Techniques de Pointage pour Cibles Distantes en Réalité Virtuelle

Marc Baloup, Veis Oudjail, Thomas Pietrzak, Géry Casiez

▶ **To cite this version:**

Marc Baloup, Veis Oudjail, Thomas Pietrzak, Géry Casiez. Techniques de Pointage pour Cibles Distantes en Réalité Virtuelle. Proceedings of the AFIHM Conférence Francophone sur l'interaction Homme-Machine (IHM 2018), Oct 2018, Brest, France. pp.8. hal-01899061

HAL Id: hal-01899061
https://hal.science/hal-01899061

Submitted on 19 Oct 2018

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.

Techniques de Pointage pour Cibles Distantes en Réalité Virtuelle
Pointing Techniques for Distant Targets in Virtual Reality

Marc Baloup
Université de Lille, Inria
marc.baloup@inria.fr

Veïs Oudjail
Université de Lille
veis.oudjail@etudiant.univ-lille1.fr

Thomas Pietrzak
Université de Lille
thomas.pietrzak@univ-lille.fr

Géry Casiez
Université de Lille
gery.casiez@univ-lille.fr

ABSTRACT

Raycasting is the most common target pointing technique in virtual reality environments. However, performance on small and distant targets is impacted by the accuracy of the pointing device and the user's motor skills. Current pointing facilitation techniques are currently only applied in the context of the virtual hand, *i.e.* for targets within reach. We propose to add a cursor on the ray, controlled by the user. This allows target pointing techniques by proximity to be applied, such as the *Bubble Cursor*. We describe a study comparing several visual feedbacks for this technique. Our results show that just highlighting the nearest target reduces the selection time by 14.8% and the error rate by 82.6% compared to *Raycasting* in the general case. For small targets, the selection time is reduced by 25.7% and the error rate by 90.8%.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **Pointing**; *Gestural input*;

KEYWORDS

Virtual reality, pointing technique, visual feedforward

RÉSUMÉ

Le *Raycasting* est la technique de pointage de cible privilégiée en réalité virtuelle. Les performances sur des cibles de petite taille et distantes sont cependant affectées par la précision du dispositif de pointage et par les capacités motrices de l'utilisateur. Les techniques de facilitation de pointage actuelles ne sont actuellement appliquées qu'au contexte de main virtuelle, c'est-à-dire pour des cibles à portée de main. Nous proposons d'ajouter un curseur sur le rayon, contrôlé par l'utilisateur. Ceci permet d'appliquer des techniques de pointage de cible par proximité, telles que le *Bubble Cursor*. Nous décrivons une étude comparant plusieurs retours visuels pour cette technique. Nos résultats montrent que la seule mise en surbrillance de la cible la plus proche réduit le temps de sélection de 14,8% et le taux d'erreur de 82,6% par rapport au *Raycasting* dans le cas général. Pour les petites cibles, le temps de sélection est réduit de 25,7% et le taux d'erreur de 90,8%.

MOTS-CLEFS

Réalité virtuelle, technique de pointage, retour visuel

1 INTRODUCTION

Le pointage est une tâche fondamentale des environnements 3D immersifs, au même titre que la manipulation d'objets et la navigation [3]. Les techniques de pointage en réalité virtuelle immersive se divisent en deux catégories principales : les techniques de *mains virtuelles* et les techniques à base de *Raycasting* [1, 14]. La technique de *main virtuelle* [15] utilise une correspondance directe entre les mouvements de la main dans le monde réel et ceux du pointeur, ou main virtuelle, dans l'environnement virtuel. Non seulement cette technique ne permet pas de sélectionner des cibles hors de portée, mais elle est aussi moins efficace que des techniques de type *Raycasting* [3]. En revanche la technique de *Raycasting* est moins efficace qu'une technique de main virtuelle pour la manipulation d'une cible, une fois sélectionnée. Dans cet article nous nous intéressons à l'amélioration des techniques de sélection par *Raycasting* en environnement 3D immersif.

Avec une technique de *Raycasting*, l'utilisateur manipule un rayon qui part de sa main, et dont il contrôle l'orientation, par exemple avec un dispositif de pointage pourvu de six degrés de liberté. Cette technique est comparable à l'utilisation d'un pointeur laser pour pointer des objets. Lorsque le rayon traverse une ou plusieurs cibles, la plus proche est sélectionnée. Cette technique permet de sélectionner des cibles distantes, mais avec une difficulté qui augmente avec la distance. Les petites cibles sont aussi difficiles à sélectionner. Ces deux difficultés sont dues à la fois aux limites de précision du dispositif de pointage, et aux limites des capacités motrices des utilisateurs. Cette technique est aussi sensible à l'occultation et aux distracteurs, car seule la cible intersectant la plus proche est sélectionnée. Le cas échéant, l'utilisateur doit déplacer sa main pour tenter d'atteindre la cible souhaitée en contournant les cibles plus proches.

Pour dépasser ces limitations, nous proposons d'ajouter un curseur sur le rayon, contrôlé par l'utilisateur. Ce curseur est contrôlé

par un degré de liberté offert par une manette manipulée par l'utilisateur et équipée d'un touchpad. L'utilisateur peut ainsi déplacer le curseur par un geste du pouce vers le haut ou vers le bas. L'ajout d'un curseur sur le rayon permet d'introduire des techniques inspirées du Bubble Cursor [8].

Dans la suite de cet article, après avoir présenté l'état de l'art sur les techniques de pointage en réalité virtuelle, nous détaillons le fonctionnement de la technique que nous avons mise en œuvre avec un casque de réalité virtuelle du type HTC Vive. Nous déclinons cette technique avec plusieurs retours visuels inspirés de Guillon *et al.* [10]. Nous présentons une comparaison de ces techniques et du *Raycasting* dans une expérience contrôlée.

2 ÉTAT DE L'ART

Nous commençons par une présentation des techniques de facilitation du pointage dans les environnements 3D immersifs, avant de détailler les travaux sur les techniques basées sur la sélection avec connaissance des cibles.

2.1 Améliorations du *Raycasting*

Étant donné les limitations du *Raycasting*, de nombreux travaux portent sur son amélioration. Argelaguet *et al.* détaillent un grand nombre de variantes [1]. Par exemple Liang *et al.* proposent d'utiliser un cône au lieu d'un simple rayon [13]. Ceci réduit le problème de sélection des petites cibles mais cette technique n'est utilisable qu'avec des densités faibles de cibles. Delamare *et al.* [7] proposent une technique de sélection en deux étapes dont la première étape est basée sur le cône. La deuxième étape consiste à sélectionner une cible parmi celles présentes dans le cône (qui est figé à la fin de la première phase) selon plusieurs techniques proposées : tourner la manette autour de l'axe du cône (*P2Roll*), ou glisser le doigt sur la surface tactile de gauche à droite (*P2Slide*) pour sélectionner l'objet souhaité. Cette étude utilise un smartphone comme outil de sélection, puisque ces techniques ont été pensées pour sélectionner des objets connectés dans le monde réel. Ces techniques pourraient très bien être utilisés en réalité virtuelle si nous disposons d'une manette à 6 degrés de liberté avec une surface tactile.

Une autre technique en deux étapes est le *Menu cone* de Ren *et al.* [16] qui repose sur l'utilisation de gestes effectués avec une manette. L'utilisateur oriente un cône de manière à ce que la cible se trouve dedans ; il approche ensuite la manette vers lui pour afficher un menu de sélection (les cibles se positionnent devant lui sous forme de cercle) ; l'utilisateur effectue enfin un geste dans la direction de la cible à sélectionner dans le cercle.

Une autre méthode consiste à tordre le rayon afin qu'il touche la dernière cible rencontrée [17]. L'efficacité de cette technique n'a cependant pas été évaluée.

Grossman *et al.* proposent des modifications du rayon permettant de faciliter la sélection de cibles. Par exemple le *Depth Ray* améliore le *Raycasting* en ajoutant un marqueur fixe au milieu du rayon [9]. Parmi toutes les cibles intersectées par le rayon, celle la plus proche de ce marqueur est sélectionnée. Cette technique permet de régler le problème d'occultation de cibles, puisque l'on peut ainsi sélectionner une cible qui se trouve derrière une autre, en déplaçant le marqueur derrière le distracteur. Pour déplacer le marqueur dans ce cas, l'utilisateur doit déplacer sa main vers l'avant. Cependant, les petites cibles et les cibles éloignées restent difficiles à atteindre puisque le *Depth Ray* ne permet pas de sélectionner les cibles qui ne sont pas traversées par le rayon, même si le marqueur se trouve très proche de la cible voulue. Il s'agit donc ni plus ni moins d'une extension à la main virtuelle. Cette technique donne de bonnes performances sur un écran volumétrique, comme dans l'étude de Grossmann *et al.* Cependant dans un environnement immersif l'horizon s'étend virtuellement à l'infini, et donc cette solution n'est pas satisfaisante dans beaucoup de cas de figure.

D'autres variantes telles que le *Lock Ray* et le *Flower Ray* permettent de lever des ambiguïtés lorsque plusieurs cibles traversent le rayon. Ces deux techniques ont deux phases. Dans la première, l'utilisateur déplace le rayon et verrouille sa position lorsque la cible souhaitée est intersectée. La deuxième phase de ces techniques permet de désambigüer les cibles traversées par le rayon. Ainsi, avec le *Lock Ray*, l'utilisateur déplace la manette dans le sens du rayon pour ne déplacer que le marqueur sur le rayon immobile. Avec le *Flower Ray*, les cibles interceptées par le rayon sont disposées en cercle de manière à ce que l'utilisateur puisse toutes les voir et sélectionner celle qu'il souhaite sans ambiguïté. Ces techniques ne règlent malheureusement pas les problèmes liés aux petites cibles et celles qui sont éloignées, puisque les cibles que le rayon ne traverse pas restent ignorées pour la sélection.

Cashion *et al.* [4] ont étendu la technique du *Raycasting* en ajoutant la fonctionnalité de zoom lorsque la densité dans la zone visée est élevée. Cette technique n'a été testée qu'avec un écran d'ordinateur. Le zoom est centré sur un point de l'écran d'ordinateur, le point étant déterminé par l'intersection entre l'écran et un rayon partant d'une manette à 6 degrés de liberté. De ce fait, la technique n'est pas transposable en réalité virtuelle. Aussi, cette technique ne règle pas le problème de l'occultation.

Kopper *et al.* [12] ont créé une autre variante du *Raycasting* en plusieurs étapes nommé *SQUAD*. La première étape est une version modifiée du *Raycasting* en ajoutant une sphère au point d'intersection le plus proche. À l'étape suivante, les cibles comprises dans la sphère lors de la validation de la première étape sont dispersées dans 4 zones à l'écran. L'utilisateur sélectionne alors une de ces zones, et les objets contenus sont dispersés à nouveau dans 4 zones, jusqu'à ce qu'il ne reste qu'une cible. Cette technique résout le problème des petites cibles, dans le cas où la densité est élevée, puisque la sphère permet de sélectionner un objet à proximité du rayon, même s'il n'est pas traversé par celui-ci. Cependant, lorsque la densité de cibles est plus faible, ou en cas d'occultation par des cibles plus proches, la sphère ne contient pas toujours la cible désirée.

Notre approche est différente, et s'inspire des techniques de facilitation de pointage en 2D.

2.2 Sélection avec connaissance des cibles

De nombreuses techniques de sélection se basent sur la connaissance des cibles potentielles. Par exemple Gupta *et al.* [11] ont développé une technique d'interaction permettant de sélectionner des objets sans les pointer. Cette technique nécessite de classer les cibles en catégories. L'utilisateur effectue un geste symbolique avec

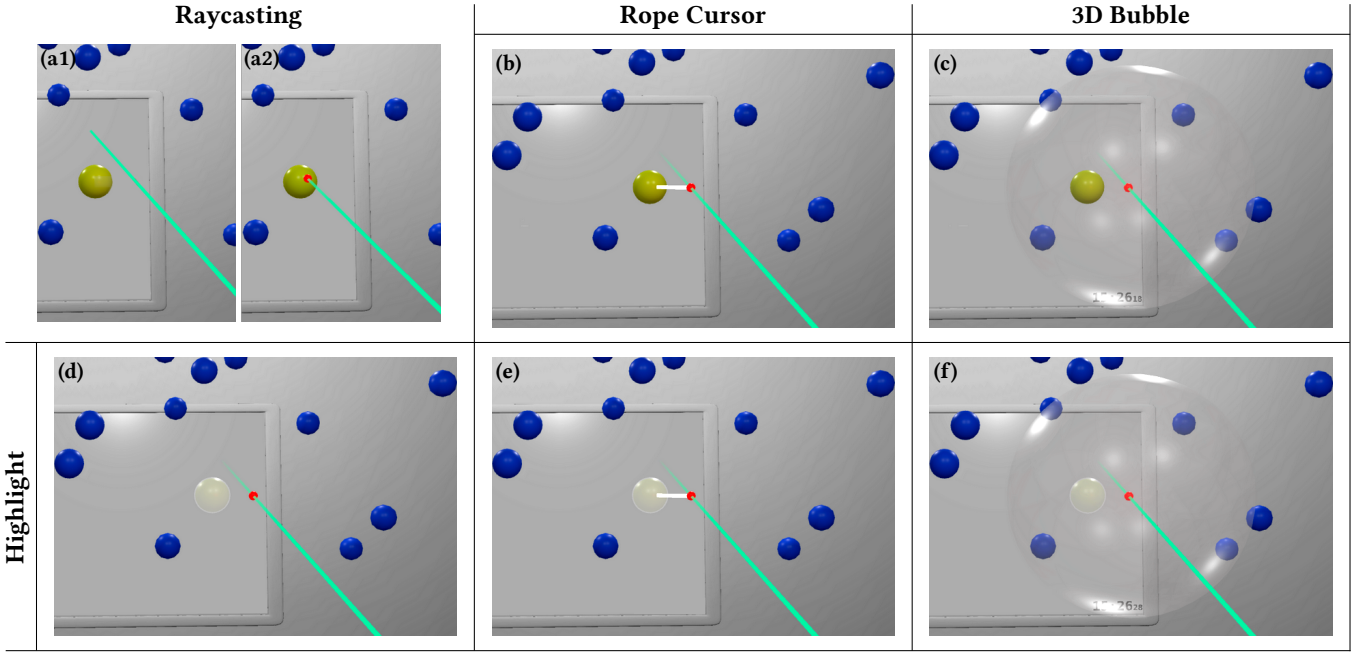

Figure 1: Techniques de pointage implémentées. Le curseur est en rouge et le rayon en cyan. (a1,a2) *Raycasting* [14] : l'utilisateur doit viser directement la cible avec le rayon (pas d'aide à la sélection, ni de contrôle du curseur). (b,e) *Rope Cursor* [10] : un rayon blanc relie le curseur à la cible la plus proche. (c,f) *3D Bubble* [18] : la bulle 3D centrée sur le curseur englobe la cible la plus proche. (d,e,f) *Highlight* : la cible la plus proche est mise en surbrillance (plus claire).

la main pour sélectionner la catégorie, puis un second pour sélectionner la cible parmi cette catégorie. Cependant la sélection avec cette technique se complique quand il y a beaucoup de cibles et de catégories. L'utilisateur doit aussi apprendre les gestes symboliques.

D'autres techniques telles que le pointage sémantique attirent le curseur de sélection sur les cibles afin de faciliter leur pointage [2]. Le désavantage principal de cette technique est sa grande sensibilité aux distracteurs [5, 6]. C'est un problème que nous cherchons justement à résoudre pour le *Raycasting*. De plus cette technique fonctionne bien grâce au pointage indirect. Le pointage par geste 3D en environnement immersif est absolu, ce qui compliquerait la conception d'une technique avec cette approche.

Un autre moyen d'utiliser les connaissances sur les cibles potentielles consiste à utiliser la distance aux cibles afin de sélectionner la plus proche. Ainsi le *Bubble Cursor* [8] utilise cette technique pour le pointage en 2D, et affiche le curseur sous forme de disque (une bulle) dont le centre est la position du curseur et le rayon la distance à la cible la plus proche. Cette technique est particulièrement efficace lorsque la densité de cibles est faible, et ce quelle que soit la taille des cibles. Le principal inconvénient de cette technique est le retour visuel gênant provoqué par les changements incessants du rayon de la bulle. Guillon *et al.* [10] ont ainsi étudié la performance de pointage avec plusieurs retours visuels. Leurs résultats montrent que la simple mise en surbrillance de la cible la plus proche est efficace. Vanacken *et al.* ont étudié une version 3D du *Bubble Cursor*, appelé *3D Bubble* [18]. Leur implémentation utilise la main virtuelle en guise de curseur 3D. Selon leur étude, le *Depth Ray* est plus efficace que la *3D Bubble*, qui est plus efficace que le *Raycasting*.

Dans notre étude, nous combinons la *3D Bubble* avec une variante de *Depth Ray* dans laquelle il est possible de manipuler la position du curseur sur le rayon.

3 TECHNIQUES DE POINTAGE

Nous allons détailler comment nous avons ré-introduit la notion de curseur avec la technique de *Raycasting*. Nous détaillerons également les variantes de retours visuels, inspirés des travaux de Guillon *et al.* [10], utilisés pour informer l'utilisateur de la cible sélectionnable.

3.1 Ajout d'un curseur sur le rayon

Notre technique se base sur une technique de *Raycasting*, à laquelle nous avons ajouté un curseur que l'utilisateur peut déplacer le long d'un rayon infini. Elle nécessite donc un degré de liberté supplémentaire pour le contrôle relatif de la position du curseur sur le rayon, que nous associons au pavé tactile d'un contrôleur HTC Vive. Il serait aussi possible d'associer ce degré de liberté à d'autres types de contrôleurs. Les déplacements de bas en haut sur le pavé tactile sont associés à un déplacement relatif du curseur le long du rayon et dans la direction opposée à l'utilisateur. La fonction de transfert utilisée est non linéaire par l'utilisation d'un gain proportionnel à la distance entre la main et le curseur [1]. Cela permet des déplacements précis quand le curseur est proche de l'utilisateur et

1. La fonction de transfert est $v_{cur} = v_{pad} \times 50 \times d_{m \leftrightarrow c}$ avec v_{cur} la vitesse du curseur sur le rayon en m s^{-1}, v_{pad} la vitesse du doigt sur la pad en m s^{-1} et $d_{m \leftrightarrow c}$ la distance manette-curseur actuelle en m.

des déplacements plus rapides quand le curseur est plus éloigné de l'utilisateur, là où la précision est moins importante compte tenu de la projection de la scène et de la résolution de l'écran.

3.2 Retours visuels

Les travaux de Guillon *et al.* ont montré une influence du retour visuel sur les performances de pointage à l'aide d'un *Bubble Cursor* dans un contexte de pointage 2D à distance [10]. Le retour d'information visuel est en effet indispensable pour mettre en évidence la cible sélectionnable. L'étude de Vanacken sur la *3D Bubble* n'utilisait que la bulle comme retour visuel. Nous allons étudier des variantes, qui sont des extensions à la 3D des retours visuels proposés par Guillon *et al.* Nous décrivons ces techniques de visualisation selon deux axes : la mise en surbrillance de la cible, et la représentation de la distance entre la cible la plus proche et le curseur.

Surbrillance de la cible. La mise en surbrillance de la cible la plus proche permet de la distinguer des autres cibles. Dans notre implémentation, la cible la plus proche du curseur est plus claire que les autres cibles (Figure 1 (d,e,f)). Ce retour visuel a l'avantage de ne pas surcharger l'espace visuel. Il peut aussi se combiner facilement avec d'autres retours visuels ci-dessous.

Représentation de la distance cible ↔ curseur. La représentation de la distance entre le curseur et la cible la plus proche permet de repérer facilement le lien entre les deux. L'affichage d'une bulle centrée sur le curseur et dont le rayon est la distance *cible* ↔ *curseur* correspond au retour visuel du *Bubble Cursor* (Figure 1 (c,f)). Guillon *et al.* ont aussi proposé le *Rope Cursor*, qui représente un rayon entre le curseur et la cible la plus proche (Figure 1 (b,e)). Ce retour visuel est plus léger, et fait un lien direct entre le curseur et la cible la plus proche. Guillon *et al.* proposent une autre technique, consistant à afficher les frontières du diagramme de Voronoï. Ce diagramme est une représentation des frontières des zones autour des cibles présentes à l'écran. Chaque zone est constituée de l'ensemble des points plus proches d'une cible que d'une autre. Nous avons implémenté cette technique en 3D avec des volumes semi-transparents (Figure 2 (a)). Cependant ce retour surcharge beaucoup l'espace visuel, alors nous l'avons écarté après des tests pilotes informels. Enfin, Guillon *et al.* ont aussi proposé de n'afficher que la cellule Voronoï de la cible la plus proche. Nous avons implémenté cette technique en 3D, mais ce retour n'est pas satisfaisant dans certains cas. Selon la configuration des cibles, le volume peut être très allongé ou même recouvrir l'écran (Figure 2 (b)). Nous l'avons donc aussi écarté pour la suite de notre étude.

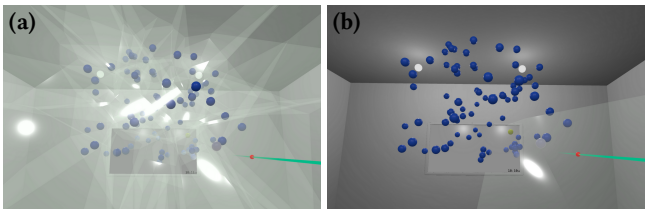

Figure 2: Régions de Voronoï. (a) Affichage du diagramme de Voronoï complet. (b) Affichage de la cellule de Voronoï de la cible la plus proche.

Combinaisons des retours visuels. Les retours visuels retenus pour notre étude sont donc la *3D Bubble*, le *Rope Cursor* ou aucune représentation de la distance *cible* ↔ *curseur*, combinés ou non avec la surbrillance de la cible la plus proche. En écartant la combinaison sans retour visuel, nous retenons 5 retours visuels (Figure 1 (b-f)).

4 EXPÉRIENCE

Le but de cette expérience est d'évaluer l'efficacité de chaque technique. Pour ce faire, on les teste avec plusieurs ensembles de cibles représentant plusieurs situations que l'on peut rencontrer lors d'une tâche de sélection. Le but est de déterminer la meilleure technique.

4.1 Matériel, tâche et participants

L'expérience s'est déroulée sur un ordinateur équipé d'un `HTC Vive`[2]. Les participants n'utilisaient qu'une seule des deux manettes, l'autre manette n'ayant aucune utilité. Les participants avaient le choix de la main utilisée pour la manette, mais ils ne pouvaient pas la tenir avec leurs deux mains. Dans l'environnement virtuel, les participants se plaçaient au centre d'une pièce de 7 m de côté et ils avaient pour consigne de ne pas se déplacer. Le centre était représenté par un carré de 70 cm de côté dessiné au sol, dans lequel les utilisateurs devaient se placer. Les instructions données aux participants étaient affichées sur un écran virtuel placé devant lui (Figure 3). L'application de l'expérience a été développée en `C#` avec `Unity 3D`.

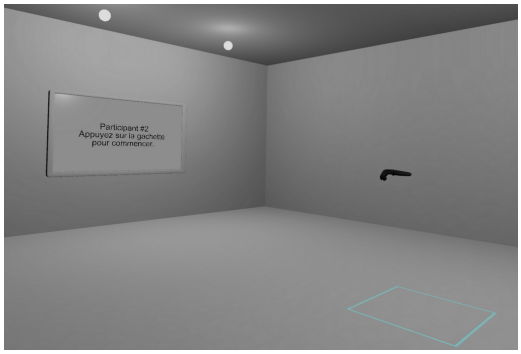

Figure 3: Scène 3D utilisée pour l'expérience. Les participants se plaçaient debout dans le carré cyan tracé au sol (en bas à droite). Les instructions apparaissaient à l'écran (à gauche). Les cibles étaient positionnées entre l'écran et le participant.

Le protocole consiste à sélectionner des cibles qui apparaissent dans l'environnement virtuel. La cible à sélectionner est de couleur jaune, tandis que les autres sont de couleur bleue. Lorsque l'utilisateur sélectionne la bonne cible (en appuyant sur la gâchette de la manette), il passe à l'essai suivant. Sinon, des feedbacks visuels et haptiques informent le participant qu'il n'a pas sélectionné la cible correctement, et qu'il doit réessayer avant de passer à l'essai suivant. Le retour haptique était produit par une brève vibration de la manette. Quant au retour visuel, il était produit par un brève clignotement de la cible à sélectionner. Les participants avaient

2. https://www.vive.com/fr/product/

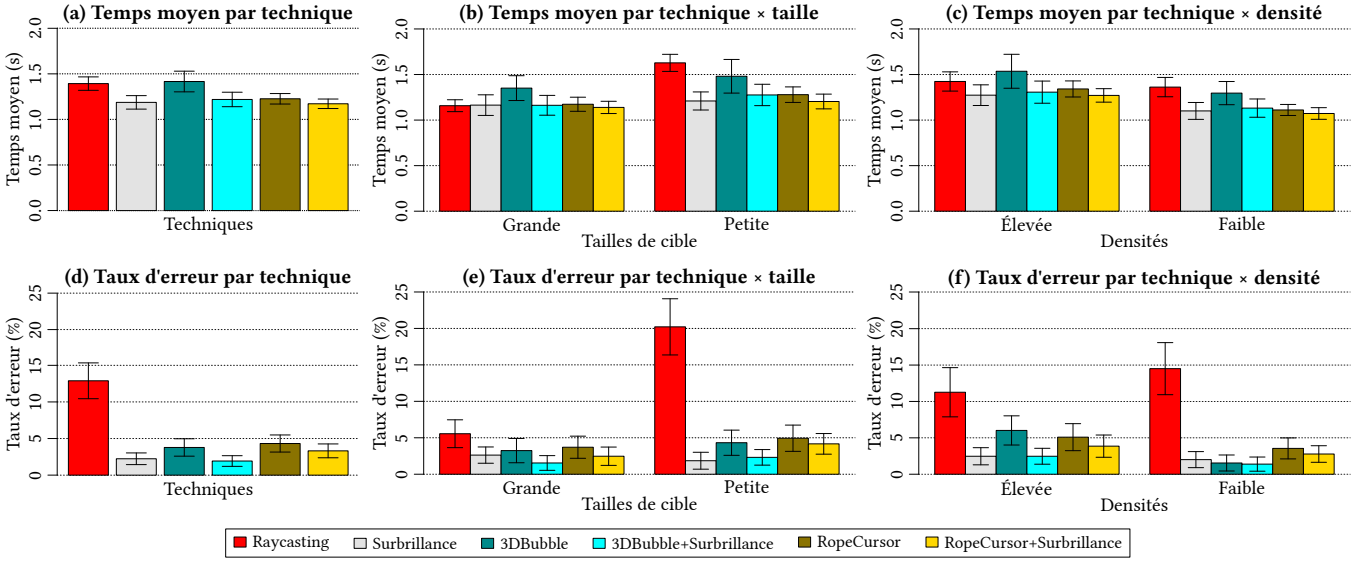

Figure 4: Temps moyens et taux d'erreur moyens relevés lors de l'expérience, avec les intervalles de confiance à 95 %.

pour instruction de réaliser la tâche en respectant un taux d'erreur autour de 4% pour le compromis vitesse / précision. Si le taux d'erreur était au-dessus de 4%, ils devaient essayer d'être plus précis. Au dessus de ce taux ils devaient essayer d'être plus rapide.

Douze participants non rémunérés, âgés de 26 ans en moyenne ($\sigma = 4,35$), ont participé à l'expérience. L'ensemble était composé de 11 hommes et 1 femme, 2 vivaient leur première expérience de réalité virtuelle, et tous étaient droitiers.

4.2 Conditions et procédure

Nous avons utilisé un plan expérimental intra-sujets. Les variables indépendantes étaient la technique utilisée (Technique), le bloc d'essai (Bloc), la densité de chaque ensemble de cibles (Densité) et la taille des cibles (Taille).

Pour ce qui est des techniques, nous en avons choisi 6, dont le *Raycasting* (*RC*) comme technique de référence ainsi que les 5 retours visuels expliqués Section 3.2 : *3DBubble* (*Bub*), *RopeCursor* (*Rope*), *Surbrillance* (*HL*), *3DBubble+Surbrillance* (*Bub+HL*), *RopeCursor+Surbrillance* (*Rope+HL*). L'ordre des 6 techniques a été contrebalancé entre chaque participant, en utilisant un carré latin.

Les tailles de cibles étaient S_{Petite} = 2 cm et S_{Grande} = 4 cm.

Il y avait 15 cibles pour D_{Faible} et 40 cibles pour $D_{Élevée}$. Celles-ci étaient disposées de manière pseudo-aléatoire dans un volume décrit par une sphère de 2 m de diamètre, dont le centre était positionné à une distance de 2 m en face de l'utilisateur. Toutes les cibles étaient visibles dans le champ de vision du participant quand il regardait le mur en face de lui de manière orthogonale.

Pour chaque technique, les participants enchaînaient 3 Blocs, dans lesquels ils devaient sélectionner 10 cibles pour chaque combinaison de Densité × Taille. Après chaque ensemble de 10 cibles, les cibles disparaissaient et le taux d'erreur depuis le début de la technique courante était affiché à l'écran virtuel. L'utilisateur devait appuyer sur la gâchette pour passer à la suite.

Pour chacune des combinaisons de Densité×Taille, nous avons généré un ensemble de cibles qui est utilisé tout au long de l'expérience pour tous les participants, toutes les techniques et tous les blocs. De plus, les cibles à sélectionner (en jaune) étaient toujours présentées dans le même ordre. De ce fait, les distances entre les cibles étaient toutes identiques d'un bloc à l'autre, et d'une technique à l'autre.

Après chaque technique, le participant devait répondre à un questionnaire NASA-TLX sur la technique, et il avait aussi la possibilité de faire une pause. Nous utilisons ce questionnaire comme indicateur pour la charge de travail de l'utilisateur. À la fin de l'expérience, l'utilisateur devait choisir la technique qu'il a préféré. Chaque participant réalisait l'expérience en environ 30 minutes.

Nous avons en résumé un design d'expérience de 12 participants × 6 Techniques × 3 Blocs × 2 Densités × 2 Tailles × 10 essais = 8640 essais au total.

4.3 Résultats

Pour déterminer l'efficacité d'une technique, nous avons mesuré le temps moyen de sélection ainsi que le taux d'erreur. Comme l'enchaînement des essais était interrompu entre chaque ensemble de 10 essais, le premier essai qui suit chaque interruption est ignoré pour les analyses.

4.3.1 *Temps de sélection.*

Le temps de sélection est défini comme le temps pris pour aller d'une cible à une autre et la sélectionner. Les essais pour lesquels l'utilisateur n'a pas sélectionné la bonne cible du premier coup sont considérés comme des erreurs et ignorés pour cette analyse.

Une Anova à mesures répétées [3] montre un effet significatif de Bloc sur le temps de sélection ($F_{1.04,11.4} = 12,3$, $p = 0,004$). Des comparaisons deux à deux montrent des différences significatives entre les blocs 1 et 2 ($p < 0,0003$), mais pas entre les blocs 2 et 3

3. Les ANOVA ont été réalisées avec R et SPSS. Des corrections de Greenhouse-Geisser ont été appliquées aux degrés de liberté quand la sphéricité était violée. Les comparaisons paires à paires utilisent la correction de Bonferroni.

($p = 0,88$) (Bloc 1 : 1,45s, 2 : 1,29s, 3 : 1,25s). Ceci peut être expliqué par l'effet d'apprentissage. Le premier bloc est donc ignoré pour le reste de cette analyse.

On observe aussi un effet significatif de Technique ($F_{2.1,23.6} = 5,81$, $p = 0,008$). Des comparaisons deux à deux montrent une différence significative entre *Raycasting* et les autres techniques à l'exception de *3DBubble* (RC : 1,39s ; HL : 1,19s ; Bub : 1,42s ; Bub+HL : 1,22s ; Rope : 1,23s ; Rope+HL : 1,17s ; $p < 0,041$). Les sélections sont en moyenne plus longues à exécuter avec les techniques *Raycasting* et *3DBubble* (sans Surbrillance) qu'avec les autres techniques testées (Figure 4 (a)).

L'Anova montre aussi un effet significatif de Taille ($F_{1,11} = 74,6$, $p < 0,0001$) et une interaction de Technique×Taille ($F_{2.3,24.9} = 22,6$, $p < 0,0001$). Des comparaisons deux à deux ne montrent pas de différence significative entre les techniques pour la taille de cible la plus grande (Figure 4 (b)). Par contre pour les petites cibles, *Raycasting* est significativement plus lente ($p < 0,0001$) que toutes les autres techniques à l'exception de *3DBubble* (RC : 1,63s ; HL : 1,28s ; Rope : 1,28s ; Rope+HL : 1,20 s). Pour les cibles de petite taille, *3DBubble* est significativement plus lente ($p < 0,02$) que *Surbrillance* et *RopeCursor+Surbrillance*.

On observe aussi un effet significatif de Densité ($F_{1,11} = 68,0$, $p < 0,0001$) et une interaction de Technique×Densité ($F_{1.7,18.6} = 3,73$, $p = 0,03$). Pour la densité *Faible*, on constate un effet significatif entre *Raycasting* (1, 36s) et les techniques *Surbrillance* (1, 10s ; $p = 0,04$) et *RopeCursor+Surbrillance* (1, 07s ; $p = 0,01$). Pour la densité *Élevée*, on constate un effet significatif entre *3DBubble* (1, 54s) et les techniques *Surbrillance* (1, 27s ; $p = 0,04$) et *RopeCursor+Surbrillance* (1, 27s ; $p = 0,035$). La *3DBubble* sans mise en surbrillance semble plus difficile à utiliser lorsque la densité est élevée. En effet, il est parfois difficile de se rendre compte de la cible actuellement englobée par la bulle en absence de la mise en surbrillance. Cette difficulté est moins présente lorsque la densité est faible car il est plus facile de positionner le curseur de manière à ce que la bulle ne semble englober, du point de vue de l'utilisateur, qu'une seule cible. Sur la Figure 1 (c), bien que dans l'espace 3D la bulle n'englobe que la cible jaune, l'utilisateur pourrait hésiter entre les 4 cibles contenues dans la projection de la sphère sur le plan de la caméra.

4.3.2 *Taux d'erreur.*

Le taux d'erreur est le nombre d'essais pour lesquels le participant n'a pas sélectionné la bonne cible du premier coup, par rapport au nombre total d'essais. Le taux d'erreur global est de 4,75%, sachant que les participants avaient pour consigne de suivre un taux de 4% en moyenne.

Une Anova à mesures répétées ne montre aucun effet significatif de Bloc sur le taux d'erreur ($F_{2,22} = 1,05$, $p > 0.05$).

L'Anova montre un effet significatif de Technique ($F_{1.8,20.2} = 17,2$, $p < 0,0001$). Des comparaisons deux à deux montrent un effet significatif entre la technique *Raycasting* (12,9%) et toutes les autres techniques (HL : 2,2% ; Bub : 3,8% ; Bub+HL : 1,9% ; Rope : 4,3% ; Rope+HL : 3,3% ; $p < 0,0001$).

On observe aussi un effet significatif de Taille ($F_{1,11} = 28,9$, $p = 0,0002$) et une interaction de Technique×Taille ($F_{1.7,18.3} = 22,0$, $p < 0,0001$). Des comparaisons deux à deux montrent que le taux d'erreur pour le *Raycasting* augmente significativement lorsque la taille de la cible diminue ($E_{RC \times Grande} = 5,6\%$; $E_{RC \times Petite} = 20,2\%$; $p < 0,0001$). Aucun effet significatif n'est relevé pour les autres techniques ($p = 1$). Le taux d'erreur de la technique de *Raycasting* plus élevé pour les petites cibles est connu de la littérature et peut s'expliquer par les mouvements involontaires de la main lors de la sélection. Les autres techniques souffrent moins de ce type de problème. Aucun effet significatif de technique sur le taux d'erreur n'a été relevé pour les cibles de grande taille. Pour les cibles de petite taille, on constate que *Raycasting* (20, 2%) a un taux d'erreur significativement plus élevé que toutes les autres techniques (< 5% ; $p < 0,0001$).

L'Anova montre aussi une interaction significative de Technique×Densité ($F_{2.9,31.8} = 4,46$, $p = 0,01$). Des comparaisons deux à deux ne montrent aucun effet significatif de Densité pour chaque technique. Pour la densité *Faible*, on constate un effet significatif entre *Raycasting* (14, 5%) et toutes les autres techniques (< 3, 6% ; $p < 0,0001$). Pour la densité *Élevée*, on constate aussi un effet significatif entre *Raycasting* (11, 3%) et toutes les autres techniques (< 6, 1% ; $p < 0,008$).

4.3.3 *Questionnaire NASA-TLX.*

L'analyse de Friedman sur les réponses au questionnaire NASA-TLX indique qu'il y a un effet significatif pour l'exigence physique ($\chi^2(5) = 18,7$, $p = 0,002$), l'exigence mentale ($\chi^2(5) = 12,1$, $p = 0,03$), l'effort ($\chi^2(5) = 23,8$, $p = 0,0002$) et la frustration ($\chi^2(5) = 25,6$, $p = 0,0001$). L'analyse post-hoc de Wilcoxon révèle que le niveau de frustration est significativement plus élevé pour la technique *Raycasting* (médiane= 14) que pour la technique *Surbrillance* (médiane= 4, $p = 0,03$). Il n'y a pas d'autre différence significative observée pour les tests post-hocs.

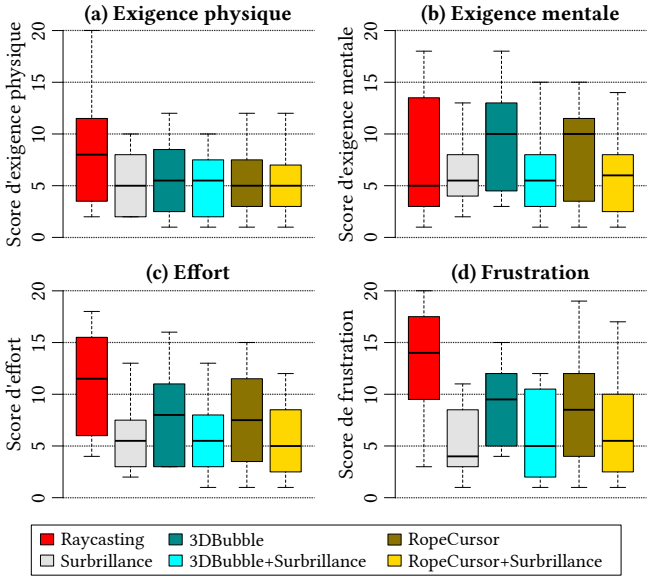

Figure 5: Diagrammes en boîte des réponses aux critères «Exigence physique», «Exigence mentale», «Effort» et «Frustration» au questionnaire NASA-TLX pour chaque technique.

4.3.4 *Préférences des participants.*

Chaque participant a pu choisir sa technique préférée parmi les 6 testées. Les résultats montrent que les participants ont préféré les

techniques avec la mise en surbrillance activée (3 à 4 votes pour chaque) tandis que les autres, *Raycasting* inclus, n'ont reçu qu'un seul vote ou aucun.

4.4 Discussion

Tout d'abord nos résultats montrent que la technique que nous avons proposée est plus efficace que la technique de *Raycasting* utilisée comme référence. La technique de *Raycasting* rend la tâche de sélection significativement plus lente que la plupart des autres techniques pour des petites cibles. Le *Raycasting* a aussi un taux d'erreur bien plus élevé que la plupart des autres techniques lors de la sélection de petites cibles. Concernant les retours visuels testés, nous avons relevé que le *3DBubble* (le retour visuel de base) est significativement plus lent que les retours visuels *Surbrillance* et *RopeCursor+Surbrillance* pour une densité élevée. Nous n'avons relevé aucun autre effet significatif entre les autres retours visuels. Ces résultats sont similaires à ceux de l'expérience de Guillon *et al.* [10] dans le cadre d'une interaction en 2D. Finalement, si nous devions choisir un retour visuel, la *Surbrillance* serait à privilégier puisque c'est celui qui ajoute le moins de charge visuelle. Nos résultats montrent qu'en ajoutant un curseur sur le rayon, avec la sélection par proximité et avec comme seul retour visuel la mise en surbrillance de la cible la plus proche, le temps de sélection est réduit de 14,8% et le taux d'erreur de 82,6% par rapport au *Raycasting* dans le cas général. Pour les petites cibles, le temps de sélection est réduit de 25,7% et le taux d'erreur de 90,8%.

Cependant, les nouvelles techniques implémentées ajoutent un contrôle supplémentaire par rapport au *Raycasting* (la position du curseur sur le rayon). L'utilisateur doit donc apprendre à manipuler ce nouveau degré de liberté. Aussi, du point de vue de la conception de l'interface, ce degré de liberté occupe un contrôle sur la manette qui pourrait être attribué à une autre tâche. Une amélioration possible serait alors de rendre facultative l'utilisation de ce contrôle supplémentaire, tout en conservant le mécanisme d'aide à la sélection que nous avons testé. En pratique, la technique par défaut utilisée serait le *Raycasting* et le curseur n'apparaitrait que lors d'un appui sur un bouton ou lorsque le pouce serait en contact sur le touchpad.

Nous avons réussi à améliorer la technique de *Raycasting* pour la sélection de cibles. Cependant, nous n'avons pas comparé notre technique avec les améliorations existantes du *Raycasting*.

Notre technique pourrait aussi améliorer le positionnement des objets sélectionnés. En effet les objets sélectionnés avec la technique de *Raycasting* peuvent ensuite être re-positionnés en les attachant au rayon. Cependant l'espace de manipulation est principalement limité à celui d'une sphère dont le rayon est la distance entre l'utilisateur et l'objet. Rapprocher un objet de l'utilisateur ne peut ainsi se faire que par une série d'allées-venues de l'utilisateur, accompagnées de relâchement et re-sélection de l'objet à chaque itération. Notre technique permettrait de facilement rapprocher l'objet de l'utilisateur en contrôlant simplement le curseur le long du rayon.

5 CONCLUSION

Nous avons présenté des nouvelles techniques de pointage utilisant un curseur sur le rayon contrôlé par l'utilisateur, appliquant le pointage de cible par proximité tel que le *Bubble Cursor*, et ayant différents retours visuels. Nous avons présenté les résultats d'une expérience comparant ces nouvelles techniques avec la technique *Raycasting*. Nos résultats montrent qu'en ajoutant un curseur sur le rayon, avec la sélection par proximité et avec comme seul retour visuel la mise en surbrillance de la cible la plus proche, le temps de sélection est réduit de l'ordre de 15% et le taux d'erreur de plus de 80% par rapport au *Raycasting* dans le cas général. Pour les petites cibles, le temps de sélection est réduit de plus de 25% et le taux d'erreur de plus de 90%. Bien que nous n'avons testé la technique qu'avec des cibles sphériques réparties dans l'espace, elle pourrait tout à fait se transposer dans un contexte écologique pour la sélection d'objets de formes arbitraires, par exemple, la sélection de pièces dans un moteur ou la sélection d'objets dans une pièce. Dans nos prochaines études, nous nous intéresserons à d'autres interactions élémentaires, telles que la manipulation de cibles.

RÉFÉRENCES

[1] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers and Graphics* 37, 3 (2013), 121 – 136. DOI : http://dx.doi.org/10.1016/j.cag.2012.12.003

[2] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. 2004. Semantic Pointing : Improving Target Acquisition with Control-display Ratio Adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 519–526. DOI : http://dx.doi.org/10.1145/985692.985758

[3] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2004. *3D User Interfaces : Theory and Practice.* Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

[4] Jeffrey Cashion, Chadwick Wingrave, and Joseph J LaViola Jr. 2012. Dense and dynamic 3d selection for game-based virtual environments. *IEEE transactions on visualization and computer graphics* 18, 4 (2012), 634–642.

[5] Géry Casiez, Nicolas Roussel, Romuald Vanbelleghem, and Frédéric Giraud. 2010. EfficacitÉ Et Robustesse Aux Distracteurs D'Un Retour Tactile Pour Faciliter Le Pointage. In *Proceedings of the 22Nd Conference on L'Interaction Homme-Machine (IHM '10)*. ACM, New York, NY, USA, 25–32. DOI : http://dx.doi.org/10.1145/1941007.1941012

[6] Géry Casiez, Nicolas Roussel, Romuald Vanbelleghem, and Frédéric Giraud. 2011. Surfpad : Riding Towards Targets on a Squeeze Film Effect. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2491–2500. DOI : http://dx.doi.org/10.1145/1978942.1979307

[7] William Delamare, Céline Coutrix, and Laurence Nigay. 2013. Mobile pointing task in the physical world : balancing focus and performance while disambiguating. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*. ACM, 89–98.

[8] Tovi Grossman and Ravin Balakrishnan. 2005. The bubble cursor : enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 281–290.

[9] Tovi Grossman and Ravin Balakrishnan. 2006. The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 3–12. DOI : http://dx.doi.org/10.1145/1166253.1166257

[10] Maxime Guillon, François Leitner, and Laurence Nigay. 2015. Investigating Visual Feedforward for Target Expansion Techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2777–2786. DOI : http://dx.doi.org/10.1145/2702123.2702375

[11] Aakar Gupta, Thomas Pietrzak, Cleon Yau, Nicolas Roussel, and Ravin Balakrishnan. 2017. Summon and Select : Rapid Interaction with Interface Controls in Mid-air. In *Proceedings of the ACM Conference on Interactive Surfaces and Spaces (ISS 2017)*. Brighton, United Kingdom. DOI : http://dx.doi.org/10.1145/3132272.3134120

[12] Regis Kopper, Felipe Bacim, and Doug A Bowman. 2011. Rapid and accurate 3D selection by progressive refinement. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*. IEEE, 67–74.

[13] Jiandong Liang and Mark Green. 1994. JDCAD : A highly interactive 3D modeling system. *Computers & Graphics* 18, 4 (1994), 499 – 506. DOI : http://dx.doi.org/10.1016/0097-8493(94)90062-0

[14] Mark R Mine. 1995. Virtual environment interaction techniques. *UNC Chapel Hill CS Dept* (1995).

[15] Mark R. Mine, Frederick P. Brooks, Jr., and Carlo H. Sequin. 1997. Moving Objects in Space : Exploiting Proprioception in Virtual-environment Interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 19–26. DOI : http://dx.doi.org/10.1145/258734.258747

[16] Gang Ren and Eamonn O'Neill. 2013. 3D selection with freehand gesture. *Computers & Graphics* 37, 3 (2013), 101–120.

[17] Frank Steinicke, Timo Ropinski, and Klaus Hinrichs. 2004. Object selection in virtual environments with an improved virtual pointer metaphor. In *Computer Vision and Graphics : International Conference, ICCVG 2004*. Springer Netherlands, Warsaw, Poland, 320–326. DOI : http://dx.doi.org/10.1007/1-4020-4179-9_46

[18] L. Vanacken, T. Grossman, and K. Coninx. 2007. Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments. In *2007 IEEE Symposium on 3D User Interfaces*. DOI : http://dx.doi.org/10.1109/3DUI.2007.340783