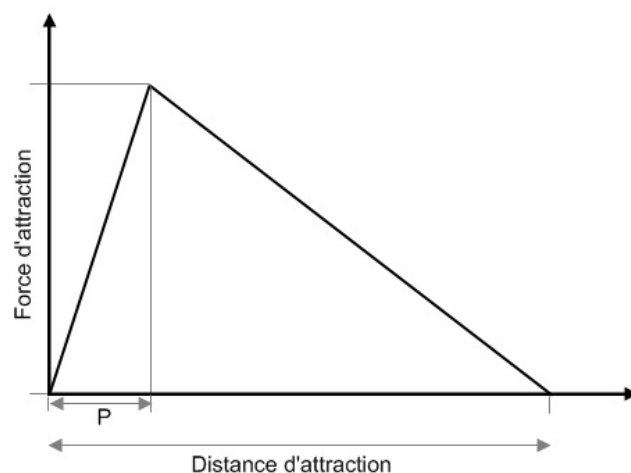


FIGURE 10 – Force exercée par l'aimant.

que forment les arêtes. Nous parlerons alors d'aimants triples. L'utilisateur a la sensation de glisser sur les arêtes ou les faces, pour finalement sélectionner un des points. [YTOH02] utilise cette méthode sur une grille de points régulière. Dans ce cas, les zones d'influence des différents éléments se chevauchent inévitablement (figure 11). Les priorités sont alors déterminées en fonction de leur dimension. Ainsi, un point sera prioritaire sur une arête, elle-même prioritaire sur un plan. Dans le cas de deux éléments de même dimension, nous nous ramenons au cas des aimants simples.

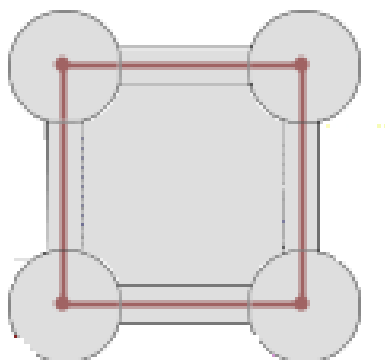


FIGURE 11 – Zones d'influence des aimants triples.

Les aimants permettent d'atteindre rapidement et avec précision la cible, car dès que le curseur est dans sa zone d'influence, il est attiré et retenu précisément à son niveau. Néanmoins, cette qualité peut devenir un défaut : le curseur peut

être attiré par des aimants qui ne sont pas la cible de l'utilisateur. Il lui faudra donc forcer pour s'en libérer, avec le risque d'être immédiatement attiré par un autre aimant proche. Il en résulte une perte de confort et une perte de temps : [PAB08] montre qu'en moyenne, le gain de temps qu'offre les aimants comparés à la sélection libre (sans aide haptique) est faible.

1.5 La constriction : naissance des murs haptiques

La technique que nous appellerons « constriction » consiste à placer des murs haptiques dans la scène. Si ces murs sont placés au même endroit que les faces d'un objet visible de la scène, alors l'utilisateur a vraiment l'impression de les toucher. Cette méthode est utilisée dans [MZ99] pour ajouter du relief à une interface $2D$, pour par exemple ajouter des bordures à un bouton. Son utilisation en $3D$ est mentionnée en [PAB08], mais uniquement dans le cadre de parcours de surface d'objets, et pas dans la sélection. Les tests alors effectués montrent que cette méthode est appréciée des utilisateurs.

En 2009, [Bur09], puis [EVC09], utilisent des polyèdres de constriction comme aides de sélection. Leur idée est de réduire progressivement la dimension du problème, en s'appuyant sur les faces, puis sur les arêtes, pour être guidé jusqu'à la cible. La figure 12 illustre cette idée :

- Le déplacement est d'abord libre à l'intérieur d'un polyèdre (1) (ici, un cube), jusqu'à ce que le curseur atteigne une face (on parle aussi de mur haptique). On passe alors d'un déplacement $3D$ à un déplacement $2D$ sur la face (2).
- Lorsque le curseur rencontre une deuxième face, il peut alors se caler sur l'arête entre les deux faces, et glisser le long de cette arête (3). Le déplacement se fait alors en $1D$.
- Lorsque le curseur rencontre d'autres arêtes, il se cale sur leur sommet commun (4) et s'immobilise.

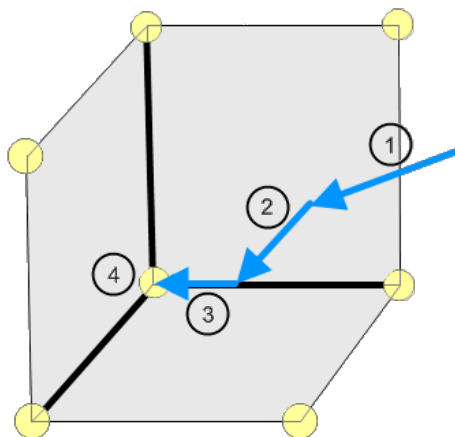


FIGURE 12 – Réductions successives de la dimensionnalité.

Dans ses expériences, [Bur09] considère que les cibles sont situées dans l'espace de telle sorte qu'elles forment un polyèdre convexe (figure 13). Il place alors des murs haptiques sur les faces de ce polyèdre (dont les sommets sont les cibles). L'utilisateur est placé au centre de ce polyèdre (ou cellule), et ne peut donc plus en sortir. Pour sélectionner une des cibles, il suffit alors de s'appuyer et de glisser sur les murs et les arêtes.

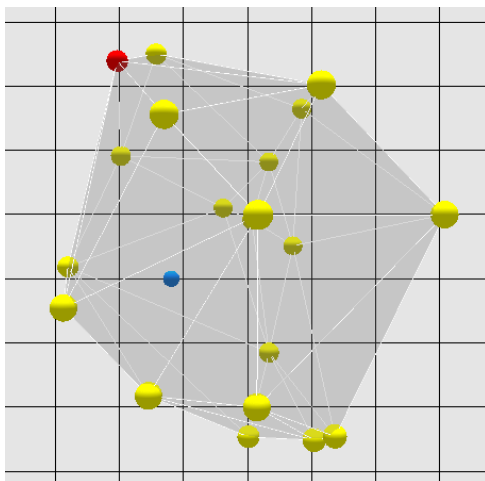


FIGURE 13 – Un polyèdre convexe à 20 sommets.

Le choix d'un polyèdre convexe est primordial. La figure 14 montre l'exemple où cette technique ne peut pas aider et même gêne l'utilisateur dans le cas d'une concavité. On y voit le curseur glisser le long d'une arête, et se diriger vers une cible, mais la concavité l'empêche alors de s'immobiliser.

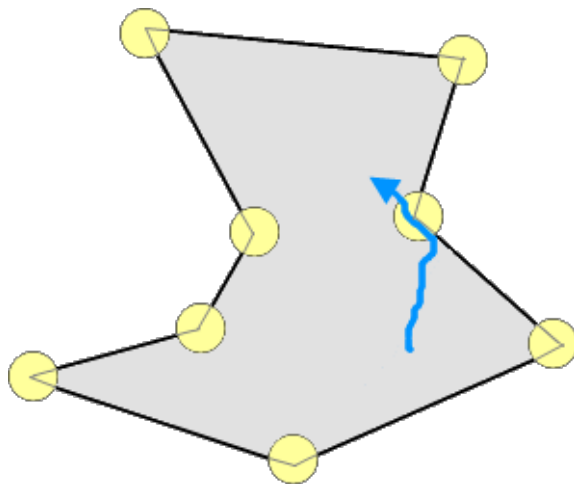


FIGURE 14 – Coupe d'un polyèdre concave.

Ainsi, [Bur09] utilise uniquement des cellules convexes. Or, dans une appli-

cation réelle, les cibles potentielles sont rarement disposées pour créer une forme convexe.

1.6 Cas réel : idée des multi-cellules

Dans le cas général, la disposition des cibles ne permet pas de toutes les englober dans une seule cellule. Un début de solution serait de découper l'espace en plusieurs cellules convexes, les cibles formant les sommets de ces cellules. Si l'utilisateur se trouve dans la même cellule que sa cible, alors il est ramené au cas mono-cellule. Mais un nouveau problème se pose alors : si la cible de l'utilisateur ne se trouve pas dans la même cellule que son curseur, comment peut-il y accéder ? Il « suffirait » alors de désactiver les murs que l'utilisateur veut traverser, pour qu'il puisse changer de cellule. La question est alors de déterminer le moment ou les conditions nécessaires pour que le système désactive ce ou ces murs (figure 15). Il faut différencier le cas où l'utilisateur veut s'appuyer sur une face pour se déplacer dessus, du cas où il veut la traverser.

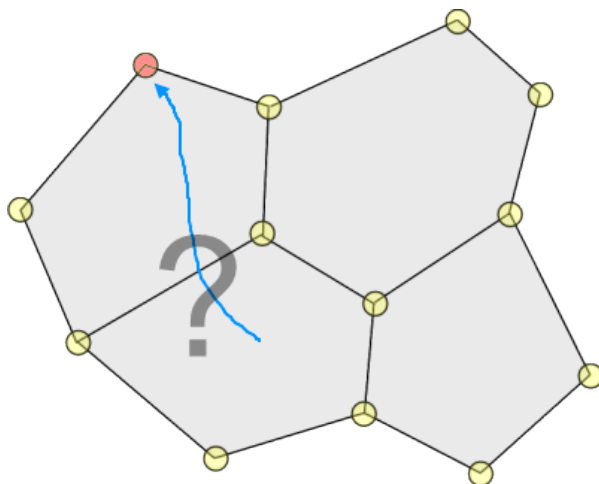


FIGURE 15 – Problème du cas multi-cellules.

Nous pouvons aussi considérer le problème dual : l'utilisateur est placé dans un monde qui au départ ne contient ni cellules, ni murs haptiques. Lorsque l'utilisateur cherche à atteindre une cible, alors le système active une cellule convexe pour le guider uniquement à proximité de la cible. La difficulté est alors de décider le moment et l'endroit où le système doit activer une cellule.

Pour activer une cellule qui aiderait l'utilisateur à atteindre sa cible, il faut pouvoir déterminer à l'avance et rapidement la trajectoire de celui-ci, et ainsi prédire sa position finale. Cette prédiction nous permet alors d'activer un effet haptique au bon endroit. Pour réaliser cette prédiction, nous avons décidé d'étudier le mouvement d'un utilisateur pendant une tâche de sélection. Cette décision est basée sur le fait que nous pouvons décomposer un mouvement de sélection en une phase d'approche, puis une phase d'ajustement ([EHC01]). L'objectif est d'extraire des informations intéressantes pour élaborer

une méthode d'anticipation du point d'arrivée d'un geste de sélection et donc pour choisir quand et où activer l'haptique.

2 Étude du mouvement

Notre objectif est de déterminer une méthode pour prédire la trajectoire de l'utilisateur. Pour cela, nous allons d'abord étudier les caractéristiques d'un mouvement pendant une tâche de sélection. Cette étude a souvent été réalisée en deux dimensions ([HLO09, GGG94]), rarement en trois dimensions ([Led99]), mais jamais, à notre connaissance, avec un PHANTOM.

2.1 Un premier test

Nous avons donc réalisé une première application permettant d'extraire les informations pertinentes. Notre premier monde virtuel contient uniquement le curseur de l'utilisateur, une cible et quelques repères visuels (voir figure 16). Le curseur est une petite sphère bleue que l'utilisateur peut contrôler en trois dimensions avec le PHANTOM. La cible est un petit cube rouge. Pour limiter l'espace de travail, nous avons placé trois murs perpendiculaires gris quadrillés. Ces murs sont uniquement des repères visuels et non pas haptiques. Enfin, les ombres du curseur et de la cible sont projetées sur les trois murs.

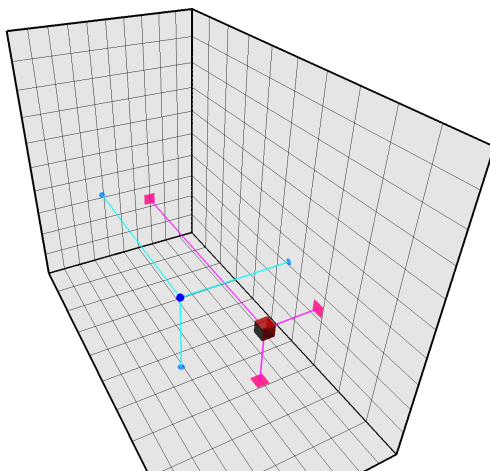
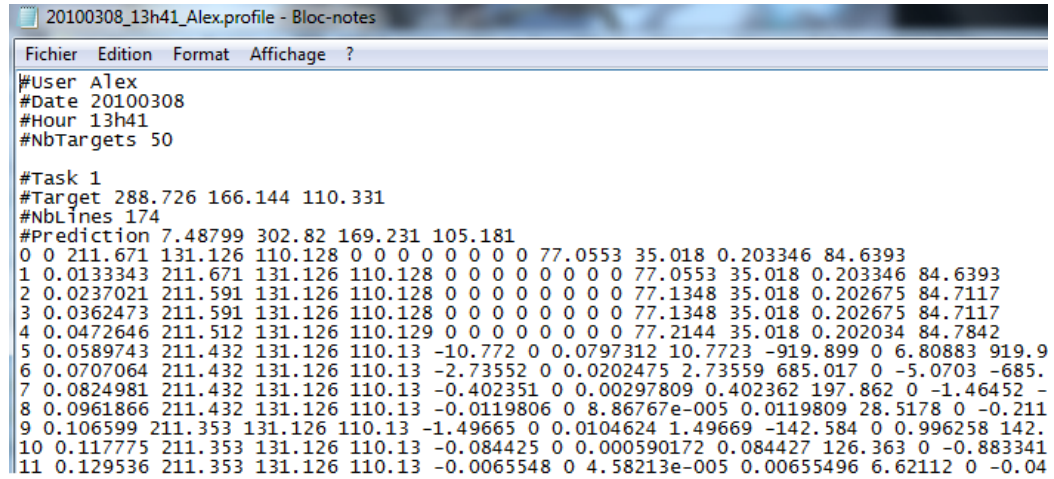


FIGURE 16 – L'Univers du premier test.

Du point de vue de l'utilisateur, il s'agit de réaliser 50 tâches de sélection. Lorsque le curseur est proche de la cible, l'utilisateur peut cliquer sur un des boutons du PHANTOM pour la sélectionner et passer directement à la suivante. La position des cibles est déterminée aléatoirement dans l'espace de travail. À chaque instant, la position et la vitesse du curseur sont enregistrées dans un fichier texte (voir figure 17). Les informations contenues dans ces fichiers texte sont ensuite extraites par un autre programme, qui va nous permettre d'afficher les courbes ou nuages de points à partir de ces données. L'un des résultats sur

lequel nous nous sommes penchés est présenté en section 2.2.



```

20100308_13h41_Alex.profile - Bloc-notes
Fichier Edition Format Affichage ?
#User Alex
#Date 20100308
#Hour 13h41
#NbTargets 50

#Task 1
#Target 288.726 166.144 110.331
#NbLines 174
#Prediction 7.48799 302.82 169.231 105.181
0 0 211.671 131.126 110.128 0 0 0 0 0 0 0 0 0 77.0553 35.018 0.203346 84.6393
1 0.0133343 211.671 131.126 110.128 0 0 0 0 0 0 0 0 0 77.0553 35.018 0.203346 84.6393
2 0.0237021 211.591 131.126 110.128 0 0 0 0 0 0 0 0 0 77.1348 35.018 0.202675 84.7117
3 0.0362473 211.591 131.126 110.128 0 0 0 0 0 0 0 0 0 77.1348 35.018 0.202675 84.7117
4 0.0472646 211.512 131.126 110.129 0 0 0 0 0 0 0 0 0 77.2144 35.018 0.202034 84.7842
5 0.0589743 211.432 131.126 110.13 -10.772 0 0.0797312 10.7723 -919.899 0 6.80883 919.9
6 0.0707064 211.432 131.126 110.13 -2.73552 0 0.0202475 2.73559 685.017 0 -5.0703 -685.
7 0.0824981 211.432 131.126 110.13 -0.402351 0 0.00297809 0.402362 197.862 0 -1.46452 -
8 0.0961866 211.432 131.126 110.13 -0.0119806 0 8.86767e-005 0.0119809 28.5178 0 -0.211
9 0.106599 211.353 131.126 110.13 -1.49665 0 0.0104624 1.49669 -142.584 0 0.996258 142.
10 0.117775 211.353 131.126 110.13 -0.084425 0 0.000590172 0.084427 126.363 0 -0.883341
11 0.129536 211.353 131.126 110.13 -0.0065548 0 4.58213e-005 0.00655496 6.62112 0 -0.04

```

FIGURE 17 – Le début d’un fichier de résultats du premier test.

Spécifications techniques : Cette application, ainsi que les suivantes, est réalisée en C++. Les graphismes et le monde virtuel sont gérés par les bibliothèques **OpenGL** et **GLUT**. Nous avons aussi utilisé **OpenHaptics** pour programmer le PHANTOM et les effets haptiques.

Conditions du test : Dix personnes ont effectué ce premier test. Toutes avaient déjà utilisé plusieurs fois un PHANTOM. La consigne donnée était de sélectionner la cible en essayant de concilier rapidité et précision.

2.2 Un profil de vitesse récurrent

En analysant ces premiers résultats, nous avons rapidement remarqué que pour une tâche donnée, la courbe représentant la vitesse du curseur en fonction du temps avait presque toujours le même profil (figure 18), et qui correspond aux différents résultats obtenus par [Led99, HLO09, GGG94]. La tâche se découpe en 3 parties :

1. la phase de décision, où l'utilisateur observe la scène, repère sa cible et décide du mouvement qu'il va effectuer. Durant cette phase, l'utilisateur est presque immobile (il ne peut pas l'être parfaitement), et sa vitesse est donc presque nulle.
2. la phase balistique, où l'utilisateur augmente sa vitesse pour parcourir rapidement une longue distance, et se rapprocher le plus possible de la cible. Lorsqu'il a atteint la zone où il voulait arriver, il ralentit brusquement. Durant cette phase, la courbe a donc la forme d'un pic.
3. la phase de réajustement, où l'utilisateur corrige sa position pour atteindre finalement sa cible. Durant cette phase, l'utilisateur essaye d'être précis et avance à une vitesse peu élevée.

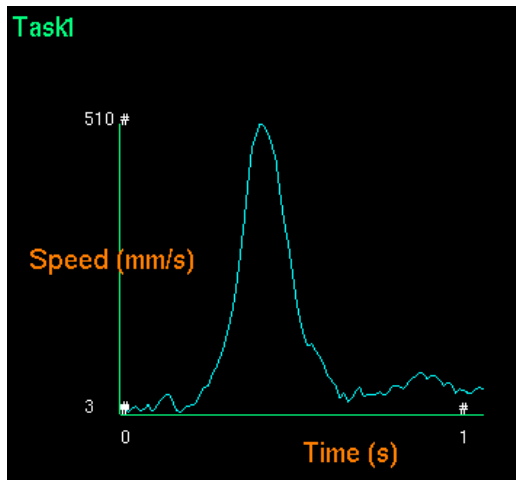


FIGURE 18 – Un profil de vitesse souvent observé.

Les deux dernières parties, la phase balistique et la phase de réajustement, correspondent bien au modèle de Woodworth [EHC01]. Nous nous sommes alors intéressés à la vitesse maximale atteinte durant la phase 2. De quoi dépend-elle ? Est-ce que cette dépendance est la même pour tous les utilisateurs ?

2.3 Un nuage de points intéressant

Une de nos hypothèses, basée sur [SL81], est la suivante : si nous voulons parcourir une longue distance, alors nous allons le faire rapidement, et si, au contraire, nous voulons atteindre une cible proche, alors nous allons nous déplacer plus lentement pour ne pas la dépasser. Nous avons alors tracé le graphe représentant la vitesse maximale atteinte durant la tâche, en fonction de la distance à parcourir (voir la figure 19).

Dans ce nuage de points, un point bleu correspond à une sélection. L'axe des ordonnées indique la vitesse maximale atteinte pendant la tâche. L'axe des abscisses donne la distance initiale (c'est-à-dire la distance au début de la tâche) entre le curseur et la cible. Pour obtenir une formulation simple de la relation entre ces deux grandeurs, nous avons effectué une régression linéaire sur l'ensemble de ces données. De plus, pour tous les utilisateurs, la forme globale du nuage est la même. Nous avons alors décidé d'utiliser ces informations pour réaliser notre prédiction. Cette réflexion est détaillée dans la section 3.

3 Analyse et réflexion

L'étude du mouvement de la section 2 a extrait certaines informations que nous avons jugé intéressantes pour établir un système de prédiction. L'idée de ce système est décrite en 3.1, tandis que ses différents paramètres sont réglés en 3.2. Enfin, l'élaboration d'un protocole de test est abordée en 3.3.

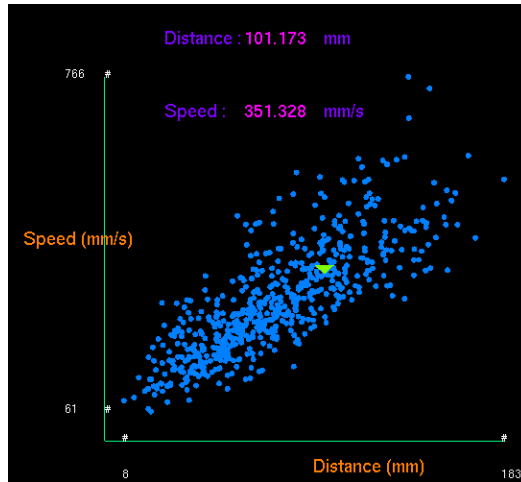


FIGURE 19 – Nuage représentant V_{\max} en fonction de D_{initial} .

3.1 Idée d’une méthode de prédiction

Notre idée est la suivante. Nous avons suggéré qu’il existait une corrélation linéaire des grandeurs présentes dans le nuage de points décrit en 2.3. Nous avons donc calculé une régression linéaire sur ces points pour obtenir l’équation de la droite :

$$V_{\max} = a \cdot D_{\text{initial}} + b, \quad (2)$$

avec V_{\max} la vitesse maximale atteinte durant la tâche, D_{initial} la distance initiale entre le curseur et la cible, et a et b des constantes.

L’idée de notre méthode est ensuite de détecter, durant le déplacement de l’utilisateur, une phase de mouvement balistique et de repérer la vitesse maximale atteinte durant cette phase. Nous utilisons alors notre équation pour en déduire la distance entre l’utilisateur et la cible visée. Il nous suffit enfin de déterminer la direction du geste pour terminer la prédiction et connaître la position visée. Les détails du calcul et de la détection de mouvement balistique sont donnés en section 4.1.3.2.

Récapitulons notre objectif et notre méthode pour l’atteindre. Nous voulons créer un système qui guide l’utilisateur vers sa cible. Pour cela, nous avons élaboré une stratégie permettant au système de prédire la destination finale de l’utilisateur, ou d’en donner une approximation. Quand une telle prédiction est réalisée, le système active une cellule convexe (*ie.* place des murs haptiques sur les faces de cette cellule) dont la cible souhaitée par l’utilisateur est un des sommets. L’utilisateur est alors ramené au cas mono-cellule décrit en section 1.5, et est donc guidé vers sa cible.

Une zone d’ombre reste encore à éclaircir. Nous voulons détecter une phase de mouvement balistique. Il faut donc que le système, à tout moment, détermine si l’utilisateur se dirige vers une cible ou non. Or, nous avons vu en section 2.2

que lorsque l'utilisateur commence à se diriger vers une cible, sa vitesse croit très rapidement jusqu'à atteindre un certain seuil, puis décroît pour amorcer sa phase d'ajustement. Nous avons alors décidé d'utiliser ce profil pour résoudre notre problème.

3.2 Réglage des paramètres de prédictions

Notre première idée était la suivante : si dans son mouvement, l'utilisateur dépasse une certaine vitesse, puis redescend en-dessous d'une autre, alors il s'agit d'un mouvement balistique. Nouveau problème : comment déterminer ces deux seuils ? Doivent-ils être de simples constantes ? Si oui, comment les fixer ? Si non, doivent-ils s'adapter à chaque utilisateur ? Dans un premier temps, nous avons décidé de fixer deux constantes, V_0 et V_1 , en testant expérimentalement les valeurs qui nous semblaient donner une bonne réponse à nos mouvements, c'est-à-dire qui lançaient les calculs de prédiction quand nous nous déplaçons pour atteindre une cible, et ne réagissaient pas autrement (les différentes valeurs numériques utilisées sont données en 4.1.3.3). Ainsi, lorsque la vitesse de l'utilisateur dépasse V_0 , puis redescend en-dessous de V_1 , le système décide qu'il s'agit d'un mouvement balistique. Notons alors t_0 l'instant auquel l'utilisateur franchit le premier seuil, et t_1 l'instant de franchissement du second seuil. La décision de lancer un calcul de prédiction est donc prise à t_1 . Le système récupère alors la valeur de la vitesse maximale atteinte entre t_0 et t_1 . Nous utilisons alors l'équation 2 pour en déduire une distance D_{initial} . Cette distance correspond à la distance initiale (au début du mouvement balistique) entre la position de l'utilisateur et sa cible. Pour trouver la position de notre prédiction dans l'espace, nous avons donc une origine (la position de l'utilisateur à t_0) et une distance (D_{initial}). Il nous manque donc une direction. Nous avons alors observé que lors de la phase balistique, le mouvement est quasiment rectiligne et se dirige vers la cible. Si P_0 est la position de l'utilisateur à l'instant t_0 , et P_1 à l'instant t_1 , alors la direction $\overrightarrow{P_0P_1}$ est la direction manquante. Nous avons ainsi tous les éléments pour calculer la position de notre prédiction.

Cette dernière affirmation n'est valable que si nous supposons connues les constantes a et b de l'équation 2. Pour rappel, ces paramètres sont obtenus par régression linéaire sur le nuage de points donnant la vitesse maximale atteinte pendant une tâche, en fonction de la distance initiale entre l'utilisateur et la cible. Or, chaque nuage ne correspond qu'à un seul utilisateur, et si globalement, les nuages ont tous la même forme, les régressions linéaires ne donnent pas les mêmes équations. Pour régler ce problème, nous avons une fois encore déterminé expérimentalement deux paramètres qui nous paraissaient placer la prédiction à une distance satisfaisante.

Un autre phénomène que nous avons observé est que le profil de vitesse d'un utilisateur est « bruité » : le mouvement de l'utilisateur n'est pas lisse. En conséquence, la vitesse de l'utilisateur peut parfois respecter les conditions pour lancer un calcul de prédiction, alors qu'il ne s'agissait pas d'un mouvement de sélection mais d'un bruit. Le système détectera alors un mouvement balistique là où il n'y en a pas. Pour pallier cet inconvénient, nous avons fixé un pas de temps t_a à respecter : si l'intervalle de temps $t_1 - t_0$ est plus petit que t_a , alors le système n'effectue pas de prédictions, car il s'agit probablement d'un bruit.

Rappelons-nous que nous voulons coupler ce système de prédiction à des murs haptiques pour guider l'utilisateur. Or, que les cellules soient prédéterminées ou définies à la volée, une même cible peut appartenir à plusieurs cellules. Nous pouvons alors nous demander si, une fois la prédiction calculée, le système doit activer une seule de ces cellules, une partie d'entre elles ou toutes. De même, dans le cadre de cette étude, nous souhaitons évaluer l'efficacité des murs haptiques vis-à-vis des solutions généralement dominantes dans les tâches de sélection : les aimants. Nous avons alors élaboré un protocole de test, détaillé en section 3.3.

3.3 Évaluation par un protocole

Notre intention est de créer une application où des utilisateurs ont pour consigne de sélectionner des cibles dans un monde en trois dimensions, aidés en cela par des effets haptiques. L'espace est découpé en cellules convexes, les différentes cibles étant les sommets de ces cellules. Ce test a pour premier objectif de comparer les différents effets qui nous intéressent : murs haptiques sur une seule cellule proche de la cible « prédite » et murs haptiques sur toutes les cellules dont la cible « prédite » est un des sommets⁶.

Pour comparer les murs haptiques avec les aimants dans ce contexte, nous avons décidé d'ajouter trois autres effets haptiques possibles. Le premier consiste à considérer la cible la plus proche de la prédiction comme un aimant⁷. Nous avons choisi cet effet car nous supposons qu'il est d'autant plus efficace en termes de performances que la prédiction est précise et juste. Il nous semblait donc important d'évaluer cet effet. Le second est une variante de l'effet « murs haptiques sur une cellule ». Il s'agit non pas de placer des murs sur les faces de la cellule la plus proche de la prédiction, mais d'en considérer les sommets comme des aimants. Cet effet permet d'avoir une marge d'erreur en comparant avec l'effet « aimant sur une seule cible ». De plus, il a un lien étroit avec l'effet « murs haptiques sur une cellule » car l'haptique est activée dans la même zone (une cellule), mais n'a pas la même dimension (des points pour l'un, des plans pour l'autre). Enfin, le troisième effet est la solution généralement adoptée : toutes les cibles de la scène sont considérées comme des aimants en permanence. Évidemment, cet effet ne dépend pas de notre méthode de prédiction.

Nous souhaitons aussi, à travers ce protocole, tester notre prédiction. Il nous faut donc une autre méthode pour pouvoir les comparer. Celle qui nous a paru la plus parlante à confronter consiste à supposer que dès que l'utilisateur ralentit, alors il est arrivé là où il voulait. Cette méthode utilise seulement des seuils de vitesse : en reprenant les notations de la section 3.2, lorsque l'utilisateur dépasse la vitesse V_0 et ralentit pour passer en-dessous de la vitesse V_1 , en respectant le pas de temps t_d ⁸, alors cette seconde méthode utilise comme « prédiction » la

6. Nous appellerons aussi « activation d'une cellule » le fait de placer des murs haptiques sur les faces de cette cellule.

7. Nous appellerons aussi « activation d'une cible » le fait de simuler un aimant au niveau de cette cible.

8. Ces conditions sont les mêmes que celles qui provoquent le calcul d'une prédiction dans notre première méthode.

position de l'utilisateur au temps t_1 . Les effets haptiques sont alors activés de la même manière que pour notre première méthode, en considérant cette position comme « prédiction ».

Les méthodes de prédiction et les effets haptiques définis ici sont testées dans l'application détaillée en section 4.

4 Protocole de test

Pour vérifier nos hypothèses, nous avons élaboré un protocole de test. Il s'agit d'une application de réalité virtuelle, où plusieurs sujets testent différents effets haptiques et sont interrogés sur leurs appréciations personnelles. Nous commencerons par décrire ce protocole en section 4.1, puis nous détaillerons le questionnaire d'évaluation subjective en section 4.2.

4.1 Description du test

Pour l'utilisateur, l'objectif du test est de sélectionner des cibles fixes apparaissant à l'écran dans un univers en 3 dimensions, en étant le plus rapide et précis possible. L'utilisateur est aidé dans sa tâche par différents effets haptiques, activés selon différentes méthodes de prédiction. Cette section décrit l'environnement virtuel ainsi que les différentes conditions du test.

4.1.1 Objectif

Rappelons que, pour nous, l'objectif est de déterminer quelle modalité haptique est la plus performante en termes de précision et de vitesse, ainsi que la plus confortable pour l'utilisateur. De plus, nous voulons tester notre modèle de prédiction, en le comparant avec un modèle plus simple (vu en 3.3). Au fil des tests, nous enregistrons différentes données et nous recueillons les appréciations des utilisateurs. Toutes ces informations sont analysées en section 5.

4.1.2 L'environnement

Cette section décrit l'environnement virtuel de notre application. Le curseur de l'utilisateur est représenté par une sphère bleue, qu'il est possible de déplacer dans les trois dimensions à l'aide du PHANTOM. Pour la disposition des cibles, nous avons choisi de garder une scène relativement simple, où il est facile pour l'utilisateur de se repérer. Une grille régulière à trois dimensions nous a alors paru un bon choix. Toujours dans cette volonté de garder un test simple, nous avons pris des cubes comme cellules convexes. L'environnement est donc une grille régulière de cubes. De plus, nous voulons tester nos différentes méthodes selon différentes directions et distances. Dans l'optique de ne pas imposer un test trop long pour les utilisateurs, nous avons décidé de leur faire effectuer des tâches de sélection à une distance de 0 (la cible de l'utilisateur est dans la même cellule que sa cible), 1 (la cible est dans une cellule adjacente à celle de l'utilisateur) et 2 (une cellule sépare celle contenant la cible de celle où se trouve l'utilisateur). Il nous fallait donc aligner trois cellules (donc trois cubes) sur les trois directions de la grille. Ainsi, ce test ne s'applique qu'à des tâches de sélection relativement

courtes. Or, dans des applications réelles, l'environnement pourrait ne pas se limiter à des cibles aussi proches les unes des autres. Nous avons donc décidé de représenter le reste de l'environnement possible en prolongeant la grille d'une cellule de chaque côté, les cibles que l'utilisateur doit atteindre dans ce test restant cloisonnées à la grille initiale de $3 \times 3 \times 3$ cubes. Nous utilisons donc finalement une grille de taille $5 \times 5 \times 5$ cubes. Une autre manière de voir la situation est de considérer la grille comme une grille de $6 \times 6 \times 6$ cibles (voir figure 20). Visuellement, les cubes sont représentés par leurs arêtes noirs, et les cibles par des sphères jaunes (la cible à atteindre durant la tâche courante est rouge).

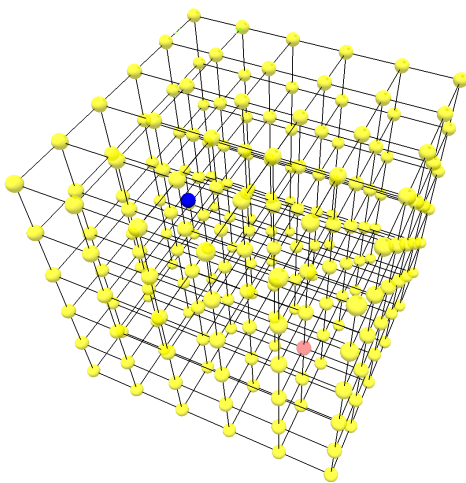


FIGURE 20 – La grille avec toutes les cibles visibles.

Néanmoins, un simple coup d'œil à la figure 20 suffit pour se rendre compte qu'une telle grille est trop complexe visuellement. Même avec beaucoup d'entraînement, il est très difficile d'y repérer la position du curseur et de la cible. De plus, toutes les cellules ne sont pas nécessaires pour chaque tâche. C'est pourquoi nous avons réduit la grille aux cubes indispensables à la tâche (uniquement en termes visuels, voir la figure 21). Ainsi, pour la sélection d'une cible à distance 0, nous n'affichons que le cube contenant l'utilisateur et la cible. Pour des sélections à distance 1 ou 2, nous n'affichons que les trois cubes alignés correspondant à la direction de la sélection (sur la figure 21, il s'agit d'une tâche demandant un mouvement de la droite vers la gauche). À propos des directions possibles dans ces tâches, nous n'avons sélectionné que les directions cardinales : GAUCHE, DROITE, HAUT, BAS, AVANT et ARRIERE. Ce choix est justifié, pour une meilleure compréhension de la disposition des cubes, et pour ne pas trop allonger la durée du test.

Une autre contrainte est la taille de l'espace de travail. Pour que l'utilisateur ne se perde pas dans l'environnement, nous avons décidé de le contraindre dans une zone de l'espace, englobant évidemment les cubes affichés, mais pas toute la grille. En effet, sans aides visuelles, si l'utilisateur sort des cubes affichés et s'en éloigne trop, alors il est très difficile pour lui d'y revenir. Un compromis a été

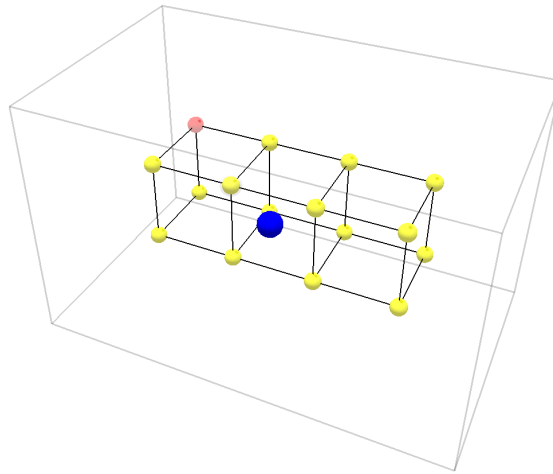


FIGURE 21 – Image typique de l’application.

choisi en plaçant des murs infranchissables proches des cubes affichés (il s’agit des murs dont les arêtes sont représentées en gris sur la figure 21. Ces murs sont placés afin d’englober exactement les cellules affichées et leurs cellules adjacentes. Dans la suite, lorsque nous parlerons de « sommets » ou de « cubes », il s’agira exclusivement des sommets ou cubes contenus à l’intérieur de l’espace de travail (donc dans l’espace défini par les murs infranchissables).

Pour finir avec l’aspect visuel de l’application, nous indiquons en haut à droite de l’écran, le nom de l’effet haptique utilisé dans la tâche de sélection courante (voir la section 4.1.3.1). Enfin, en haut à gauche, nous inscrivons les consignes données à l’utilisateur pour avancer dans le test, comme « *Cliquez sur la cible rouge.* », et le numéro de la tâche courante.

Voyons maintenant à quoi correspondent dans cette application les méthodes et effets définis en section 3.3.

4.1.3 Conditions du test

En 3.3, nous avons décrit les différents effets haptiques et méthodes de prédictions que nous voulons tester et comparer. Ici, nous allons préciser leur comportement dans le cadre de notre application de test.

4.1.3.1 Modalités haptiques

Dans cette application, nous considérons 5 modalités haptiques. À une tâche de sélection correspond une seule modalité. Le moment et l’emplacement de l’activation de l’effet haptique correspondant sont expliqués en section 3.2. Les cinq modalités différentes sont les suivantes :

- **Aimant x1** : Un seul sommet, visible ou non, est activé. Un sommet activé se comporte comme un aimant pour l’utilisateur. Sa zone d’attraction est une sphère de rayon égal au tiers de la longueur d’une arête entre

deux sommets. Ce rayon d'attraction a été choisi pour pouvoir être attiré rapidement par la cible, tout en gardant une grande zone de l'espace libre de toute attraction. Visuellement, une sphère grise semi-transparente englobe le sommet activé.

- **Aimants x8** : Il s'agit d'une extension de la modalité **Aimant x1**. Le système active ici les huit sommets d'un cube, que celui-ci soit entièrement visible ou non (au moins un de ces sommets est visible). Pour chaque sommet activé, le rayon d'attraction et la représentation visuelle sont les mêmes que pour la modalité **Aimant x1**.
- **Bloc x1** : Il s'agit d'une variante de la modalité **Aimants x8**. Le système n'active pas les sommets, mais active un cube, ou plutôt les faces haptiques qui le composent. Une face activée se comporte comme un mur infranchissable lorsque l'utilisateur essaye de sortir du cube, mais n'oppose aucune résistance pour rentrer dans le cube. Visuellement, pour chaque arête d'une face activée, cette arête est dessinée entièrement en marron si les deux sommets à ses extrémités sont visibles, avec un dégradé du marron au blanc transparent si un seul sommet est visible, ou entièrement en blanc transparent si aucun des sommets n'est visible (voir la figure 22, où 8 murs ont été affichés selon différentes positions). Cette décision a été prise afin de ne pas surcharger visuellement la scène.
- **Blocs x8** : Il s'agit d'une extension de la modalité **Bloc x1**. Les huit cubes entourant un des sommets visibles sont activés. Pour chaque cube activé, les paramètres des faces et la représentation visuelle sont les mêmes que pour la modalité **Bloc x1**.
- **Full Aimants** : Tous les sommets sont activés. Il s'agit du seul effet dont l'activation est permanente.

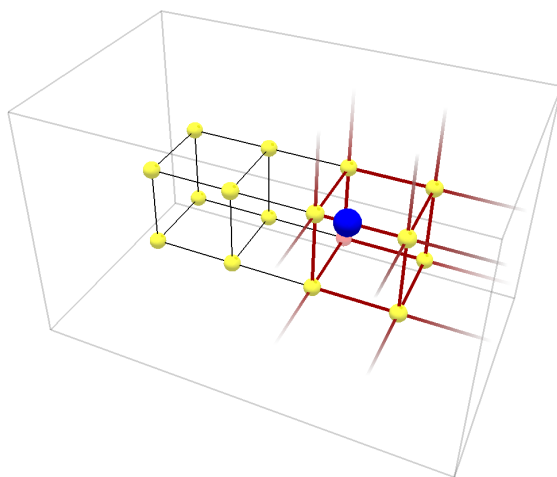


FIGURE 22 – Activation de 8 cubes. Selon leur position, nous n'affichons pas les mêmes arêtes.

4.1.3.2 Prédiction

Comme vu en section 3.3, nous voulons comparer deux méthodes de prédiction. À une tâche donnée, nous attribuons une seule de ces méthodes. Reprenons les notations de 3.2 et notons S_{near} le sommet le plus proche de la prédiction (quand le système en produit une), et C_{near} le cube le plus proche d'elle (la distance utilisée est la distance euclidienne classique, et la distance entre un point de l'espace et un cube est la distance entre ce point et le centre de ce cube). La table 1 décrit les effets de l'activation des différentes modalités haptiques, quand le système produit une prédiction.

Aimant x1 :	Le sommet S_{near} est activé, avec les conséquences décrites en section 4.1.3.1.
Aimants x8 :	Les huit sommets composant le cube C_{near} sont activés.
Bloc x1 :	Le cube C_{near} est activé.
Blocs x8 :	Les huit cubes entourant le sommet S_{near} sont activés.
Full Aimants :	Comme précisé en section 4.1.3.1, cette modalité ne dépend d'aucune prédiction : tous les sommets sont activés en permanence.

TABLE 1 – Les différentes activations des effets haptiques.

Lorsque le système fait une nouvelle prédiction, il désactive au préalable les sommets ou cubes précédemment activés. Dans le cas des cubes, afin d'éviter de rester bloqué dans son élan, le système désactive aussi les murs haptiques lorsque l'utilisateur, après avoir dépassé le seuil V_0 , dépasse un autre seuil V_2 supérieur à V_0 .

4.1.3.3 Valeurs numériques

Dans notre protocole, nous avons utilisé les valeurs numériques suivantes :

$$\begin{aligned}
 V_0 \text{ (murs)} &= 100.0 \text{ mm.s}^{-1} \\
 V_0 \text{ (aimants)} &= 40.0 \text{ mm.s}^{-1} \\
 V_1 &= 30.0 \text{ mm.s}^{-1} \\
 V_2 \text{ (murs)} &= 200.0 \text{ mm.s}^{-1} \\
 a &= 5.0 \text{ s}^{-1} \\
 b &= 70.4 \text{ mm.s}^{-1} \\
 t_d &= 0.1 \text{ s}
 \end{aligned}$$

Ces valeurs ont été choisies empiriquement au cours d'une phase de tests préliminaires visant à calibrer les différents effets haptiques et les méthodes de prédiction. Le choix de prendre deux V_0 différents selon les modalités tient à notre évaluation personnelle. Choisir une seule valeur commune favorisait trop l'une ou l'autre des modalités.

4.1.4 Déroulement du test

Initialement, l'utilisateur se trouve en mode **Pause**. Ce mode intervient au début de chaque tâche. Un aimant, représenté par une sphère rouge semi-transparente, bloque l'utilisateur au niveau de sa position de départ pour sa prochaine tâche (voir figure 23). Un des sommets visibles s'illumine (*ie.* devient rouge). Il s'agit de la nouvelle cible à sélectionner. À tout moment, l'utilisateur peut se libérer du mode **Pause** pour débiter sa tâche en cliquant sur le bouton

secondaire du PHANTOM. L'aimant qui le retenait disparaît. L'utilisateur doit se diriger vers la cible et la sélectionner en cliquant dessus avec le bouton principal du PHANTOM. À ce moment, le mode **Pause** revient et ainsi de suite. Pour chaque modalité haptique à l'exception de la modalité **Full Aimants**, il y a 2 séquences de 32 tâches consécutives à effectuer. La séquence *A* correspond au premier calcul de prédiction vu en section 4.1.3.2, et la séquence *B* au second, vu en 3.3. Pour chaque séquence, les tâches sont réparties de la façon suivante :

- 8 tâches correspondent aux 8 sommets du cube central. Seul ce cube est affiché, et l'utilisateur commence sa tâche au milieu du cube.
- Pour chacune des six directions principales, il y a 4 tâches. Par exemple, pour la direction GAUCHE, trois cubes sont affichés : le cube central, celui à sa gauche et celui à sa droite. L'utilisateur est placé au centre du cube de droite. Parmi les 8 sommets du cube de gauche, le système sélectionne aléatoirement 2 sommets sur sa face gauche, et 2 sur sa face droite, qui vont correspondre à 4 tâches différentes.
- Le système sélectionne de la même manière 4 tâches pour chacune des directions DROITE, HAUT, BAS, AVANT et ARRIERE.

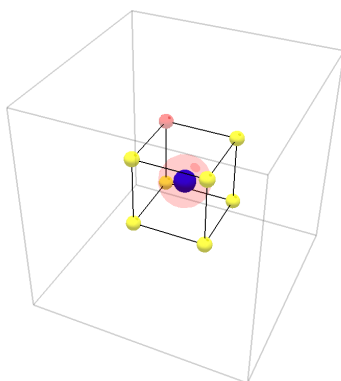


FIGURE 23 – Le curseur est bloqué en mode **Pause**.

Pour la modalité **Full Aimants**, celle-ci ne dépendant pas d'un calcul de prédiction, l'utilisateur ne doit en effectuer qu'une seule séquence. Nous arrivons donc à un total de $32 \times 9 = 288$ sélections.

L'ordre des modalités à tester est :

Aimant x1, Aimants x8, Bloc x1, Blocs x8, Full Aimants,

et toutes ses permutations circulaires. Les deux séquences d'une même modalité sont effectuées l'une juste après l'autre, en commençant par la séquence *A* pour deux des modalités, et par la séquence *B* pour les deux autres. Dans une séquence, les 32 tâches sont mélangées aléatoirement. À la fin de chaque séquence, l'utilisateur est invité à répondre à différentes questions d'un questionnaire décrit en section 4.2.

4.1.5 Données enregistrées

À chaque instant, nous sauvegardons la position et la vitesse du curseur du PHANTOM. À la fin des 288 tâches, nous enregistrons ces données dans un fichier *.profile*. Les informations que nous extrayons ensuite de ces fichiers sont détaillées en 5.2. En Annexe A, nous détaillons l’aspect programmation de cette sauvegarde et donnons l’exemple d’un fichier *.profile* réduit.

4.2 Questionnaire

Afin de connaître le ressenti de l’utilisateur, nous avons élaboré un questionnaire électronique d’évaluation subjective, auquel il doit répondre au fur et à mesure de l’avancement du test.

4.2.1 Questions

Pour chaque séquence, cinq questions sont posées à l’utilisateur, sous la forme d’une phrase affirmative. Ce sont les mêmes questions pour chaque modalité, seul évidemment le nom de la modalité change. Pour chaque question, 7 boutons radio, notés de 0 à 6, permettent à l’utilisateur d’apprécier l’intitulé de la question. Comme le sous-entend la barre de couleurs dégradés du rouge au vert (voir figure 24), la note 0 signifie que l’utilisateur n’est pas du tout d’accord avec l’intitulé, et la note 6 signifie qu’il est tout à fait d’accord. En dessous de chaque question, une zone de texte permet à l’utilisateur de noter des commentaires ou autres appréciations personnelles qui lui semblent pertinents. Les questions sont les suivantes :


- *La modalité XXX - Séquence Y est intuitive.* Nous demandons ici si l’utilisateur arrive à bien prendre en main la modalité, et s’il comprend ce qui se passe, en termes d’effets haptiques.
- *La modalité XXX - Séquence Y est agréable.* Nous demandons ici si la modalité est confortable, ou si elle n’est pas trop fatigante, physiquement ou mentalement.
- *La modalité XXX - Séquence Y me paraît adaptée à la tâche.* La tâche, ici, est la sélection d’une cible parmi plusieurs points. Nous demandons ici si l’aide fournie par l’effet haptique a été jugée efficace ou non.
- *La modalité XXX - Séquence Y permet de sélectionner rapidement la cible.* Nous demandons ici si l’utilisateur a senti que sa performance était rapide ou non.
- *La modalité XXX - Séquence Y permet de sélectionner avec précision la cible.* Nous demandons ici si l’utilisateur a senti qu’il arrivait à atteindre sa cible précisément.

Lorsque les deux séquences d’une même modalité ont été réalisées, deux autres questions sont alors posées (voir la figure 25) :

- *Pour la modalité XXX, on note des différences entre la Séquence A et la Séquence B.* L’utilisateur a le choix entre deux boutons radio, l’un **Oui**, l’autre **Non**. Il s’agit simplement de savoir s’il a ressenti des différences. Il peut alors éventuellement indiquer les différences dans la zone de texte en dessous.
- *Pour la modalité XXX, j’ai préféré . . .* Trois boutons radio sont présents : **Séquence A**, **Séquence B** et **Aucune**.

La modalit  Aimant x1 - S quence A est intuitive.

0
 1
 2
 3
 4
 5
 6



Remarque :

FIGURE 24 – Une question typique du questionnaire.

Enfin,   la fin du questionnaire, une note globale entre 0 et 6 est demand e pour chaque s quence de chaque modalit . Pour enregistrer les donn es recueillies dans ce questionnaire, un bouton **Enregistrer** est situ  en bas de celui-ci. Lorsque l'utilisateur clique dessus, deux situations peuvent se pr senter.

- Si l'utilisateur a oubli  de r pondre   une question, une boite de dialogue s'ouvre et lui indique quelle question n'a pas  t  compl t e.
- Sinon, le syst me cr e un nouveau fichier et y inscrit les r ponses dans le format indiqu  en section 4.2.3.

4.2.2 Cr ation du questionnaire

L'utilisateur r pond   certaines questions   la fin de chaque s quence. Or, l'ordre des s quences n'est pas fixe selon les utilisateurs, comme pr cis  en 4.1.4. Afin de ne pas perturber l'utilisateur, les s quences dans le questionnaire doivent se trouver dans le m me ordre que dans le test. Nous avons donc cr e un programme en **C++** r cup rant le nom de l'utilisateur, ainsi que l'ordre de passage des s quences, et cr ant ensuite le questionnaire ad quat au format **html**.

4.2.3 Fichiers de r sultats

Lorsque l'utilisateur appuie sur le bouton **Enregistrer** du questionnaire, si toutes les questions ont  t  remplies, alors un programme **Javascript** cr e un fichier *.result* au nom de l'utilisateur. Un tel fichier a la structure suivante :

- Les neuf premi res lignes sont chacune compos es de cinq entiers compris entre 0 et 6 (inclus), correspondant aux notes attribu es   chaque question pour les s quences dans l'ordre suivant :
 - **Aimant x1 - S quence A**, puis **S quence B**,
 - **Aimants x8 - S quence A**, puis **S quence B**,
 - **Bloc x1 - S quence A**, puis **S quence B**,
 - **Blocs x8 - S quence A**, puis **S quence B**, et enfin
 - **Full Aimants**.
- Le fichier comprend ensuite une ligne vide, puis quatre lignes de deux entiers. Le premier de ces entiers vaut 1 si l'utilisateur a not  des diff rences entre les deux s quences d'une m me modalit , ou 0 sinon. Le deuxi me

Comparaison Aimant x1

Pour la modalité Aimant x1, on note des différences entre la Séquence A et la Séquence B.

Oui Non

Remarque :

Pour la modalité Aimant x1, j'ai préféré ...

Séquence A Séquence B Aucune

Remarque :

FIGURE 25 – Les questions de comparaison de séquences.

vaut 1 si l'utilisateur a préféré la séquence A, 2 s'il a préféré la séquence B, ou 0 s'il n'a préféré aucune des deux séquences. S'ensuit une ligne vide.

- Enfin, les neuf dernières lignes, contenant chacune un entier entre 0 et 6 (inclus), donne les notes globales attribuées à chaque modalité.

Nous avons donc plusieurs types de données à analyser : les données recueillies pendant le déplacement de l'utilisateur, et les différentes notes et remarques personnelles du questionnaire. Ces données sont traitées respectivement en sections 5.2 et 5.3.

5 Résultats

Les différents résultats obtenus durant ces tests vont nous permettre de valider ou non nos modèles. La section 5.1 donne les différentes conditions dans lesquelles se sont déroulées les expériences. Les sections 5.2 et 5.3 analysent les différents résultats obtenus. Enfin, la section 5.4 regroupe les remarques intéressantes que nous avons recueillies, et les observations que nous avons pu faire.

5.1 Conditions du test

Quinze personnes ont réalisé cette expérience. Toutes ont déjà manipulé le PHANToM au moins une fois auparavant. Le test commence par un entraînement. Il s'agit de la même application, mais moins longue, au cours de laquelle aucune donnée n'est enregistrée. Il s'agit de présenter les différentes

modalités à l'aide de quelques tâches (5 pour chaque modalité). Les consignes données à l'utilisateur sont les suivantes :

- « L'objectif du test est de sélectionner la cible rouge. »
- « À différents instants, le système essaye de prédire votre trajectoire, et active différents effets haptiques en conséquence pour vous aider à sélectionner la cible. »
- « Pour chaque modalité, il y a deux séquences de 32 tâches à effectuer. Les deux séquences correspondent à deux méthodes de prédiction. Ces deux méthodes sont basées sur votre vitesse. »
- « L'important est que votre curseur soit sur la cible, que l'effet haptique soit activé ou non à cet endroit. »

Le test, entraînement et questionnaire compris, dure environ 30 minutes.

5.2 Résultats expérimentaux

Parmi nos fichiers *.profile* décrits en section 4.1.5, nous avons extrait cinq caractéristiques qui nous semblaient intéressantes à analyser :

- Temps : le temps passé pour chaque tâche.
- Distance parcourue : la distance parcourue par le curseur dans l'environnement pendant toute la durée de la tâche. Cette distance est calculée en cumulant les distances entre deux positions successives de l'utilisateur. Cette mesure, inspirée par [MKS01], a pour objectif de quantifier l'effort réalisé par l'utilisateur pour atteindre la cible et constitue un indicateur de performance supplémentaire.
- Erreur absolue : la distance entre la position finale de l'utilisateur (*ie.* au moment du clic) et la cible.
- Erreur de prédiction absolue : la distance entre une prédiction et la cible.
- Temps à proximité : le temps passé par l'utilisateur à proximité de la cible. Nous avons défini le seuil de distance de proximité comme étant le tiers de la distance entre deux sommets adjacents, ce qui correspond à la zone d'attraction d'un aimant.

Nous utiliserons dans la suite les notations suivantes :

- **A1A** : Aimant $\times 1$, séquence *A*
- **A1B** : Aimant $\times 1$, séquence *B*
- **A8A** : Aimants $\times 8$, séquence *A*
- **A8B** : Aimants $\times 8$, séquence *B*
- **B1A** : Bloc $\times 1$, séquence *A*
- **B1B** : Bloc $\times 1$, séquence *B*
- **B8A** : Blocs $\times 8$, séquence *A*
- **B8B** : Blocs $\times 8$, séquence *B*
- **FA** : Full Aimants

Nous avons réalisé une analyse statistique des résultats sur les données récoltées, en effectuant une **ANOVA** (**AN**alysis **Of** **VA**riance) à mesures répétées, le facteur indépendant étant la modalité haptique utilisée. Nous effectuons ensuite une comparaison pair à pair en utilisant les tests HSD de Tukey. Nous présentons en annexe B les données issues des analyses statistiques. Nous y présentons également les regroupements significatifs entre modalités.

5.2.1 Temps

La technique a une influence significative sur le temps passé pour chaque tâche ($F(8, 109) = 15.17, p < 0.001$). La table 2 regroupe la moyenne des temps passés par tâche pour chaque modalité. La table 7 fournie en annexe B nous indique que **A1B** est significativement plus rapide que les autres modalités ($p < 0.05$), sauf pour **FA** et **A8B** ($p > 0.10$) avec qui il n’y a pas de différences. À l’opposé, **B1A** a tendance à être moins rapide que **B8A** et **B1B** ($0.05 < p < 0.1$) et significativement moins rapide que les autres modalités. Plus généralement, toutes les techniques utilisant des murs haptiques s’avèrent significativement plus lentes que les techniques avec des aimants. Ce résultat était attendu : les aimants sont reconnus pour être rapide. Lorsque l’utilisateur entre dans leur zone d’attraction, ils sont instantanément dirigés vers la cible. L’avantage de **A1B** résulte du fait qu’avec un seul aimant activé à chaque fois, il y a moins de risques d’être attiré par un mauvais aimant.

A1B	A8B	FA	A8A	A1A	B8B	B8A	B1B	B1A
1.52	1.61	1.68	1.86	2.18	2.65	2.75	2.89	2.94

TABLE 2 – Le temps moyen par tâche pour chaque modalité (en secondes).

5.2.2 Distance parcourue

La technique a une influence significative sur la distance parcourue pour chaque tâche ($F(8, 109) = 13.29, p < 0.001$). La table 3 regroupe la moyenne des distances parcourues par tâche pour chaque modalité. Cette table nous indique qu’en moyenne, la modalité qui permet d’atteindre sa cible en parcourant le moins de distance est **A1B**. De plus, la table 9 nous informe que **A1B** a tendance à requérir moins de distance que **A8A**, et en demande significativement moins que les autres modalités. À l’autre extrême, **B8B** demande la même distance que **B8A**, a tendance à en demander plus que **B1A**, et en demande significativement plus que les autres modalités. De même que pour le critère « Temps », les modalités utilisant des murs haptiques sont significativement moins performantes que les modalités avec aimants : les premières demandent à parcourir une plus grande distance que les secondes. Ce résultat s’explique aisément : pour utiliser les murs, il faut d’abord se poser dessus, puis glisser vers un sommet. Pour un aimant, l’utilisateur est directement attiré vers le sommet, d’où un gain de distance.

A1B	A8B	FA	A8A	A1A	B8B	B8A	B1B	B1A
75.73	79.35	85.30	87.36	92.70	113.81	116.87	117.18	125.40

TABLE 3 – La distance parcourue moyenne par tâche pour chaque modalité (en millimètres).

5.2.3 Erreur absolue

La technique a une influence significative sur l’erreur absolue à la fin de chaque tâche ($F(8, 109) = 3.18, p = 0.0028$). La table 4 regroupe la moyenne

de l'erreurs absolue pour chaque modalité. Un utilisateur commet moins d'erreur en moyenne avec **FA**, ce qui n'est pas surprenant car les aimants doivent bloquer l'utilisateur sur la cible précisément. Les autres modalités avec aimants obtiennent des résultats variés, à cause d'une mauvaise prédiction qui va parfois désactiver les aimants au moment du clic. Plus étonnant, **B8A** obtient de bons résultats. Ceci peut s'expliquer par le fait que si la cible est bien le sommet commun aux 8 blocs haptiques, alors la présence des murs tout autour de ce sommet permet d'être très précis. Les faibles résultats de **B1B** et **B1A** sont la conséquence des changements trop fréquents dans l'activation des blocs.

FA	A8B	B8A	A1B	A8A	B8B	A1A	B1B	B1A
0.20	0.31	0.53	0.57	0.60	0.93	0.94	1.26	1.44

TABLE 4 – L'erreur absolue par tâche pour chaque modalité (en millimètres).

5.2.4 Erreur de prédiction absolue

Nous voulons évaluer l'efficacité de la prédiction (celle décrite en 3.1, utilisant la régression linéaire) en fonction de l'effet haptique utilisé. D'après les résultats statistiques de l'**ANOVA**, l'effet haptique a une influence significative sur l'erreur absolue d'une prédiction ($F(8, 4) = 88.24$, $p < 0.001$). La table 5 regroupe la moyenne des erreurs absolues par prédiction pour chaque modalité. La première remarque que nous pouvons faire est que globalement, quelle que soit la modalité utilisée, la prédiction est assez mauvaise (le meilleur résultat est une moyenne de 17.45mm pour **A1A**, alors que la taille d'une arête d'un cube dans la scène est 20mm. Comme l'indique la table 14, la prédiction donne significativement les pires résultats pour **FA**. Ceci peut s'expliquer facilement : lorsque tous les aimants sont activés en permanence, l'utilisateur est souvent attiré involontairement par de mauvaises cibles. Cette attraction augmente instantanément la vitesse du curseur, et le système pense donc qu'il s'agit d'un geste balistique et calcule une prédiction, qui par conséquent est très mauvaise et n'est même que rarement dans la direction de la vraie cible. Néanmoins, cette baisse de performances n'a évidemment aucune incidence dans notre application, car avec **FA**, l'activation des aimants ne dépend pas des prédictions. Plus généralement, plus l'effet est contraignant, plus une mauvaise prédiction sera gênante pour l'utilisateur. La seule solution pour lui est alors d'« envoyer » la prédiction loin de lui par un à-coup. La nouvelle prédiction qui en résulte ne provient pas d'un mouvement naturel et est généralement très mauvaise. Par conséquent, la distance moyenne entre une prédiction et la cible est relativement élevée. Les effets haptiques les plus contraignants sont les deux effets avec 8 blocs (en pratique, les utilisateurs se sont souvent retrouvés complètement coincés car entourés de murs), et cela se reflète sur les performances pour ce critère.

A1A	B1A	B1B	A1B	A8A	A8B	B8A	B8B	FA
17.45	18.84	20.00	23.10	23.24	25.14	25.29	26.11	29.96

TABLE 5 – L'erreur absolue moyenne d'une prédiction pour chaque modalité (en millimètres).

5.2.5 Temps à proximité

La technique a une influence significative sur le temps passé à proximité de la cible ($F(8, 109) = 22.502, p < 0.001$). La table 6 regroupe la moyenne de temps passé près de la cible par tâche pour chaque modalité. Sans surprise, la plupart des modalités avec aimants obtiennent de meilleurs résultats que les modalités avec blocs, car il suffit que la cible soit effectivement activée et que l'utilisateur entre dans la zone d'attraction, pour qu'il soit attiré précisément sur la cible et qu'il décide de cliquer. Le mauvais résultat de **A1A** est dû à une mauvaise prédiction : l'utilisateur est effectivement proche de la bonne cible, mais l'unique sommet activé n'est pas le bon à cause d'une mauvaise prédiction. Il n'arrive alors pas à se décider de cliquer tant que la bonne cible n'est pas activée. De manière générale, les utilisateurs ont tendance à ne pas cliquer sur la cible tant qu'une aide haptique n'est pas présente, et contrairement à ce que préconisait l'une des consignes. Ce comportement vient de leur volonté de cliquer précisément sur la cible, ce qui pour eux implique une aide haptique. En effet, sans cette aide, ils pensent ne pas être assez précis. Ce phénomène explique aussi, comme indiqué dans les tables 16 et 15 en annexe B, que **FA** obtient significativement de meilleurs résultats par rapport aux autres modalités (sauf pour la modalité **A1B** avec laquelle on ne relève qu'une tendance statistique à être meilleure ($p = 0.064$)). En effet, toutes les cibles étant activées en permanence, la décision demandée à l'utilisateur est plus simple à résoudre : quand il arrive près de la cible, il est attiré vers elle, et donc il sait qu'il est précisément sur la cible. Il n'a pas à « attendre » que l'effet haptique soit bien placé.

FA	A1B	A8B	A8A	B8B	A1A	B8A	B1A	B1B
0.57	0.66	0.66	0.83	1.06	1.08	1.15	1.46	1.49

TABLE 6 – La moyenne de temps passé près de la cible par tâche pour chaque modalité (en secondes).

5.3 Résultats de l'évaluation subjective

Comme indiqué dans la section 4.2, nous avons évalué 5 critères. Les utilisateurs ont évalué si chaque modalité étaient intuitive, agréable, adaptée à la tâche, rapide et précise. Enfin, une note globale était demandée. La figure 26 montre les moyennes des notes obtenues pour ces questions pour tous les testeurs réunis. Chaque modalité, sauf **Full Aimants**, est séparée en deux colonnes *A* et *B*, correspondant aux deux séquences. La colonne *A* est notre méthode de prédiction avec régression linéaire ; la colonne *B* correspond à la méthode décrite en 3.3 utilisant uniquement des seuils de vitesse.

La première observation que nous pouvons faire est une nette préférence pour les aimants sur les murs haptiques : sur tous les critères, les 5 séquences faisant intervenir des aimants surpassent les 4 séquences avec les blocs.

En ce qui concerne la comparaison entre les deux méthodes de prédiction, il apparaît que la méthode *B* obtient presque tout le temps une bien meilleure note que la méthode *A*. Nous concluons de cette différence de notes que les paramètres que nous avons fixés pour la prédiction, s'ils donnaient de bons résultats expérimentalement pour les « pré-testeurs » (2 personnes connaissant bien la technique), n'étaient pas adaptés à tous. Pour la modalité **Blocs x8**, les deux méthodes ne se distinguent presque pas, et ce sur tous les critères. Ceci

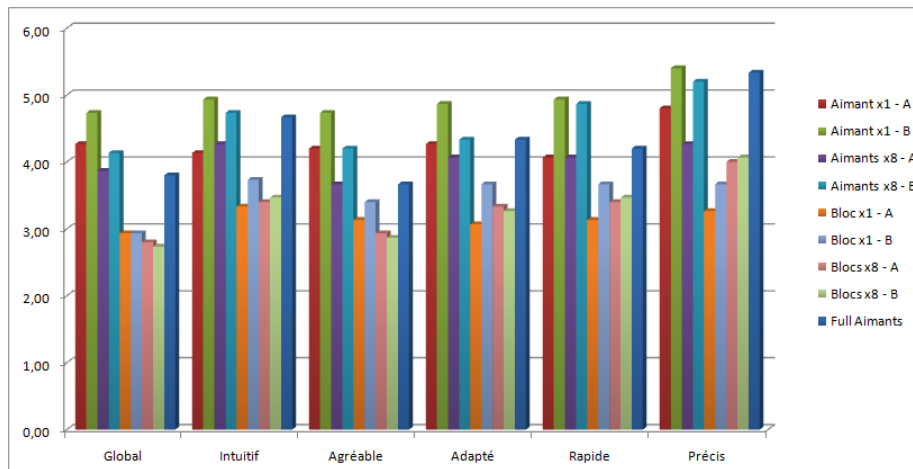


FIGURE 26 – Notes moyennes extraites du questionnaire.

s’explique par le fait que la zone d’effet de cette aide est très grande comparée aux autres modalités. Ainsi, la différence de position entre les deux prédictions devient négligeable : l’utilisateur est malgré tout susceptible d’être guidé, même si la prédiction n’est pas bonne.

La perception qu’a l’utilisateur sur ses performances n’est pas toujours conforme à la réalité. Ainsi, une mauvaise note (4.0/6.0, 7^{ème} rang) est attribuée à la précision de **B8A (Blocs x8 - Séquence A)**, alors que cette modalité obtient de bons résultats (0.53mm d’erreur en moyenne, 3^{ème} rang) sur les performances réelles. Une raison à cela est qu’un utilisateur a tendance à attribuer une meilleure note à une modalité qu’il aura appréciée, et ce quelle que soit la question.

5.4 Commentaires

L’objectif initial était d’établir un système permettant à l’utilisateur de réaliser un geste naturel de sélection, et de s’apercevoir de la présence de l’haptique uniquement à la fin du mouvement. En pratique, dans le cas des blocs, il s’est avéré qu’après un temps d’adaptation, le geste n’est plus naturel : il commence alors par un à-coup, pour « lancer » l’haptique vers la cible, puis finit par un mouvement vers la cible rapide mais néanmoins plus lent que pour un geste naturel. Ce comportement est instinctif et difficile à corriger. Presque tous les utilisateurs ont fini par l’adopter, sans qu’il leur soit suggéré. Si cet à-coup n’est, au début, pas toujours efficace, il le devient de plus en plus au cours du test.

Les résultats que nous obtenons sont contradictoires avec nos espérances et les résultats de [Bur09, EVC09]. Les murs haptiques n’ont pas obtenu de bonnes performances, et n’ont pas été particulièrement appréciés par les utilisateurs. Une cause plausible de ces résultats est la disposition des cibles. En effet, l’idée de créer une grille régulière très simple est justifiée par notre volonté de minimiser les erreurs de planification de l’utilisateur. Plus l’environnement est compréhensible, moins l’utilisateur fera d’erreurs dans sa trajectoire. De plus,

nous avons décidé de mettre peu d'indices visuels, car nous cherchions à valider notre système de prédiction et nos différentes aides haptiques, et non pas à expérimenter différentes aides visuelles permettant de mieux planifier le mouvement. Néanmoins, il s'est avéré que ces choix nous ont conduit à une situation favorable pour les différents aimants, en particulier pour la modalité **Full Aimants**. En effet, une grille régulière, orientée selon les directions cardinales de l'environnement, permet facilement d'éviter les aimants non souhaités, et donc d'aller rapidement et précisément sur la bonne cible. Cela s'est traduit par le fait que, contrairement à nos attentes, les utilisateurs se sont peu retrouvés dans le cas de figure où le pointeur est « capturé » par un sommet différent de leur cible.

Comme indiqué en 5.2.5, la plupart des utilisateurs nous ont confié qu'ils n'arrivaient pas à se décider à cliquer sur la cible si l'aide haptique n'est pas présente. Ainsi, même s'ils arrivaient sur la cible, si un des aimants (par exemple) n'était pas activé à cet endroit, ils préféraient tourner autour de la cible ou effectuer des mouvements d'aller-retour pour tenter d'activer le bon sommet. Ce comportement est contradictoire avec l'une des consignes énoncées en 5.1, mais est difficile à corriger.

Conclusions

Dans ce rapport, nous avons commencé par introduire une méthode d'aide haptique pour faciliter la sélection de cibles dans un nuage de points en trois dimensions, basé sur des polyèdres de constriction. Cette méthode n'avait été utilisée que dans le cas mono-cellule, et les utilisateurs en avaient retiré une opinion favorable, dans un domaine où les aimants prédominent. Nous avons alors présenté le processus qui nous a conduit à une méthode étendant les polyèdres de constriction au cas multi-cellules. Cette méthode consiste à se ramener au cas mono-cellule, en prédisant la trajectoire de l'utilisateur, et donc la position de sa cible, pour ne placer qu'une seule cellule dans la zone la plus susceptible d'aider l'utilisateur. Nous avons ensuite détaillé le protocole de test qui nous a permis de tester notre nouvelle méthode d'aide, et de la comparer aux techniques existantes. Enfin, nous avons présenté les différents résultats, subjectifs et « réels », que nous avons extraits de ce test.

Nous pouvons retenir de ces résultats que les performances des murs haptiques dans ce contexte ne sont pas à la hauteur des espérances laissées par le cas mono-cellule. La cause de cela n'est pas la technique des murs en elle-même, mais plutôt l'évaluation des différents paramètres gérant les calculs de prédiction et l'activation des différents effets. Nous avons fixé ces paramètres expérimentalement, dans un souci de confort d'utilisation et de performances, en supposant que la phase d'entraînement du test suffirait au testeur final pour s'approprier la technique et la maîtriser correctement. Dans les faits, il en résulte des prédictions qui ne correspondent pas aux intentions de l'utilisateur, et des activations/désactivations d'effets haptiques qui le perturbent plus qu'elles ne le guident. De plus, le choix d'un environnement trop simple (une grille régulière) pour une meilleure perception de la disposition de la scène, favorise les effets à base d'aimants : l'utilisateur peut facilement éviter les sommets, mais pas les faces des cellules.

Perspectives

Une manière de prendre en compte les résultats obtenus est de dire que toute technique nécessite un minimum d'apprentissage, et que, par conséquent, les utilisateurs réussiront au bout d'un certain temps à utiliser notre méthode basée sur le couple prédiction/murs haptiques, avec les paramètres fixés actuellement. Une seconde manière consiste à considérer que ce n'est pas l'utilisateur qui doit s'adapter à la technique, mais le contraire. Ainsi, pour étendre ces travaux, nous pouvons améliorer la méthode de prédiction pour qu'elle s'adapte à l'utilisateur, en déterminant les différents paramètres par une phase de calibration, ou en les corrigeant au fil du temps.

Certains utilisateurs nous ont suggéré de tester la possibilité pour le testeur d'activer manuellement un effet haptique (par exemple en appuyant sur une touche ou par un clic de souris). Si cette condition n'a pas été retenue pour le protocole de test car nous voulions une technique d'activation automatique, il peut être intéressant de la tester, seule ou en la combinant avec une méthode de prédiction.

Comme nous l'avons déjà précisé, l'environnement virtuel utilisé pendant ce test était trop simple. Pour pouvoir complexifier la scène tout en permettant à l'utilisateur de se repérer, il peut être intéressant d'utiliser des systèmes de stéréoscopie, malgré les inconvénients présentés en 1.1.2, voire de porter l'application à l'échelle humaine en la testant sur un Workbench, pour une meilleure immersion de l'utilisateur. Ainsi, les aimants ne seraient plus aussi favorisés.

Complexifier la scène pose d'autres questions :

- Comment devons-nous générer l'environnement ? En utilisant des cibles aléatoires ? Des cas provenant d'applications réelles ?
- Comment répartir les cibles en cellules ? Devons-nous les pré-découper avant de lancer l'application ? Si oui, de quelle manière ? Si non, devons-nous calculer une cellule « optimale » à la volée ?
- Dans certaines applications, en particulier les jeux, les cibles ne sont pas fixes mais mobiles, et leur mouvement dépendent même parfois des déplacements de l'utilisateur (ou du joueur). Pouvons-nous étendre nos techniques à ce cas ?

Références

- [AFT03] B. Arnaldi, P. Fuchs, and J. Tisseau. *Traité de la réalité virtuelle*, volume 1, chapter 1. Les Presses de l'École des Mines de Paris, 2003.
- [Bur09] Alexandre Burel. Aide haptique à la sélection de cibles dans un nuage de points en réalité virtuelle. Master's thesis, Université de Strasbourg, 2009.
- [CEB78] S. Card, W. English, and B. Burr. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a crt. *Ergonomics*, 21 :601–613, 1978.
- [EHC01] Digby Elliott, Werner F. Helsen, and Romea Chua. A century later : Woodworth's (1899) two-component model of goal-directed aiming. *Psychological bulletin*, 127(3) :342–357, 2001.
- [EVC09] Caroline Essert-Villard and Antonio 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.
- [Fit54] P. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47 :381–391, 1954.
- [GGG94] James Gordon, Maria Felice Ghilardi, and Claude Ghez. Accuracy of planar reaching movements. *Experimental Brain Research*, 99(1) :97–111, 1994.
- [GM99] Chris Gunn and Paul Marando. Experiments on the haptic rendering of constraints : Guiding the user. *Proceedings of Advanced Simulation Technology and Training Conference*, 1999.
- [GR04] T. Grossman and R. Balakrishnan. Pointing at trivariate targets in 3d environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 447–454, 2004.
- [HLO09] Joel C. Huegel, Andrew J. Lynch, and Marcia K. O'Malley. Validation of a smooth movement model for a human reaching task. In *Conference on Rehabilitation Robotics*, 2009.
- [Led99] Nicolas Leduc. Authentification de signatures à l'aide du modèle delta log-normal en 3d. Rapport, 1999.
- [LWW09] Y. Liu, F. Wang, and C. Wang. The research on visual fatigue factor in stereoscopic image observation. In *Conference on Computer-Aided Industrial Design and Conceptual Design*, 2009.
- [MKS01] I. Scott MacKenzie, Tatu Kauppinen, and Miika Silfverberg. Accuracy measures for evaluating computer pointing devices. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 9–16, 2001.
- [MZ99] Timothy Miller and Robert Zeleznik. The design of 3d haptic widgets. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 97–102, 1999.
- [PAB08] Flavien Picon, Mehdi Ammi, and Patrick 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, mars 2008.

- [SL81] J.F. Soechting and F. Lacquaniti. Invariant characteristics of a pointing movement in man. *The Journal of Neuroscience*, 1(7) :710–720, 1981.
- [SS97] J. Kenneth Salisbury and Mandayam A. Srinivasan. Phantom-based haptic interaction with virtual objects. *IEEE Computer Graphics and Applications*, 17 :6–10, 1997.
- [YTOH02] Toshio Yamada, Daisuke Tsubouchi, Tetsuro Ogi, and Michitaka Hirose. Desk-sized immersive workplace using force feedback grid interface. In *Proceedings of the IEEE Virtual Reality Conference 2002*, page 135, 2002.

A Fichier de profile

Les données du PHANTOM sont recueillies à chaque instant ; en pratique, nous les obtenons toutes les dixièmes de seconde environ. La position du curseur est donnée par la fonction *hlGetDoublev* avec pour arguments la constante *HL_PROXY_POSITION* et un tableau de trois *doubles*. Sa vitesse est donnée par la fonction *hlGetFloatv* avec pour arguments la constante *HD_CURRENT_VELOCITY* et un tableau de trois *floats*. À la fin des 288 tâches, on enregistre ces données dans un fichier *.profile*.

La structure d'un tel fichier est le suivant :

<i>Fichier .profile</i>	:=	<i>Entête</i> <i>Tâche</i> ₁ ... <i>Tâche</i> ₂₈₈
<i>Entête</i>	:=	#User nom_utilisateur #Date date_test #Hour heure_test #NbTargets 288 #TotalTime temps_total #GridParameters dix_paramètres #PredictionParameters sept_paramètres #CameraParameters six_paramètres
<i>Tâche</i>	:=	#Pause temps_pause #Task numero_tâche #HapticMode identifiant_effet_haptique #Target trois_paramètres #NbLines nombre_données #NbPredictions nombre_predictions <i>Ligne</i> ₁ ... <i>Ligne</i> _{nombre_données+nombre_predictions}
<i>Ligne</i>	:=	neuf_paramètres OU #Prediction quatre_paramètres

Quelques points méritent des éclaircissements. Les dix paramètres de la grille correspondent à diverses variables de configuration de la scène, et sont les mêmes pour tous les utilisateurs. Le premier correspond à la taille des sommets, les trois suivants sont le nombre de sommets selon les trois axes principaux, les trois suivants sont les coordonnées minimales d'un sommet, et les trois derniers sont les coordonnées maximales. Dans notre protocole, nous avons les paramètres suivants :

#GridParameters 2 6 6 6 -50 -25 -30 50 75 70

Les huit paramètres de prédiction sont les différents seuils et variables gérant les prédictions. Le premier est le V_0 défini dans la section 4.1.3.2, pour l'activation des cubes. Le second est V_0 pour l'activation des sommets. Le troisième est V_1 , valable pour les cubes et les sommets. Le quatrième est le seuil V_2 de désactivation des murs. Le cinquième est le pas de temps t_d . Les deux der-

niers sont les coefficients de la régression linéaire utilisé pour un des calculs de prédiction. Dans notre protocole, nous avons :

```
#PredictionParameters 100 40 30 200 0.1 5 70.4017
```

Les six paramètres de caméra sont les coordonnées des deux points intéressants pour la caméra : sa position, et l'endroit où elle regarde. Dans notre protocole, nous avons :

```
#CameraParameters 61 136 163 -24 1 -17
```

Les trois paramètres pour #Target sont les coordonnées de la cible associée à la tâche.

Pour les données correspondant aux *Lignes*, deux cas se présentent :

- soit il s'agit d'une prédiction, dans ce cas, nous enregistrons l'instant auquel le système a calculé la prédiction (*ie.* le moment écoulé depuis le début de la tâche), ainsi que les coordonnées de celle-ci,
- soit il s'agit des données du PHANTOM, dans ce cas, nous donnons l'index de la ligne, l'instant, les trois coordonnées de la position de l'utilisateur, les trois coordonnées de vitesse v_x , v_y et v_z selon les trois axes, ainsi que sa « vraie » vitesse ($\sqrt{v_x^2 + v_y^2 + v_z^2}$).

Exemple réduit :

```
#User Alex_ocampo
#Date 20100507
#Hour 10h30
#NbTargets 288
#TotalTime 1926.47
#GridParameters 2 6 6 6 -50 -25 -30 50 75 70
#PredictionParameters 100 40 30 200 0.1 5 70.4017
#CameraParameters 61 136 163 -24 1 -17

#Pause 18.506

#Task 0
#HapticMode 2
#Target 10 15 10
#NbLines 182
#NbPredictions 1
0 0 -19.9595 25.0554 20.0714 1.90427 17.2869 -17.3403 24.5591
1 0.0116882 -20.2097 21.0009 15.6805 1.87585 19.6129 -17.0815 26.0761
2 0.0234127 -20.1799 21.1176 15.4094 4.38671 -11.1148 -39.9456 41.6945
3 0.0351849 -20.1326 20.9351 14.9782 3.90381 -14.2427 -35.5482 38.4938
4 0.0477734 -20.0808 20.7165 14.5063 2.32245 -14.4167 -21.1483 25.6999
5 0.0587006 -20.0589 20.4572 14.3069 2.53207 -24.7095 -23.0571 33.8911
6 0.0704689 -20.0369 20.1985 14.1069 1.2802 -30.4905 -11.6576 32.6682
7 0.0822372 -20.0147 19.8656 13.9049 2.32264 -35.9973 -21.15 41.8153
8 0.0939941 -19.9928 19.6827 13.7052 1.14183 -12.8369 -10.3976 16.559
9 0.105766 -19.9752 19.5368 13.5451 0.496265 -16.835 -4.51901 17.438
10 0.117901 -20.0322 19.3165 13.375 0.841405 -7.6321 -7.63295 10.8268
11 0.129297 -20.0233 19.2438 13.2947 0.178227 -1.54012 -1.61682 2.24005
12 0.141054 -20.0233 19.2438 13.2947 0.0496509 -12.7584 -0.450417 12.7665
13 0.152824 -20.0231 19.1693 13.2931 0.00147842 -0.379899 -0.0134118 0.380138
14 0.164583 -20.0231 19.1693 13.2931 6.09436e-005 -0.0155729 -0.00055286 0.0155828
15 0.17802 -20.0274 19.1311 13.3316 4.73169e-006 -0.00120909 -4.29243e-005 0.00120986
16 0.18815 -20.0274 19.1311 13.3316 1.81468e-006 -0.000463704 -1.64621e-005 0.000464
17 0.199883 -20.0274 19.1311 13.3316 12.6316 -2.61571e-005 1.38716 12.7075
18 0.211697 -19.9527 19.1311 13.3398 0.517687 -1.07201e-006 0.0568509 0.520799
19 0.223417 -19.878 19.1311 13.348 0.0761505 -1.57676e-007 0.00829925 0.0766014

...

206 2.42584 10.0559 35.009 50.0231 0.408807 -1.61637 -1.00628 1.9474
207 2.43872 10.0559 35.009 50.0231 0.0167544 -0.0662448 -0.0412408 0.079811
208 2.44962 10.0559 35.0092 50.0233 -15.2533 1.97223 -0.845669 15.4035
209 2.46115 10.0558 35.0095 50.0237 -3.38734 1.5498 -6.82401 7.77451
210 2.4729 10.0559 35.0091 50.0233 -0.878442 -10.4236 -15.5737 18.7607
211 2.48468 10.056 35.009 50.0233 -0.476361 -3.22269 -10.3673 10.8671
212 2.49643 10.0561 35.0087 50.023 -0.309151 -8.27852 -6.79836 10.7167
213 2.50873 10.057 35.0072 50.0226 -0.0861968 -2.30617 -1.89551 2.98644

#Pause 0.552887

#Task 67
#HapticMode 10
#Target -10 15 30
#NbLines 135
#NbPredictions 4
0 0 0.0121985 25.0222 0.0326623 17.5816 24.1617 -34.0798 45.3248
1 0.0112305 0.299197 25.1236 13.5551 1.85251 8.97058 -17.8715 20.0822
#Prediction 0.0112305 11.3111 36.2774 54.7095
2 0.0240479 0.30023 25.7249 14.1831 0.199373 28.263 31.9963 42.6919
3 0.0343628 0.225125 25.6128 14.1451 -10.2506 -2.22362 -2.01332 10.6805
4 0.0464172 0.25392 14.2255 -24.3971 -15.9361 17.0904 33.7826
5 0.0579224 -0.300629 25.3552 14.4256 -38.5753 -16.1645 17.3788 45.2921
6 0.0696411 -0.751735 25.3183 14.4643 -40.1752 -11.4557 12.1513 43.5079

...
```

B Analyse statistique

Cette annexe regroupe les différents tableaux de résultats obtenus par **ANOVA**, décrit en 5.2.

B.1 Temps

L'analyse comparative montre que la modalité haptique utilisée a une influence significative sur le temps passé pour chaque tâche ($F(8, 109) = 15.17$, $p < 0.001$). Le tableau 7 regroupe les résultats de la comparaison par paires.

	A8A	A8B	A1A	A1B	B8A	B8B	B1A	B1B
A8B	0.0242	-	-	-	-	-	-	-
A1A	0.0043	$3.3e^{-7}$	-	-	-	-	-	-
A1B	0.0020	0.4007	$2.9e^{-9}$	-	-	-	-	-
B8A	$2.2e^{-15}$	$< 2e^{-16}$	$3.6e^{-7}$	$< 2e^{-16}$	-	-	-	-
B8B	$1.6e^{-12}$	$< 2e^{-16}$	$2.5e^{-5}$	$< 2e^{-16}$	0.3812	-	-	-
B1A	$< 2e^{-16}$	$< 2e^{-16}$	$1.7e^{-11}$	$< 2e^{-16}$	0.0990	0.0116	-	-
B1B	$< 2e^{-16}$	$< 2e^{-16}$	$2.2e^{-10}$	$< 2e^{-16}$	0.2056	0.0323	0.7007	-
FA	0.1043	0.5285	$7.6e^{-6}$	0.1414	$< 2e^{-16}$	$< 2e^{-16}$	$< 2e^{-16}$	$< 2e^{-16}$

TABLE 7 – Les comparaisons par paires des modalités pour le temps passé par tâche.

Ces statistiques permettent de classer les modalités de la manière indiquée par la table 8.

Modalité	Moyenne (s)						
A1B	1.52	A					
A8B	1.61	A					
FA	1.68	A	B				
A8A	1.86		B				
A1A	2.18			C			
B8B	2.65				D		
B8A	2.75				D	E	
B1B	2.89					E	F
B1A	2.94						F

TABLE 8 – Classification pour le temps passé par tâche.

B.2 Distance parcourue

L'analyse comparative montre que la modalité haptique utilisée a une influence significative sur la distance parcourue pour chaque tâche ($F(8, 109) = 13.29$, $p < 0.001$). Le tableau 9 regroupe les résultats de la comparaison par paires.

	A8A	A8B	A1A	A1B	B8A	B8B	B1A	B1B
A8B	0.2497	-	-	-	-	-	-	-
A1A	0.1525	0.0099	-	-	-	-	-	-
A1B	0.0639	0.4828	0.0010	-	-	-	-	-
B8A	$7.5e^{-10}$	$2.9e^{-13}$	$2.3e^{-6}$	$1.3e^{-15}$	-	-	-	-
B8B	$1.1e^{-14}$	$< 2e^{-16}$	$2.8e^{-10}$	$< 2e^{-16}$	0.1120	-	-	-
B1A	$1.1e^{-9}$	$4.5e^{-13}$	$3.0e^{-6}$	$2.1e^{-15}$	0.9525	0.0992	-	-
B1B	$3.6e^{-8}$	$2.9e^{-11}$	$4.5e^{-5}$	$2.0e^{-13}$	0.5145	0.0250	0.5536	-
FA	0.6909	0.1215	0.3014	0.0244	$8.4e^{-9}$	$2.2e^{-13}$	$1.2e^{-8}$	$3.2e^{-7}$

TABLE 9 – Les comparaisons par paires des modalités pour la distance parcourue par tâche.

Ces statistiques permettent de classier les modalités de la manière indiquée par la table 10.

Modalité	Moyenne (mm)						
A1B	75.73	A					
A8B	79.35	A	B				
A8A	85.30		B	C			
FA	87.36		B	C			
A1A	92.70			C			
B1B	113.81				D		
B1A	116.87				D		
B1A	117.18				D	E	
B1B	125.40					E	

TABLE 10 – Classification pour la distance parcourue par tâche.

B.3 Erreur absolue

L'analyse comparative montre que la modalité haptique utilisée a une influence significative sur l'erreur absolue à la fin de chaque tâche (quand l'utilisateur clique sur le bouton principal du PHANTOM ($F(8, 126) = 3.18, p < 0.01$). Le tableau 11 regroupe les résultats de la comparaison par paires.

	A8A	A8B	A1A	A1B	B8A	B8B	B1A	B1B
A8B	0.10903	-	-	-	-	-	-	-
A1A	0.05778	0.00047	-	-	-	-	-	-
A1B	0.87969	0.14670	0.04050	-	-	-	-	-
B8A	0.70083	0.22303	0.02254	0.81589	-	-	-	-
B8B	0.06864	0.00062	0.93813	0.04860	0.02747	-	-	-
B1A	$3.0e^{-6}$	$3.7e^{-10}$	0.00549	$1.4e^{-6}$	$4.3e^{-7}$	0.00429	-	-
B1B	0.00021	$1.2e^{-7}$	0.07124	0.00012	$4.4e^{-5}$	0.05977	0.32988	-
FA	0.02496	0.52224	$3.5e^{-5}$	0.03654	0.06315	$4.9e^{-5}$	$5.1e^{-12}$	$2.9e^{-9}$

TABLE 11 – Les comparaisons par paires des modalités pour l'erreur absolue par tâche.

Ces statistiques permettent de classifier les modalités de la manière indiquée par la table 12.

Modalité	Moyenne (mm)				
FA	0.20	<i>A</i>			
A8B	0.31	<i>A</i>	<i>B</i>		
B8A	0.53		<i>B</i>		
A1B	0.57		<i>B</i>		
A8A	0.60		<i>B</i>		
B8B	0.93			<i>C</i>	
A1A	0.94			<i>C</i>	
B1B	1.26				<i>D</i>
B1A	1.44				<i>D</i>

TABLE 12 – Classification pour l’erreur absolue par tâche.

B.4 Erreur absolue de prédiction

L’analyse comparative montre que la modalité haptique utilisée a une influence significative sur l’erreur absolue d’une prédiction ($F(8, 4) = 88.24$, $p < 0.001$). Le tableau 13 regroupe les résultats de la comparaison par paires.

	A8A	A8B	A1A	A1B	B8A	B8B	B1A	B1B
A8B	0.00350	-	-	-	-	-	-	-
A1A	$< 2e^{-16}$	$< 2e^{-16}$	-	-	-	-	-	-
A1B	0.83588	0.00303	$< 2e^{-16}$	-	-	-	-	-
B8A	0.00034	0.80142	$< 2e^{-16}$	0.00036	-	-	-	-
B8B	$5.7e^{-7}$	0.10617	$< 2e^{-16}$	$1.0e^{-6}$	0.11188	-	-	-
B1A	$2.1e^{-9}$	$< 2e^{-16}$	0.06335	$2.8e^{-8}$	$< 2e^{-16}$	$< 2e^{-16}$	-	-
B1B	$1.8e^{-8}$	$< 2e^{-16}$	$1.6e^{-5}$	$4.7e^{-7}$	$< 2e^{-16}$	$< 2e^{-16}$	0.09209	-
FA	$< 2e^{-16}$	$1.3e^{-13}$	$< 2e^{-16}$	$< 2e^{-16}$	$3.9e^{-16}$	$2.2e - 11$	$< 2e^{-16}$	$< 2e^{-16}$

TABLE 13 – Les comparaisons par paires des modalités pour l’erreur absolue par prédiction.

Ces statistiques permettent de classifier les modalités de la manière indiquée par la table 14.

Modalité	Moyenne (mm)						
A1A	17.45	<i>A</i>					
B1A	18.84		<i>B</i>				
B1B	20.00			<i>C</i>			
A1B	23.10				<i>D</i>		
A8A	23.24				<i>D</i>		
A8B	25.14					<i>E</i>	
B8A	25.29					<i>E</i>	
B8B	26.11					<i>E</i>	
FA	29.96						<i>F</i>

TABLE 14 – Classification pour l’erreur absolue par prédiction.

B.5 Temps passé proche de la cible

L'analyse comparative montre que la modalité haptique utilisée a une influence significative sur le temps passé à proximité de la cible ($F(8, 109) = 22.502$, $p < 0.001$). Le tableau 15 regroupe les résultats de la comparaison par paire.

	A8A	A8B	A1A	A1B	B8A	B8B	B1A	B1B
A8B	0.00061	-	-	-	-	-	-	-
A1A	$8.6e^{-8}$	$< 2e^{-16}$	-	-	-	-	-	-
A1B	.00033	0.86985	$< 2e^{-16}$	-	-	-	-	-
B8A	$1.5e^{-11}$	$< 2e^{-16}$	0.16241	$< 2e^{-16}$	-	-	-	-
B8B	$6.5e^{-7}$	$< 2e^{-16}$	0.70297	$< 2e^{-16}$	0.07522	-	-	-
B1A	$< 2e^{-16}$	$< 2e^{-16}$	$1.3e^{-15}$	$< 2e^{-16}$	$3.7e^{-11}$	$< 2e^{-16}$	-	-
B1B	$< 2e^{-16}$	$< 2e^{-16}$	$< 2e^{-16}$	$< 2e^{-16}$	$8.2e^{-13}$	$< 2e^{-16}$	0.58495	-
FA	$5.5e^{-8}$	0.04409	$< 2e^{-16}$	0.06439	$< 2e^{-16}$	$< 2e^{-16}$	$< 2e^{-16}$	$< 2e^{-16}$

TABLE 15 – Les comparaisons par paires des modalités pour le temps passé à proximité de la cible par tâche.

Ces statistiques permettent de classer les modalités de la manière indiquée par la table 16.

Modalité	Moyenne (s)						
FA	0.75	A					
A1B	1.11		B				
A8B	1.66		B				
A8A	1.75			C			
B8B	1.94				D		
A1A	2.77				D	E	
B8A	3.02					E	
B1A	3.91						F
B1B	4.70						F

TABLE 16 – Classification pour le temps passé à proximité par tâche.