



HAL
open science

Guide de conception d'une technique de désignation 3D

Cédric Dumas, Frédéric Jourdan, Patricia Plénacoste, Laurence Perron,
Amine Chellali

► **To cite this version:**

Cédric Dumas, Frédéric Jourdan, Patricia Plénacoste, Laurence Perron, Amine Chellali. Guide de conception d'une technique de désignation 3D. Revue des Sciences et Technologies de l'Information - Série TSI: Technique et Science Informatiques, 2009, 28 (6-7), pp.799-829. 10.3166/tsi.28.799-829 . hal-00480547

HAL Id: hal-00480547

<https://hal.science/hal-00480547v1>

Submitted on 4 May 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Guide de conception d'une technique de désignation 3D

**Cédric Dumas^(1,2), Frédéric Jourdan⁽¹⁾, Patricia Plénacoste⁽³⁾,
Laurence Perron⁽⁴⁾, Amine Chellali^(1,2)**

*(1) Ecole des Mines de Nantes
4, avenue Alfred Kastler F-44307 Nantes cedex 3
Frederic.Jourdan@emn.fr
Amine.Chellali@emn.fr*

*(2) IRCCYN
1, rue de la Noë F-44321 Nantes cedex 3
Cedric.Dumas@ircryn.ec-nantes.fr*

*(3) LIFL & INRIA Futurs
Bâtiment M3 F-59655 Villeneuve d'Ascq cedex
Patricia.Plenacoste@univ-lille1.fr*

*(4) Orange France Télécom R&D
2 avenue Pierre Marzin F-22307 LANNION cedex
Laurence.Perron@orange-ftgroup.com*

RÉSUMÉ. Nous présentons un tour d'horizon des techniques de désignation existantes en IHM, aussi bien en 2D qu'en 3D. Leur analyse nous a permis de proposer un guide de conception d'une technique de désignation 3D pour les développeurs d'environnements virtuels. Une illustration de cette méthode est proposée avec le pointage sémantique en 3D.

ABSTRACT. We present a panel of pointing techniques, both in 2D and 3D. Their analyse led us to propose here a guide for 3D pointing techniques design, dedicated to programmers and designers of virtual environments. An example of 3D semantic pointing is given following this guide.

MOTS-CLÉS : désignation, pointage, interaction 3D, environnement virtuel.

KEYWORDS : pointing, 3D interaction, virtual environments.

1. Introduction générale

Si l'affichage tridimensionnel (3D) s'est généralisé sur le plan technique en ce qui concerne les ordinateurs depuis la fin du XXème siècle, son développement du côté des utilisateurs commence seulement à se diffuser en masse avec des applications grand public (GoogleEarth de Google™), sortant la 3D d'applications métier pointues ou d'applets spécialisées. Les mondes virtuels, les jeux en ligne ou plus vraisemblablement la TV3D devraient forcer à moyen terme la génération des flux d'information à s'enrichir systématiquement de la troisième dimension. Côté dispositif d'interaction, en dehors de domaines plus matures comme la réalité virtuelle, les avancées de grande diffusion sont récentes, comme l'apparition de manettes de jeux 3D (Wii de Nintendo™) ou de périphériques de navigation (NULOOQ de Logitech™), et sont encore relativement rares.

Si l'on peut donc constater l'évolution technique des dispositifs d'entrée et de sortie, l'utilisation de ces évolutions reste encore marginale, entre marketing (Aero sous Windows Vista™) et applications spécialisées (Time Machine sous Mac OS X™) ; elles promettent cependant un enrichissement certain de la visualisation dans les environnements de travail informatique. Pour ce qui est des moyens d'interagir avec ces représentations, les choses se font encore plus lentement. A commencer par la désignation, où le point d'action habituel de toute application se cantonne encore à un point 2D, matérialisé par une flèche statique appelée à juste titre *pointeur*.

L'évolution du couple exclusif clavier/souris est pourtant rendu nécessaire, la vision classique de l'informatique se diversifie, les mentalités et les attentes changent grâce à des approches innovantes à grande échelle comme l'écran tactile de la console Nintendo™ DS, l'utilisation de caméras comme périphérique d'entrée (EyeToy de Sony™) ou encore les écrans multi-tactiles (iPhone de Apple™).

Un des problèmes de l'Interaction Homme Machine (IHM) autour de l'industrialisation de nouvelles techniques d'interaction (périphériques inclus) publiées ces vingt dernières années commence donc à se résoudre avec le développement et le gain en maturité du marché de l'informatique, la montée en puissance de la culture des technologies de l'information et des attentes qui en découlent.

Dans ce contexte, il nous semble utile d'apporter des méthodes de conception en 3D comme il en existe tant pour les interfaces graphiques dites WIMP (*Windows, Icons, Menus, Pointing device*), afin d'accompagner le développement des environnements virtuels. Nous proposons ici une heuristique, construite à partir de l'analyse de techniques existantes et de notre expérience, qui pourra guider les concepteurs dans leurs choix en matière de désignation 3D.

Notion de désignation

Les Environnements Virtuels Collaboratifs (EVC) sont basés sur l'utilisation d'une interface tridimensionnelle. La projection d'une interface 3D sur la surface d'un ou plusieurs écrans permet à l'utilisateur d'interagir avec des objets (modélisés, représentés et perçus) en trois dimensions. Dans ces conditions, la notion de désignation et de sélection d'objets est à repenser par rapport à nos interfaces 2D traditionnelles où le couple souris/pointeur s'est imposé. Interagir dans un environnement virtuel présuppose que l'on sache concrétiser son intention :

1. être capable de se déplacer vers un objet virtuel,
2. être capable de désigner un objet virtuel,
3. être capable de sélectionner cet objet virtuel,

afin de pouvoir le manipuler, réaliser des actions dessus.

Ces trois actions, associées à des rétroactions (*feedbacks*) appropriées de l'environnement, permettent d'initier toutes les tâches basées sur de la gestuelle dans un environnement virtuel : la sélection et la manipulation d'objets 3D, la navigation, etc.

Dans cet article, nous nous focalisons sur le point (2) uniquement en faisant le point sur les différentes techniques de désignation, dans une logique d'action sur un objet (navigation/sélection/manipulation), c'est-à-dire en rapport avec la fonction ergotique du geste (nous ne nous intéressons pas ici à sa fonction épistémique ou sémiotique [1]).

L'objectif de cet article est donc de faire le point sur l'évolution des techniques de désignation en 2D/3D de ces dernières années et d'en voir la portée pour les environnements virtuels. Il s'articule en trois parties : les techniques de désignation, leur analyse comparative, et une synthèse sous la forme d'un guide de conception.

2. Les techniques de désignation et de sélection

Diverses techniques de désignation en 2 et 3 dimensions ont été explorées, décrites ou implémentées, quelquefois évaluées. Elles sont issues de contextes variés tels que les classiques bureaux WIMP ou les écrans à grande surface d'affichage en 2 dimensions, ou encore les systèmes (semi-)immersifs en 3 dimensions de la réalité virtuelle. Nous présentons ici une quinzaine d'exemples dont l'ensemble constitue un tour d'horizon de l'existant en la matière. Nous avons regroupé ces exemples en fonction des techniques mises en œuvre dans leur réalisation.

2.1. Curseur surfacique ou volumique

Le point d'action dans l'espace virtuel est classiquement représenté par un pointeur, c'est-à-dire un curseur ponctuel. Un objet est désigné ou sélectionné lorsque ce pointeur est positionné en un point appartenant à l'objet. Afin de faciliter

la sélection de petits objets, plusieurs techniques remplacent le pointeur de dimension pixel par un curseur de dimension plus importante : une aire en 2D ou un volume en 3D. Un objet est sélectionné lorsque son intersection avec le curseur est non vide.

En 2 dimensions, le terme *area cursor* [2,3] désigne un curseur qui prend la forme d'une zone rectangulaire ou circulaire (cf. figure I.a). Dans le cas d'un environnement dense (comprenant des cibles nombreuses et rapprochées), un *area cursor* peut intersecter plusieurs objets, induisant une ambiguïté quant à la cible désignée. Pour remédier à ce problème, la technique du *bubble cursor* [4] propose un *area cursor* circulaire dont le rayon s'adapte dynamiquement de façon à toujours englober une cible unique (cf. figure I.b).

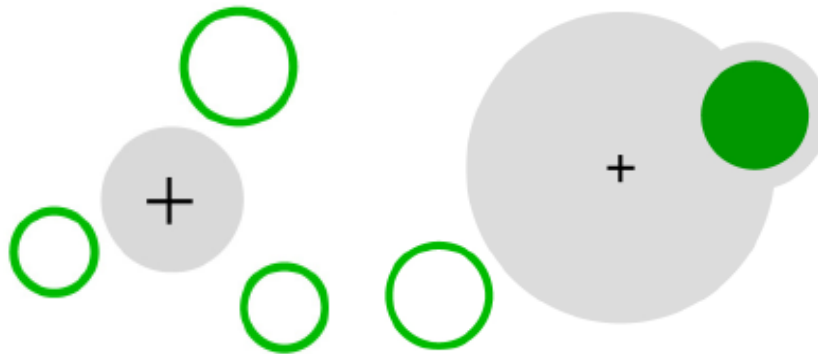


Figure I. a) à gauche, un *area cursor* circulaire ; **b)** à droite un *bubble cursor* (les cibles sont en vert, les curseurs en gris) - extrait de [4].

Des techniques analogues existent pour les interfaces en 3 dimensions. Zhai *et al.* ont proposé le *silk cursor* [5], « curseur de soie » constitué d'une boîte cubique semi-transparente. Un objet situé derrière le curseur est vu à travers deux épaisseurs de soie, alors qu'un objet situé à l'intérieur du curseur n'est vu qu'à travers une seule épaisseur de soie. La transparence partielle du curseur lui permet de ne pas masquer les objets qui apparaissent derrière lui.

Dans les environnements tridimensionnels, on utilise souvent un « rayon pointeur » basé sur la métaphore du pointeur laser. Un rayon est projeté depuis la main de l'utilisateur dans l'espace 3D, et l'objet sélectionné est l'objet le plus proche intersecté par le rayon (*ray selection*). Ce type de pointeur peut également être « agrandi » pour devenir un curseur volumique en forme de cône dont le rayon forme l'axe (cf. figure II). En cas de sélection multiple, l'objet sélectionné peut être l'objet le plus proche de l'axe du cône (*cone selection*). Dans la technique de *shadow cone selection*, l'utilisateur désigne l'objet à sélectionner en déplaçant sa main de façon à ce que la cible reste en permanence dans le cône [6].

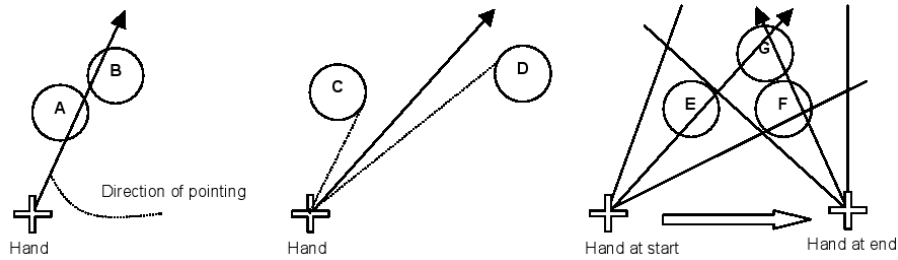


Figure II. De gauche à droite, *ray selection* (l'objet A est sélectionné), *cone selection* (l'objet D est sélectionné), *shadow cone selection* (l'objet G est sélectionné) - extrait de [6].

2.2. Lever l'ambiguïté en cas de sélection multiple

Introduite dans la section précédente, la question de l'ambiguïté entre plusieurs cibles potentielles lors d'une opération de désignation ou de sélection n'est pas spécifique à l'utilisation de curseurs surfaciques ou volumiques. L'étude de cette question dans le contexte d'un chevauchement de fenêtres 2D ou d'une occlusion d'objets 3D est à l'origine d'une série de techniques que nous regroupons dans cette section.

La technique du *target chooser* [7] est destinée à faciliter la sélection d'une fenêtre en 2D. À l'activation de cette technique, un rayon est lancé perpendiculaire à la surface de l'écran et la fenêtre dont le centre est le plus proche du rayon est sélectionnée. Un retour visuel est fourni pour signaler à l'utilisateur la fenêtre concernée, celui-ci pouvant ensuite circuler dans la liste des fenêtres par de petits mouvements de la souris. En 3D, les techniques de *depth ray* et *lock ray* [8] permettent également de parcourir la liste des différentes cibles possibles le long d'un rayon pointeur, dans le contexte des dispositifs d'affichage volumétriques (cf. figure III).

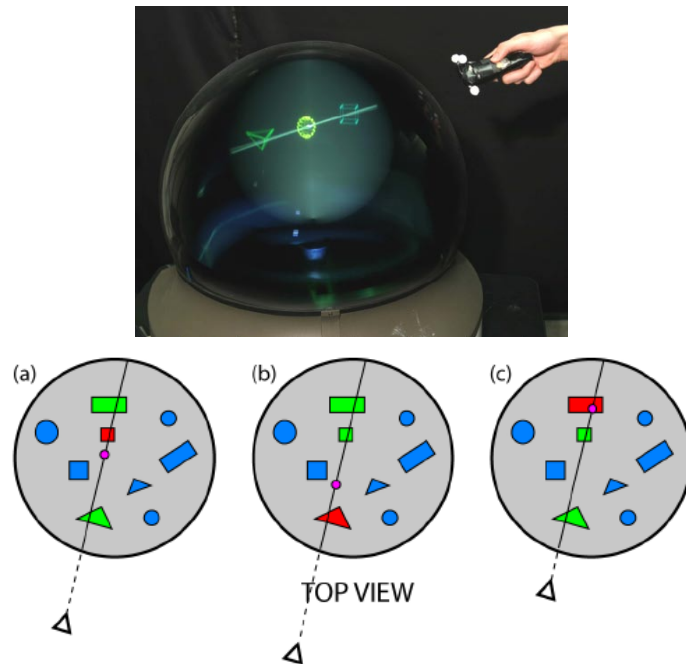


Figure III. En haut, un système d'affichage volumétrique et un pointeur rayon. En dessous, schéma illustrateur du *depth ray* : le point rouge peut être déplacé le long du rayon afin de sélectionner l'une des trois cibles potentielles – extrait de [8].

Une autre possibilité utilisable aussi bien en 2 qu'en 3 dimensions est de modifier temporairement la géométrie de la scène représentée pour présenter à l'utilisateur une vue d'ensemble des cibles potentielles. Avec le *splatter* [9], un groupe d'objets 2D, qui se chevauchent, est présenté en une vue éclatée où les objets sont disposés en cercle autour du point d'activation (cf. figure IV). C'est sur cette vue que l'utilisateur sélectionne ensuite la cible de son choix au moyen du pointeur. En 3D, une vue éclatée analogue est utilisée par la technique du *flower ray* [8] avec un pointeur rayon (cf. figure V).

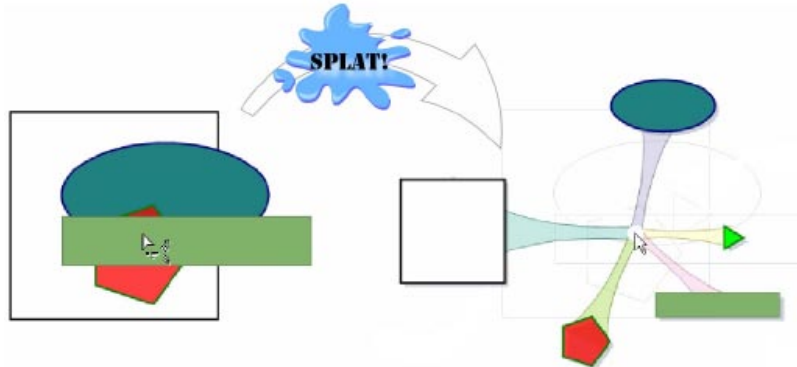


Figure IV. Illustration de la technique de *splatter* sur un groupe de 5 objets superposés dont l'un (le petit triangle vert) est au départ invisible) - extrait de [9].

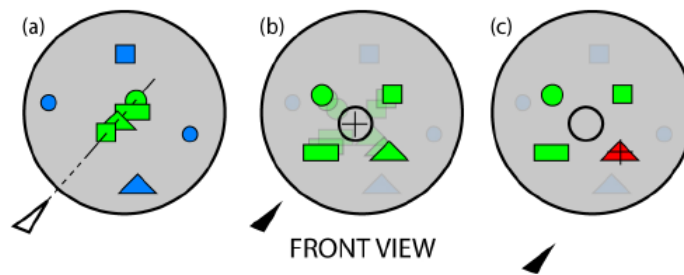


Figure V. La technique de *flower ray* dans le cadre de l'utilisation d'un système d'affichage volumétrique) - extrait de [8].

Dans le cas de fenêtres superposées, le *fold'n drop* [10] permet de replier et déplier les fenêtres par manipulation directe, selon la métaphore du carnet que l'on feuillette (cf. figure VI).

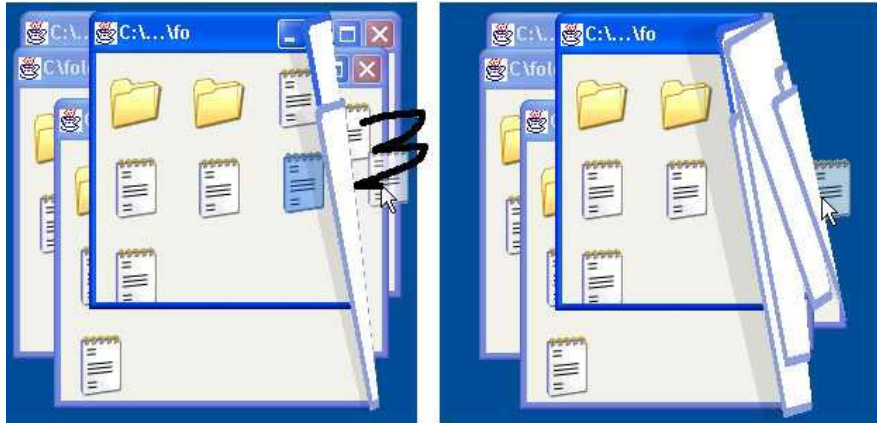


Figure VI. Manipulation directe de fenêtres avec le *fold'n drop* - extrait de [10]

2.3. Zoom et Radar

Lorsque la désignation d'une cible exige un geste trop précis (cibles trop petites dans l'environnement virtuel) ou trop ample (espace virtuel trop vaste), le système peut afficher une représentation de l'espace virtuel agrandie (*zoom*) ou rétrécie (*radar*) pour faciliter le pointage.

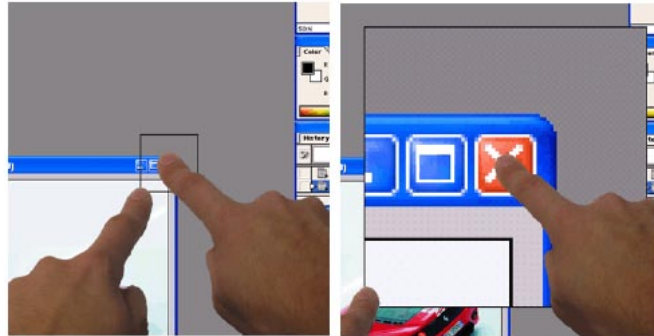


Figure VII. Utilisation d'un zoom - extrait de [11].

On pourra trouver des exemples récents d'utilisation de ces techniques chez Benko *et al.* [11], qui emploient un zoom déclenché par une interaction bimanuelle sur un écran tactile (cf. figure VII), tandis qu'Aliakseyeu *et al.* [12] proposent avec *bubble radar* (cf. figure VIII) une juxtaposition de deux modes d'interaction dont l'un est un radar, l'autre étant un *bubble cursor*.

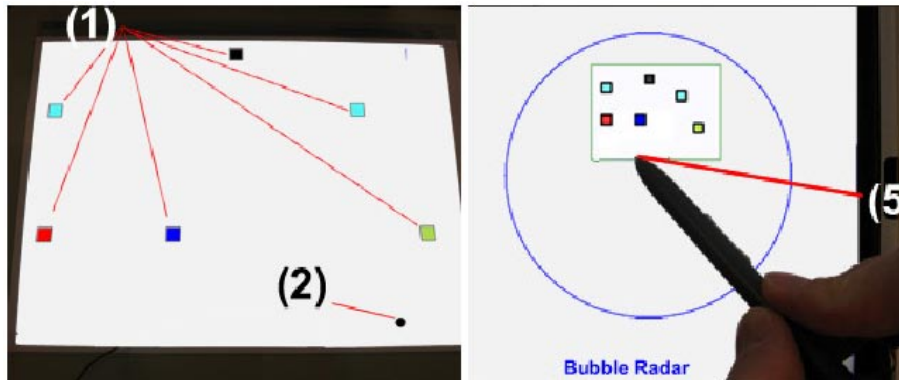


Figure VIII. Le radar (rectangle vert à droite) est une vue réduite de l'espace de travail de gauche où sont représentées les six cibles (1) (fléchées en rouge à gauche) - extrait de [12].

2.4. Atteindre des cibles éloignées

Dans le cadre de l'utilisation de dispositifs d'affichage sur une grande surface (écrans muraux, tables écrans...), de nombreuses propositions ont été faites pour faciliter la désignation de cibles situées à grande distance dans l'espace virtuel. Un espace de travail en 2D où les cibles sont des icônes (bureau) constitue le contexte de cette famille de travaux, à laquelle cette section est consacrée.

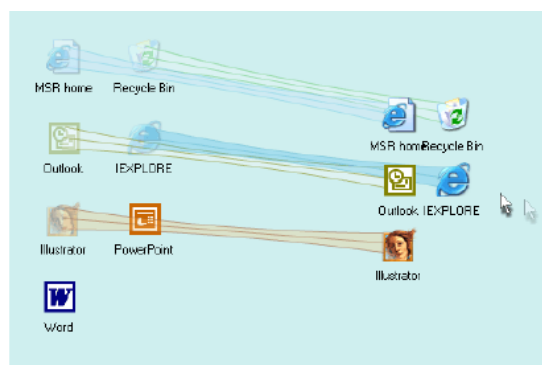


Figure IX. Illustration du *drag and pick* - extrait de [13].

Pour réduire la distance virtuelle entre le pointeur et la cible, deux types de stratégies ont été explorées. La première consiste à rapprocher la cible du pointeur et est mise en œuvre par les techniques (cf. figure IX) de *drag and pop* et *drag and pick* [13]. Ces techniques sont adaptées du classique *drag and drop* et sont initiées de la même manière : on clique avec le périphérique de pointage, puis on le déplace en direction des éléments graphiques. Les icônes situées dans la direction indiquée par le mouvement de la souris, considérées par le système comme cibles

potentielles, sont alors dupliquées. Le double de ces cibles se rapproche du pointeur, de sorte qu'un mouvement de faible amplitude suffit à l'utilisateur pour sélectionner la cible souhaitée. *Drag and pop* et *drag and pick* sont établies sur ce principe, la nuance étant que dans le premier cas le mouvement est initié à partir d'une icône (pour déposer une icône sur une autre), alors que dans le second cas le mouvement est initié à partir d'un espace vide (sélection d'une cible éloignée).

La seconde stratégie consiste à envoyer le pointeur vers les cibles par un geste de

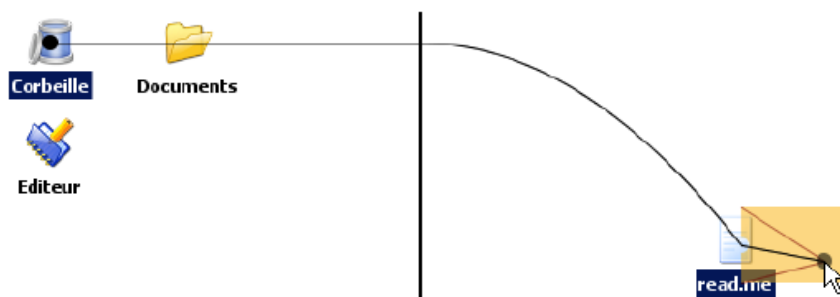


Figure X. Exemple de *drag and throw* (métaphore du tir à l'arc) - extrait de [14].

petite amplitude. C'est l'approche retenue pour le *drag and throw* (cf. figure X) et le *push and throw* (cf. figure XI) [14], techniques de lancer d'icônes conçues pour les dispositifs d'affichage à surfaces multiples. À l'activation de ces techniques, une zone rectangulaire « de décollage » est affichée autour du pointeur. Les mouvements du pointeur dans cette zone sont traduits en termes de trajectoire de l'icône qui peut ainsi être projeté à grande distance du point initial d'activation. La différence entre les deux techniques réside dans la métaphore employée pour gouverner la trajectoire de l'icône lancée : métaphore du tir à l'arc pour l'une, métaphore du pantographe pour l'autre.

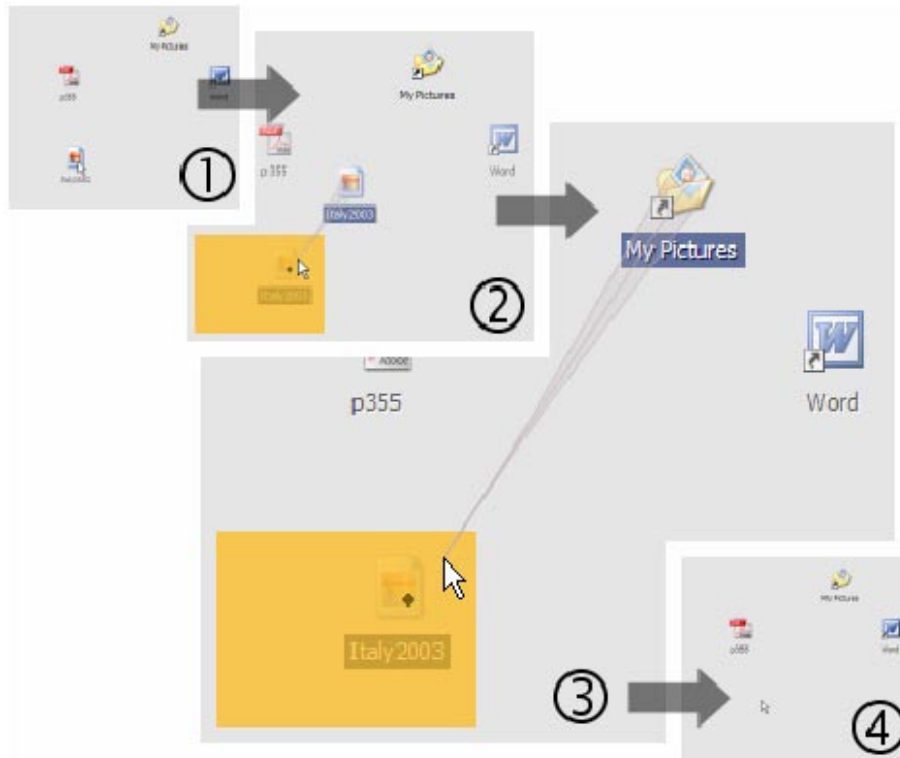


Figure XI. Scénario en images du déroulement d'un *push and throw*. L'icône initialement située en bas à gauche est déplacée dans le dossier « My Pictures » - extrait de [14].

S'inspirant des exemples précédents, Collomb *et al.* ont récemment proposé de combiner les deux stratégies (cible vers pointeur et pointeur vers cible) avec la technique du *push and pop* [15]. L'interaction se déroule comme illustré sur la figure XII. (1) L'utilisateur commence à tirer l'icône à déplacer, comme pour exécuter un *drag and drop*. (2) Le système affiche autour du pointeur une mini-zone représentant l'espace d'affichage tout entier. Cette « zone de décollage » est similaire à un radar, sinon que les icônes y sont représentées avec leur taille d'origine, et non rétrécies comme dans le cas d'un radar classique. (3) L'utilisateur relâche l'icône à déplacer au-dessus de la copie de l'icône de la corbeille. (4) Il ne reste plus qu'à relâcher le bouton du périphérique de pointage.

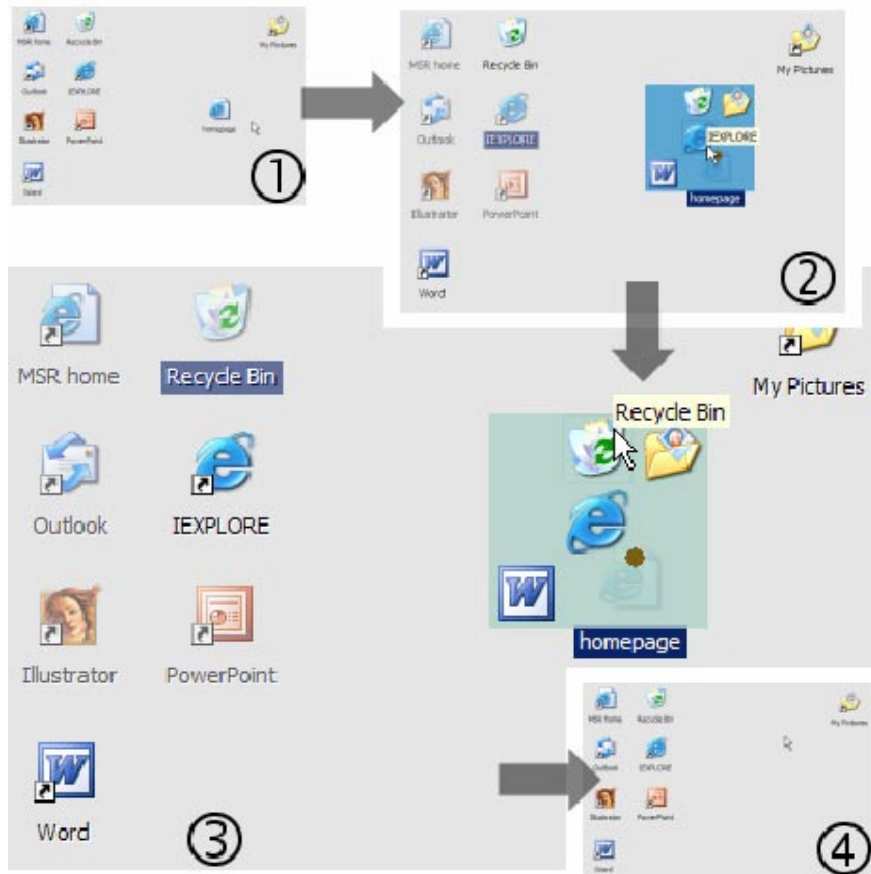


Figure XII. Déroulement du *push and pop* : l'icône située au milieu à droite est déplacée dans la corbeille - extrait de [15].

D'une inspiration différente des précédents exemples, signalons aussi le *tractorbeam* [16] : un périphérique spécifiquement 3D (cf. figure XIII), à savoir un stylo à 6 Degrés De Liberté (DDL) est utilisé pour interagir avec un dispositif d'affichage en 2D (une table utilisée comme écran). Ce dispositif (voir figure XIII) permet de pointer à distance des cibles sur une table interactive, particulièrement efficace pour de grosses cibles (par rapport au touché de la surface), les auteurs ont ajouté d'autres techniques pour améliorer le pointage sur de tailles inférieures, en particulier l'aimantation du pointeur sur les bords d'un objet proche.

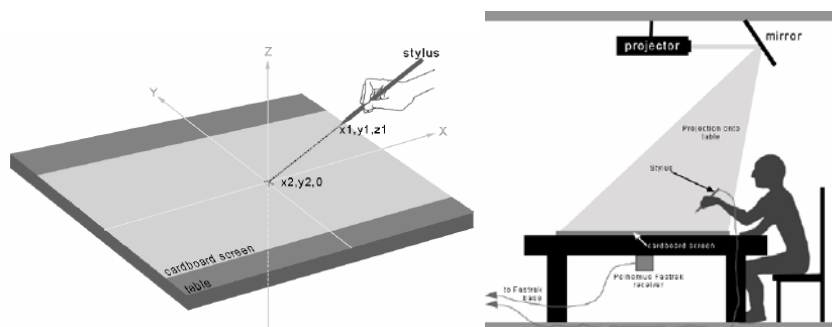


Figure XIII. Le tractorbeam - extrait de [16].

2.5. Control-display ratio non linéaire

Nous avons regroupé ici des approches qui jouent sur une modification dynamique du rapport entre les déplacements du périphérique de pointage et de l'objet contrôlé (*control-display ratio* ou CDR). Un exemple bien connu est « l'accélération de la souris » pour contrôler un pointeur 2D. On peut donner de ces approches des exemples de techniques en 3D et en 2D.

Avec la technique *go-go* [17], la vitesse de déplacement d'un pointeur 3D dans un environnement de réalité virtuelle est modifiée en fonction de la distance entre la main et la poitrine de l'utilisateur. L'utilisateur peut aussi bien atteindre par un même type de geste les objets situés dans son environnement immédiat, que les objets situés à grande distance.

Le *pointage sémantique* [18] est une technique où le CDR est adapté en fonction de la sémantique des objets affichés (cf. figure XIV). Plus précisément, la vitesse du pointeur dépend de la position du pointeur par rapport aux objets affichés. Le principe consiste à déplacer le pointeur plus lentement sur (ou au voisinage) des cibles potentielles et plus rapidement sur un espace vide. Selon les auteurs, leur technique revient à modifier la taille des objets affichés dans l'espace moteur et non dans l'espace visuel, évitant ainsi les déformations de l'affichage observées avec les techniques détaillées à la section précédente.



Figure XIV. Pointage sémantique : (a) une boîte de dialogue dans l'espace visuel, (b) la même représentée dans l'espace moteur sous l'effet du pointage sémantique - extrait de [18].

Les mêmes auteurs ont également proposé une version poussée à l'extrême de ce principe, appelée *object pointing* [19], dans laquelle le pointeur ne peut que se

trouver au-dessus d'une cible, « sautant » de cible en cible selon les mouvements du dispositif de pointage.

Enfin une adaptation de leurs travaux à la 3D a été proposée dans [20], étendue à tout type de formes (pas seulement des rectangles), mais pour une désignation d'objet dans le plan de projection avec un périphérique 2D (avec les 2 DLL d'une souris par exemple). Un calcul de CDR non linéaire dans ce plan est proposé, utilisant les fonctionnalités matérielles des cartes graphiques pour être temps réel,

3. Comparaisons et Analyse

Nous effectuons ici une analyse comparative des techniques présentées précédemment, sous l'aspect des périphériques mis en œuvre, du contexte, et du type de tâches auxquelles elles sont destinées.

3.1. Techniques 2D : classification selon les périphériques utilisés

Le tableau suivant présente les techniques de désignation en 2 dimensions sous l'aspect des périphériques d'entrée et de sortie qu'elles mettent en œuvre. Les différentes classes de périphériques d'entrée 2D que nous prenons en compte sont :

1. absolus directs : écran tactile, crayon optique...
2. absolus indirects : tablette...
3. relatifs indirects : souris...
4. non gestuels : trackball...

Pour chaque technique, on marque ainsi sa compatibilité avec les périphériques :

- **Orig** désigne le(s) périphérique(s) avec lequel elle est originellement présentée,
- **Adap** désigne les classes de périphériques non gestuels auxquelles la technique est à notre sens adaptable,
- **Incpt** désigne les classes de périphériques auxquelles la technique n'est pas adaptable, c'est-à-dire incompatible,
- **Dédié** précise si la technique est dédiée au périphérique de sortie utilisé.

	absolu direct	absolu indirect	relatif indirect	non gestuel	périphérique de sortie	dédié ?
area cursor	Orig	Orig	Orig	Adap	écran classique	
bubble cursor			Orig	Adap	écran classique	
target chooser			Orig	Adap	écran classique	
Splatter			Orig		écran classique	
fold'n drop			Orig		écran classique	
Zoom	Orig	Incpt	Incpt		écran multitactile	Dédié
bubble radar	Orig				table écran + tablette	Dédié
drag and pop/pick	Orig				grand écran multiple	
drag/pick and throw	Orig				grand écran multiple	
push and pop	Orig				grand écran multiple	Dédié
pointage sémantique		Orig			écran classique	

Périphérique d'entrée

Les techniques de sélection en 2 dimensions que nous avons présentées dans la première section de ce document ont toutes été pensées dans le cadre de la métaphore classique d'un bureau comportant fenêtres et/ou icônes. Ce même type de contexte explique l'homogénéité des périphériques utilisés. Il s'agit dans chaque cas de périphériques gestuels (le geste effectué avec le périphérique est transmis au pointeur), qui se répartissent principalement entre périphériques absolus directs (stylos pour écrans tactiles) et périphériques relatifs indirects (classiques souris).

À quelques exceptions près (*zoom*, *bubble radar*), les travaux dont ces techniques sont issues ne visent pas à exploiter les spécificités de périphériques particuliers, mais à proposer des techniques d'interaction plus générales. Pour cette raison on n'observe pas de spécialisation par rapport aux périphériques d'entrée : chaque technique¹ est adaptable à toutes les classes de périphériques gestuels (non signalé dans le tableau pour des raisons de lisibilité). En revanche, l'importance du geste dans nombre de techniques (lancer de pointeur...) les rend incompatibles avec les périphériques non gestuels.

¹ Mise à part la technique de zoom, qui dans la version que nous présentons est dédiée à un dispositif spécifique à plusieurs pointeurs pilotés par les doigts de l'utilisateur.

Périphérique de sortie

En ce qui concerne les périphériques de sortie, le même type de contexte entre les différentes techniques explique là encore le peu de variété rencontrée et en particulier le grand nombre de techniques mettant en œuvre un écran classique. Mais à la différence des périphériques d'entrée, on peut distinguer toute une famille de techniques orientées vers les spécificités d'un type de périphérique de sortie. Il s'agit des dispositifs d'affichage de grande taille (*large display*), souvent composés de plusieurs écrans, qui ont suscité de nombreux travaux (*drag and pop*, *drag and throw...*). Bien que ces techniques se présentent comme dédiées à de tels dispositifs (*drag and pop*), nous pensons qu'elles demeurent utilisables avec des écrans plus classiques.

3.2. Techniques 2D : classification selon le contexte

Nous regroupons ici les différentes techniques 2D de désignation et sélection selon le contexte dans lequel elles ont été introduites. Nous envisageons également les différents types de tâches de sélection auxquelles les techniques sont adaptées.

Pour les tâches de sélection, nous considérons les deux axes ci-après. Le premier concerne le nombre d'objets à sélectionner, dans les tâches de sélection d'un objet isolé et de sélection d'un groupe d'objets. Le second axe est celui de la densité de l'environnement virtuel, qui peut être dense (objets nombreux, rapprochés) ou clairsemé (objets peu nombreux, éloignés).

Du point de vue du contexte, nous distinguons quatre groupes non disjoints de techniques, répondant aux quatre problématiques suivantes :

1. faciliter la sélection en toute généralité dans une interface graphique (GUI) classique,
2. permettre la sélection dans les cas de chevauchement entre objets,
3. trouver des techniques adaptées aux dispositifs d'affichage de grande taille,
4. améliorer la technique du *drag and drop*.

Le premier de ces groupes concerne les techniques de sélection dans un contexte général d'une interface graphique classique (bureau, icônes), sans particularisation sur une situation ou un matériel. On y trouve la technique d'*area cursor* et ses dérivés successifs *bubble cursor* et *bubble radar*. Dans l'article original [3], il s'agissait de rendre la sélection plus facile pour les personnes âgées. Bien que conçue avant tout pour sélectionner un objet isolé, cette technique de curseur surfacique peut être employée pour sélectionner un groupe d'objets. Elle s'avère en revanche inadaptée en environnement dense. Cette lacune est corrigée par la technique de *bubble cursor*, au prix d'une limitation à la sélection d'un objet unique. Les tâches concernées par le *bubble radar* sont la sélection d'un objet isolé, mais aussi la désignation d'un point quelconque dans l'espace libre.

Toujours dans ce premier groupe, la technique de *pointage sémantique* vise à rendre plus efficace la sélection d'une cible quelle que soit sa situation dans l'espace virtuel. Cette meilleure possibilité d'exploration de la totalité de l'espace virtuel est obtenue grâce à l'exploitation de connaissances sur la nature des objets présents dans l'environnement. Permettant de traverser rapidement les zones vides de cibles potentielles, cette technique est bien adaptée aux environnements clairsemés. Elle ne reste cependant pas spécialisée à ces derniers, puisque le découplage entre les espaces visuels et moteurs donne la possibilité de densifier l'environnement représenté (cibles importantes de grande taille dans l'espace moteur, mais de petite taille à l'écran).

Trois des techniques présentées l'ont été dans le contexte du chevauchement d'objets (fenêtres ou éléments d'un dessin 2D), cas particulier d'environnement dense. L'objectif du *target chooser* est la sélection d'une fenêtre dans un chevauchement. Avec le *splatter*, il s'agit de sélectionner un élément dans le contexte d'occlusions en dessin 2D. Enfin le *fold'n drop* vise à permettre les opérations de glisser-déposer entre fenêtres qui se chevauchent.

Nombreuses sont les techniques développées dans le cadre des *large displays* (dispositifs d'affichage de grande taille, tels les tables écrans ou les écrans muraux, éventuellement multiples). Les techniques de *zoom* sont principalement destinées à la sélection de petites cibles, quelle que soit la densité de l'environnement. L'utilisation d'un *radar* est quant à elle pertinente dans le cas des environnements clairsemés. Les techniques de *drag and pop/pick*, *drag/pick and throw* et *push and pop* sont spécifiquement conçues pour les grandes surfaces d'affichage et sont pour cette raison peu appropriées aux environnements denses.

On peut enfin remarquer que plusieurs techniques visent à améliorer les opérations de glisser-déposer (*drag and drop*) :

- dans le contexte de chevauchement de fenêtres pour le *fold'n drop*,
- dans celui des *large displays* pour les *drag and pop/pick*, *drag/pick and throw* et *push and pop*.

L'ensemble des remarques précédentes se trouve synthétisé dans le tableau suivant, où les croix X indiquent la compatibilité des techniques avec les différents types de tâches et de contextes :

	Contexte				Tâches			
	GUI classique	chevauchement	<i>large displays</i>	<i>drag and drop</i>	objet isolé	groupe d'objets	environ. dense	environ. clairsemé
area cursor	X				X	X		X
bubble cursor	X				X		X	X
target chooser		X			X		X	
splatter		X			X	(X)	X	
fold'n drop		X		X	X		X	
zoom			X		(X)	(X)	X	X
bubble radar	X		X		X			X
drag and pop/pick			X	X	X			X
drag/pick and throw			X	X	X			X
push and pop			X	X	X			X
pointage sémantique	X				X		X	X

3.3. Techniques 3D

Dans un environnement 3D, la sélection s'effectue communément par collision entre la cible désirée et la main (réelle ou virtuelle) de l'opérateur ou bien par intersection de la cible avec un rayon projeté à partir de cette même main. À la première catégorie appartiennent les techniques *silk cursor* et *go-go*, alors que les *shadow cone*, *depth/lock ray* et *flower ray* relèvent de la seconde catégorie. La technique de *tractorbeam*, hybride entre 2D et 3D, emprunte aux deux catégories.

Périphériques entrée et sortie

Dans la plupart des cas, le périphérique d'entrée utilisé met en œuvre un capteur spatial à 6 DDL, permettant de suivre la position et l'orientation de la main de l'utilisateur. Le *silk cursor* se limite quant à lui aux mouvements de translation, un gant de données à 3 degrés de liberté serait donc suffisant. Plus généralement, on peut d'ailleurs remarquer que trois DDL suffisent à la désignation par intersection

avec un pointeur 3D, tandis que la désignation par un rayon met en jeu cinq DDL. Nous reviendrons sur ce point plus loin.

Plutôt centrées sur la réalité virtuelle ou augmentée, nos techniques de référence utilisent des périphériques de sortie avancés :

non immersifs : écran cathodique stéréoscopique (*silk cursor*, *go-go*) ;

immersifs : casque de réalité virtuelle (*go-go*) ;

immersifs spatialisés : de type CAVE (*shadow cone*) ;

dispositifs de rendu volumétrique 3D (*depth/lock ray*, *flower ray*) ;

table interactive (*tractorbeam*).

Pensées par leurs auteurs comme dédiées à ces dispositifs, ces techniques ont pour but de tirer partie des spécificités des environnements 3D pour lesquels elles ont été conçues. Elles ne sont donc pas nécessairement utilisables dans d'autres environnements, comme avec la classique combinaison d'un écran ordinaire et d'une simple souris (2 DDL disponibles pour des techniques nécessitant 5 DDL).

Contexte et zone d'action

Si le *silk cursor* [5] a été présenté dans un contexte très général d'exploration de nouvelles techniques d'interaction en 3 dimensions, les autres techniques de référence interviennent dans des contextes plus spécialisés. Le *tractorbeam* a été spécifiquement conçu pour les tables interactives. Les *depth ray* et *flower ray*, pour les nouveaux dispositifs de rendu volumétrique tridimensionnel. Quant au *shadow cone*, il est destiné à être employé dans un environnement immersif spatialisé multi-utilisateurs, pour s'adapter au fait que la vue n'y est exacte que pour un seul de ces utilisateurs.

Le type de tâche pour lesquelles ces techniques ont été présentées est pour chacune d'entre elles la sélection d'un objet unique. Les techniques de *depth/lock ray*, *flower ray* et *shadow cone* sont spécifiquement conçues pour être efficaces dans un environnement dense. Les techniques *go-go* et *tractorbeam* ont pour objectif l'intégration fluide de la désignation entre des objets proches et des objets éloignés de l'utilisateur (en évitant la navigation).

Ces deux techniques proposent en effet deux modes d'interaction différents selon que la cible est proche ou lointaine, avec une transition naturelle d'un mode à l'autre. Le *silk cursor* permet quant à lui d'interagir avec les objets proches. Avec le *shadow cone*, basé sur un pointeur rayon, un seul mode permet d'interagir avec l'ensemble de l'environnement (quelque soit la distance de la cible), au prix de manipulations parfois difficiles pour résoudre les problèmes d'ambiguïtés de la cible visée.

En ce qui concerne le rayon de leurs zones d'action, la situation des techniques dédiées aux dispositifs volumétriques est particulière : du fait de leur petite taille, tous les objets peuvent être considérés comme proches, le principal problème provenant dans ce cas de la densité de l'environnement.

3.4. Bilan

Notre but étant la désignation en 3D, nous avons naturellement été conduits à traiter séparément les techniques 2D et 3D, pour lesquelles les problématiques sont différentes. En ce qui concerne les techniques 2D, il s'agit de savoir dans quelle mesure elles peuvent être généralisées à la 3D. Parmi les techniques que nous avons présentées, certaines techniques 3D peuvent déjà être considérées comme des généralisations. On peut ainsi former les couples suivants :

2D	3D
<i>area cursor</i>	↔ <i>silk cursor</i>
<i>bubble cursor</i>	↔ <i>shadow cone</i>
<i>splatter</i>	↔ <i>flower ray</i>
<i>target chooser</i>	↔ <i>depth/lock ray</i>

Parmi les techniques 2D, le pointage sémantique partage avec *go-go* la variation dynamique de control-display ratio.

Pour ce qui est des techniques 3D, les exemples utilisés étaient pour la plupart dédiés à des périphériques particuliers. Afin d'être capable de choisir la technique la plus adaptée, en fonction de la tâche plutôt que d'un périphérique ou d'un dispositif d'affichage donné, nous voulons proposer une méthode de conception pour la désignation en environnement virtuel.

Nous retiendrons de l'ensemble de ces techniques qu'elles se différencient sur plusieurs points fondamentaux :

1. les périphériques d'entrée/sortie qui leurs sont spécifiques,
2. le mode de déplacement du point (ou zone) d'action, i.e. le CDR utilisé,
3. la définition de leur point (ou zone) d'action, et la méthode de calcul des éléments désignés,

Un choix pertinent de ces éléments est particulièrement utile en environnement virtuel, où l'introduction de la troisième dimension, la complexité de la gestion des points de vue et de la visibilité des objets, ou encore la grande variété de périphériques qui peuvent servir à désigner, rendent la mise au point d'une méthode de désignation nettement plus ardue qu'en 2D, chaque nouvelle application ayant des spécificités qui empêchent les raccourcis (comme celui de l'utilisation systématique de la souris).

Si le point (1) est spécifique à chaque tâche et les éléments caractéristiques du point (2) connus pour la 3D depuis [17] et formalisés depuis [18], le point (3) nécessite une description plus complète, avant l'exposé du guide, autour de la modélisation du point (ou zone) d'action.

4. Modélisation du point d'action

Le point d'action est la réalité d'un utilisateur dans son environnement virtuel.

Comme nous avons pu l'observer en ce qui concerne les techniques de désignation répertoriées dans la partie précédente, la souris reste le périphérique incontournable pour les applications 2D, ainsi que dans le domaine de la recherche. Seuls certains environnements virtuels se démarquent de ces deux degrés de liberté pour la désignation.

4.1. Le pointeur

L'être humain a des capacités d'action relativement étendues dans le monde réel, grâce à ses sens et à ses capacités motrices. Si on se limite à la seule étude du geste technique, on retrouve une multitude de capacités de manipulation d'outils dans des situations très variées qui ont permis à l'humain de produire tout ce qu'il a pu par ailleurs imaginer (ou presque). Du côté informatique, la généralisation du modèle WIMP pour l'ensemble des applications informatiques a ramené toutes ces possibilités d'action aux deux degrés de liberté de la souris. Cette dernière s'instrumente (molettes, multiples boutons...) et d'autres périphériques se diffusent comme nous l'avons déjà remarqué, mais le point d'action de l'utilisateur, capable d'interaction bi-manuelle, tridimensionnelle, vocale... reste très largement seulement un point 2D.

Ce point d'action est pourtant la seule matérialisation de l'utilisateur dans l'environnement de travail virtuel. On y réalise des tâches complexes, l'ordinateur peut être le seul instrument de notre activité, mais on nous impose pourtant un handicap lourd, une factorisation à outrance du geste sous forme de longues suites de clics à la souris.

On opposera à ce constat que les interfaces tactiles et à stylos (TabletPC, PDA) se généralisent, que les environnements virtuels n'ont jamais eu à souffrir de cette limitation avec l'interaction gestuelle et son cortège de périphériques 3D, ou encore que les applications de travail collaboratif synchrone ou post-WIMP ont permis d'outrepasser cette limitation. Il ne s'agit cependant pas encore des cas les plus répandus.

La perception du point d'action

Le point d'action, lorsqu'il s'agit d'un point 2D, est manipulé à travers des périphériques gestuels, avec une corrélation entre mouvement réel (la main) et mouvement virtuel (le pointeur) qui permet à l'utilisateur une grande performance, même si le déplacement n'est pas dans le même plan ou si le ratio n'est pas linéaire (CDR). Nous avons pu remarquer cependant dans la partie précédente que l'utilisation de CDR dynamique (ou adaptative pour le pointage sémantique) doit nécessiter une phase d'apprentissage un peu plus longue [18].

Mais la perception du point d'action dans l'environnement se fait d'autant mieux que son comportement correspond aux gestes de l'utilisateur, et qu'il est facile de se créer un modèle mental du fonctionnement du pointeur.

Le choix d'une technique de désignation est donc un compromis entre une interaction gestuelle directe naturelle à l'utilisateur et l'utilisation d'optimisation dans le déplacement du point d'action qui permettent d'améliorer ses performances dans un environnement donné.

Le choix du point d'action

Le choix du point d'action est fortement conditionné par le type de périphérique : le pointeur de la souris, le point d'encre des stylets, la main virtuelle des datagloves, ... Cependant, contrairement aux interfaces 2D WIMP, on ne peut que rarement se contenter en 3D d'un pointeur 2D de taille fixe, la profondeur et la complexité des scènes faisant apparaître des objets de tailles très diverses, y compris relativement petits ou bien composites, et pour lesquels une orientation fixe ne conviendra pas. Il est en effet souvent nécessaire d'orienter le pointeur 3D dans la direction de l'objet le plus proche afin de rendre moins ambiguë la désignation, de même qu'il est utile de tenir compte de la taille du pointeur vis à vis de la taille de l'objet afin d'obtenir un placement optimal.

Le point d'action en 3D doit donc être un objet dynamique qui tient compte du contexte applicatif, c'est-à-dire :

1. de la taille des objets proches,
2. de la densité d'objets,
3. du type des tâches que l'utilisateur aura à effectuer (navigation, manipulation, etc.), et qui seront initiées par la désignation,
4. des éventuelles rétroactions des objets désignés,
5. de ses caractéristiques dynamiques (son animation) liées au type de périphérique et à son CDR,
6. des actions de l'utilisateur, et de la prévention des erreurs de ces actions (par exemple, gérer la sortie de l'espace de travail, la perte visuelle du pointeur, etc...).

C'est ce dernier point qui peut aider largement les utilisateurs en termes de retour d'information sur les actions effectuées, le point d'action est au centre de l'attention de l'utilisateur, en lui faisant porter des messages adaptés (rétroaction syntaxique, lexicale et sémantique) et pertinents, on contribue alors à une meilleure perception de l'activité.

Le concepteur d'applications 3D ne peut donc que rarement faire le compromis d'un point d'action par défaut (comme la flèche des boîtes à outils WIMP), il devra repenser les choses suffisamment en amont et intégrer une conception spécifique à son travail de développement.

4.2. Les objets désignés

En trois dimensions, il existe plusieurs stratégies pour déterminer quel(s) objet(s) sont désignés à partir d'un point ou d'une zone d'action. Elles dépendent du nombre de degrés de liberté du périphérique d'entrée et de l'algorithme de calcul des objets désignés.

Nous ne discutons pas ici des types de périphériques utilisables ou des CDR possibles, mais uniquement des principes de sélection des objets 3D (« quel objet est désigné ? ») pour faire de la désignation, du point de vue de la scène 3D. C'est à notre sens le point central, qui permet une classification cohérente des techniques, la plus à même d'aider le concepteur dans ses choix.

Note : Nous présentons les schémas suivants en deux dimensions (en coupe), mais ils correspondent bien sûr à des techniques de désignation en 3D, les explications sont d'ailleurs données pour un espace tridimensionnel.

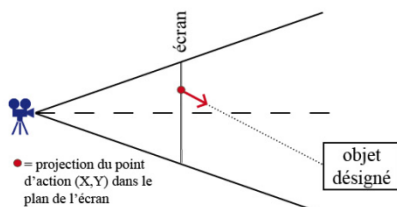
Les méthodes de désignation simples

L'algorithme de désignation ne tient compte ici que des éléments géométriques de la scène. En fonction de la qualité de la modélisation et des besoins de l'application, il pourra agir à différentes granularités de la scène :

- au niveau des données géométriques de base (souvent les facettes),
- au niveau d'un objet déterminé dans le cas d'une première structuration de la scène (liste d'objets présents par exemple),
- au niveau d'un sous-objet si on a une scène plus structurée (en graphe de scène, en arbre, etc).

On peut faire un parallèle avec le niveau lexical d'une scène 3D. On gère alors la désignation en termes de description physique du monde.

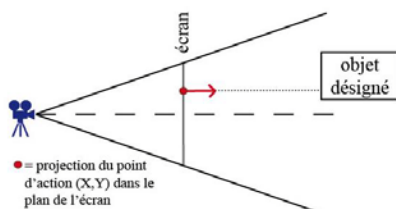
A – Désignation déplacement + orientation



Il s'agit d'un lancer de rayon défini par un point déplacé dans le plan de projection avec un périphérique à 2 DDL (ou déplacé en 3D dans le cas d'un périphérique 3 DDL) et orienté par 2 autres DDL dans l'espace. Le rayon intersecte le(s) objet(s) désigné(s). Cette technique nécessite d'afficher le parcours du rayon afin de

rendre perceptible l'orientation du pointeur.

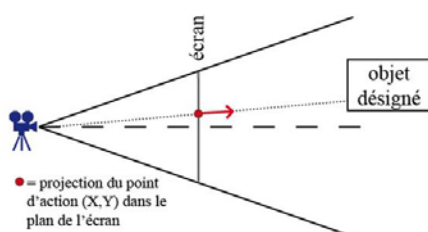
B – Désignation rayon parallèle aux axes



Un rayon est lancé à partir d'un point déplacé dans le plan de projection, orthogonalement à ce plan. Ce type de sélection évite l'occultation par un éventuel pointeur dans le plan de projection de l'objet désigné. Il nécessite d'afficher le rayon car l'orientation n'est pas dans l'axe de l'œil de l'observateur (la caméra

virtuelle). Le fait de déplacer le point de départ en profondeur (périphérique 3 DDL) peut permettre de désigner un objet situé derrière un autre.

C – Désignation rayon dans l'axe de l'observateur

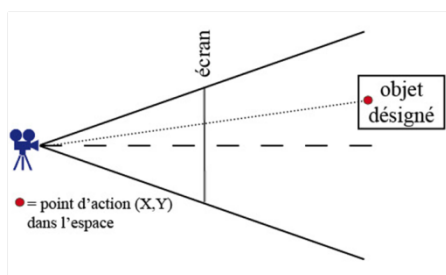


C'est la technique de désignation la plus classique avec un périphérique de type 2 DDL (souris, ...) pour faire de l'interaction 3D. Le rayon part de la caméra et passe par un point déplacé dans le plan de projection. Cette technique de désignation est assez naturelle car elle suit la perspective.

Cependant, on la rencontre souvent utilisée avec un pointeur 2D standard issu directement d'un environnement WIMP pour désigner dans un environnement 3D, ce qui a plusieurs effets de bord :

- l'absence de rétroaction dans l'environnement 3D de la méthode de désignation (par l'affichage d'un rayon, le déplacement du pointeur en 3D, etc) n'aide pas l'utilisateur à percevoir l'environnement en 3D,
- la représentation du pointeur 2D (plane et statique dans le plan de projection) n'est pas faite en 3D avec le modèle d'éclairage ad-hoc, cela contribue à renforcer ce sentiment de sélection dans un plan et nuit à l'homogénéité de l'environnement de travail et donc à sa perception globale,
- en revanche, les actions de désignation apparaissent plus simples mais la scène doit être conçue comme une surface type WIMP (pas de chevauchement ni d'occultation, et suffisamment d'espaces, entre les objets), ce qui est très rarement le cas et limite l'utilisation de la 3D.

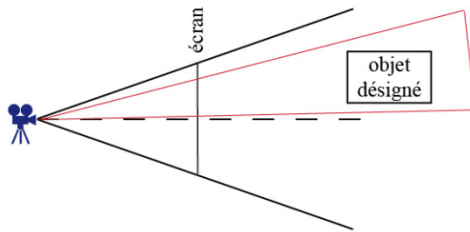
D – Désignation 3D



Elle consiste à déplacer le point d'action directement en 3D à partir d'un périphérique 3 DDL. Le contact du point

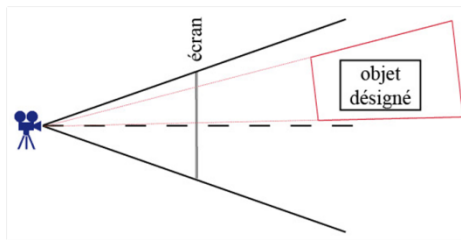
d'action avec un objet permet sa désignation.

E – Désignation par cône 3D



La désignation s'opère sur les objets inclus dans le cône de désignation. Le cône est centré sur un point qui se déplace dans le plan de projection, il peut être manipulé avec 2 DDL. Avec plus de degrés de libertés, on peut paramétrer dynamiquement l'ouverture du cône ou sa longueur.

F – Désignation par un segment de cône 3D

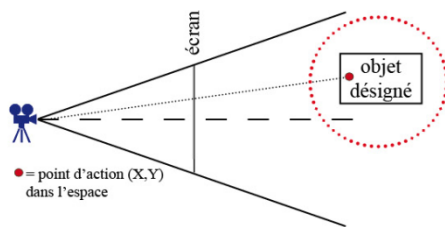


Un segment de cône est ici manipulé en trois dimensions en terme de position à partir d'un périphérique 3 DLL. Avec plus de degrés de libertés, l'orientation peut également être pilotée par le périphérique d'entrée (sinon le sommet du cône reste au niveau de la caméra).

Les méthodes de désignation structurelles

Ces algorithmes de sélection des objets à désigner tiennent compte de la géométrie des objets. Il l'interprète afin de gérer les ambiguïtés de désignation (objets proches, nombreux, petits, etc). On peut faire un parallèle avec le niveau syntaxique d'une scène 3D : on gère alors la désignation en tenant compte de l'organisation de la scène.

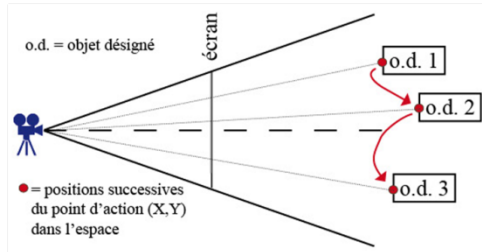
G – Désignation 3D dans les zones englobantes



Elle consiste à désigner un objet en s'en approchant suffisamment pour être dans sa zone préalablement établie. Cela permet de désigner un objet sans forcément le toucher, ce qui permet d'atteindre son objectif plus facilement, en particulier avec des objets petits ou composites, ou encore avec des

environnements denses. La zone peut être une boîte ou une sphère englobante, ou encore une subdivision de l'espace. Si les objets de la scène ont des tailles très variables, le volume de la zone doit être une fonction non linéaire de la taille de l'objet (on doit la grossir pour des objets de petites tailles, et rester proche de la taille réelle de l'objet pour de gros objets).

H – Désignation structurelle



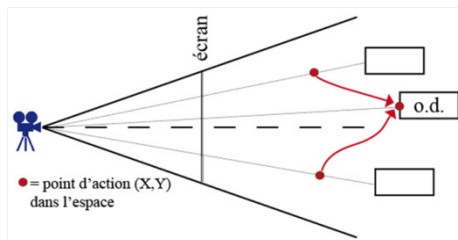
C'est la connaissance de la structure géométrique de la scène qui permet de passer d'un objet à un autre. Le point d'action se déplace en fonction des déplacements du périphérique, et s'arrête sur des positions visibles (qui peuvent être éventuellement pré-calculées) des objets 3D de la scène, avec des techniques de type

aimantation. Cette technique fonctionne avec des périphériques en absolu et en relatif. L'aimantation peut n'avoir lieu qu'à une distance donnée de chaque objet, auquel cas on déplace encore un pointeur dans l'espace.

Les méthodes de désignation évoluées

Ces algorithmes de sélection des objets à désigner tiennent compte de la sémantique des objets. Il l'interprète afin de ne gérer que les objets pour lesquels cela a du sens dans l'application. On peut faire un parallèle avec le niveau sémantique d'une scène 3D : on gère alors la désignation en tenant compte des données représentatives de la sémantique de l'application.

I – Désignation sémantique



On tient compte cette fois de la connaissance du type des objets et de leurs propriétés pour interpréter les déplacements du point d'action vers tel ou tel objet. Le comportement du point d'action dépend dans ce cas fortement du type de tâche que doit réaliser l'utilisateur et du type d'objet manipulé.

Comme la méthode précédente, on peut l'utiliser avec un nombre variable de DDL.

Les environnements virtuels sont souvent très riches, et peuvent facilement distinguer le décor des avatars, des outils, des documents,... et plus encore pour permettre une sélection basée sur la sémantique des objets. Par exemple dans un EVC, la désignation peut se limiter aux seuls avatars, le but de l'application étant de communiquer avec ces derniers. Les autres éléments de l'interface 3D ne sont alors pas pris en compte pour la désignation.

On peut associer à cette méthode un déplacement du point d'action basé sur un CDR sémantique [18], tout comme sur un CDR linéaire ou autre, en effet il s'agit bien de deux aspects différents : le choix de la méthode de déplacement du point d'action d'un côté (CDR), et la méthode de sélection de l'objet désigné de l'autre. Ces deux éléments sont également dissociés dans la méthode que nous proposons (partie suivante).

4.3. Bilan

La désignation d'objets 3D doit respecter et tirer avantage de l'environnement où elle est utilisée : animations et rétroactions sur les actions de l'utilisateur en veillant à leur cohérence, contexte et périphériques, densité et taille des objets, distances, et surtout algorithmique de sélection. Tous ces éléments vus jusqu'ici nous permettent de découper le problème de la désignation en différents aspects qui, réunis, forment une aide à la conception d'une technique de désignation.

5. Guide de conception d'une technique de désignation

Depuis peu, on observe l'émergence de travaux pour adapter les méthodes existantes en ergonomie, en psychologie, en sciences cognitives, en didactique aux spécificités des environnements virtuels [21, 22, 23]. Des guides et méthodes répondant à un problème spécifique sont alors issues des évaluations effectuées. Chacun de ces guides et méthodes répondent à un usage spécifique : allant des propriétés des technologies d'interaction [24, 25] dans ces environnements à la conception de scénarios pour permettre des apprentissages et l'analyse de ces derniers. Beaucoup de disciplines participent à l'enrichissement des connaissances pour améliorer la conception des environnements virtuels. Toutefois il n'existe pas encore à ce jour de guide de conception dédié aux concepteurs eux-mêmes pour choisir la technique de désignation adéquate en fonction du contexte d'usage.

Nous proposons ici un tel guide de conception qui peut être utilisé comme une heuristique par les concepteurs et les experts des environnements 3D. Il peut être utilisé durant les phases de prototypage, de la conception avec des croquis papiers jusqu'aux versions informatiques interactives, par les designers ou les développeurs, afin d'aider à la conception ou d'évaluer leur modèle de désignation.

5.1. Guide de conception

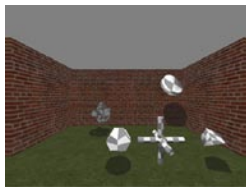
La conception d'un environnement virtuel implique une conception centrée utilisateur, qui tienne compte des tâches à effectuer et du contexte de l'activité. La diversité et la complexité des situations rendent la réalisation de ce type de guide de conception particulièrement ardue. Au delà des méthodes de conception centrées sur l'utilisateur, son activité et son contexte, il est possible de fournir au concepteur des guides généraux sur une phase particulière de la conception, comme celui que nous proposons ci-après sur la désignation 3D. Il généralise les connaissances que nous avons de la désignation.

L'utilisation de ce guide requiert d'être passé par des étapes amont dans la conception et de connaître déjà un certain nombre de choses sur l'activité de l'utilisateur, l'environnement 3D, les types d'objets présents et leurs fonctions.

<p>Choix du périphérique</p> <p>DDL isotonique / isométrique direct / indirect discret / continu</p>	<p><i>détermination du périphérique et de son mode d'utilisation, à partir en particulier du nombre de degrés de liberté nécessaires, d'éventuels retours haptiques. Il doit tenir compte de la tâche et des caractéristiques de l'utilisateur.</i></p>
<p>Filtrage</p> <p>absolu / relatif Control-to-Display Ratio</p>	<p><i>interprétation des données du périphérique, qui consiste à transformer les données du périphérique physique en données adaptées à l'environnement virtuel et au Mode de désignation (voir ci-dessous)</i></p>
<p>Point d'action</p> <p>représentation(s) 3D modèle d'animation gestion de l'orientation dynamique (variation taille, etc)</p>	<p><i>détermination de la forme et du comportement du (des) point(s) d'action, en particulier si le pointeur change d'apparence en fonction de la densité ou du type d'objets, ou si son orientation ou sa taille changent en fonction de la position et de la taille des objets.</i></p>
<p>Mode de désignation</p> <p>rayon / point / volume (cone,...) syntaxique / sémantique</p>	<p><i>choix de l'algorithme de sélection des objets (cf. partie 4.4), des règles de désignation : mode de calcul de la distance entre objets, critère de choix de l'objet le plus proche.</i></p>
<p>Rétroaction directe sur les objets sélectionnables</p> <p>modification de l'environnement modification du point d'action indices visuels / virtuels</p>	<p><i>affichage des rétroactions, des indices qui permettent d'aider à percevoir l'espace 3D, le mode de fonctionnement de la désignation, et surtout les objets proches d'être désignés. Si des objets sont non sélectionnables, il faut que cela apparaisse clairement (grisé, appartenance au décor, etc).</i></p>

<p>Rétroaction sur les objets sélectionnés</p> <p>visibilité des objets sélectionnés mise en valeur des objets sélectionnés</p>	<p><i>indiquer clairement quels sont le ou les objets finalement désignés (puis éventuellement sélectionnés) : changement de couleur, d'échelle, d'éclairage, ou bien de forme (encadrement, détourage, etc)</i></p>
<p>Mode de désélection</p> <p>techniques pour revenir à un état sans objet sélectionné</p>	<p><i>Un mode simple pour relâcher le(s) objet(s) en cours de sélection doit être conçu. Il doit être fait en relation avec les éventuels modes de manipulation des objets afin d'enchaîner de façon lisse les différentes modalités (désignation / sélection / manipulation)</i></p>

5.2 Illustration



Un exemple simple² a été développé afin d'illustrer ce guide sur une technique de désignation sémantique 3D :

- Choix du périphérique : souris (2D + molette en Z)
- Filtrage : CDR sémantique, le point d'action ralentit dans l'espace visuel lorsque l'on traverse un objet.
- Point d'action : un modèle 3D statique de croix, projection des ombres portées de la croix et des objets désignables.
- Mode de désignation : sélection d'un objet lorsque le point d'action traverse son volume englobant (ici une boîte englobante parallèle aux axes). L'objet doit appartenir à une liste donnée d'objets (cf. désignation sémantique, partie 4.4).
- Rétroaction sur les objets désignés : les objets deviennent transparents.

5.3 Travaux futurs

La conception des différentes techniques d'interaction dans un environnement virtuel nécessite de prendre en compte simultanément la navigation, la désignation et la manipulation des objets. Elle doit se faire en relation avec les tâches que veulent réaliser les utilisateurs. Dans la suite de nos travaux sur ces différents aspects, nous

² Un exemple de pointage sémantique 3D : <http://www.emn.fr/x-info/cdumas/designation3D/>

continuerons à modéliser des heuristiques, avec l'objectif à terme de proposer un ensemble cohérent de guides de conception pour les environnements virtuels.

6. Bibliographie

- [1] Cadoz, C., *Le geste canal de communication homme/machine : la communication "instrumentale"*, Technique et science informatique, 13(1), pp. 31-61, 1994.
- [2] Kabbash, P. et Buxton, W., *The "prince" technique: Fitts' law and selection using area cursors*, Proceedings of ACM CHI Conference on Human Factors in Computing systems (CHI'95), pp. 273-279, 1995.
- [3] Worden, A., Walker, N., Bharat, K. et Hudson, S., *Making computers easier for older adults to use: area cursors and sticky icons*, Proceedings of ACM CHI Conference on Human Factors in Computing systems (CHI'97), pp. 266-271, 1997.
- [4] Grossman, T. et Balakrishnan, R., *The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area*, Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'05), pp. 281-290, 2005.
- [5] Zhai, S., Buxton, W. et Milgram, P., *The "silk cursor": investigating transparency for 3D target acquisition*, Proceedings of ACM CHI Conference on Human Factors in Computing systems (CHI'94), pp. 459-464, 1994.
- [6] Steed, A. et Parker, C., *3D selection strategies for head tracked and non head tracked operation of spatially immersive displays*, 8th International Immersive Projection Technology Workshop, 2004.
- [7] Robertson, G., Czerwinski, M., Baudisch, P., Robbins, D., Smith, G. et Tan, D., *The large-display user experience*, IEEE Computer Graphics and Applications, 25(4), pp. 44-51, 2005.
- [8] Grossman, T. et Balakrishnan, R., *The design and evaluation of selection techniques for 3D volumetric displays*, Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'06), p. 3-12, 2006.
- [9] Ramos, G., Robertson, G., Czerwinski, M., Tan, D., Baudisch, P., Hinckley, K. et Agrawata, M., *Tumble! Splat! Helping users access and manipulate occluded content in 2D drawings*, Proceedings of the working conference on Advanced visual interfaces (AVI'06), p. 428-435, 2006.
- [10] Dragicevic, P., *Combining crossing-based and paper-based Interaction paradigms for dragging and dropping between overlapping windows*, Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'04), pp. 193-196, 2004.
- [11] Benko, H., Wilson, A. et Baudisch, P., *Precise Selection Techniques for Multi-Touch Screens*, Proceedings of ACM CHI Conference on Human Factors in Computing systems (CHI'06), pp. 1263-1272, 2006.
- [12] Aliakseyeu, D., Nacenta, M., Subramanian, S. et Gutwin, C., *Bubble radar; efficient pen-based interaction*, Proceedings of the working conference on Advanced visual interfaces (AVI'06), pp 19-26, 2006.

- [13] Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B. et Zierlinger, A., *Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems*, Proceedings of the INTERACT' Conference, pp. 57-64, 2003.
- [14] Hascoët, M., *Throwing models for large displays*, Proceedings of the British HCI Group Conference (HCI'03), pp. 73-77, 2003.
- [15] Collomb, M., Hascoet, M., Baudisch, P. and Lee, B., *Improving drag-and-drop on wall-size displays*, Proceedings of the 2005 Conference on Graphics Interface, pp. 25-32, 2005.
- [16] Parker, J., Nunes, N. Mandryk, R., Inkpen, K., *TractorBeam selection aids: improving target acquisition for pointing input on tabletop displays*, technical report CS-2004-10, université d'Halifax, Nouvelle-Écosse, Canada, 2004.
- [17] Poupyrev, I., Billingham, M., Weghorst, S. et Ichikawa, T., *The go-go interactive technique, non linear mapping for direct manipulation in VR*, Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'96), pp. 76-80, 1996.
- [18] Blanch, R., Guiard, Y., Beaudoin-Lafon, M., *Semantic pointing: improving target acquisition with control-display ratio adaptation*, Proceedings of ACM CHI Conference on Human Factors in Computing systems (CHI'04), pp. 519-526, 2004.
- [19] Guiard, Y., Blanch, R., Beaudoin-Lafon, M., *Object pointing: a complement to bitmap pointing in GUIs*, Proceedings of the 2004 Conference on Graphics Interface, pp. 9-16, 2004.
- [20] Elmqvist, N., Fekete, J.-D., *Semantic Pointing for Object Picking in Complex 3D Environments*. Proceedings of the 2008 Conference on Graphics Interface, pp. 243-250, 2008.
- [21] Dubois E, Mansoux B., Bach C, Scapin D, Masserey G, Viala J., *un modèle préliminaire des domaines des systèmes mixtes*, Actes de la 16^{ème} conférence de l'Association Francophone d'Interaction Homme-Machine (IHM'04), pp. 61-68, 2004.
- [22] Bach, C, Scapin D.L *Adaptation des critères ergonomiques aux interactions hommes environnement virtuel*, Actes de la 15^{ème} conférence de l'Association Francophone d'Interaction Homme-Machine (IHM'03), pp 24-31, 2003.
- [23] Burkhart, J.M., Plénacoste P., Perron, L., *Concevoir et évaluer l'interaction utilisateur-environnement virtuel*, dans le Traité de la Réalité Virtuelle, volume 2, éditions Ecole des Mines de Paris, pp. 473-521, 2006.
- [24] Hillaire, S., Lécuyer, A., Cozot, R., & Casiez, G., *Using an Eye-Tracking System to Improve Depth-of-Field Blur Effects and Camera Motions in Virtual Environments*, Proceedings of the 15th Annual IEEE Virtual Reality 2008, pp. 47-50, march 2008.
- [25] Casiez, G., & Chaillou, C. *Effects of DOF Separation on Elastic Devices for the Navigation in 3D Virtual Environments with Force Feedback*, Proceedings of IEEE World Haptics 2005, 483-486, 2005.

Cédric Dumas est maître assistant à l'école des Mines de Nantes et à l'IRCCyN. Spécialisé en interaction homme machine, il mène des recherches sur l'interaction dans les environnements virtuels, ses caractéristiques visuelle et motrice, les styles d'interaction associés, pour des utilisateurs seuls ou en situation collaborative.

Frédéric Jourdan est ingénieur de recherche à l'IRCCyN. Spécialisé sur les problèmes d'infographie et de modélisation 3D, il travaille sur des projets touchant à la visualisation d'information, à la réalité virtuelle ou encore à la CAO.

Patricia Plénacoste est maître de conférence en psychologie au LIFL et au centre INRIA Futurs de Lille. Elle s'intéresse aux problèmes liés à l'usage des environnements virtuels, dont la désignation et la manipulation d'objets en environnement virtuel 3D, avec une approche centrée sur la cognition et l'interaction instrumentale.

Laurence Perron est psychologue-ergonome et mène des recherches sur la réalité virtuelle, augmentée et mixte au sein de Orange Labs Lannion. Elle travaille plus particulièrement sur les environnements virtuels collaboratifs en lien avec les diverses formes de collaboration humaine (rôle des représentations humaines, non verbal vs verbal, asymétrie, typologie de tâches...). Elle s'intéresse plus particulièrement aux règles de construction et d'évaluation de ces environnements, avec une approche socio-technique centrée sur l'utilisateur.

Amine Chellali termine un doctorat en informatique dans l'équipe Robotique de l'IRCCyN. Ses travaux portent sur les environnements virtuels collaboratifs. Il s'intéresse plus particulièrement à la construction des référentiels d'action dans la coopération à distance et l'élaboration d'un référentiel commun entre les participants.

