

STARFISH : une aide graphique et haptique à la sélection en réalité virtuelle

Jonathan Wonner*
ENS Cachan Bretagne

Jérôme Grosjean†
LSIIT UMR 7005

Antonio Capobianco‡
LSIIT UMR 7005

Dominique Bechmann§
LSIIT UMR 7005

ABSTRACT

Nous présentons la méthode STARFISH d'aide à la sélection de cibles dans des environnements de réalité virtuelle. Cette méthode contraint graphiquement ou haptiquement le pointeur de l'utilisateur à l'intérieur d'un volume en forme d'"étoile de mer", dont les extrémités des branches sont placées au niveau des cibles. L'utilisateur peut "glisser" sur la surface des branches pour atteindre sa cible rapidement et sans difficultés. La forme est créée à partir d'une surface implicite dont les positions du pointeur et des cibles les plus proches forment le squelette. STARFISH est actuellement en cours d'expérimentation.

Index Terms: H.5.1 [Information interfaces and presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; H.5.2 [Information interfaces and presentation]: User Interfaces—Haptic I/O

1 INTRODUCTION

Les environnements de réalité virtuelle offrent la possibilité à l'utilisateur d'interagir avec les entités d'une scène 3D. La sélection est l'une des tâches les plus fondamentales, mais est difficile dans de tels environnements. Les principaux obstacles à la sélection sont une perception faussée de la profondeur, et la densité de la scène. De nombreuses techniques ont été proposées afin d'aider l'utilisateur à sélectionner une cible. Nous pouvons séparer la plupart de ces techniques en deux catégories [5] : les méthodes à base de pointeur [2, 3], et celles utilisant un rayon [12].

Dans leur enquête, *Hinckley et al.* [7] suggèrent que le retour multi-sensoriel améliore l'interaction dans une scène 3D. Des techniques de sélection s'appuyant sur des retours de force ont été proposées pour des environnements de réalité virtuelle disposant de dispositifs haptiques. *Picon* [17] associe un aimant à chaque élément géométrique de la scène (face, arête, sommet). L'utilisateur peut ainsi sélectionner un sommet en se laissant guider par les aimants. Cependant, dans le cas d'une scène dense en aimants, l'utilisateur peut être souvent attiré vers une entité qui n'est pas sa cible. Se libérer de son attraction pour se rediriger vers l'entité ciblée induit une perte de temps, et surtout une perte de confort pour l'utilisateur. Un autre type d'aide haptique moins contraignant est basée sur des polyèdres de constriction. Dans [6], les options d'un menu 3D sont les sommets d'un polyèdre dont les faces sont des "murs" haptiques infranchissables. L'utilisateur peut atteindre sa cible rapidement en "glissant" sur une des faces, puis le long d'une arête, et enfin se stabiliser dans un des coins du polyèdre, sans être contraint par des aimants. Cependant, cette méthode n'est possible que dans le cas où les cibles forment un polyèdre convexe.

Le contexte de notre travail est d'étendre l'idée des polyèdres de constriction au cas général où nous n'avons pas a priori sur la

disposition des cibles. Nous proposons une méthode de sélection de cibles, STARFISH (pour *Selection of TARgets From Implicit Surfaces with Haptics*), représentée en figure 1. STARFISH permet à l'utilisateur d'atteindre une cible en se déplaçant sur une surface fermée. Le pointeur est contraint soit graphiquement, soit haptiquement, à l'intérieur d'un volume formé par une surface implicite. Nous émettons l'hypothèse que notre méthode permet d'atteindre une cible avec autant de précision que les aimants tout en apportant plus de confort, en particulier en milieu dense.

Nous présentons en section 2 un état de l'art sur les techniques de sélection, graphique et haptique, en réalité virtuelle. La méthode STARFISH est ensuite détaillée en section 3. La section 4 est consacrée à une discussion sur le réglage des différents paramètres et sur les objectifs de l'expérimentation en cours.

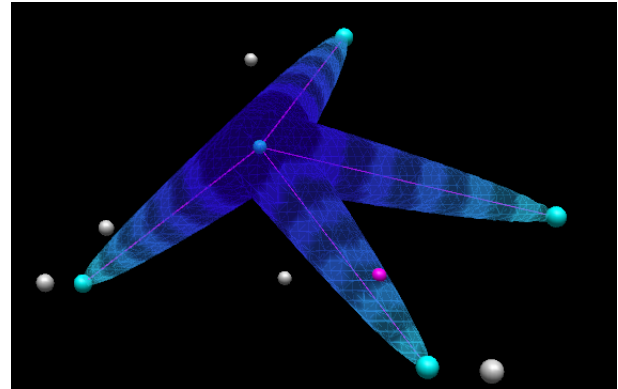


Figure 1: STARFISH dans une scène 3D. Le pointeur de l'utilisateur (en rose) est contraint dans le volume bleu en étoile de mer, dont les extrémités des branches sont des cibles potentielles.

2 ÉTAT DE L'ART

De nombreuses applications de réalité virtuelle nécessitent une sélection précise d'une entité ou d'une position 3D. Les principaux obstacles incluent la perception de la profondeur, l'occlusion et la densité des cibles [19]. Des techniques ont été élaborées afin d'aider l'utilisateur à sélectionner sa cible. Nous pouvons distinguer les techniques seulement graphiques, et les techniques utilisant l'haptique.

2.1 Aide graphique

La plupart des techniques de sélection sont des solutions ad hoc à certaines caractéristiques de la scène considérée. La taille de la scène est l'une de ces caractéristiques : si l'utilisateur est souvent amené à sélectionner des cibles hors de portée de main, une technique du type lancer de rayon est adaptée ; sinon, une main virtuelle, permettant d'accéder à toutes les cibles à portée de bras, sera préférée. *Bowman et al.* [3] propose des guides pour choisir la technique la plus adaptée.

Dang [5] présente de nombreuses techniques de sélection et en propose une classification. Le premier groupe de cette classifica-

*e-mail: wonner@unistra.fr

†e-mail: grosjean@unistra.fr

‡e-mail: a.capobianco@unistra.fr

§e-mail: bechmann@unistra.fr

tion contient les techniques basées sur le contrôle d'un point pour sélectionner une cible, comme une main virtuelle, [1, 22, 12, 18]. Les techniques du second groupe consistent à contrôler une ligne ou une courbe, comme le lancer de rayon, [2, 9, 15]. Ces techniques sont néanmoins difficiles à utiliser lorsque les cibles deviennent ponctuelles. Aligner un pointeur ou un rayon sur un point de l'espace demande un haut degré de précision. Pour diminuer la difficulté de cette tâche, nous pensons qu'il est nécessaire de guider le geste de l'utilisateur.

2.2 Aide haptique

La possibilité de toucher physiquement, à l'aide de dispositifs haptiques, les entités d'une scène virtuelle, augmentent la sensation de présence à l'intérieur de la scène [8, 16]. L'utilisation de l'haptique permet également d'améliorer la vitesse et la précision des interactions de l'utilisateur [10, 14]. Dans le cadre des interfaces graphiques, [13, 11] proposent des guides pour le design de widgets haptiques 3D, comme des boutons-poussoirs. Les aimants sont également souvent utilisés, en particulier en CAO, pour attirer l'utilisateur sur une grille [21], ou pour contraindre l'utilisateur sur la surface d'une entité [17]. Néanmoins, des aimants trop nombreux entraînent une perte de confort et de performance. Les polyèdres de constriction [4] reposent sur l'emploi de "murs" haptiques infranchissables plutôt que sur des aimants. L'utilisateur peut être guidé vers les cibles, mais sans être contraint et retenu en permanence. Cette technique n'a été évaluée, à notre connaissance, que dans le cas de cibles disposées selon un polyèdre convexe régulier, pour les options d'un menu 3D [6].

Notre méthode STARFISH, détaillée dans la section suivante, repose sur l'utilisation d'une surface de constriction créée à partir d'une surface implicite, afin de générer des "canaux" vers les cibles, quelle que soit leur disposition, comme l'illustre la figure 1. Nous émettons l'hypothèse que cette approche permet d'améliorer les performances et le confort de l'utilisateur lors d'une tâche de sélection de cibles, en particulier en milieu dense.

3 STARFISH

STARFISH est une technique basée sur l'utilisation d'un pointeur. Elle repose sur l'activation par l'utilisateur d'une surface sur laquelle il peut s'appuyer pour atteindre rapidement et avec précision sa cible, sans avoir à faire de geste précis. Dans notre implémentation, cette surface est générée au moment de l'appui sur une touche du périphérique, et supprimée lorsque l'utilisateur relâche cette touche. L'utilisateur ne bénéficie de l'aide de STARFISH que lorsqu'il le souhaite, pour ne pas entraver son déplacement, et ne pas surcharger inutilement la scène. Lorsque la surface est présente, le pointeur de l'utilisateur est contraint à l'intérieur du volume formé par la surface. La section 3.1 décrit le processus de création de cette surface. Les méthodes graphiques et haptiques pour contraindre le pointeur sont détaillées dans la section 3.2.

3.1 Création de la forme

Créer la forme en étoile de mer de STARFISH nécessite deux étapes. Nous formons un squelette à partir des cibles environnantes, puis nous générons une surface implicite à partir de ce squelette.

3.1.1 Génération du squelette

STARFISH consiste en la création d'une surface implicite basée sur un squelette en segments. Ces segments ont pour extrémités la position P_0 du pointeur au moment de l'activation de STARFISH, et des cibles C_i les plus proches. Afin de limiter les calculs et la surcharge visuelle, nous ne considérons au maximum que les N_{max} cibles les plus proches à une distance d'influence maximale D_{max} . La figure 2 représente le squelette considéré pour STARFISH en 2D.

Afin que chaque branche de l'étoile de mer n'interfère pas avec les autres, nous imposons un angle minimum Θ_{min} entre chaque

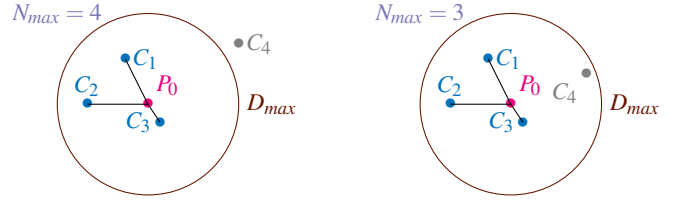


Figure 2: Squelettes de STARFISH pour deux scènes 2D. À gauche, la cible C_4 n'est pas considérée car hors de la zone d'influence. À droite, C_4 n'est pas parmi les N_{max} cibles les plus proches.

segment, comme l'illustre la figure 3. Si deux segments forment un angle inférieur à Θ_{min} , alors nous ne conservons que la cible la plus proche. Si les deux cibles sont à égale distance, alors nous en choisissons une de manière arbitraire.

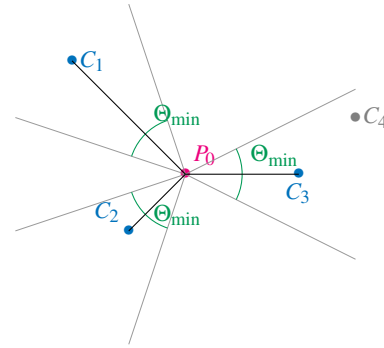


Figure 3: Gestion des conflits pour la création du squelette. La cible C_4 n'est pas considérée car l'angle $C_3P_0C_4$ est inférieur à Θ_{min} , et C_3 est plus proche de P_0 que C_4 .

3.1.2 Génération de la surface

Afin de générer notre surface implicite, nous avons besoin d'une fonction d de distance au squelette, et d'une fonction E de potentiel. La fonction E indique le potentiel d'un point P de l'espace en fonction de sa distance au squelette. Elle doit être décroissante (plus un point est loin du squelette, plus son potentiel est faible) et positive sur son intervalle de définition. Dans notre implémentation, nous avons utilisé la fonction E suivante, dont le graphe est représenté en figure 4 :

$$E(P) = \begin{cases} 0 & \text{si } d(P) \geq R_s, \\ \left(1 - \frac{d(P)^2}{R_s^2}\right)^2 & \text{sinon.} \end{cases}, \quad (1)$$

où la distance R_s représente le rayon d'influence de notre fonction implicite. Si un point a une distance au squelette supérieure à R_s , alors son potentiel est nul.

Nous définissons également la fonction d de distance au squelette :

$$d(P) = \min_{i=1 \dots n} (d_i(P)), \quad (2)$$

où n est le nombre de segments composant le squelette (avec $1 \leq n \leq N_{max}$), et les d_i sont les fonctions de distance à chacun des segments du squelette. Pour un segment $[P_0C_i]$ du squelette, nous souhaitons que la surface implicite générée permette de guider le pointeur de l'utilisateur jusqu'à C_i . La figure 5 représente l'aspect souhaité de la surface.

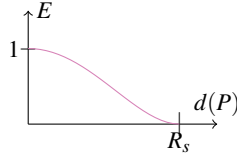


Figure 4: Graphe de la fonction de potentiel E à support fini.

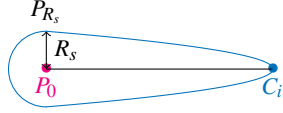


Figure 5: Représentation en 2D de la surface générée par STARFISH pour un segment. En s'appuyant sur la surface, le pointeur de l'utilisateur peut se stabiliser sur C_i .

Nous souhaitons une fonction de distance d_i qui vérifie :

- $d_i(C_i) = d_i(P_{R_s}) = R_s$, et
- $d_i(P) = 0$ si et seulement si $P = P_0$.

Nous proposons de définir cette fonction d_i de distance au segment $[P_0C_i]$ de la manière suivante :

$$d_i(P) = \begin{cases} \frac{(P_0P)^2}{R_s} & \text{si } t_i(P) < 0, \\ \frac{(PP_h)^2}{R_s} + R_s \cdot t_i(P) & \text{sinon.} \end{cases} \quad (3)$$

où P_h est le projeté orthogonal de P sur la droite (P_0C_i) , et $t_i(P)$ est un facteur représentant la position de P_h sur l'axe (P_0C_i) , avec en particulier $t_i(P) = 0$ si $P_h = P_0$, et $t_i(P) = 1$ si $P_h = C_i$. Dans notre implémentation, nous avons :

$$t_i(P) = \frac{\overrightarrow{P_0P_h} \cdot \vec{u}_i}{P_0C_i}, \quad (4)$$

où $\vec{u}_i = \frac{\overrightarrow{P_0C_i}}{P_0C_i}$ est un vecteur directeur unitaire de la droite (P_0C_i) .

La figure 6 illustre les valeurs prises par t_i .

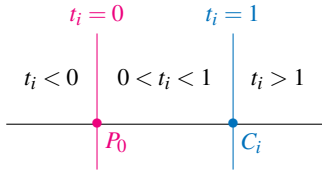


Figure 6: Représentation des valeurs prises par t_i . $t_i(P)$ est compris entre 0 et 1 si P_h est sur le segment $[P_0C_i]$.

Cette fonction vérifie bien les caractéristiques souhaitées. La figure 7 représente plusieurs surfaces iso-distance pour un unique segment $[P_0C_i]$. La surface passant par C_i est celle qui nous intéresse. Tous les points P de cette surface ont une distance au squelette $d(P) = d_i(P) = R_s$, et un potentiel $E(P) = 0$.

La technique STARFISH consiste à générer la surface S_0 fermée iso-potentielle $E = 0$, puis à contraindre le pointeur de l'utilisateur,

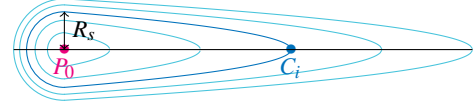


Figure 7: Surfaces iso-distance pour un squelette composé d'un unique segment dans une scène 2D.

initialement en P_0 , dans le volume formé par cette surface. La figure 8 représente la surface S_0 dans le cas d'un squelette avec plusieurs segments. Le pointeur peut atteindre facilement les cibles et se déplacer de l'une à l'autre en s'appuyant sur cette surface. La figure 9 représente l'aspect de STARFISH, avec 1, 2 et 6 cibles à proximité.

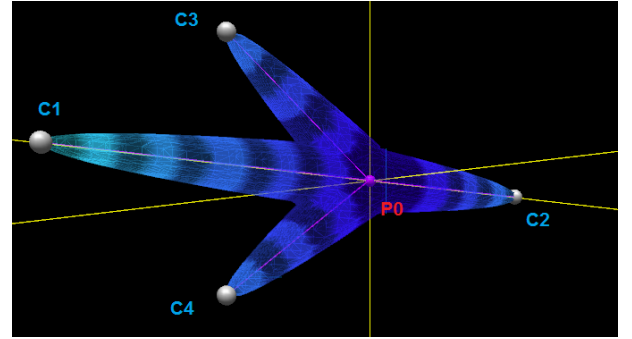
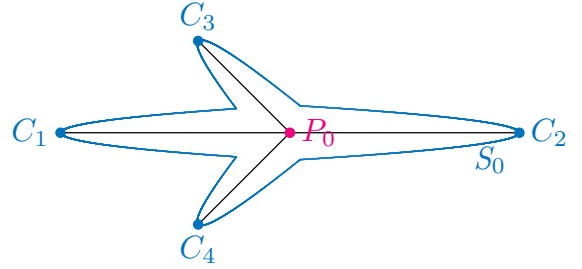


Figure 8: Surface iso-potentielle d'un squelette avec plusieurs segments dans une scène 2D (en haut) et 3D (en bas).

3.2 Contrainte du pointeur

Nous proposons en sections 3.2.1 et 3.2.2 deux méthodes, l'une haptique et l'autre graphique, pour contraindre le pointeur à l'intérieur du volume formé par S_0 .

3.2.1 Contrainte haptique

Pour les environnements comportant un dispositif haptique, une solution est de le contraindre haptiquement à l'intérieur du volume défini par l'iso-potentielle, en lui appliquant une force lorsqu'il essaye de traverser la surface S_0 . À chaque instant, nous appliquons au pointeur de position P la force

$$\vec{F}(P) = F(P) \cdot \vec{N}_{\text{surf}}, \quad (5)$$

avec :

$$F(P) = \begin{cases} 0 & \text{si } E(P) \geq E_{\min} \\ F_{\max} \left(1 - \frac{E(P)}{E_{\min}} \right) & \text{sinon} \end{cases}, \quad (6)$$

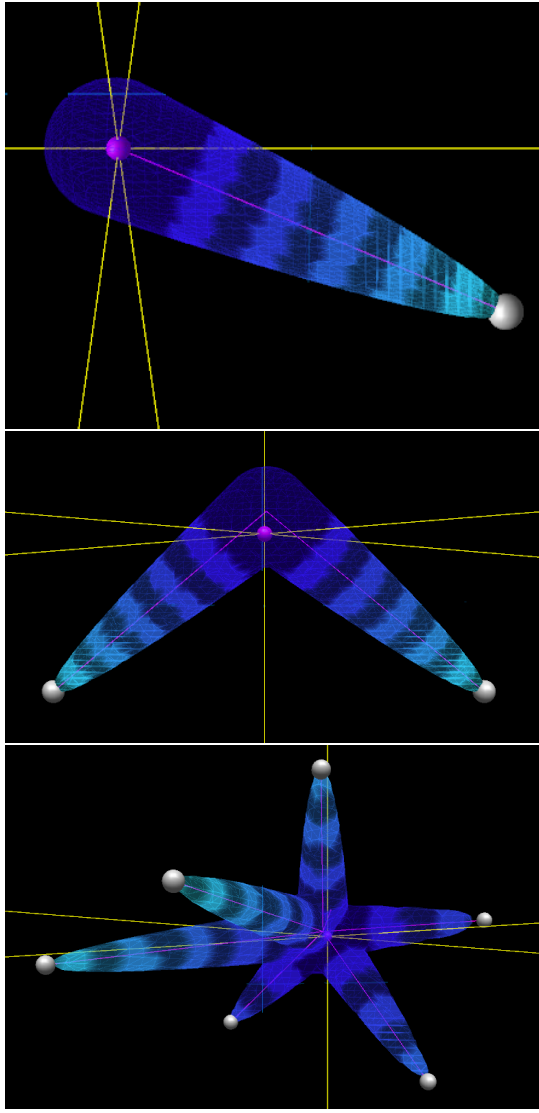


Figure 9: Aspect de STARFISH pour respectivement 1, 2, et 6 segments.

où F_{max} est une force de valeur arbitrairement grande selon le dispositif haptique employé, \vec{N}_{surf} est un vecteur unitaire dans la direction du gradient du champ de potentiel en P , et E_{min} est un potentiel de valeur arbitraire. Les deux surfaces S_0 et $S_{E_{min}}$ délimitent un volume à l'intérieur duquel le retour de force croît de manière continue. La figure 10 représente le graphe de F en fonction de E . L'annexe A détaille le calcul de \vec{N}_{surf} .

La solution haptique est simple du point de vue de l'implémentation. Il suffit d'envoyer une force sur le pointeur pour rendre la surface infranchissable. En revanche, en l'absence de dispositif haptique, contraindre les déplacements du pointeur dans le volume est plus problématique.

3.2.2 Contrainte graphique

Nous proposons ici une alternative graphique pour contraindre le pointeur de l'utilisateur à l'intérieur de S_0 . Le principe de cette méthode est de créer un second pointeur, dit *captif*, qui doit suivre le pointeur *libre* tout en restant dans une zone de potentiel strictement

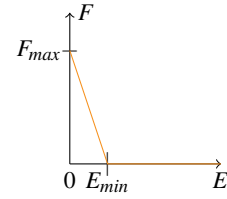


Figure 10: Graphe représentant l'amplitude de la force à appliquer sur le pointeur, en fonction du potentiel.

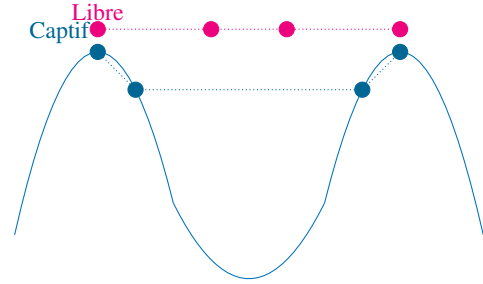


Figure 11: Exemple de situation où la méthode naïve de suivi de surface échoue. Un petit déplacement du pointeur libre entraîne un saut du pointeur captif entre deux branches.

positif. Visuellement, seul le pointeur captif est affiché, tandis que l'utilisateur ne contrôle la position que du pointeur libre.

Au moment de l'activation de STARFISH, le pointeur captif est créé à la même position que le pointeur libre. Lorsque le pointeur libre est à l'intérieur de la forme en étoile de mer, la position du pointeur captif est identique à la position du pointeur libre. Lorsque le pointeur libre est à l'extérieur de la forme, le pointeur captif longe la surface pour se rapprocher le plus possible du pointeur libre.

Une solution naïve pour longer la surface est de placer le pointeur captif sur le point de la surface le plus proche du pointeur libre. Cependant, cette solution peut entraîner des "sauts" entre deux branches de la surface, comme l'illustre la figure 11. Au moment du saut, le pointeur libre n'étant pas affiché, le pointeur captif semble avoir le même comportement que s'il était attiré par un aimant. Cela entraîne une perte de contrôle et de confort.

Nous souhaitons que le pointeur captif suive le pointeur libre en longeant la surface. L'utilisateur doit pouvoir se stabiliser sur l'extrémité d'une des branches, même si le pointeur libre ne s'arrête pas précisément sur cette extrémité. Le passage entre deux branches se fait en épousant grossièrement la forme de la surface, comme l'illustre la figure 13. Nous proposons une méthode de suivi de surface, basée sur de petits déplacements successifs dans des directions orthogonales à la normale à la surface \vec{N}_{surf} . L'algorithme utilisé est le suivant, en supposant que le pointeur captif est déjà sur la surface S_0 :

- 1) Calcul du vecteur $\vec{D}_L = \vec{P}_C P_L$, où P_C est la position du pointeur captif, et P_L la position du pointeur libre.
- 2) Calcul de \vec{N}_{surf} , vecteur normal à la surface en P_C (voir l'annexe A).
- 3) Si \vec{D}_L est colinéaire à \vec{N}_{surf} ,
le pointeur captif n'est pas déplacé (voir figure 12.A).
Arrêt de l'algorithme.

Sinon,

le pointeur captif effectue un déplacement de vecteur

$\vec{Dep} = \alpha \cdot \vec{N}_{surf} \wedge (\vec{D}_L \wedge \vec{N}_{surf})$, où α est un facteur arbitrairement petit (voir figure 12.B).

- 4) Calcul de la nouvelle normale \vec{N}_{surf} à la surface.
- 5) Le pointeur captif est itérativement déplacé dans la direction \vec{N}_{surf} jusqu'à retrouver un potentiel nul, et donc la surface S_0 (voir figure 12.C).
- 6) **Retour à l'étape 1.**

Les différentes étapes de cet algorithme sont illustrées sur la figure 12.

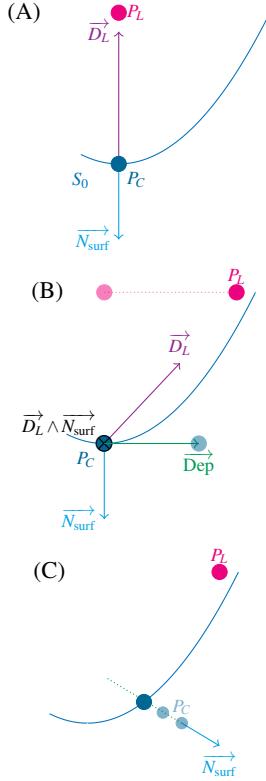


Figure 12: (A) \vec{D}_L est colinéaire à \vec{N}_{surf} , le pointeur n'est pas déplacé. (B) Déplacement de vecteur \vec{Dep} . (C) Déplacement dans la direction \vec{N}_{surf} , pour retrouver un potentiel nul.

Cet algorithme permet de suivre le pointeur libre en se déplaçant le long de la surface, sans faire de sauts.

4 DISCUSSION

Nous traitons dans cette section des réglages des différents paramètres de STARFISH, ainsi que des objectifs de l'expérimentation en cours.

4.1 Réglages des paramètres

La méthode STARFISH nécessite le réglage de nombreux paramètres : D_{max} , N_{max} , Θ_{min} , R_s , F_{max} , E_{min} , α et β (voir l'annexe A pour ce dernier paramètre). Ces réglages dépendent entre autres de l'application et du dispositif utilisé. Par exemple, F_{max} représente la force maximale appliquée sur le pointeur, et doit être calibrée selon le dispositif haptique employé.

Nous souhaitons que STARFISH aide l'utilisateur à atteindre sa cible, en particulier dans un milieu dense. Nous estimons ainsi que

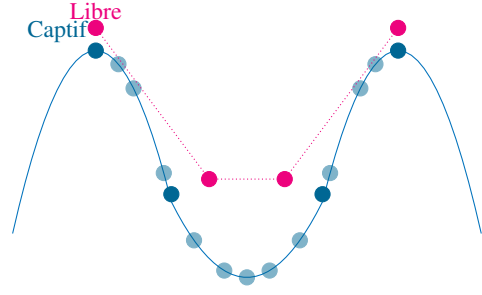


Figure 13: Suivi de la surface S_0 par l'algorithme de la section 3.2.2.

les quatre paramètres de forme (D_{max} , N_{max} , Θ_{min} et R_s) doivent être ajustés en fonction de la densité des cibles proches. En effet, en choisissant une valeur de D_{max} ou de N_{max} trop petite, ou une valeur de Θ_{min} trop grande, nous réduisons le nombre de cibles retenues pour le squelette. En particulier, la cible de l'utilisateur risque de ne pas être considérée. À l'inverse, avec une valeur de D_{max} trop grande, le squelette pourrait contenir une cible lointaine et possiblement hors de portée de geste. Une valeur de N_{max} élevée pourrait entraîner la formation de nombreuses branches entraînant une surcharge visuelle. De plus, régler Θ_{min} à une valeur trop faible, ou R_s trop grande, risque de "fusionner" certaines branches entre elles ; l'utilisateur pourrait alors ne pas être guidé correctement. Enfin pour de petites valeurs de R_s , les branches sont étroites et choisir la branche dans laquelle entrer devient malaisé. La complexité du réglage de ces paramètres réside dans leur dépendances à la géométrie de la scène et à leur propre interdépendance. Un réglage automatique de ces paramètres selon la densité de la scène pourrait avoir un impact important sur l'aide apporté par STARFISH, mais nécessite des études supplémentaires.

Le passage d'une cible à l'intérieur de l'étoile de mer, à une cible proche à l'extérieur, se fait manuellement en supprimant puis en activant de nouveau STARFISH. Il est possible d'effectuer cette opération automatiquement, par exemple dès que le pointeur arrive à l'extrémité d'une branche. L'utilisateur peut ainsi naviguer de proche en proche, et être guidé jusqu'à sa cible.

4.2 Expérimentation en cours

Nous souhaitons quantifier l'aide apportée par notre méthode, en la comparant avec d'autres techniques de sélection. En particulier, le temps de sélection et le nombre d'erreurs sont deux des critères objectifs de performance que nous évaluons. Les critères subjectifs importants que nous considérons incluent le confort, l'intuitivité, la fatigue et la surcharge mentale.

Les techniques de sélection que nous désirons comparer sont :

Aide graphique : STARFISH avec contrainte graphique, lancer de rayon, et main virtuelle.

Aide haptique : STARFISH avec contrainte haptique, aimants permanents et aimants activés sur commande.

Nous émettons l'hypothèse qu'en milieu dense, STARFISH avec contrainte haptique offre plus de confort que les aimants. Des aimants activés en permanence entravent la progression de l'utilisateur. Avec des aimants activés sur commande explicite, par exemple par un appui de bouton, le pointeur de l'utilisateur risque d'être attiré sur une cible qui n'est pas la sienne. À l'inverse, STARFISH donne une indication sur les directions qu'il est possible d'emprunter, et offre un guide sans diriger tout seul le pointeur.

De plus, nous pensons que STARFISH, avec contrainte graphique, permet d'être plus rapide que la main virtuelle, car le suivi automatique de la surface permet d'atteindre une cible sans effectuer de geste précis.

STARFISH peut être utilisé dans des environnements de réalité virtuelle avec ou sans dispositif haptique. Nous ne savons pas encore laquelle, de la version haptique ou son alternative graphique, est la plus performante et la plus appréciée.

5 CONCLUSION ET TRAVAUX FUTURS

Dans cet article, nous proposons une nouvelle approche pour aider à la sélection d'une cible dans un environnement de réalité virtuelle. Cette approche consiste à contraindre l'utilisateur à l'intérieur d'un volume dont la forme évoque une étoile de mer, et dont les extrémités des branches sont placées au niveau des cibles. L'utilisateur peut ainsi se déplacer sur le bord du volume et se stabiliser sur les cibles. Afin de s'adapter à la disposition des cibles, notre technique, STARFISH, génère une surface fermée définie implicitement. Nous proposons également deux méthodes, l'une graphique et l'autre haptique, afin de contraindre l'utilisateur à l'intérieur du volume. STARFISH peut ainsi être employée sur des environnements intégrant, ou non, un dispositif haptique.

Dans notre implémentation, l'activation de STARFISH requiert une commande explicite de l'utilisateur (appui sur un bouton du dispositif). De même, si l'utilisateur souhaite sortir de l'étoile de mer, il doit relâcher le bouton. Nous pensons qu'il est possible d'activer automatiquement STARFISH. Une première idée consiste à intégrer un algorithme de prédiction de cibles basé sur le mouvement de l'utilisateur [20]. STARFISH est alors activé dès que l'algorithme estime que le pointeur est proche de sa cible.

ANNEXES

A CALCUL DE LA NORMALE

La surface S_0 étant une iso-potentielle, calculer sa normale en un point P revient à calculer la direction du gradient du champ de potentiel $\vec{\nabla}E$ en ce point. Dans notre implémentation, nous calculons ce gradient comme étant la somme :

$$\vec{\nabla}E = \frac{1}{\beta} \sum_{i=1}^6 E(P_i) \cdot \vec{PP}_i, \quad (7)$$

où les P_i sont les centres des faces d'un cube de centre P et de rayon arbitraire β , comme l'illustre la figure 14. Nous posons alors la

normale à la surface $\vec{N}_{\text{surf}} = \frac{\vec{\nabla}E}{\|\vec{\nabla}E\|}$.

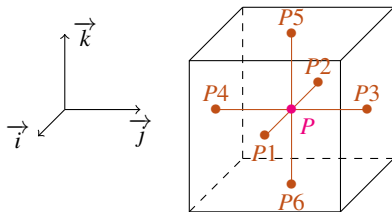


Figure 14: Calcul du gradient de potentiel $\vec{\nabla}E$. Dans le repère $(\vec{i}, \vec{j}, \vec{k})$, ce gradient s'écrit $\vec{\nabla}E = (E_1 - E_2, E_3 - E_4, E_5 - E_6)$, où $E_i = E(P_i)$.

REFERENCES

- [1] E. A. Bier. Skitters and jacks: interactive 3d positioning tools. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, pages 183–196, 1987.
- [2] D. Bowman. *Interaction Techniques for Common Tasks in Immersive Virtual Environments: Design, Evaluation and Application*. PhD thesis, Georgia Institute of Technology, 1999.
- [3] D. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., 2004.
- [4] A. Burel. Aide haptique à la sélection de cibles dans un nuage de points en réalité virtuelle, 2009.
- [5] N.-T. Dang. A Survey and Classification of 3D Pointing Techniques. In *Proceedings of 2007 IEEE International Conference on Research, Innovation and Vision for the Future*, pages 71–80, 2007.
- [6] C. Essert-Villard and A. Capobianco. Hardborders : a new haptic approach for selection tasks in 3d menus. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology 2009*, pages 243–244, 2009.
- [7] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell. A survey of design issues in spatial input. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 213–222, 1994.
- [8] B. E. Insko. *Passive haptics significantly enhances virtual environments*. PhD thesis, 2001.
- [9] J. Liang and M. Green. Jdcad: A highly interactive 3d modeling system. *Computers and Graphics*, 18:499–506, 1994.
- [10] K. Lundin, M. Cooper, A. Persson, D. Evstedt, and A. Ynnerman. Enabling design and interactive selection of haptic modes. *Virtual Real.*, 11:1–13, 2007.
- [11] T. Miller and R. Zeleznik. The design of 3d haptic widgets. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 97–102, 1999.
- [12] M. R. Mine. *Virtual Environment Interaction Techniques*. Technical report, University of North Carolina, 1995.
- [13] I. Oakley, A. Adams, S. Brewster, and P. Gray. Guidelines for the design of haptic widgets. In *16th British HCI Group Annual Conference*, pages 195–211, 2002.
- [14] I. Oakley, M. R. McGee, S. Brewster, and P. Gray. Putting the feel in look and feel. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422, 2000.
- [15] A. Olwal and S. Feiner. The flexible pointer: An interaction technique for selection in augmented and virtual reality. In *User Interface Software and Technology*, 2003.
- [16] M. A. Otaduy and M. C. Lin. Introduction to haptic rendering. In *ACM SIGGRAPH 2005 Courses*, 2005.
- [17] F. Picon, M. Ammi, and P. Bourdot. Case study of haptic methods for selection on cad models. In *Proceedings of the 2008 IEEE Conference 2008 on Virtual Reality*, pages 209–212, 2008.
- [18] I. Poupyrev, M. Billingham, S. Weghorst, and T. Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80, 1996.
- [19] A. Steed. Towards a general model for selection in virtual environments. In *Proceedings of the IEEE conference on Virtual Reality*, pages 131–, 2006.
- [20] J. Wonner, J. Grosjean, A. Capobianco, and D. Bechmann. Speed : Prédiction de cibles. In *Proc. of IHM 2011*, 2011. To appear.
- [21] T. Yamada, T. Ogi, D. Tsubouchi, and M. Hirose. Desk-sized immersive workplace using force feedback grid interface. In *Proceedings of the IEEE Virtual Reality Conference 2002*, pages 135–, 2002.
- [22] S. Zhai, W. Buxton, and P. Milgram. The silk cursor: investigating transparency for 3d target acquisition. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, pages 459–464, 1994.