

Amélioration du Raycasting par utilisation de la sélection par proximité et du filtrage

Marc BALOUP

Inria, Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRISTAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France

marc.baloup@inria.fr

Thomas PIETRZAK

Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRISTAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France

thomas.pietrzak@univ-lille.fr

Géry CASIEZ

Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRISTAL - Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France

gery.casiez@univ-lille.fr

Improving Raycasting using Proximity Selection and Filtering

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 enhancements to *Raycasting* : filtering the ray, and adding a controllable cursor on the ray to select the nearest target. We describe a series of studies for the design of the visual feedforward, filtering technique, as well as a comparative study between different 3D pointing techniques. Our results show that highlighting the nearest target is one of the most efficient visual feedforward technique. We also show that filtering the ray reduces error rate in a drastic way. Finally we show the benefits of *RayCursor* compared to *Raycasting* and another technique from the literature.

Key words : Virtual reality, pointing technique, visual feedforward

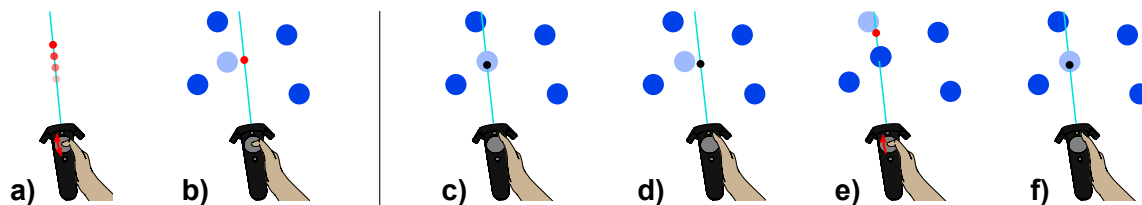
Résumé. *Raycasting* est la technique de pointage de cible la plus courante dans les environnements de réalité virtuelle. Cependant, sa performance sur des cibles à la fois petites et distantes est affectée par la précision du dispositif de pointage et les capacités motrices de l'utilisateur. Les techniques actuelles de facilitation du pointage ne sont actuellement appliquées que dans le contexte de la main virtuelle, c'est-à-dire pour les cibles à portée de main. Nous proposons des améliorations au *Raycasting* : le filtrage du rayon, et l'ajout d'un curseur contrôlable sur celui-ci pour sélectionner la cible la plus proche. Nous décrivons une série d'études pour la conception d'aides visuelles, la technique de filtrage, ainsi qu'une étude comparative entre différentes techniques de pointage 3D. Nos résultats montrent que la mise en évidence de la cible la plus proche est l'une des techniques les plus efficaces de rétroaction visuelle. Nous montrons également que le filtrage du rayon réduit le taux d'erreur de manière drastique. Enfin, nous montrons les avantages de *RayCursor* par rapport à *Raycasting* et à une autre technique de la littérature.

Mots-clés : Réalité virtuelle, technique de pointage, retours visuels

Édité par Joëlle Coutaz, Université de Grenoble et Patrick Girard, Université de Poitiers

Soumis le 17 juin 2019 – Accepté le 30 septembre 2019

Figure 1. Illustration du RayCursor manuel : a) l'utilisateur contrôle un curseur le long du rayon grâce au déplacement relatif de son pouce sur le pavé tactile; b) la cible la plus proche du curseur est mise en surbrillance. Illustration du RayCursor semi-auto : c) par défaut, il s'utilise comme Raycasting. Le curseur (en noir) se positionne automatiquement à l'intersection du rayon avec une cible; d) la cible reste mise en surbrillance même lorsque le curseur s'éloigne, jusqu'à ce qu'il soit plus proche d'une autre cible; e) l'utilisateur peut reprendre le contrôle du curseur en utilisant le pavé tactile, pour sélectionner une cible occultée (le curseur devient rouge pour indiquer le mode manuel); f) si l'utilisateur n'utilise plus le pavé tactile pendant 1s, le curseur retrouve le comportement décrit en c).



1 INTRODUCTION

Le pointage est une tâche fondamentale dans tout système interactif comportant un espace d'interaction 2D ou 3D. De nombreuses techniques ont été proposées et améliorées au fil des décennies pour les environnements 2D, afin d'exploiter les degrés de liberté offerts par les nouveaux périphériques d'entrée (souris d'ordinateur, interfaces tactiles ou oculomètres) ou les caractéristiques des objets affichés, en essayant par exemple de réduire artificiellement la distance aux objets (Grossman & Balakrishnan, 2005; Guiard, Blanch, & Beaudouin-Lafon, 2004) ou en augmentant leur largeur (Blanch, Guiard, & Beaudouin-Lafon, 2004). Les environnements 3D offrent également de multiples techniques pour accomplir cette tâche. Ces techniques peuvent être divisées en deux grandes catégories : les mains virtuelles et les techniques basées sur le *Raycasting* (Bowman, Kruijff, LaViola, & Poupayrev, 2004; Mine, 1995). Les techniques de mains virtuelles offrent un isomorphisme entre la main réelle et la main virtuelle (Argelaguet & Andujar, 2013). Malgré de nombreuses techniques destinées à améliorer *Raycasting* (Cashion, Wingrave, & LaViola Jr, 2012; de Haan, Koutek, & Post, 2005; Grossman & Balakrishnan, 2006; Kopper, Bacim, & Bowman, 2011; Liang & Green, 1994; Ren & O'Neill, 2013; Ro et al., 2017; Schmidt, Baillot, Brown, Tomlin, & Swan, 2006; Steinicke, Ropinski, & Hinrichs, 2004) (voir (Argelaguet & Andujar, 2013) pour un aperçu plus exhaustif), le *Raycasting* standard et la main virtuelle restent les deux techniques par défaut disponibles avec des systèmes tels que le HTC Vive et leurs environnements de programmation.

Avec la technique de *Raycasting*, l'utilisateur manipule un rayon dont l'origine et l'orientation sont définies par celles d'un périphérique d'entrée à 6 degrés de liberté, d'une manière similaire à celle d'un pointeur laser. Lorsque le rayon intersecte plusieurs objets, celui qui est le plus proche de l'utilisateur peut être sélectionné. Cette technique permet de sélectionner des cibles avec une difficulté croissante lorsque les cibles sont plus éloignées ou plus petites, en raison des limites du suivi du mouvement et des capacités motrices humaines. Cette technique est également affectée par l'occultation et les cibles distrayantes, car seule la cible la plus proche peut être sélectionnée. Ainsi, pour sélectionner des cibles cachées il faut changer la position et l'orientation du rayon.

De nombreuses techniques ont déjà été conçues et améliorées pour surmonter ces limites. De nouveaux périphériques d'entrée et de nouveaux contextes d'interaction offrent de nouvelles possibilités d'améliorer les techniques de pointage. Les nouveaux contrôleurs, disponibles sur le HTC Vive, offrent des degrés de liberté supplémentaires tels que des pavés tactiles qui n'étaient pas disponibles auparavant.

Nous proposons une technique de *Raycasting* améliorée, appelée *RayCursor*¹, qui utilise un curseur sur le rayon. Cette technique a été conçue principalement pour les environnements immersifs utilisant le HTC Vive, mais elle pourrait également être utilisée dans d'autres

1. Ressources additionnelles disponibles à l'adresse ns.inria.fr/loki/raycursor

contextes 3D comportant un contrôleur 6 degrés de liberté avec un pavé tactile et des boutons. L'utilisateur peut contrôler le curseur le long du rayon en utilisant les déplacements relatifs de son pouce sur le pavé tactile. Comme pour le *Bubble Cursor* (Grossman & Balakrishnan, 2005 ; Vanacken, Grossman, & Coninx, 2007), la cible la plus proche du curseur peut être sélectionnée lorsque l'utilisateur appuie sur un bouton.

Après avoir présenté les travaux relatifs aux techniques de sélection 3D basées sur *Raycasting* et les techniques de facilitation du pointage, nous décrivons la conception de *RayCursor* et ses caractéristiques : retour visuel, fonction de transfert du curseur et filtrage du rayon. Nous expliquons la mise en place générale de nos études expérimentales, et une série d'expériences examinant chacune des caractéristiques de la technique. Enfin, nous détaillons une expérience utilisateur, comparant notre technique au *Raycasting* standard et la technique de la littérature la plus proche (Ro et al., 2017).

2 ÉTAT DE L'ART

La section état de l'art porte sur les techniques de sélection basées sur *Raycasting*. Nous commençons par présenter les différents mécanismes de désambiguïsation qui ont été proposés pour ces techniques. Nous présentons ensuite des techniques qui ajoutent des degrés de liberté supplémentaires au rayon avant de couvrir les techniques de facilitation du pointage qui reposent sur la sélection par proximité.

2.1 Techniques de désambiguïsation

Argelaguet et Andujar fournissent une taxonomie des différentes techniques conçues pour améliorer le *Raycasting* (Argelaguet & Andujar, 2013). La plupart de ces techniques sont basées sur l'utilisation d'un volume au lieu d'un rayon, ce qui nécessite l'utilisation de techniques de désambiguïsation pour la sélection. Ils distinguent trois groupes de mécanismes de désambiguïsation : manuel, heuristique et comportemental.

L'approche manuelle nécessite des étapes supplémentaires pour sélectionner manuellement une cible parmi celles qui sont mises en évidence. Par exemple, dans *Flower Ray*, Grossman *et al.* affichent dans un menu circulaire les objets traversés par le rayon (Grossman & Balakrishnan, 2006). La technique *Menu Cone* affiche également les cibles à désambiguïser dans un menu et l'utilisateur fait un geste pour sélectionner la cible désirée (Ren & O'Neill, 2013). De la même manière, la technique *SQUAD*, proposée par Kopper *et al.*, adapte le *Raycasting* pour projeter une sphère sur la surface en intersection la plus proche du rayon, afin de déterminer quels objets sont susceptibles d'être sélectionnés (Kopper et al., 2011). Les objets sélectionnables sont ensuite répartis sur quatre quadrants et l'utilisateur affine la sélection jusqu'à ce que l'objet souhaité puisse être sélectionné. *SQUAD* a montré des performances nettement meilleures que *Raycasting* pour les cibles de petites tailles et les faibles densités de cibles, mais elle présente également une dégradation significative des performances pour les cibles de grandes tailles et les densités importantes, en raison du nombre accru d'étapes pour sélectionner une cible. Cashion *et al.* proposent une variante de *SQUAD*, appelée *Expand*, qui ajoute la possibilité de zoomer (Cashion et al., 2012). Ils montrent qu'*Expand* est plus rapide que *SQUAD* pour des densités d'objets élevées. Sans recourir aux menus, le *Depth Ray* (Grossman & Balakrishnan, 2006) utilise un curseur fixe au milieu d'un rayon dont la longueur est également fixe. Lorsque le rayon intersecte plusieurs objets, celui qui est le plus proche du curseur peut être sélectionné. L'utilisateur peut alors ajuster la position du *Depth Ray* en changeant la position et l'orientation de sa main. Les auteurs l'ont utilisé avec un écran volumétrique, autour duquel l'utilisateur peut se déplacer. En réalité virtuelle (RV), l'espace est grand, potentiellement infini, et ce curseur fixe n'est ni plus ni moins qu'une main virtuelle distante. Grossman et Balakrishnan ont également proposé la technique du *Lock Ray* qui consiste à verrouiller tous les objets en intersection, avant d'en sélectionner un à l'aide du curseur (Grossman & Balakrishnan, 2006), en bouclant entre les objets sélectionnés (Hinckley, Pausch, Goble, & Kassell, 1994). Cependant, cela n'améliore pas le temps de mouvement. En utilisant un smartphone pour sélectionner des objets dans le monde physique, Delamare *et al.* ont proposé deux tech-

niques pour désambigüiser les cibles sélectionnées dans un cône (Delamare, Coutrix, & Nigay, 2013). Avec le P2Roll, l'utilisateur effectue un geste de roulement pour sélectionner la cible souhaitée. Avec le P2Slide, il réalise un geste de glissement sur la surface tactile. Leurs techniques n'ont été évaluées qu'avec un maximum de 16 cibles.

L'approche heuristique permet de déterminer la cible que l'utilisateur souhaite sélectionner selon des choix prédéterminés. La technique Flashlight, par exemple, met en évidence l'objet qui est le plus proche de l'axe central du cône de sélection (Liang & Green, 1994). Le Skicky-Ray, basé sur *Raycasting* (Steinicke et al., 2004), laisse quant à lui sélectionnable le dernier objet intersecté, jusqu'à ce qu'un autre soit touché. Le rayon virtuel est dévié vers les objets qui peuvent être sélectionnés, n'ayant plus ainsi de retour visuel pour sélectionner un autre objet. Cette technique n'a pas été évaluée. Schimdt *et al.* ont proposé différents algorithmes probabilistes basés sur le pointage pour déduire la cible que l'utilisateur veut sélectionner, mais cela nécessite un réglage complexe de schémas de pondérations selon l'application (Schmidt et al., 2006).

Enfin, les approches comportementales prennent en compte les actions de l'utilisateur avant la confirmation de la sélection pour déterminer l'objet à sélectionner. Par exemple, IntenSelect utilise une fonction de notation basée sur le temps pour déterminer le score des objets qui se situent dans un volume de sélection conique. L'objet avec le score le plus élevé peut être sélectionné (de Haan et al., 2005). De la même manière, le Smart Ray pondère continuellement les cibles en fonction de leur proximité au curseur du rayon (Grossman & Balakrishnan, 2006). Cependant, cette dernière technique est moins performante que les techniques telles que Flower Ray ou Depth Ray (Grossman & Balakrishnan, 2006).

En résumé, les techniques d'interaction actuelles qui améliorent le *Raycasting* nécessitent un mécanisme de désambigüisation qui ajoute des étapes supplémentaires pour faire la sélection. Lorsqu'elles sont évaluées, ces techniques sont plus performantes pour la sélection de petits objets dans des environnements denses, mais elles sont également moins performantes que le *Raycasting* pour la sélection de cibles de tailles importantes. Grossman et Balakrishnan ont montré que le Depth Ray est plus performant que le Lock Ray, Flower Ray ou Smart Ray en raison du temps plus court nécessaire pour la phase de désambigüisation (Grossman & Balakrishnan, 2006). Au lieu d'avoir à désambigüiser entre différentes cibles en plusieurs étapes, une autre approche consiste à ajouter des degrés de liberté supplémentaires au *Raycasting* pour aider à ajuster la sélection de la cible tout en manipulant le rayon.

2.2 Ajout de degrés de liberté supplémentaires

Grossman et Balakrishnan ont simplement ajouté un degré de liberté supplémentaire (ddl) en ajoutant un curseur fixe au milieu d'un rayon (Grossman & Balakrishnan, 2006), ce qui s'est avéré être la technique la plus efficace pour sélectionner des cibles sur un petit écran volumétrique. Cependant, dans le contexte d'environnements immersifs tels que les casques RV, l'utilisation de cette technique nécessiterait des déplacements importants de l'utilisateur pour désambigüiser les cibles. Au lieu d'avoir un curseur fixé sur un rayon, Ro *et al.* ont introduit la possibilité d'ajuster la profondeur du rayon en utilisant les déplacements relatifs d'un doigt sur l'écran tactile d'un smartphone (Ro et al., 2017). Cependant, leur technique n'a pas été comparée à d'autres techniques et la fonction de transfert utilisée pour contrôler la longueur du rayon n'est pas détaillée. Des études récentes combinent les mouvements des mains avec le suivi de la tête et des yeux pour le pointage en réalité augmentée, comme le Pinpointing (Kytö, Ens, Piumsomboon, Lee, & Billinghamurst, 2018). Cette technique est beaucoup plus précise que les techniques basées uniquement sur le regard. Cependant, elle présente des inconvénients communs avec le *Raycasting* : sensibilité à l'occultation, tremblement des mains et précision des périphériques d'entrée.

2.3 Techniques de facilitation du pointage

Diverses stratégies ont été étudiées pour faciliter le pointage. Par exemple, le pointage sémantique élargit les cibles dans l'espace moteur (Blanch et al., 2004) Cette technique a été

conçue pour le pointage 2D, mais une autre étude l'a étendue à la 3D en utilisant une souris d'ordinateur sur écran standard (Elmqvist & Fekete, 2008). Cette technique améliore les performances de pointage dans des environnements qui présentent une faible densité de cibles mais est affectée par des cibles qui se trouvent sur le chemin du curseur (cibles distractives) (Casiez, Roussel, Vanbelleghem, & Giraud, 2011). Une autre stratégie consiste à remplacer le pointage par des gestes symboliques (Gupta, Pietrzak, Yau, Roussel, & Balakrishnan, 2017). Elle permet d'atténuer les problèmes dus aux gestes de pointage, mais elle n'est pas adaptée à des cibles arbitraires. À l'inverse, des études proposent des gestes 3D pour pointer des cibles sur des écrans 2D mais ces résultats sont difficilement applicables dans le cadre de la sélection de cibles 3D (Nancel et al., 2013; Vogel & Balakrishnan, 2005).

L'une des techniques de facilitation du pointage les plus efficaces en 2D consiste à sélectionner la cible la plus proche du curseur. Par exemple, le Bubble Cursor affiche un disque (une bulle) centré sur un curseur de souris dont le rayon est ajusté en fonction de la distance à la cible la plus proche (Grossman & Balakrishnan, 2005). Cette technique est particulièrement efficace lorsque la densité des cibles est faible, quelle que soit leur taille. Le principal inconvénient reste le retour visuel introduit par la bulle qui change constamment de rayon. Guillon *et al.* ont évalué l'impact de plusieurs retours visuels sur la performance du Bubble Cursor et ont constaté qu'une simple mise en surbrillance de la cible la plus proche est efficace (Guillon, Leitner, & Nigay, 2015).

Vanacken *et al.* ont développé une version 3D du Bubble Cursor, appelée 3D Bubble, en utilisant une technique de main virtuelle pour contrôler un pointeur 3D (Vanacken et al., 2007). Leur technique utilise une sphère semi-transparente en 3D englobant la cible la plus proche. Ils montrent que le Depth Ray est plus efficace que la 3D Bubble, elle-même plus efficace que *Raycasting*. De même, Vickers a défini un cube sensible autour d'un curseur 3D manipulé par une manette (Vickers, 1972). Lorsqu'un objet se trouve dans le cube sensible, le curseur saute sur l'objet.

En résumé, les techniques de désambiguïsation semblent efficaces pour sélectionner de petites cibles dans des environnements denses, mais elles augmentent globalement le temps de sélection en raison des étapes supplémentaires qu'elles introduisent. Les techniques de sélection utilisant la proximité des cibles semblent efficaces mais elles nécessitent l'utilisation d'un retour d'information approprié, notamment en 3D. Inspiré par la 3D Bubble, le Depth Ray et le rayon de longueur ajustable introduit par Ro *et al.*, nous proposons *RayCursor*, une technique combinant plusieurs des avantages que ces techniques offrent, sans introduire d'étapes de désambiguïsation supplémentaires que de nombreuses techniques requièrent.

3 RAYCURSOR

Nous décrivons la conception de *RayCursor*, une amélioration de *Raycasting*. Tout d'abord, nous ajoutons un curseur sur le rayon, que l'utilisateur peut manipuler. Ensuite, nous ajoutons une stratégie consistant à sélectionner la cible la plus proche du curseur, de la même manière que Bubble Cursor (Grossman & Balakrishnan, 2005; Vanacken et al., 2007). Nous évoquons différentes variantes de retours visuels, inspirés des travaux de Guillon *et al.* pour la 2D (Guillon et al., 2015). Ensuite, nous décrivons différentes fonctions de transfert possibles pour le contrôle du curseur. Enfin, nous détaillons les techniques de filtrage que nous avons utilisées pour réduire les effets des tremblements du rayon, en utilisant le *1€ Filter* (Casiez, Roussel, & Vogel, 2012).

3.1 Ajout d'un curseur sur le rayon

L'idée d'un curseur contrôlable a été introduite par Ro *et al.* pour sélectionner des cibles dans des environnements de réalité augmentée en utilisant un téléphone mobile (Ro et al., 2017). Cependant, cette technique n'a pas été conçue pour faciliter le choix de petites cibles. Au lieu d'utiliser un téléphone portable, l'utilisateur de *RayCursor* effectue des déplacements vers l'avant ou vers l'arrière sur le pavé tactile d'un contrôleur Vive situé sous son pouce. Il serait possible d'implémenter cette technique sur n'importe quel autre contrôleur à 6 ddl ayant

au moins un degré de liberté supplémentaire tel qu'une molette.

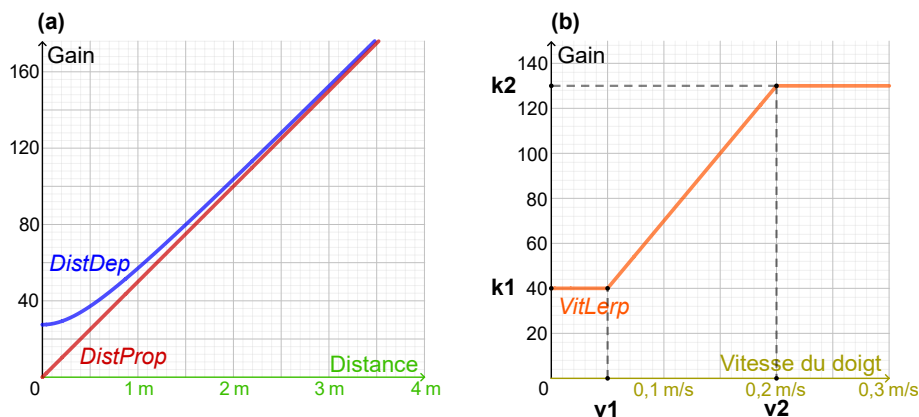
3.2 Fonction de transfert pour le contrôle du curseur

La fonction de transfert, qui calcule les déplacements du curseur, est un aspect essentiel de la technique d'interaction. En effet, des recherches antérieures ont montré que la fonction de transfert influence les performances de pointage (Casiez & Roussel, 2011). Nous considérons deux variables pour la conception de la fonction de transfert :

- v_{pad} la vitesse du point de contact sur le pavé tactile en m/s . Il s'agit d'une variable habituelle pour des fonctions de transfert non linéaires.
- d_{cur} la distance entre la manette et le curseur en m . Il s'agit de la distance dans l'espace virtuel, c'est-à-dire la distance entre la représentation virtuelle de la manette et le curseur. Nous utilisons cette variable car la vitesse du curseur projetée à l'écran paraît d'autant plus importante que celui-ci est proche de l'utilisateur. Nous émettons l'hypothèse que des fonctions de transfert exploitant d_{cur} influencent le temps de sélection.

La vitesse du curseur est donc $v_{cur} = g(v_{pad}, d_{cur}) \times v_{pad}$. Nous proposons plusieurs fonctions de transfert, que nous évaluons dans le cadre d'une expérience dédiée.

Figure 2. Courbes des fonctions de gain en fonction de (a) la distance manette-curseur et de (b) la vitesse du doigt sur le pad.



3.2.1 Fonction de transfert dépendant de la vitesse du doigt

Les fonctions de transfert non linéaires usuelles adaptent la vitesse du curseur à celle du périphérique d'entrée (Casiez & Roussel, 2011), de manière à tirer parti des phases balistique et corrective d'une tâche de pointage. Pendant la phase balistique, l'utilisateur cherche à se déplacer aussi vite que possible pour se rapprocher de la cible alors que dans la phase corrective, l'utilisateur a généralement besoin d'être précis pour sélectionner la cible. Pour ce faire, les fonctions de transfert non linéaires utilisent un gain plus faible à basse vitesse et un gain plus élevé à haute vitesse. Nous avons conçu cette fonction de transfert comme une interpolation linéaire bornée (voir Figure 2 (b)) :

$$VitLerp(v_{pad}) = \begin{cases} k_1 & \text{si } x \leq v_1 \\ k_2 & \text{si } x \geq v_2 \\ k_1 + \frac{k_2 - k_1}{v_2 - v_1}(v_{pad} - v_1) & \text{sinon} \end{cases}$$

3.2.2 Fonction de transfert dépendant de la position du curseur

Avec une fonction de transfert qui utilise un gain constant, la vitesse du curseur affiché à l'écran semble dépendre de sa distance à l'utilisateur dans l'espace virtuel : plus rapide quand il est proche, et plus lente quand il est éloigné. Nous avons d'abord conçu une fonction de transfert

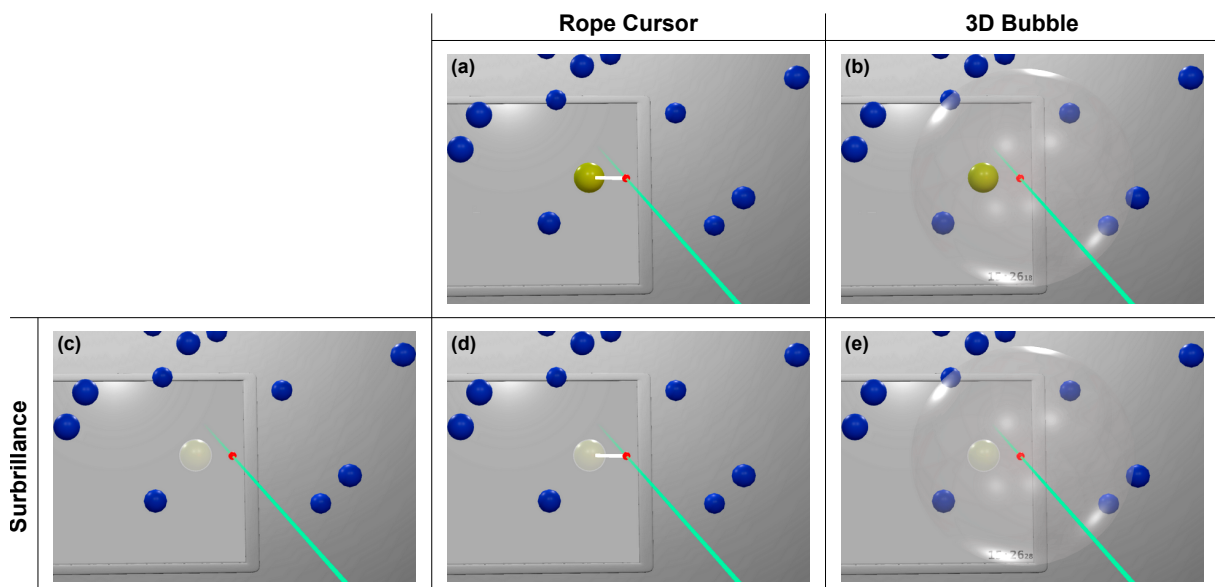
basée sur un gain qui est proportionnel à la distance du curseur par rapport au contrôleur dans l'espace virtuel, pour atténuer ce problème (voir *DistProp* dans la Figure 2 (a)). Bien que cette fonction permette de rendre visuellement cohérente la vitesse de déplacement du curseur à des positions moyennes et éloignées, elle provoque des déplacements trop lents lorsque le curseur est très proche du contrôleur. Nous avons donc développé une deuxième version de cette fonction appelée *DistDep* (voir Figure 2 (a)), qui garantit un gain minimum pour des distances plus proches. La formule de cette fonction est la suivante, avec k un facteur de proportionnalité, d_{cur} la distance entre le contrôleur et le curseur, et d une constante correspondant à une distance estimée entre l'œil de l'utilisateur et la manette :

$$DistDep(d_{cur}) = k \times \sqrt{d_{cur}^2 + d^2}$$

3.3 Retours visuels

Les travaux de Guillon *et al.* ont montré une influence du retour visuel rétroactif sur l'efficacité du pointage avec la technique *Bubble Cursor* pour le pointage 2D (Guillon *et al.*, 2015). L'étude de Vanacken *et al.* sur le 3D Bubble ne prenait en compte que le retour visuel de la bulle (Vanacken *et al.*, 2007). Nous adaptons à la 3D les retours visuels conçus par Guillon *et al.* pour l'interaction 2D. Nous décrivons ci-dessous ces retours visuels selon deux dimensions : la mise en évidence de la cible et la représentation de la distance entre le curseur et la cible la plus proche.

Figure 3. RayCursor avec différents retours visuels. Le curseur est rouge et le rayon est cyan. (c,d,e) Surbrillance : la cible la plus proche est mise en surbrillance. (a,d) Rope Cursor (Guillon *et al.*, 2015) : une ligne blanche relie le curseur à la cible la plus proche. (b,e) 3D Bubble (Vanacken *et al.*, 2007) : une bulle 3D centrée sur le curseur, adapte son rayon pour englober la cible la plus proche.



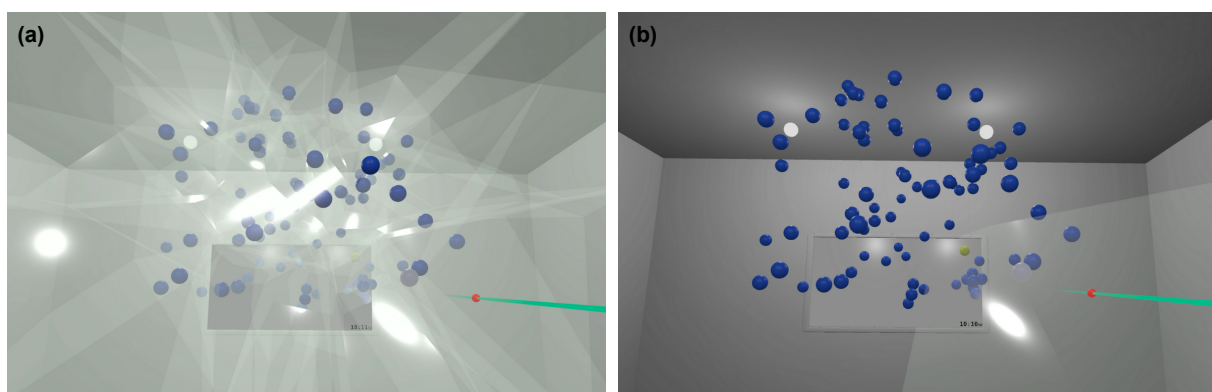
3.3.1 Surbrillance de la cible

Dans notre implémentation, la mise en évidence de la cible la plus proche consiste à appliquer une couleur plus claire que les autres cibles (Figure 3 (c,d,e)). L'avantage de ce retour visuel par rapport aux suivants est que l'encombrement visuel est minimal. Nous pouvons également le combiner avec d'autres retours visuels.

3.3.2 Représentation de la distance cible-curseur

Représenter la distance entre le curseur et la cible la plus proche est susceptible d'aider à déterminer la cible qui peut être sélectionnée à l'aide du curseur. Il existe plusieurs manières de représenter cette distance. Dessiner une sphère semi-transparente, centrée sur le curseur et dont le rayon est la distance *cible-curseur*, est une adaptation 3D du *Bubble Cursor* (Figure 3 (b,e)). Guillon *et al.* ont également proposé le *Rope Cursor*, que nous adaptons à la 3D en affichant un segment blanc entre le curseur et la cible la plus proche (Figure 3 (a,d)). Ce retour visuel provoque moins d'encombrement visuel que la bulle. Une autre proposition de Guillon *et al.* est la région de Voronoï de chaque cible. Ces régions représentent les zones d'influence de chaque cible, en prenant en compte les distances entre elles. Bien que nous ayons implémenté cette rétroaction visuelle sous forme de volumes semi-transparentes, nous l'avons rejetée parce que l'encombrement visuel créé la rend difficile à utiliser (Figure 4).

Figure 4. RayCursor avec le retour visuel représentant les régions de Voronoï. (a) Affichage simultané de toutes les régions de Voronoï. (b) Affichage de la région de Voronoï de la cible la plus proche.



Dans la section 5.1, nous présentons une étude comparative entre les retours visuels suivants ainsi que des combinaisons entre eux (Figure 3) : *3D Bubble*, *Rope Cursor* et *Surbrillance*.

3.4 Filtrage du rayon

La précision de la sélection des petites cibles est un problème courant avec *Raycasting*. Elle est due à la fois à des tremblements de la main et à des entrées bruitées. Bowman *et al.* (Bowman, Wingrave, Campbell, Ly, & Rhoton, 2002) décrivent aussi un autre problème, qu'ils ont nommé « l'effet Heisenberg de l'interaction spatiale », qui correspond au décalage accidentel du rayon lors de l'action de sélection (appui sur un bouton). Nous proposons de réduire ces problèmes en filtrant le rayon. Nous appliquons le filtrage, à la fois pour *Raycasting* et notre technique, en utilisant le *1€ Filter* car il est rapide, simple à régler, et offre un bon compromis entre réduction du bruit et introduction de latence. Nous avons conçu deux modes de filtrage. Dans le premier mode, nous filtrons l'orientation du rayon mais uniquement pour calculer l'intersection avec des objets virtuels. Le rayon affiché n'est pas filtré. Nous appelons ce mode $1\epsilon_M$, car le rayon est seulement filtré dans l'espace moteur. Dans le second mode, nous filtrons l'orientation du rayon dans les espaces moteur et visuel ($1\epsilon_{VM}$). L'avantage de $1\epsilon_M$ par rapport à $1\epsilon_{VM}$ est que l'utilisateur ne subit absolument aucun temps de latence supplémentaire mais bénéficie quand même du filtrage pour le point d'intersection du rayon avec des objets. En revanche, avec $1\epsilon_{VM}$, l'utilisateur dispose d'un retour visuel plus cohérent sur les résultats de ses actions.

Nous décrivons une étude comparative de ces deux modes de filtrage et un *Raycasting* non filtré dans la section 5.3.

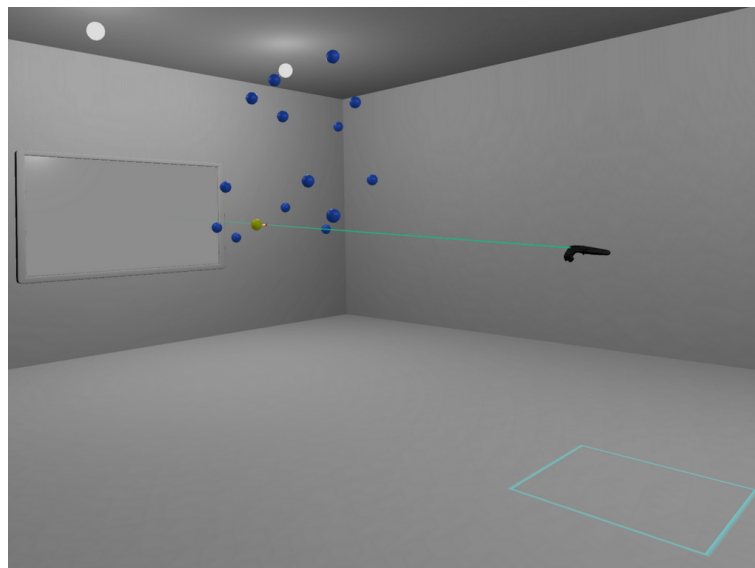
4 CONFIGURATION EXPÉRIMENTALE GÉNÉRALE

Toutes les expériences de cet article se basent sur la même configuration telle que décrite dans cette section.

4.1 Matériel

Pour nos expériences, nous avons utilisé un ordinateur équipé d'un casque de réalité virtuelle *HTC Vive* (Vive, 2019). Les participants manipulaient le contrôleur HTC Vive avec leur main dominante, et n'avaient pas le droit d'utiliser l'autre. Le rayon était contrôlé par les 6 degrés de liberté de mouvement du contrôleur, et le curseur était manipulé par le pouce sur le pavé tactile (Figure 1). La sélection était effectuée au moment de l'appui sur la gâchette du contrôleur, en utilisant l'index. L'application de l'expérience a été développée en *C#* avec *Unity 3D* et la librairie *SteamVR*.

Figure 5. Scène 3D utilisée pour les expériences. Le participant était debout dans le carré cyan tracé au sol. Les instructions étaient affichées sur l'écran virtuel en face de lui. Les cibles étaient positionnées entre l'utilisateur et l'écran virtuel.



4.2 Tâches

Sauf indication contraire, les participants devaient se conformer aux instructions suivantes dans toutes les expériences. Ils devaient se tenir au milieu d'une pièce carrée de 7 m de côté. La position exacte était indiquée par un carré sur le sol de 70 cm de côté. Les instructions de l'expérience étaient affichées sur un écran virtuel devant l'utilisateur (Figure 5). Nous avons demandé aux participants de sélectionner des cibles à différentes positions, tailles et densités, en fonction des conditions expérimentales. Toutes les cibles étaient visibles lorsque l'utilisateur regardait le mur virtuel avec l'écran virtuel. Afin de permettre une comparaison équitable des temps d'exécution et des taux d'erreur entre les conditions, nous avons généré, avant les expériences, une séquence unique de cibles pour chaque condition, utilisée pour tous les participants, toutes les techniques et tous les blocs. L'ordre des techniques était contrebalancé entre les participants en utilisant un carré latin.

Toutes les cibles étaient bleues sauf celle qui devait être sélectionnée, qui était jaune. Le participant pouvait sélectionner une cible en appuyant sur la gâchette. Une vibration de 10 ms du contrôleur informait le participant lorsque la bonne cible est sélectionnée. Lorsqu'un participant ne sélectionnait pas la bonne cible, le contrôleur vibrait pendant 200 ms, la cible correcte clignotait en vert, la mauvaise cible sélectionnée clignotait en rouge (sauf lorsque l'utilisateur ne sélectionnait aucune cible avec *Raycasting*), et l'essai était marqué comme une erreur. Les par-

participants ne pouvaient pas passer à la cible suivante avant d'avoir correctement sélectionné la cible en cours. Le taux d'erreur a été calculé comme étant le rapport entre le nombre total d'essais pour lesquels la bonne cible n'a pas été sélectionnée à la première tentative et le nombre total d'essais. Nous avons demandé aux participants de maintenir un taux d'erreur d'environ 4% afin d'équilibrer leur compromis vitesse/précision. Le taux d'erreur était affiché sur l'écran virtuel pendant les pauses.

Les participants répondaient à un questionnaire NASA-TLX pour chaque technique ou fonction testée. Nous utilisons ce questionnaire comme indicateur de la charge de travail de l'utilisateur. À la fin de l'expérience, l'utilisateur devait aussi choisir sa technique préférée.

5 ÉTUDE DES CARACTÉRISTIQUES DE RAYCURSOR

Nous présentons une série d'expériences pour évaluer chacune des caractéristiques de *RayCursor*. Nous commençons par l'étude de différents retours visuels pour indiquer la cible à sélectionner, car elle nous a semblé être la caractéristique la plus susceptible d'affecter les performances. Nous évaluons ensuite les différentes fonctions de transfert que nous avons conçues et évaluons l'effet du filtrage du rayon.

5.1 Retours visuels

Nous avons proposé plusieurs retours visuels, basés sur les travaux existants sur des techniques similaires d'interaction 2D et 3D (Grossman & Balakrishnan, 2005 ; Guillon et al., 2015 ; Vanacken et al., 2007). L'étude de Guillon *et al.* sur le cas 2D a conclu que la mise en surbrillance de la cible la plus proche est le retour le plus efficace, tout en limitant l'encombrement visuel. Nous décrivons une étude similaire qui compare des versions 3D de ces retours visuels, avec *Raycasting* comme condition témoin. Étant donné les travaux existants en 2D, notre hypothèse est que la mise en surbrillance de la cible la plus proche est le retour le plus efficace en 3D (H1). Nous avons utilisé la fonction de transfert *DistDep* pour contrôler le curseur, et le rayon n'a été filtré dans aucune des conditions.

5.1.1 Méthodologie

Douze participants (tous droitiers, age moyen=26, $\sigma = 4,3$) ont pris part à cette expérience. Pour deux d'entre eux il s'agissait de leur première expérience de réalité virtuelle.

Nous avons utilisé un protocole expérimental intra-sujet, avec les facteurs : TECHNIQUE, DENSITÉ de cibles, TAILLE des cibles et BLOC. Les 6 techniques sont *Raycasting* (RC) en tant que référence, ainsi que *RayCursor* avec les 5 retours visuels décrits Figure 3 : *RopeCursor* (Rope), *Surbrillance* (HL), *3DBubble* (Bub), *3DBubble+Surbrillance* (Bub+HL), *RopeCursor+Surbrillance* (Rope+HL). L'ordre des techniques a été équilibré entre les participants à l'aide d'un carré latin. Les 2 tailles de cibles étaient $S_{Grande} = 8$ cm et $S_{Petite} = 4$ cm. Les deux densités de cibles étaient de 15 cibles (D_{Faible}) et 40 cibles ($D_{Elevée}$). Toutes les cibles étaient positionnées de manière pseudo-aléatoire dans une sphère de 2m de diamètre, dont le centre était positionné à 2m devant le participant.

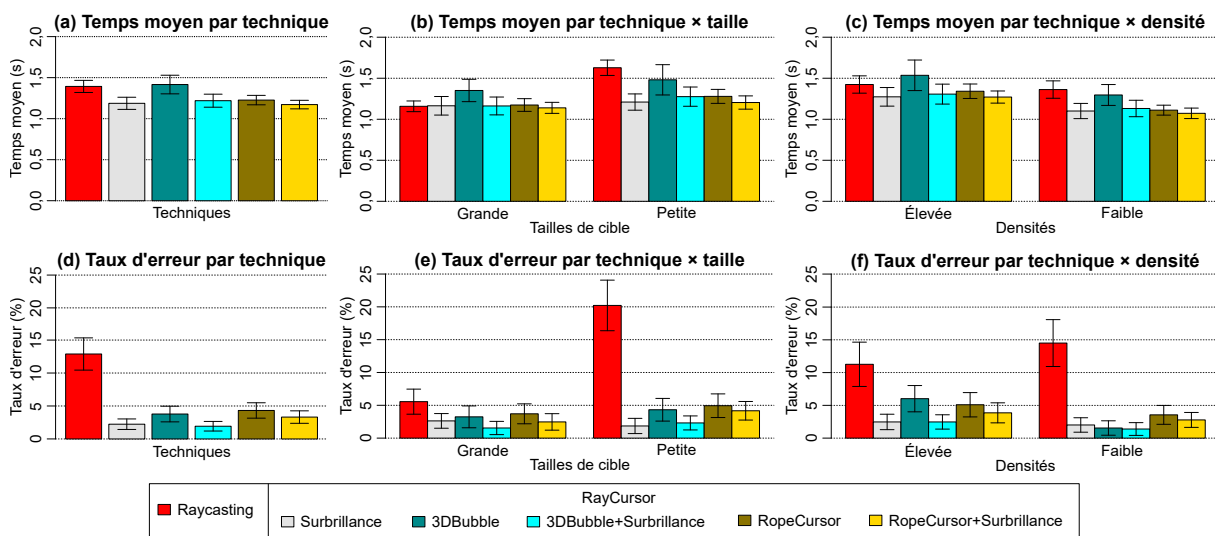
Les participants pouvaient prendre des pauses entre chaque bloc. Le protocole expérimental de l'expérience était donc : 12 participants \times 6 TECHNIQUES \times 3 BLOCS \times 2 DENSITÉS \times 2 TAILLES \times 10 cibles = 8640 essais au total. L'expérience durait environ 30 minutes par participant.

5.1.2 Résultats

Nos deux variables dépendantes sont le temps de sélection et le taux d'erreur.

Temps de sélection Dans cette analyse, le temps de sélection correspond au temps écoulé entre deux sélections. Par conséquent, le premier essai de chaque séquence de 10 cibles est écarté, de même que les essais entraînant une erreur. Nous avons appliqué une transformation

Figure 6. Temps moyen et taux d'erreur relevés lors de l'expérience sur les retours visuels, avec représentation des intervalles de confiance à 95 %.



de Box-Cox avec $\lambda = -1,2$ pour corriger les problème de normalité des résidus des temps de sélection (Box & Cox, 1964).

Une ANOVA² révèle un effet significatif de BLOC ($F_{1,2,12,7} = 21,2, p < 0,001, \eta_G^2 = 0,04$). Les comparaisons par paires montrent des différences significatives entre le bloc 1 et les deux suivants ($p < 0,003$, Bloc 1 : 1,45s, 2 : 1,29s, 3 : 1,25s). Nous supposons que cette différence est due à un effet d'apprentissage. Nous supprimons donc le premier bloc du reste de l'analyse.

Les analyses suivantes révèlent un effet significatif de TECHNIQUE ($F_{5,55} = 5,4, p < 0,001, \eta_G^2 = 0,1$). Les comparaisons par paires montrent un effet significatif entre *Raycasting* et toutes les autres techniques sauf *3DBubble* (RC : 1,39s, HL : 1,19s, Bub : 1,42s, Bub+HL : 1,22s, Rope : 1,23s, Rope+HL : 1,17s, $p < 0,027$).

Les analyses montrent un effet significatif de TAILLE ($F_{1,11} = 108,3, p < 0,001, \eta_G^2 = 0,07$) et une interaction TECHNIQUE x TAILLE ($F_{5,55} = 28,1, p < 0,001, \eta_G^2 = 0,05$). Les comparaisons par paires ne montrent un effet significatif entre les techniques que pour les *petites* cibles. *Raycasting* est significativement plus lent ($p < 0,0004$) que toutes les autres techniques, à part *3DBubble* (RC : 1,63s, Bub : 1,48s, HL : 1,21s, Rope : 1,28s, Rope+HL : 1,20s).

Nous observons également un effet significatif de DENSITÉ ($F_{1,11} = 176,3, p < 0,001, \eta_G^2 = 0,11$) et une interaction TECHNIQUE x DENSITÉ ($F_{5,55} = 5,2, p < 0,001, \eta_G^2 = 0,01$). Pour la *faible* densité, nous observons une différences significative ($p < 0,01$) entre *Raycasting* et toutes les autres techniques, excepté *3DBubble* (RC : 1,36s, Bub : 1,30s, Bub+HL : 1,13s, HL : 1,10s, Rope : 1,11s, Rope+HL : 1,07s).

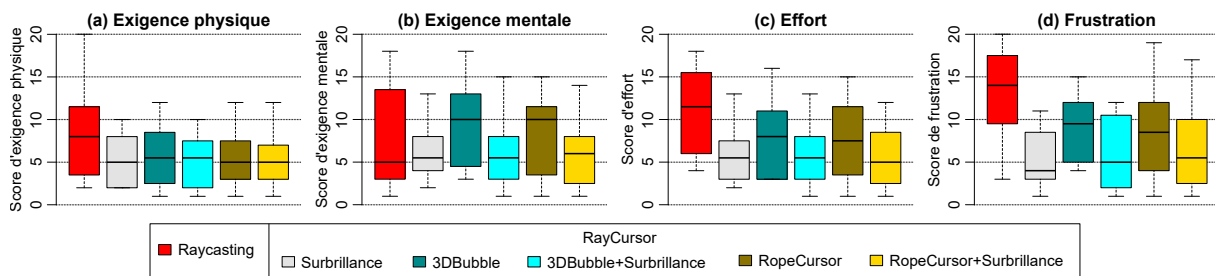
Taux d'erreur Le taux d'erreur global est de 4,7%, sachant que les participants avaient pour instruction de rester autour de 4%. Les données ont été pré-traitées à l'aide d'une Aligned Rank Transform (ART) pour tenir compte de leur distribution non normale (Wobbrock, Findlater, Gergle, & Higgins, 2011). L'ANOVA à mesures répétées révèle un effet significatif de TECHNIQUE ($F_{5,829} = 30,3, p < 0,001$). Les comparaisons par paires montrent un effet significatif entre *Raycasting* (12,9%) et toutes les autres techniques. (HL : 2,2%, Bub : 3,8%, Bub+HL : 1,9%, Rope : 4,3%,

2. Toutes les analyses statistiques des 4 expériences ont été effectuées avec R, avec $\alpha = 0,05$ et la correction de Greenhouse-Geisser a été appliquée aux degrés de liberté quand la sphéricité était violée. Nous avons utilisé la correction de Bonferroni, dans laquelle la p -value était multipliée par le nombre de comparaisons. Les analyses statistiques détaillées sont disponibles à l'adresse ns.inria.fr/loki/raycastor.

Rope+HL : 3,3%, $p < 0,006$). Nous observons aussi un effet significatif de TAILLE ($F_{1,829} = 140,7$, $p < 0,0001$) et une interaction TECHNIQUE \times TAILLE ($F_{5,829} = 26,4$, $p < 0,0001$). Les comparaisons par paires montrent que le taux d'erreur de *Raycasting* est significativement plus  lev e pour des *petites* cibles (20,2%) que pour des *grandes* cibles (5,6%). Pour les *petites* cibles, *Raycasting* a un taux d'erreur plus  lev e que pour toutes les autres techniques ($< 5\%$, $p < 0,0001$). Les analyses montrent aussi une interaction TECHNIQUE \times DENSIT  ($F_{5,829} = 5,8$, $p < 0,0001$). Les comparaisons par paires montrent que *Raycasting* a un taux d'erreur plus  lev e pour une *faible* densit  que les autres techniques ($< 3,6\%$, $p < 0,0001$).

Questionnaire NASA-TLX L'analyse de Friedman sur les r ponses 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 r v le que le niveau de frustration est significativement plus  lev e pour la technique *Raycasting* (m diane=14) que pour la technique *Surbrillance* (m diane=4, $p = 0,03$). Il n'y a pas d'autre diff rence significative observ e pour les tests post-hocs.

Figure 7. Diagrammes en bo te des r ponses aux crit res « Exigence physique », « Exigence mentale », « Effort » et « Frustration » au questionnaire NASA-TLX de la premi re exp rience, pour chaque technique.



Pr f rences des participants Chaque participant a pu choisir sa technique pr f r e parmi les 6 test es. Les r sultats montrent que les participants ont pr f r  les techniques avec la mise en surbrillance activ e (3   4 votes pour chaque) tandis que les autres, *Raycasting* inclus, n'ont re u qu'un seul vote ou aucun.

5.1.3 Discussion

Les r sultats montrent que tous les retours visuels utilis s avec *RayCursor* sont globalement plus efficaces que *Raycasting*. En particulier, *RayCursor* est globalement plus rapide pour les petites cibles tout en gardant un faible taux d'erreur. La mise en surbrillance de la cible la plus proche est l'un des effets visuels les plus efficaces (H1). Appliqu    *RayCursor*, il est aussi jug  moins frustrant que *Raycasting* par les utilisateurs. L'affichage d'une bulle ne permet pas d'am liorer les performances de *RayCursor* par rapport au *Raycasting*, en particulier pour les petites cibles ou les environnements denses, certainement en raison de l'encombrement visuel plus  lev  qu'elle introduit. Ces r sultats sont en accord avec l' tude de Guillon *et al.* concernant les retours visuels en 2D.

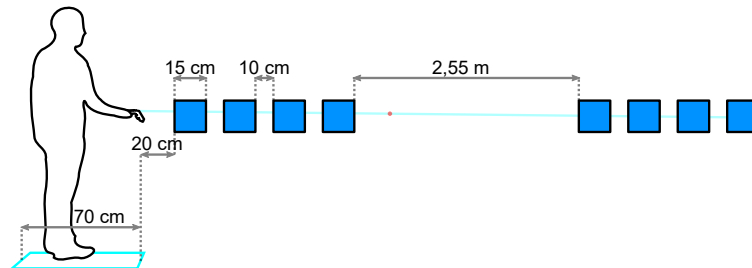
5.2 Fonction de transfert du curseur

Notre hypoth se est que le choix de la fonction de transfert utilis e influence les performances de *RayCursor* (H2). Pour v rifier H2, nous avons compar  la performance de *VitLerp*, *DistProp*, et *VitLerp* \times *DistDep* (une combinaison entre *VitLerp* et *DistDep*) avec 9 participants (tous droitiers, age moyen=27,7, $\sigma = 6,5$). Les param tres de chaque fonction de transfert ont  t  ajust s empiriquement pour maximiser les performances. Pour *VitLerp*, les param tres  taient $k_1 = 30$, $k_2 = 150$, $v_1 = 0,05$ et $v_2 = 0,15$. Pour *DistProp*, le param tre  tait $k = 50$. La

combinaison de *VitLerp* et *DistDep* (*VLDD*) consiste à multiplier les deux gains. Pour *VLDD*, les paramètres étaient $k_1 = 20$, $k_2 = 100$, $v_1 = 0,05$, $v_2 = 0,15$, $k = 1$ et $d = 0,55$.

Les variables indépendantes étaient FONCTIONS de transfert (*VitLerp*, *DistProp*, *VLDD*), POSITION de la cible (*Proche*, *Loin*), DISTANCE avec la cible précédente (*Courte*, *Longue*), et BLOC (3).

Figure 8. Positionnement des cibles dans l'espace virtuel pour l'expérience sur les fonctions de transfert.

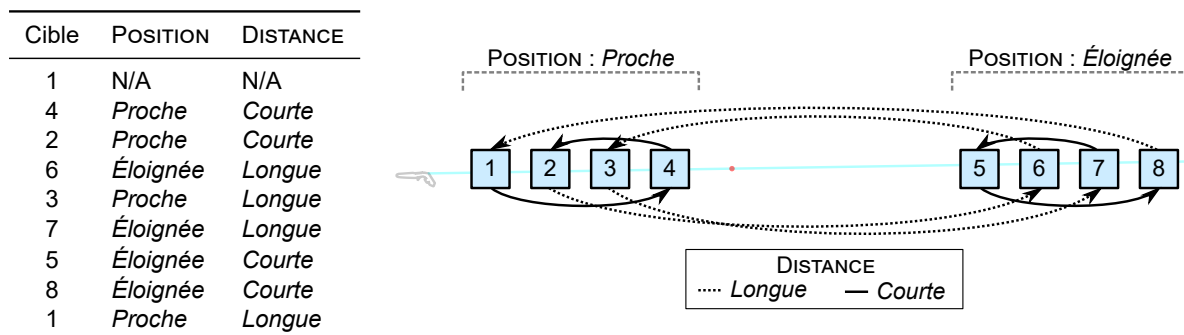


Huit cibles sous forme de cubes étaient positionnées tel que décrit Figure 8. L'utilisateur devait placer son contrôleur de manière à être aligné aux cibles, tout en restant dans le carré au sol.

Pour chaque technique, les participants enchaînaient 3 BLOCS, dans lesquels ils devaient effectuer 4 enchaînements de 9 essais. Après chaque enchaînement, 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.

L'ordre des cibles pour les 4 enchaînements était défini de manière à ce qu'il y ait le même nombre d'essais pour chaque condition de POSITION×DISTANCE, tel que montré en exemple Figure 9. Les 4 enchaînements étaient identiques pour tous les blocs, toutes les fonctions de transfert et tous les participants. L'ordre des fonctions de transfert était contrebalancé entre les participants en utilisant un carré latin.

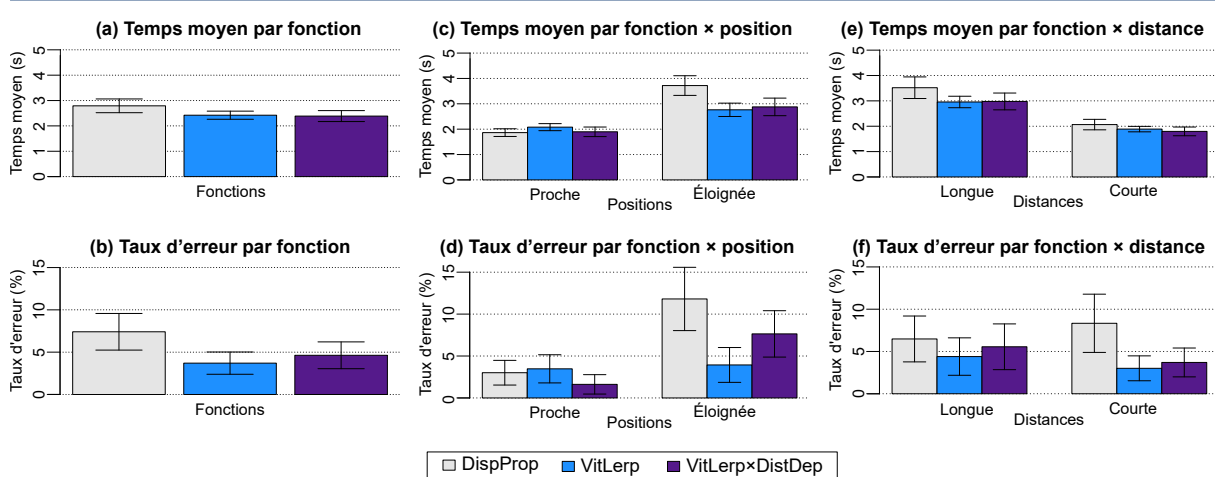
Figure 9. Exemple d'enchaînement de cibles pour parcourir les niveaux des variables indépendantes Position et Distance.



Nous avons en résumé un plan expérimental de 9 participants × 3 FONCTIONS × 3 BLOCS × 2 POSITIONS × 2 DISTANCES × 9 essais = 2 916 essais au total. L'expérience durait environ 20 minutes par participant.

Temps de sélection Nous avons utilisé une transformation Box-Cox ($\lambda = -0,4$) pour corriger les problèmes de normalité des résidus des temps de sélection. Une ANOVA à mesures répétées montre un effet significatif de FONCTION ($F_{2,16} = 3,8$, $p = 0,046$, $\eta_G^2 = 0,07$). Les comparaisons par paires montrent une différence significative marginale entre *DistProp* et *VLDD* (*DP* : 2,79s; *VLDD* : 2,39s; $p = 0,05$). *DistProp* est légèrement plus lente en moyenne que *VLDD*. Nous observons aussi un effet significatif de POSITION ($F_{1,8} = 223$, $p < 0,001$, $\eta_G^2 = 0,55$) et une interaction

Figure 10. Temps moyen et taux d'erreur pour l'expérience sur les fonctions de transfert, avec représentation des intervalles de confiance à 95 %.



de FONCTION×POSITION ($F_{2,16}=25,1, p<0,001, \eta_G^2=0,16$). Aucune différence significative n'a été observée pour une position *Proche*. Cependant, pour la position *Loin*, les comparaisons par paires montrent une différence significative entre *DistProp* (3,72s) et les deux autres fonctions (VL : 2,76s ; VLDD : 2,88s ; $p<0,002$).

Taux d'erreur Nous avons pré-traité les données à l'aide d'une ART pour prendre en compte leur distribution non-normale. Une ANOVA à mesures répétées montre un effet significatif de FONCTION ($F_{2,304} = 4,55, p < 0,0004$). Les comparaisons par paires montrent un effet significatif entre *DistProp* et *VitLerp* (DP : 7,4% ; VL : 3,7% ; $p < 0,002$). Les participants font généralement moins d'erreurs avec *VitLerp* qu'avec *DistProp*. Nous observons aussi un effet significatif de POSITION ($F_{1,304} = 29,6, p < 0,001$), et une interaction de FONCTION×POSITION ($F_{2,304} = 12,9, p < 0,0001$). Les comparaisons par paires ne montrent pas de différence significative entre les fonctions de transfert pour des cibles *Proches*. Cependant, nous observons une différence significative pour des cibles éloignées de l'utilisateur entre *DistProp* et *VitLerp* (DP : 11,8% ; VL : 3,9% ; $p < 0,008$).

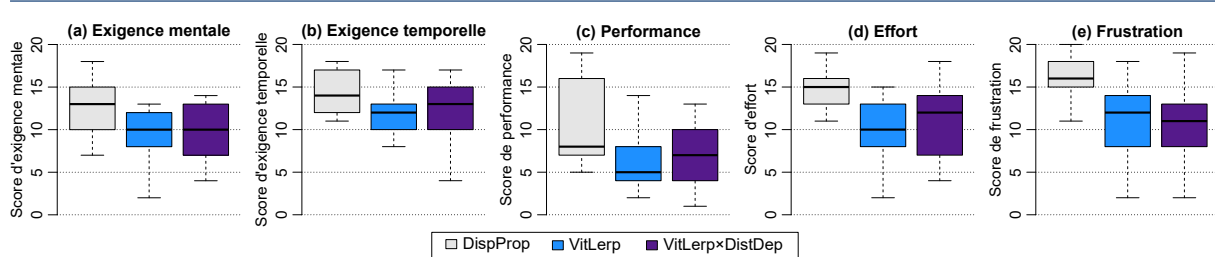
Questionnaire NASA-TLX L'analyse de Friedman sur les réponses au questionnaire NASA-TLX indique qu'il y a un effet significatif pour l'exigence mentale ($\chi^2(2)=8,24, p=0,01$), l'exigence temporelle ($\chi^2(2) = 7,93, p = 0,02$), la performance ($\chi^2(2) = 8,63, p = 0,01$), l'effort ($\chi^2(2) = 11,0, p = 0,004$) et la frustration ($\chi^2(5) = 12, p = 0,002$). L'analyse post-hoc de Wilcoxon révèle que le niveau de frustration est significativement plus élevé pour *DistProp* (médiane= 16) que pour *VitLerp×DistDep* (médiane= 11, $p = 0,045$). Il n'y a pas d'autre différence significative observée pour les tests post-hocs.

Préférences des participants Chaque participant a pu choisir sa technique préférée parmi les 3 testées. Les résultats montrent que les participants ont préféré les fonctions de gain *VitLerp* et *VitLerp×DistDep* (4 à 5 votes pour chaque) tandis que la fonction *DistProp* n'a reçu aucun vote.

5.2.1 Discussion

En conclusion, la fonction de transfert influence les performances de *RayCursor* (H2 confirmée) : *VitLerp* est plus rapide et plus efficace que *DistProp* pour les cibles éloignées de l'utilisateur et est globalement la fonction la plus efficace.

Figure 11. Diagrammes en boîte des réponses aux critères « Exigence mentale », « Exigence temporelle », « Performance », « Effort » et « Frustration » au questionnaire NASA-TLX de la dernière expérience, pour chaque fonction.



5.3 Filtrage du rayon

Dans les expériences précédentes, les petites cibles étaient difficiles à sélectionner à cause des tremblements de la main et des entrées bruitées. Dans cette expérience, nous voulons vérifier l'hypothèse que filtrer le rayon réduit les erreurs de sélection pour *Raycasting* (H3). Bien que le filtrage a déjà été utilisé dans la littérature pour filtrer un rayon (Kopper et al., 2011 ; Vogel & Balakrishnan, 2005), il était utilisé dans un contexte où ce rayon intersecte un écran physique. Nous n'avons pas trouvé de travaux évaluant formellement les effets du filtrage sur le taux d'erreur et le temps de sélection de *Raycasting*. Vogel et al. et Kopper et al. ont tous les deux utilisé ce qui correspond à une version primitive du *1€ Filter* (Casiez et al., 2012). Nous l'avons aussi utilisé car il semble fournir un bon compromis entre un bon filtrage et une faible latence. Dans cette expérience, nous considérons uniquement des cibles éloignées, ce qui est la situation la plus difficile.

5.3.1 Méthodologie

Neuf participants (tous droitiers, age moyen=26,9, $\sigma = 6,25$) ont pris part à cette expérience. Tous avaient déjà eu une expérience en réalité virtuelle.

Nous avons utilisé un protocole expérimental intra-sujet. Les variables indépendantes sont : TECHNIQUE et BLOC. Les trois techniques sont décrites section 3.4 : *Raycasting* non filtré (RC), *Raycasting* filtré dans l'espace moteur ($1€_M$) et *Raycasting* filtré dans l'espace visuel et moteur ($1€_{VM}$). L'ordre des 3 techniques a été contre-balancé entre les participants en utilisant un carré latin. Nous avons fixé les paramètres du *1€ Filter* à $min_{cutoff} = 0,1$ et $\beta = 50$ suite à des évaluations empiriques, de manière à réduire les tremblements tout en minimisant la latence. La cible sélectionnable était mise en surbrillance, en accord avec l'expérience sur le retour visuel. Pour chaque technique, les participants devaient réaliser 5 BLOCS de 20 cibles.

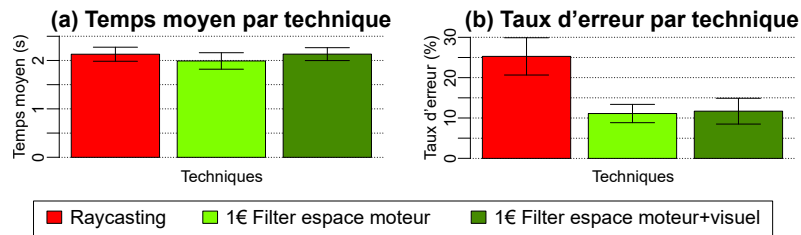
En résumé, le protocole expérimental était : 9 participants \times 3 TECHNIQUES \times 5 BLOCS \times 20 essais = 2700 essais au total. L'expérience durait environ 15 minutes par participant.

5.3.2 Résultats

Nous analysons le temps de sélection et le taux d'erreur pour étudier les effets du filtrage. Il y avait une interruption entre les blocs, nous avons donc retiré le premier essai de chaque bloc de l'analyse. Pour le temps de sélection, nous avons également écarté les essais ayant donné lieu à une erreur.

Temps de sélection Nous avons appliqué une transformation de Box-Cox ($\lambda = -0.2$) pour corriger les problèmes de normalité des résidus des temps de sélection. Une ANOVA à mesures répétées² ne montre aucun effet de BLOC ($F_{4,32} = 2,2, p = 0,09$) ni d'effet de TECHNIQUE ($F_{1,2,9,8} = 2,0, p = 0,18$).

Figure 12. Temps moyen et taux d'erreur résultants de l'expérience sur le 1€ Filter, avec représentation des intervalles de confiance à 95 %.

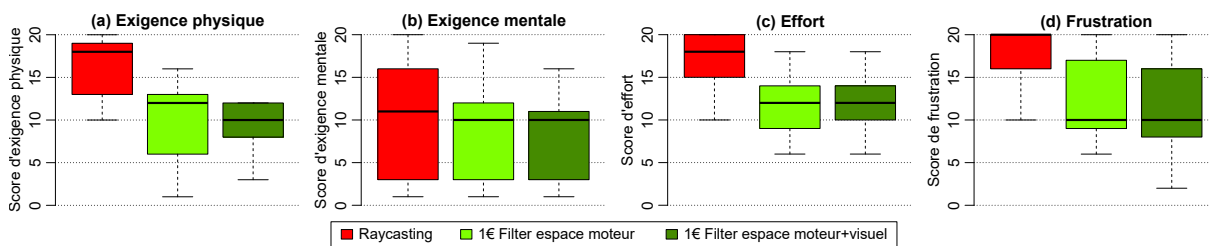


Taux d'erreur Le taux d'erreur global est de 16,02%, alors que les participants devaient rester autour de 4%. Nous avons appliqué une transformation logarithmique pour corriger les résidus de taux d'erreur non normaux. Les analyses ne montrent pas d'effet significatif de BLOC ($F_{4,32} = 0,94, p > 0,05$). Cependant, nous observons un effet significatif de TECHNIQUE ($F_{2,16} = 12,2, p < 0,001, \eta_G^2 = 0,21$). Les comparaisons par paires montrent un effet significatif entre le *Raycasting* sans filtre (25,2%) et les autres techniques (1€_{VM} : 11,7%, 1€_M : 11,1%, $p < 0,0001$).

Questionnaire NASA-TLX L'analyse de Friedman sur les réponses au questionnaire NASA-TLX ne révèle pas d'effet significatif pour l'exigence physique ($\chi^2(2) = 15,8, p = 0,0003$), l'exigence mentale ($\chi^2(2) = 11,1, p = 0,003$), l'effort ($\chi^2(2) = 13,7, p = 0,001$) et la frustration ($\chi^2(2) = 11,0, p = 0,004$).

L'analyse post-hoc de Wilcoxon révèle que le niveau d'exigence physique est significativement plus élevé pour *Raycasting* (médiane=18) que pour 1€_{VM} (médiane=10; $p = 0,015$), et que le niveau d'effort est significativement plus élevé pour *Raycasting* (médiane=18) que pour 1€_M (médiane=12; $p = 0,049$). Les tests post-hocs ne montrent pas d'autre différence significative.

Figure 13. Diagrammes en boîte des réponses aux critères « Exigence physique », « Exigence mentale », « Effort » et « Frustration » au questionnaire NASA-TLX de la deuxième expérience pour chaque technique.



Préférences des participants Chaque participant a pu choisir sa technique préférée parmi les 3 testées. Les résultats montrent que les participants ont largement préféré la technique avec le rayon filtré par 1€ Filter dans l'espace visuel et moteur (7 votes) suivi par le rayon filtré par 1€ Filter dans l'espace moteur seulement (2 votes) et aucun vote pour le *Raycasting*.

5.3.3 Discussion

Les résultats montrent que filtrer le rayon avec le 1€ Filter réduit les erreurs de sélection de plus de 50% (H3 confirmée). Lorsque le rayon est filtré dans l'espace moteur, le fait que le rayon soit aussi filtré dans l'espace visuel ou non n'influence pas le taux d'erreur. Cependant, nos discussions avec les participants révèlent que 7/9 préfèrent que l'espace visuel et l'espace moteur soient tous les deux filtrés. Qualitativement, nous avons observé que le filtrage réduit l'exigence physique, ainsi que l'effort nécessaire pour réaliser la tâche.

5.4 Discussion

Parmi les retours visuels que nous avons proposés pour *RayCursor*, les plus efficaces d'entre eux montrent la cible qui sera sélectionnée sans ambiguïté. C'est le cas notamment pour toutes celles qui utilisent la mise en surbrillance et le Rope Cursor. Bien que plusieurs des retours visuels aient des performances similaires, nous suggérons de simplement mettre en surbrillance la cible la plus proche, puisqu'il s'agit du retour visuel avec le moins d'encombrement visuel. Ceci suggère également que la technique 3DBubble (Vanacken et al., 2007) bénéficierait certainement aussi de ce type de retour visuel.

Nos études sur la fonction de transfert pour contrôler le curseur ont montré qu'une fonction non linéaire de la vitesse du curseur était la plus efficace. Ceci est en accord avec la littérature et les fonctions de transfert courantes pour les souris d'ordinateur et les pavés tactiles de bureau (Casiez & Roussel, 2011). Nous avons proposé une fonction simplifiée, paramétrée avec deux seuils de vitesse d'entrée et deux gains extrêmes. Le développeur de l'application peut adapter ces paramètres en fonction de la taille du pavé tactile et des distances que l'utilisateur doit parcourir. Nous avons observé de manière informelle que les fréquents mouvements vers l'avant et vers l'arrière du curseur sur de longues distances ralentissent la technique. Cela nous a amené l'idée de combiner *Raycasting* et *RayCursor*, avec un mécanisme pour téléporter le curseur près d'un point d'intérêt, réduisant ainsi le temps de déplacement.

Enfin, nous avons montré que le filtrage du rayon diminue les erreurs de sélection. C'est à la fois bénéfique pour *Raycasting* et *RayCursor*. Bien que nos expériences aient montré que la présence ou non du filtrage dans l'espace visuel n'influence pas les performances lorsque le rayon était filtré dans l'espace moteur, la préférence des utilisateurs tend à suggérer qu'il est préférable de filtrer les deux espaces.

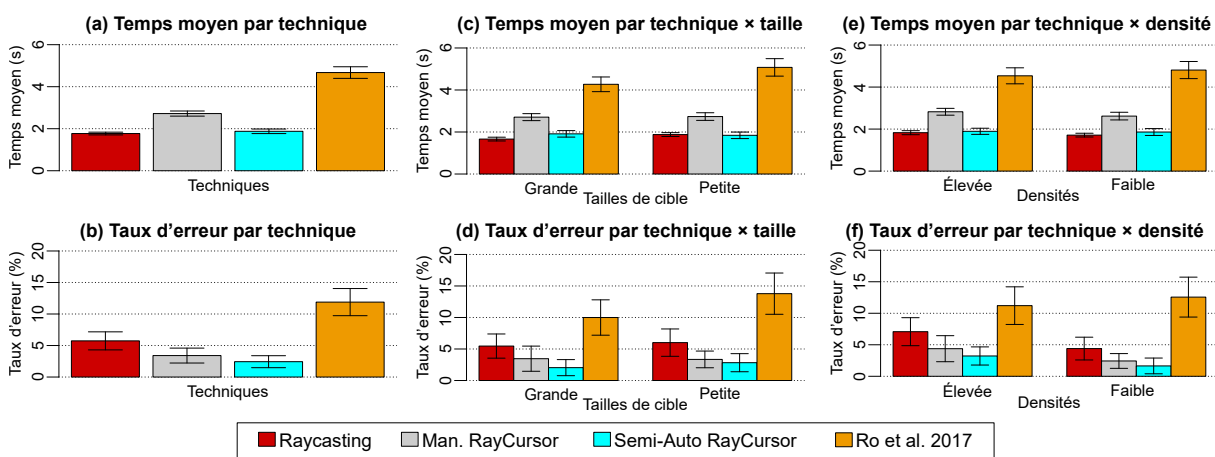
Dans ce qui suit, nous présentons une dernière expérience comparant 1) *RayCursor* utilisant la mise en surbrillance, la fonction de transfert non-linéaire qui utilise la vitesse et le rayon filtré ; 2) la même version de *RayCursor* mais utilisant un positionnement semi-automatique du curseur le long du rayon ; 3) *Raycasting* utilisant un filtrage ; et 4) une technique récente de la littérature, qui utilise un rayon de longueur ajustable (Ro et al., 2017).

6 ÉTUDE COMPARATIVE

Nous avons évalué toutes les caractéristiques de *RayCursor* et proposé des paramètres optimaux. Dans cette section, nous comparons deux variantes de la technique avec *Raycasting* et la technique la plus proche de la nôtre dans la littérature (Ro et al., 2017). Une version préliminaire de cette étude est disponible dans (Baloup, Pietrzak, & Casiez, 2019). Les deux variantes sont un contrôle manuel du curseur et un contrôle semi-automatique du curseur. La version semi-automatique (Figure 1, c-f) est un hybride entre *Raycasting* et *RayCursor*. Le contrôle automatique du curseur est activé lorsque l'utilisateur ne touche pas le pavé tactile. Dans ce mode, lorsque le rayon intersecte une cible, le curseur se déplace automatiquement sur le rayon au point d'intersection. Si le rayon sort de la cible, la cible reste sélectionnable à l'aide du mécanisme de sélection par proximité. Si une autre cible est plus proche du curseur ou intersecte le rayon, cette nouvelle cible est sélectionnée. Si l'utilisateur pose son doigt sur le pavé tactile, il peut contrôler la position du curseur comme dans la version manuelle de *RayCursor* (le contrôle automatique est désactivé). Si l'utilisateur relâche le pavé tactile plus d'une seconde, la technique revient au comportement automatique du curseur. Le délai d'une seconde laisse le temps à l'utilisateur de relever et reposer son doigt pour déplacer le curseur sur une longue distance.

Raycasting et les deux variantes de *RayCursor* sont filtrées en utilisant le *1€ Filter* avec les réglages précédemment définis. En effet, lorsque *Raycasting* n'est pas filtré, son taux d'erreur est beaucoup plus élevé que pour les autres techniques selon notre étude dans la section 5.3. Cette expérience nous permettra de mieux comprendre les améliorations de la performance du *Raycasting* lorsqu'il est filtré. Nous avons aussi implémenté la technique de Ro et al. car c'est la technique la plus proche de la nôtre (Ro et al., 2017). Cela nous permet également de

Figure 14. Temps moyen et taux d'erreur résultants de l'étude comparatif, avec représentation des intervalles de confiance à 95 %.



mesurer l'avantage direct de l'utilisation du mécanisme de sélection par proximité. Comme Ro *et al.* n'ont pas détaillé la fonction de transfert utilisée pour contrôler la longueur de leur rayon, nous avons utilisé pour leur technique la même fonction de transfert utilisée pour *RayCursor*. Notre hypothèse est que *RayCursor* avec contrôle semi-automatique du curseur est plus rapide et moins sujette aux erreurs que les autres techniques testées (H4).

6.1 Méthodologie

Douze participants (1 gaucher, age moyen=27,6, $\sigma = 5,8$) ont pris part à l'expérience. Deux d'entre eux n'ont jamais utilisé de système de réalité virtuelle avant l'expérience. Six d'entre eux ont participé à une des expériences précédentes. Le temps entre chaque expérience était d'au moins 4 semaines, ce qui nous permet de supposer que l'effet d'apprentissage était négligeable. Le plan expérimental intra-sujet a par ailleurs l'avantage de réduire les différences individuelles.

Nous avons utilisé un protocole expérimental intra-sujet, avec comme facteur : TECHNIQUE, DENSITÉ de cibles, TAILLE des cibles et BLOC. Les 4 techniques utilisées sont : *Raycasting* filtré avec 1ϵ Filter (RC_f), *RayCursor* avec le contrôle manuel du curseur ($ManRCur$), *RayCursor* avec le contrôle semi-automatique du curseur ($AutoRCur$) et la technique de Ro *et al.* (Ro *et al.*, 2017) (Ro). L'ordre des techniques a été contrebalancé entre les participants grâce à un carré latin. Les 2 tailles de cible étaient $S_{Grande} = 8cm$ et $S_{Petite} = 4cm$ de diamètre. Les 2 densités étaient de 30 cibles (D_{Faible}) et 60 cibles ($D_{Elevée}$). Les cibles étaient réparties équitablement dans 2 sphères de 60 cm de diamètre, en face de l'utilisateur, et centrées à 80 cm du sol. La première sphère était à 1m en face de l'utilisateur et la deuxième à 4m. Toutes les cibles étaient dans le champ de vision du participant lorsqu'il regardait en direction de l'écran virtuel. Le participant devait sélectionner alternativement une cible dans la sphère proche et une autre dans la sphère éloignée. Cette condition correspond à la situation la plus défavorable pour $ManRCur$ car le participant devait constamment déplacer le curseur sur de longues distances. Ceci a pour but de mettre en avant les limites de *RayCursor*.

Les participants avaient la possibilité de faire une pause entre les techniques. Le design de l'expérience était donc : 12 participants \times 4 TECHNIQUES \times 3 BLOCS \times 2 DENSITÉS \times 2 TAILLES \times 9 cibles = 5184 essais au total. L'expérience durait environ 30 min par participant.

6.2 Résultats

La figure 14 montre les taux d'erreur et les temps de sélection. Nous discutons aussi des préférences des participants.

6.2.1 Temps de sélection

Dans cette analyse, le temps de sélection correspond au temps entre deux sélections. De ce fait, le premier essai de chaque séquence de 10 cibles est ignoré, ainsi que les essais erronés. Nous avons aussi supprimé les valeurs extrêmes, lorsque les temps de sélection étaient au dessus de $mean + 3 \times sd$ pour chaque technique.

Nous avons appliqué une transformation de Box-Cox avec $\lambda = -0,3$ pour corriger les résidus de temps de sélection non normaux. Une ANOVA à mesures répétées montre un effet significatif de BLOC ($F_{2,22} = 21,3$, $p = 0,001$, $\eta_G^2 = 0,05$), avec les comparaisons par paires révélant une différence significative entre les blocs 1 et 3 ($p < 0,016$, Bloc 1 : 2,96s, 2 : 2,72s, 3 : 2,59s). Ces résultats ne suggérant pas clairement d'effet d'apprentissage ou de fatigue, nous avons gardé tous les blocs pour les analyses suivantes.

Nous observons aussi un effet significatif de TECHNIQUE ($F_{1,4,15,9} = 120,0$, $p < 0,0001$, $\eta_G^2 = 0,71$). Des comparaisons par paires montrent un effet significatif entre *Ro* et les autres techniques (RC_f : 1,77s, *ManRCur* : 2,72s, *AutoRCur* : 1,88s, *Ro* : 4,67s, $p < 0,0001$), et aussi entre *ManRCur* et les autres techniques ($p < 0,0001$). La technique de *Ro et al.* est significativement plus lente que *ManRCur*, qui est elle-même, significativement plus lente que les deux autres techniques. Avec le *RayCursor* semi-automatique, l'utilisateur a, dans la plupart des cas, juste à viser la bonne cible avec le rayon. Ces techniques sont donc plus rapides que les autres qui obligent l'utilisateur à déplacer le curseur le long du rayon pour atteindre la cible.

L'analyse montre un effet significatif de TAILLE ($F_{1,11} = 11,0$, $p < 0,007$, $\eta_G^2 = 0,02$) et une interaction de TECHNIQUE \times TAILLE ($F_{3,33} = 11,8$, $p < 0,0001$, $\eta_G^2 = 0,03$). Des comparaisons par paires montrent seulement une différence significative pour la technique de *Ro*, entre les *Petites* et les *Grandes* cibles ($Ro_{Petites}$: 5,08s, $Ro_{Grandes}$: 4,27s, $p = 0,05$). Nous observons que notre implémentation de la technique de *Ro et al.* est plus lente pour les *Petites* cibles. Nous trouvons aussi un effet significatif de DENSITÉ ($F_{1,11} = 5,8$, $p < 0,035$, $\eta_G^2 = 0,01$) et une interaction de TECHNIQUE \times DENSITÉ ($F_{3,33} = 3,4$, $p < 0,03$, $\eta_G^2 = 0,01$). Une analyse post-hoc révèle que le temps de sélection augmente pour *Raycasting* avec des densités *Élevée* ($p < 0,001$, *Élevée* : 1,82s, *Faible* : 1,71s).

6.2.2 Taux d'erreur

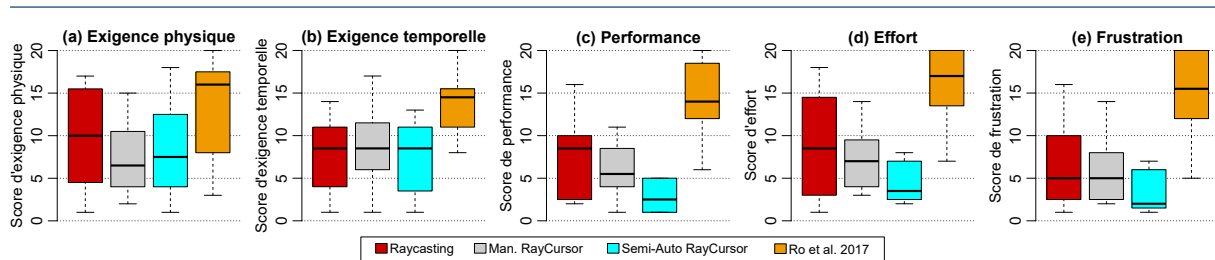
Le taux d'erreur global est de 5,9%, sachant que les participants avaient pour instruction de rester autour de 4% d'erreur. Les données ont été pré-traitées avec une ART pour prendre en compte leur distribution non normale. Une ANOVA à mesures répétées ne montre pas d'effet de BLOC ($F_{2,562} = 2,9$, $p > 0,05$). Cependant, elle montre un effet significatif de TECHNIQUE ($F_{3,549} = 25,5$, $p < 0,0001$). Des comparaisons par paires montrent des différences entre la technique de *Ro et al.* (11,9%) et toutes les autres techniques (RC_f : 5,7%, *ManRCur* : 3,8%, *AutoRCur* : 2,4%, $p < 0,0001$), aussi bien qu'entre RC_f et *AutoRCur* ($p = 0,013$). Le taux d'erreur élevé de la technique de *Ro et al.* est probablement dû au manque de sélection par proximité, comparé au *RayCursor* manuel. Aussi, le *RayCursor* semi-automatique a un plus faible taux d'erreur que *Raycasting*. Nous expliquons cet effet par l'utilisation du principe de sélection par proximité lorsque le rayon dévie de la cible visée lorsque qu'on essaye de le sélectionner. Nous relevons aussi un effet significatif de TAILLE ($F_{1,549} = 11,9$, $p < 0,001$) montrant un taux d'erreur plus élevé pour des petites cibles (*Petite* : 6,5%, *Grande* : 5,2%). Aucun effet de DENSITÉ ($F_{1,549} = 0,01$, $p > 0,05$) n'a été trouvé.

6.2.3 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 ($\chi^2(3) = 10,7$, $p = 0,01$), l'exigence temporelle ($\chi^2(3) = 12,6$, $p = 0,006$), la performance ($\chi^2(3) = 18,9$, $p = 0,0003$), l'effort ($\chi^2(3) = 15,9$, $p = 0,001$) et la frustration ($\chi^2(3) = 23,2$, $p < 0,0001$). L'analyse post-hoc de Wilcoxon révèle que le score d'exigence temporelle est significativement plus élevé pour *Ro* (médiane = 14,5) que pour les autres techniques (médianes = 8,5 $p < 0,04$). Elle montre aussi que l'estimation des participants de leur performance est significativement plus basse pour *Ro* (médiane = 14) que pour les autres techniques

(médianes $\leq 8,5$ $p < 0,02$). Enfin elle montre que le score d'effort est significativement plus élevé pour *Ro* (médiane = 17) que pour les autres techniques (médianes $\leq 8,5$ $p < 0,03$) et que le score de frustration est également significativement plus élevé pour *Ro* (médiane = 15,5) que pour les autres techniques (médianes ≤ 5 $p < 0,02$). Il n'y a pas d'autre différence significative observée pour les tests post-hocs.

Figure 15. Diagrammes en boîte des réponses aux critères « Exigence physique », « Exigence temporelle », « Performance », « Effort » et « Frustration » au questionnaire NASA-TLX de la dernière expérience, pour chaque technique. L'échelle de performance du NASA-TLX est inversée afin que les valeurs élevées représentent des contre-performances sur toutes les échelles.



6.2.4 Préférence des participants

À la fin de l'expérience, les participants devaient classer les techniques selon leurs préférences. Un test de Friedman révèle un effet significatif de TECHNIQUE pour les préférences utilisateurs ($\chi^2(3) = 25,3$, $p < 0,0001$). Une analyse post-hoc de Wilcoxon montre une différence significative entre la technique de *Ro et al.* (rang médian = 4) et toutes les autres techniques (rangs médians : *RC_f* : 2, *ManRCur* : 2, *AutoRCur* : 1, $p < 0,0001$) et entre le *RayCursor* semi-automatique et manuel ($p = 0,041$). Ces résultats subjectifs sont en accord avec nos analyses de taux d'erreur et de temps de sélection. Une majorité de participant ont placé le *RayCursor* semi-automatique en premier (9/12 participants).

6.3 Discussion

Cette expérience montre que *RayCursor* semi-automatique obtient des temps de sélection similaires à ceux du *Raycasting* filtré, pour différentes tailles et densités de cibles. Cependant, le *RayCursor* semi-automatique réduit significativement les taux d'erreur sur le *Raycasting* filtré dans les différentes conditions. Ceci montre l'efficacité 1) du *1€ Filter* pour réduire les mouvements bruités; 2) de la sélection par proximité pour les cibles en 3D; et 3) du placement semi-automatique du curseur le long du rayon. Lors de la sélection de cibles distantes, le *RayCursor* semi-automatique aurait pu être affecté négativement par le rayon qui intersectait des cibles proches et faisait soudainement sauter le curseur, mais ce ne fut pas le cas. Si des sauts se produisaient pour des cibles d'éloignement similaire, l'utilisation d'une fonction d'hystérésis aiderait à résoudre le problème. Notre *RayCursor* manuel a obtenu des performances de temps inférieures par rapport aux techniques mentionnées précédemment, certainement en raison du déplacement incessant du curseur nécessaire pour déplacer le curseur en avant et en arrière d'un essai à l'autre. Cependant, en comparant le *RayCursor* manuel à la technique de *Ro et al.*, elle montre clairement les avantages de la sélection par proximité.

7 CONCLUSION

Nous avons présenté *RayCursor*, une nouvelle technique d'interaction pour la sélection de cibles 3D dans des environnements immersifs. Cette technique est une amélioration de *Raycasting* qui utilise un curseur sur le rayon pour sélectionner la cible la plus proche. L'affichage d'une bulle est le retour visuel typique d'une telle technique (Grossman & Balakrishnan, 2005; Vanacken et al., 2007). Cependant Guillon et al. ont montré que, dans un contexte 2D, mettre en surbrillance la cible la plus proche est plus efficace, et produit moins d'encombrement vi-

suel (Guillon, Leitner, & Nigay, 2014 ; Guillon et al., 2015). Nous avons étendu leurs résultats à l'interaction 3D, avec des conclusions similaires.

Malgré l'existence de travaux antérieurs sur le filtrage du rayon, nous avons décrit la première étude d'un *Raycasting* filtré en évaluant formellement ses avantages. Nous avons montré que le filtrage du rayon réduit fortement les erreurs de sélection. Ceci est à la fois bénéfique pour *Raycasting* et *RayCursor*.

Nos résultats démontrent également que les fonctions de transfert comme celles utilisées sur les interfaces de bureau sont efficaces pour le contrôle d'un curseur sur un rayon. Nous recommandons d'utiliser une fonction sigmoïde qui dépend de la vitesse du curseur. Cependant, d'autres contrôleurs VR n'ont pas de touchpad mais un joystick. Des études plus approfondies sont nécessaires afin d'appliquer *RayCursor* à ce type de contrôleur. En particulier il faudrait dans ce cas déterminer la fonction de transfert adéquate.

Nous montrons aussi qu'une technique hybride entre *Raycasting* et *RayCursor* a le plus faible taux d'erreur, tout en étant aussi rapide que *Raycasting*. Néanmoins, nous n'avons pas évalué l'efficacité de notre technique sur des cibles de formes non sphériques. En effet, avec des formes non sphériques, la manière dont est calculée la distance entre le curseur et chaque cible pourrait affecter les performances de la technique.

En 3D, la sélection précède souvent la manipulation d'un objet 3D. Un avantage secondaire et important de notre technique est sa capacité à manipuler l'objet le long du rayon une fois qu'il est sélectionné, par exemple pour le rapprocher de l'utilisateur, ce que le *Raycasting* standard ne permet pas de faire. Les travaux futurs se concentreront sur l'utilisation de *RayCursor* pour permettre à la fois la sélection et la manipulation d'objets 3D dans des environnements immersifs.

REMERCIEMENTS

Ce travail a été partiellement financé par le projet Inria IPL Avatar.

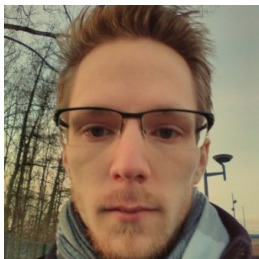
RÉFÉRENCES

- ▶ Argelaguet, F., & Andujar, C. (2013). A survey of 3d object selection techniques for virtual environments. *Computers and Graphics*, 37(3), 121 - 136. Consulté sur <http://www.sciencedirect.com/science/article/pii/S0097849312001793> doi : 10.1016/j.cag.2012.12.003
- ▶ Baloup, M., Pietrzak, T., & Casiez, G. (2019). Raycursor : A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 chi conference on human factors in computing systems* (pp. 101:1–101:12). New York, NY, USA : ACM. doi : 10.1145/3290605.3300331
- ▶ Blanch, R., Guiard, Y., & Beaudouin-Lafon, M. (2004). Semantic pointing : Improving target acquisition with control-display ratio adaptation. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 519–526). New York, NY, USA : ACM. doi : 10.1145/985692.985758
- ▶ Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I. (2004). *3d user interfaces : Theory and practice*. Redwood City, CA, USA : Addison Wesley Longman Publishing Co., Inc.
- ▶ Bowman, D. A., Wingrave, C. A., Campbell, J., Ly, V., & Rhoton, C. (2002). Novel uses of pinch gloves(tm) for virtual environment interaction techniques. *Virtual Reality*, 6(3), 122–129.
- ▶ Box, G. E., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 211–252.
- ▶ Cashion, J., Wingrave, C., & LaViola Jr, J. J. (2012). Dense and dynamic 3d selection for game-based virtual environments. *IEEE transactions on visualization and computer graphics*, 18(4), 634–642. doi : 10.1109/TVCG.2012.40
- ▶ Casiez, G., & Roussel, N. (2011). No more bricolage !: Methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th annual acm symposium on user interface software and technology* (pp. 603–614). New York, NY, USA : ACM. doi : 10.1145/2047196.2047276
- ▶ Casiez, G., Roussel, N., Vanbelleghem, R., & Giraud, F. (2011). Surfpad : Riding towards targets on a squeeze film effect. In *Proceedings of the sigchi conference on human factors in computing*

- systems (pp. 2491–2500). New York, NY, USA : ACM. doi : 10.1145/1978942.1979307
- ▶ Casiez, G., Roussel, N., & Vogel, D. (2012). 1€ filter : A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 2527–2530). New York, NY, USA : ACM. doi : 10.1145/2207676.2208639
 - ▶ de Haan, G., Koutek, M., & Post, F. H. (2005). Intenselect : Using dynamic object rating for assisting 3d object selection. In *Proceedings of the 11th eurographics conference on virtual environments* (pp. 201–209). Aire-la-Ville, Switzerland, Switzerland : Eurographics Association. doi : 10.2312/EGVE/IPT_EGVE2005/201-209
 - ▶ Delamare, W., Coutrix, C., & Nigay, L. (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* (pp. 89–98). New York, NY, USA : ACM. doi : 10.1145/2493190.2493232
 - ▶ Elmqvist, N., & Fekete, J.-D. (2008, mai). Semantic pointing for object picking in complex 3d environments. In *Proceedings of graphics interface 2008* (pp. 243–250). Toronto, Ont., Canada : Canadian Information Processing Society. Consulté sur <http://dl.acm.org/citation.cfm?id=1375714.1375755>
 - ▶ Grossman, T., & Balakrishnan, R. (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* (pp. 281–290). New York, NY, USA : ACM. doi : 10.1145/1054972.1055012
 - ▶ Grossman, T., & Balakrishnan, R. (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* (pp. 3–12). New York, NY, USA : ACM. doi : 10.1145/1166253.1166257
 - ▶ Guiard, Y., Blanch, R., & Beaudouin-Lafon, M. (2004). Object pointing : A complement to bitmap pointing in guis. In *Proceedings of graphics interface 2004* (pp. 9–16). School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada : Canadian Human-Computer Communications Society. Consulté sur <http://dl.acm.org/citation.cfm?id=1006058.1006060>
 - ▶ Guillon, M., Leitner, F., & Nigay, L. (2014). Static Voronoi-Based Target Expansion Technique for Distant Pointing. In F. Garzotto et al. (Eds.), *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI 2014)* (p. 41-48). Como, Italy : ACM. Consulté sur <https://hal.inria.fr/hal-01006840> doi : 10.1145/2598153.2598178
 - ▶ Guillon, M., Leitner, F., & Nigay, L. (2015). Investigating visual feedforward for target expansion techniques. In *Proceedings of the 33rd annual acm conference on human factors in computing systems* (pp. 2777–2786). New York, NY, USA : ACM. doi : 10.1145/2702123.2702375
 - ▶ Gupta, A., Pietrzak, T., Yau, C., Roussel, N., & Balakrishnan, R. (2017). Summon and select : Rapid interaction with interface controls in mid-air. In *Proceedings of the 2017 acm international conference on interactive surfaces and spaces* (pp. 52–61). New York, NY, USA : ACM. doi : 10.1145/3132272.3134120
 - ▶ Hinckley, K., Pausch, R., Goble, J. C., & Kassell, N. F. (1994). A survey of design issues in spatial input. In *Proceedings of the 7th annual acm symposium on user interface software and technology* (pp. 213–222). New York, NY, USA : ACM. doi : 10.1145/192426.192501
 - ▶ Kopper, R., Bacim, F., & Bowman, D. A. (2011, March). Rapid and accurate 3d selection by progressive refinement. In *2011 ieee symposium on 3d user interfaces (3dUI)* (pp. 67–74). Washington, DC, USA : IEEE Computer Society. doi : 10.1109/3DUI.2011.5759219
 - ▶ Kytö, M., Ens, B., Piumsomboon, T., Lee, G. A., & Billinghamurst, M. (2018). Pinpointing : Precise head-and eye-based target selection for augmented reality. In *Proceedings of the 2018 chi conference on human factors in computing systems* (pp. 81:1–81:14). New York, NY, USA : ACM. doi : 10.1145/3173574.3173655
 - ▶ Liang, J., & Green, M. (1994). Jdcad : A highly interactive 3d modeling system. *Computers & Graphics*, 18(4), 499 - 506. doi : 10.1016/0097-8493(94)90062-0
 - ▶ Mine, M. R. (1995). *Virtual environment interaction techniques* (Rapport technique). Chapel Hill, NC, USA : University of North Carolina.
 - ▶ Nancel, M., Chapuis, O., Pietriga, E., Yang, X.-D., Irani, P. P., & Beaudouin-Lafon, M. (2013). High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 831–840). New York, NY, USA : ACM. doi : 10.1145/2470654.2470773

- ▶ Ren, G., & O'Neill, E. (2013, mai). 3d selection with freehand gesture. *Computers & Graphics*, 37(3), 101–120. doi : 10.1016/j.cag.2012.12.006
- ▶ Ro, H., Chae, S., Kim, I., Byun, J., Yang, Y., Park, Y., & Han, T. (2017, Oct). A dynamic depth-variable ray-casting interface for object manipulation in ar environments. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (p. 2873-2878). IEEE Systems, Man, and Cybernetics Society. doi : 10.1109/SMC.2017.8123063
- ▶ Schmidt, G., Baillot, Y., Brown, D. G., Tomlin, E. B., & Swan, J. E. I. (2006, March). Toward disambiguating multiple selections for frustum-based pointing. In *Proceedings of the 3d user interfaces* (pp. 87–94). Washington, DC, USA : IEEE Computer Society. doi : 10.1109/VR.2006.133
- ▶ Steinicke, F., Ropinski, T., & Hinrichs, K. (2004). Object selection in virtual environments with an improved virtual pointer metaphor. In *Computer vision and graphics : International conference, ICCV 2004* (pp. 320–326). Warsaw, Poland : Springer Netherlands. doi : 10.1007/1-4020-4179-9_46
- ▶ Vanacken, L., Grossman, T., & Coninx, K. (2007, March). Exploring the effects of environment density and target visibility on object selection in 3d virtual environments. In *2007 IEEE Symposium on 3d user interfaces*. Washington, DC, USA : IEEE Computer Society. doi : 10.1109/3DUI.2007.340783
- ▶ Vickers, D. L. (1972). *Sorcerer's apprentice : Head-mounted display and wand* (Thèse de doctorat non publiée).
- ▶ Vive. (2019, 7 January). *Htc vive vr headset*. (<https://www.vive.com/us/product/vive-virtual-reality-system/>, retrieved January 7th, 2019)
- ▶ Vogel, D., & Balakrishnan, R. (2005). Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th annual ACM symposium on user interface software and technology* (pp. 33–42). New York, NY, USA : ACM. doi : 10.1145/1095034.1095041
- ▶ Wobbrock, J. O., Findlater, L., Gergle, D., & Higgins, J. J. (2011). The aligned rank transform for nonparametric factorial analyses using only ANOVA procedures. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 143–146). New York, NY, USA : ACM. doi : 10.1145/1978942.1978963

BIOGRAPHIE



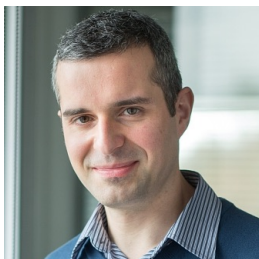
Marc BALOUP

Marc Baloup est doctorant dans l'équipe de recherche Loki, commune entre Inria et l'Université de Lille au sein de l'UMR 9189 CRISTAL. Ses travaux de recherche entrent dans le cadre de l'IPL Avatar, avec pour objectifs d'améliorer l'interaction d'un avatar avec son environnement et aussi d'améliorer le contrôle de l'avatar incarné par l'utilisateur.



Thomas PIETRZAK

Thomas Pietrzak est Maître de Conférences à l'Université de Lille, et membre de l'équipe de recherche Loki du laboratoire CRISTAL (UMR CNRS 9189) et du centre Inria Lille – Nord Europe. Son domaine de recherche est l'Interaction Homme-Machine. Il s'intéresse particulièrement à la conception, l'implémentation et l'évaluation de techniques d'interaction et de dispositifs interactifs, en particulier haptiques. Il publie régulièrement dans des conférences telles qu'ACM CHI, ACM UIST, et IHM, la conférence organisée par l'AFIHM. Il a été membre du conseil d'administration de l'AFIHM pendant 8 ans.



Géry CASIEZ

Géry Casiez est Professeur des Universités en Informatique à l'Université de Lille, membre de l'équipe de recherche Loki et directeur adjoint de l'UMR 9189 CRISTAL. Ses intérêts de recherche en Interaction Homme-Machine portent principalement sur le développement de nouveaux périphériques et techniques d'interaction, l'étude des facteurs humains et le développement de métriques et modèles prédictifs de performance. Il publie et participe régulièrement aux comités de programmes de conférences comme ACM CHI, ACM UIST et IHM. Il a été président de l'AFIHM pendant 4 ans. En 2018 il a été nommé membre junior de l'Institut Universitaire de France pour 5 ans.