



HAL
open science

Interaction en réalité augmentée mobile : exploration et pointage de cibles numériques

Patrick Perea

► **To cite this version:**

Patrick Perea. Interaction en réalité augmentée mobile : exploration et pointage de cibles numériques. Interface homme-machine [cs.HC]. Université Grenoble Alpes [2020-..], 2020. Français. NNT : 2020GRALM028 . tel-03012809

HAL Id: tel-03012809

<https://theses.hal.science/tel-03012809v1>

Submitted on 18 Nov 2020

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.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Patrick PEREA

Thèse dirigée par **Laurence NIGAY**, Professeur, Université Grenoble Alpes, et
co-encadrée par **Denis MORAND**, Architecte Senior, Schneider Electric

préparée au sein du **Laboratoire d'Informatique de Grenoble**
dans l'**École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

**Interaction en réalité augmentée mobile :
exploration et pointage de cibles numériques**

**Mobile augmented reality interaction: digital
objects exploration and pointing**

Thèse soutenue publiquement le **6 Juillet 2020**,
devant le jury composé de :

Monsieur Georges-Pierre BONNEAU

Professeur, Université de Grenoble-Alpes, Président

Monsieur Martin HACHET

Directeur de recherche, Inria Bordeaux-Sud-Ouest, Rapporteur

Monsieur Éric LECOLINET

Professeur, Télécom Paris, Rapporteur

Madame Nadine COUTURE

Professeur, ESTIA, Examinatrice

Monsieur Denis MORAND

Architecte Senior, Schneider Electric, Co-encadrant de thèse

Madame Laurence NIGAY

Professeur, Université Grenoble Alpes, Directrice de thèse



Abstract

This thesis contributes to the research field of Human-Computer Interaction (HCI). The focus of the research is on user interaction with handheld Augmented Reality (AR) systems.

AR allows the addition of digital content, primarily graphics, to the user's physical environment. The resulting mixed environment includes digital objects registered in the physical world. This mixed environment, partially visible on the mobile device's screen, defines several constraints for interaction. Our work is dedicated to the selection of a digital target in this mixed environment and we address two questions : (1) how to improve the perception of the mixed environment beyond what is perceived by the camera's field of view of the mobile device, for finding off-screen digital targets (2) how to enhance pointing at a digital target registered in the physical environment. Our contributions answer these two questions.

We first propose three interactive off-screen guidance techniques, which we evaluate by conducting two laboratory experiments. These techniques are based on Halo, a visualization technique of off-screen objects based on circles, and differ in the way the aggregation of the off-screen objects is displayed on screen. The results of the two experiments show that our three techniques effectively extend users' knowledge of the mixed environment, and limit the visual intrusion on the mobile device's screen in comparison with the traditional arrow-based visualization technique.

We then define two interaction techniques enhancing the selection of digital targets. These techniques are based on (1) target expansion techniques, which facilitate target selection by allocating a larger active area to each target and (2) a cursor jump, which shortens the distance between the cursor and the content of the digital target. The two techniques differ in the way users manipulate the cursor once it has jumped to the digital target. We propose two possibilities : physical pointing by moving the mobile device to the desired content, or relative pointing using thumb strokes on the screen. The result of a laboratory experiment confirms that target expansion techniques improve the pointing performance, while relative pointing is preferred by participants.

Our contributions are applied to the field of industrial maintenance, in charge of repairing or preventing failures on production machines. As part of a Schneider Electric-CIFRE thesis, the Schneider Electric's *Augmented Operator Advisor* product (AR maintenance assistance application) includes one of the proposed targeting assistance techniques.

Résumé

Cette thèse s'inscrit dans le domaine de l'Interaction Homme-Machine (IHM) et est dédiée à l'interaction en Réalité Augmentée (RA) sur dispositifs mobiles.

La RA permet l'ajout de contenu numérique, principalement graphique, à l'environnement physique de l'utilisateur. L'environnement mixte ainsi obtenu comprend des objets numériques ancrés dans le monde physique. Cet environnement mixte partiellement perceptible sur l'écran du dispositif mobile définit plusieurs contraintes pour l'interaction. Nos travaux sont dédiés à la sélection d'une cible numérique dans cet environnement mixte. Pour cela les deux questions traitées sont : (1) comment améliorer la perception de l'environnement mixte au-delà de ce qui est perçu par le champ de vision de la caméra du dispositif mobile, pour trouver une cible numérique (2) comment améliorer le pointage sur dispositif mobile d'une cible numérique ancrée dans l'environnement physique. Nous apportons des contributions à ces deux questions.

Nous proposons dans un premier temps trois techniques interactives de guidage hors-champ, que nous évaluons au travers de deux études expérimentales en laboratoire. Ces techniques sont basées sur Halo, une solution permettant la visualisation d'objets hors-champ à l'aide de cercles, et diffèrent par la manière de visualiser l'agrégation des cibles numériques hors-champ. Les résultats des deux études montrent que nos trois techniques étendent efficacement la connaissance de l'utilisateur sur l'environnement mixte hors-champ, et limitent l'intrusion visuelle à l'écran du dispositif mobile par rapport aux solutions traditionnelles à base de flèches.

Nous avançons ensuite deux techniques d'interaction pour faciliter le pointage de cibles numériques en RA. Ces techniques sont basées sur (1) les techniques d'expansion de cibles, qui facilitent la sélection de cibles en allouant à chaque cible une zone active plus grande et (2) un saut de curseur, qui raccourcit la distance entre le curseur et les informations regroupées au sein de la cible numérique. Les deux techniques diffèrent par la manière de manipuler le curseur une fois qu'il a sauté sur la cible numérique. Nous offrons deux possibilités : pointage physique en déplaçant l'appareil mobile vers l'information souhaitée, ou pointage relatif en utilisant le pouce sur l'écran. Le résultat d'une étude expérimentale en laboratoire confirme que les techniques d'expansion de cibles diminuent le temps nécessaire à l'acquisition d'une cible, tandis que la manipulation relative du curseur avec le pouce est très largement préférée par les participants.

Nos contributions sont appliquées au contexte de la production industrielle, et plus spécifiquement à la maintenance industrielle en charge de réparer ou de prévenir les pannes sur les machines de production. L'entreprise partenaire de cette thèse CIFRE, Schneider Electric, intègre dans le produit *Augmented Operator Advisor* (application de RA d'aide à la maintenance) une des techniques d'aide au pointage proposées.

Remerciements

Je remercie les membres de mon jury d'avoir participé à la validation de ces trois ans de thèse. Je remercie en particulier George-Pierre Bonneau qui a présidé ce jury, les deux rapporteurs Eric Lecolinet et Martin Hachet pour leurs retours et commentaires sur le manuscrit, ainsi que Nadine Couture d'avoir accepté de faire partie de ce jury.

Un grand merci à Laurence, pour ta présence, ton implication et la motivation que m'as apporté. Tu as toujours été disponible pour m'aider dans cette thèse. Travailler avec toi a été un réel plaisir.

Je remercie Denis Morand et Maxime Faraco, pour m'avoir accompagné pendant ces 3 ans côté Schneider Electric. Vous avez fait preuve d'une grande écoute et avez toujours été disponibles pour répondre à mes questions.

Je souhaite remercier chaleureusement mes parents et mon frère, qui ont été d'un soutien indéfectible pendant ces 3 ans et tout au long de mes études.

Enfin, je remercie mes collègues de l'équipe IIHM, dont certains sont devenus des amis, et sans qui ces trois ans de thèse auraient été plus mornes. Merci à vous, qui faites vivre le labo.

Table des matières

1	Introduction	18
1.1	Domaine	18
1.2	Questions de recherche	20
1.3	Contributions	21
1.4	Collaboration industrielle : domaine applicatif	22
1.5	Structure du manuscrit	23
2	Contexte et application cible : maintenance industrielle	25
2.1	Réalité augmentée pour la maintenance	25
2.2	Tâches de maintenance	26
2.3	Trois scénarios d’usage de la réalité augmentée	26
2.4	Application cible existante	30
2.4.1	Architecture globale	30
2.4.2	Interface utilisateur	30
2.4.3	Système de suivi	31
2.5	Conclusion	33
I	Techniques de visualisation de l’espace hors-champ	34
3	Etat de l’art	35
3.1	Introduction	35
3.2	Le besoin en RA mobile	36
3.3	Approches pour la visualisation de l’espace hors-champ	37
3.3.1	Zoom	37
3.3.2	Overview+Detail	39
3.3.3	Focus+Context	41
3.3.4	Vues contextuelles	43
3.3.5	Autre modalité non graphique, la modalité lumineuse	44
3.3.6	Ce qu’il faut retenir	45
3.4	Vues contextuelles	46
3.4.1	Indices entièrement visibles à l’écran	46
3.4.2	Indices partiellement visibles à l’écran	51

3.4.3	Synthèse des vues contextuelles	55
3.5	Conclusion	57
4	Conception de techniques de visualisation hors-champ pour la réalité augmentée mobile	58
4.1	Introduction	58
4.2	Adaptation de Halo pour la Réalité Augmentée	59
4.2.1	Projection	59
4.2.2	Intrusion visuelle constante	60
4.2.3	Gestion de la profondeur	61
4.2.4	POIs devant/derrière l'utilisateur	67
4.3	Agrégation de POIs hors-champ	68
4.3.1	Halo3D : Barycentre	68
4.3.2	Halo3D* : Chevauchements autorisés	71
4.3.3	Halo3D** : Courbes	73
4.4	Conclusion	74
5	Exploration expérimentale de trois techniques de visualisation hors-champ pour la réalité augmentée mobile	75
5.1	Introduction	75
5.2	Expérience 1 : Tâche abstraite de pointage	76
5.2.1	Introduction	76
5.2.2	Plan d'expérience	76
5.2.3	Résultats	81
5.2.4	Discussion	83
5.2.5	Conclusion de l'expérience 1	84
5.3	Expérience 2 : Tâche d'exploration	85
5.3.1	Introduction	85
5.3.2	Plan d'expérience	85
5.3.3	Résultats	91
5.3.4	Discussion	92
5.3.5	Conclusion de l'expérience 2	95
5.4	Conclusion	95
II	Techniques de pointage en RA	96
6	Etat de l'art	97
6.1	Introduction	97
6.2	Le besoin en RA mobile	99
6.3	Modélisation du pointage	100
6.3.1	Modélisation 2D	100
6.3.2	Modélisation 3D	101
6.4	Assistance au pointage 2D	102

6.4.1	Pointage direct	102
6.4.2	Pointage indirect	105
6.5	Assistance au pointage en Réalité Augmentée mobile	110
6.5.1	Pointage direct	110
6.5.2	Pointage indirect	112
6.5.3	Stabilisation du flux d'entrée	114
6.6	Mise en contexte de l'interaction	115
6.7	Synthèse et conclusion	116
7	Conception de techniques de pointage pour la réalité augmentée mobile	120
7.1	Introduction	120
7.2	Expansion de cibles en Réalité Augmentée	122
7.2.1	Adaptation à la Réalité Augmentée	123
7.2.2	Retour visuel	124
7.3	Sélection d'un élément du menu d'un POI	127
7.4	Conclusion	132
8	Exploration expérimentale des deux techniques de pointage pour la réalité augmentée mobile	134
8.1	Introduction	134
8.2	Plan d'expérience	135
8.2.1	Appareil et participants	135
8.2.2	Techniques de visualisation développées	135
8.2.3	Tâche expérimentale et environnement	135
8.2.4	Design expérimental et procédure	138
8.2.5	Hypothèses	139
8.2.6	Données collectées	139
8.3	Résultats	142
8.4	Discussion	143
8.5	Conclusion de l'expérience	146
9	Conclusion	147
9.1	Contributions	147
9.1.1	Techniques de visualisation hors-champ en RA	147
9.1.2	Techniques de pointage en RA	148
9.1.3	Diffusion et brevet	149
9.2	Perspectives	150
9.2.1	Perspectives à court terme : extension des travaux effectués	150
9.2.2	Perspectives à long terme : ouverture vers d'autres travaux	153
	Annexes	155
	A System Usability Scale	156

B RAW TLX (Task Load Index)	157
C Questionnaire comparatif des études expérimentales 1, 2 et 3	158
Bibliographie	161

Table des figures

1.1	Continuum Réalité-Virtualité de Milgram pour décrire différentes combinaisons d'éléments réels et virtuels. Image inspirée de [38].	18
1.2	Processus de fonctionnement de la réalité augmentée.	19
1.3	Application de réalité augmentée pour la maintenance de machine de production développée par Schneider Electric. L'application permet d'examiner l'état d'un moteur, les statistiques de la machine, ou encore de consulter une vidéo d'entraînement. Image issue de [37].	23
2.1	Accès aux données d'aide à la maintenance d'une armoire électrique, regroupées sous forme de POIs (a) 3 POIs fermés placés sur une armoire électrique. (b) 2 des 3 POIs ouverts, affichant des informations sur la température et la pression de la machine, ainsi que la possibilité d'accéder à deux documentations techniques et à une vidéo explicative. (c) Lancement de la vidéo explicative sur le montage et le démontage d'un composant de la machine (image issue de [80]).	27
2.2	(a) Opérateur de maintenance devant une chaîne cuboïde de montage. Les points d'intérêt de maintenance RA sont situés sur toute la longueur de la ligne. (b) Opérateur de maintenance au centre d'une ligne d'équipements en forme de U. Les points d'intérêts (POIs) de maintenance RA sont situés autour de l'utilisateur.	27
2.3	Exemple de sélection d'une information technique sur une machine à l'aide (b)-(d) du curseur centré sur l'écran et (c)-(e) d'un pointage direct avec le doigt. (a) 6 POIs placés sur une armoire électrique. (b) Sélection d'un POI en l'alignant avec le curseur central et en appuyant sur le bouton valider. (c) Sélection d'un POI en le touchant directement avec le doigt. (d) Ouverture d'un menu à 3 items et sélection du deuxième élément en alignant le curseur avec l'élément et en appuyant sur le bouton valider. (e) Ouverture d'un menu à 3 items et sélection du deuxième élément en le touchant directement avec le doigt.	29
2.4	Architecture de l'application RA d'aide à la maintenance.	31
2.5	Interface de l'application de RA <i>Vijeo</i> : schéma électrique superposé à la machine et reliant plusieurs POIs	32

3.1	Interfaces de zoom proposées en RA : en haut, zoom égocentrique élargissant le champ de vision de l'utilisateur jusqu'à 360 degrés. En bas, zoom excentrique offrant une vue radar 2D. Image issue de [82].	38
3.2	Interface Overview+Detail affichée sur a) ordinateur (image issue de Google Maps) b) téléphone mobile (image issue de [26])	39
3.3	a) Application <i>Spatial Wearable Conference Space</i> , affichant sur une carte l'emplacement des participants à la conférence. Image issue de [19] b) Application <i>Google Maps RA</i> , affichant sur la carte l'itinéraire à suivre.	41
3.4	a) La distorsion est appliquée à la vue entière. b) Le focus et le contexte ne sont pas déformés. Seule la zone Glue présente une distorsion. Image issue de [116].	41
3.5	Application AR Magic Lens. a) Marqueurs visuels disposés sur une table (pour y superposer le globe terrestre) et poignée tenue par l'utilisateur où est placé un autre marqueur (pour y superposer la loupe). b) Loupe (vue détaillée) permettant de visualiser un ensemble de données propre du globe terrestre virtuel (vue globale) tel que la couche de nuages. Image issue de [20].	42
3.6	Vue contextuelle à l'aide de CityLights : des lignes épaisses sur les bordures des fenêtres indiquent la position, taille et distance des objets textuels non visibles. Les triangles dans les coins des fenêtres indiquent que davantage d'objets invisibles se trouvent dans l'espace hors-champ associé. Image issue de [73].	43
3.7	a) Sparkle avec tous les LEDS allumés. b) LEDs placés sur la face arrière d'une tablette. Image issue de [81].	44
3.8	a) Système MonocularAR attaché au casque HoloLens. b) LEDs placés sur le contour d'une moitié de la visière de l'HoloLens. Image issue de [50].	45
3.9	Flèches dont la distance aux POIs est symbolisée par a) la taille b) la longueur. Image issue de [27].	47
3.10	a) Visualisation <i>EdgeRadar</i> : les carrés bleus sur le bord de l'écran indiquent la position et la distance des POIs hors-champ. Image issue de [58]. b) <i>EdgeSplit</i> : l'utilisateur sélectionne les POIs hors-champ directement sur l'écran. Image issue de [64].	48
3.11	Visualisation <i>FlyingARrow</i> . (a)-(d) Objet hors-champ en rouge, champ de vision de l'utilisateur en gris. (b) Flèche 3D dans le champ de vision de l'utilisateur. (c) Animation de la flèche vers l'objet hors-champ. (d) Son dans le casque de RA lorsque la flèche atteint le POI. Image issue de [49].	48
3.12	a) Projection des coordonnées 3D en une vue fisheye 2D. (b)-(c) Visualisation Aroundplot pour 50 POIs hors-champ lorsque b) l'utilisateur est statique c) l'utilisateur déplace l'appareil mobile vers le haut et la droite. Images issues de [65].	49
3.13	Visualisation EyeSee360 ici appliquée à la réalité virtuelle. EyeSee360 est affiché sur un téléphone, lui-même inclus dans un casque de réalité virtuelle Google Cardboard. Image issue de [48].	50

3.14	Visualisation EyeSeeX affichée sur un simulateur. (a) Visualisation de trois objets hors-champ sans réduction ou compression de l'information. (b) Stratégie 1 : réduction de l'information par élimination des objets situés derrière l'utilisateur. (c) Stratégie 2 : compression de l'information. Image issue de [45].	51
3.15	a) Halos indiquant l'emplacement de 5 POIs hors-champ (Image issue de [13]). b) Chevauchements entre halos pour 8 POIs hors-champ (Image issue de [27]). c) Double arc agrégeant des halos qui se chevauchent (Image issue de [13]). d) Visualisation haloDot (Image issue de [43]).	52
3.16	a) Wedges indiquant l'emplacement de 5 POIs hors-champ (Image issue de [59]). b) Chevauchements entre wedges pour 30 POIs hors-champ (Image issue de [28]).	53
3.17	3D halos indiquant la position de 3 POIs : le premier situé sur le côté gauche de l'espace hors-champ et placé devant la caméra, le deuxième situé sur le côté gauche de l'espace hors-champ et placé derrière la caméra, le troisième situé derrière le building mis en évidence en vert. Image issue de [105].	54
3.18	(a) Projection de la technique Halo sur une sphere autour de la tête de l'utilisateur (image issue de [46]). (b) Visualisation Halo, Wedge et flèches dans un casque de RA (image issue de [52]).	55
4.1	Projection d'un POI hors-champ sur le plan de l'écran et calcul du halo correspondant	59
4.2	Dynamisme du halo lors d'un mouvement de gauche à droite (a) un POI dans le champ de vision de la caméra. (b) Après un déplacement de gauche à droite de la caméra, le POI est hors-champ et est indiqué par un arc de cercle. (c) L'arc de cercle est agrandi car la distance entre la projection du POI et le bord de l'écran a augmenté.	60
4.3	a) Calcul du rayon des halos à partir de la droite passant par P' jusqu'au centre de l'écran. b) Visualisation des halos de 100 POIs sans agrégation : l'intrusion visuelle des halos est plus grande dans les coins de l'écran. . . .	62
4.4	a) Calcul du rayon des halos en considérant un rectangle interne. b) Visualisation des halos de 100 POIs sans agrégation : le rectangle interne devient perceptible.	62
4.5	a) Calcul du rayon des halos en considérant un rectangle interne à bords arrondis. b) Visualisation des halos de 100 POIs sans agrégation : le rectangle interne à bords arrondis devient perceptible.	62
4.6	Exemples d'intrusion de halos dans le coin d'un écran de 2560 pixels de largeur en utilisant la méthode du rectangle à bords arrondis, dont les cercles possèdent un rayon égal à (a) 5% de la largeur de l'écran, (b) 10% de la largeur de l'écran, (c) 20% de la largeur de l'écran.	63

4.7	Halos et profondeur (a) Vue orthographique par-dessus de 2 POIs hors-champ devant et derrière une machine. (b) Visualisation des 2 POIs : les deux halos ne permettent pas de connaître la différence de profondeur entre les deux POIs. (c) Visualisation des 2 POIs avec la profondeur traduite par la transparence du halo. Le halo le plus opaque à gauche indique que le POI correspondant est plus proche que le deuxième POI.	64
4.8	Visualisation de la distance entre l'utilisateur et un POI hors-champ, obtenue en couplant le trait visuel <i>épaisseur</i> avec la transition <i>atténuée</i> (a-b) ou <i>abrupte</i> (c-d). Avec la transition <i>atténuée</i> , le halo devient de plus en plus fin à mesure que l'utilisateur se rapproche du POI hors-champ. Avec la transition <i>abrupte</i> , le halo devient de plus en plus épais à mesure que l'utilisateur se rapproche du POI hors-champ.	64
4.9	Vue orthographique par-dessus de 3 POIs hors-champ. Un seul des trois POIs est significativement plus proche de l'utilisateur que des deux autres POIs.	65
4.10	Dynamisme du halo lors d'un mouvement pour atteindre un POI situé derrière l'utilisateur (a) le halo, affiché comme une ligne droite se trouve derrière l'utilisateur. (b-c) Déplacement de gauche à droite de la caméra, le POI se rapproche du champ de vision de l'utilisateur, et la taille du halo diminue. (d) Le POI est maintenant visible dans le champ de vision de l'utilisateur, le halo a disparu de l'écran.	68
4.11	Agrégation de 8 POIs (a) 8 POIs visibles sur l'écran. (b) L'utilisateur se déplace de gauche à droite. 3 halos s'agrègent. (c) Tous les POIs sont agrégés.	69
4.12	Processus de désagrégation. a) Recherche des nœuds dont les halos fils ne se chevauchent plus ou n'existent plus. b) Élimination des parents du nœuds désagrégés.	70
4.13	Halo3D guide l'utilisateur vers le POI P en deux étapes : (a) Halo3D guide d'abord l'utilisateur vers le barycentre des POI agrégés (b) une fois que les POIs se sont désagrégés, Halo3D guide vers les POI P.	71
4.14	Trois techniques avec agrégation. (a) Halo3D, affichant le halo du barycentre du cluster. (b) Halo3D*, chevauchements entre halos autorisés conformément à un seuil. (c) Halo3D**, POIs agrégés affichés à l'aide d'une courbe, révélant la distribution spatiale des POIs du cluster.	72
4.15	(a) : Étapes 1 et 2 - Tous les halos et leurs intersections sont calculés. L'arc (AB) est contenu dans un halo et ne fera pas partie de la courbe finale. (b) : Étape 3 - Les arcs composant la courbe finale sont calculés. (c) : Étape 4 - Exemple d'une courbe quadratique reliant deux arcs, en utilisant trois points (x1, X, x2) et trois tangentes (lignes en pointillés noirs). (d) : Étape 5 - Le nombre de points d'intérêt agrégés est affiché au sein de la courbe.	73
5.1	Techniques de visualisation développées (a) Arrows2D (b) AroundPlot* (c) Halo3D.	77

5.2	Validation d'un POI avec la technique Halo3D : (a) un cercle vert indique la direction à suivre pour atteindre la cible, (b) la cible verte rentre sur l'écran, (c) la tâche est validée par l'alignement du curseur avec la cible . . .	78
5.3	Etude 1 : environnements générés avec 100 POIs (a) placés sur une sphère de support (b) disposés en 3 clusters	79
5.4	Déroulement de l'expérience 1	79
5.5	Participant effectuant une rotation sur place pour chercher un POI hors-champ	80
5.6	Temps mesurés lors de l'expérience	81
5.7	Temps de pointage hors-champ en secondes pour les 3 techniques et les 2 amplitudes de mouvement. Les barres d'erreur révèlent des intervalles de confiance de 95%.	81
5.8	Tâche de localisation : capture d'écran de la grille. En rouge : cellules sélectionnées par un participant. En jaune : champ de vision de la caméra de l'appareil mobile.	86
5.9	Tâche d'exploration : (a) Sélection d'un POI en utilisant Halo3D** (b) Un participant se déplace devant une réplique d'une machine de production pour chercher les POIs.	87
5.10	Exemple d'un environnement contenant 35 POIs et emplacement de départ de l'utilisateur	87
5.11	Déroulement de l'expérience 2	88
5.12	Taux d'erreur dans les environnements à faible et à forte densités. Les barres d'erreur indiquent des intervalles de confiance à 95%.	91
5.13	Temps d'exploration pour toutes les techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.	91
5.14	Notes sur 5 (1 = très mauvais, 5 = très bien) données aux techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.	91
5.15	Intrusion à l'écran des techniques sur 5 (1 = très mauvais, 5 = très intrusif) selon les participants. Les barres d'erreur indiquent des intervalles de confiance à 95%.	92
5.16	(a) Halo3D* affichant un arc complet pour chaque cluster (b) Halo3D** n'affichant qu'une seule courbe lorsque le nombre de POIs est élevé.	93
6.1	(a) Pointage indirect : les deux mains tiennent le dispositif mobile et la sélection s'effectue en alignant le croix centrale avec la cible. (b) Pointage direct : une seule main tient le dispositif mobile et la sélection s'effectue en touchant la cible avec un doigt de l'autre main. (c) Pointage direct : la main qui tient le dispositif mobile sert aussi à interagir. La sélection s'effectue alors avec le pouce de la main qui tient le dispositif.	98
6.2	Technique de sélection Thumbspace a) configuration d'une zone qui servira de touchpad, (b) miniature de l'interface affichée dans Thumbspace. Image issue de [66].	103

6.3	Technique de sélection Escape. (a) L'utilisateur appuie son doigt près de la cible souhaitée. (b) L'utilisateur effectue un geste dans la direction indiquée par la cible. (c) La cible est sélectionnée, malgré la forte densité de POIs. (d) Exemple d'application sur écran tactile. Image issue de [119].	104
6.4	Technique de sélection 2DDragger : (a) la cible la plus proche du doigt de l'utilisateur est pré-sélectionnée, (b) l'utilisateur passe de cible voisine en cible voisine en faisant glisser son doigt sur l'écran. Image issue de [102].	104
6.5	a) Technique de sélection Drag and pop : toutes les icônes potentielles sont ramenées près du curseur, b) Technique de sélection Drag and pick : toutes les icônes situées dans la direction du déplacement de la souris sont ramenées près du curseur. Images issues de [12].	105
6.6	Grille de curseurs affichés à l'écran avec a) Technique de sélection Rake Cursor : sélection du curseur actif grâce au regard (cercle rouge). Image issue de [21]. b) Technique de sélection Ninja Cursor : sélection du curseur comme celui le plus proche du centre d'une cible. Les autres curseurs sont mis en attente. Image issue de [68].	106
6.7	Partitionnement de 4 clusters (34 POIs) à l'aide de a) Voronoi (b) Starbust. Image issue de [15].	107
6.8	Technique de sélection Bubble Cursor : agrandissement circulaire du curseur de sorte qu'une seule cible ne soit sélectionnable à la fois. Image issue de [44].	109
6.9	Technique de sélection Fan Cursor : la vitesse et la direction du mouvement du curseur modifie la taille et l'orientation de l'éventail. Image issue de [103].	109
6.10	(a) Technique de sélection Dual-Finger Midpoint RayCasting : un curseur est affiché au milieu des deux doigts posés sur l'écran. (b) Technique de sélection Dual-Finger Offset RayCasting : le curseur est affiché au-dessus du doigt posé sur l'écran. Image issue de [104].	111
6.11	Technique d'interaction gestuelle : le doigt est reconnu par la caméra, et le contact du doigt avec la surface est désigné par un cercle. Image issue de [7].	111
6.12	Technique de sélection DrillSample : (a) l'utilisateur souhaite sélectionner une cible cachée (le dernier bloc rose). (b) DrillSample affiche une vue "décomposée" pour faciliter la sélection de la cible. Image issue de [78].	112
6.13	Techniques de sélection en réalité augmentée : (a) curseur centré sur l'écran ; (b) Pointage relatif avec curseur stabilisé sur l'objet physique. (c) Pointage direct sur la vidéo ; (d) Shift & Freeze : Shift [109] combiné avec arrêt sur image. Image issue de [108].	113
6.14	Mise en contexte de l'interaction : (a) Saut de menu sur le curseur (b) Curseurs multiples (c) Saut de curseur (d) «Champ de force» : aide au déplacement du curseur.	115

7.1	Ouverture d'un POI et sélection d'un élément dans le menu correspondant dans le cas du pointage indirect (a) Situation initiale, l'utilisateur se trouve devant une machine augmentée de plusieurs POIs. (b) Survol d'un POI en alignant le POI avec la croix centrale (c) Sélection du POI en appuyant sur un bouton de validation, le menu correspondant s'ouvre. (d) Survol d'un élément du menu en alignant l'élément avec la croix centrale. (e) Sélection de l'élément du menu en appuyant sur un bouton de validation.	121
7.2	Techniques d'expansion de cibles en RA mobile : sélection de POIs.	122
7.3	Diagramme de Voronoï : partitionnement de l'espace en cellules tel que (1) chaque cellule ne contient qu'une seule cible, et (2) tous les pixels d'une cellule sont plus proches du point correspondant que des autres points. Image issue de [100].	123
7.4	Deux types d'aide visuelle de la technique d'expansion de cibles a) CELL PAINTING : mise en évidence de la cellule de Voronoï dans laquelle se trouve le curseur (b) TARGET : mise en évidence de la cible de la cellule de Voronoï dans laquelle se trouve le curseur.	125
7.5	États d'un POI : diagramme de transition d'états pour le cas du pointage indirect avec un curseur centré. Les transitions sont annotées par les événements utilisateur qui les déclenchent et les actions système résultantes (ou sorties graphiques produites).	126
7.6	Pointage à l'intérieur de la cellule Voronoï d'un POI. (a) Technique de pointage tactile : détection de la proximité du doigt au-dessus de la surface d'affichage. (b) Technique de curseur centré sur l'écran. Les cellules de Voronoï sont ici affichées à des fins d'explication.	126
7.7	États d'un POI : diagramme de transition d'états pour le cas du pointage direct avec détection de proximité. Les transitions sont annotées par les événements utilisateur qui les déclenchent et les actions système résultantes (ou sorties graphiques produites).	127
7.8	Ouverture d'un POI grâce à une technique d'expansion de cibles (a) Situation initiale, un POI se trouve proche du curseur (dans la cellule de Voronoï du POI). Celui-ci est mis en évidence. (b) Sélection du POI en appuyant sur le bouton, le menu correspondant s'ouvre.	128
7.9	Démonstration des techniques. (a) Pointage physique (b) Sélection d'un POI en appuyant sur le bouton avec le pouce droit, le menu correspondant s'ouvre (c) Pointage physique pour déplacer le curseur fixe et centré sur l'écran vers l'élément du menu souhaité (d) Sélection d'un élément du menu en appuyant sur le bouton avec le pouce droit. (e) Saut du curseur sur le menu en appuyant sur l'écran avec le pouce gauche. Pouce gauche maintenu à l'écran (f1) Pointage physique pour déplacer le curseur sur l'élément de menu (f2) Pointage relatif pour déplacer le curseur sur le menu en faisant glisser le pouce gauche sur l'écran (g) Sélection d'un élément du menu en appuyant sur le bouton avec le pouce droit tout en maintenant le pouce gauche sur l'écran.	129

7.10	Stratégies pour la sélection d'un élément dans un menu : (a) Saut de menu vers le curseur (b) Curseurs multiples (c) Saut de curseur (d) «Champ de force» : aide au déplacement du curseur.	129
7.11	États d'un POI et du menu associé : diagramme de transition d'états pour le cas de l'interaction bi-manuelle VJP et VJR. Les transitions sont annotées par les événements utilisateur qui les déclenchent et les actions système résultantes (ou sorties graphiques produites).	131
8.1	(a) Un participant devant le modèle d'une machine de production. (b) Tâche de pointage pour sélectionner/ouvrir un POI. (c) Tâche de pointage pour sélectionner le deuxième élément d'un menu.	136
8.2	Configuration des POIs en cercle et des cellules de Voronoï correspondantes.	136
8.3	Champ de vision de la caméra (rectangle bleu), partition de Voronoï (en gris), et POI à sélectionner (en vert) (a) 6 POIs du cercle interne sont visibles. (b) Déplacement de l'appareil mobile pour sélectionner le POI vert : le POI opposé sur le cercle devient hors-champ.	137
8.4	Déroulement de l'expérience.	138
8.5	Temps (s) de sélection/d'ouverture d'un POI pour toutes les techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.	140
8.6	Temps (s) de sélection d'un élément dans un menu pour toutes les techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.	140
8.7	Temps (s) total de complétion d'une tâche de pointage (T1 :sélection d'un POI + T2 :sélection d'un élément dans un menu) pour toutes les techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.	140
8.8	Fatigue physique sur 5 (1 = très mauvais, 5 = très fatigant) ressentie par les participants après utilisation des techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.	140
8.9	Notes sur 5 (1 = très mauvais, 5 = très bien) données aux techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.	141
8.10	(a) POI ouvert dans la section inférieure de l'écran (rectangle bleu) : courte distance entre le curseur et l'élément vert du menu. (b) POI ouvert dans la section supérieure de l'écran (rectangle bleu) : importante distance entre le curseur et l'élément vert du menu.	143
8.11	Calques de réalité augmentée : monde physique - POIs - menus.	145
9.1	Adaptation automatique des techniques de visualisation hors-champ en fonction de la disposition spatiale des cibles hors-champ. (a) Halo3D convient lorsque les cibles sont condensées autour d'une position. (b) Halo3D** s'adapte mieux lorsque les cibles d'un paquet s'étalent dans l'espace hors-champ. (c) Halo3D* est la technique la plus appropriée lorsque les cibles sont très dispersées dans l'espace hors-champ.	151

9.2	Adaptation automatique des techniques de visualisation hors-champ en fonction du nombre de chevauchements visibles entre les cercles à l'écran.	
	(a) Halo3D* ne convient pas lorsqu'un nombre important de chevauchements est présent à l'écran. Dans ce cas, deux techniques sont plus adaptées : Halo3D (b) ou Halo3D** (c).	152
9.3	Pointage tactile à l'intérieur d'une cellule Voronoï d'une cible : détection de la proximité du doigt au-dessus de la surface d'affichage.	152

1

Sommaire

1.1	Domaine	18
1.2	Questions de recherche	20
1.3	Contributions	21
1.4	Collaboration industrielle : domaine applicatif	22
1.5	Structure du manuscrit	23

1.1 Domaine

Cette thèse s’inscrit dans le domaine de l’Interaction Homme-Machine (IHM) et est dédiée à l’interaction en Réalité Augmentée (RA) sur dispositifs mobiles.

La RA permet de superposer du contenu numérique à l’environnement physique de l’utilisateur. Caudell et Mizell ont introduit pour la première fois le terme ”réalité augmentée” en 1992 pour désigner un système qui affiche des informations numériques enregistrées dans le monde physique [32]. Milgram et al. ont proposé en 1994 une taxonomie [76, 77]

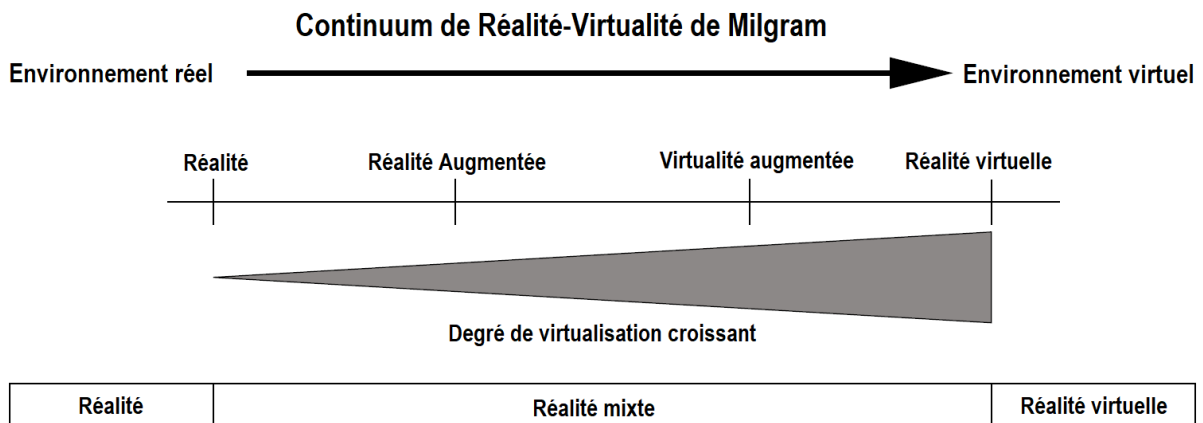


FIGURE 1.1 – Continuum Réalité-Virtualité de Milgram pour décrire différentes combinaisons d’éléments réels et virtuels. Image inspirée de [38].

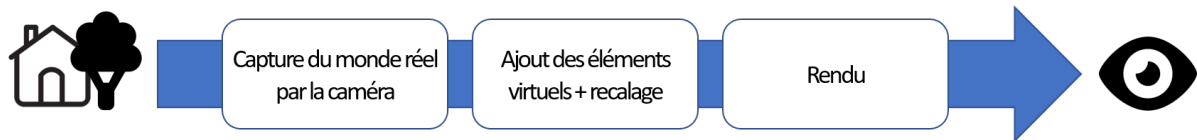


FIGURE 1.2 – Processus de fonctionnement de la réalité augmentée.

qui situe la réalité augmentée dans un continuum de la réalité-virtualité. Ce continuum est une échelle continue qui englobe toutes les compositions possibles d'objets réels et virtuels (voir Figure 1.1). Ainsi, la réalité augmentée se caractérise par l'observation simultanée de l'environnement réel et d'objets virtuels superposés, tandis que dans la réalité virtuelle, l'utilisateur est totalement immergé dans un monde 3D généré par ordinateur. Enfin, Azuma et al. [5, 6] définissent un système de RA par un système qui :

- Combine réel et virtuel
- Est interactif et en temps réel
- Est synchronisé en 3D (aligne et fait correspondre l'orientation des éléments virtuels avec celui des objets physiques)

Les informations ajoutées au monde physique peuvent être graphiques [6], auditives [40] ou basées sur du retour haptique [11, 86]. Dans nos travaux, nous nous focalisons sur la RA qui repose sur l'ajout d'éléments numériques graphiques ancrés dans le monde physique pour définir un environnement mixte. En RA, la superposition d'éléments numériques graphiques sur un environnement physique peut être réalisée sur l'écran d'un dispositif mobile (RA mobile avec un téléphone, une tablette etc.), sur l'écran d'un casque de RA (en anglais Head-Mounted Display - HMD) ou encore en projetant directement le contenu virtuel sur l'environnement physique, aussi noté RA spatiale). Quelle que soit la méthode employée, l'environnement mixte ainsi obtenu comprend des objets numériques ancrés dans le monde physique.

La RA connaît une forte expansion dans de nombreux secteurs : le divertissement (jeux vidéos), l'éducation (apprentissage interactif), et les industries (aide à la conception, assemblage, maintenance etc.). Ce phénomène émergent a notamment été possible grâce à la démocratisation des téléphones et tablettes mobiles. La majeure partie de ces appareils contiennent la technologie permettant des applications de RA (comme une caméra et une centrale inertielle). Ainsi, la RA mobile est un cas spécifique où un dispositif mobile équipé d'une caméra et tenu entre les mains est utilisé pour afficher l'environnement mixte (voir Figure 1.2). Les éléments numériques générés par l'ordinateur sont ainsi directement superposés aux images de la caméra sur l'écran du téléphone ou de la tablette. Les téléphones et tablettes étant largement répandus, la RA mobile est facilement accessible aux utilisateurs. C'est pourquoi nos travaux se concentrent sur les systèmes de RA mobiles.

1.2 Questions de recherche

Avec la multiplication des smartphones et de leur capacité, la RA mobile s'installe progressivement auprès du grand public et des industries. Au delà de l'effet de découverte et du divertissement que peut provoquer la RA, il est nécessaire d'envisager une utilisation réelle, et d'aller plus loin que l'affichage d'informations à l'écran. Le jeu *Pokemon Go* est un exemple d'application de RA où l'interaction avec le contenu numérique de RA reste très limitée : l'application ne présente pas d'interaction avec l'environnement augmenté/mixte autre que la récupération des données (pokémons). Bien que la technologie soit disponible sur de nombreux dispositifs mobiles commerciaux, l'enjeu de l'interaction est une étape incontournable pour le développement d'applications de RA mobile qui soient utiles (permettant la réalisation de nombreuses tâches de complexité variée) et utilisables.

L'interaction en RA mobile soulève de nombreux problèmes. Les problèmes sont d'abord ceux identifiés pour l'interaction sur supports mobiles : tout d'abord, la taille limitée de l'écran restreint la quantité d'information qu'il est possible d'afficher [35]. Sur écrans tactiles, l'interaction directe avec le doigt peut aussi masquer la cible souhaitée à l'écran, empêchant un pointage précis (fat finger) [88]. De plus, l'interaction directe avec le pouce de la main qui tient l'appareil rend inaccessible certaines zones de l'écran [18, 33] qui sont en dehors de la zone fonctionnelle du pouce. L'interaction avec la main qui tient l'appareil peut également devenir problématique lorsque l'appareil utilisé est plus grand, comme une tablette ou un smartphone [91]. Un appareil trop grand ou trop lourd devient ainsi suffisamment encombrant pour qu'il puisse être tenu de manière stable avec une seule main. Cette instabilité est d'autant plus perceptible en RA, car la vue temps réel de la caméra et le contenu numérique associé et ancré dans le monde physique peuvent être facilement perturbé par les tremblements naturels des mains [108]. Au delà des problèmes hérités de l'interaction sur dispositif mobile, la RA ajoute des problèmes interactionnels supplémentaires dont la vision partielle de l'environnement mixte limitée à l'angle de vue de la camera [65] et l'instabilité des cibles à l'écran due aux problèmes de suivi de mouvement et d'enregistrement du contenu numérique sur les objets physiques [107].

Dans ce contexte de besoins interactionnels aigus, nous traitons la tâche élémentaire universelle de sélection d'une cible à l'écran. Or en RA mobile, l'environnement augmenté/mixte est partiellement perceptible sur l'écran du dispositif mobile. Cela définit plusieurs contraintes pour l'interaction : (1) la cible numérique n'est pas visible dans le champ de vision de la caméra. Il est donc nécessaire d'étendre la connaissance de l'utilisateur sur l'environnement RA. (2) Une cible numérique peut contenir plusieurs informations pertinentes par rapport à sa localisation dans le monde physique. Cela implique donc de faciliter le pointage de ces informations sans briser le lien qui les unit à leur environnement physique. Nous traitons donc les deux questions suivantes :

1. L'exploration d'un environnement en RA pour rechercher des cibles numériques

nécessite l'acquisition d'informations qui dépassent le champ de vision de la caméra du dispositif mobile. Des techniques de visualisation sont donc nécessaires pour étendre la connaissance sur l'espace hors-champ. Or l'écran de l'appareil mobile a une taille limitée, et les objets hors-champ peuvent être nombreux. De plus, l'affichage d'informations supplémentaires sur l'espace hors-champ ne doit pas distraire les utilisateurs de leurs tâches principales ou masquer la zone de travail de l'utilisateur, située sur la partie centrale de l'écran.

→ **Comment améliorer la perception de l'environnement mixte RA au delà du champ de vision de la caméra du dispositif mobile, sans surcharger l'interface graphique sur un écran de taille limitée ?**

2. Une fois la cible visible dans le champ visuel de la caméra, l'utilisateur doit être en mesure de la sélectionner sur l'écran. Comme la tâche de sélection est élémentaire et universelle, de très nombreux travaux de recherche en IHM sont dédiés aux techniques de sélection en considérant les aspects matériels, logiciels et les capacités humaines. Nous étudions cette tâche élémentaire en RA. En RA, les cibles numériques regroupent des informations pertinentes par rapport à leur localisation dans le monde physique. Par exemple, dans une application de RA développée pour un musée, les cibles sont positionnées à l'emplacement de chaque œuvre. Ces cibles enrichissent l'environnement réel avec des informations telles que la biographie d'un peintre, les étapes de création d'une sculpture, les différentes couches de peinture d'une toile etc. Il est donc pas uniquement question de sélectionner une cible, mais aussi d'interagir avec les éléments numériques contenus à l'intérieur de ces cibles. Les techniques d'assistance au pointage ont pour objectif de faciliter la sélection des cibles, en minimisant le nombre d'erreurs et le temps nécessaire pour acquérir une cible numérique. Par exemple, les techniques d'expansion de cibles [55, 56] pour les interfaces graphiques facilitent le pointage en agrandissant la taille des cibles. Mais ces techniques sont-elles applicables en RA mobile pour augmenter l'efficacité du pointage des cibles et des informations qui y sont contenue ?

→ **Comment améliorer le pointage sur dispositif mobile d'une cible numérique ancrée dans l'environnement physique, et des informations qui y sont contenues ?**

Nous apportons des contributions à ces deux questions.

1.3 Contributions

En réponse à nos deux questions de recherche, nos contributions sont doubles :

Nous proposons dans un premier temps trois techniques interactives de guidage hors-champ, que nous évaluons au travers de deux études expérimentales en laboratoire. Ces techniques sont basées sur Halo [13], une solution permettant la visualisation d'objets hors-champ à l'aide de cercles, et diffèrent par la manière de visualiser l'agrégation des cibles

numériques hors-champ. Les résultats de deux études expérimentales montrent que nos trois techniques étendent efficacement la connaissance de l'utilisateur sur l'environnement mixte hors-champ, et limitent l'intrusion visuelle à l'écran du dispositif mobile par rapport aux solutions traditionnelles à base de flèches.

Nous avançons ensuite deux techniques d'interaction pour faciliter le pointage de cibles numériques en RA. Ces techniques sont basées sur (1) les techniques d'expansion de cibles, qui facilitent la sélection de cibles en allouant à chaque cible une zone active plus grande et (2) un saut de curseur, qui raccourcit la distance entre le curseur et les informations regroupées au sein de la cible numérique. Les deux techniques diffèrent par la manière de manipuler le curseur une fois qu'il a sauté sur la cible numérique. Nous offrons deux possibilités : pointage physique en déplaçant l'appareil mobile vers l'information souhaitée, ou pointage relative en utilisant le pouce sur l'écran. Le résultat d'une expérience en laboratoire confirme que les techniques d'expansion de cibles diminuent le temps nécessaire à l'acquisition d'une cible, tandis que la manipulation relative du curseur avec le pouce est très largement préférée par les participants.

1.4 Collaboration industrielle : domaine applicatif

Cette thèse CIFRE a été réalisée en collaboration avec Schneider Electric. Le domaine applicatif cible est la maintenance des machines dans les usines de production.

Nous explorons dans cette thèse l'interaction en réalité augmentée mobile en contexte industriel. Avec l'avènement d'appareils mobiles de plus en plus puissants, la RA est de plus en plus utilisée pour la maintenance industrielle, dans le contexte de "l'industrie 4.0" [22, 84, 97]. En particulier, nous nous concentrons sur les systèmes de RA mobiles pour les opérateurs de maintenance des usines de production. Ces systèmes permettent aux opérateurs d'obtenir des informations supplémentaires et en temps réel sur une machine. Ainsi, un utilisateur se trouvant face à une machine de production (voir Figure 1.3) pourra consulter les données statistiques de la machine (vitesse de production etc.) , des vidéos d'entraînement sur les procédures de maintenance étape par étape ou encore obtenir des informations sur les endroits de la machine qui nécessitent une révision. L'augmentation est donc fournie sur la base des connaissances disponibles sur l'emplacement de l'opérateur et des informations fournies par l'automate de la machine de production.

Nos travaux partent de ce contexte applicatif mais nos contributions sont plus générales. En effet, la collaboration avec Schneider Electric a permis d'instancier une approche itérative centrée sur les besoins des utilisateurs. Nous avons adopté une démarche de recherche qui partait des besoins utilisateurs. A partir de ces besoins utilisateurs nous avons défini des besoins génériques auxquels nous avons cherché à répondre. Les évaluations expérimentales ont été conduites en laboratoire ou contexte pseudo-industriel en contrôlant les variables indépendantes. En cela les contributions sont générales mais issues d'un besoin utilisateur dans un contexte applicatif particulier.

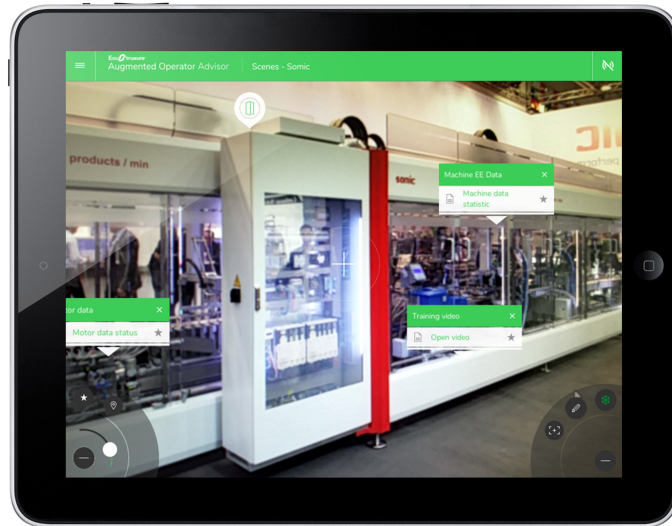


FIGURE 1.3 – Application de réalité augmentée pour la maintenance de machine de production développée par Schneider Electric. L’application permet d’examiner l’état d’un moteur, les statistiques de la machine, ou encore de consulter une vidéo d’entraînement. Image issue de [37].

Nos contributions sont aussi valides dans le domaine applicatif de la maintenance des machines de production. Pour cela, nous avons complété les évaluations contrôlées par la collecte des retours d’experts dans le domaine de la production industrielle, parfois en contexte. Enfin, les techniques d’assistance au pointage en RA mobile présentées dans cette thèse ont été intégrées dans les systèmes professionnels de Schneider.

Cette collaboration industrielle nous semble particulièrement effective. En effet, elle a permis de mener des travaux de recherche fondamentaux en IHM qui vont au-delà d’un domaine applicatif particulier, tout en donnant lieu à un transfert industriel effectif.

1.5 Structure du manuscrit

L’exploration d’un environnement RA conduit les utilisateurs à (1) chercher des objets numériques situés autour d’eux, et (2) à interagir avec ces éléments. Ce manuscrit est donc structuré en deux parties. La première partie détaille les techniques de visualisation de l’espace hors-champ, visant à étendre la connaissance de l’utilisateur sur l’environnement RA. La deuxième partie traite des techniques d’assistance au pointage, dont le but est de faciliter la sélection de cibles numériques RA et des informations qui y sont contenues.

Chaque partie est structurée de la même manière avec un chapitre qui dresse l’état de l’art en relation avec la question de recherche, un chapitre dédié à la conception de techniques et un chapitre décrivant les évaluations expérimentales conduites. En plus de ces deux parties, nous commençons le manuscrit par un chapitre qui présente le contexte industriel cible et les scénarios afin de poser les besoins utilisateurs sur lesquels reposent nos travaux. Ainsi, le manuscrit contient les chapitres suivants :

- Dans le chapitre 2, nous présentons le contexte industriel auquel sont rattachées nos recherches, les scénarios d'usage en RA, ainsi que l'environnement matériel et logiciel de RA utilisée dans nos études expérimentales.

Partie 1 : Techniques de visualisation de l'espace hors-champ

- Dans le chapitre 3, nous dressons un état de l'art des techniques de visualisation de l'espace hors-champ.
- Dans le chapitre 4, nous introduisons trois techniques de visualisation hors-champ pour minimiser l'intrusion visuelle à l'écran, gérer les environnement denses en cibles numériques et visualiser la répartition spatiale des objets hors-champ.
- Dans le chapitre 5, nous présentons deux études expérimentales pour évaluer l'apport de nos trois techniques de visualisation hors-champ sur la connaissance des utilisateurs de l'environnement hors-champ.

Partie 2 : Techniques de pointage en RA

- Dans le chapitre 6, face aux très nombreux travaux en IHM sur le pointage, nous dressons un état de l'art des techniques d'assistance de pointage, en focalisant sur les techniques développées sur appareils mobiles et en RA.
- Dans le chapitre 7, nous introduisons deux techniques d'assistance au pointage en RA facilitant la sélection des cibles et des informations qui y sont contenues.
- Dans le chapitre 8, nous présentons une étude expérimentale pour évaluer nos deux techniques de pointage RA dans un contexte pseudo-industriel. Cette tâche est composée de la sélection d'une cible et d'un élément du menu contenu au sein de cette cible.

Enfin, nous concluons ce manuscrit en résumant nos contributions et en présentant les pistes de recherche pour les techniques de visualisation hors-champ et d'assistance au pointage en RA.

2

Contexte et application cible : maintenance industrielle

Sommaire

2.1	Réalité augmentée pour la maintenance	25
2.2	Tâches de maintenance	26
2.3	Trois scénarios d’usage de la réalité augmentée	26
2.4	Application cible existante	30
2.4.1	Architecture globale	30
2.4.2	Interface utilisateur	30
2.4.3	Système de suivi	31
2.5	Conclusion	33

Le contexte applicatif de nos travaux de recherche est la maintenance industrielle de machines de production. Nos travaux sont dédiés aux techniques d’interaction pour une application sur supports mobiles de réalité augmentée destinée aux opérateurs de maintenance. Dans ce chapitre, nous décrivons ce contexte applicatif et nous introduisons les utilisateurs et leurs tâches. Nous introduisons également l’application existante.

2.1 Réalité augmentée pour la maintenance

La Réalité Augmentée (RA) est une technologie qui combine éléments réels et éléments numériques en temps réel et sur un même support (téléphones, tablettes, casque de RA, etc.).

En contexte industriel, la RA permet notamment d’optimiser la maintenance des machines de production et vise trois objectifs :

1. **Réduire le temps de maintenance** : sans RA, les opérateurs de maintenance sont tenus de se diriger vers le poste de contrôle de la machine, écran tactile fixe permettant de vérifier son état courant et de lancer des opérations. Pour consulter les valeurs de contrôle (température, pression, etc.) d’un élément de la machine,

l'opérateur est donc obligé de retourner vers le poste de contrôle de la machine (mobilité physique). Sur le poste fixe, l'interface graphique permet à l'opérateur d'accéder aux valeurs de contrôle qu'il souhaite consulter (mobilité logique). La RA permet d'unifier la mobilité logique et physique, et de consulter les valeurs de contrôle là où se trouve l'élément contrôlé.

2. **S'abstraire du niveau d'expertise de l'utilisateur** : la RA permet de guider un opérateur novice à travers la procédure de maintenance qu'il doit réaliser. Les instructions affichées à l'écran lui indiquent pas à pas les étapes de la maintenance, pour réaliser la tâche en toute sécurité.
3. **S'abstraire du niveau d'habilitation de l'utilisateur** : la RA donne également la possibilité à un utilisateur de pouvoir consulter les valeurs de contrôle se trouvant à l'intérieur d'une machine dont l'accès est interdit. A titre d'exemple, certaines armoires électriques ne peuvent être ouvertes qu'avec une habilitation particulière, pour des raisons de sécurité. Grâce à la RA, un opérateur n'a pas besoin d'ouvrir la machine et peut consulter les valeurs de contrôle depuis l'extérieur. Sur la Figure 2.5, le schéma électrique est superposé par dessus la machine fermée.

Globalement, ces trois objectifs permettent la diminution du temps d'arrêt d'une machine de production, et limite donc l'impact négatif sur son rendement journalier.

2.2 Tâches de maintenance

Les opérateurs de maintenance assurent l'entretien des équipements de production. Leur objectif est de garantir le bon fonctionnement des machines pour qu'elles puissent fournir un produit de qualité. Ils effectuent des contrôles périodiques pour certifier que les équipements fonctionnent à pleine capacité. Ainsi, leur rôle est de :

- Surveiller régulièrement les équipements et machines d'un site industriel
- Réaliser des maintenances préventives, pour anticiper tout problème
- Réaliser des maintenances correctives, en cas de défaillances existantes
- Contrôler le fonctionnement d'un équipement après une intervention
- Documenter toute maintenance effectuée pour la traçabilité

2.3 Trois scénarios d'usage de la réalité augmentée

Nous décrivons trois scénarios dans le domaine de la maintenance industrielle. Ces cas d'usage sont représentatifs d'un grand nombre de scénarios de RA. Elles mettent en évidence des suites de tâches interactives différentes. Nous illustrons ces scénarios avec une chaîne de fabrication de cartes électroniques. Ces lignes d'assemblage comportent, entre autres, un four à soudure, et des chariots sur lesquels sont placées les cartes électroniques.

Un POI permet d'accéder à plusieurs types d'informations techniques liées à un emplace-



FIGURE 2.1 – Accès aux données d'aide à la maintenance d'une armoire électrique, regroupées sous forme de POIs (a) 3 POIs fermés placés sur une armoire électrique. (b) 2 des 3 POIs ouverts, affichant des informations sur la température et la pression de la machine, ainsi que la possibilité d'accéder à deux documentations techniques et à une vidéo explicative. (c) Lancement de la vidéo explicative sur le montage et le démontage d'un composant de la machine (image issue de [80]).

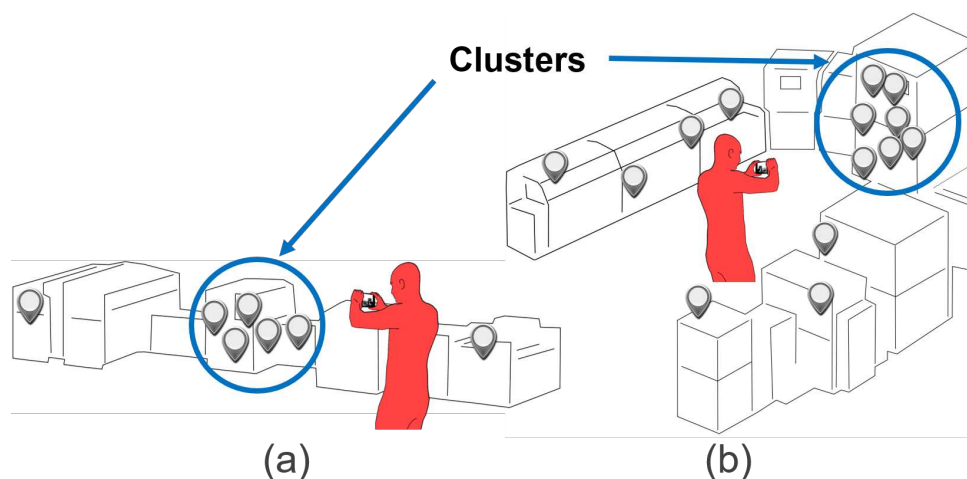


FIGURE 2.2 – (a) Opérateur de maintenance devant une chaîne cuboïde de montage. Les points d'intérêt de maintenance RA sont situés sur toute la longueur de la ligne. (b) Opérateur de maintenance au centre d'une ligne d'équipements en forme de U. Les points d'intérêts (POIs) de maintenance RA sont situés autour de l'utilisateur.

ment physique spécifique de la machine. Une fois qu'un POI cible apparaît dans le champ de vision de la caméra de l'appareil mobile (voir Figure 2.1a), les opérateurs de maintenance peuvent accéder aux informations de la machine que le POI regroupe (température, pression, cadence de production actuelle de la machine, etc.) (voir Figure 2.1b), et ouvrir des documentations ou des vidéos d'entraînement (voir Figure 2.1c).

Scénario 1 : Avec des lignes d'assemblage en forme de cuboïde (voir Figure 2.2a) les opérateurs sont toujours face à la machine. Les POIs sont par conséquent situés devant les opérateurs. Lors des campagnes de maintenance, les opérateurs se présentent devant la chaîne de montage pour effectuer une tâche unique répétée X fois à différentes localisations. Chaque emplacement est marqué par un POI (voir Figure 2.3) et contient des informations sur la machine (statut, température du four, panne, accès au diagramme électrique, comme

montré à la Figure 2.3d et Figure 2.3e) etc.).

→ La machine étant un cuboïde, les opérateurs de maintenance tournent autour d'elle pour chercher les points d'intérêts et examiner l'état de la ligne de production.

Scénario 2 : Certaines lignes d'assemblage sont définies par un ensemble de machines formant un "U" (voir Figure 2.2b). Les opérateurs de maintenance se déplacent au sein du "U" pour examiner l'état de la ligne et/ou y faire des réparations. → Comme la ligne est en forme de U, les opérateurs de maintenance effectuent une rotation autour d'eux-mêmes pour examiner les POIs, et donc l'état de la machine.

Les scénarios ci-dessus peuvent impliquer un nombre important de POIs (environnements denses). De plus, les POIs peuvent être répartis de manière diffuse sur la machine, ou de manière concentrée. On parle alors de clusters de POIs (Figure 2.2). Par exemple, la commande de la machine de revêtement UV est un petit appareil (60cm x 2m) qui rassemble toutes les informations sur l'état de la machine. Cet espace limité peut contenir jusqu'à 6 ou 7 points d'intérêt.

Les scénarios 1 et 2 diffèrent par la forme des machines et ont un impact direct sur la réalisation de la tâche par les opérateurs : le premier scénario concerne des machines en forme de U. Dans ce cas, les opérateurs de maintenance doivent effectuer une rotation sur place pour rechercher les points d'intérêt qui les entourent. Le deuxième scénario inclut des machines cuboïdales : les opérateurs tournent autour de la machine pour amener les points d'intérêt dans le champ de vision de la caméra.

Scénario 3 : Pour visualiser les informations regroupées dans un POI, les opérateurs sélectionnent le POI (voir Figure 2.3b et Figure 2.3c). Lorsqu'un POI est sélectionné, la liste des informations est affichée sous forme de menu (voir Figure 2.3d et Figure 2.3e) et l'opérateur peut sélectionner un des items. L'état d'une machine dépend de plusieurs variables, qui peuvent être documentées dans des POIs. L'ouverture de plusieurs POIs (voir Figure 2.1b) est souvent nécessaire pour avoir une vision complète de l'état de la machine. Les opérateurs de maintenance doivent donc très souvent (1) sélectionner plusieurs POIs pour acquérir des données en temps réel, (2) lorsqu'un POI est ouvert, sélectionner l'information technique souhaitée (voir Figure 2.1 et Figure 2.3), et (3) fermer les POIs une fois la tâche de maintenance terminée.

Pointer des objets numériques est une tâche interactive élémentaire universellement présente en RA mobile. Le pointage est généralement effectué soit avec un curseur placé au centre de l'écran [93] [96] (voir Figure 2.3 b-d), soit en touchant directement la cible sur l'écran (avec un stylo ou des doigts nus) [89] [96] (voir Figure 2.3 c-e). Dans les deux cas, les techniques de pointage sont altérées par l'instabilité de la scène augmentée avec laquelle les utilisateurs interagissent à l'écran. En effet, différents facteurs tels que définis dans [107] peuvent nuire à la stabilité de la scène augmentée comme :

— Le tremblement naturel de la main entraîne de petits changements dans le point

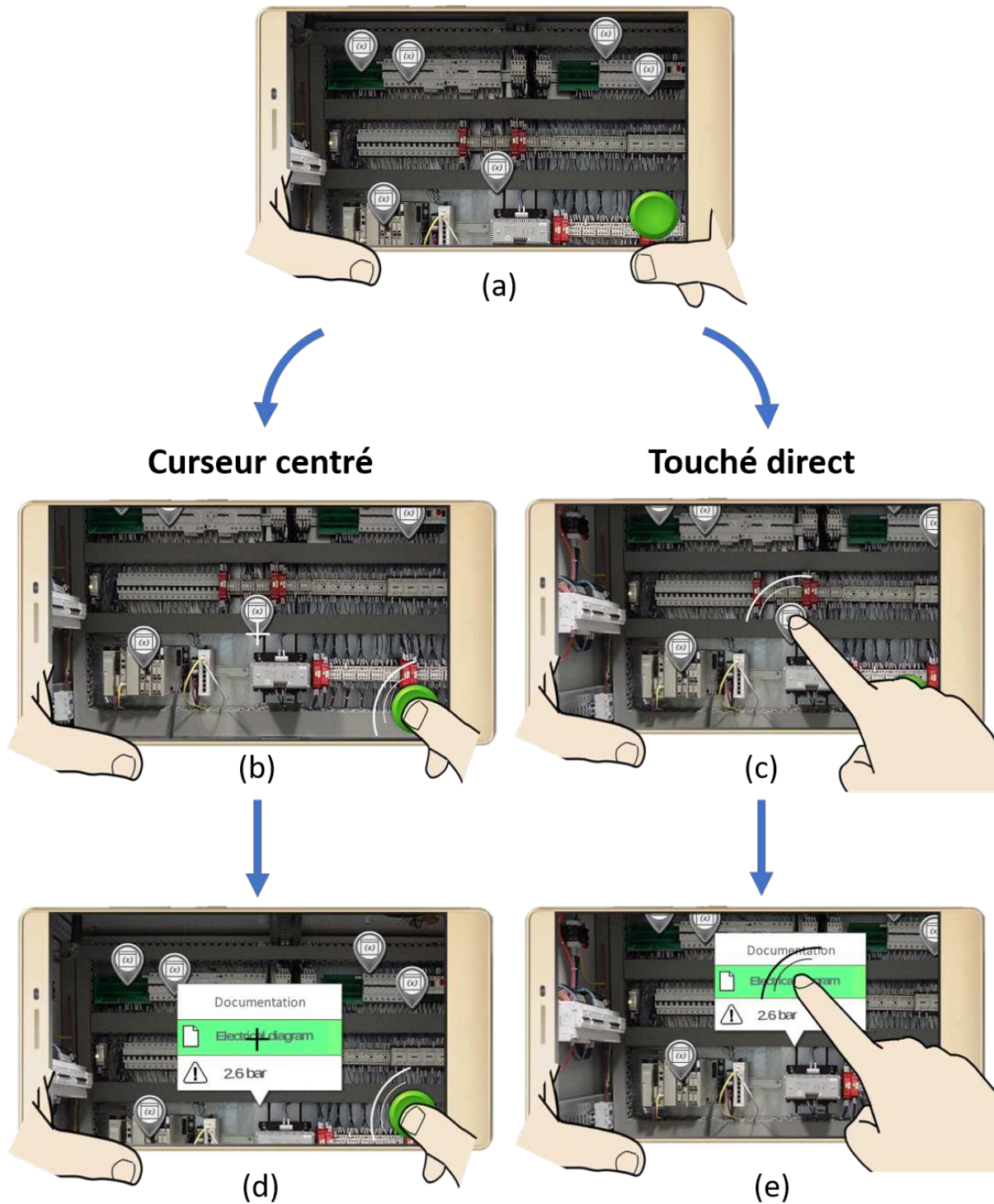


FIGURE 2.3 – Exemple de sélection d’une information technique sur une machine à l’aide (b)-(d) du curseur centré sur l’écran et (c)-(e) d’un pointage direct avec le doigt. (a) 6 POIs placés sur une armoire électrique. (b) Sélection d’un POI en l’alignant avec le curseur central et en appuyant sur le bouton valider. (c) Sélection d’un POI en le touchant directement avec le doigt. (d) Ouverture d’un menu à 3 items et sélection du deuxième élément en alignant le curseur avec l’élément et en appuyant sur le bouton valider. (e) Ouverture d’un menu à 3 items et sélection du deuxième élément en le touchant directement avec le doigt.

de vue de la caméra rendant le contenu à l'écran instable.

- Le fait de tenir l'appareil d'une main tout en interagissant avec l'autre main peut entraîner une instabilité impactant la précision d'une sélection avec le doigt [107].

De ce fait, les deux mains sont généralement utilisées par les opérateurs de maintenance pour maintenir l'appareil mobile (voir Figure 2.3a), apporter plus de stabilité à la scène, et évaluer avec précision l'état de la machine de production.

2.4 Application cible existante

Pour mieux cerner le contexte des travaux de recherche entrepris, nous décrivons l'application d'aide à la maintenance en RA conçue et développée par Schneider Electric.

2.4.1 Architecture globale

L'application d'aide à la maintenance se compose de :

1. La machine de production à augmenter
2. Une base de données dans laquelle sont stockées toutes les données récoltées de la machine (température, pression, etc.)
3. Un serveur de RA, intermédiaire entre l'application de RA et la base de données. Celui-ci est installé sur l'ordinateur de contrôle de la machine. L'écran de cet ordinateur est fixé sur la machine de production et permet de consulter les valeurs de contrôle de l'équipement et de lancer des opérations. Ce serveur collecte les valeurs de contrôle disponibles de la machine, et les transmet à la base de données pour les stocker.
4. Une application de RA sur un dispositif mobile qui manipule les valeurs de contrôle de la machine en communiquant avec le serveur RA de la machine.

Lorsqu'un opérateur souhaite accéder depuis son dispositif mobile à une information sur l'état de la machine, l'application de RA dialogue avec le serveur de RA pour récupérer les données en temps réel. Ce dernier les récupère depuis la base de données et les renvoie à l'application (voir Figure 2.4).

2.4.2 Interface utilisateur

L'interface graphique de l'application affiche des POIs à des endroits spécifiques de la machine (voir Figure 2.5). Chaque POI regroupe une ou plusieurs informations utiles pour la maintenance de l'équipement. Dans le cas où un POI regroupe plusieurs informations, un menu s'ouvre au-dessus de celui-ci pour permettre à l'utilisateur de choisir l'item de son choix. Cette interface graphique a servi de base pour toutes les études expérimentales décrites dans ce mémoire de thèse. Il convient néanmoins de noter qu'il ne s'agit que d'une partie de l'interface. En effet, l'application permet aussi de lancer des *procédures*.

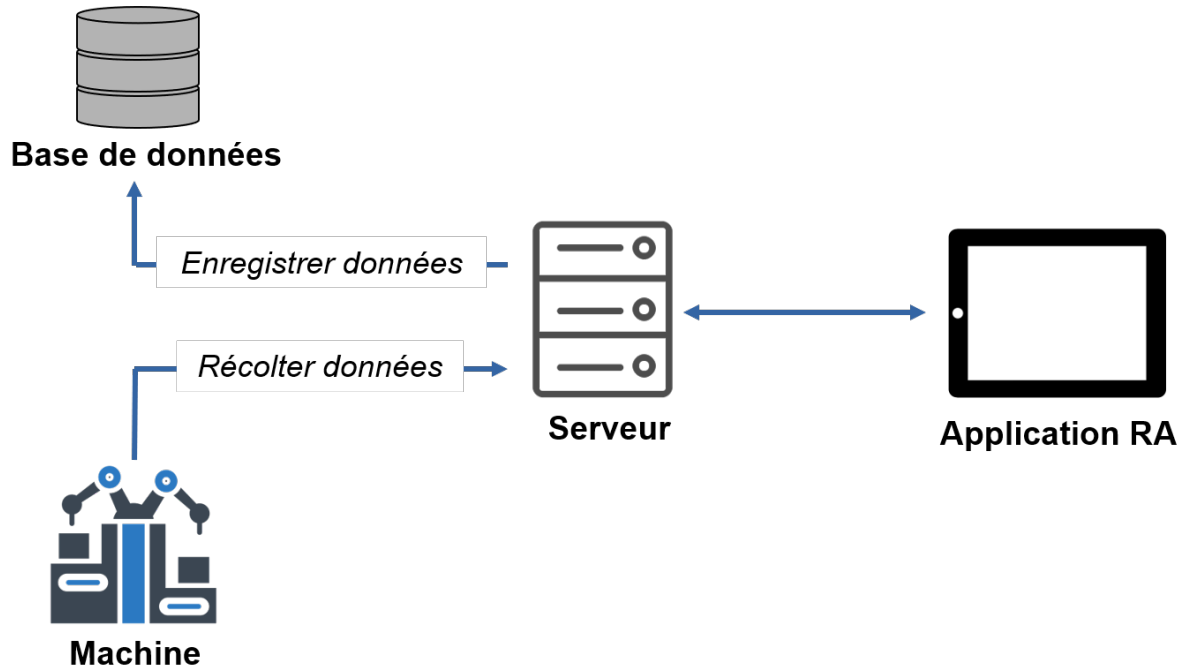


FIGURE 2.4 – Architecture de l’application RA d’aide à la maintenance.

Les procédures sont constituées d’un ensemble d’étapes que le technicien doit suivre pour effectuer une tâche de maintenance. L’opérateur est ainsi guidé pas à pas dans l’accomplissement de sa tâche. Ce cas diffère des trois scénarios présentés en section 2.3 : dans ces scénarios, l’utilisateur souhaite avoir une vision globale de l’état de la machine et de l’ensemble des POIs qui s’y trouvent. Ici, l’opérateur ne souhaite connaître que la position d’un seul POI, celui où l’étape courante de la procédure doit être effectuée. L’opérateur n’a donc qu’une seule destination affichée à l’écran à la fois, et se rendra à la prochaine destination une fois que l’étape courante de la procédure sera terminée.

2.4.3 Système de suivi

Le système de suivi a une importance particulière dans le cadre de la RA : il assure que les informations numériques soient superposées en temps réel aux éléments du monde physique correspondants, et ce quelque soit la position et l’orientation de l’utilisateur (aussi désigné comme l’alignement numérique-physique). L’objet des travaux de recherche de cette thèse n’est pas le système de suivi, et les travaux présentés dans cette thèse sont indépendants du système de suivi utilisé. Nous détaillons cependant la méthode de suivi employée lors de toutes les études expérimentales décrites dans ce manuscrit. Nous présentons ainsi *Tango*, méthode utilisée pour assurer ce suivi.

Tango est une plateforme logicielle et matérielle de RA développée par Google en 2014. Elle est destinée à équiper les appareils mobiles des technologies nécessaires à la RA. Les téléphones/tablettes sont dotés de capteurs proches d’une Kinect afin de percevoir leur environnement en profondeur et de cartographier une pièce en 3D. Tango fournit une parfaite connaissance de l’espace grâce à une combinaison de suivi de mouvements (via une centrale inertielle) et de perception de la profondeur (via un capteur de profondeur).

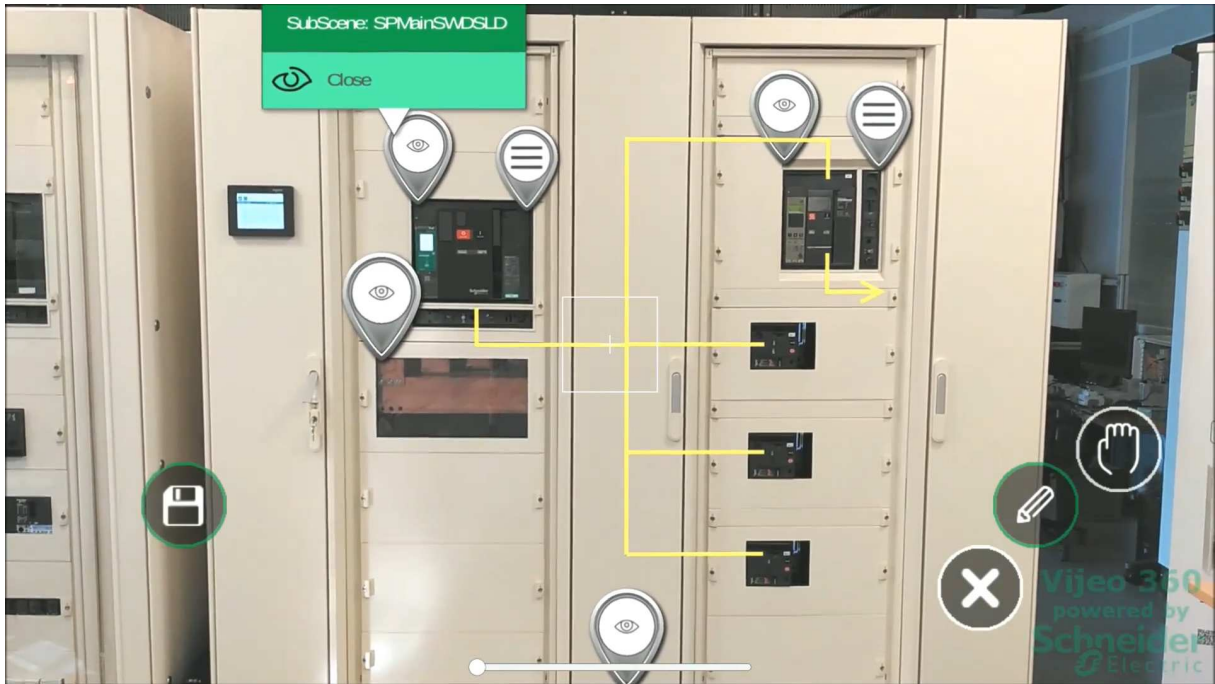


FIGURE 2.5 – Interface de l’application de RA *Vijeo* : schéma électrique superposé à la machine et reliant plusieurs POIs

En l’absence de ces capteurs, d’autres technologies de suivi nécessitent des marqueurs visuels (QR code, images spécifiques de détection, etc.) pour déterminer la position de la caméra au sein de l’environnement 3D. Un marqueur visuel sert de référence pour construire un repère 3D (0,0,0) à partir duquel sera calculée la position de la caméra.

Roberto et. al [92] ont étudié la précision du suivi de mouvements et de la perception de profondeur d’une tablette Google Tango. Pour cela, le déplacement de la tablette a été évalué entre deux positions connues dans le monde réel. Deux études expérimentales ont été menées, l’une pour évaluer un petit espace de travail RA (table d’une surface de un mètre carré) et l’autre pour de grands environnements (bureau fermé d’une superficie d’environ 50 mètres carrés). Les résultats ont montré que le suivi des mouvements de la tablette Tango est 2,3 fois plus précis sur un petit espace de travail que sur de grands environnements intérieurs. Les erreurs de suivi de mouvements de la tablette Tango étaient d’environ 6 et 14 centimètres respectivement pour les petits et grands scénarios intérieurs.

Alors que le suivi de Tango est précis à court terme grâce à la centrale inertielle et au capteur de profondeur, la précision à long terme se détériore rapidement en raison de l’erreur accumulée par ces mêmes capteurs [85]. Pour atténuer cette dérive, Google a mis au point une solution d’apprentissage automatique, appelée *Area Learning*, permettant à Tango d’obtenir une mémoire de l’environnement. Il est ainsi en mesure de se repositionner dans une zone déjà apprise, en retraçant sa position jusqu’à la position de départ relative (0, 0, 0) de chaque session. L’apprentissage de l’espace se fait en parcourant la région dans différentes directions. Tango mémorise des caractéristiques de l’environnement tels que des objets avec des coins et des bords. Les informations visuelles obtenues via la caméra et les capteurs de profondeur sont converties en descriptions stockées sur l’appareil. Celles-ci

seront ainsi comparées par rapport à la position actuelle du périphérique pour le localiser précisément dans la zone apprise. La solution d'apprentissage automatique *Area Learning* permet donc de corriger les erreurs de dérive des capteurs et reste précis dans le temps. Nous utilisons cette solution dans toutes les études expérimentales présentées dans ce manuscrit.

Le principal avantage de *Tango* est la possibilité pour l'utilisateur d'obtenir 6 degrés de liberté (3 axes d'orientation ainsi que 3 axes de mouvement) sans nécessité de marqueurs visuels dans le champs de vision de la caméra. Cependant, il nécessite un matériel dédié pour fonctionner, le capteur de profondeur.

La technologie *Tango* n'a été déployée que sur les appareils mobiles *Lenovo Phab 2 Pro* (sortis en Novembre 2016) et *Asus Zenfone AR* (sortis en Juillet 2017). Google a depuis arrêté le projet *Tango* le 1er mars 2018 pour se centrer sur *ARCore*, le kit de développement RA pour Android, concurrent de *ARKit* pour appareils mobiles Apple.

2.5 Conclusion

Dans ce chapitre, nous avons présenté le contexte applicatif de nos travaux de recherche : le contexte industriel, les cas d'usages des opérateurs de maintenance pour (1) chercher des POIs autour d'eux et (2) trouver l'information technique pertinente au sein d'un POI ouvert, et enfin l'application RA d'aide à la maintenance des équipements industriels conçue par Schneider Electric.

Étendre le champ de vision de l'utilisateur pour faciliter la recherche de POIs et améliorer le pointage d'objets numériques en RA constituent l'objectif applicatif de nos travaux de recherche.

Dans ce manuscrit, nous retrouvons ces deux sujets de recherche respectivement dans la partie 1 et 2. La partie 1 est dédiée à de nouvelles approches pour la visualisation de l'espace hors-champ, tandis que la partie 2 est consacrée à de nouvelles techniques de pointage en RA.

PARTIE I

TECHNIQUES DE VISUALISATION DE L'ESPACE HORS-CHAMP

3

Sommaire

3.1	Introduction	35
3.2	Le besoin en RA mobile	36
3.3	Approches pour la visualisation de l'espace hors-champ	37
3.3.1	Zoom	37
3.3.2	Overview+Detail	39
3.3.3	Focus+Context	41
3.3.4	Vues contextuelles	43
3.3.5	Autre modalité non graphique, la modalité lumineuse	44
3.3.6	Ce qu'il faut retenir	45
3.4	Vues contextuelles	46
3.4.1	Indices entièrement visibles à l'écran	46
3.4.2	Indices partiellement visibles à l'écran	51
3.4.3	Synthèse des vues contextuelles	55
3.5	Conclusion	57

3.1 Introduction

Naviguer dans un espace de grande taille, qu'il soit réel (environnement industriel) ou numérique (cartes, documents textuels, etc.) nécessite que les utilisateurs acquièrent en permanence des informations qui dépassent leur champ de vision. Que ce soit en utilisant un écran, une caméra ou l'œil humain, l'espace observable est limité.

Alors que la vision binoculaire humaine atteint 190 degrés, celle-ci n'est pas uniforme. La zone de commodité permettant d'observer et de lire des informations avec haute précision est comprise entre 20 et 40 degrés. Au-delà, la compréhension du champ de vision diminue : les couleurs restent visibles jusqu'à un angle compris entre 60 et 120 degrés. Au-delà, seule la lumière est perceptible. Par exemple, des écrans d'ordinateurs 16/9 vendus dans le commerce permettent d'obtenir un champ de vision de 90 à 120 degrés. Les écrans

ultrawide sont très répandus dans les domaines artistiques tels le graphisme, car l'utilisateur nécessite l'ouverture simultanée de multiple fenêtres. En Réalité Augmentée (RA), le champ de vision dépend de longueur focale de la caméra utilisée pour observer l'environnement. Une caméra grand angle (> 120 degrés) permet de couvrir plus d'éléments, mais déforme l'image (perspectives faussées, etc.).

Deux facteurs expliquent donc la réduction de l'espace observable : (1) le champ de vision restreint de l'appareil utilisé pour observer l'environnement (caméra ou œil humain) limite la connaissance de l'environnement, (2) l'écran utilisé pour consulter les informations possède une taille fixe, et limite le nombre d'éléments pouvant y être affichés. L'objectif des techniques de visualisation hors-champ est donc de permettre à l'utilisateur d'étendre sa connaissance sur le contexte environnant. Dans le cas de la RA mobile, les éléments numériques superposés à la réalité (Points d'intérêts - POIs) peuvent être nombreux et disséminés dans l'environnement (voir les exemples du chapitre précédent). L'objectif est donc de permettre à l'utilisateur d'étendre son champ de vision pour le rendre conscient des POIs qui l'entourent.

Nous présentons une revue des techniques de visualisation hors-champ, en détaillant particulièrement les vues contextuelles, qui constituent le sujet de nos travaux. Pour chaque technique, nous décrivons son principe et l'illustrons par des visualisations sur un écran fixe ou mobile. Nous détaillons ensuite son application à la RA mobile, quand la technique le permet.

3.2 Le besoin en RA mobile

En RA mobile, la taille limitée des écrans implique que les utilisateurs ne peuvent visualiser qu'une portion d'un grand espace d'informations. Le besoin d'acquérir des informations qui dépassent le cadre de l'écran est donc fréquent. L'ajout d'informations pour étendre le champ de vision de l'utilisateur et augmenter sa connaissance spatiale est nécessaire. Nous nous intéressons à un type d'information, les points d'intérêts (POIs) disséminés dans l'environnement de l'utilisateur et non visibles sur l'écran : les POIs hors-champ.

La visualisation de l'espace hors-champ permet de situer la vue courante de l'utilisateur par rapport à l'environnement entier. Cependant, il convient de ne pas surcharger le centre de l'écran avec des informations supplémentaires sur les POIs hors-champ. L'attention principale des utilisateurs est en effet concentrée sur la zone centrale de l'écran. Les interactions visuelles et tactiles dans cette zone peuvent y être fréquentes, telles que la consultation et la modification d'informations associées à des POIs visibles : dans les scénarios 1 et 2 (voir section 2.3), les opérateurs de maintenance doivent visualiser les données associées aux POIs visibles tels que les valeurs numériques, la documentation ou des graphiques. Nous appelons cette zone *zone de travail* ou *vue détaillée*, car il s'agit d'une portion spécifique d'un grand espace d'informations qui est assez détaillée pour réaliser la tâche. L'affichage d'informations supplémentaires sur l'espace hors-champ, que nous

nommons *vue globale*, ne doit donc pas distraire les utilisateurs de leurs tâches principales ou trop masquer la *vue détaillée* (zone de travail de l'utilisateur). Nous soulignons que la zone sur les bords de l'écran est moins utilisée pour interagir avec le système ou pour visualiser des informations. L'affichage de repères visuels sur les POIs hors-champ dans une zone périphérique de l'écran est par conséquent un moyen non intrusif d'étendre le champ de vision de l'utilisateur. Ces repères visuels sur les POIs hors-champ peuvent être affichés sous forme textuelle (page web, etc.), ou graphique (cartes, etc.).

Enfin, un grand nombre d'objets hors-champ peuvent correspondre à l'intérêt des utilisateurs. Comme illustré par les deux scénarios (voir section 2.3), la distribution des POIs peut varier considérablement pour différentes machines, tant en termes de densité globale que de groupes de POIs localement denses (voir Figure 2.2) : les POIs hors-champ peuvent être situés tout autour des utilisateurs (scénario 1) ou organisés en clusters à des emplacements spécifiques (scénarios 1 et 2). Ces deux situations peuvent entraîner un écran excessivement occupé si le nombre de POIs hors-champ est élevé. Dans le premier cas, l'écran complet sera surchargé, tandis que dans le deuxième cas, l'encombrement visuel pourra être localisé à des endroits spécifiques de l'écran.

Nous déduisons deux critères pour les techniques de visualisation hors-champ en RA mobile : (1) minimiser l'intrusion visuelle pour éviter de surcharger la zone de travail de l'utilisateur, et (2) gérer les environnements denses en POIs. Ces deux critères structurent notre synthèse des techniques du Tableau 3.2.

3.3 Approches pour la visualisation de l'espace hors-champ

Quatre approches sont utilisées pour étendre la connaissance spatiale de l'utilisateur [35] : les techniques Zoom séparent temporellement les vues détaillée et globale de l'environnement. Les interfaces Overview+Detail utilisent une séparation spatiale pour présenter les deux vues, tandis que les techniques Focus+Context les affichent sur une seule vue combinée. Enfin, les vues contextuelles, dérivées des techniques Focus+Context, superposent la vue globale à la vue détaillée sur le contour de l'écran.

Cette taxonomie s'applique autant aux écrans classiques qu'à des écrans plus petits de dispositifs mobiles et inclut donc la RA mobile. Notre revue est organisée selon ces quatre approches.

3.3.1 Zoom

Les interfaces de type *Zoom* impliquent une séparation temporelle entre la vue détaillée et la vue globale de l'environnement. A tout instant, une seule des deux vues est affichée sur l'écran. L'utilisateur passe de l'une à l'autre en zoomant ou en dé-zoomant sur l'interface. Les niveaux de zoom permettent donc d'accéder à différentes représentations de

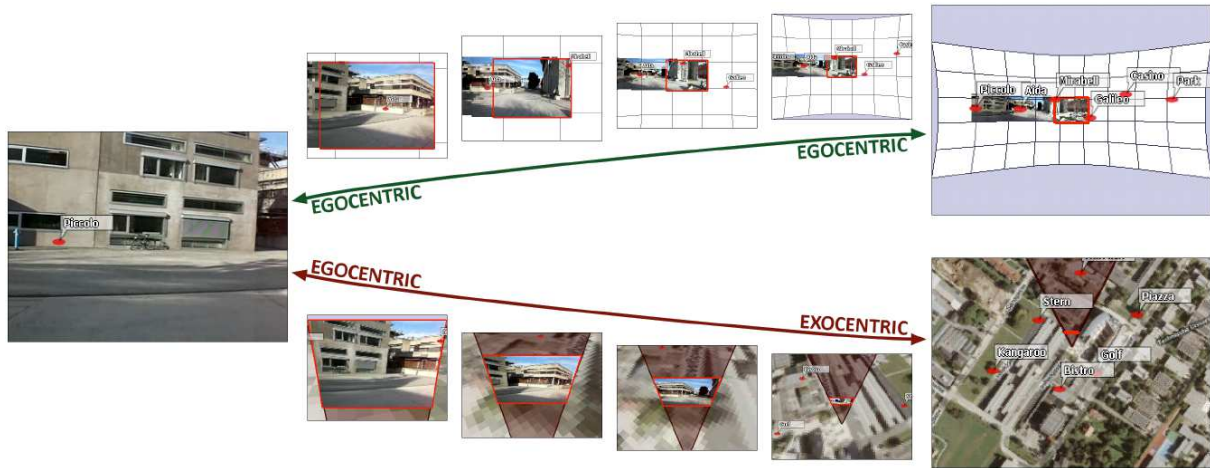


FIGURE 3.1 – Interfaces de zoom proposées en RA : en haut, zoom égocentrique élargissant le champ de vision de l'utilisateur jusqu'à 360 degrés. En bas, zoom excentrique offrant une vue radar 2D. Image issue de [82].

l'environnement.

La manière d'interagir avec les interfaces de type zoom dépend du type d'appareil utilisé. Sur ordinateur, le zoom à l'aide de la roulette centrale de la souris est le plus communément utilisé. Sur des applications de cartographie comme Google Maps, le double clic à l'aide du bouton gauche (resp. bouton droit) de la souris permet d'augmenter (resp. diminuer) le niveau de zoom. Certains logiciels de traitement d'images donnent également la possibilité de modifier le niveau de zoom en utilisant la combinaison CTRL+roulette de la souris. Enfin, le zoom sur surfaces tactiles peut être réalisé en écartant deux doigts posés sur l'écran, ou à l'aide d'un double tap. Dans tous les cas, il est rare de pouvoir annuler une action de zoom sans avoir à dé-zoomer manuellement.

En RA mobile, Mulloni et al. [82] ont proposé deux interfaces de type zoom pour étendre la connaissance de l'utilisateur sur l'environnement hors-champ : la première permet d'obtenir une vue panoramique à 360 degrés, la deuxième une vue radar de dessus (voir Figure 3.1). Dans le premier cas, la transition se fait à l'aide des boutons "+" et "-" placés sur les côtés du téléphone. L'utilisateur conserve son point de vue égocentrique et obtient une vision élargie des points d'intérêts autour de lui. L'avantage principal est que la vue augmentée du monde réel est conservée sur l'écran. En revanche, l'utilisateur n'a pas d'informations sur la distance qui sépare les points d'intérêts. Dans le deuxième cas, l'utilisateur bascule sur une vue radar excentrique en abaissant le téléphone vers le sol. Les points d'intérêts lui sont présentés sur une carte ainsi que sa position et son orientation. Il obtient des informations sur la distance qui sépare les POIs mais perd la vue temps réel de la caméra. Le lien entre monde physique et monde numérique est brisé, il ne s'agit plus de RA.

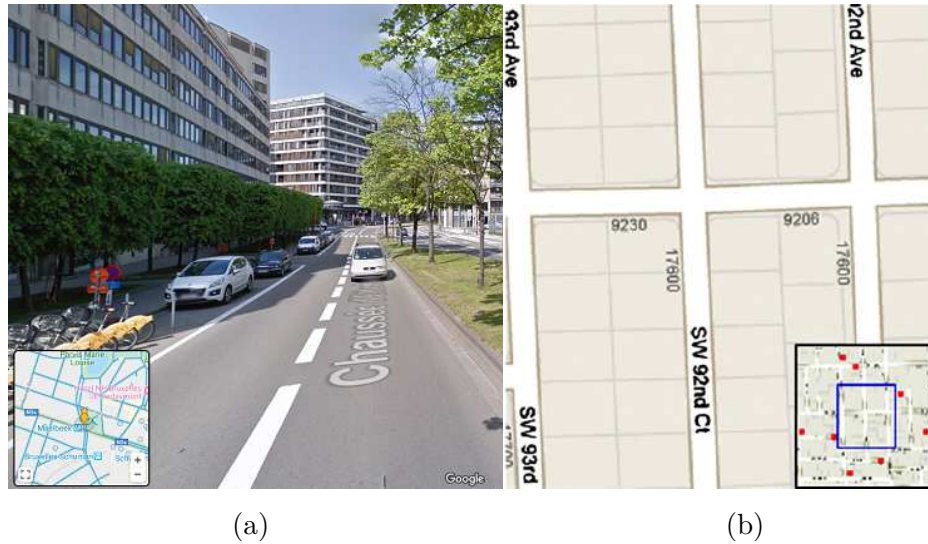


FIGURE 3.2 – Interface Overview+Detail affichée sur a) ordinateur (image issue de Google Maps) b) téléphone mobile (image issue de [26])

3.3.2 Overview+Detail

Les techniques Overview + Detail affichent simultanément la vue globale de l'environnement (Overview) et la vue détaillée sur une section de cet environnement (Detail). Ces deux vues sont spatialement séparées en deux fenêtres distinctes, la vue globale étant superposée à la vue détaillée. Ce type d'interface est couramment utilisé dans les applications cartographiques (tel Google Maps, voir Figure 3.2a), et les bénéfices sont triples :

1. La navigation est plus rapide car les individus peuvent se déplacer sur la carte en utilisant directement la vue globale au lieu de la vue détaillée [16].
2. La vue globale aide les utilisateurs à se repérer dans un grand espace [87]
3. Les interfaces de type Overview+Detail donnent aux utilisateurs un sentiment de contrôle [99].

Cependant, la vue globale masque une partie de la vue détaillée (zone de travail). Le changement d'attention nécessaire lors du passage d'une vue à l'autre a un impact négatif sur les performances des utilisateurs [63], car leur attention s'écarte de la zone de travail. Cet aspect est confirmé par Hornbaek et al. [63], en comparant sur ordinateur les interfaces Overview+Detail aux interfaces Zoom. L'étude comportait deux tâches : une première tâche de navigation, dont le but est de trouver un objet spécifique sur la carte, et une deuxième tâche d'exploration, consistant à scanner la carte pour trouver l'ensemble des objets répondant à des critères spécifiques. Les résultats ont montré que les participants étaient plus rapides à l'aide de l'interface Zoom, particulièrement lors des tâches de navigation. En effet, le nombre de transitions entre les fenêtres vue globale et vue détaillée impacte fortement le temps pour rechercher des objets sur une carte. Une action (déplacement, zoom) sur la vue détaillée modifie instantanément la vue globale, pour garder une cohérence entre les deux fenêtres. La vue globale peut donc être distrayante et attirer constamment l'attention des sujets. De plus, le changement de contexte entre les deux vues implique un effort mental additionnel pour se replacer dans le nouveau repère

[14]. Enfin, la navigation à l'aide de la vue globale peut-être difficile car ce type d'interface ne permet pas d'effectuer des déplacements fins lorsque le niveau de zoom est proche de nul. Les utilisateurs doivent donc corriger le déplacement à l'aide de la vue détaillée, ce qui implique des transitions supplémentaires entre les deux fenêtres.

Les nombreux travaux sur les visualisations de type Overview+Detail [1, 16, 29, 25, 63, 87, 99, 111] ont donné lieu à un ensemble de principes de conception qui inclut :

1. La vue globale et la vue détaillée nécessitent d'être couplées pour qu'un changement sur une des vues (déplacement du champ visuel et sélection) affecte immédiatement l'autre fenêtre [1].
2. Le facteur de zoom, défini comme le ratio entre le plus grand et le plus petit grossissement des deux fenêtres, doit être inférieur à 25 [87] ou à 30 [99].
3. La taille de la vue globale a une grande influence sur le nombre d'informations qui peuvent y être affichées ainsi que sur la facilité de navigation. Hornbaek K et al. [63] précisent que la taille de la vue globale devrait être égale à 1/16 de celle de la vue détaillée pour les applications sur ordinateur. Une vue globale de taille plus importante a l'inconvénient de cacher un nombre d'informations non négligeable sur l'écran. La taille utile de la vue globale dépend également de la tâche à réaliser [87] : une large fenêtre serait par exemple nécessaire pour des tâches de monitoring (tâches de supervision permettant l'acquisition de données (mesures, état de fonctionnement, alarmes etc.) pour la surveillance du bon état de fonctionnement des équipements).

Enfin, sur supports mobiles (voir Figure 3.2b), la manipulation directe de la vue globale pour déplacer la vue détaillée permet (1) de compenser la petite taille de la vue globale, due aux dimensions restreintes de l'écran et (2) d'améliorer les performances de navigation de l'utilisateur [25].

En RA, l'application *Spatial Wearable Conference Space* [19] utilise une visualisation de type Overview+Detail pour étendre la connaissance de l'utilisateur sur l'environnement hors-champ. Un petit écran radar indique la localisation de chaque participant dans l'espace de conférence par rapport à l'emplacement de l'utilisateur (voir Figure 3.3a). Plus récemment, la nouvelle version de Google Maps en RA permet de visualiser un itinéraire directement sur la vue principale de la caméra. L'application inclut également une visualisation de type Overview+Detail : une carte est affichée sur la partie inférieure de l'écran (voir Figure 3.3b), et permet de situer l'utilisateur par rapport à l'itinéraire global.

Ce type de visualisation est donc efficace pour étendre la connaissance spatiale de l'utilisateur, mais impose un effort supplémentaire à l'utilisateur pour se réorienter lorsqu'il passe d'une vue à l'autre.

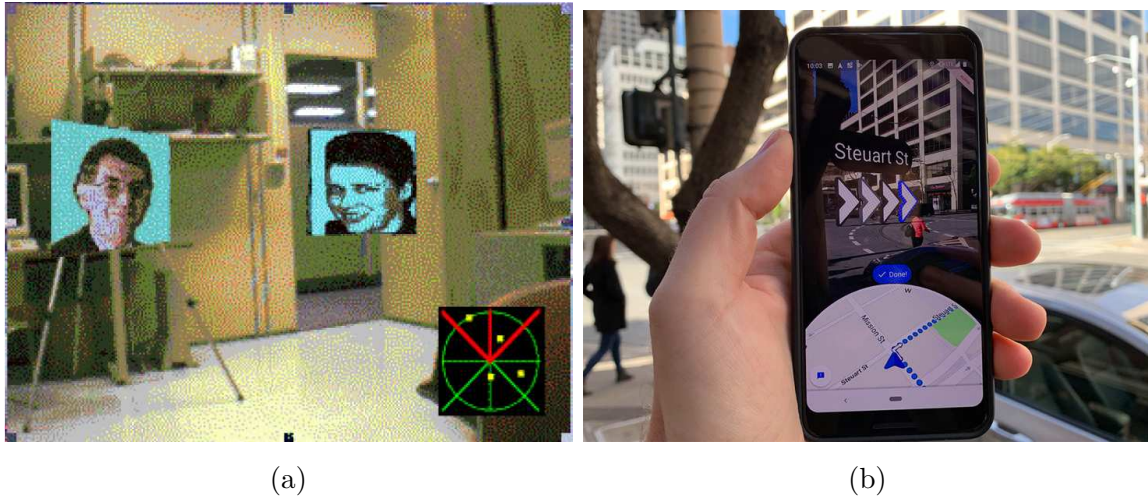


FIGURE 3.3 – a) Application *Spatial Wearable Conference Space*, affichant sur une carte l'emplacement des participants à la conférence. Image issue de [19] b) Application *Google Maps RA*, affichant sur la carte l'itinéraire à suivre.

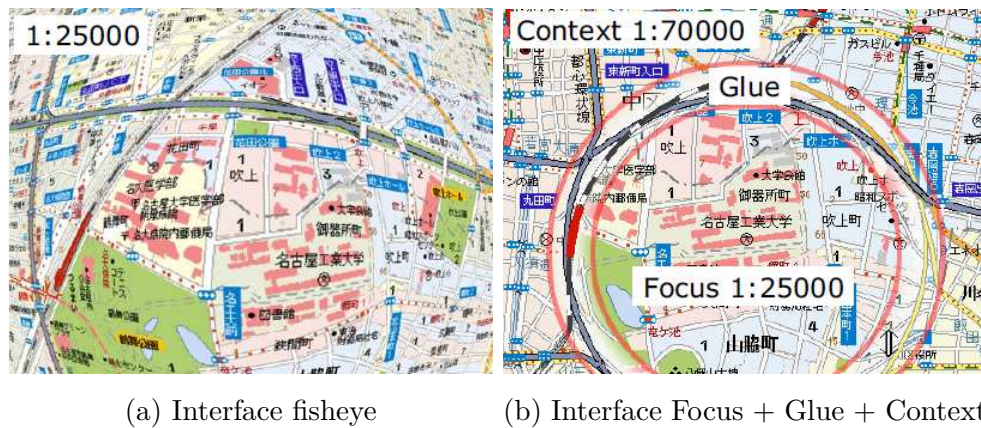


FIGURE 3.4 – a) La distorsion est appliquée à la vue entière. b) Le focus et le contexte ne sont pas déformés. Seule la zone Glue présente une distorsion. Image issue de [116].

3.3.3 Focus+Context

Focus+Context est une technique de visualisation qui affiche sur une seule et même fenêtre une vue générale de l'environnement. Elle offre une transition entre la vue détaillée sur laquelle travaille l'utilisateur (le focus) et la vue globale entourant cette zone (le contexte) [35]. Les techniques de type Focus+Context suppriment donc la séparation temporelle (section 3.3.1 Zoom) et spatiale (section 3.3.2 Overview+Detail) entre les deux vues.

La vue "Focus" se trouve au centre de l'écran et représente une vue zoomée et détaillée de l'environnement. La vue "Context" qui l'entoure constitue la vue globale de l'environnement et permet à l'utilisateur de percevoir à petite échelle le cadre entourant la région ciblée. Les techniques Focus+Context permettent donc d'observer simultanément différents niveaux de zoom.

Les visualisations de type "fisheye" (voir Figure 3.4a) constituent un exemple d'interfaces

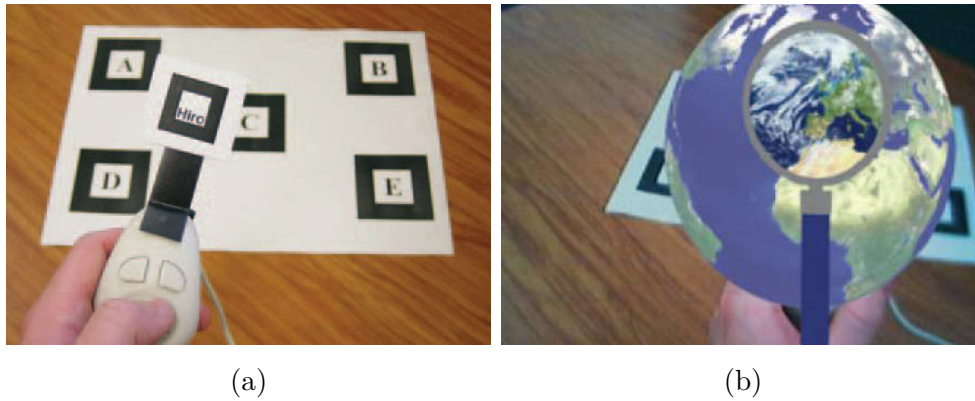


FIGURE 3.5 – Application AR Magic Lens. a) Marqueurs visuels disposés sur une table (pour y superposer le globe terrestre) et poignée tenue par l'utilisateur où est placé un autre marqueur (pour y superposer la loupe). b) Loupe (vue détaillée) permettant de visualiser un ensemble de données propre du globe terrestre virtuel (vue globale) tel que la couche de nuages. Image issue de [20].

Focus+Context. Le terme fisheye provient de l'objectif photographique du même nom. Ce type d'objectif possède une distance focale très courte et conduit à un angle de champ très grand. Il déforme ainsi l'image en courbant fortement les lignes droites qui ne passent pas par le centre. En visualisation, la distorsion fisheye agrandit la région locale autour de la souris, tout en laissant le graphique autour inchangé. La région ciblée est donc magnifiée, tout en préservant l'environnement autour. Cette technique est par exemple utilisée sur les ordinateurs Apple : les icônes de la barre de tâches s'agrandissent lorsque le curseur se rapproche d'elles. En revanche, selon le niveau de distorsion utilisé, les vues de type "fisheye" peuvent engendrer des difficultés pour trouver une cible, dues à une mauvaise interprétation des données contenues dans la zone agrandie et déformée [35]. Le F+G+C (Focus+Glue+Context) [116] est une technique qui permet de pallier à la déformation de toute la topologie de l'espace : la vue détaillée et la vue globale ne sont cette fois pas déformées, et toute la distorsion se concentre uniquement sur une zone restreinte de l'écran, la zone "Glue" (voir Figure 3.4b). Elle se situe entre la vue détaillée et la vue globale et sert de transition pour passer d'une visualisation à petite échelle à une visualisation à grande échelle. En appliquant cette technique à des cartes routières, la lisibilité des itinéraires est augmentée [116].

AR Magic Lens [20] est un exemple de technique Focus+Context en RA, qui permet de visualiser des données virtuelles : les utilisateurs peuvent par exemple inspecter un globe terrestre virtuel (voir Figure 3.5) à travers une loupe, également virtuelle. Cette lentille permet de magnifier le globe pour y observer des sous-ensembles de données tels que les nuages, la pollution lumineuse, etc. Elle est superposée au marqueur visuel d'une poignée tenu par l'utilisateur, tandis que le globe est affiché sur les marqueurs visuels placés sur une table. La loupe est donc contrôlée par les mouvements de main de l'utilisateur, tandis que le globe est contrôlé par le placement de la caméra autour de la table. Comme la lentille ne magnifie qu'une partie du globe, l'utilisateur conserve une vue globale sur la

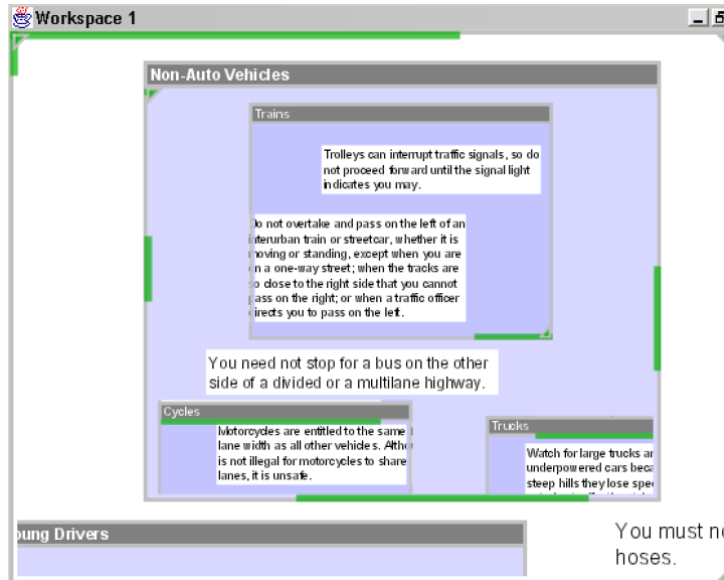


FIGURE 3.6 – Vue contextuelle à l’aide de CityLights : des lignes épaisses sur les bordures des fenêtres indiquent la position, taille et distance des objets textuels non visibles. Les triangles dans les coins des fenêtres indiquent que davantage d’objets invisibles se trouvent dans l’espace hors-champ associé. Image issue de [73].

Terre. Dans ce cas, la loupe représente la vue détaillée, tandis que la Terre constitue la vue globale.

Les techniques Focus+Context ont donc l’avantage d’afficher la vue détaillée et la vue globale simultanément, sans que les deux fenêtres ne se recouvrent. Cependant, ces techniques impliquent une réduction de la taille de la vue détaillée pour permettre à la vue globale d’être affichée le long des bords de l’écran.

3.3.4 Vues contextuelles

Les vues contextuelles sont dérivées des techniques Focus+Context [59, 73]. Cependant, elles ne réduisent pas la taille de la zone de la vue détaillée pour afficher une vue complète de l’espace environnant. Au lieu de cela, ces techniques superposent le long des bordures de l’écran des indices visuels pour situer la vue détaillée dans son environnement. L’environnement n’est plus complètement représenté. Seuls des indices sur l’environnement sont représentés par exemple sous des formes abstraites (proxies) [35].

Sur le navigateur Safari, la barre de défilement n’est pas affichée à côté de la page web, dans un espace réservé, mais par-dessus. Ainsi, la taille de la vue détaillée est préservée, tout en permettant à l’utilisateur de se situer dans la page. City lights [73] est un autre exemple de vue contextuelle qui utilise les bords de l’écran pour étendre la connaissance de l’utilisateur sur l’espace hors-champ. Appliquée à un environnement textuel, la visualisation superpose des lignes épaisses sur les bords de la vue détaillée pour indiquer la position ainsi que la taille des objets textuels non visibles à l’écran (voir Figure 3.6). L’utilisateur est ainsi en mesure d’estimer la localisation de l’objet hors-champ (position de la ligne par rapport à

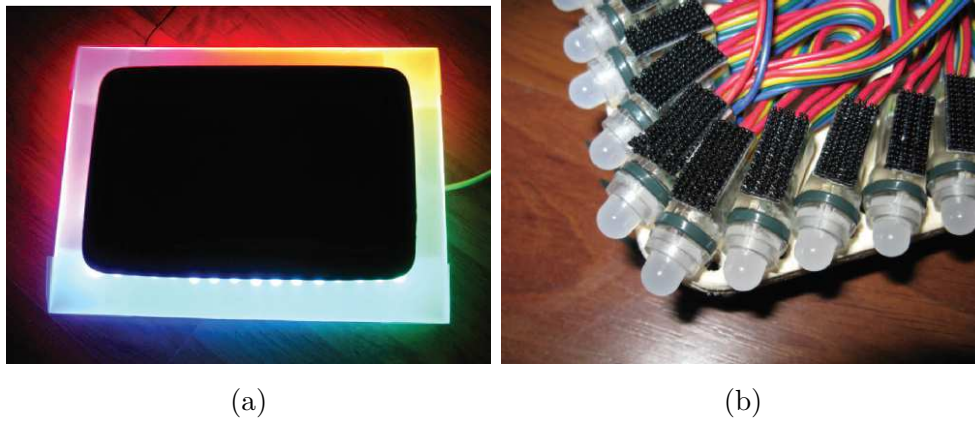


FIGURE 3.7 – a) Sparkle avec tous les LEDs allumés. b) LEDs placés sur la face arrière d’une tablette. Image issue de [81].

la largeur de l’écran), sa taille (longueur de la ligne par rapport à la largeur de l’écran), et sa distance (couleur de la ligne).

Objet de nos travaux, ce type de visualisation a été appliquée au cas de la RA mobile. Nous détaillons les techniques existantes dans la section 3.4.

3.3.5 Autre modalité non graphique, la modalité lumineuse

La modalité lumineuse a été proposée pour permettre la visualisation des POIs hors-champ sans empiéter sur l’écran. Dans le cas d’applications où les informations affichées sont distribuées sur toute la surface de l’écran, afficher des indications supplémentaires sur l’espace hors-champ n’est pas possible. Ce cas concerne les applications pour la gestion de crise utilisées par les pompiers ou encore les Portable Pilot Units utilisés par les pilotes en mer. Ces systèmes informatisés portables aident l’utilisateur à prendre des décisions pour gérer des situations problématiques, voir dangereuses. Il convient donc de ne pas perturber la tâche principale de l’utilisateur, qui s’effectue sur tout l’écran. Il est en revanche nécessaire d’opter pour une approche qui exploite une autre modalité que celle graphique sur l’écran.

Sparkle [81] est une technique de visualisation sans aucune intrusion sur l’écran (voir Figure 3.7a). La visualisation hors-champ est basée sur la lumière. Le système repose ainsi sur une série de LEDs positionnés sur le contour de la face arrière d’une tablette (voir Figure 3.7b) pour indiquer la position et la distance des POIs situés en dehors du champ de vision (et donc non affichés sur l’écran de la tablette) : une lumière intense (resp. peu intense) représente un POI proche (resp. loin) du bord de l’écran. Cependant, Sparkle est limité : le système n’est utilisable que dans une pièce sombre, pour que les LEDs soient correctement visibles.

En réalité mixte, PeriMR [51] est un casque utilisant un éclairage périphérique pour étendre la connaissance de l’utilisateur sur l’espace hors-champ. Gruenefeld et al. [51] ont choisi trois principes pour leur prototype : (1) la taille de l’outil doit permettre son inser-

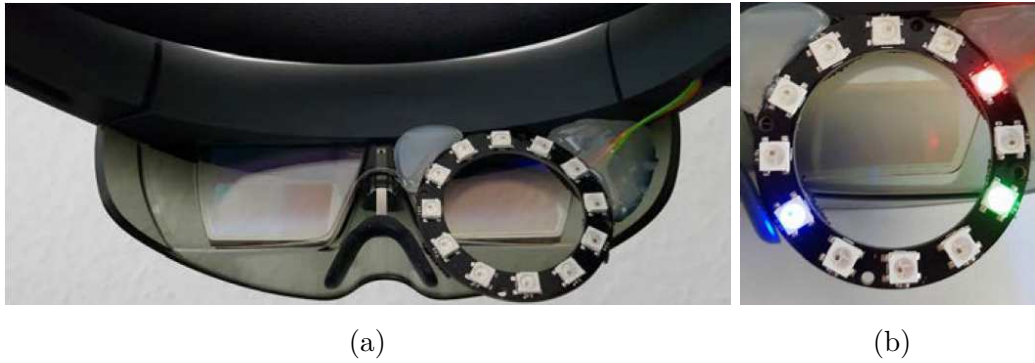


FIGURE 3.8 – a) Système MonocularAR attaché au casque HoloLens. b) LEDs placés sur le contour d’une moitié de la visière de l’HoloLens. Image issue de [50].

tion autant dans les casques de réalité virtuelle que dans les casques semi-transparents de RA, (2) l’outil doit être re-configurable en fonction du patron de conception choisi (emplacement des LEDs sur le casque), et enfin (3) l’outil doit être abordable financièrement. Le prototype a été développé avec un casque de réalité virtuelle basé sur le modèle de Google Cardboard. Ce type de casque est un support où un téléphone peut être placé entre les lentilles et le bord externe du casque. Pour choisir l’emplacement des LEDs, trois positions ont été étudiées : entre les lentilles du casque et le téléphone, entre les yeux et les lentilles du casque, et enfin sur le contour interne du casque. Les deux dernières positions ont été privilégiées, et la dernière écartée dues aux reflets des LEDs sur les lentilles.

MonocularAR [50] utilise également un éclairage périphérique pour étendre le champ de vision restreint des casques de RA, comme l’Hololens. Le système n’augmente qu’un seul œil avec 12 LEDs positionnés radialement (voir Figure 3.8a). La couleur de la lumière est celle de l’objet hors-champ représenté, ou un mélange des couleurs des objets distants s’ils sont placés dans la même direction (voir Figure 3.8b). Cependant, MonocularAR s’est révélé moins performant (temps de complétion et préférence des utilisateurs) dans un environnement peu dense (jusqu’à 3 POIs) par rapport à une technique ”on-screen” où les indices visuels sont affichés directement sur l’écran. L’étude n’impliquait pas d’environnements à forte densité.

Bien que les techniques de visualisation hors-champ offrent une vision périphérique des POIs distants sans aucune intrusion à l’écran, elles nécessitent l’ajout de matériel. De plus, l’espace hors-champ est discrétisé par les LEDs, ce qui diminue la précision du système.

3.3.6 Ce qu’il faut retenir

Les quatre principales approches détaillées précédemment étendent la connaissance spatiale de l’environnement en apportant une vue globale à la vue détaillée de l’utilisateur. Mais lorsque la taille de l’écran est limitée, comme celui des appareils mobiles (téléphones, tablettes, etc.), il convient de limiter l’intrusion à l’écran de la visualisation hors-champ (voir critères de la section 3.2). Dans le respect de ce critère, il est nécessaire de (1) ne pas déformer la vue détaillée à l’écran et (2) ne pas masquer ni réduire la vue détaillée

Technique	Distorsion	Occultation	Description
Zoom	NON	OUI	Affichage d'une seule des deux vues détaillée ou globale à l'instant t
Overview+Detail	NON	OUI	Occultation de la vue détaillée par la vue globale
Focus+Contexte	OUI	OUI	Occultation de la vue détaillée par la vue globale sur la bordure de l'écran
Vues contextuelles	NON	PARTIELLE	Superposition de la vue globale sur la vue détaillée et occultation partielle de celle-ci par les indices visuels placés sur le bord de l'écran

TABLE 3.1 – Synthèse des 4 approches pour la visualisation de l'espace hors-champ

actuelle de l'utilisateur. Nous résumons dans le Tableau 3.1 chaque type de visualisation hors-champ par rapport à ces deux exigences.

Nous nous concentrons dans la suite sur les vues contextuelles car elles n'impliquent aucune distorsion à l'écran et cachent un minimum d'informations de la vue détaillée. Elles s'adaptent donc parfaitement aux écrans d'appareils mobiles tout en évitant les tâches supplémentaires liées au multiplexage temporel du zoom.

3.4 Vues contextuelles

Les vues contextuelles sont particulièrement adaptées aux écrans dont la taille est limitée. Comme adopté par Gustafson et al. [59] et Trapp et al. [105], nous passons en revue les vues contextuelles en distinguant les techniques dont les indices visuels sont entièrement visibles à l'écran, des techniques dont les indices visuels ne sont que partiellement visibles à l'écran. Pour chacune d'entre elles, nous différencions les visualisations basées sur une vue orthographique descendante 2D, des visualisations hors-champ 3D ou adaptées à la RA mobile (techniques affichant des directions 3D vers des points d'intérêt situés dans un espace 3D).

3.4.1 Indices entièrement visibles à l'écran

3.4.1.1 Visualisation d'objets hors-champ en 2D

Les flèches sont couramment utilisées pour donner des informations de direction vers des points d'intérêt (POIs) hors-champ. Deux techniques 2D proposées par Burigat et al. [27] exploitent respectivement la taille (voir Figure 3.9a) et la longueur des flèches (voir Figure 3.9b) pour indiquer la distance de ces POIs par rapport à l'utilisateur : plus la flèche est grande (resp. longue), plus le POI affiché à l'écran est proche de l'utilisateur. Cependant, les visualisations n'ont pas été évaluées dans des environnements denses, mais

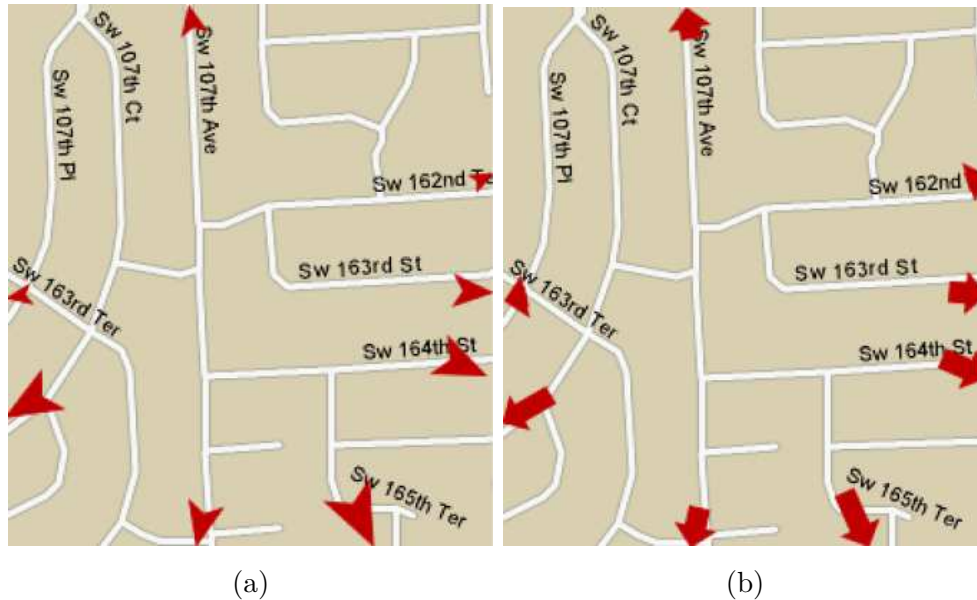


FIGURE 3.9 – Flèches dont la distance aux POIs est symbolisée par a) la taille b) la longueur. Image issue de [27].

avec un maximum de 8 POIs.

EdgeRadar [58] est une technique de visualisation 2D qui utilise une petite partie de chaque bord de l'écran pour afficher l'emplacement des POIs hors-champ (voir Figure 3.10a). Le centre de l'écran définit la vue détaillée ou zone de travail de l'utilisateur tandis que les informations sur la vue globale ou contexte hors-champ ne s'affichent que sur les bords de l'écran. Les POIs distants sont indiqués par des icônes carrées, placées sur les bords de l'écran. Ainsi, la distance entre l'icône et la bordure de l'écran indique à quel point le POI hors-champ est distant. La technique est également utile pour fournir des informations sur la relation spatiale entre les POIs : EdgeRadar permet de visualiser le rapprochement/éloignement de la totalité des POIs du contexte les uns par rapport aux autres. Cependant, EdgeRadar atteint sa limite lorsque la densité de POIs augmente ou lorsque les POIs sont regroupés : dans de tels cas, la bordure de l'écran est surchargée avec de petites icônes carrées.

Basé sur EdgeRadar, EdgeSplit [64] permet la visualisation et la sélection de POIs hors-champ. Les objets hors-champ sont représentés par des carrés autour de la bordure de l'écran. Les bordures sont divisées en zones rectilignes, chaque zone hébergeant un objet. L'utilisateur peut sélectionner une zone pour obtenir des informations sur le POI hors-champ (voir Figure 3.10b). EdgeSplit souffre des mêmes inconvénients que EdgeRadar : une densité élevée de POIs hors-champ ou des POIs répartis en clusters diminueront l'utilisabilité de la visualisation. En effet, les zones sélectionnables deviendront de plus en plus petites, ce qui impactera leur sélection.



FIGURE 3.10 – a) Visualisation *EdgeRadar* : les carrés bleus sur le bord de l'écran indiquent la position et la distance des POIs hors-champ. Image issue de [58]. b) *EdgeSplit* : l'utilisateur sélectionne les POIs hors-champ directement sur l'écran. Image issue de [64].

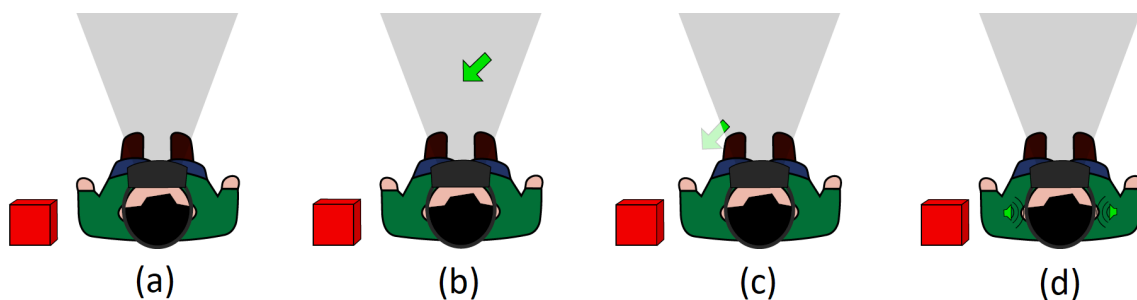


FIGURE 3.11 – Visualisation *FlyingARrow*. (a)-(d) Objet hors-champ en rouge, champ de vision de l'utilisateur en gris. (b) Flèche 3D dans le champ de vision de l'utilisateur. (c) Animation de la flèche vers l'objet hors-champ. (d) Son dans le casque de RA lorsque la flèche atteint le POI. Image issue de [49].

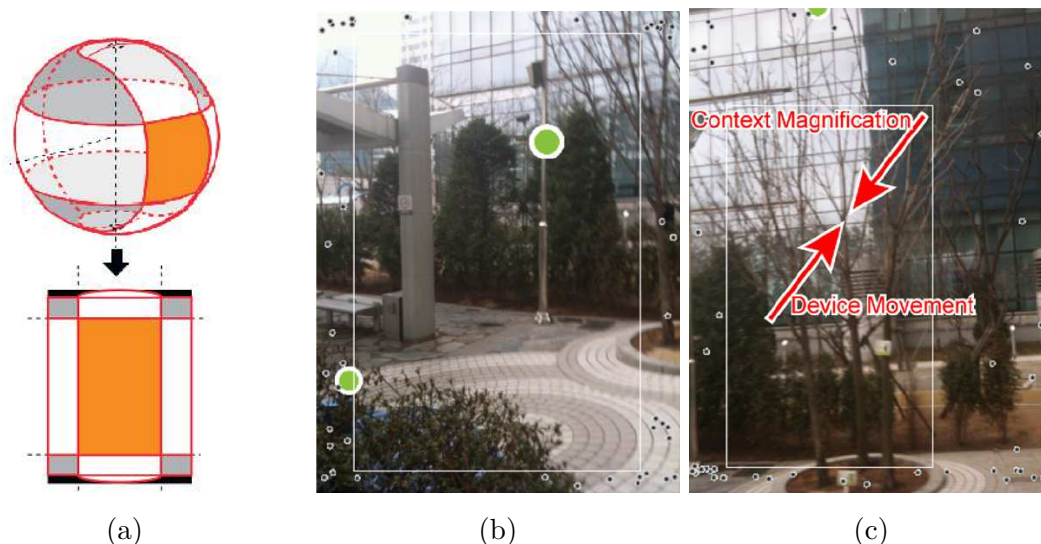


FIGURE 3.12 – a) Projection des coordonnées 3D en une vue fisheye 2D. (b)-(c) Visualisation Aroundplot pour 50 POIs hors-champ lorsque b) l'utilisateur est statique c) l'utilisateur déplace l'appareil mobile vers le haut et la droite. Images issues de [65].

3.4.1.2 Visualisation d'objets hors-champ en 3D/RA

Les flèches 3D [34] affichent des directions 3D pour atteindre des POIs hors-champ. Implémentées d'un point de vue égocentrique, les flèches 3D permettent aux utilisateurs de prendre en compte toutes les directions qui les entourent. La visualisation indique par exemple si un POI est localisé devant ou derrière l'utilisateur. Les flèches 3D ont été comparées aux flèches 2D et au radar 2D [34]. La tâche consistait à rechercher cinq points d'intérêt dans un ordre spécifié. Les résultats ont montré que les utilisateurs effectuaient plus rapidement les tâches avec les flèches 3D. Cependant, lorsque la densité de points d'intérêts est élevée, l'occultation entre les flèches réduit considérablement les performances de navigation des utilisateurs. Avec FlyingARrow [49], Gruenefeld et al. proposent une technique où les flèches 3D sont animées et associées à un signal acoustique : la technique repose sur la théorie de la perception amodale appliquée à un mouvement. Un utilisateur est capable de compléter un mouvement uniforme lorsque seule une partie de sa trajectoire est visible. Pour chaque POI hors-champ, une flèche 3D se déplace du champ de vision de l'utilisateur au POI hors-champ, et joue un son uniquement lorsque l'objet distant a été atteint (voir Figure 3.11). La direction est donc indiquée par l'orientation de la flèche et la distance est codée par la durée d'animation de la flèche.

En RA mobile, Hyungeun et al. ont proposé Aroundplot [65]. Inspiré de EdgeRadar [58], AroundPlot affiche sur chacun des bords de l'écran des indices visuels sur l'emplacement des points d'intérêt distants (voir Figure 3.12b). La technique projette les coordonnées sphériques 3D de l'environnement en une vue fisheye 2D orthogonale (voir Figure 3.12a). Elle affiche ensuite sur la bordure de l'écran un petit point pour chaque POI hors-champ. Cependant, la petite taille de cette zone soulève des problèmes de perceptibilité. Pour résoudre cette difficulté, la technique grossit dynamiquement la zone du contexte dans la direction du déplacement de l'appareil mobile (voir Figure 3.12c). A titre d'exemple, un

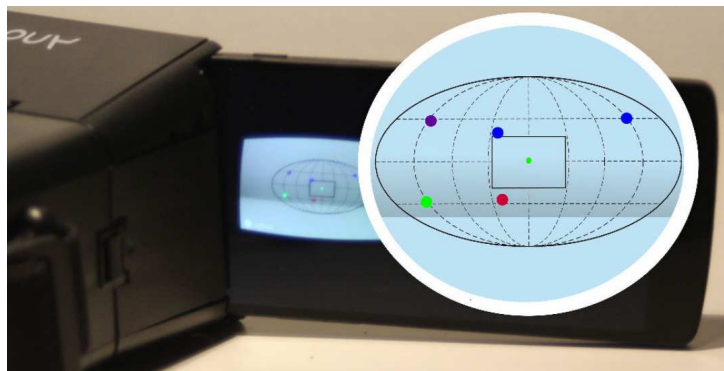


FIGURE 3.13 – Visualisation EyeSee360 ici appliquée à la réalité virtuelle. EyeSee360 est affiché sur un téléphone, lui-même inclus dans un casque de réalité virtuelle Google Cardboard. Image issue de [48].

ensemble de points affichés sur la bordure droite de l'écran indique des POIs accessibles si l'utilisateur effectue une rotation vers la droite. Le déplacement de l'utilisateur vers les POIs est interprété par AroundPlot comme un intérêt de sa part envers cette partie de l'environnement. La visualisation agrandit donc la zone du contexte correspondante pour lui permettre d'avoir plus de détails et une meilleure visibilité sur cette partie de l'environnement. Hyungeun et al. [65] ont comparé AroundPlot à un radar 2D et aux flèches 3D pour les cas d'environnements peu denses (5 POIs) et denses (50 POIs). Les résultats ont montré que les utilisateurs obtenaient de meilleures performances avec AroundPlot (temps de complétion et nombre d'erreurs). En effet, le radar 2D empêche les utilisateurs d'estimer la hauteur (position y) des POIs et l'occultation entre les flèches 3D rend difficile l'interprétation des indices visuels.

EyeSee360 [48] et EyeSee [47], deux techniques de visualisation respectivement adaptées aux casques de réalité virtuelle et de RA, reprennent les visualisations EdgeRadar [58] et AroundPlot [65] : le rectangle intérieur représente la vue détaillée, et la zone à l'extérieur du rectangle désigne la vue globale (voir Figure 3.13). De la même façon qu'AroundPlot, ce rectangle est superposé à la vue détaillée du monde réel. La bordure intérieure séparant la vue détaillée de la vue globale est cependant elliptique et les points représentant les POIs hors-champ sont plus gros. La distance est ici codée grâce à un gradient de couleur, du bleu au rouge. Ces couleurs sont basées sur celles d'une carte de chaleur. Ici le rouge représente un objet proche et le bleu un POI éloigné de l'utilisateur. EyeSee360 s'est révélé légèrement plus performant que les flèches 2D et Halo pour estimer la direction des objets hors-champ, mais aucune différence notable entre les techniques pour l'estimation de la distance de ces objets n'a été observée. En revanche, EyeSee360 n'a été évalué qu'avec des environnements peu denses : 3, 5 et 11 POIs. Une comparaison avec FlyingARrow [49] montre que EyeSee360 possède de meilleures performances pour estimer la localisation d'un POI hors-champ et le ramener dans le champ de vision de la caméra. L'étude n'a cependant été menée qu'avec un seul POI distant.

EyeSeeX [45] va plus loin en proposant des techniques pour minimiser l'intrusion dans

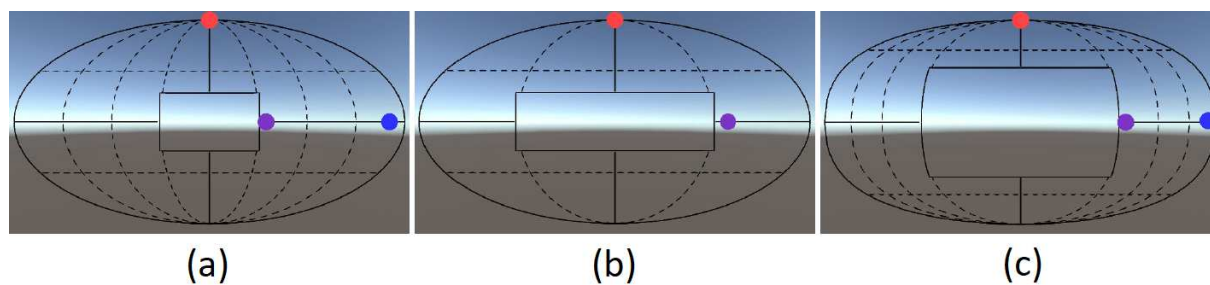


FIGURE 3.14 – Visualisation EyeSeeX affichée sur un simulateur. (a) Visualisation de trois objets hors-champ sans réduction ou compression de l'information. (b) Stratégie 1 : réduction de l'information par élimination des objets situés derrière l'utilisateur. (c) Stratégie 2 : compression de l'information. Image issue de [45].

la vue détaillée (voir Figure 3.14) : une première stratégie est de diminuer l'espace hors-champ à considérer par la visualisation. En passant de 360 à 180 degrés, les POIs placés derrière l'utilisateur sont exclus et le nombre de POIs hors-champ à prendre en compte est réduit. Une deuxième stratégie est de compresser l'information (voir Figure 3.14b). Dans ce cas, deux options : la visualisation des POIs hors-champ est compressée à l'aide d'une fonction, comme la racine carrée. Ainsi, les informations proches du bord interne sont élargies, tandis que les informations proches du bord externe sont davantage compressées (voir Figure 3.14c). Une deuxième option est de compresser directement les axes (X ou Y). A titre d'exemple, si cette compression est appliquée à l'axe des Y, la technique supprimera toute information de hauteur des POIs. Les POIs hors-champ seront donc visualisés sur une seule ligne.

3.4.2 Indices partiellement visibles à l'écran

3.4.2.1 Visualisation d'objets hors-champ en 2D

Halo [13] et Wedge [59] sont deux techniques de visualisation hors-champ basées sur la théorie de la complétion amodale. Celle-ci souligne que le système visuel humain est capable de compléter des parties d'une forme partiellement visible [98].

La technique Halo [13] indique l'emplacement et la distance des POIs hors-champ à l'aide d'arcs 2D : la visualisation calcule un cercle dont le centre est le POI distant et le rayon est la distance entre celui-ci et le bord de l'écran. Seule une partie du cercle apparaît à l'écran pour éviter toute surcharge visuelle (voir Figure 3.15a). L'utilisateur complète mentalement la figure à partir de l'arc visible pour comprendre l'emplacement du POI hors-champ (perception amodale). Le halo a été comparé aux flèches dont la taille change en fonction de la distance du POI hors-champ [27]. L'étude demandait aux utilisateurs de localiser et de comparer des POIs hors-champ sur une carte 2D. Les résultats ont montré que les utilisateurs identifiaient plus efficacement la position du POI hors-champ avec Halo qu'avec les flèches. Comme indiqué par Baudisch et. al [13] pour les cibles statiques et par Gustafson et. al [58] pour les cibles en mouvement, Halo atteint ses limites dans les environnements denses en POIs : les arcs se chevauchent et créent un encombrement

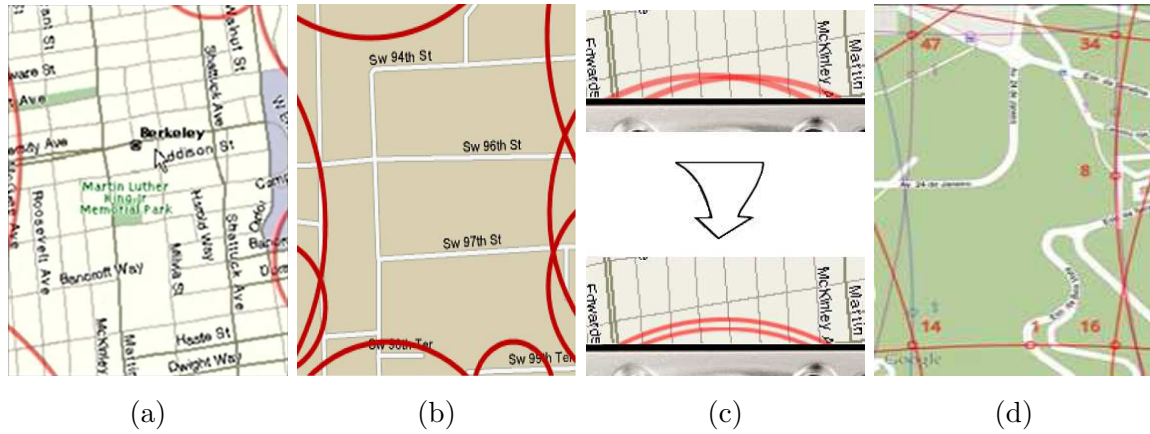


FIGURE 3.15 – a) Halos indiquant l'emplacement de 5 POIs hors-champ (Image issue de [13]). b) Chevauchements entre halos pour 8 POIs hors-champ (Image issue de [27]). c) Double arc agrégeant des halos qui se chevauchent (Image issue de [13]). d) Visualisation haloDot (Image issue de [43]).

visuel important (voir Figure 3.15b).

Pour réduire la surcharge visuelle à l'écran, une solution consiste à agréger les halos qui se chevauchent. Baudish et al. ont suggéré que les arcs qui se chevauchent fortement soient fusionnés en un seul multi-arc [13]. Le multi-arc est constitué d'arcs concentriques plus minces centrés sur le barycentre des POI agrégés. Lorsque le groupe dépasse 3 POIs, le multi-arc affiche un double anneau épais à l'écran (voir Figure 3.15c). Bien que cette approche réduise légèrement le nombre de halos qui se recouvrent à l'écran, le mécanisme présente deux inconvénients : (1) il n'empêche pas tous les halos de se chevaucher car il s'applique uniquement aux halos qui se recouvrent fortement et (2) il augmente l'intrusion visuelle des halos agrégés à l'écran (double arcs).

La technique HaloDot [43] adopte une autre approche : une grille invisible est superposée sur l'espace hors-champ pour la diviser en cellules. Deux POIs distants sont agrégés s'ils sont contenus au sein de la même cellule. HaloDot calcule un halo pour chaque cellule contenant des POIs et affiche le nombre de POIs agrégés sur l'arc correspondant. Cependant, la technique HaloDot (1) ne fournit aucune information sur la distribution spatiale des POIs agrégés d'une cellule et (2) n'empêche pas les halos de se chevaucher : si deux POIs sont proches l'un de l'autre mais situés dans des cellules différentes, les halos se chevaucheront à l'écran.

Enfin, pour réduire la surcharge visuelle à l'écran, le filtrage des POIs constitue une approche complémentaire. Baudish et al. [13] ont proposé d'afficher uniquement les POIs hors-champ dont la distance par rapport au bord de l'écran est inférieure à un seuil prédéfini. Dans tous les cas, les utilisateurs perdent partiellement conscience des POIs environnants.

La technique de visualisation Wedge [59] traite le problème posé par les environnements



FIGURE 3.16 – a) Wedges indiquant l'emplacement de 5 POIs hors-champ (Image issue de [59]). b) Chevauchements entre wedges pour 30 POIs hors-champ (Image issue de [28]).

denses. Wedge affiche des triangles au lieu de cercles comme avec Halo (voir Figure 3.16a). L'utilisateur prolonge mentalement les segments du triangle à l'écran pour estimer leur intersection, et l'emplacement du POI hors-champ. La technique fournit des indications de distance grâce à l'ouverture entre les segments et à l'intrusion dans l'espace visible : plus l'objet est éloigné, plus l'intrusion dans l'espace visible est grande et l'ouverture du triangle est large. Enfin, Wedge évite le chevauchement entre les triangles en les faisant pivoter. Cependant, le principal inconvénient de Wedge reste l'intrusion à l'écran. Les utilisateurs ont également préféré Halo lors d'une recherche des POIs les plus proches de leur position : la différence visuelle entre les POIs proches et éloignés était plus facile à percevoir avec Halo, de par la différence de taille des arcs de cercle. De plus, Wedge n'a été testé qu'avec 5 POIs regroupés dans une petite zone. Burigat et. al [28] ont montré qu'en augmentant le nombre de POIs jusqu'à 30, Wedge n'était plus assez robuste pour éviter les chevauchements (voir Figure 3.16b).

Pour simplifier la visualisation, deux optimisations ont été ajoutées à Wedge [28] : (1) réduction de la visibilité des POIs éloignés, en jouant sur la transparence du trait. Ce mécanisme est basé sur l'hypothèse que les POIs éloignés sont moins importants pour l'utilisateur dans beaucoup de scénarios du monde réel, (2) suppression des Wedges pour les POIs dont la distance est supérieure à un seuil. L'environnement hors-champ est donc tronqué. Pour une recherche de POIs dans des environnements denses (40 POIs) et sur supports mobiles, Wedge obtient des performances moindres que les interfaces Overview+Detail [28]. Pour cette comparaison, il était demandé aux utilisateurs de déplacer la carte et non le dispositif mobile pour explorer l'environnement hors-champ. En explorant la carte, l'apparence des triangles Wedge (taille, ouverture, etc.) est modifiée, tandis que celle des icônes représentant les POIs hors-champ dans l'Overview n'est pas modifiée. Ces observations suggèrent que Wedge requiert un effort mental plus important,

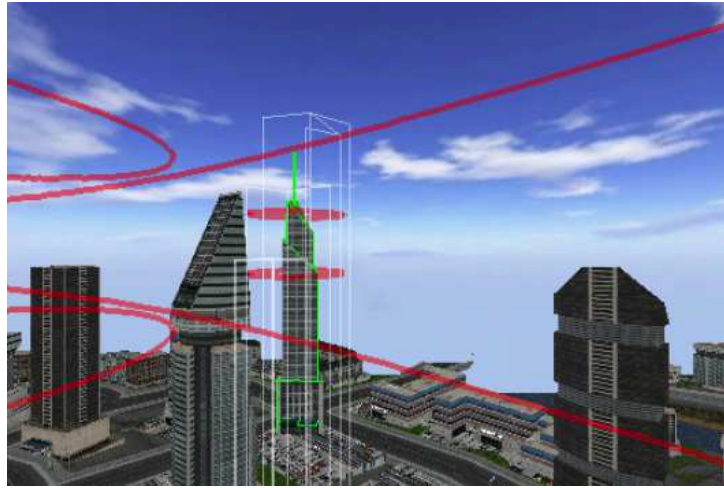


FIGURE 3.17 – 3D halos indiquant la position de 3 POIs : le premier situé sur le côté gauche de l’espace hors-champ et placé devant la caméra, le deuxième situé sur le côté gauche de l’espace hors-champ et placé derrière la caméra, le troisième situé derrière le building mis en évidence en vert. Image issue de [105].

car les triangles doivent être ré-interprétés après chaque action sur la carte. Toutefois, ces résultats ont été obtenus en comparant Wedge à une interface Overview+Detail dont les deux fenêtres affichent une vue orthographique. En RA, la fenêtre principale (Detail) est une vue égocentrique, capturée par la caméra de l’appareil mobile. L’utilisation d’une interface Overview+Detail en RA impliquerait donc un effort mental additionnel pour se réorienter lorsque l’utilisateur passe d’une vue égocentrique (Detail) à une vue orthographique (Overview).

3.4.2.2 Visualisation d’objets hors-champ en 3D/RA

La technique 3D Halo de Trapp et al. [105] est une adaptation de Halo [13] pour les environnements de réalité virtuelle 3D. Le cercle est placé en perspective dans la scène (voir Figure 3.17) et son rayon indique la distance physique à vol d’oiseau entre le POI et l’utilisateur. Cependant, plus un POI hors-champ est éloigné, plus l’intrusion du cercle à l’écran est grande. Il devient alors difficile de compléter mentalement le cercle et d’estimer l’emplacement du POI.

Plus récemment, Gruenefeld et al. [52] ont adapté les techniques Halo [13], Wedge [59] ainsi que les flèches [27] aux environnements 3D et RA à l’aide d’un casque (head-mounted device/HMD) (voir Figure 3.18). L’adaptation de ces techniques à la RA ou à la réalité virtuelle se fait d’une part, en projetant les POIs hors-champ sur une sphère imaginaire autour de la tête de l’utilisateur, puis en projetant les techniques sur la surface intérieure de cette sphère (voir Figure 3.18a). Comparée aux flèches, la technique HMD-Halo s’est révélée plus performante en termes de temps de complétion des tâches et de taux d’erreur, alors que la technique HMD-Wedge a subjectivement été perçue comme la meilleure. Cependant, les trois techniques ont été comparées dans un environnement comportant moins de 10 POIs. De plus, les POIs ont tous été placés dans un champ de vision de



FIGURE 3.18 – (a) Projection de la technique Halo sur une sphere autour de la tête de l'utilisateur (image issue de [46]). (b) Visualisation Halo, Wedge et flèches dans un casque de RA (image issue de [52]).

90 degrés. Gruenefeld et al. ont également comparé les performances des techniques de visualisation hors-champ Halo et Wedge dans les environnements RA avec celles dans des environnements de réalité virtuelle (RV) en utilisant un HMD [46]. Mais là encore, les études ne portaient que sur 8 POIs et aucun mécanisme n'a été proposé pour gérer un grand nombre de POIs hors-champ.

3.4.3 Synthèse des vues contextuelles

Les vues contextuelles sont efficaces pour donner des informations sur l'environnement hors-champ sans réduire la taille de la vue détaillée, ni la cacher. En revanche, les indices visuels sur les POIs hors-champ doivent avoir une intrusion limitée à l'écran, pour ne pas perturber l'attention de l'utilisateur. De plus, la gestion d'une densité élevée de POIs hors-champ ou de POIs regroupés en clusters est importante afin de ne pas surcharger l'interface graphique. Le respect de ces deux critères garantit des visualisations hors-champ non intrusives et adaptées aux environnements de RA mobile (voir section 3.2). Cependant, ces critères ne sont pas complètement remplis par les techniques de visualisation existantes. Le Tableau 3.2 résume notre revue des techniques existantes selon ces deux critères.

La littérature n'a révélé aucune visualisation hors-champ à la fois adaptée à la RA, garantissant une intrusion visuelle limitée et capable de gérer un nombre important de POIs hors-champ ou répartis en clusters. Cependant, AroundPlot [65], EyeSeeX [45] et HaloDot [43] semblent des techniques prometteuses : AroundPlot est la seule visualisation testée en RA avec un nombre important de POIs. La magnification dynamique de la vue globale permet d'espacer les indices visuels en présence d'une densité élevée de POIs ou de POIs répartis en clusters. Mais cette magnification n'est présente que lorsque l'utilisateur est en mouvement. AroundPlot possède les mêmes inconvénients que EdgeRadar [58] lorsque l'utilisateur reste statique. EyeSeeX limite l'intrusion à l'écran et évite la surcharge de l'interface graphique en limitant le nombre de POIs hors-champ considérés par la visualisation. Cependant, en ne considérant qu'une partie des POIs, la connaissance de l'utilisateur sur l'environnement hors-champ est incomplète et tronquée. Enfin, Halo-

-	Technique	Intrusion limitée	Gestion de densité élevée/ POIs regroupés en clusters
2D	Flèches 2D [27]	OUI	NON
	EdgeRadar [58]	OUI	NON
	EdgeSplit [58]	OUI	NON
	Halo [13]	OUI	NON
	Wedge [59]	OUI	NON
	HaloDot [43]	OUI	PARTIELLE
RV	Flèches 3D [34, 65]	OUI	NON
	3D Halo [105]	NON	NON
	HaloVR [52]	OUI	NON
	WedgeVR [52]	OUI	NON
RA	AroundPlot [65]	PARTIELLE	PARTIELLE
	HaloAR [52]	OUI	NON
	WedgeAR [52]	OUI	NON
	EyeSee [47]	OUI	NON
	EyeSee360 [48]	OUI	NON
	EyeSeeX [45]	OUI	PARTIELLE
	FlyingARrow [49]	NON	NON

TABLE 3.2 – Synthèse : techniques de visualisation hors-champ basées sur des vues contextuelles selon nos deux critères (voir section 3.2)

Dot offre une intrusion limitée à l'écran et gère une densité élevée de POIs ou des POIs répartis en clusters à l'aide d'un mécanisme d'agrégation. Cependant, la visualisation ne fournit aucune information sur la distribution spatiale des POIs agrégés et n'empêche pas tous les halos de se chevaucher. Toutefois, la technique Halo [13] est facile à comprendre [59] et indique direction et distance des POIs hors-champ en minimisant l'espace utilisé sur l'écran. Dans le chapitre suivant, nous nous basons donc sur Halo pour concevoir des techniques de visualisation hors-champ adaptées en RA et respectant les deux critères énoncés en section 3.2.

3.5 Conclusion

Nous avons présenté quatre approches pour étendre la connaissance spatiale de l'environnement, en apportant une vue globale à la vue détaillée de l'utilisateur : les techniques Zoom, Overview+Detail, Focus+Context et les vues contextuelles. Mais lorsque la taille de l'écran est limitée, comme celui des appareils mobiles (téléphones, tablettes, etc.), il convient (1) de garder une intrusion limitée à l'écran et (2) d'éviter de surcharger l'interface graphique lorsque le nombre de POIs est important ou lorsque les POIs sont groupés en clusters.

Dans le respect du premier critère, il est impératif de ne pas déformer, ni masquer, ni réduire la vue détaillée à l'écran de taille réduite, pour ne pas perturber la tâche principale de l'utilisateur. Nous avons ainsi choisi d'étudier les vues contextuelles, car elles n'impliquent aucune distorsion à l'écran et cachent un minimum d'informations de la vue détaillée.

Nous avons proposé une revue des vues contextuelles existantes sur écrans mobiles, en réalité virtuelle ainsi qu'en RA. Parmi celles-ci, Halo [13] indique direction et distance des POIs hors-champ en minimisant l'espace utilisé sur l'écran. Nous retiendrons aussi AroundPlot [65] et les flèches 2D [27], techniques auxquelles nous faisons référence dans la suite de ce manuscrit.

Nous proposons dans le chapitre suivant trois techniques de visualisation hors-champ RA basées sur Halo [13], dont l'intrusion à l'écran est limitée, permettant autant de gérer une forte densité de POIs que des POIs regroupés en clusters, mais aussi d'afficher la répartition spatiale des POIs.

4

Conception de techniques de visualisation hors-champ pour la réalité augmentée mobile

Sommaire

4.1	Introduction	58
4.2	Adaptation de Halo pour la Réalité Augmentée	59
4.2.1	Projection	59
4.2.2	Intrusion visuelle constante	60
4.2.3	Gestion de la profondeur	61
4.2.4	POIs devant/derrière l'utilisateur	67
4.3	Agrégation de POIs hors-champ	68
4.3.1	Halo3D : Barycentre	68
4.3.2	Halo3D* : Chevauchements autorisés	71
4.3.3	Halo3D** : Courbes	73
4.4	Conclusion	74

4.1 Introduction

Dans ce chapitre, nous présentons trois nouvelles techniques de visualisation hors-champ pour la réalité augmentée (RA) mobile. Comme motivé au regard de l'état de l'art dans le chapitre précédent, les techniques proposées reposent sur une vue contextuelle qui n'implique aucune distorsion à l'écran, et minimise l'intrusion à l'écran. Les trois techniques exploitent le principe de Halo [13] pour indiquer direction et distance des points d'intérêts (POIs) hors-champ. Nous structurons ce chapitre selon les deux critères établis au chapitre précédent section 3.2 : (1) minimiser l'intrusion visuelle, et (2) gérer les environnements denses en POIs. Ainsi la section section 4.2 traite du premier critère en expliquant les méthodes employées pour adapter Halo à la RA mobile et la section section 4.3 traite du deuxième critère en détaillant les solutions conçues pour l'agrégation dans le cas de nombreux POIs hors-champ.

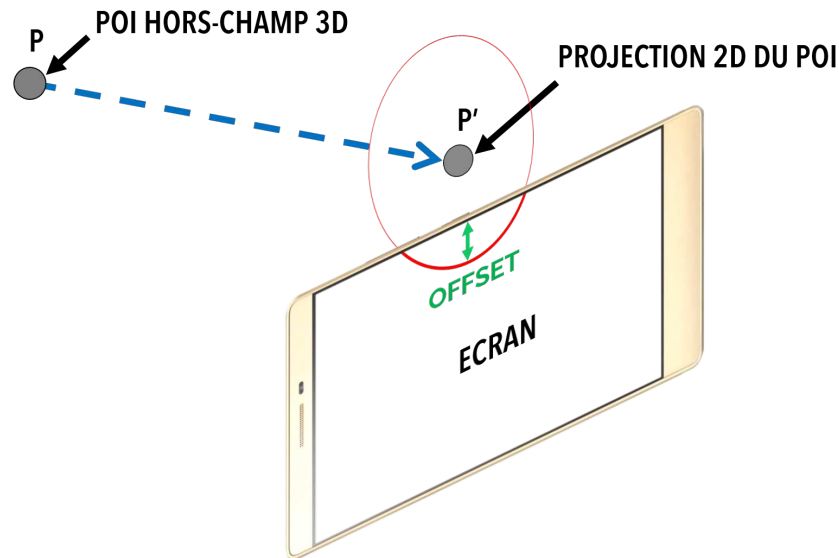


FIGURE 4.1 – Projection d’un POI hors-champ sur le plan de l’écran et calcul du halo correspondant

4.2 Adaptation de Halo pour la Réalité Augmentée

Halo [13] est une technique permettant de visualiser la localisation et la distance des POIs hors-champ sur un plan orthographique 2D. Ces informations sont affichées en garantissant une intrusion visuelle constante sur les bords de l’écran. Nous détaillons dans cette section l’adaptation de la technique Halo [13] à la RA mobile.

4.2.1 Projection

En RA mobile, les utilisateurs évoluent au sein d’une vue égocentrique 3D. Pour adapter Halo [13] à un environnement 3D, nous projetons le POI distant sur le plan de l’écran de l’appareil mobile. Nous déterminons ensuite le rayon du cercle en calculant la distance qui sépare la projection P' du POI du bord de l’écran le plus proche. A cette distance nous rajoutons un offset constant pour que le cercle soit partiellement visible sur l’écran (voir Figure 4.1). Nous avons fixé cet offset à 100 pixels de façon empirique pour un écran de résolution 2560x1440 pixels (dispositif Lenovo Phab 2 Pro décrit dans le chapitre 2). Pour le cas plus général, cet offset pourrait être proportionnel à la résolution de l’écran.

Ainsi, lorsqu’un POI initialement dans le champ visuel de la caméra disparaît dans l’espace hors-champ, un arc de cercle apparaît sur le bord de l’écran le plus proche de la projection P' du POI. Cet arc s’agrandit à mesure que la projection P' s’éloigne du bord de l’écran (voir Figure 4.2). En RA, l’arc de cercle représente donc l’amplitude de mouvement nécessaire pour ramener un POI distant dans le champ de vision de la caméra et donc à l’écran : plus le cercle est grand et plus le mouvement nécessaire pour ramener le POI dans le champ de vision sera important.

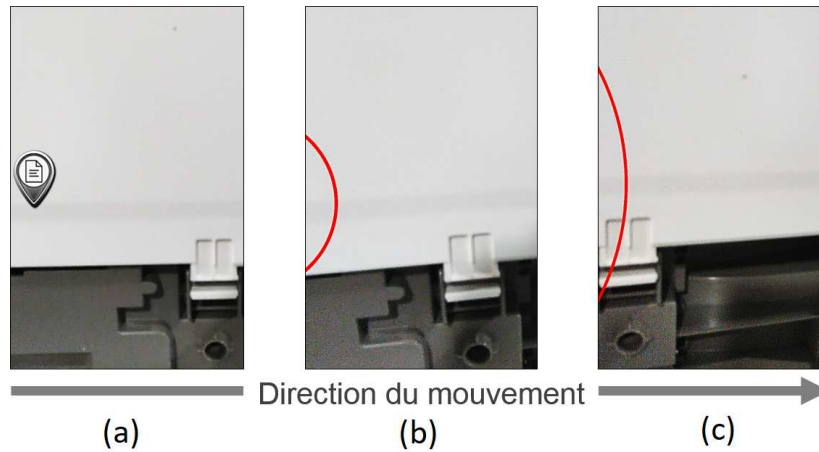


FIGURE 4.2 – Dynamisme du halo lors d'un mouvement de gauche à droite (a) un POI dans le champ de vision de la caméra. (b) Après un déplacement de gauche à droite de la caméra, le POI est hors-champ et est indiqué par un arc de cercle. (c) L'arc de cercle est agrandi car la distance entre la projection du POI et le bord de l'écran a augmenté.

4.2.2 Intrusion visuelle constante

L'avantage principal de Halo [13] est l'intrusion visuelle constante sur chacun des bords de l'écran, garantissant de ne pas impacter la zone centrale de l'écran (focus) dédiée à la tâche principale de l'utilisateur. Le rayon du cercle détermine donc l'intrusion visuelle du halo à l'écran. Nous détaillons la conception de la méthode de calcul du rayon d'un halo en trois étapes.

4.2.2.1 Méthode 1

Une première approche pour calculer le rayon du cercle d'un halo est de tracer une droite passant par la projection P' jusqu'au centre de l'écran C . A partir de l'intersection entre cette droite et le bord de l'écran le plus proche de P' , il suffit de rajouter un offset constant le long du vecteur $\vec{P'C}$ pour que le halo devienne visible à l'écran (voir Figure 4.3a). Cette méthode ne garantit cependant pas une intrusion visuelle constante sur l'écran : à mesure que le cercle se déplace vers les coins de l'écran, son rayon augmente : ainsi à la Figure 4.3a, R_2 est supérieur à R_1 . L'exigence d'intrusion visuelle constante n'est plus respectée. Ce phénomène est très perceptible pour un grand nombre de POIs placés autour de l'utilisateur sans agrégation (voir Figure 4.3b). L'intrusion des halos est plus grande dans les coins de l'écran.

4.2.2.2 Méthode 2

Pour garantir une intrusion visuelle constante, nous considérons un rectangle R dont les bords se situent à une distance $OFFSET$ des bords de l'écran (voir Figure 4.4a). L'intersection entre P' et le point du rectangle le plus proche de P' garantit une intrusion visuelle constante du halo, quelle que soit sa position sur l'écran : deux projections de POIs hors-champ alignées horizontalement (ou verticalement) possèdent maintenant le même rayon. Pour un grand nombre de POIs non agrégés, l'affichage de leurs halos laisse

deviner le rectangle interne R , et donc l'offset constant sur chacun des bords de l'écran (voir Figure 4.4b).

En revanche, pour des projections de POIs situés dans les coins de l'espace 2D hors-champ (P1 et P2 sur la Figure 4.4a), le coin du rectangle R est le point le plus proche d'eux. Plusieurs projections de POIs peuvent se trouver dans cet espace, et seraient donc représentés par le même halo. Ceci impliquerait des halos cloisonnés dans le coin de l'écran tant que les projections de POIs restent dans cet espace 2D hors-champ.

4.2.2.3 Méthode 3

Nous résolvons l'inconvénient de la méthode 2 en considérant un rectangle intérieur à bords arrondis (voir Figure 4.5a). Cette forme est composée de quatre cercles de même rayon placés aux coins du rectangle. Lorsqu'une projection P' de POI se trouve dans le coin de l'espace 2D hors-champ, nous déterminons le point du cercle le plus proche de P' . Cette méthode permet de calculer à chaque projection de POI un halo distinct. Pour un grand nombre de POIs non agrégés, l'affichage de leur halos permet de percevoir le caractère arrondi des coins du rectangle interne (voir Figure 4.5b). Comme pour l'offset (section 4.2.1), le rayon des cercles du rectangle interne peut être calculé par rapport à un pourcentage de la largeur totale de l'écran (en pixels). La Figure 4.6 montre une visualisation des halos pour différents pourcentages de la largeur de l'écran.

Pour la conception finale de notre technique, nous avons fixé de façon empirique un rayon de cercle égal à 5% de la largeur de l'écran (donc un rayon de 128 pixels de large pour une taille d'écran de 2560 pixels de large). Cela permet de garder une intrusion visuelle minimale à l'écran, tout en permettant de différencier les halos des POIs situés dans les coins de l'espace 2D hors-champ.

4.2.3 Gestion de la profondeur

En projetant les POIs sur le plan de l'appareil mobile, les trois dimensions de la position d'un POI sont réduites à deux dimensions. Il y a donc une première perte d'information de profondeur ou de distance physique qui sépare les POIs de l'utilisateur. Nous décrivons ensuite (section 4.2.4) une deuxième perte d'information, le positionnement du POI devant ou derrière l'utilisateur.

4.2.3.1 Traits visuels

La profondeur peut être traduite visuellement en exploitant les caractéristiques visuelles du trait du cercle du halo, comme :

- la transparence (voir Figure 4.7)
- la couleur
- l'épaisseur

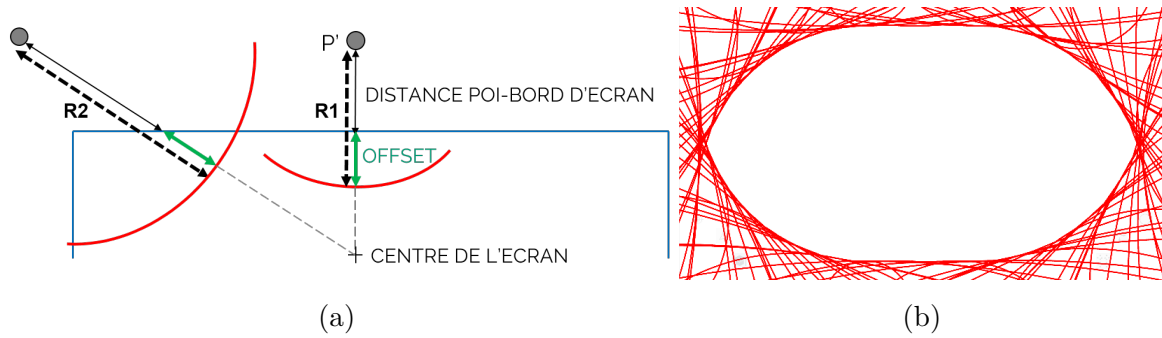


FIGURE 4.3 – a) Calcul du rayon des halos à partir de la droite passant par P' jusqu'au centre de l'écran. b) Visualisation des halos de 100 POIs sans agrégation : l'intrusion visuelle des halos est plus grande dans les coins de l'écran.

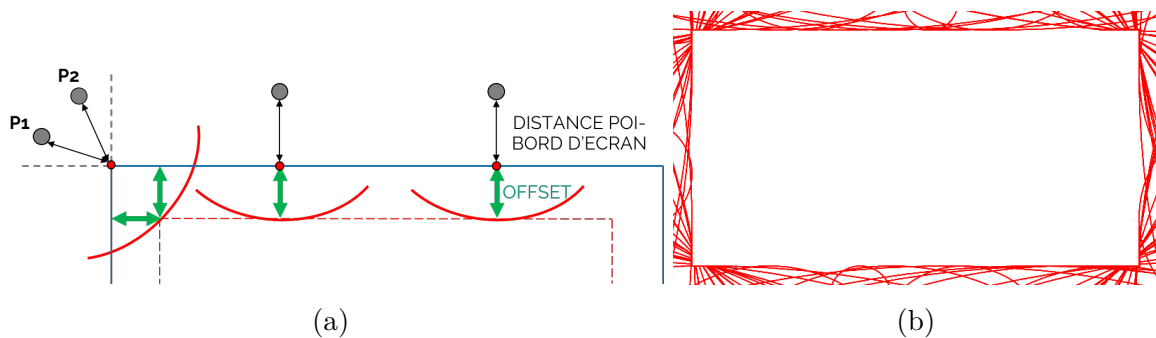


FIGURE 4.4 – a) Calcul du rayon des halos en considérant un rectangle interne. b) Visualisation des halos de 100 POIs sans agrégation : le rectangle interne devient perceptible.

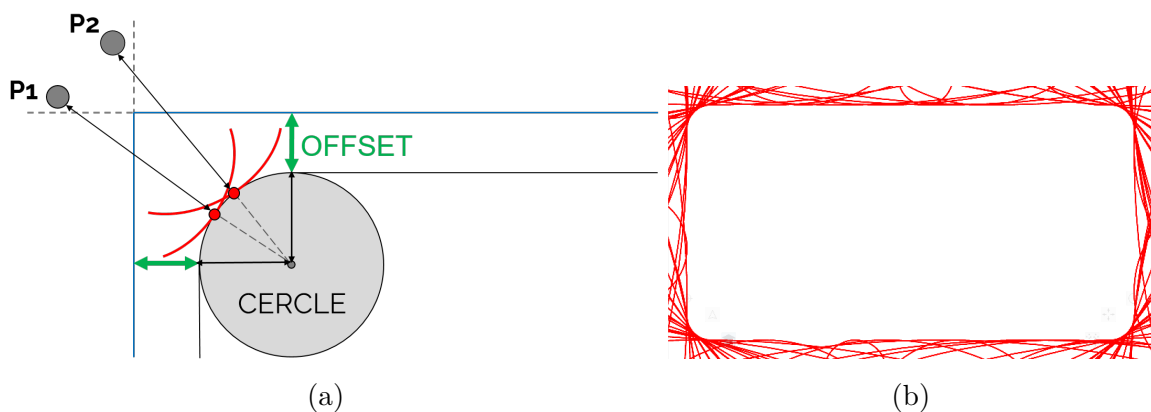


FIGURE 4.5 – a) Calcul du rayon des halos en considérant un rectangle interne à bords arrondis. b) Visualisation des halos de 100 POIs sans agrégation : le rectangle interne à bords arrondis devient perceptible.

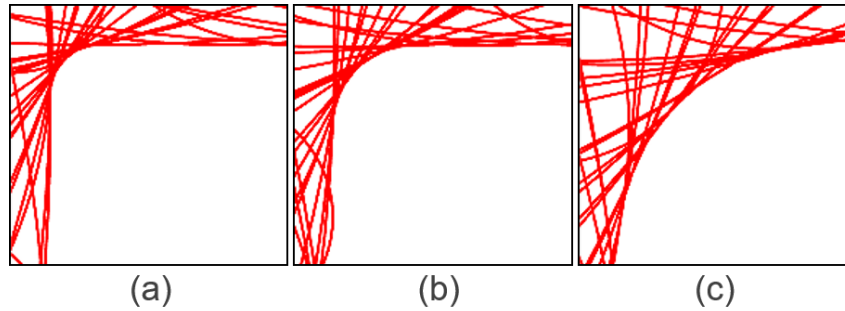


FIGURE 4.6 – Exemples d'intrusion de halos dans le coin d'un écran de 2560 pixels de largeur en utilisant la méthode du rectangle à bords arrondis, dont les cercles possèdent un rayon égal à (a) 5% de la largeur de l'écran, (b) 10% de la largeur de l'écran, (c) 20% de la largeur de l'écran.

D'autres traits visuels ont également été explorés : la continuité du trait du cercle (pointillés, hachures) et le caractère multicolore du trait. Par exemple, plus un POI serait proche de l'utilisateur, et plus le halo correspondant deviendrait hachuré/multicolore. En s'éloignant du POI, l'arc reprendrait son aspect original (sans coupures) / de couleur uniforme. Cependant, ces solutions augmentent considérablement la variabilité visuelle du halo. En RA, l'environnement est déjà dense en couleurs et en formes. Un contexte industriel est d'autant plus chargé qu'il y a de machines, câbles, capteurs dans le champ de vision de la caméra. L'ajout d'une variabilité visuelle supplémentaire à chaque halo augmenterait donc la difficulté à percevoir les emplacements des points d'intérêt hors-champ. Nous considérons donc trois caractéristiques visuelles, à savoir la transparence, la couleur et l'épaisseur, car elles préservent l'uniformité visuelle du halo affiché à l'écran.

Au-delà du trait visuel, son évolution avec la distance POI-utilisateur implique un deuxième choix de conception. En considérant l'épaisseur de l'arc par exemple, il s'agit de définir si le trait visuel choisi doit être le plus visible (épais) lorsque le POI est loin ou lorsqu'il est proche de l'utilisateur. Nous considérons deux types de transitions : la transition atténuée et la transition abrupte. La transition atténuée rend le cercle de moins en moins visible à mesure que l'utilisateur se rapproche physiquement d'un POI (exemple : de plus en plus transparent, fin ou blanc). Au contraire la transition abrupte rend le cercle de plus en plus apparent (exemple : de plus en plus opaque, épais, ou rouge) à mesure que l'utilisateur se rapproche physiquement de la cible correspondante. La Figure 4.8 schématise ces deux transitions pour le trait visuel épaisseur, et pour un POI hors-champ.

Deux axes de conception sont donc à considérer pour déterminer la manière de visualiser la distance qui sépare un POI d'un utilisateur : (1) le trait visuel (transparence, couleur ou épaisseur) et (2) l'évolution de cette caractéristique visuelle selon la distance POI-utilisateur que nous désignons par transition visuelle. Le couple (caractéristique visuelle, transition visuelle) définit la technique visuelle permettant de représenter la profondeur.

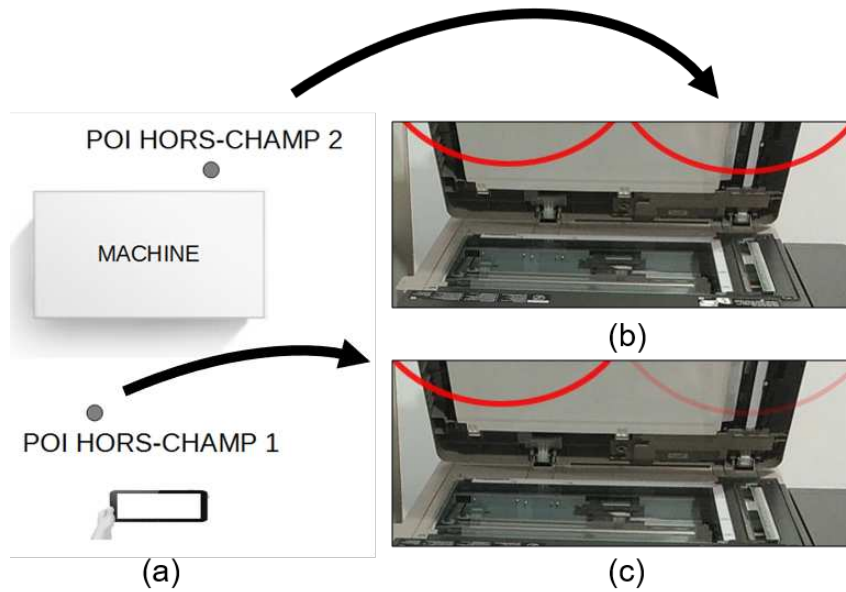


FIGURE 4.7 – Halos et profondeur (a) Vue orthographique par-dessus de 2 POIs hors-champ devant et derrière une machine. (b) Visualisation des 2 POIs : les deux halos ne permettent pas de connaître la différence de profondeur entre les deux POIs. (c) Visualisation des 2 POIs avec la profondeur traduite par la transparence du halo. Le halo le plus opaque à gauche indique que le POI correspondant est plus proche que le deuxième POI.

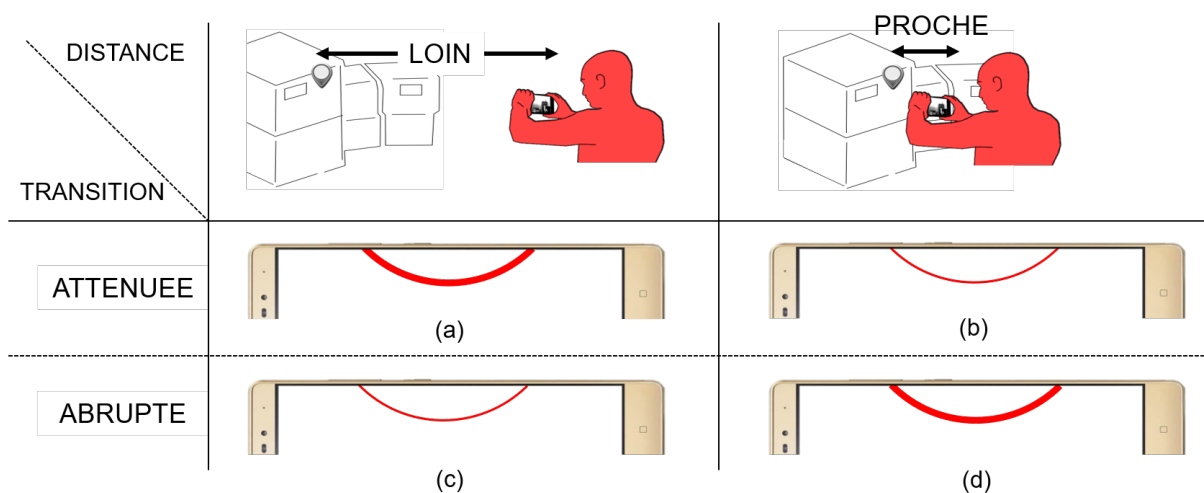


FIGURE 4.8 – Visualisation de la distance entre l'utilisateur et un POI hors-champ, obtenue en couplant le trait visuel *épaisseur* avec la transition *atténuée* (a-b) ou *abrupte* (c-d). Avec la transition *atténuée*, le halo devient de plus en plus fin à mesure que l'utilisateur se rapproche du POI hors-champ. Avec la transition *abrupte*, le halo devient de plus en plus épais à mesure que l'utilisateur se rapproche du POI hors-champ.

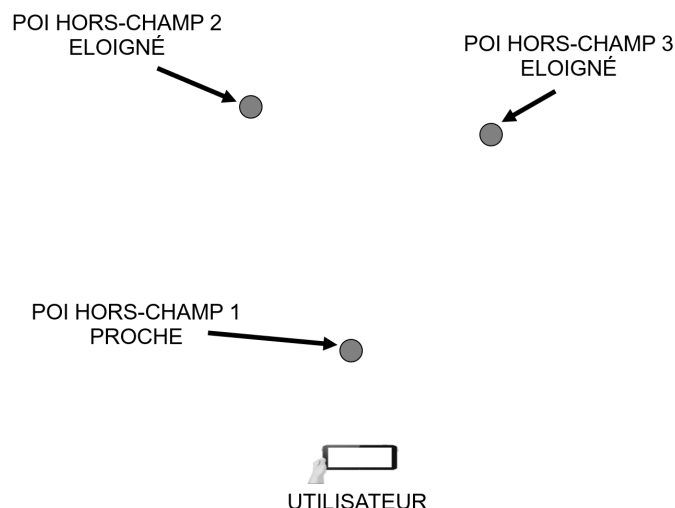


FIGURE 4.9 – Vue orthographique par-dessus de 3 POIs hors-champ. Un seul des trois POIs est significativement plus proche de l'utilisateur que des deux autres POIs.

4.2.3.2 Étude pilote

Pour figer ce paramètre de conception (caractéristique visuelle, transition visuelle), nous avons mené une étude pilote. Nous émettons les deux hypothèses suivantes : (1) la technique visuelle (caractéristique visuelle, transition visuelle) qui minimise la charge visuelle à l'écran sera préférée. (2) La technique visuelle (caractéristique visuelle, transition visuelle) qui rend plus visible les POIs proches de ceux loin de l'utilisateur sera préférée.

Tâche L'objectif de la tâche est de déterminer lequel des trois POIs placés devant l'utilisateur est le plus proche de sa position. Pour cela nous plaçons trois cibles de sorte qu'une des trois soit significativement plus proche du participant que les deux autres (voir Figure 4.9). Pendant toute l'étude expérimentale, l'utilisateur tient l'appareil mobile tel que les cibles ne soient pas visibles à l'écran. Ainsi, seuls les halos apparaissent à l'écran. Nous demandons au participant d'interpréter les informations visuelles du cercle pour déterminer oralement lequel des POIs est le plus proche.

Participants/Matériel 6 participants issus de l'Université ont participé à cette étude pilote. L'étude a été réalisée avec la tablette Apple iPad Air 2, et la SDK Vuforia, librairie de RA nécessaire pour reconnaître et suivre des objets 3D en temps réel. Les techniques ont été développées en C# et affichées à l'écran de la tablette à l'aide du moteur de rendu Unity3D.

Conditions Trois traits visuels sont ici considérés : la transparence, l'épaisseur et la couleur du cercle. Nous prenons également en compte les deux types de transitions visuelles : abrupte et atténuée. 6 conditions sont donc considérées et chaque condition est répétée 6 fois. Cela correspond à 36 tâches de choix du POI le plus proche (6 conditions x 6 itérations) pour chaque participant.

Procédure Le participant débute l'expérience par une première phase d'apprentissage

générale. Cette phase consiste à manipuler l'appareil mobile pour comprendre le dynamisme des halos, et la relation qui existe entre la taille de cercle et l'amplitude de mouvement à effectuer pour ramener le POI dans le champ de vision de la caméra. Le participant réalise ensuite la tâche demandée pour tous les attributs visuels d'un premier type de transition donnée. Après une session de débriefing pour noter les commentaires et les préférences du participant, celui-ci continue l'étude expérimentale avec les trois attributs visuels avec le deuxième type de transition.

Pour chaque tâche nous enregistrons le taux d'erreur ainsi que le temps pour accomplir la tâche (temps chronométré entre l'instant où une nouvelle configuration de POIs est affichée, et l'instant où le participant indique oralement sa réponse). Entre chacun des cas, le système modifie le POI le plus proche à déterminer.

Résultats Nous n'avons pas observé d'effets notables des différentes techniques visuelles autant sur le temps de complétion que le taux d'erreur (majoritairement nul). Nous confirmons ainsi les résultats de l'étude réalisée par Muller et al. sur le système de visualisation hors-champ Sparkle [81] : le participant est capable de s'adapter quelque soit l'encodage employé. Une fois la relation entre trait visuel et distance comprise, il est en mesure d'accomplir les tâches demandées.

Nous nous appuyons donc sur les résultats qualitatifs pour choisir une technique visuelle. Les commentaires recueillis auprès des 6 candidats indiquent que la technique visuelle préférée est définie par le couple [épaisseur-transition visuelle abrupte] :

- *Trait visuel épaisseur* : l'épaisseur et la transparence ont été largement préférées par rapport à la couleur. Tout d'abord, le trait visuel de la couleur est plus adapté pour traduire un type d'entité (différencier les POIs d'alerte des POIs de documentation par exemple) qu'un attribut scalaire comme la distance [112, p. 213]. De plus, l'épaisseur et la transparence permettent de modifier la saillance visuelle sur l'écran en fonction de la distance entre l'utilisateur et un POI. Il est par exemple possible d'obtenir un cercle plus fin ou semi-transparent, ce qui diminue la saillance visuelle. Ce changement de saillance n'est pas possible avec la couleur : que le cercle soit bleu, rouge ou vert, le cercle sera toujours assez visible sur l'écran. Les utilisateurs préfèrent donc une technique qui puisse modifier sa saillance en fonction de la distance utilisateur - POI. Enfin, l'épaisseur est préférée car le cercle reste tout à fait perceptible lorsque le POI est éloigné. En effet, les participants ont rapporté qu'il est plus facile de percevoir un trait fin qu'un trait plus ou moins transparent.
- *Transition visuelle abrupte* : les participants ont justifié leur préférence en effectuant un parallèle avec la perception du monde réel. Lorsqu'un objet est situé au loin, il est difficile de correctement percevoir tous les détails qui le composent. Plus l'utilisateur se rapproche, et plus il obtient une vision nette et précise de cet objet. Les participants préfèrent donc un système qui reproduit ce phénomène (sorte de

métaphore du monde réel), avec un cercle qui devient de plus en plus visible à mesure qu'ils se rapprochent d'un POI.

Pour gérer la profondeur, nous conservons donc la technique visuelle qui rend le cercle plus épais (resp. plus fin), à mesure que l'utilisateur se rapproche (resp. s'éloigne) du POI.

4.2.4 POIs devant/derrière l'utilisateur

Au-delà de la perte d'informations de distance en profondeur du POI par rapport à l'utilisateur, une autre information n'est plus représentée par la projection du POI sur le plan du dispositif mobile : le fait que le POI soit devant ou derrière l'utilisateur. En effet, qu'un POI se trouve devant ou derrière l'utilisateur, la projection du POI ne change pas de position, et le rayon du halo correspondant reste identique.

Une première solution pour différencier un POI hors-champ placé devant l'utilisateur d'un POI hors-champ placé derrière est d'utiliser un trait visuel supplémentaire, parmi ceux décrits dans la section 4.2.3. Cependant, l'utilisateur est déjà sollicité pour comprendre (1) l'emploi d'un cercle pour désigner un POI hors-champ, (2) la relation entre la taille de ce cercle et l'amplitude de mouvement nécessaire pour ramener le POI dans son champ de vision, (3) la profondeur désignée par un trait visuel et (4) la technique d'agrégation de halos (section 4.3). L'ajout d'un trait visuel supplémentaire pour indiquer si un POI se trouve derrière risque de rendre la visualisation trop complexe. De plus, il convient de noter que le système de visualisation hors-champ ne sera utilisé que ponctuellement : il ne constitue pas le centre d'attention principal d'un utilisateur, d'où son affichage en périphérie de l'écran. Si trop d'efforts sont requis pour comprendre la visualisation, l'utilisateur ne cherchera pas à s'en servir.

Nous préférons exploiter les propriétés existantes du cercle pour indiquer la position des POIs derrière l'utilisateur. Similaire à la technique proposée par Trapp [105], nous supposons qu'atteindre un POI placé derrière soi implique un mouvement de grande amplitude (par exemple, se retourner pour ramener le POI dans le champ de vision de la caméra). Plus l'amplitude du mouvement à réaliser pour atteindre le POI est importante, et plus le cercle s'agrandit jusqu'à être semblable à une ligne. Nous proposons donc d'exploiter cet aspect en affichant une ligne au lieu d'un cercle dès qu'un POI passe derrière l'utilisateur.

La ligne est en fait un cercle de très grande taille, et est affichée tant que le POI se trouve derrière l'utilisateur, quelque soit la position de la projection du POI par rapport au bord de l'écran.

Ainsi, déterminer si un POI se trouve derrière soi se fait de manière dynamique, en suivant le cercle. A titre d'exemple (voir Figure 4.10), une ligne placée sur le bord droit de l'écran indique à l'utilisateur qu'un POI hors-champ sera visible s'il effectue une rotation vers sa droite. La ligne reste visible tant que le POI se trouve dans l'espace hors-champ situé derrière l'utilisateur. A un point de la rotation, le POI passe dans l'espace hors-champ

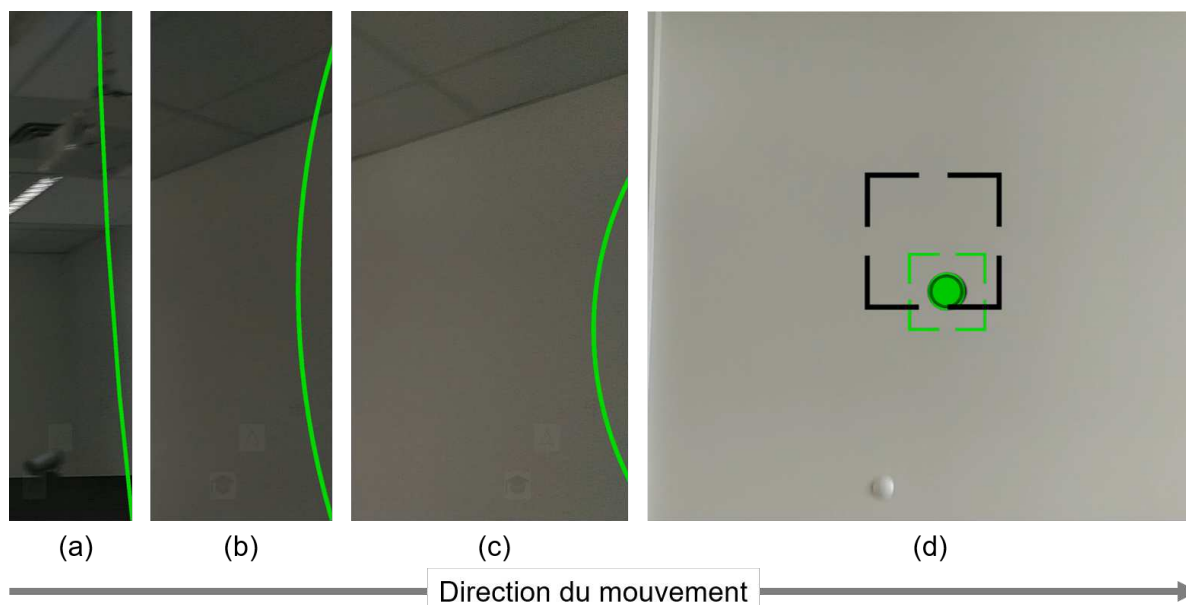


FIGURE 4.10 – Dynamisme du halo lors d’un mouvement pour atteindre un POI situé derrière l’utilisateur (a) le halo, affiché comme une ligne droite se trouve derrière l’utilisateur. (b-c) Déplacement de gauche à droite de la caméra, le POI se rapproche du champ de vision de l’utilisateur, et la taille du halo diminue. (d) Le POI est maintenant visible dans le champ de vision de l’utilisateur, le halo a disparu de l’écran.

situé devant l’utilisateur. A partir de cet instant, le dynamisme du cercle est ré-activé : le halo diminue de taille à mesure que le POI hors-champ se rapproche du champ visuel de l’utilisateur, jusqu’à ce que le POI devienne visible depuis la caméra.

C’est donc l’animation du cercle qui guide l’utilisateur jusqu’aux POIs hors-champ situés derrière lui.

4.3 Agrégation de POIs hors-champ

Avoir en permanence affichés sur l’écran les halos correspondant à tous les POIs hors-champ conduit à une surcharge visuelle de l’interface graphique. Dans le cas où les POIs sont nombreux (Figure 4.3b, Figure 4.4b, Figure 4.5b), ceci entraîne le chevauchement des nombreux halos. Pour éviter la surcharge visuelle due à un environnement dense en POIs (deuxième critère de la section 3.2), il convient d’agréger les halos. Nous proposons trois solutions/variantes pour décider quand agréger des POIs et comment afficher les POIs agrégés. Les trois techniques résultantes sont respectivement notées Halo3D, Halo3D* et Halo3D**.

4.3.1 Halo3D : Barycentre

Halo3D agrège des cercles dès qu’ils se chevauchent à l’écran. Le nouveau cercle affiché remplace ceux qui se chevauchent et est centré sur leur barycentre. Chaque halo d’agrégation comporte un numéro indiquant le nombre de POIs agrégés. Dans un souci

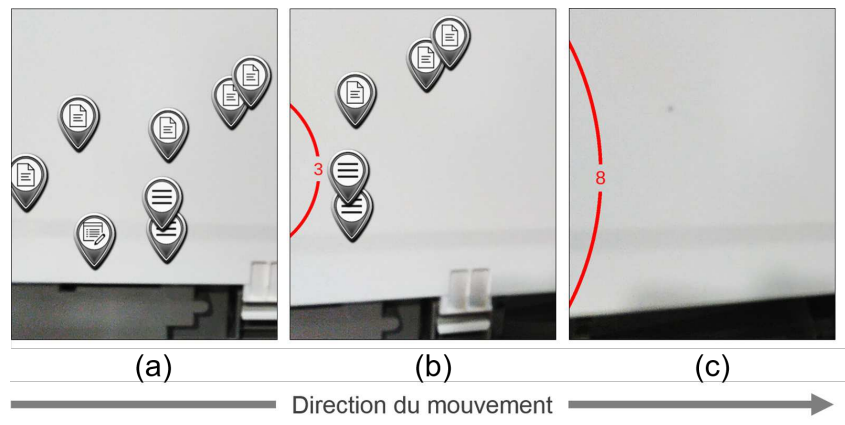


FIGURE 4.11 – Agrégation de 8 POIs (a) 8 POIs visibles sur l'écran. (b) L'utilisateur se déplace de gauche à droite. 3 halos s'agrègent. (c) Tous les POIs sont agrégés.

de minimisation de l'intrusion dans l'espace visible, ce numéro est affiché sur le cercle. L'arc inclut donc une petite ouverture pour permettre l'affichage du nombre de POIs agrégés (voir Figure 4.11). Cette ouverture est localisée à l'intersection entre le cercle et le segment défini par le centre du cercle et le centre de l'écran. Le nombre de POIs agrégés reste ainsi visible quelque soit la position du cercle sur l'écran.

Conserver un historique des cercles agrégés permet de restaurer les halos lors de la désagrégation. Nous utilisons pour cela une structure de données de type arbre, dont chaque nœud représente un halo. Les feuilles de l'arbre constituent donc les halos d'origine, correspondant à chaque POI hors-champ. Un nœud comportant au moins deux fils représente un halo d'agrégation, les fils étant des cercles qui se chevauchent. Chaque halo d'agrégation peut également chevaucher d'autres cercles, chacun représenté par son arbre à un ou plusieurs niveaux. L'algorithme d'agrégation est donc réitéré en ne comparant que la racine de chaque arbre. Le processus se termine lorsque plus aucun chevauchement n'est détecté.

Pour détecter tous les cas de désagrégation, l'algorithme descend dans un premier temps jusqu'aux nœuds dont tous les fils sont constitués de feuilles. Un nœud est constitué de feuilles s'il n'existe aucun POI d'agrégation parmi les POIs agrégés. Ce nœud doit faire l'objet d'une désagrégation si :

1. Un des POIs agrégés ne se chevauche plus avec l'un des autres POIs agrégés du nœud.
2. Un des POIs agrégés est rentré dans le champ de vision de l'utilisateur, son halo n'étant par conséquent plus affiché à l'écran.

Cette vérification est effectuée de parent en parent jusqu'à ce qu'une désagrégation à effectuer soit trouvée. L'algorithme réitère jusqu'à ce que plus aucune désagrégation ne soit détectée.

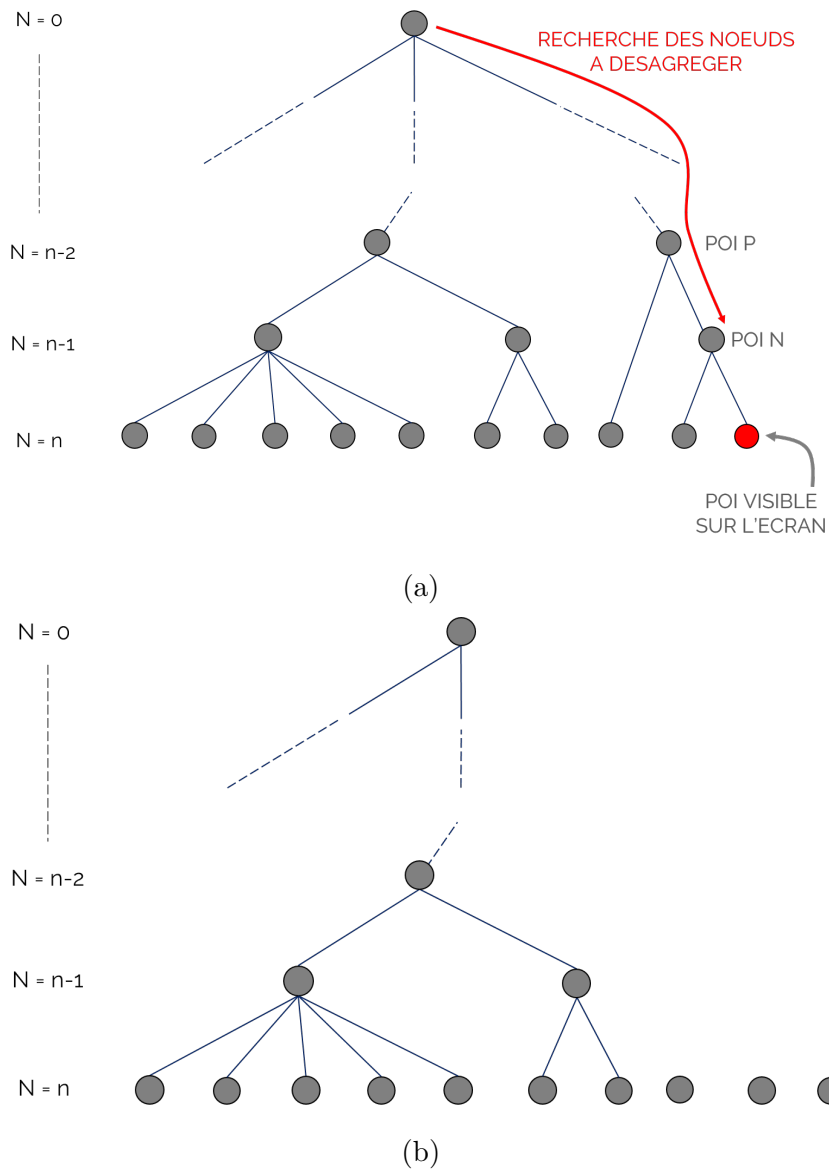


FIGURE 4.12 – Processus de désagrégation. a) Recherche des nœuds dont les halos fils ne se chevauchent plus ou n'existent plus. b) Élimination des parents des nœuds désagrégés.

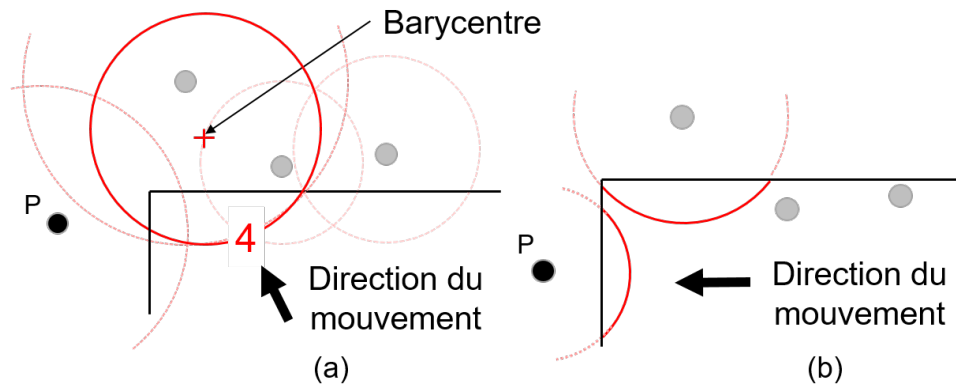


FIGURE 4.13 – Halo3D guide l'utilisateur vers le POI P en deux étapes : (a) Halo3D guide d'abord l'utilisateur vers le barycentre des POI agrégés (b) une fois que les POIs se sont désagrégés, Halo3D guide vers les POI P.

L'agrégation basée sur le barycentre permet donc d'éliminer tous les chevauchements entre les halos, et de rendre l'interface plus lisible. En revanche, cette technique peut induire un guidage indirect vers un POI. En effet, considérons un groupe de POIs dont le barycentre est situé dans la partie supérieure gauche de l'espace hors-champ (voir Figure 4.13). Pour atteindre le POI P situé en bas de ce cluster, Halo3D guidera l'utilisateur en deux étapes : tout d'abord vers le centre de gravité du cluster (voir Figure 4.13a), puis vers le POI P (voir Figure 4.13b). Ce guidage indirect est causé par la perte d'information sur la distribution spatiale des POIs lorsqu'il y a agrégation.

Pour minimiser ce problème, nous proposons deux variantes de Halo3D : Halo3D* et Halo3D**. Halo3D* réduit le nombre d'agrégations réalisées en autorisant le chevauchement des halos conformément à un seuil (taille maximale de chevauchements entre deux arcs), tandis que la technique Halo3D** modifie la façon dont les POIs agrégés sont affichés à l'écran.

4.3.2 Halo3D* : Chevauchements autorisés

Halo3D* repose sur l'hypothèse que de petits chevauchements entre halos ne perturbe pas la compréhension de l'environnement hors-champ par l'utilisateur (voir Figure 4.14b).

Après plusieurs essais, nous avons fixé un seuil de 70 pixels pour la taille maximale de chevauchement entre deux arcs. L'algorithme d'Halo3D qui détermine si deux cercles se chevauchent (voir section 4.3.1) prend en paramètres le centre et le rayon des cercles à analyser. Halo3D* réutilise cet algorithme mais l'applique avec des rayons diminués du seuil de 70 pixels. A l'écran, la taille des cercles reste inchangée mais de petits chevauchements sont maintenant visibles.

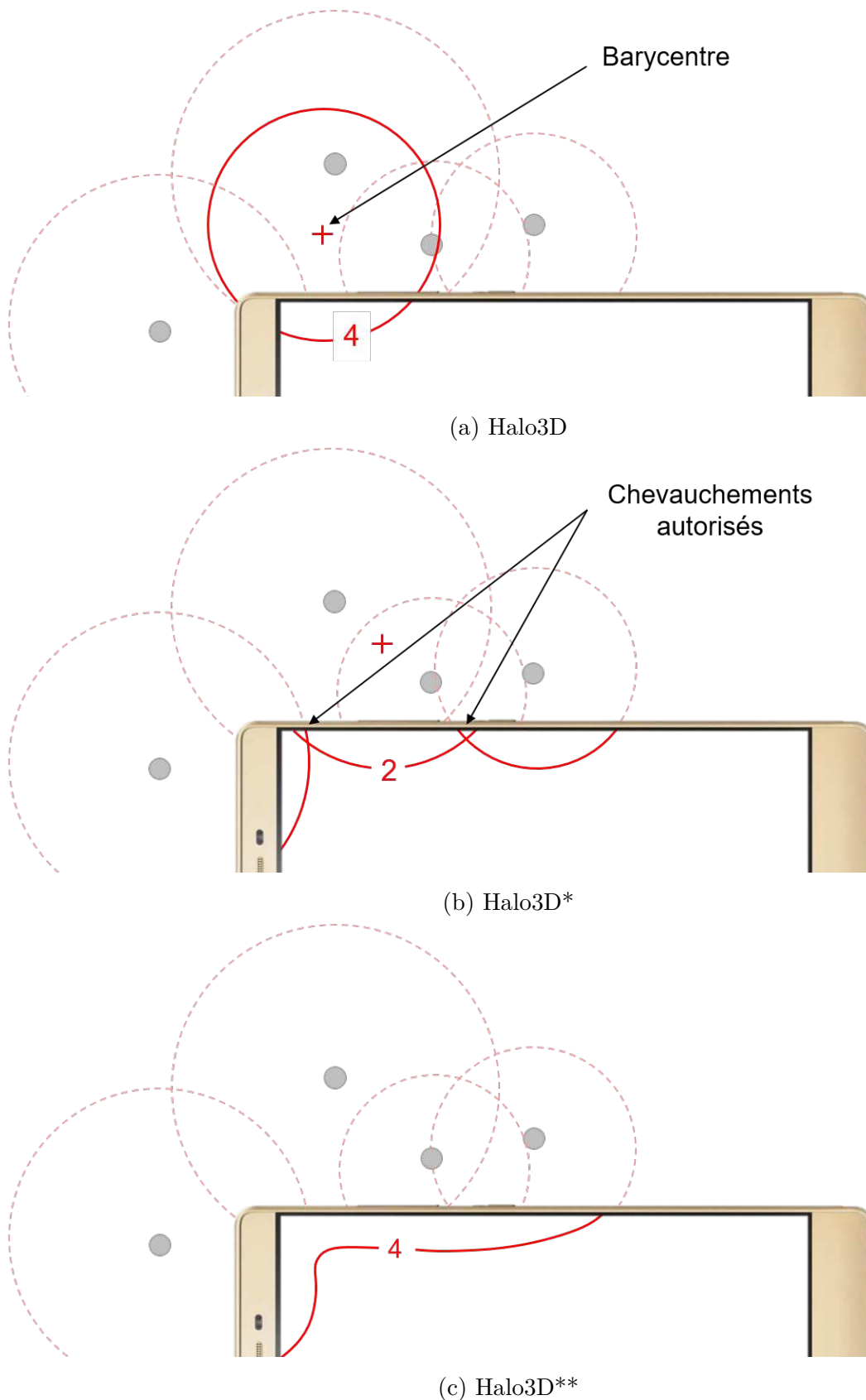


FIGURE 4.14 – Trois techniques avec agrégation. (a) Halo3D, affichant le halo du barycentre du cluster. (b) Halo3D*, chevauchements entre halos autorisés conformément à un seuil. (c) Halo3D**, POIs agrégés affichés à l'aide d'une courbe, révélant la distribution spatiale des POIs du cluster.

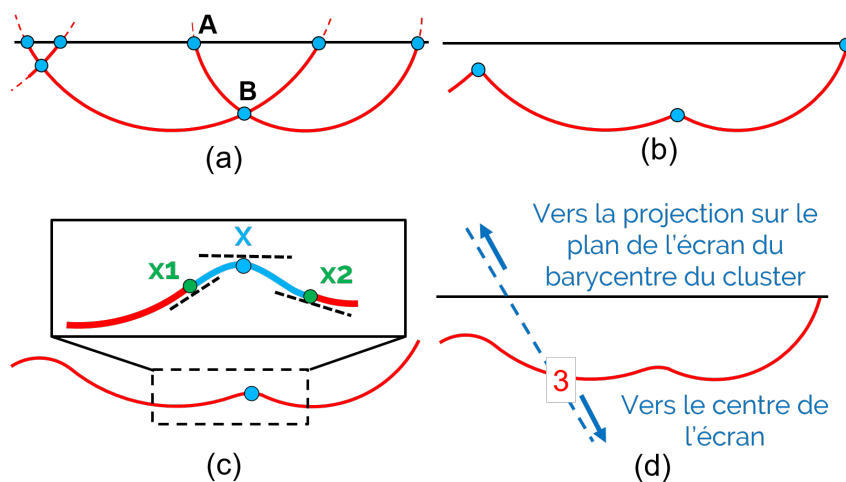


FIGURE 4.15 – (a) : Étapes 1 et 2 - Tous les halos et leurs intersections sont calculés. L’arc (AB) est contenu dans un halo et ne fera pas partie de la courbe finale. (b) : Étape 3 - Les arcs composant la courbe finale sont calculés. (c) : Étape 4 - Exemple d’une courbe quadratique reliant deux arcs, en utilisant trois points (x_1 , X , x_2) et trois tangentes (lignes en pointillés noirs). (d) : Étape 5 - Le nombre de points d’intérêt agrégés est affiché au sein de la courbe.

4.3.3 Halo3D** : Courbes

Halo3D** affiche la distribution spatiale des POIs hors-champ agrégés en affichant des courbes au lieu de cercles (voir Figure 4.14c).

Une courbe permet aux utilisateurs de visualiser l’étendue d’un cluster de POIs hors-champ et est calculée en 5 étapes : tout d’abord, le halo de tous les POIs hors-champ sont calculés selon le même principe que Halo3D. L’algorithme calcule ensuite les points d’intersection avec les bords de l’écran et entre les différents cercles (voir Figure 4.15a). Dans une troisième étape, l’algorithme détermine les arcs qui feront partie de la courbe finale (voir Figure 4.15b) : un arc est une section d’un cercle situé entre deux points d’intersection sur ce même cercle. Les arcs contenus dans un halo (par exemple l’arc (AB) à la Figure 4.15a) ne font pas partie de la courbe finale. Les intersections entre les arcs sont ensuite lissées. Pour ce faire, l’algorithme utilise des courbes quadratiques pour lier les arcs entre eux. Comme le montre la Figure 4.15c, le point d’intersection X entre deux arcs est l’un des points de la courbe quadratique qui lie les arcs entre eux. Deux autres points (x_1 , x_2) sont également calculés, un sur chaque arc se chevauchant, de sorte que leur distance à X soit égale à un tiers de la longueur de l’arc correspondant. En connaissant les coordonnées de chaque point et les tangentes à ces points, les courbes quadratiques peuvent facilement être calculées. L’algorithme affiche donc une courbe lisse à l’écran qui représente la distribution spatiale des POI agrégés. Enfin, le nombre de points d’intérêt agrégés est affiché au sein de la courbe, au point d’intersection entre la courbe et la droite passant par la projection du barycentre du cluster et le centre de l’écran (voir Figure 4.15d).

4.4 Conclusion

Dans ce chapitre, nous avons présenté notre contribution en termes de techniques de visualisation hors-champ. Les techniques présentées reposent sur la technique Halo (2D) [13] que nous avons adaptée au cas de la réalité augmentée mobile (3D). Trois variantes pour l'agrégation de POIs hors-champ définissent les techniques notées Halo3D, Halo3D* et Halo3D**. Halo3D** a fait l'objet d'un brevet déposé par Schneider Electric. Les avantages des trois techniques sont triples : (1) fournir des informations sur l'environnement hors-champ tout en limitant l'intrusion visuelle sur l'écran, (2) gérer une forte densité de POIs, (3) indiquer l'amplitude de mouvement nécessaire pour ramener un POI hors-champ dans le champ de vision de l'utilisateur. Les deux critères établis pour un système de visualisation hors-champ en RA mobile (voir section 3.2) sont donc vérifiés.

Nous présentons dans le chapitre suivant deux études expérimentales des éléments de conception détaillés dans ce chapitre.

5

Exploration expérimentale de trois techniques de visualisation hors-champ pour la réalité augmentée mobile

Sommaire

5.1	Introduction	75
5.2	Expérience 1 : Tâche abstraite de pointage	76
5.2.1	Introduction	76
5.2.2	Plan d'expérience	76
5.2.3	Résultats	81
5.2.4	Discussion	83
5.2.5	Conclusion de l'expérience 1	84
5.3	Expérience 2 : Tâche d'exploration	85
5.3.1	Introduction	85
5.3.2	Plan d'expérience	85
5.3.3	Résultats	91
5.3.4	Discussion	92
5.3.5	Conclusion de l'expérience 2	95
5.4	Conclusion	95

5.1 Introduction

Dans ce chapitre, nous présentons nos travaux expérimentaux liés aux techniques de visualisation de points d'intérêt (POIs) hors-champ dont la conception est détaillée au chapitre précédent. Nous décrivons deux études expérimentales. La première a pour objectif d'évaluer Halo3D pour une tâche abstraite de pointage similaire au scénario 2 (voir section 2.3) : acquérir des POIs placés autour de l'utilisateur. La deuxième a pour but d'explorer expérimentalement les deux variantes de Halo3D (Halo3D* et Halo3D**) décrites dans le chapitre précédent (voir section 4.3). Pour comparer Halo3D, Halo3D* et Halo3D**, nous considérons un contexte plus écologique que celui de la première étude. Pour cela nous nous inspirons du scénario 1 (voir section 2.3) et la tâche consiste à chercher et sélectionner des POIs placés sur une machine de production.

En synthèse des études expérimentales détaillées dans les deux sections suivantes nous soulignons les résultats expérimentaux suivants : Halo3D est 18% plus rapide que la technique focus+context AroundPlot* pour pointer sur un POI et perçue comme 34% moins intrusive que la technique Arrows2D à base de flèches. En contexte pseudo-écologique, les deux variantes Halo3D* et Halo3D** affichant la distribution spatiale des POIs dans les clusters (1) permettent une compréhension efficace de l'environnement hors-champ et (2) nécessitent moins d'effort que AroundPlot* pour localiser un POI dans l'environnement.

5.2 Expérience 1 : Tâche abstraite de pointage

5.2.1 Introduction

Halo3D est une technique de visualisation pour la Réalité Augmentée (RA) mobile, qui gère les environnements denses en points d'intérêt (POIs) sans être intrusif à l'écran. Parmi les techniques de visualisation en RA mobile présentées au chapitre 4, nous notons que la technique AroundPlot (voir section 3.4.1.2) est la seule technique à avoir été étudiée expérimentalement en environnement dense avec un grand nombre de POIs. La technique AroundPlot affiche sur chacun des bords de l'écran des indices visuels sur l'emplacement des POIs. Bien que la technique AroundPlot ne garantisse pas une intrusion constante le long des bords de l'écran, nous la comparons à la technique Halo3D. Nous considérons aussi une technique de référence basée sur des flèches 2D car la technique est largement utilisée dans les systèmes de RA industriels actuels pour pointer vers des POIs de maintenance. Inspirée du scénario 2 (voir section 2.3), l'étude considère une tâche de pointage pour acquérir des POIs hors-champ situés autour des participants.

5.2.2 Plan d'expérience

5.2.2.1 Appareil et participants

L'étude a été réalisée sur un téléphone Lenovo phab2PRO intégrant la technologie Tango de Google. Il offre à l'appareil une parfaite connaissance de l'espace grâce à une combinaison de suivi de mouvement et de perception de la profondeur (voir section 2.4.3). Le téléphone possède également un écran de 6,4 pouces avec une résolution de 2560 x 1440 pixels qui a été pleinement utilisée dans cette expérimentation. L'étude comprenait 12 volontaires non rémunérés d'un laboratoire universitaire en informatique (3 femmes et 9 hommes ; âge compris entre 27 et 40 ans, moyenne=30.5 ans, écart-type=3,8).

5.2.2.2 Techniques de visualisation développées

Le développement de Halo3D (voir Figure 5.1c) suit le modèle de conception décrit au chapitre 4, sauf pour la présentation de la profondeur (voir section 4.2.3) du POI (distance physique qui sépare le POI de l'utilisateur). Pour l'expérience, l'information de profondeur n'est pas utile, le participant tournant sur lui-même pour pointer un POI placé autour de lui. La technique Halo3D développée ne gère donc pas l'information de profondeur du

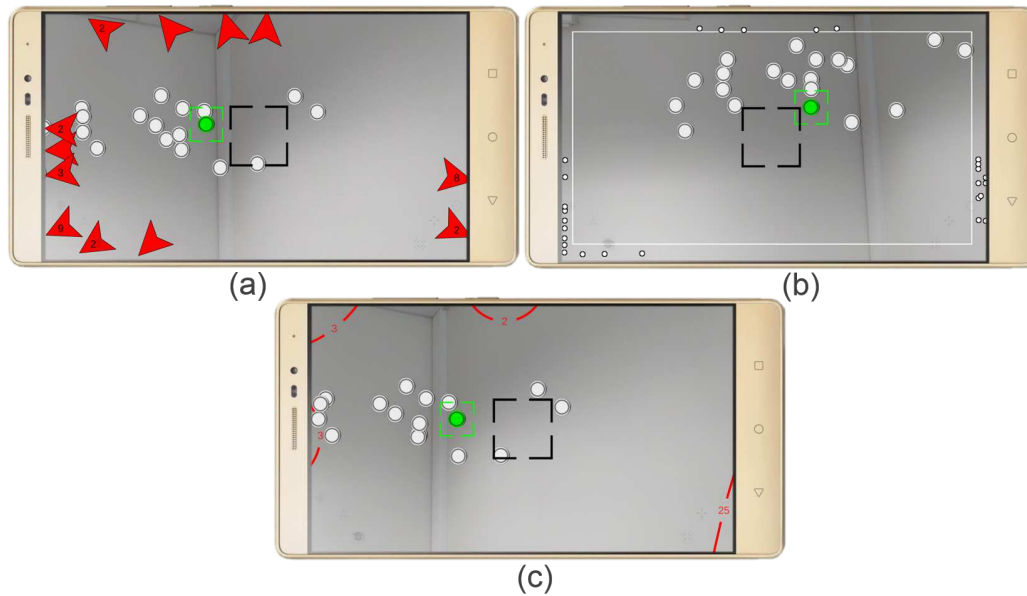


FIGURE 5.1 – Techniques de visualisation développées (a) Arrows2D (b) AroundPlot* (c) Halo3D.

POI traduite visuellement par l'épaisseur du trait de l'arc de cercle. Ainsi tous les arcs de cercle sont présentés avec la même épaisseur de trait.

Nous avons implémenté la technique basée sur des flèches 2D, notée Arrows2D, avec la même conception que le système de guidage RA hors-champ de Schneider (voir Figure 5.1a). Les flèches affichées ont une taille de 150 pixels. Nous avons également développé le même mécanisme d'agrégation que pour Halo3D : deux flèches s'agrègent lorsqu'elles se chevauchent visuellement à l'écran. La taille des flèches a été choisie comme un compromis entre le nombre de flèches affichées et leur intrusion à l'écran. En effet, des flèches plus petites (<100 pixels) entraînent moins de chevauchement entre elles, mais impliquent plus de flèches à l'écran car moins d'agrégations sont réalisées.

AroundPlot* est une technique inspirée de AroundPlot [65]. Nous avons répliqué la visualisation originale de AroundPlot : tout d'abord, un rectangle blanc est dessiné à l'écran et délimite la vue actuelle par rapport au contexte environnant affiché sur les bords de l'écran. De petits points noirs avec un contour blanc sont ensuite affichés sur les bords de l'écran pour indiquer l'emplacement des POIs hors-champ (voir Figure 5.1b). Nous avons également reproduit le redimensionnement dynamique de AroundPlot : le rectangle intérieur n'est pas fixé à l'écran mais grossit dans la direction du mouvement physique du dispositif mobile. Ce mécanisme améliore la lisibilité de l'environnement hors-champ. Pour une comparaison objective des visualisations des trois techniques, AroundPlot* est basée sur le même algorithme de projection de POIs que Halo3D et Arrows2D : le POI hors-champ est projeté sur le plan de l'écran de l'appareil mobile. AroundPlot* positionne sur l'écran la petite icône représentant un POI le long de la ligne définie par le centre de l'écran et la projection du POI. Enfin, avec AroundPlot*, aucune agrégation n'était nécessaire car les points de la zone de contexte sont petits et ne se chevauchent donc que

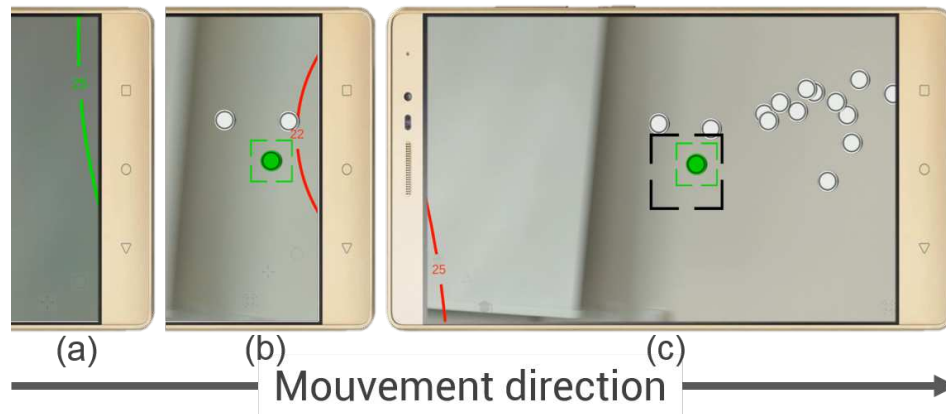


FIGURE 5.2 – Validation d'un POI avec la technique Halo3D : (a) un cercle vert indique la direction à suivre pour atteindre la cible, (b) la cible verte rentre sur l'écran, (c) la tâche est validée par l'alignement du curseur avec la cible

très rarement.

Les trois techniques ont été développées en C# et affichées sur l'écran de l'appareil mobile à l'aide du moteur de rendu Unity3D.

5.2.2.3 Tâche expérimentale et environnement

La tâche expérimentale et l'environnement sont inspirés du scénario 2 (voir section 2.3) : pointer vers des points d'intérêt hors-champ placés autour des utilisateurs. Les participants ont été invités à ramener le POI hors-champ désigné par la couleur verte dans le champ de vision de la caméra. Lorsque le POI apparaissait sur l'écran, les participants devaient valider l'acquisition en le plaçant entièrement à l'intérieur du curseur carré situé au centre de l'écran (voir Figure 5.2). La sélection est automatiquement validée dès que la cible est entièrement à l'intérieur du carré central (pas de temporisation ou "dwell time").

Deux paramètres guident la conception des environnements : premièrement, la densité des points d'intérêt est variable, comme l'illustre le scénario 2. Pour mesurer les performances des techniques de visualisation en fonction de la densité des POIs hors-champ, nous avons généré deux environnements, comme réalisé dans l'étude expérimentale d'Around-Plot [65] : un environnement à faible densité avec 5 POIs et un environnement à forte densité avec 50 POIs.

Un deuxième paramètre à considérer est l'amplitude du mouvement. Comme illustré dans le scénario 2 dans le cas de machines en U, les opérateurs peuvent être amenés soit à tourner complètement autour d'eux-mêmes pour acquérir un POI ou soit à effectuer un mouvement de faible amplitude. Dans l'étude, nous avons considéré deux amplitudes de mouvement : un petit mouvement défini par une distance angulaire inférieure à 80 degrés et un grand mouvement défini par une distance angulaire supérieure à 120 degrés (un POI derrière l'utilisateur). Cette distance est mesurée à partir d'un POI spécial appelé point de départ, identique pour toutes les tâches de pointage. Ainsi, il a été demandé aux

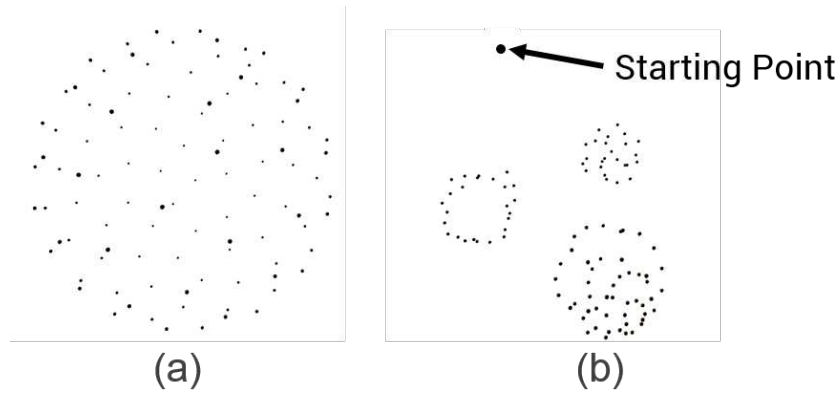


FIGURE 5.3 – Etude 1 : environnements générés avec 100 POIs (a) placés sur une sphère de support (b) disposés en 3 clusters

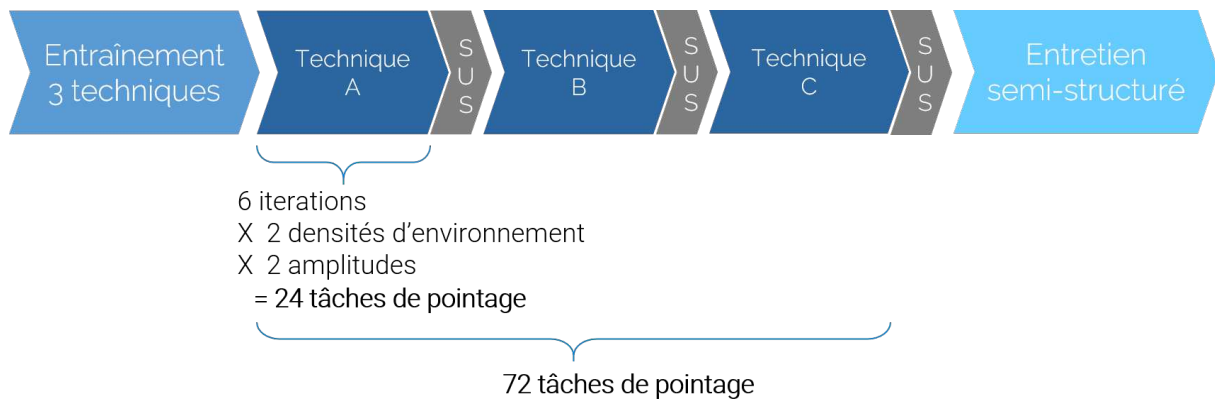


FIGURE 5.4 – Déroulement de l'expérience 1

participants de revenir au point de départ après chaque tâche de pointage.

En tenant compte de ces paramètres, nous avons généré les environnements comme suit : nous avons d'abord placé 100 POIs sur une sphère à l'aide de l'algorithme de Fibonacci (voir Figure 5.3a). Les 100 points ont ensuite été regroupés autour de 3 centres afin de créer un environnement organisé en clusters : un de chaque côté des participants et un autre derrière (voir Figure 5.3b). Les POIs placés sur les côtés des participants impliquent des mouvements de faible amplitude. Ceux placés derrière les participants impliquent des mouvements de grande amplitude. Comme les participants effectuent 6 itérations pour chaque condition de densité, nous avons créé 6 environnements pour les conditions éparées et denses afin d'éviter tout effet d'apprentissage. Nous avons ainsi généré 12 environnements différents.

5.2.2.4 Design expérimental et procédure

Design L'étude est basée sur un design intra-sujets : de 3 (*Visualisation* : Halo3D, Arrows2D, Aroundplot*) x 2 (*Densité* : nombre de POIs hors-champ : 5, 50) x 2 (*Amplitude de mouvement* : faible et grande amplitude de mouvement) conditions. Les techniques de visualisation ont été contrebalancées entre les participants en utilisant un carré latin. Pour chaque technique, toutes les combinaisons des conditions de densité et d'amplitude ont



FIGURE 5.5 – Participant effectuant une rotation sur place pour chercher un POI hors-champ

été présentées dans un ordre aléatoire pour contrer les effets d'apprentissage [39]. Chaque condition a été répétée 6 fois. Cela correspond à 24 tâches de pointage (6 itérations x 2 densités de POIs x 2 amplitudes) pour chaque technique de visualisation et 72 tâches par participant (voir Figure 5.4). Ces tâches ont été réalisées au sein de 12 environnements différents mais équivalents (6 itérations x 2 densités de POIs) comme expliqué ci-dessus.

Procédure Les participants ont effectué l'expérience debout (voir Figure 5.5). Pour rechercher les POIs, les participants effectuaient une rotation sur place en utilisant les indices visuels affichés sur l'écran du téléphone. Ils se sont d'abord formés sur chaque technique de visualisation. La phase d'entraînement impliquait un environnement différent de ceux de l'étude pour éviter tout effet d'apprentissage. Une fois qu'ils se sont estimés complètement à l'aise avec une technique de visualisation, ils passaient à l'entraînement de la technique de visualisation suivante. Le temps d'apprentissage moyen était de 19.1s avec Arrows2D, 24.2s avec AroundPlot* et de 26.8s avec Halo3D.

L'expérience entière avait une durée moyenne de 30 minutes.

5.2.2.5 Hypothèses

Nous avons formulé les hypothèses suivantes sur Halo3D :

- **H1.** Il est plus rapide d'accomplir les tâches avec Halo3D par rapport à AroundPlot*. Les points affichés par AroundPlot* étant petits, il sera plus long de localiser à l'écran un point vert qu'un arc de cercle vert plus grand.
- **H2.** En présence d'une forte densité de POIs, Halo3D est mentalement moins fatigant en comparaison à AroundPlot* et à Arrows2D. Ceci est dû aux propriétés d'intrusion visuelle constante sur l'écran et à l'absence de chevauchements de Halo3D.

5.2.2.6 Données collectées

Pour vérifier nos hypothèses, nous avons mesuré, pour chaque technique, le temps pris par les participants pour comprendre la localisation du POI hors-champ indiquée par la visualisation (temps d'interprétation de la visualisation), le temps de guidage écoulé jusqu'à ce que le POI surligné en vert apparaisse à l'écran (temps de guidage hors-champ),

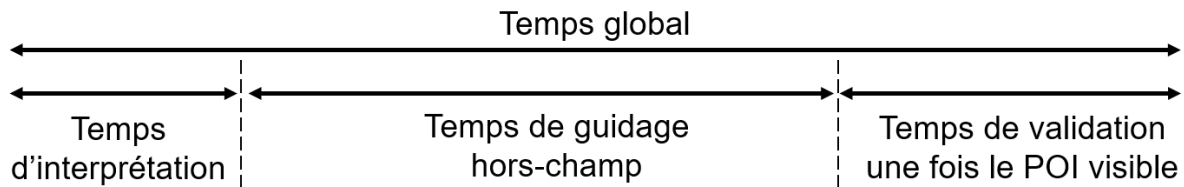


FIGURE 5.6 – Temps mesurés lors de l'expérience

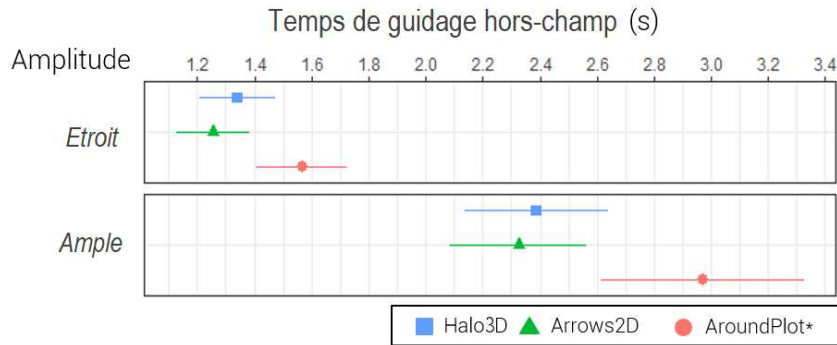


FIGURE 5.7 – Temps de pointage hors-champ en secondes pour les 3 techniques et les 2 amplitudes de mouvement. Les barres d'erreur révèlent des intervalles de confiance de 95%.

ainsi que le temps global de la tâche de pointage jusqu'à la validation (voir Figure 5.6). Nous avons également enregistré le nombre d'*overshoots*, soit le nombre de dépassements de cibles pour chaque tâche. Un dépassement survient lorsque le participant manque le POI cible : le participant suit le repère visuel, mais le mouvement est si rapide que le POI apparaît à l'écran, puis disparaît.

À la fin de l'utilisation d'une technique, les participants étaient invités à remplir un formulaire SUS [24] (voir Annexe A).

À la fin de la session, il a été demandé aux participants d'évaluer, pour chaque technique de visualisation, l'intrusion à l'écran, le niveau de charge mentale de la tâche et la facilité d'apprentissage des techniques (voir Annexe C). Les questions ont été évaluées sur une échelle de Likert à 5 valeurs [72]. Enfin, il a également été demandé aux participants de classer les techniques de visualisation par ordre de préférence.

5.2.3 Résultats

Nous rapportons les résultats en tenant compte des effets de chacun des trois facteurs (*Visualisation*, *Densité* et *Amplitude*) sur le temps nécessaire pour ramener à l'écran le POI hors-champ ainsi que le nombre d'*overshoots*. Toutes les itérations de chaque essai ont été agrégées pour chaque participant. Le Tableau 5.1 présente les résultats moyens. Un test de Shapiro-Wilk a indiqué que nos données étaient significativement différentes de la distribution normale. Ainsi, nous avons appliqué une transformation alignée par rangs sur les données [114], puis exécuté une analyse de variance à mesures répétées pour

-	Halo3D	Arrows2D	AroundPlot*
Temps d'interprétation (s)	0.9 ; 0.92(0.16)	0.91 ; 0.9(0.18)	1.17 ; 1.13(0.37)
Temps de guidage hors-champ (s)	1.86 ; 1.76(0.47)	1.79 ; 1.71(0.46)	2.27 ; 2.18(0.64)
Overshoots	0.35 ; 0(0.57)	0.21 ; 0(0.65)	0.21 ; 0(0.46)

TABLE 5.1 – Moyenne, médiane (et écart-type) des temps d'interprétation, des temps de guidage hors-champ, ainsi que du nombre d'*overshoots* pour Halo3D, Arrows2D et AroundPlot*

-	Fatigue mentale	Facilité d'apprentissage	Intrusion	Classement
Halo3D	2 ; 2(1.04)	4.25 ; 4(0.75)	2.33 ; 2(1.23)	3.5 ; 4(1.09)
Arrows2D	1.67 ; 1.5(0.78)	4.42 ; 4(0.51)	3.42 ; 3.5(1.31)	3.83 ; 4(0.72)
AroundPlot*	2.75 ; 3(1.22)	3.67 ; 3.5(1.15)	1.92 ; 2(0.9)	3.08 ; 3(1.16)

TABLE 5.2 – Moyenne, médiane (et écart-type) des notes sur une échelle de 5 valeurs (1 = très mauvais, 5 = très bien) données aux techniques pour la fatigue mentale, la facilité d'apprentissage, l'intrusion visuelle à l'écran, et le classement général

étudier les interactions possibles entre les facteurs.

5.2.3.1 Temps et overshoots

Une ANOVA à trois facteurs a montré que la visualisation ($F(2,132)=11.3$ et $p < 0.001$) et l'amplitude ($F(1,132) = 210.1$ et $p < 0.001$) avaient un impact significatif sur le temps de guidage hors-champ (voir Figure 5.7). Comme attendu, le temps de guidage hors-champ est nettement plus important lors des mouvements amples (moyenne=2,56s, écart-type=0,7s) que lors des mouvements de faible amplitude (moyenne=1,39s, écart-type=0,35s). Une analyse post-hoc avec comparaisons par paires et corrections de Bonferroni a indiqué que les participants passaient beaucoup plus de temps à exécuter les tâches avec AroundPlot* qu'avec Halo3D ($p < 0.001$) et Arrows2D ($p < 0.001$). Aucune différence statistiquement significative n'a été trouvée entre Halo3D et Arrows2D.

Une étude de l'impact de la visualisation sur le temps d'interprétation avec un test post-hoc avec comparaisons par paires et corrections de Bonferroni a confirmé que statistiquement, les participants passaient beaucoup plus de temps à interpréter la visualisation AroundPlot* que les visualisations Halo3D ($p < 0.001$) et Arrow2D ($p < 0.001$).

Enfin, nous n'avons pas trouvé d'impact significatif des différents facteurs sur le nombre d'*overshoots*.

5.2.3.2 Préférences subjectives

Les scores SUS indiquent que Halo3D (74,75/100) et Arrows2D (81/100) ont été perçus comme plus utilisables que AroundPlot* (70/100).

Le Tableau 5.2 présente les résultats de notre questionnaire (voir Annexe C). Un test de Friedman a montré que les techniques de visualisation avaient un impact significatif (avec $\alpha = 0.05$) sur l'intrusion perçue à l'écran ($\chi_2^2 = 9.41$, $p = 0.009$), l'effort mental requis pour accomplir la tâche ($\chi_2^2 = 11.2$, $p = 0.004$) et la facilité d'apprentissage des techniques de visualisation ($\chi_2^2 = 7.27$, $p = 0.03$). Mais aucune différence significative n'a été constatée sur le classement général.

Une analyse post-hoc avec comparaisons par paires et corrections de Bonferroni a révélé qu'Arrows2D a été perçu comme beaucoup plus intrusif qu'AroundPlot* ($p=0.007$). Concernant l'effort mental, la technique AroundPlot* a été jugée beaucoup plus exigeante que Arrows2D ($p=0.009$). Enfin, pour la facilité d'apprentissage, le test post-hoc n'a révélé aucune différence statistiquement significative entre les facteurs.

5.2.4 Discussion

5.2.4.1 Temps de complétion de la tâche

Les résultats confirment notre hypothèse **H1** : les participants ont passé beaucoup plus de temps à exécuter les tâches avec AroundPlot* qu'avec Halo3D et Arrows2D. Le temps nécessaire à l'exécution des tâches inclut (1) le temps nécessaire pour interpréter la visualisation et comprendre quelle direction doit être suivie, et (2) le temps nécessaire pour ramener le POI hors-champ dans le champ de vision. Comme le montre le Tableau 5.1, les participants ont amené 18 fois plus rapidement les POIs dans le champ de vision de la caméra avec Halo3D qu'avec AroundPlot*. Les faibles performances d'AroundPlot* sont dues à la forte densité du nuage de points dans la zone du contexte. Comme indiqué dans le Tableau 5.1, le temps consacré à rechercher le point vert sur l'écran avec AroundPlot* était 24 fois supérieur à celui nécessaire pour interpréter les repères visuels de Halo3D et de Arrows2D. Les flèches d'une taille de 150 pixels ont aidé les participants à localiser facilement le repère visuel vert à l'écran. Halo3D offrait un temps moyen similaire pour localiser le repère visuel vert sur l'écran (c'est-à-dire un arc de cercle) tout en étant moins intrusif à l'écran que Arrows2D. Ces résultats montrent les avantages de Halo3D.

5.2.4.2 Difficulté mentale

Le classement des techniques de visualisation est corrélé à la difficulté mentale ressentie lors des tâches.

Arrows2D a été légèrement préféré à Halo3D : comme la forme de la flèche indique intuitivement une direction, il est mentalement plus facile de déduire une direction d'une flèche que d'un arc de cercle. Cela pourrait expliquer pourquoi les participants ont trouvé la technique Halo3D légèrement plus exigeante sur le plan mental que Arrows2D, contrairement à notre hypothèse **H2**. La technique AroundPlot* a été jugée nettement plus exigeante qu'Arrows2D : AroundPlot* fournit un aperçu complet de l'espace hors-champ dans une petite section de l'écran, la zone du contexte. Dans les environnements de forte

densité, ces points sont proches les uns des autres et une attention visuelle importante est nécessaire pour repérer le point vert à l'écran. La fatigue mentale est donc liée à la densité des informations affichées et à la taille de la zone de l'écran dans laquelle elles y sont affichées.

5.2.4.3 Effet de la configuration des environnements

Le nombre de POIs (5 vs 50) n'a pas eu d'impact significatif sur les performances des participants. Ce résultat suggère que le mécanisme d'agrégation n'a pas d'impact sur les performances tout en offrant une solution pour gérer les environnements à forte densité.

De plus, la densité n'a pas eu d'impact sur le nombre d'*overshoots*. Cela est principalement dû à la distribution spatiale des POIs : les POIs ont été regroupés en clusters pour simuler un scénario réaliste. A mesure que les participants se rapprochent d'un POI, la pointe du cluster contenant le point d'intérêt commence à apparaître à l'écran. Les participants savaient donc qu'ils se rapprochaient du POI cible et ralentissaient leurs mouvements, minimisant ainsi le risque d'*overshoots*.

5.2.5 Conclusion de l'expérience 1

Les résultats indiquent que Halo3D est une technique efficace pour les tâches de pointage 3D en RA mobile, en particulier au sein d'environnements denses. Halo3D est aussi performant que Arrows2D tout en étant moins intrusif. De plus, Halo3D est plus rapide qu'AroundPlot*. Cependant, deux participants ont verbalement fait part de leurs préoccupations concernant l'agrégation visuelle d'Halo3D : "Le point d'intérêt entré à l'écran se trouvait en dehors du cercle" ou "Je m'attendais à ce que le point d'intérêt rentre à l'écran vers le centre du cercle. Mais il est entré à un endroit différent et j'ai dû réajuster mon mouvement". Ces commentaires indiquent que les participants n'ont pas été en mesure de bien comprendre la distribution spatiale des POIs au sein d'un cluster. Les participants ont dû changer de direction de pointage lorsque les POIs se sont désagrégés, car la direction suivie auparavant était fautive. Ce problème a été décrit dans le chapitre 4 (voir section 4.3.1 et Figure 4.13) comme un guidage en deux étapes et est confirmé expérimentalement.

Pour résoudre ce problème, nous avons décrit au chapitre 4 deux variantes de Halo3D permettant de visualiser la distribution spatiale des POIs : Halo3D* et Halo3D** (voir section 4.3). Nous proposons dans l'étude suivante une évaluation comparative de ces deux variantes de Halo3D.

5.3 Expérience 2 : Tâche d'exploration

5.3.1 Introduction

Dans cette deuxième étude expérimentale, nous visons un environnement plus écologique. L'objectif est de déterminer quelle technique favorise l'exploration de l'environnement hors-champ. Inspiré du scénario 1 (voir section 2.3), le but de la tâche est d'explorer un environnement pseudo-industriel et de trouver tous les POIs ancrés à une machine de production. Cette étude diffère de l'expérience 1 en trois points : (1) Environnement pseudo-écologique : l'environnement à l'écran est visuellement chargé avec des câbles et des capteurs sur une face d'une machine de production. (2) Tâche d'exploration : les participants sont libres de suivre n'importe quelle trajectoire pour sélectionner tous les POIs. 3) Techniques comparées : nous comparons cinq techniques de visualisation : Arrows2D, AroundPlot*, Halo3D, Halo3D* et Halo3D**.

5.3.2 Plan d'expérience

5.3.2.1 Appareil et participants

L'expérience a été réalisée avec le même appareil mobile que celui de la première étude : un téléphone Lenovo phab2PRO intégrant la technologie Tango de Google. L'écran du téléphone possède une résolution de 2560 x 1440 pixels et une taille de 6,4 pouces. Pour reproduire un environnement industriel complexe, nous avons photographié une machine de production Schneider de 9m x 3m. Nous avons imprimé les photographies sur des panneaux forex d'une épaisseur de 3mm pour construire l'unité (voir Figure 5.9b). La taille du modèle résultant est de 3 m de long et 0,75 m de large. Nous avons sollicité 15 volontaires non rémunérés (3 femmes, 12 hommes ; âgés de 23 à 47 ans, moyenne=31 ans, écart-type=6,4) dans un laboratoire universitaire en informatique. Aucun des participants n'avait d'expertise concernant les machines de production. Il s'agissait de participants différents de ceux de l'expérience 1.

5.3.2.2 Techniques de visualisation développées

Les techniques Arrows2D et de AroundPlot* sont celles développées pour l'expérience 1 (voir section 5.2.2.2).

Halo3D, Halo3D* et Halo3D** suivent les modèles de conception détaillés au chapitre 4 (voir section 4.3). Tandis que dans l'expérience 1, l'information de profondeur n'avait aucune incidence sur la tâche à réaliser (l'utilisateur faisant uniquement des rotations sur lui-même), dans cette expérience, la distance d'un POI est une information qui peut éventuellement être pertinente. Pour décider si l'information de profondeur est à gérer par les techniques Halo3D, nous avons conduit une expérience pilote avec 6 participants. La tâche consistait à chercher et sélectionner des POIs distribués sur une machine de production. Nous avons émis l'hypothèse que la visualisation de la profondeur allait

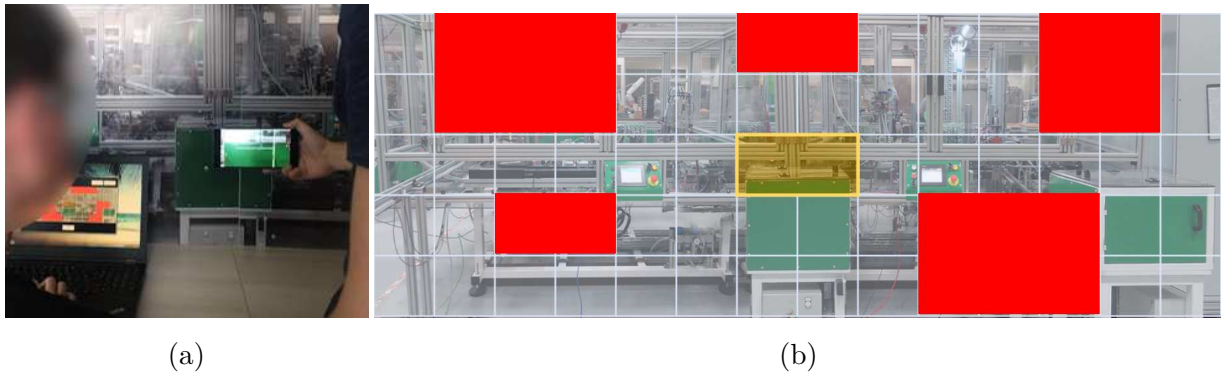


FIGURE 5.8 – Tache de localisation : capture d'écran de la grille. En rouge : cellules sélectionnées par un participant. En jaune : champ de vision de la caméra de l'appareil mobile.

optimiser le temps de complétion de la tâche. Tous les participants ont trouvé l'utilisation de la profondeur compliquée et inutile. En effet ils ont rapporté que l'ajout de l'indication de profondeur par l'épaisseur du trait était compliqué à exploiter car ils devaient déjà comprendre la direction et l'amplitude du geste à faire pour ramener le POI dans le champ de vision de la caméra, ainsi qu'appréhender le mécanisme d'agrégation. Aussi les participants ont rapporté ne pas utiliser l'indication de distance. Leur stratégie observée était de se guider par les halos pour obtenir les cibles visibles sur l'écran et, à partir de là, d'évaluer mentalement quel était le chemin le plus court pour l'atteindre. Nous avons donc écarté l'affichage de la profondeur comme pour l'expérience 1.

5.3.2.3 Tâche expérimentale et environnement

La tâche expérimentale s'inspire de notre premier scénario industriel (voir section 2.3) : les participants devaient rechercher et sélectionner des POIs placés devant eux sur une machine. 35 POIs leur ont été placés sur une face d'une machine de production. La tâche expérimentale était divisée en deux sous-tâches : (1) une tâche consistant à localiser sur une grille tous les POIs hors-champ de la machine, en fonction de la visualisation affichée à l'écran et (2) une tâche d'exploration pour rechercher et sélectionner tous les POIs sur la machine.

Tâche de localisation. Cette première tâche est inspirée de la "tâche de localisation" des études Halo [13] et Wedge [59] : les participants devaient localiser tous les POIs hors-champ à l'aide d'une technique donnée. Pour cela, une grille leur a été présentée sur un ordinateur, composée de 14 colonnes et de 5 lignes. La grille possède comme image de fond la machine de production (voir la Figure 5.8b). Après avoir examiné la visualisation présentée sur l'écran de l'appareil, il leur a été demandé de remplir la grille pendant que l'expérimentateur tenait l'appareil mobile à une position fixe (voir Figure 5.8a). La position fixe de l'expérimentateur avait pour but d'obtenir un champ de vision correspondant à l'encadré jaune dans la Figure 5.8b). L'expérimentateur maintenait le dispositif mobile à côté des participants, de sorte qu'ils puissent regarder la visualisation affichée sur l'écran, et remplir la grille en même temps (voir Figure 5.8a). Les participants ont ainsi été invités

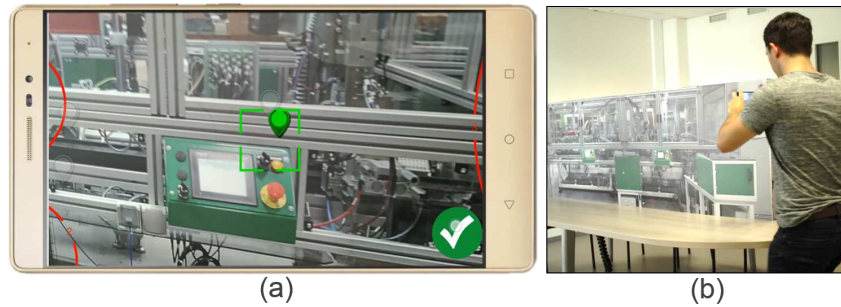


FIGURE 5.9 – Tâche d'exploration : (a) Sélection d'un POI en utilisant Halo3D** (b) Un participant se déplace devant une réplique d'une machine de production pour chercher les POIs.

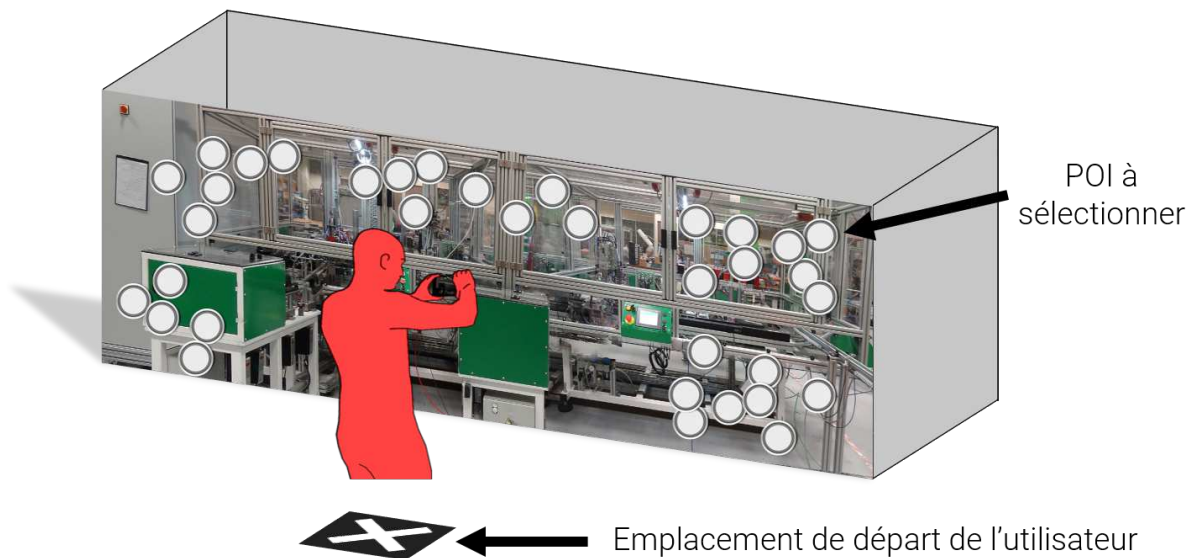


FIGURE 5.10 – Exemple d'un environnement contenant 35 POIs et emplacement de départ de l'utilisateur

à cliquer sur toutes les cellules dans lesquelles ils pensaient trouver un ou plusieurs POIs.

Tâche d'exploration. Les participants ont été invités à rechercher et à sélectionner un par un chaque POI dans l'ordre de leur choix. Un POI visible à l'écran devient *sélectionnable* lorsque le participant est à moins de 30 cm de celui-ci. Si tel est le cas, un bouton *Valider* apparaît dans l'angle inférieur droit de l'écran, facilement sélectionnable avec le pouce (voir Figure 5.9a). La sélection s'effectue en alignant le POI avec le curseur central de l'écran et en appuyant simultanément sur le bouton *Valider*. Lorsque le POI est sélectionné, il disparaît de l'environnement RA. Cette tâche (1) simule une opération de maintenance effectuée sur un POI (par exemple, contrôler la valeur de la température à un endroit spécifique d'une machine) et (2) oblige les participants à utiliser la technique de visualisation hors-champ pour ramener à l'écran les POIs restants, comme le montre Figure 5.9. En effet, les participants étant proches de la machine, les POIs restants sont hors du champ de vision de la caméra de l'appareil mobile.

Pour éviter tout effet d'apprentissage d'une technique à l'autre, 5 environnements équivalents

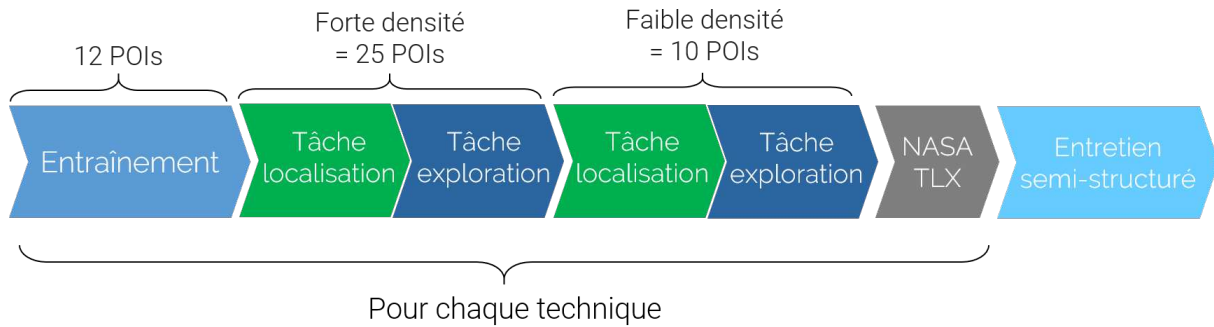


FIGURE 5.11 – Déroulement de l'expérience 2

(un pour chaque technique) contenant chacun 35 POIs ont été générés (voir Figure 5.10). Chaque environnement est composé de 5 clusters de POIs, et chaque cluster contient respectivement 8,6,8,8 et 5 POIs. Nous avons ensuite construit les 5 environnements en modifiant l'emplacement des clusters sur la machine. Les environnements générés sont les mêmes pour tous les participants.

5.3.2.4 Design expérimental et procédure

Nous utilisons un protocole intra-sujet en mesure répétées à une seule variable : les techniques de visualisation Arrows2D, AroundPlot, Halo3D, Halo3D* et Halo3D**.

Toutes les techniques ont été contrebalancées entre les participants : nous avons divisé les participants en 5 groupes de 3 personnes. Chaque groupe a débuté l'expérimentation avec une technique différente. Pour s'assurer que le couple (technique x environnements) soit équilibré entre tous les participants, les environnements ont été présentés dans le même ordre. Comme les techniques sont contrebalancées, chaque technique est associée aux 5 environnements entre tous les participants. Au total, nous avons enregistré 175 tâches de pointage par participant (5 environnements x 35 POIs).

Les 5 techniques ainsi que les tâches expérimentales ont d'abord été expliquées aux participants. En raison du nombre élevé de techniques, les participants ont été invités, pour chaque technique, à réaliser la session d'évaluation immédiatement après la session d'entraînement [31] (voir Figure 5.11). Les tâches d'entraînement consistaient à localiser les POIs hors-champ sur la grille et à les sélectionner sur la machine. Les environnements d'entraînement contenaient 12 POIs, placés à des endroits différents de ceux des environnements d'évaluation. Pour chaque technique, l'expérimentation n'a débuté que lorsque les participants estimaient être à l'aise avec la tâche. Dans le cas contraire, il leur a été demandé d'effectuer un autre bloc d'entraînement avec 12 POIs. Les séances d'entraînement avant chaque visualisation ont été conçues pour permettre aux participants de complètement se familiariser avec les techniques et ainsi d'atténuer le transfert éventuel de compétences entre elles. Ceci nous a permis d'utiliser une conception expérimentale intra-sujet sans avoir besoin de pleinement contre-balancer les techniques entre les participants.

Les participants ont commencé la tâche à partir d'un emplacement spécifique devant la machine (voir Figure 5.10). L'emplacement était marqué d'une croix blanche sur le sol. Après avoir examiné la visualisation présentée sur l'écran de l'appareil, il leur a d'abord été demandé de remplir la grille de la tâche de localisation. Les participants devaient ensuite rechercher et sélectionner les points d'intérêt sur la machine. Lorsqu'il ne restait plus que 10 POIs, les participants devaient à nouveau remplir la grille. Cela nous a permis d'étudier la technique qui facilite le mieux la compréhension de l'environnement hors-champ pour les cas d'environnements à forte densité (première tâche de localisation) et à faible densité (deuxième tâche de localisation) (voir Figure 5.11).

Après avoir achevé les tâches de localisation et d'exploration pour une technique et un environnement, les participants ont rempli le formulaire Raw TLX [61] (voir Annexe B) et revenaient à l'emplacement de départ pour continuer avec la technique suivante (voir Figure 5.11). Une séance durait environ 40 minutes.

5.3.2.5 Hypothèses

Nos hypothèses sont les suivantes :

- **H1.** Halo3D*, Halo3D** et AroundPlot* sont moins sujets aux erreurs pour la tâche de localisation que d'autres techniques. En effet, Arrows2D ne fournit aucune information de distance et Halo3D n'affiche pas la distribution spatiale des POIs.
- **H2.** Halo3D et ses variantes sont mentalement moins fatigantes par rapport à AroundPlot*, car AroundPlot* affiche tous les POIs dans une petite section de l'écran pour la tâche d'exploration.

5.3.2.6 Données collectées

Pour vérifier nos hypothèses, nous avons mesuré, pour chaque technique, le temps nécessaire pour (1) localiser les POI hors-champ sur la grille et (2) valider tous les POIs sur la machine. Pour la tâche de localisation, nous avons calculé le nombre de réponses incorrectes dans la grille. Une erreur est comptée lorsque les participants ont soit sélectionné une cellule incorrecte (c'est-à-dire une cellule ne contenant pas de POI hors-champ) soit omis une cellule correcte (à savoir une cellule contenant un ou plusieurs POI hors-champ). Une fois les tâches de localisation et d'exploration exécutées pour une technique, les participants ont été invités à remplir un formulaire Raw TLX [61] (voir Annexe B). À la fin de la session, il a été demandé aux participants d'évaluer, pour chaque technique de visualisation, l'intrusion à l'écran, le niveau de charge mentale de la tâche d'exploration et la facilité d'apprentissage des techniques (voir Annexe C). Les questions étaient à évaluer sur une échelle de Likert à 5 valeurs [72]. Enfin, les participants ont également été invités à classer les 5 techniques.

-	Taux d'erreur (forte densité)	Taux d'erreur (faible densité)
Halo3D	25.67 ; 26(5.19)	8.93 ; 9(3.05)
Arrows2D	27.74 ; 28(5.31)	9.07 ; 8(4.48)
AroundPlot*	26.27 ; 26(5.92)	11.4 ; 10(3.56)
Halo3D*	29.2 ; 29(3.4)	10.67 ; 11(2.84)
Halo3D**	24.94 ; 25(7.15)	9.67 ; 9(4.29)

TABLE 5.3 – Moyenne, médiane (et écart-types) du nombre d'erreurs en forte et faible densité pour Halo3D, Arrows2D, AroundPlot*, Halo3D* et Halo3D**

-	Temps d'exploration (s)
Halo3D	93.4 ; 84.9(26.36)
Arrows2D	82.85 ; 83.25(15.69)
AroundPlot*	77.79 ; 79.32(15.73)
Halo3D*	76.07 ; 77.38(14.09)
Halo3D**	75.7 ; 80.12(14.76)

TABLE 5.4 – Moyenne, médiane (et écart-types) du temps d'exploration Halo3D, Arrows2D, AroundPlot*, Halo3D* et Halo3D**

-	Fatigue mentale	Facilité d'apprentissage	Intrusion	Classement
Halo3D	2.2 ; 2(0.86)	3.8 ; 4(1.57)	1.87 ; 2(0.64)	3.67 ; 4(0.49)
Arrows2D	2.6 ; 2(1.12)	3.87 ; 4(1.41)	2.93 ; 3(1.44)	2.87 ; 3(0.92)
AroundPlot*	3.07 ; 3(1.16)	3.73 ; 4(1.39)	2.13 ; 2(1.25)	3.27 ; 3(1.28)
Halo3D*	2.27 ; 2(0.96)	4.07 ; 5(1.39)	2 ; 2(0.85)	3.8 ; 4(0.94)
Halo3D**	2.4 ; 2(1.12)	3.67 ; 4(1.35)	2.13 ; 2(0.99)	3.67 ; 4(1.05)

TABLE 5.5 – Moyenne, médiane (et écart-type) des notes sur une échelle de 5 valeurs (1 = très mauvais, 5 = très bien) données aux techniques pour la fatigue mentale, la facilité d'apprentissage, l'intrusion visuelle à l'écran, et le classement général

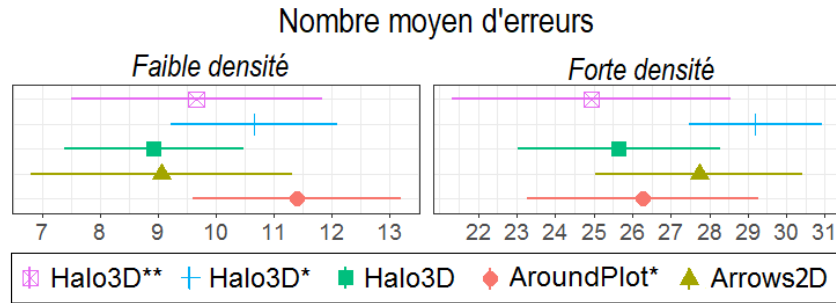


FIGURE 5.12 – Taux d'erreur dans les environnements à faible et à forte densités. Les barres d'erreur indiquent des intervalles de confiance à 95%.

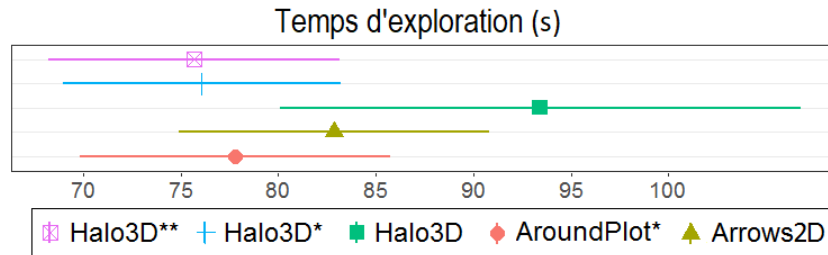


FIGURE 5.13 – Temps d'exploration pour toutes les techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.

5.3.3 Résultats

5.3.3.1 Taux d'erreur et temps de complétion

Nous rapportons les résultats en prenant en compte les effets de chaque visualisation sur le taux d'erreur lors de la tâche de localisation (voir Tableau 5.3 et Figure 5.12) et le temps nécessaire pour sélectionner tous les POIs sur la machine lors de la tâche d'exploration (voir Tableau 5.4 et Figure 5.13). Un test de Shapiro-Wilk a indiqué que nos données ne suivaient pas une distribution normale. Le test non-paramétrique de Kruskal-Wallis n'a révélé aucun effet de la visualisation sur le taux d'erreur pour les environnements à forte densité ($\chi_4^2 = 6.69$, $p=0.15$) et à faible densité ($\chi_4^2 = 7.89$, $p=0.09$), ainsi que sur le temps d'exploration ($\chi_4^2 = 6.17$, $p=0.18$).

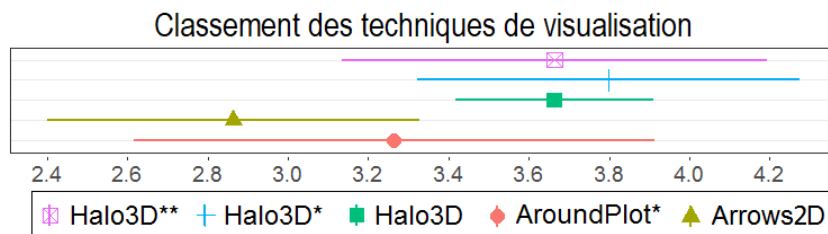


FIGURE 5.14 – Notes sur 5 (1 = très mauvais, 5 = très bien) données aux techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.

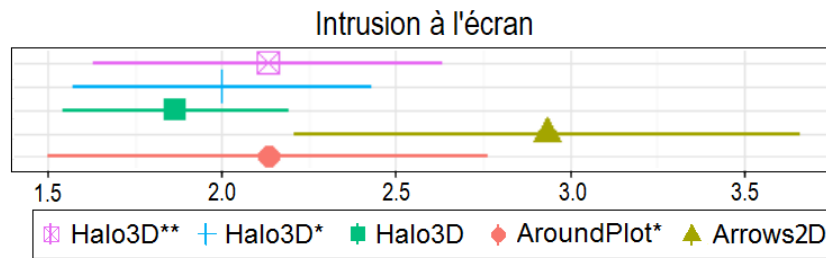


FIGURE 5.15 – Intrusion à l'écran des techniques sur 5 (1 = très mauvais, 5 = très intrusif) selon les participants. Les barres d'erreur indiquent des intervalles de confiance à 95%.

5.3.3.2 Préférences subjectives

Résultats Raw TLX. Un test de Friedman a révélé un effet de la visualisation sur l'effort ($\chi_4^2 = 10.69$, $p=0.03$), et la fatigue physique ($\chi_4^2 = 11.4$, $p=0.02$) ressentie pendant la tâche d'exploration. Cependant, dans les deux cas, un test post-hoc avec comparaisons par paires et corrections de Bonferroni n'a pas révélé de différences statistiquement significatives entre les facteurs. Aucune différence statistiquement significative n'a également été constatée dans toutes les autres mesures : frustration ($\chi_4^2 = 8.7$, $p=0.06$), exigence mentale ($\chi_4^2 = 6.87$, $p=0.14$), performance ($\chi_4^2 = 9.6$, $p=0.05$) et exigence temporelle ($\chi_4^2 = 2.14$, $p=0.71$).

Résultats du questionnaire en fin de session. Le Tableau 5.5 ainsi que la Figure 5.14 présentent les résultats de notre questionnaire (voir Annexe C). Un test de Friedman a révélé un effet important de la visualisation sur l'intrusion à l'écran ($\chi_4^2 = 10.33$, $p=0.035$). Cependant, un test post-hoc avec comparaisons par paires et corrections de Bonferroni n'a pas révélé de différences statistiquement significatives entre les facteurs. Aucune différence statistiquement significative n'a également été constatée dans toutes les autres mesures : facilité d'apprentissage ($\chi_4^2 = 2.93$, $p=0.57$), fatigue mentale ($\chi_4^2 = 6.08$, $p=0.19$) et note globale ($\chi_4^2 = 4.76$, $p=0.31$).

5.3.4 Discussion

5.3.4.1 Quantité d'information sur l'environnement hors-champ

AroundPlot* est la seule technique qui affiche directement l'environnement hors-champ, y compris les relations spatiales entre les POIs. En effet, il existe une correspondance directe entre les POIs hors-champ et les points affichés dans la zone du contexte. Bien que nous n'ayons pas trouvé de différences statistiquement significatives concernant le taux d'erreur, 8 participants ont déclaré que la technique AroundPlot* incitait à remplir la grille avec plus de précision par rapport aux autres techniques. AroundPlot* permet de bien comprendre l'environnement hors-champ, mais cette précision a été rapportée par les participants comme non utile pour la tâche d'exploration. Un participant a déclaré : "Je n'ai pas besoin de connaître exactement l'emplacement des points d'intérêt sur la machine, mais seulement la zone où ils se trouvent et combien ils sont : c'est suffisant pour moi". Un autre participant a fait une analogie avec la vision périphérique humaine : "Lorsque vous

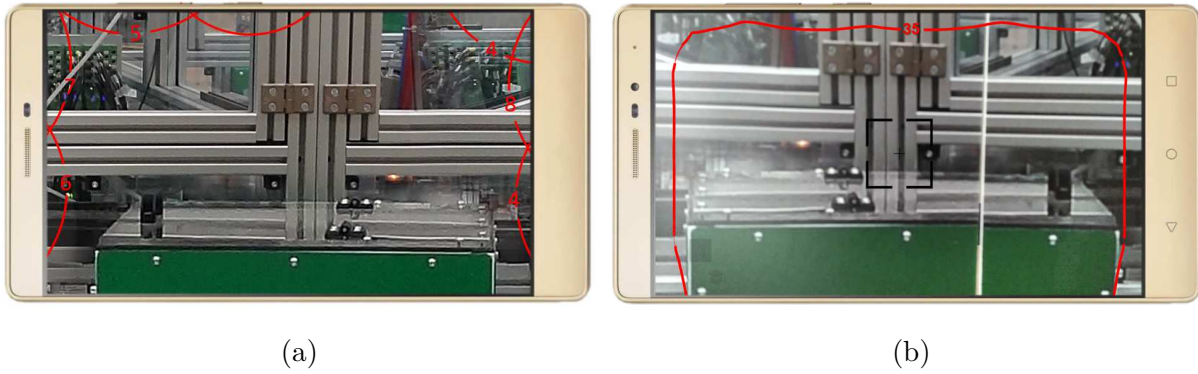


FIGURE 5.16 – (a) Halo3D* affichant un arc complet pour chaque cluster (b) Halo3D** n'affichant qu'une seule courbe lorsque le nombre de POIs est élevé.

regardez un objet devant vous, vous pouvez le voir très distinctement. Cependant, tous les objets situés dans votre vision périphérique sont flous, mais vous savez approximativement où ils se trouvent". Cette analogie convient parfaitement aux techniques basées sur le halo, car elles fournissent uniquement direction et distance vers les objets hors-champ.

Contrairement à AroundPlot*, la technique Arrows2D fournit uniquement une direction vers un POI hors-champ. La technique Arrows2D a été classée comme la pire technique pour la tâche de localisation ainsi que dans le classement final (voir Figure 5.14). Tous les participants ont déclaré : "Je n'ai aucune idée de la distance qui sépare les points, car aucune information de distance n'est fournie". Ce retour d'information des participants conforte notre hypothèse **H1**, mais l'expérience n'a pas révélé d'impact statistiquement significatif des techniques sur le taux d'erreur. De plus, il n'a pas été montré de différence de temps statistiquement significative entre les trois techniques basées sur Halo et Arrows2D au cours de la tâche d'exploration. Ceci confirme les résultats de l'étude 1 : une visualisation basée sur le halo offre des performances temporelles similaires à celles d'Arrows2D, tout en fournissant des informations de distance supplémentaires.

5.3.4.2 Techniques basées sur Halo

Bien que l'expérience n'ait pas mis en évidence un taux d'erreur plus élevé et statistiquement significatif au cours de la tâche de localisation, ce qui contredit notre hypothèse **H1**, 7 participants ont déclaré avoir été induits en erreur par Halo3D : les POIs qu'ils ont placés sur les grilles ne correspondaient pas à ce qu'ils ont trouvé sur la machine pendant la phase d'exploration. En effet, Halo3D n'indique que l'emplacement et la distance du centre de gravité des clusters hors-champ. Comme un cluster peut contenir des POI plus ou moins étalés, certains d'entre eux indiqués par un arc d'agrégation peuvent être situés en dehors du cercle de l'arc affiché.

Halo3D* résout les problèmes ci-dessus en réduisant le nombre d'agrégation. Les chevauchements autorisés entre les arcs ne perturbent pas les participants, car Halo3D* n'a pas été jugé plus exigeant sur le plan de la charge cognitive que Halo3D. Les participants ont eu des performances similaires avec Halo3D* et Halo3D**. Cependant, les commentaires

des participants ont permis de comprendre pourquoi ils s'estimaient plus à l'aise avec Halo3D*. Un participant a déclaré qu'”il était plus facile de deviner l'emplacement d'un POI avec Halo3D*, car l'arc complet était visible à l'écran” (voir Figure 5.16a). Avec Halo3D**, les arcs qui se chevauchent sont coupés à leur intersection pour lisser la courbe (voir Figure 4.15d). Ainsi, les courbes Halo3D** n'affichent jamais d'arcs complets de bordure d'écran à bordure d'écran. Plus le nombre d'arcs composant la courbe est important, plus il devient difficile de déterminer l'emplacement des POIs agrégés.

5.3.4.3 Environnements à forte densité vs faible densité

Nous avons considéré qu'un environnement avec 35 POIs était un environnement à forte densité. Cependant, 35 POIs ont suffi à révéler les limites de plusieurs techniques : au début de la tâche d'exploration, Halo3D** affiche une seule courbe continue partant du bord gauche de l'écran, passant par le bord supérieur et se terminant sur le bord droit de l'affichage (voir Figure 5.16b). Comme indiqué par 6 participants, nous obtenons une visualisation qui ”ne fournit aucune information utile sur l'environnement hors-champ”, tout comme trente flèches seraient inutiles à l'écran.

Dans les environnements à forte densité, AroundPlot* affiche des groupes de points qui se chevauchent dans la zone du contexte. Comme mentionné par 9 participants, AroundPlot* oblige les utilisateurs à se concentrer intensément sur cette petite zone de l'écran, afin de déterminer le nombre et la distance des POIs. Il est plus difficile de déterminer le nombre de POIs hors-champ avec AroundPlot*, car les participants auraient besoin de compter chaque point dans cette section de l'écran. Avec le mécanisme d'agrégation et le nombre de POIs agrégés affichés à l'écran, les techniques basées sur halo conviennent mieux aux cas d'utilisation industriels (voir la section 2.3) : le nombre de POIs informe les opérateurs de maintenance des sections de la machine qui requiert le plus d'attention. Ainsi, les participants ont attribué à AroundPlot* une charge cognitive supérieure à celle des techniques à base de halos. Bien que les réactions de ces participants soient cohérentes avec notre hypothèse **H2**, l'expérimentation n'a pas révélé d'impact statistiquement significatif des techniques sur la charge mentale.

Dans les environnements à faible densité, 5 participants ont déclaré ne pas utiliser la visualisation AroundPlot* au cours de la tâche d'exploration car ”il était difficile de localiser les points restants à l'écran”. Contrairement à l'étude 1, ce comportement n'a pas d'impact sur le temps requis pour effectuer la tâche : ces participants étaient toujours en mesure de rechercher manuellement les points sur la machine. Enfin, les environnements à faible densité caractérisent la seule situation dans laquelle les participants ont trouvé Arrows2D utile lors de la tâche d'exploration, comme indiqué par 4 participants. En effet, les flèches ne se chevauchent pas souvent et sont plus faciles à localiser sur l'écran.

5.3.5 Conclusion de l'expérience 2

Cette deuxième expérience avait pour objectif de comparer Halo3D* et Halo3D**, deux variantes de Halo3D qui fournissent des informations supplémentaires sur la dispersion spatiale des POIs. Dans un contexte plus réaliste, les résultats ne montrent aucun impact statistiquement significatif des techniques sur le taux d'erreur et la charge mentale. Néanmoins, les participants ont indiqué avoir été induits en erreur par Halo3D mais non par Halo3D* et Halo3D**.

5.4 Conclusion

Nous avons d'abord réalisé une première expérience de Halo3D avec une tâche de pointage. Les résultats montrent que les participants sont 18% plus rapides avec Halo3D qu'avec AroundPlot* et aussi rapide qu'Arrows2D tout en étant perçu comme 34% moins intrusif à l'écran. En contexte pseudo-industriel, les résultats de la seconde expérience révèlent que les deux variantes de Halo3D, Halo3D* et Halo3D** permettent une compréhension efficace de l'environnement hors-champ.

Les résultats, même si certains sont obtenus dans un environnement pseudo écologique doivent être confirmés dans un contexte industriel : en particulier le fait que l'utilisateur ait des tâches à réaliser (ouvrir un POI, étudier une valeur entre chaque recherche de POI) peut avoir une influence sur les résultats.

De ces deux expériences, nous déduisons qu'aucune technique n'est parfaite pour toutes les situations. Dans un environnement peu dense, Arrows2D ou Halo3D peuvent être les plus adaptées. Dans un environnement très dense, Halo3D peut induire en erreur si trop d'agrégations sont effectuées, Halo3D* peut afficher trop de chevauchements à l'écran et Halo3D** peut dessiner une seule courbe qui longe tous les bords de l'écran si les POIs sont très étalés. Dans ce contexte, une piste de recherche serait d'adapter automatiquement la technique de visualisation à l'environnement hors-champ : se poserait alors le problème de changement de visualisation et celui du transfert d'expertise d'une technique à l'autre.

PARTIE II

TECHNIQUES DE POINTAGE EN RA

6

Sommaire

6.1	Introduction	97
6.2	Le besoin en RA mobile	99
6.3	Modélisation du pointage	100
6.3.1	Modélisation 2D	100
6.3.2	Modélisation 3D	101
6.4	Assistance au pointage 2D	102
6.4.1	Pointage direct	102
6.4.2	Pointage indirect	105
6.5	Assistance au pointage en Réalité Augmentée mobile	110
6.5.1	Pointage direct	110
6.5.2	Pointage indirect	112
6.5.3	Stabilisation du flux d'entrée	114
6.6	Mise en contexte de l'interaction	115
6.7	Synthèse et conclusion	116

6.1 Introduction

Les chapitres de la partie précédente traitent du guidage vers des cibles numériques, les points d'intérêts (POIs). De façon complémentaire à ce travail, nous nous intéressons dans cette partie (chapitre 6, chapitre 7, chapitre 8) au pointage d'un POI affiché à l'écran en Réalité Augmentée (RA) mobile.

Le pointage est une action élémentaire utilisée dans une grande variété de tâches (sélectionner une zone, une icône, une option dans un menu, réaliser un glisser-déposer, etc.). Cette action est également commune à divers contextes tels que les ordinateurs de bureau, les appareils mobiles tactiles, les environnements virtuels, etc. En fonction des contextes, le moyen d'effectuer une sélection diffère : la souris pour les ordinateurs, le doigt sur les écrans tactiles, le curseur dans les environnements virtuels.

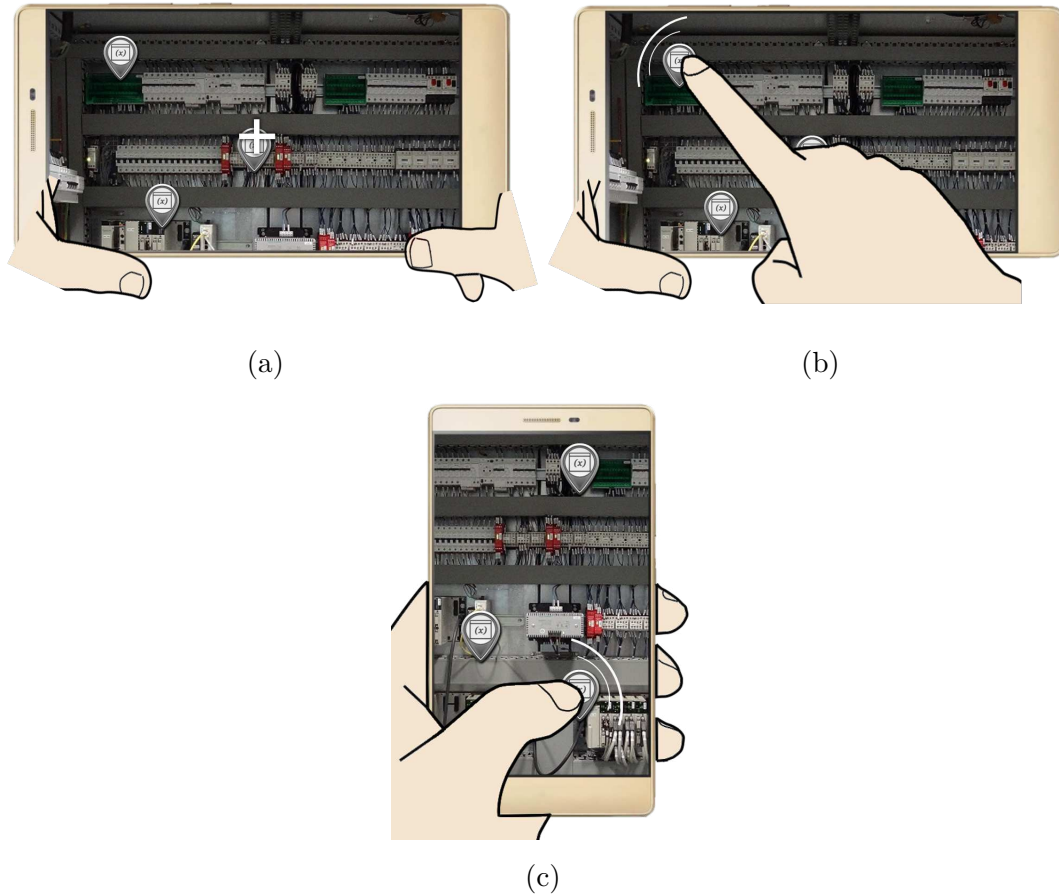


FIGURE 6.1 – (a) Pointage indirect : les deux mains tiennent le dispositif mobile et la sélection s’effectue en alignant la croix centrale avec la cible. (b) Pointage direct : une seule main tient le dispositif mobile et la sélection s’effectue en touchant la cible avec un doigt de l’autre main. (c) Pointage direct : la main qui tient le dispositif mobile sert aussi à interagir. La sélection s’effectue alors avec le pouce de la main qui tient le dispositif.

Un geste de pointage s’effectue par l’intermédiaire d’un point actif, contrôlé par l’utilisateur, pour interagir avec le système. Ce point actif est défini par une position dans l’espace d’affichage. Sur ordinateur, il peut être représenté par un curseur (flèche, croix centrale, main, etc.). Sur écrans tactiles, aucun curseur ne représente le point actif. L’interaction se fait à l’aide d’un doigt ou d’un stylet. Le contact du curseur ou du doigt avec une cible crée ainsi une interaction permettant à l’utilisateur des actions tels que la sélection, le déplacement, le dessin etc. La réalité augmentée rassemble les deux types de points actifs : la croix centrale permet de pointer tout en tenant l’appareil mobile des deux mains. L’utilisateur contrôle le curseur et doit l’aligner avec une cible pour la sélectionner (pointage indirect, voir Figure 6.1a). Une sélection avec un doigt ou un stylet permet de pointer une cible en la touchant sur l’écran (pointage direct, voir Figure 6.1b et Figure 6.1c). Ce type de pointage reprend les codes de l’interaction tactile, largement utilisée avec les téléphones mobiles.

Malgré les différentes manières de pointer une cible, deux facteurs sont liés à la difficulté d’un geste de pointage : (1) la taille de la cible à atteindre et (2) la longueur du geste à effectuer (voir section 6.3). D’autres facteurs existent selon la manière de pointer comme

le phénomène du fat-finger (occultation de la cible par le doigt qui sert à pointer [88]) propre au pointage direct. La performance d'un geste de pointage est donc défini par deux grandeurs : la précision et la durée du geste. Cette définition s'applique également en RA mobile, où l'utilisateur peut sélectionner un POI soit avec le doigt, soit à l'aide d'un curseur affiché sur l'écran. Un de nos objectifs est d'augmenter les performances du geste de pointage d'un POI en RA mobile.

Dans ce chapitre, nous présentons une revue des techniques d'assistance au pointage. Nous rappelons d'abord les éléments principaux de modélisation du geste humain vers une cible. Nous présentons ensuite un aperçu des techniques de pointage 2D, en distinguant celles définies pour optimiser le pointage tactile de celles basées sur un curseur. Puis nous présentons les techniques de pointage en RA mobile, là encore en distinguant les solutions basées sur un pointage tactile avec un doigt ou un stylet, de celles basées sur un curseur centré à l'écran.

Avant de présenter ces techniques de pointage (sur supports mobiles et en RA mobile), nous énonçons les critères recherchés pour le pointage en RA mobile.

6.2 Le besoin en RA mobile

Tout d'abord, les difficultés pour pointer une cible en RA mobile sont héritées de celles liées au pointage sur supports mobiles : sur écrans tactiles, l'interaction directe avec le pouce de la main qui tient le dispositif peut rendre inaccessibles certaines zones à l'écran en dehors de la zone fonctionnelle du pouce [18, 33] (voir Figure 6.1c) ou dans le cas général de l'interaction avec un doigt cacher la cible souhaitée à l'écran, empêchant un pointage précis (fat finger) [88]. Lorsque l'appareil utilisé est plus grand, comme une tablette ou un smartphone, le maintien de l'appareil d'une main pour gérer la sélection de cibles de l'autre main augmente l'instabilité du geste de pointage [91]. De plus, et spécifique à la RA mobile, l'instabilité des cibles à l'écran due aux problèmes de suivi de mouvement et d'enregistrement du contenu numérique sur les objets physiques [107] augmente la difficulté du pointage.

Les techniques d'assistance au pointage permettent de réduire la difficulté de la tâche, soit en agrandissant les cibles, soit en diminuant la distance nécessaire pour les atteindre, soit les deux (comme expliqué dans la section 6.3). Cependant, les appareils couramment utilisés en RA mobile ont une taille d'écran limitée. Il convient donc d'éviter les techniques d'assistance au pointage qui introduisent une intrusion visuelle supplémentaire à l'écran, pour ne pas surcharger la zone de travail de l'utilisateur. De plus, les éléments numériques affichés à l'écran contiennent des informations qui augmentent le monde physique. Chaque POI offre donc des informations pertinentes par rapport à sa localisation dans le monde physique. Il est important de ne pas briser le lien qui unit monde physique et monde numérique, pour conserver la sémantique propre aux informations affichées.

De plus, pointer sur des cibles numériques (POIs) peut être une étape intermédiaire pour accéder à des informations supplémentaires. C'est le cas de widgets numériques, comme un menu, enregistrés sur un objet physique (voir Figure 2.3). Dans le cas de la maintenance industrielle (voir section 2.3), un POI regroupe une ou plusieurs informations sur l'état de la machine (température, pression, documentation technique, etc.). Ces éléments sont regroupés au sein d'un menu, visible uniquement lorsque le POI a été sélectionné. Accéder à une information technique nécessite donc (1) de pointer le POI correspondant et le sélectionner pour l'ouvrir, et (2) de pointer l'information souhaitée au sein du menu affiché et la sélectionner pour la consulter. Il est donc pas uniquement question de sélectionner une cible, mais aussi d'interagir avec des éléments numériques liés à la cible.

Nous déduisons trois critères pour notre étude des techniques d'aide au pointage de POIs en RA mobile : (1) minimiser l'intrusion visuelle pour éviter de surcharger la zone centrale de l'écran (critère identifié dans la partie 1 de ce manuscrit), (2) ne pas briser le lien qui unit les cibles numériques à leur environnement physique, et (3) considérer la sélection d'informations après avoir sélectionné une cible.

6.3 Modélisation du pointage

6.3.1 Modélisation 2D

La loi de Fitts modélise le temps de réalisation d'une tâche de pointage (MT) à l'aide de deux facteurs : la distance à la cible (D) à partir du point de départ et la largeur de la cible (W). Le temps est linéaire en un indice de difficulté (ID) de la tâche. ID est une fonction de D et W [41, 101] :

$$MT = a + b * ID$$

où a et b sont des constantes positives et

$$ID = \log_2(1 + D/W)$$

Selon la loi de Fitts, réduire le temps nécessaire pour effectuer un geste de pointage revient à réduire l'indice de difficulté ID de cette tâche : cela équivaut à, soit réduire la distance à la cible D, soit à augmenter la largeur de la cible W, soit les deux.

Le *Modele de Corrections itératives* de Crossman et al [36] suppose qu'un geste de pointage d'un point de départ vers une cible est composé d'une série de sous-mouvements discrets effectués sur la base d'une rétroaction sensorielle. L'indice de difficulté pour atteindre la cible diminue linéairement car chaque sous-mouvement se rapproche de 50% de la cible. Le modèle est cependant incomplet car il ne considère pas le bruit neuro-moteur (perturbations aléatoires dues à l'activité nerveuse), les mouvements ciblés rapides ou encore les erreurs (cibles manquées) [75].

Le *Modèle de l'Impulsion Initiale Optimisée* [75] décompose le geste de pointage en deux sous-mouvements : un premier sous-mouvement balistique de grande amplitude et rapide et, si la cible est manquée, des sous-mouvements correctifs de faibles amplitudes et plus lents. Lors de la phase balistique, l'utilisateur effectue un premier geste vers la cible, sans rétro-contrôle. Si le geste n'est pas arrivé sur la cible, plusieurs gestes de correction sont nécessaires. Il s'agit de la phase correctrice. Cette phase est réalisée en boucle fermée : l'utilisateur observe le résultat de son action, puis intègre la réaction du système à sa planification de mouvements, lui permettant ainsi d'affiner la précision de son geste.

6.3.2 Modélisation 3D

Rohs et al. [94] ont proposé un modèle en deux parties, pour caractériser une interaction en réalité augmentée mobile avec un curseur affiché au centre de l'écran. Dans ce modèle, la tâche de pointage est décomposée en 2 phases : (1) une première phase nommée pointage physique, où la cible n'apparaît pas sur l'écran mais est directement observable dans le monde physique. A un instant donné pendant le geste de pointage, la cible passe derrière l'écran et n'est donc plus observable à l'œil nu. Pour une largeur d'écran S , le point intermédiaire est donc situé à une distance de $S / 2$. C'est là que débute la deuxième phase de pointage intitulée pointage virtuel. (2) une phase de pointage virtuel où l'utilisateur aligne la cible, maintenant visible à l'écran, avec le curseur central. Pour une largeur d'écran S , le temps de réalisation d'une tâche de pointage est alors défini :

$$MT = a + b * \log_2(D/S + 1) + c * \log_2(S/2/W + 1)$$

En 2011, Rohs et al. [95] ont montré que ce calcul du temps de réalisation en deux parties pouvait être appliqué à des cibles du monde réel placées autour des utilisateurs. Les tâches consistaient à pointer les bâtiments de l'environnement physique avec un curseur centré sur l'écran.

Pour des tâches de pointage 3D en raycasting, Wingrave et al. [113] ont adapté la loi de Fitts. Les variables A et W de l'indice de difficulté de la loi de Fitts $ID = \log_2(1 + A/W)$ sont définis comme suit : A désigne la distance angulaire entre le pointeur et la cible tandis que W se réfère à la largeur de la cible sur l'écran du casque de réalité virtuelle. Une étude expérimentale a montré l'adéquation de cette modélisation : l'étude a considéré plusieurs cibles dont la distance varie par rapport à l'utilisateur (4, 10 et 20 mètres), ainsi que leur positions horizontales (+35,+15,0,-15,-35 degrés) et verticales (+35,0,-35 degrés). Faire varier la distance des cibles par rapport à l'utilisateur a ainsi une influence directe sur leur taille W visible sur l'écran du casque de réalité virtuelle, tandis que modifier leurs positions horizontales et verticales impacte la distance angulaire A entre le pointeur et les cibles. Les résultats ont montré que cette variante de la loi de Fitts adaptée au pointage 3D s'applique aux données de l'étude expérimentale.

6.4 Assistance au pointage 2D

Inspiré par les modélisations du geste humain vers une cible en considérant les paramètres influençant la difficulté d'un geste de pointage, les techniques d'assistance au pointage permettent de diminuer le temps de pointage. Sur supports mobiles, ce geste peut être effectué soit à l'aide d'un doigt ou d'un stylet (pointage direct), soit grâce à un curseur (pointage indirect). Selon ces deux types de pointage (direct et indirect), différentes approches d'optimisation du pointage ont été proposées. Nous les détaillons ci-dessous.

6.4.1 Pointage direct

Sur écrans tactiles, l'interaction directe avec le doigt soulève trois problèmes de nature différente [17] : (1) le doigt peut masquer la cible souhaitée à l'écran, empêchant un pointage précis, (2) la zone de contact du doigt à l'écran mène à une sélection ambiguë et (3) l'interaction avec le pouce lors de l'utilisation du dispositif mobile à une seule main est réduite à la zone fonctionnelle du pouce [18, 33] rendant inaccessible certaines zones de l'écran. Deux catégories de techniques permettent de pallier à ses problèmes : les techniques "target-agnostic" se réfèrent aux systèmes dont la connaissance préalable de l'existence des cibles n'est pas requise. Ceci fait opposition aux techniques "target-aware", qui nécessitent de connaître la position des cibles dans l'espace pour fonctionner.

Techniques "target-agnostic". La technique TapTap de [96] et al. est une technique qui améliore la précision de pointage en introduisant deux phases : un premier tap sur l'écran définit une région d'intérêt. Cette zone est ensuite agrandie, permettant aux utilisateurs de mieux percevoir l'emplacement de la cible souhaitée. Par conséquent, un deuxième tap permet la sélection précise de la cible à l'écran. Roudaut et al. ont également proposé MagStick [96], un curseur magnétisé contrôlé par un manche télescopique. Comparés au toucher direct simple, des résultats expérimentaux ont montré que ces deux techniques conduisaient à moins d'erreurs de pointage.

Similaire à TapTap [96], ZoomTap [57] empêche l'occultation visuelle en affichant une vue agrandie de la zone sélectionnée par l'utilisateur. Cependant, cette vue zoomée ne se ferme pas tant que l'utilisateur n'a pas tapé à l'extérieur de la zone, permettant ainsi la sélection de multiples cibles. Une vue zoomée de l'environnement est également beaucoup plus sensible au tremblement des mains, surtout lorsque le dispositif mobile est tenu à une seule main.

Une autre approche consiste à déplacer la zone cachée sous le doigt vers un emplacement non caché : la sélection Take-Off (aussi appelé Offset cursor)[90] place le curseur à 1.27 cm (0.5 pouce) au-dessus du doigt pour éviter toute occultation. Cependant, les utilisateurs ne sont pas en mesure de connaître la position exacte du curseur jusqu'à ce qu'un premier contact avec l'écran soit établi. Par conséquent, il nécessite un mouvement correctif pour placer avec précision le curseur sur la cible. Shift [109] étend le pointage tactile direct en

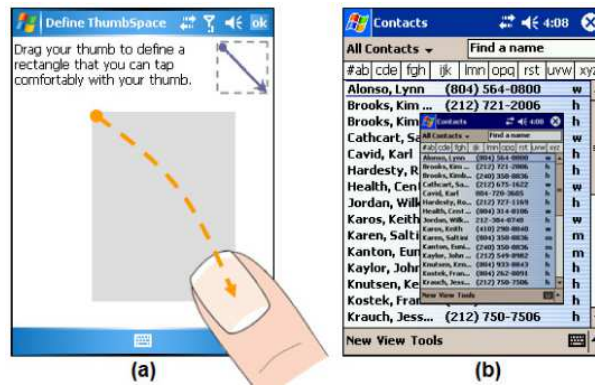


FIGURE 6.2 – Technique de sélection Thumbspace a) configuration d’une zone qui servira de touchpad, (b) miniature de l’interface affichée dans Thumbspace. Image issue de [66].

affichant une extension visuelle au-dessus du doigt : le curseur reste sous le doigt, seule la vue cachée par le doigt est décalée. Ainsi, les utilisateurs peuvent facilement viser la cible et n’ont pas à prédire la position du curseur.

Thumbspace [66] propose une solution pour pallier au manque d’accessibilité du pouce aux sections supérieures de l’écran : Thumbspace permet de définir une région de l’écran qui servira de touchpad (voir Figure 6.2a). Lorsque cet espace est activé, la région affiche une vue miniature de l’écran (voir Figure 6.2b). L’utilisateur peut ainsi sélectionner un objet sans avoir à interagir en haut de l’écran. L’objet en question est mis en surbrillance dans la fenêtre principale. Ainsi, une action dans le Thumbspace a un effet direct sur l’interface graphique originale. L’utilisateur peut ajuster sa sélection en effectuant du pointage relatif : avant de lever le doigt de l’écran pour valider la sélection, l’utilisateur a la possibilité de manipuler le curseur en déplaçant son pouce. Similaire à Object Pointing [53], le curseur saute sur la cible la plus proche selon la direction du mouvement effectué par le doigt. Le pointage relatif permet donc de limiter le nombre de mouvements à effectuer pour sélectionner une cible sur l’écran, et peut s’appliquer à la RA mobile. Bien que réduisant la distance nécessaire à l’acquisition d’une cible, afficher une miniature de l’interface reste très intrusif à l’écran.

Enfin, BezelCursor [71] propose une autre approche pour pallier au manque d’accessibilité du pouce sur les écrans tactiles. La technique permet la sélection de n’importe quel objet sur l’écran à l’aide d’une seule action fluide, décomposée en deux parties : un premier geste pour activer le pointeur virtuel, et un deuxième geste pour contrôler ce même pointeur. Un curseur apparaît à l’écran et peut être contrôlé grâce aux mouvements du pouce. Ainsi, le pointage relatif de BezelCursor utilise l’écran tactile comme un trackpad. Le curseur sélectionne l’élément en dessous et se désactive lorsque le pouce se lève de l’écran.

Techniques ”target-aware”. Burst [57] permet la sélection de multiples cibles avec un seul doigt tout en minimisant les problèmes d’occultation. Burst [57] sélectionne les six cibles les plus proches du point de contact avec l’écran, et les organise en anneau autour du doigt de l’utilisateur. Afin de sélectionner plusieurs cibles, l’utilisateur maintient son

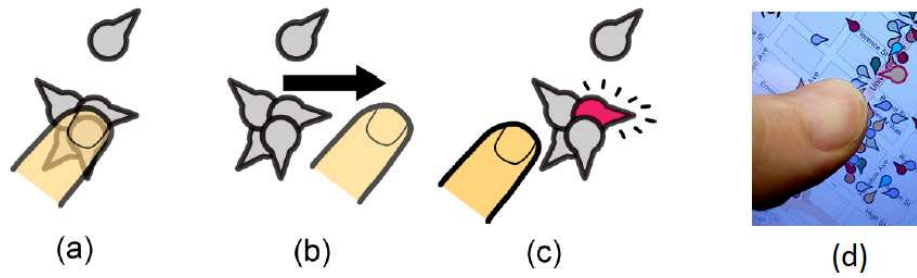


FIGURE 6.3 – Technique de sélection Escape. (a) L'utilisateur appuie son doigt près de la cible souhaitée. (b) L'utilisateur effectue un geste dans la direction indiquée par la cible. (c) La cible est sélectionnée, malgré la forte densité de POIs. (d) Exemple d'application sur écran tactile. Image issue de [119].

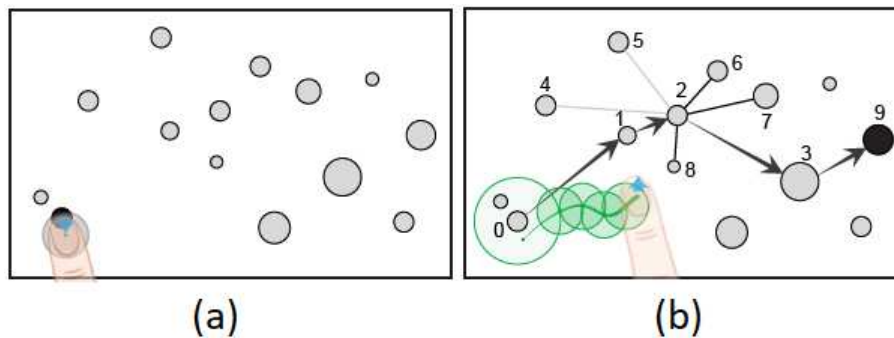


FIGURE 6.4 – Technique de sélection 2DDragger : (a) la cible la plus proche du doigt de l'utilisateur est pré-sélectionnée, (b) l'utilisateur passe de cible voisine en cible voisine en faisant glisser son doigt sur l'écran. Image issue de [102].

doigt appuyé sur l'écran et le fait glisser sur les cibles. Le menu circulaire disparaît lorsque l'utilisateur relâche le doigt. Cependant, le menu circulaire ne permet plus de représenter les cibles dans leur contexte spatial. En RA mobile, chaque POI correspond à un emplacement physique particulier. Les déplacer brise le lien qui unit le monde numérique et le monde physique.

Dans le cas d'une forte densité de POIs à l'écran, Escape [119] facilite la sélection d'une cible. Chaque cible est affichée à l'écran avec une information de direction (voir Figure 6.3d). Lorsque l'utilisateur souhaite sélectionner une cible parmi un ensemble de cibles proches, il pose son doigt près de celle-ci (voir Figure 6.3a) et effectue le geste dans la direction affichée par la cible (voir Figure 6.3b). Le système détecte le geste et sélectionne la cible correspondante (voir Figure 6.3d). Outre l'intrusion visuelle supplémentaire provoquée par la nécessité d'afficher une information de direction pour chaque cible, Escape [119] fonctionne dans le cas où les cibles sont petites et nombreuses.

Avec 2DDragger [102], Su et al. proposent une technique de pointage sur des écrans tactiles permettant d'accéder facilement à de petites cibles, qu'elles soient distantes ou situées dans des régions denses. Poser le doigt sur l'écran pré-sélectionne automatiquement la cible la plus proche du point de contact (voir Figure 6.4a). En faisant glisser le doigt sur l'écran, l'utilisateur passe de cible voisine en cible voisine en fonction de la direction

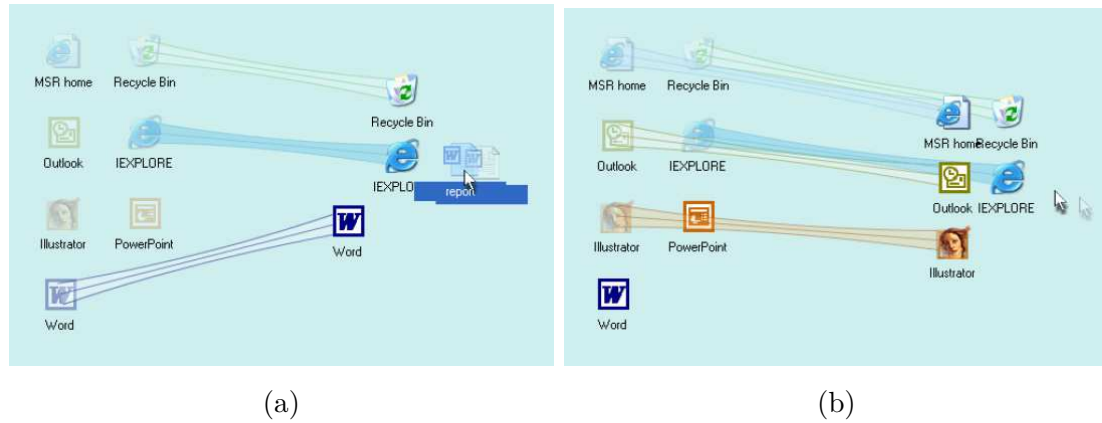


FIGURE 6.5 – a) Technique de sélection Drag and pop : toutes les icônes potentielles sont ramenées près du curseur, b) Technique de sélection Drag and pick : toutes les icônes situées dans la direction du déplacement de la souris sont ramenées près du curseur. Images issues de [12].

de son mouvement (voir Figure 6.4b). Une fois la cible souhaitée atteinte, relâcher le doigt de l'écran permet de valider la sélection. 2DDragger est donc insensible à la distribution et à la taille des cibles. Cette technique répond parfaitement aux besoins des terminaux mobiles en palliant à la zone fonctionnelle restreinte du pouce sur l'écran. 2DDragger prend également en compte les besoins en RA, où de nombreux POIs peuvent être distribués en paquets sur l'écran. Cependant, la technique proposée traite uniquement de la sélection d'une cible, et rien ne mentionne l'accès aux informations qui pourraient être liées à cette cible.

6.4.2 Pointage indirect

Basé sur la loi de Fitts (section 6.3), réduire l'amplitude nécessaire pour atteindre la cible, et/ou augmenter la taille de la cible constituent deux approches pour diminuer le temps du geste de pointage [9]. Nous détaillons des techniques selon ces approches dans cette section. Les techniques décrites sont intrinsèquement "target-aware" (une exception non décrite est la technique "angle mouse" qui est "target-agnostic" [115]).

6.4.2.1 Réduction de l'amplitude nécessaire à l'acquisition d'une cible

Pour réduire la distance du point de départ à la cible, une solution est de déplacer les cibles afin de les ramener près du curseur. Drag-and-Pop [12] est une extension de l'interaction glisser-déposer. La technique ramène près du curseur les icônes potentielles en fonction du fichier que l'utilisateur a commencé à déplacer sur l'écran. A titre d'exemple, le déplacement d'un fichier textuel ramènera près du curseur des icônes du type Word, NotePad, etc (voir Figure 6.5a). Drag-and-Pick [12] modifie l'interaction Drag-and-Pop en ramenant près du curseur toutes les icônes situées dans la direction du déplacement (voir Figure 6.5b). Ainsi, un relâchement du curseur sur une icône, un dossier etc. activera l'application associée comme si elle avait été double cliquée. Ces techniques permettent un pointage plus rapide, mais ne sont pas applicables à la RA mobile : chaque POI ap-

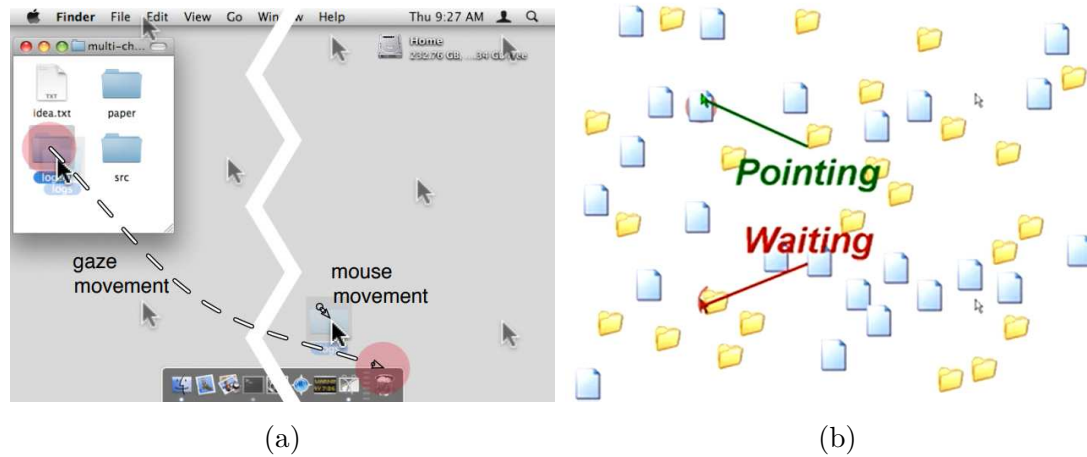


FIGURE 6.6 – Grille de curseurs affichés à l'écran avec a) Technique de sélection Rake Cursor : sélection du curseur actif grâce au regard (cercle rouge). Image issue de [21]. b) Technique de sélection Ninja Cursor : sélection du curseur comme celui le plus proche du centre d'une cible. Les autres curseurs sont mis en attente. Image issue de [68].

porte une ou plusieurs informations par rapport à une localisation précise dans le monde physique. Les déplacer casserait le lien qui unit monde numérique et monde physique. Drag-and-Pop et Drag-and-Pick conservent un lien visuel entre la position originale d'une icône et sa nouvelle localisation, mais ceci augmente considérablement l'intrusion visuelle des techniques à l'écran.

Une autre solution pour réduire l'amplitude A consiste à modifier le déplacement du curseur. Object Pointing [53] suppose qu'il n'est pas nécessaire d'acquérir de l'information en continu de la souris quand seules des informations discrètes sont requises. Object Pointing propose ainsi d'éliminer l'espace vide entre les cibles : le curseur se déplace de cible en cible, dans la direction du mouvement de la souris. Bien que cette technique réduise le temps de pointage, elle est difficilement applicable à la RA. En RA mobile, le curseur est déplacé conjointement avec le mouvement de l'appareil mobile. Avec Object Pointing, le curseur se déplacerait automatiquement vers les nouvelles cibles apparaissant à l'écran. Cependant, la sélection de cibles se ferait toujours de manière dynamique, en déplaçant l'appareil mobile, et peut donc augmenter la fatigue engendrée par le port du téléphone ou de la tablette. Cela introduirait également une grande variabilité visuelle à l'écran, le curseur se déplaçant à chaque mouvement de l'utilisateur. Cette variabilité visuelle s'ajouterait à celle induite par la réalité augmentée comme le flux vidéo provenant de la caméra, les POIs qui s'ouvrent/ferment etc.

Enfin, multiplier le nombre de curseurs à l'écran est également une possibilité pour réduire l'amplitude A . Avec Rake Cursor [21], Blanch et Ortega couplent deux entrées pour améliorer la tâche de pointage : la souris et la position du regard. Rake Cursor affiche ainsi à l'écran une grille de curseurs contrôlée par la souris (voir Figure 6.6a). La sélection du curseur actif est réalisée par le regard. Le temps de pointage diminue, car la distance nécessaire pour atteindre la cible est réduite. Ninja Cursor [68] utilise également

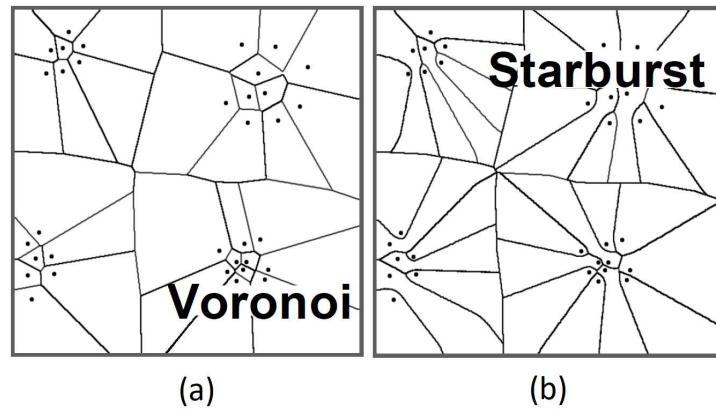


FIGURE 6.7 – Partitionnement de 4 clusters (34 POIs) à l’aide de a) Voronoi (b) Starburst. Image issue de [15].

une grille de curseurs à l’écran pour réduire la distance de pointage (voir Figure 6.6b). Cependant aucun système de suivi de regard n’est ici utilisé. Par conséquent, il existe une ambiguïté pour déterminer le curseur actif lorsque deux curseurs pointent sur des cibles simultanément. Pour lever cette ambiguïté, Ninja Cursor s’appuie sur un algorithme qui définit le curseur le plus proche du centre d’une cible comme étant le curseur actif. Les autres curseurs concurrents sont replacés en dehors de leur cible et mis dans une file d’attente. Lorsque le curseur actif s’éloigne de sa cible, le premier curseur de la file d’attente est retiré de la queue. Il peut alors se placer à l’intérieur de la cible pour laquelle il avait été mis en attente.

Dans les deux cas, l’augmentation du nombre total d’éléments visuels sur l’écran et la recherche du curseur actif peut conduire à une distraction visuelle et à une augmentation du temps de pointage [103]. En RA mobile, afficher une grille de pointeurs sur un petit écran, par-dessus la vue temps réel de la caméra et les POIs surchargerait l’écran et pourrait conduire à une distraction visuelle importante.

6.4.2.2 Agrandissement des cibles

Expansion de cibles

Acquérir une petite cible sur un écran peut être une tâche difficile, de part la précision nécessaire pour aligner le curseur avec l’élément souhaité. Les techniques d’expansion de cibles facilitent le pointage en agrandissant la zone de sélection des cibles. Elles reposent sur un partitionnement de l’espace en cellules, de sorte que (1) chaque cellule contient une seule cible et (2), tous les pixels d’une cellule soient plus proches de la cible correspondante que de toute autre cible. Cela permet également de diminuer le risque que le curseur sorte de la zone de sélection de la cible lors de la validation de la sélection. L’utilisateur n’ayant plus besoin d’aligner le curseur avec la cible, il peut valider la sélection n’importe où à l’intérieur de la cellule contenant l’élément souhaité.

Deux éléments caractérisent les techniques d’expansion de cibles : l’algorithme d’expansion et l’aide visuelle associée.

La tessellation de Voronoi [42] est un exemple d'algorithme de partitionnement de l'espace qui maximise l'utilisation de l'espace vide entre les cibles (voir Figure 6.7a). La distance à parcourir pour sélectionner une cible est plus courte, la fatigue engendrée par la tâche de pointage est moindre. Cependant, lorsque les cibles sont regroupées en grappes denses aussi notées *clusters*, les cibles du cluster deviennent difficilement sélectionnables : leur cellule est de petite taille à cause de la densité de cibles, et l'avantage de l'expansion de cibles est grandement diminué (voir Figure 6.7a). Pour traiter ce problème, Baudish et al. [15] ont proposé Starburst. Starburst alloue, pour chaque cible, de l'espace à la périphérie du cluster (voir Figure 6.7b). Le pointage s'effectue donc autour du cluster, sans avoir à amener le curseur à l'intérieur du cluster.

Pour indiquer à l'utilisateur dans quelle cellule se trouve le curseur, il est nécessaire d'offrir une aide visuelle. Cette aide est essentielle pour tirer pleinement parti des zones étendues. Guillon et al. [55] proposent 3 axes pour caractériser l'aide visuelle des techniques d'expansion de cibles : (1) l'axe dynamique définit la vitesse de mise à jour de l'aide visuelle (statique, discrète ou continue), (2) l'axe d'observabilité détermine si la technique affiche les limites de la zone étendue (explicite) ou si elles ne les affichent pas (implicite), et enfin (3) l'axe des éléments augmentés décrit quel élément parmi le curseur, la cible ou l'espace entre les cibles est visuellement augmenté pour fournir aux utilisateurs des retours sur les cibles étendues. Une comparaison entre différentes techniques exploitant les 3 axes [55] montre que de meilleures performances sont obtenues lorsque la technique est implicite, discrète et basée sur les cibles. Cette technique agrandit la cible contenue dans la cellule de Voronoi où se trouve le curseur, mais n'affiche pas la cellule à l'écran : elle est donc qualifiée d'implicite. Le retour visuel s'effectue uniquement lorsque le curseur rentre dans une cellule de Voronoi. C'est donc une technique discrète. La technique consiste à mettre en évidence la cible. Notons que cette technique implicite, discrète et basée sur les cibles consiste à mettre en évidence la cible la plus proche du curseur (le calcul de la partition de Voronoi étant alors inutile pour son implémentation).

Curseurs étendus

Les curseurs étendus permettent d'optimiser le geste de pointage en utilisant un curseur avec une plus grande région d'activation, au lieu d'un pointeur traditionnel dont la zone d'activation est un point. Les curseurs étendus sont souvent distingués des techniques d'expansion de cibles dans les états de l'art sur le pointage [9, 54]. La taxonomie de Guillon et al. [55] les unifie en considérant qu'il s'agit d'une aide visuelle qui augmente le curseur et non la cible.

Bubble Cursor [44] est un type de curseur qui re-dimensionne dynamiquement sa zone d'activation en fonction de la proximité des cibles. Cela augmente virtuellement la taille de la cible pour exploiter l'espace vide autour d'elle. Le curseur s'agrandit de sorte qu'une seule cible ne puisse être sélectionnée à la fois (voir Figure 6.8). En augmentant la taille du curseur, la distance nécessaire pour acquérir une cible est réduite. Les résultats des tâches d'acquisition de cibles en 1D et 2D ont montré que le Bubble Cursor réduisait

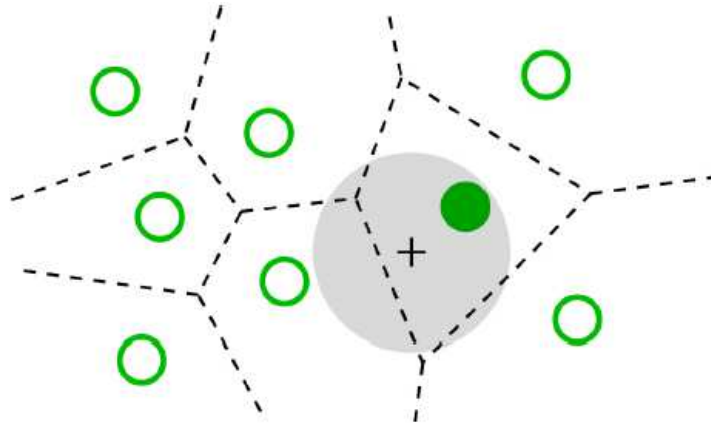


FIGURE 6.8 – Technique de sélection Bubble Cursor : agrandissement circulaire du curseur de sorte qu’une seule cible ne soit sélectionnable à la fois. Image issue de [44].

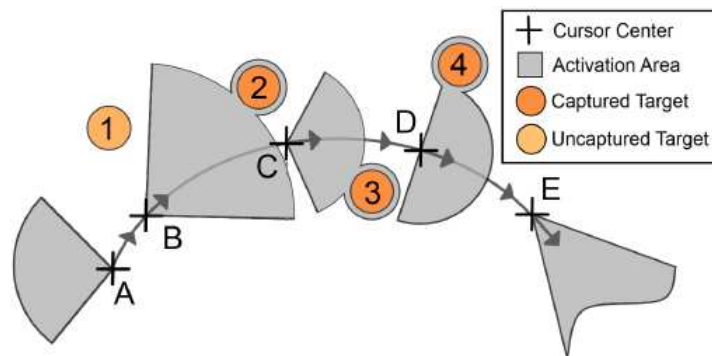


FIGURE 6.9 – Technique de sélection Fan Cursor : la vitesse et la direction du mouvement du curseur modifie la taille et l’orientation de l’éventail. Image issue de [103].

efficacement les temps de pointage par rapport à un curseur classique et à la technique Object Pointing [53]. Plusieurs techniques ont été inspirées par Bubble Cursor. Bubble Lens [79] se comporte comme un Bubble Cursor lorsque les cibles sont largement espacées. Mais en présence de fortes densités de cibles, Bubble Lens se comporte comme une loupe. Les cibles étant grossies, l’utilisateur peut réaliser un geste de pointage plus précis. La loupe est affichée automatiquement en analysant le mouvement du curseur, sur la pente descendante du premier sous-mouvement correctif (voir section 6.3). Comparée à Bubble Cursor, Bubble Lens réduit le temps de pointage de 10,2%, avec 37,9% plus de précision.

Fan Cursor [103] est une autre technique exploitant la vitesse du mouvement. Avec Fan Cursor, la zone d’activation se re-dimensionne dynamiquement selon la vitesse du mouvement. Fan Cursor permet ainsi de pointer précisément des cibles proches à grande vitesse et des cibles lointaines à faible vitesse. Fan Cursor est un curseur en forme d’éventail qui utilise la direction et la vitesse du curseur pour transformer sa zone de sélection (voir Figure 6.9). Plus le curseur est rapide, plus l’éventail est large. La longueur du curseur est calculée de manière à ce qu’il touche la cible désignée. Une première étude expérimentale a été menée pour comparer trois styles d’affichage pour le curseur : en éventail, circulaire et implicite (uniquement le réticule). La cible sélectionnée a été mise en surbrillance quel que soit le style. Les résultats ont montré que les utilisateurs étaient plus rapides et plus

précis lorsqu'ils utilisaient le style implicite. En effet, le changement constant de la taille du curseur provoque une forte distraction visuelle et ralentit les mouvements.

Les techniques de curseur étendu permettent un gain de performance (temps de pointage et précision) sans modifier la position ou la taille visible des cibles. Seul le curseur change dynamiquement de taille, explicitement ou implicitement. Lorsque le changement est explicite (bubble cursor ou bubble lens), la modification constante de la taille du curseur provoque une intrusion visuelle à l'écran importante entraînant une distraction visuelle [103], une intrusion importante à l'écran et est visuellement dérangeant. Lorsque l'écran utilisé est restreint, comme celui d'appareils mobiles ou des casques de RA/RV, toute perturbation visuelle doit être évitée. Si le changement de taille du curseur s'effectue de manière implicite, seul le visuel de la cible désignée par le curseur est mis en surbrillance. Comme pour les techniques implicites d'expansion de cibles, l'intrusion est alors minimale et convient parfaitement aux écrans de petite taille, comme en RA mobile. En revanche, nous nous intéressons également à la mise en contexte d'une tâche de pointage, telle que la sélection d'un item dans un menu ouvert lors de la sélection d'une cible. Ce sujet n'est pas traité par les techniques d'assistance au pointage (voir section 6.6).

6.5 Assistance au pointage en Réalité Augmentée mobile

Sur la base du canevas analytique pour la conception de techniques d'interaction en RA mobile défini dans [108], le pointage en RA mobile peut être effectué soit de manière directe (stylo ou doigts nus [89, 96]), soit avec un curseur centré sur l'écran [93, 96]. Pour chacun des deux cas, nous résumons les techniques d'assistance au pointage proposées pour la RA mobile. Nous verrons que ces techniques sont inspirées/adaptées des techniques de pointage que nous avons présentées dans la section précédente.

6.5.1 Pointage direct

Le pointage direct avec les doigts ou à l'aide d'un stylet est l'une des deux modalités de pointage sur les écrans mobiles. En RA, l'écran tactile est utilisé pour pointer sur des objets physiques ou numériques.

Avec Touch Projector, Boring et. al [23] proposent une technique permettant aux utilisateurs d'interagir avec des images sur des écrans distants en touchant directement l'écran de leur appareil mobile. L'entrée tactile est projetée sur la cible distante, comme si elle s'y était produite. Cependant, le pointage direct conduit à une ambiguïté car le contact entre le doigt et l'écran n'est pas un point mais plutôt une surface. Cela peut occasionner des erreurs lors de la sélection d'objets précis dans la scène.

Dual-Finger Midpoint RayCasting et Dual-Finger Offset RayCasting [104] sont des techniques de pointage RA inspirées par les interactions à deux doigts proposées par Benko

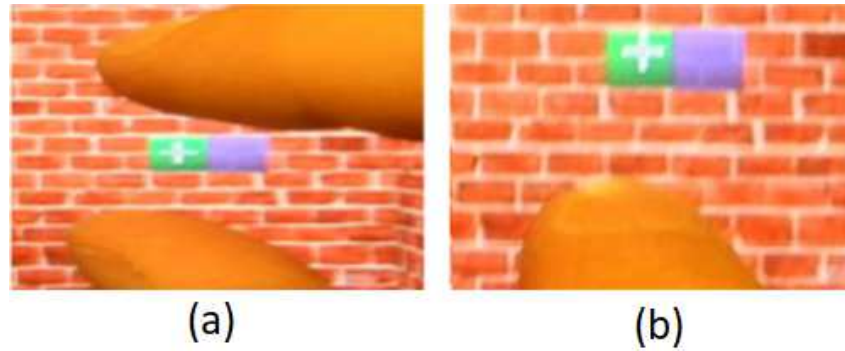


FIGURE 6.10 – (a) Technique de sélection Dual-Finger Midpoint RayCasting : un curseur est affiché au milieu des deux doigts posés sur l'écran. (b) Technique de sélection Dual-Finger Offset RayCasting : le curseur est affiché au-dessus du doigt posé sur l'écran. Image issue de [104].

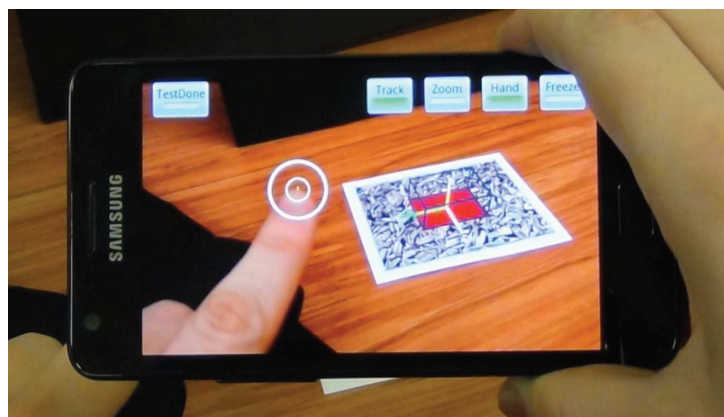


FIGURE 6.11 – Technique d'interaction gestuelle : le doigt est reconnu par la caméra, et le contact du doigt avec la surface est désigné par un cercle. Image issue de [7].

et al [17]. La technique Midpoint affiche un pointeur en forme de croix au barycentre des deux doigts posés sur l'écran (voir Figure 6.10a). Avec la technique Offset, un seul doigt est nécessaire pour pointer dans l'environnement de RA. Le pointeur virtuel affiché sur l'écran est positionné au-dessus du doigt avec un léger offset (voir Figure 6.10b). Les deux techniques se comportent ensuite comme un ray-casting classique, évaluant l'intersection entre la croix et un élément numérique pour déterminer l'élément pointé. Comme les techniques de la sous-section 6.4.1, l'objectif de ces deux techniques est de faciliter la sélection des cibles petites et partiellement cachées sous la zone couverte par le doigt.

Tandis que le ray-casting permet de pointer des objets distants non accessibles directement par la main, une autre forme de pointage direct par interaction gestuelle directe sur les objets à portée de main est possible [7]. Contrairement à l'interaction tactile, les informations saisies proviennent ici de la position des doigts détectée dans le champ de vision de la caméra (voir Figure 6.11). Ce type d'interaction résout le problème d'occultation de la cible par le doigt sur l'écran.

Outre l'occultation des cibles par le doigt lors d'un pointage direct, les éléments numériques peuvent également être cachés par d'autres objets numériques de la scène. Comme la tech-

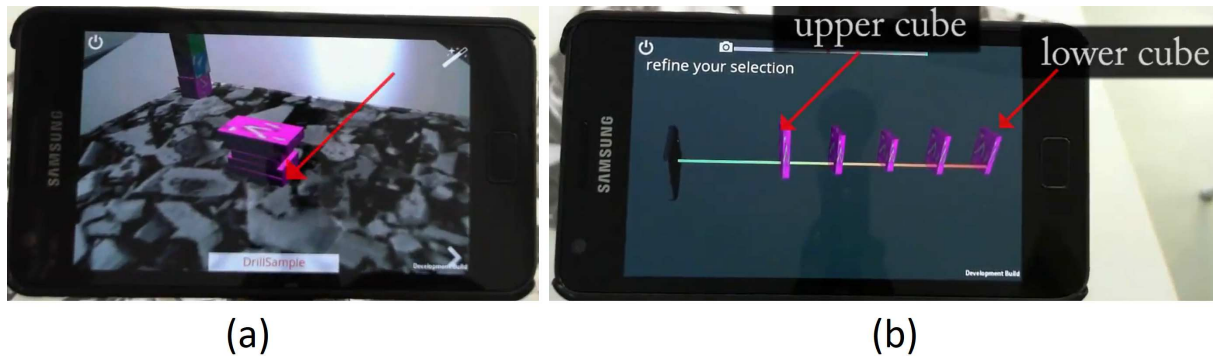


FIGURE 6.12 – Technique de sélection DrillSample : (a) l'utilisateur souhaite sélectionner une cible cachée (le dernier bloc rose). (b) DrillSample affiche une vue "décomposée" pour faciliter la sélection de la cible. Image issue de [78].

nique Burst [57] pour l'occultation des cibles par le doigt (voir sous-section 6.4.1), DrillSample [78] est une technique de pointage RA qui propose une étape supplémentaire de désambiguïsation lorsque l'occultation entre plusieurs objets (voir Figure 6.12a) ne permet pas de déterminer avec précision l'élément numérique pointé : après un premier toucher sur l'écran, la technique propose une vue "décomposée" où les objets numériques cachés sont éloignés les uns des autres (voir Figure 6.12b). Cela permet de lever l'ambiguïté et de sélectionner avec plus de précision des objets fortement cachés ou présentant une similarité visuelle élevée. DrillSample préserve le contexte spatial des objets, en affichant les éléments numériques du plus proche de la caméra à l'élément le plus éloigné. L'utilisateur peut manipuler l'angle de vue (rotation, zoom, etc.) à l'aide de son doigt pour faciliter sa sélection. Comparé aux techniques basées sur le ray-casting ou à Expand [30], une technique décomposant l'ensemble des éléments cachés sans préservation du contexte spatial, DrillSample a obtenu les meilleurs résultats (temps de complétion et précision des sélections) dans les cas où les cibles sont cachées et difficilement différenciables. La technique du ray-casting reste une bonne technique de sélection dans le cas où les objets sont entièrement visibles.

Bien que plus intuitif, le pointage direct nécessite de maintenir l'appareil mobile à une main tandis que l'autre est utilisée pour pointer dans l'environnement de RA. Ce type d'interaction provoque une instabilité, qui rend la tâche de pointage plus ardue. Le pointage indirect est une manière de pallier à cette instabilité.

6.5.2 Pointage indirect

Contrairement au pointage direct sur l'écran, le curseur central n'est pas altéré par des problèmes d'occultation ou de point de contact ambiguë (voir Figure 6.13a). De plus, le curseur autorise le survol des cibles (événement "hover"), état d'interaction manquant avec les appareils tactiles standard actuels.

Le curseur centré sur l'écran et le pointage direct avec les doigts ont été comparés sur des enfants entre 5 et 10 ans [91]. La tâche consistait à sélectionner des cibles dans un

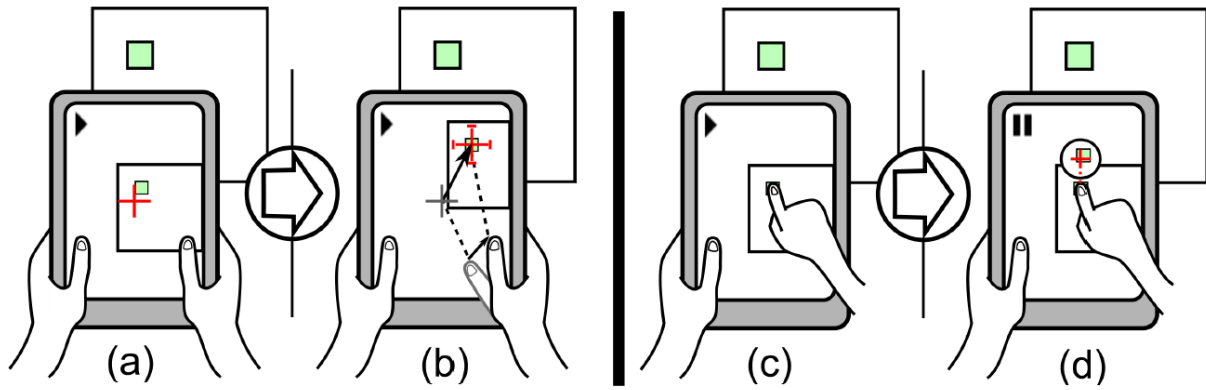


FIGURE 6.13 – Techniques de sélection en réalité augmentée : (a) curseur centré sur l'écran ; (b) Pointage relatif avec curseur stabilisé sur l'objet physique. (c) Pointage direct sur la vidéo ; (d) Shift & Freeze : Shift [109] combiné avec arrêt sur image. Image issue de [108].

jeu en réalité augmentée. Les résultats montrent que la sélection des cibles avec le doigt est beaucoup plus rapide que l'interaction avec le curseur, avec une moyenne de 1,2 s (29 %) de moins par sélection. En fonction du type d'interaction, les utilisateurs emploient différentes compétences cognitives et physiques : l'interaction directe avec les doigts exige que les enfants maintiennent l'appareil stable avec une main, tandis que l'autre main est utilisée pour sélectionner un objet numérique sur l'écran. En revanche, l'interaction avec un curseur oblige les enfants à réorienter précisément l'appareil (probablement avec les deux mains) pour que le curseur soit aligné avec la cible. Afin que cette réorientation soit précise, l'utilisateur doit posséder une bonne compréhension des relations spatiales qui existent entre l'appareil et l'environnement augmenté. Par rapport au pointage direct, l'interaction avec un curseur repose donc davantage sur la compréhension spatiale de l'environnement. Enfin, l'utilisation de l'interaction avec le doigt peut devenir problématique lorsque l'appareil utilisé est plus grand, comme une tablette ou un smartphone [91]. Un appareil trop grand ou trop lourd devient suffisamment encombrant pour qu'il puisse être tenu de manière stable avec une seule main. Ici, l'interaction à l'aide d'un curseur est plus adaptée car elle permet de tenir l'appareil mobile à deux mains tout en laissant la possibilité d'interagir avec l'application.

Une approche pour surmonter l'instabilité du contenu à l'écran consiste à utiliser un curseur stabilisé dans le référentiel de l'objet physique, à savoir la technique du Relative Pointing [108]. Le curseur n'est plus altéré par le tremblement des mains. La modification du point de vue de la caméra modifie automatiquement la position du curseur à l'écran, mais pas dans l'environnement physique. Le curseur est ainsi ancré dans le monde physique et n'est plus fixe au centre de l'écran. La position du curseur peut cependant être modifiée par le déplacement des doigts à l'écran (voir Figure 6.13b). Les résultats d'une étude expérimentale [107] montrent que la technique du Relative Pointing est moins sujet aux erreurs qu'avec un curseur fixe centré sur l'écran. Ces deux techniques peuvent être combinées avec succès : utilisation d'un curseur centré sur l'écran pour un pointage

physique grossier et rapide, puis utilisation du pointage relatif si plus de précision est nécessaire.

6.5.3 Stabilisation du flux d'entrée

Orthogonalement aux techniques de pointage direct et indirect, une autre approche en RA mobile consiste à figer le flux vidéo pour faciliter le pointage. Cette stratégie permet (1) de stabiliser les cibles à l'écran et (2) de travailler avec une pose confortable pour interagir avec le monde augmenté.

Lee et al. [70] ont montré que figer l'image réduit le nombre d'erreurs lors de la réalisation de tâches de dessin RA avec un stylo. De même pour manipuler un cube virtuel, la technique figeant l'image a été comparée à l'interaction directe sur l'écran et à l'interaction gestuelle [7], en bougeant le doigt devant la caméra. La tâche consistait à faire correspondre un cube virtuel à une pose et une position spécifiées. Les résultats ont montré que la technique qui fige l'image de la caméra était préférée et la moins sujette aux erreurs.

Une autre application présentée par Guven et al. [60] est l'annotation de l'environnement augmenté. Pour annoter le monde physique, l'utilisateur fige le flux vidéo provenant de la caméra, annote en créant une étiquette textuelle ou un contenu audio, puis redémarre le flux vidéo. Lorsque le flux vidéo reprend, les étiquettes créées sont interpolées à leur position physique, là où la capture d'écran a été prise. Figer la vidéo permet ici de travailler avec précision, tout en limitant la fatigue physique causée par le maintien de l'appareil mobile. Il convient de noter que cette approche est également employée dans les environnements industriels : une fois l'opérateur de maintenance placé devant la section de la machine souhaitée, figer l'image coupe le flux vidéo provenant de la caméra mais n'arrête pas les informations RA provenant de la machine en temps réel (température, pression, etc.). Cela permet donc à l'utilisateur de consulter les valeurs de contrôle de la machine en temps réel, sans avoir à maintenir le dispositif mobile devant lui.

Cependant, figer l'image pose deux problèmes en RA : (1) la mise à jour en temps réel du monde réel est perdue. Cela est problématique dans des environnements industriels dynamiques, car les POIs sont fortement connectés à ce qui peut être observé sur la machine. (2) Les utilisateurs peuvent être désorientés lorsqu'ils reprennent la vue en direct de la caméra, car le point de vue a changé. Shift & Freeze [108] surmonte ce problème en combinant Shift [109] (voir sous-section 6.4.1) et les techniques d'arrêt sur image [70]. Après une pression sur l'écran, le flux visuel de la caméra est figé et une légende apparaît au-dessus du doigt (voir Figure 6.13c-d). La pause de la vidéo permet aux utilisateurs de pointer avec précision la cible souhaitée. L'affichage temps réel reprend une fois que le doigt n'est plus en contact avec l'écran. Ainsi, figer ou non l'image est un quasi-mode qui est déclenché par le système au moment où l'utilisateur commence l'interaction en touchant l'écran et se termine quand l'utilisateur relève le doigt.

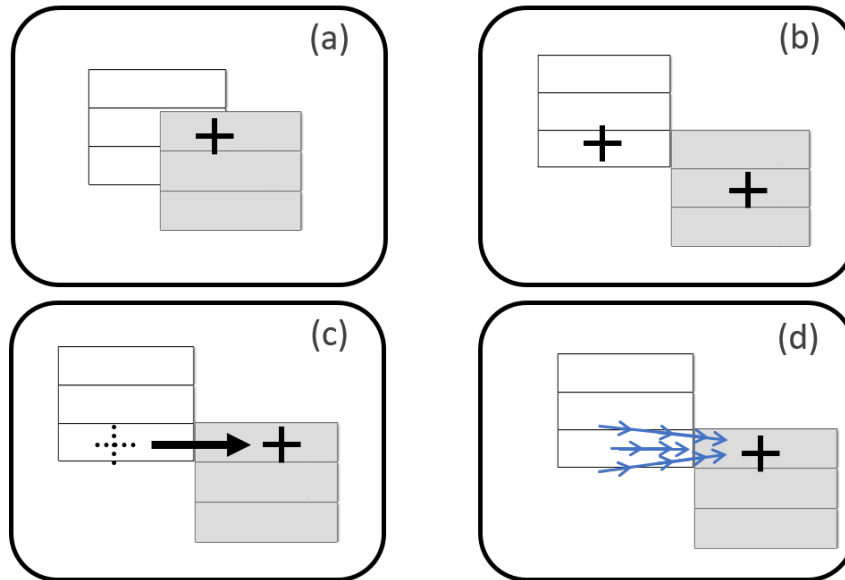


FIGURE 6.14 – Mise en contexte de l'interaction : (a) Saut de menu sur le curseur (b) Curseurs multiples (c) Saut de curseur (d) «Champ de force» : aide au déplacement du curseur.

6.6 Mise en contexte de l'interaction

Les techniques d'assistance au pointage ci-dessus focalisent sur l'optimisation de l'acquisition de cibles. L'interaction après la sélection d'une cible n'est pas considérée. Au contraire, les travaux sur les menus hiérarchiques se concentrent sur l'interaction après avoir pointé sur un élément du premier niveau du menu. Ceci est pertinent pour notre étude qui se concentre non seulement sur l'acquisition d'un POI, mais également sur la sélection d'un élément du menu contenu dans ce POI. Pour faciliter la sélection d'un élément d'un sous-menu après avoir pointé sur un élément d'un menu parent, nous identifions trois stratégies schématisées à la Figure 6.14.

Une première approche consiste à faire sauter le menu vers le curseur (Figure 6.14a) pour raccourcir la distance de pointage. Kobayashi et Igarashi considèrent les menus qui se chevauchent [67] : le sous-menu saute à la position actuelle du curseur, chevauchant ainsi les menus parents. Toutefois, l'intrusion visuelle provoquée par ce chevauchement est gênante car elle occulte une partie du menu parent et induit une légère perte de contexte. Les Bubbling menus [106] améliorent la sélection des éléments du menu en les étendant grâce à un Bubble Cursor [44]. Pour faciliter la sélection d'un élément de second niveau, le sous-menu correspondant se déplace verticalement, en suivant le mouvement du curseur dans le menu parent. Cependant, le déplacement d'un menu ((Figure 6.14a) en RA mobile n'est pas approprié, car il brise la relation entre le monde numérique et le monde physique : les menus affichent uniquement des informations pertinentes par rapport à leur emplacement dans le monde physique. Dans les environnements industriels, un menu affichant la température d'un moteur à l'intérieur d'une machine n'a de sens qu'à l'endroit où cette valeur est mesurée.

Une autre stratégie qui ne modifie pas la disposition visuelle des menus repose sur l'utilisation de plusieurs curseurs (voir sous-section 6.4.2 et Figure 6.14b). Par exemple, le curseur satellite [120] associe chaque cible à un curseur à proximité pour optimiser la distance de pointage. Cependant, cette solution s'applique principalement aux grands écrans. En RA mobile, la taille de l'écran est limitée. Comme expliqué dans la section sous-section 6.4.2, même si l'attention des utilisateurs est concentrée sur la cible souhaitée (c'est-à-dire un élément d'un menu), il sera visuellement distrait par les autres curseurs car l'écran est de petite taille.

Au-delà du déplacement du sous-menu et de la multiplication des curseurs, une dernière approche consiste à aider le déplacement du curseur vers le menu (Figure 6.14c). Avec les curseurs sautants [3], la sélection d'un élément d'un menu parent ouvre un sous-menu et fait sauter le curseur directement au centre du premier élément de ce sous-menu. Plutôt que de contrôler complètement le curseur, Ahlström [2] propose d'appliquer une force virtuelle qui pousse automatiquement le curseur vers le premier élément du sous-menu. Ahlström et al. [3] ont effectué une comparaison de l'approche par application d'une force (Figure 6.14d) et du saut de curseur (Figure 6.14c, appelé *Jumping Menu System* dans [3]) pour le cas des menus déroulants en cascade : les performances étaient similaires mais les auteurs soulignent les avantages de l'approche d'un curseur qui saute sur celle de l'application d'une force. En particulier "l'utilisateur n'est pas obligé de modifier sa manière habituelle d'interagir et peut choisir de profiter des sauts de curseur que dans des situations particulièrement lourdes...". C'est pourquoi nous avons choisi d'étudier expérimentalement l'approche par curseur sautant dans le chapitre suivant.

6.7 Synthèse et conclusion

Dans ce chapitre, nous avons passé en revue les techniques d'assistance au pointage. Notre sujet d'étude est la sélection de POIs en RA mobile. Pour cela, nous avons énoncé trois critères (voir section 6.2), tels qu'une technique d'assistance au pointage :

1. N'introduit pas de surcharge visuelle à l'écran, pour ne pas perturber la zone de travail de l'utilisateur.
2. Ne brise pas le lien qui unit la cible à son environnement. Dans le cas de la réalité augmentée, c'est le lien entre le monde numérique et le monde physique qui doit être conservé.
3. Considère la tâche de pointage dans son ensemble (sélection d'une cible et d'un élément numérique au sein de cette cible).

Les tableaux 6.1 et 6.2 résument notre revue des techniques existantes selon ces trois critères.

Pointer en RA mobile peut être effectué soit de manière directe (stylet ou doigts nus [89, 96], soit à l'aide d'un curseur centré sur l'écran [93, 96]. Dans les deux cas, des techniques

-	-	Technique	Intrusion limitée	Lien cible-environnement conservé
2D	Pointage direct	Tap Tap [96]	PARTIELLE	OUI
		MagStick [96]	OUI	OUI
		Take-Off [90]	OUI	OUI
		Shift [109]	OUI	OUI
		Thumbspace [109]	NON	OUI
		ZoomTap [109]	NON	OUI
		Burst [109]	NON	OUI
		Escape [119]	PARTIELLE	OUI
		BezelCursor [71]	NON	OUI
		2DDragger [102]	OUI	OUI
	Pointage indirect	Drag-and-Pop [12]	NON	OUI
		Drag-and-Pick [12]	NON	OUI
		Object Pointing [53]	NON	OUI
		Rake Cursor [21]	NON	OUI
		Ninja Cursor [68]	NON	OUI
		Bubble Cursor [44]	NON	OUI
		Bubble Lens [79]	NON	OUI
		Fan Cursor [103]	OUI	OUI
		Satellite Cursor [120]	NON	OUI
Voronoi [55]		OUI	OUI	
Starburst [15]	OUI	OUI		
RA	Pointage direct	Touch Projector [23]	OUI	OUI
		Dual-Finger Midpoint [104]	OUI	OUI
		Dual-Finger Offset [104]	OUI	OUI
		Expand [30]	OUI	NON
		DrillSample [78]	NON	NON
		Shift and Freeze [108]	OUI	PARTIELLE
	Pointage indirect	Relative Pointing [108]	OUI	OUI

TABLE 6.1 – Évaluation des techniques d’assistance au pointage, listées dans leur ordre de présentation dans ce chapitre.

Technique	Intrusion limitée	Lien cible-environnement conservé
Menus qui se chevauchent [67]	NON	NON
Bubbling Menus [106]	NON	NON
Curseur sautant [3]	OUI	OUI
Menus avec champs de force [2]	OUI	OUI

TABLE 6.2 – Évaluation des techniques d’assistance au pointage dans des menus contextuels

[103, 55, 15, 96, 90, 109, 102, 108, 104] répondent favorablement à nos deux premiers critères.

Dans le cas d’un pointage direct, de nombreuses solutions existent : une approche pour contourner le problème du ”fat finger” est de déplacer le curseur ou la vue cachée par le doigt [17]. Pour résoudre l’instabilité de la scène (due au tremblement naturel de la main), il est possible de figer l’image [70], mais la relation ”temps réel” entre le monde numérique et le monde physique est brisée. La technique Shift & Freeze [108] fige le flux vidéo de la caméra juste le temps du pointage et répond à nos besoins. Dans le cas d’une forte densité de POIs, la technique Escape [119] est une option pour sélectionner avec précision une cible.

Néanmoins, dans nos travaux, nous privilégions les techniques de pointage avec un curseur pour les raisons suivantes, certaines issues du domaine applicatif cible. Les tablettes tactiles sont davantage employées en RA industrielle car leur taille d’écran apporte un confort visuel utile pour l’affichage de plusieurs informations techniques en simultané. Il devient inconfortable de pointer avec le doigt, notamment lorsque la taille et le poids de l’appareil mobile rendent difficile de le tenir à une seule main. De plus, à la différence du pointage direct, le curseur centré sur l’écran n’est pas perturbé par des problèmes d’occultation du doigt, ou de stabilité de la scène. Augmenter le curseur comme Bubble Cursor [44] ou Fan Cursor [103] n’est ici pas une approche envisagée, de part la gêne visuelle engendrée sur un écran de taille réduite. En revanche, les techniques d’expansion de cibles [55] constitueraient une approche efficace pour réduire la distance nécessaire à l’acquisition d’un objet en RA mobile. Nous nous concentrons dans la suite des travaux sur une technique de pointage avec curseur couplée à des cibles étendues.

Pour notre étude, optimiser la sélection d’un POI n’est pas l’unique objectif : en effet, l’utilisateur doit également naviguer au sein des éléments numériques contenus dans cette cible. Il n’est donc pas uniquement question de faciliter le pointage d’une cible, mais également des informations qu’elle contient. La littérature sur les menus en cascades apporte des éléments de réponses sur la manière d’optimiser le pointage séquentiel de plusieurs cibles. Les deux critères énoncés précédemment s’appliquent également aux menus en cascade. Nous résumons dans le Tableau 6.2 les techniques existantes selon ces

critères : seules les techniques [3, 2] qui déplacent le curseur et non les menus répondent aux deux critères fixés pour notre étude en RA mobile.

La littérature n'a révélé aucune technique d'assistance au pointage avec curseur (1) adaptée à la RA, (2) garantissant un minimum d'intrusion visuel à l'écran, (3) ne brisant pas le lien qui unit la cible à son environnement physique, et (4) considérant la tâche de pointage dans son ensemble. Cependant, la technique du relative pointing [108], d'expansion de cibles [55] et du curseur sautant [3] définissent des techniques prometteuses à appliquer au cas de la RA mobile. Basée sur un curseur centré à l'écran, la technique du relative pointing [108] n'implique aucun problème d'occultation de cibles ou d'instabilité. La technique d'expansion de cibles permet d'améliorer la sélection des POIs. Enfin, faire sauter le curseur permet d'accéder directement à un menu ouvert sans avoir à déplacer davantage le curseur. Dans le chapitre suivant, nous nous basons donc sur ces trois techniques pour concevoir une technique d'assistance au pointage de POIs adaptée en RA et respectant les trois critères énoncés en section 6.2.

7

Conception de techniques de pointage pour la réalité augmentée mobile

Sommaire

7.1	Introduction	120
7.2	Expansion de cibles en Réalité Augmentée	122
7.2.1	Adaptation à la Réalité Augmentée	123
7.2.2	Retour visuel	124
7.3	Sélection d'un élément du menu d'un POI	127
7.4	Conclusion	132

7.1 Introduction

Ce chapitre est consacré à la tâche composée de deux parties : (1) sélectionner un point d'intérêt (POI) en Réalité Augmentée (RA) mobile, et (2) sélectionner un élément dans le menu associé au POI. Nous illustrons cette tâche composée dans le domaine de la maintenance industrielle (voir *scénario 3* complet, section 2.3).

Un opérateur de maintenance souhaite obtenir des informations techniques sur le four à souder d'une chaîne d'assemblage. Il se positionne devant la machine (Figure 7.1a), et sélectionne le POI rassemblant les informations souhaitées (Figure 7.1b-c). Un menu s'ouvre (Figure 7.1c) contenant des informations sur la température, la pression etc. à cet endroit de la ligne d'assemblage. Ici, la sélection du POI est une première étape, et la sélection de l'information de température du four est la deuxième étape de la tâche (Figure 7.1d-e). Ce scénario illustre l'intérêt de considérer la tâche de pointage RA dans sa globalité.

Comme énoncé en synthèse du chapitre précédent, nous considérons le cas du pointage indirect avec un curseur affiché à l'écran : en RA mobile, les techniques de pointage sont altérées par l'instabilité de la scène augmentée avec laquelle les utilisateurs interagissent à l'écran. Alors que le pointage direct renforce cette instabilité en impliquant la tenue de l'appareil d'une main tout en interagissant sur l'écran de l'autre [107], nous optons pour

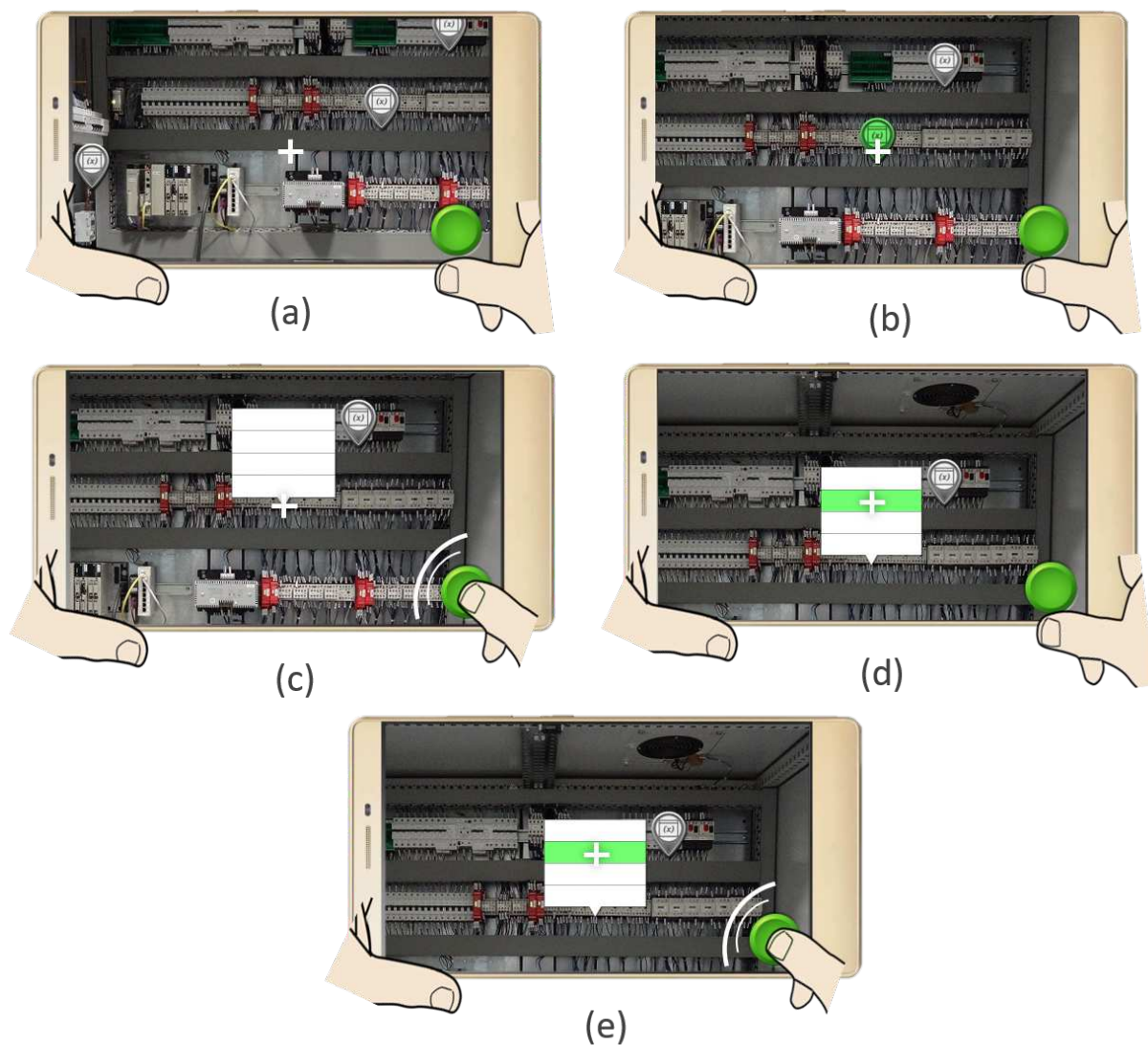


FIGURE 7.1 – Ouverture d'un POI et sélection d'un élément dans le menu correspondant dans le cas du pointage indirect (a) Situation initiale, l'utilisateur se trouve devant une machine augmentée de plusieurs POIs. (b) Survol d'un POI en alignant le POI avec la croix centrale (c) Sélection du POI en appuyant sur un bouton de validation, le menu correspondant s'ouvre. (d) Survol d'un élément du menu en alignant l'élément avec la croix centrale. (e) Sélection de l'élément du menu en appuyant sur un bouton de validation.

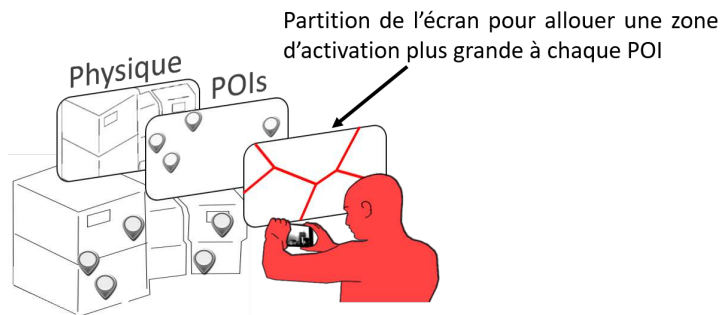


FIGURE 7.2 – Techniques d'expansion de cibles en RA mobile : sélection de POIs.

la sélection de cibles avec un curseur. De ce fait, nous réduisons cette instabilité tout en évitant les problèmes d'occultation de la cible par le doigt. La figure Figure 7.1 résume la sélection d'un POI et la sélection d'un élément dans le menu correspondant dans le cas du pointage indirect. Pour faciliter les tâches de pointage en RA mobile, nous adoptons une approche d'expansion de cible qui facilite la sélection d'un POI en lui allouant une zone de sélection plus grande.

Nous décrivons la conception de techniques d'interaction facilitant la tâche de pointage composée. Pour cela nous organisons le chapitre selon les deux étapes de la tâche. Nous détaillons tout d'abord l'adaptation des techniques d'expansion de cibles à la RA, pour optimiser la sélection d'un POI. Puis nous introduisons deux nouvelles techniques de pointage basées sur le saut de curseur, afin d'optimiser la sélection des éléments numériques contenus au sein de ce POI.

7.2 Expansion de cibles en Réalité Augmentée

Une approche par expansion de cibles n'a pas encore été appliquée pour le pointage en RA mobile et est particulièrement prometteuse lorsque l'on se concentre sur l'interaction avec les POIs en RA mobile. Nous schématisons le contenu affiché à l'écran comme contenant deux calques : l'un est le flux vidéo représentant le monde physique et l'autre les POIs augmentant le monde physique. Ce contexte (voir Figure 7.2) lève les limitations identifiées des techniques d'expansion de cibles. En effet :

- Plusieurs applications de RA offrent deux modes distincts : l'édition (pour relier les informations numériques aux objets physiques) et l'exploration (pour localiser les POIs) [22, 83]. Alors que le mode d'édition nécessite un positionnement in situ des POIs et donc la nécessité de pointer n'importe quelle position à l'écran, le mode d'exploration implique la sélection de ces POIs. Ainsi, la connaissance préalable des cibles nous permet d'explorer les techniques d'expansion de cibles pour faciliter la sélection des POIs.
- Pour améliorer les tâches de pointage, les techniques d'expansion de cibles allouent des zones de sélection plus grandes aux POIs. Ces techniques nécessitent un espace moteur vide entre les POIs, ce qui est souvent le cas en RA car un POI est lié à un objet ou un emplacement physique.

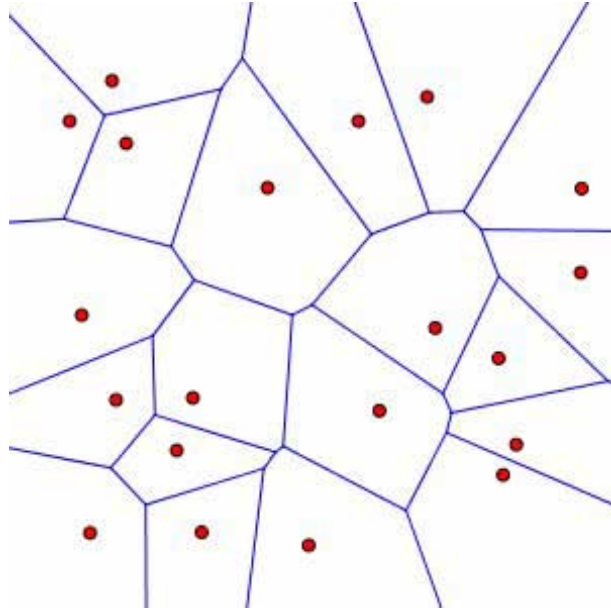


FIGURE 7.3 – Diagramme de Voronoï : partitionnement de l'espace en cellules tel que (1) chaque cellule ne contient qu'une seule cible, et (2) tous les pixels d'une cellule sont plus proches du point correspondant que des autres points. Image issue de [100].

Pour décrire la conception des techniques d'expansion de cibles en RA, nous présentons les deux éléments de base [55] : l'algorithme d'expansion et l'aide visuelle concernant les POIs étendus.

7.2.1 Adaptation à la Réalité Augmentée

Comme expliqué dans le chapitre précédent les techniques d'expansion de cibles permettent d'optimiser le geste de pointage. En étendant la zone active des cibles, l'utilisateur n'est plus contraint d'aligner le curseur avec un POI pour le sélectionner. Le diagramme de Voronoï est un exemple de partitionnement de l'espace en polygones convexes, aussi appelés cellules de Voronoï (Figure 7.3). Chaque cellule contient qu'une seule cible, telle que tous les pixels d'une cellule sont plus proches de la cible correspondante que de toute autre cible. Les cellules de Voronoï résultantes définissent la taille étendue des POIs. Pour calculer un diagramme de Voronoï basé sur les POIs en RA mobile, deux aspects sont à traiter :

1. Adaptation à une scène 3D : l'ensemble des POIs est placé dans un environnement 3D autour de l'utilisateur et leur affichage en 2D sur l'écran.
2. Adaptation à une scène dynamique : le partitionnement de Voronoï s'applique à une configuration fixe de points. En RA, l'utilisateur évolue dans un environnement en temps réel. Le point de vue de la caméra est amené à être modifié (déplacement, ré-orientation etc.). Ceci implique donc de considérer dynamiquement les POIs qui apparaissent et disparaissent de l'écran pour calculer le partitionnement de Voronoï.

Une façon immédiate de calculer le pavage à l'écran c'est-à-dire la partition de Voronoï est de considérer la projection des POIs 3D sur le plan de la tablette. Chaque projection d'un POI sur le plan de la tablette est un germe pour calculer le pavage à l'écran. Cependant, en projetant l'ensemble des POIs, certaines cellules de Voronoï peuvent être vides. Elle font en effet référence à des POIs qui ne sont pas dans le champ de vision de la caméra, et ne sont pas visibles à l'écran. Nous appliquons donc le partitionnement de Voronoï uniquement aux POIs visibles à l'écran. Ainsi, à tout instant, aucune des cellules du pavage n'est vide. Cela implique donc un calcul dynamique du partitionnement de Voronoï.

7.2.2 Retour visuel

Une aide visuelle est essentielle pour tirer pleinement parti des zones étendues pendant la tâche de pointage [55]. En RA mobile, la minimisation de l'intrusion visuelle est cruciale pour ne pas surcharger l'écran déjà occupé par le flux vidéo de la caméra (voir aussi la première partie de ce manuscrit). L'environnement physique présent devant l'utilisateur est affiché à l'écran. Par dessus cet environnement physique sont affichés les éléments numériques augmentant l'environnement physique (voir Figure 7.2). Il est important de noter que l'arrière-plan, l'environnement physique, a une signification pour les utilisateurs. Par exemple les opérateurs de maintenance ont besoin de voir la machine physique pour comprendre les informations fournies par les POIs (voir chapitre 2).

Pour concevoir l'aide visuelle ou rétroaction visuelle de la technique d'expansion, nous nous appuyons sur l'espace de conception de Guillon et al. [55] tout en visant à minimiser l'intrusion visuelle. Comme exposé dans le chapitre précédent, Guillon et al. [55] définissent un espace de conception à 3 axes pour caractériser les rétroactions visuelles des techniques d'expansion de cibles et ont évalué l'impact de plusieurs rétroactions visuelles sur les performances de pointage. Dans un espace 2D, ils ont constaté qu'une mise en évidence statique de la cible la plus proche est efficace, un résultat récemment confirmé pour l'interaction 3D par Baloup et al. [10]. Sur la base de ces résultats, nous disposons de deux choix de conception :

- CELL PAINTING (Figure 7.4a) : technique qui affiche uniquement la cellule de Voronoï dans laquelle le curseur est actuellement situé. Cette technique permet à l'utilisateur de visualiser la zone de sélection d'une cible.
- TARGET (Figure 7.4b) : technique qui met en évidence la cible de la cellule de Voronoï dans laquelle se trouve le curseur. La mise en évidence peut consister par exemple en un grossissement de la cible ou un changement de couleur de la cible.

Nous avons implémenté les deux types de rétroaction visuelle CELL PAINTING et TARGET, comme illustré à la Figure 7.4. Nous avons ensuite conduit une étude pilote auprès de trois professionnels. L'objectif de l'étude était de déterminer quelle rétroaction visuelle est préférée, dans le cas où l'arrière-plan (l'environnement physique) a une signification pour la tâche de pointage. Ceci explique pourquoi nous avons conduit l'étude pilote avec des professionnels. Les trois participants ont été invités à sélectionner huit POIs associés

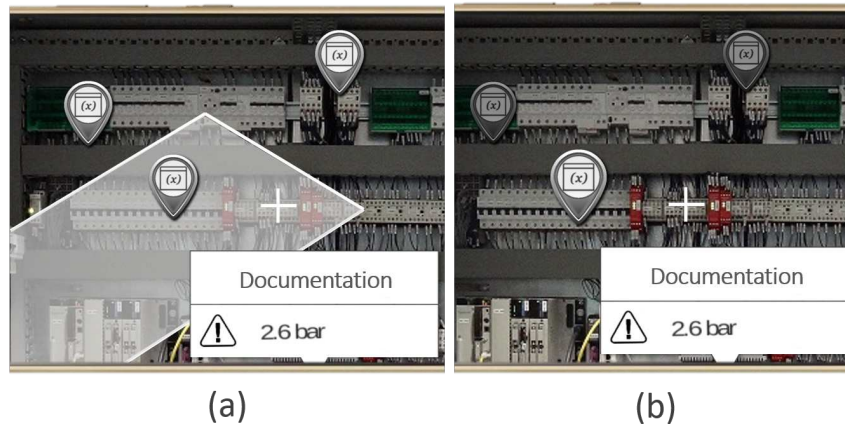


FIGURE 7.4 – Deux types d’aide visuelle de la technique d’expansion de cibles a) CELL PAINTING : mise en évidence de la cellule de Voronoï dans laquelle se trouve le curseur (b) TARGET : mise en évidence de la cible de la cellule de Voronoï dans laquelle se trouve le curseur.

à une machine, sans ordre imposé. Nous avons utilisé un modèle de ligne de production d’une taille de 3m x 0,75m, identique à celle de l’étude expérimentale 2 de la section 5.3 (voir Figure 5.9b). Avant de commencer, le mécanisme d’expansion de cibles a été expliqué aux participants. Les trois professionnels ont préféré la technique TARGET en raison de son intrusion visuelle minimale à l’écran. De plus, ils ont indiqué que TARGET leur permettait de mieux percevoir la machine physique, qui est le cœur d’une tâche de maintenance.

Il est important de mentionner qu’avec TARGET, les cellules de Voronoï ne sont pas explicitement affichées sur l’écran. A tout moment, seul le POI situé dans la cellule de Voronoï du curseur est mis en évidence. Il s’agit en effet d’une rétroaction visuelle de type *implicite*, tel que défini dans l’espace de conception de Guillon et al. [56]). Ainsi, il n’est pas nécessaire de calculer le partitionnement complet de Voronoï. Comme expliqué également par Baudisch et al. dans [15], un algorithme simple consiste à calculer le POI le plus proche du curseur. L’automate présenté sur la Figure 7.5 résume les états des POIs dans le cas d’expansion de POIs pour du pointage indirect avec un curseur fixe sur l’écran.

Afficher en permanence la cellule de Voronoï dans lequel le curseur se trouve provoque une gêne visuelle reportée lors de notre étude pilote. Par contre, dans le cas du pointage direct, une technique qui afficherait de façon temporaire la cellule de Voronoï comme CELL PAINTING pourrait aider au pointage direct. Pour cela, la détection de la proximité des doigts au-dessus de la surface d’affichage (voir Figure 7.6b) permettrait au système d’anticiper la zone de l’écran qui intéresse l’utilisateur [118] (événement survol "hover"). Afficher la cellule de Voronoï avant le moment du toucher permet ainsi la sélection d’une cible avec le doigt en cliquant n’importe où dans la cellule, sans avoir à toucher directement le POI, réduisant ainsi les problèmes d’occultation du POI par le doigt. La distance de pointage serait raccourcie, et le temps de complétion d’un geste de pointage serait diminué.

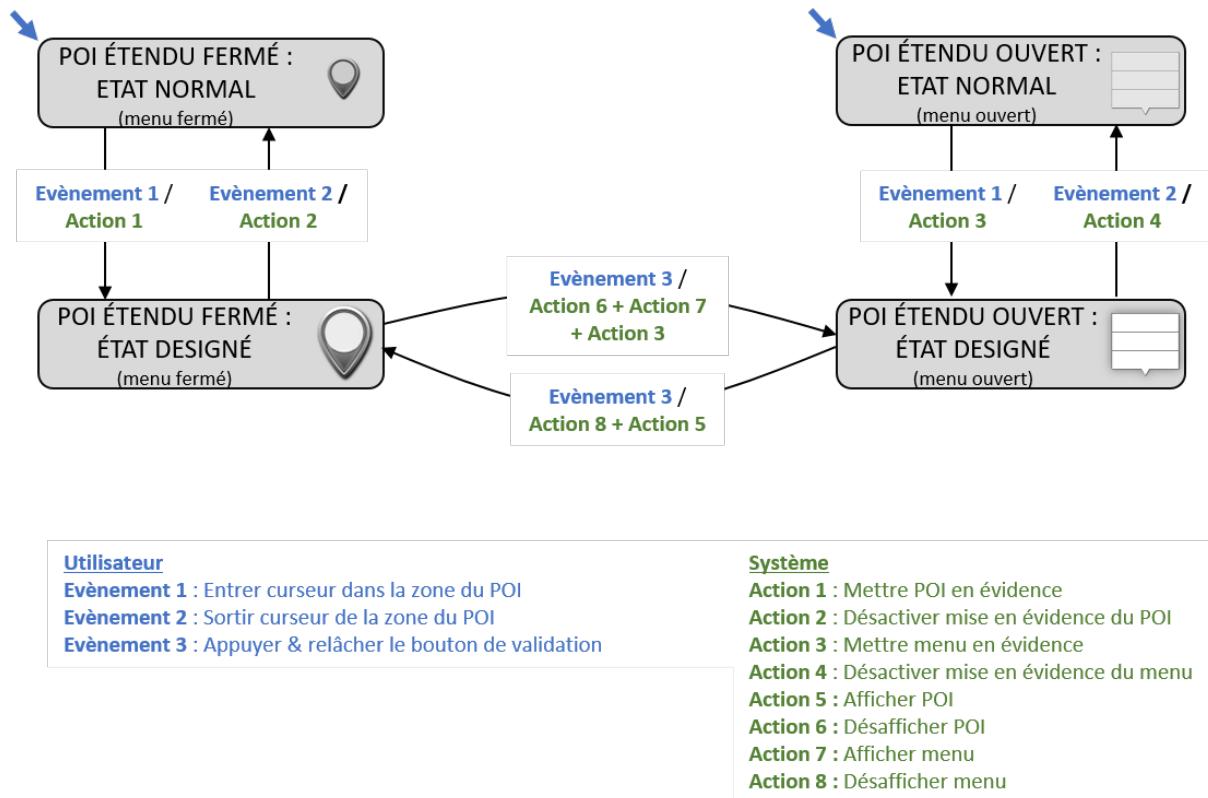


FIGURE 7.5 – États d'un POI : diagramme de transition d'états pour le cas du pointage indirect avec un curseur centré. Les transitions sont annotées par les événements utilisateur qui les déclenchent et les actions système résultantes (ou sorties graphiques produites).

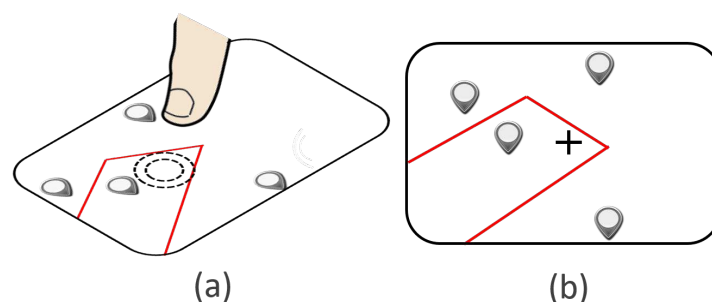


FIGURE 7.6 – Pointage à l'intérieur de la cellule Voronoï d'un POI. (a) Technique de pointage tactile : détection de la proximité du doigt au-dessus de la surface d'affichage. (b) Technique de curseur centré sur l'écran. Les cellules de Voronoï sont ici affichées à des fins d'explication.

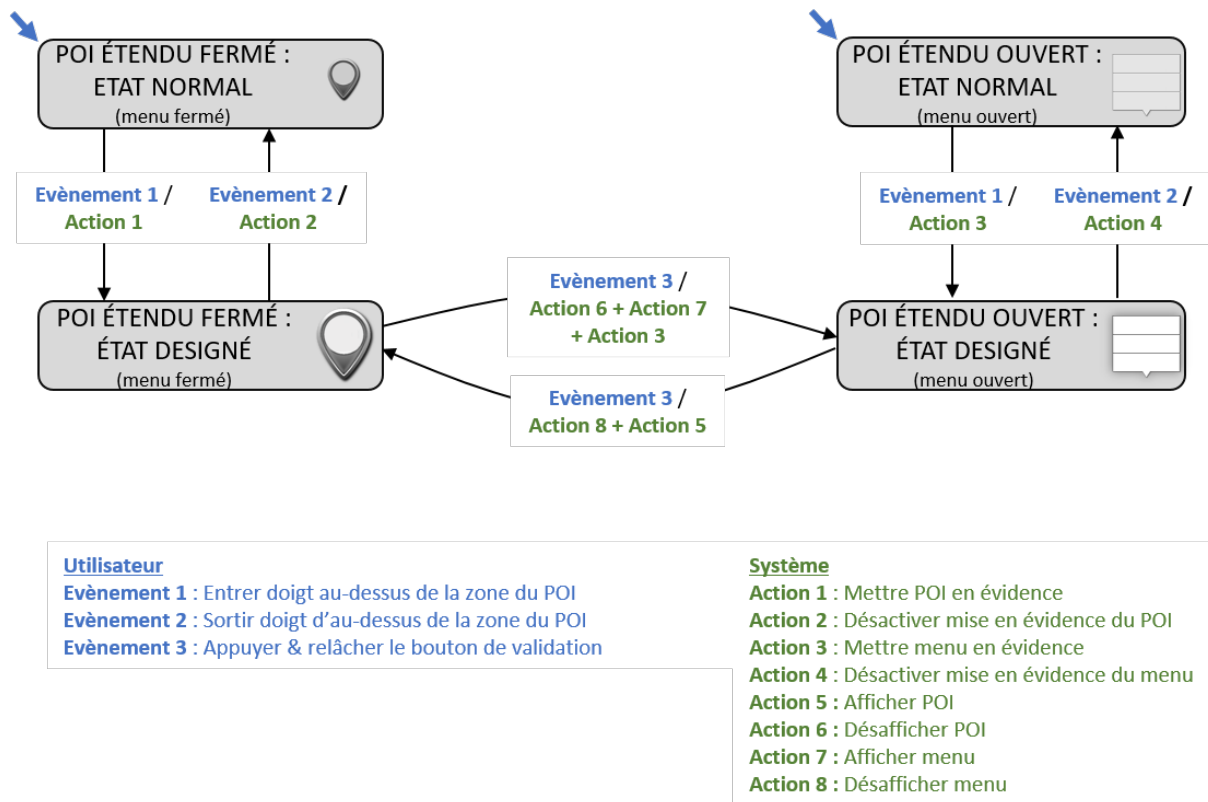


FIGURE 7.7 – États d'un POI : diagramme de transition d'états pour le cas du pointage direct avec détection de proximité. Les transitions sont annotées par les événements utilisateur qui les déclenchent et les actions système résultantes (ou sorties graphiques produites).

Cette approche a été appliquée pour l'expansion d'une cible unique par Yang et al. [118] : en pointage directe tactile, la technique détecte la proximité du doigt au-dessus de la surface d'affichage et élargit progressivement la cible, un widget, à mesure que le doigt s'approche de la surface. Il s'agit d'une variante de widgets en expansion [74] appliquée à l'interaction directe tactile en reposant sur la détection de la proximité du doigt de la surface pour obtenir les événements de survol ("hover").

En synthèse, les deux automates des figures 7.5 et 7.7 présentent les différents états des POIs et actions utilisateur dans le cas d'expansion de POIs pour du pointage indirect avec un curseur fixe sur l'écran et du pointage directe tactile.

Afficher la cellule de Voronoi (CELL PAINTING) constitue une piste de recherche à approfondir pour le cas du pointage direct. Pour notre étude d'une technique de pointage indirect avec un curseur affiché à l'écran, la solution de conception adoptée est de mettre en évidence le POI (TARGET) de la cellule de Voronoi dans laquelle se trouve le curseur.

7.3 Sélection d'un élément du menu d'un POI

Un POI permet d'accéder à des informations techniques liées à un emplacement physique spécifique sur la machine (voir *scénario 3*, section 2.3). Pour accéder à ces informations

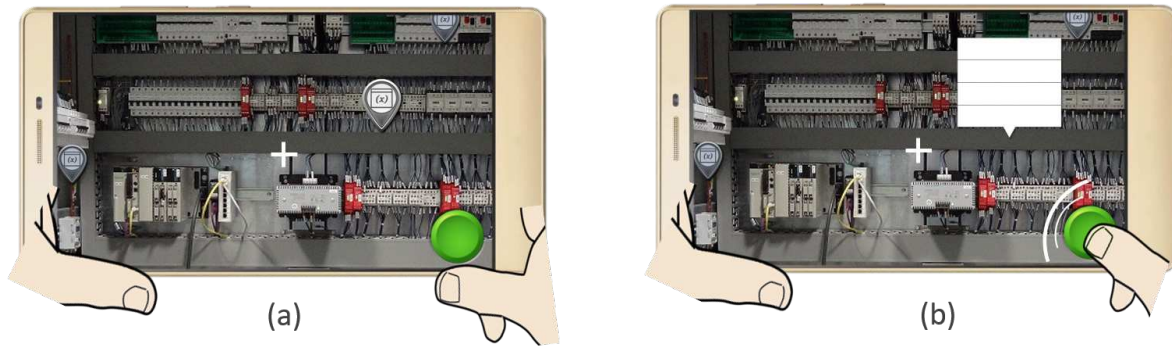


FIGURE 7.8 – Ouverture d'un POI grâce à une technique d'expansion de cibles (a) Situation initiale, un POI se trouve proche du curseur (dans la cellule de Voronoï du POI). Celui-ci est mis en évidence. (b) Sélection du POI en appuyant sur le bouton, le menu correspondant s'ouvre.

(pannes éventuelles, température d'un four de soudage etc.), les opérateurs de maintenance sélectionnent le POI en alignant le curseur avec la cible. La liste des informations est alors affichée sous forme de menu, placé au-dessus du POI (voir Figure 7.1).

Grâce aux techniques d'expansion de cibles, la sélection d'un POI peut être réalisée en amenant le curseur central à proximité du POI (voir Figure 7.8a). Il est alors possible de sélectionner/ouvrir un POI sans avoir à aligner le curseur sur le POI (voir Figure 7.8b). Un POI sélectionné ouvre un menu rassemblant toutes les informations utiles par rapport à ce POI (voir Figure 2.3d-e et Figure 7.1c). Le menu est affiché à la position du POI car il est étroitement lié à un objet physique ou à l'emplacement de ce POI. Cependant, après avoir sélectionné le POI avec la technique d'expansion de cibles, le curseur n'est plus directement positionné sur le menu (voir Figure 7.8b).

Une première technique, que nous notons Voronoï (V) est de laisser l'utilisateur amener le curseur sur le menu par un mouvement physique. Cette technique optimise uniquement le pointage du POI et est présentée à la Figure 7.9. Pour optimiser en plus de la sélection d'un POI, celle d'un élément de son menu, nous nous reposons sur la taxonomie que nous avons établie au chapitre précédent (voir section 6.6). Trois stratégies sont alors possibles pour aider à la sélection d'un élément d'un menu, sans que l'utilisateur ramène le curseur sur le menu : faire sauter le menu vers le curseur (voir Figure 7.10a), utiliser plusieurs curseurs (voir Figure 7.10b), ou aider le déplacement du curseur vers le menu (voir Figure 7.10c-d). Comme expliqué au chapitre précédent (voir section 6.6), nous conservons la solution de saut de curseur (voir Figure 7.10c), car elle ne brise pas le lien qui unit la cible numérique avec l'environnement physique, n'augmente pas l'intrusion visuelle à l'écran, et est préférée au guidage du curseur par application d'un champ de force (voir Figure 7.10d).

Faire sauter le curseur directement sur un menu ouvert permet de diminuer la distance de pointage. En effet, le curseur étant déjà positionné sur le menu, l'utilisateur n'a plus

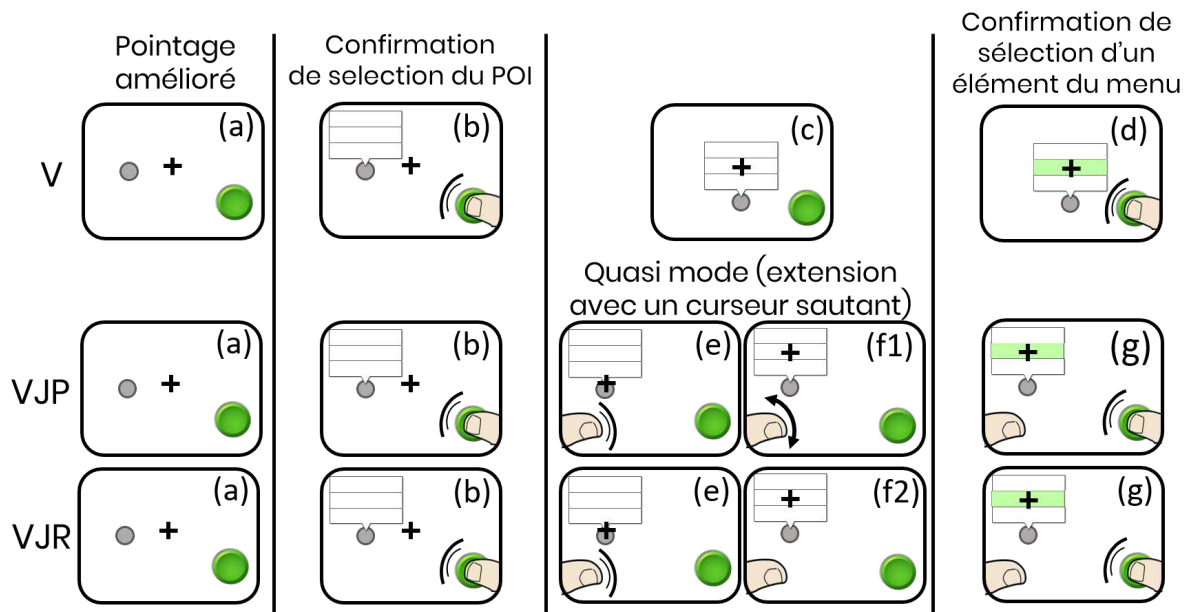


FIGURE 7.9 – Démonstration des techniques. (a) Pointage physique (b) Sélection d'un POI en appuyant sur le bouton avec le pouce droit, le menu correspondant s'ouvre (c) Pointage physique pour déplacer le curseur fixe et centré sur l'écran vers l'élément du menu souhaité (d) Sélection d'un élément du menu en appuyant sur le bouton avec le pouce droit. (e) Saut du curseur sur le menu en appuyant sur l'écran avec le pouce gauche. Pouce gauche maintenu à l'écran (f1) Pointage physique pour déplacer le curseur sur l'élément de menu (f2) Pointage relatif pour déplacer le curseur sur le menu en faisant glisser le pouce gauche sur l'écran (g) Sélection d'un élément du menu en appuyant sur le bouton avec le pouce droit tout en maintenant le pouce gauche sur l'écran.

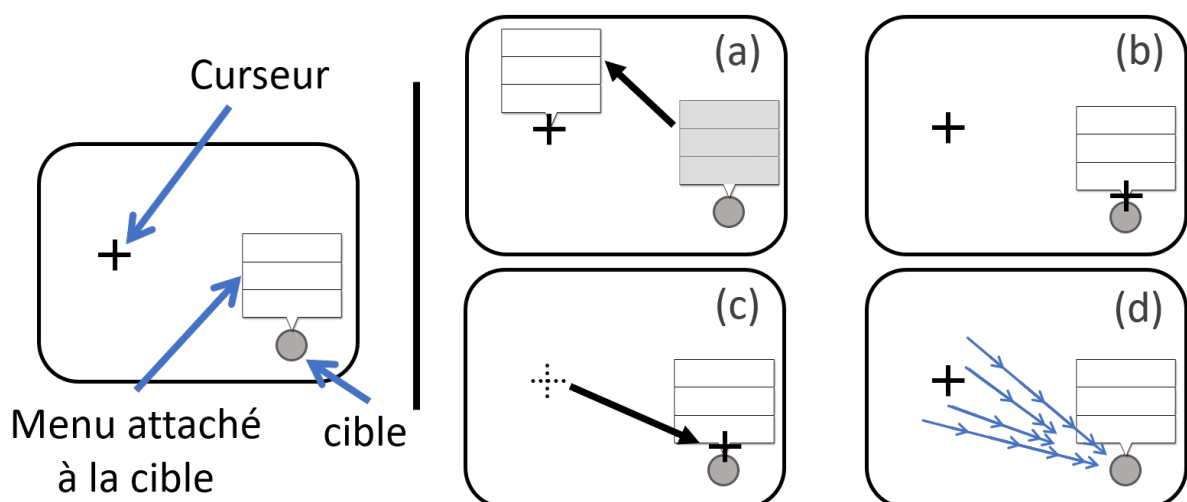


FIGURE 7.10 – Stratégies pour la sélection d'un élément dans un menu : (a) Saut de menu vers le curseur (b) Curseurs multiples (c) Saut de curseur (d) «Champ de force» : aide au déplacement du curseur.

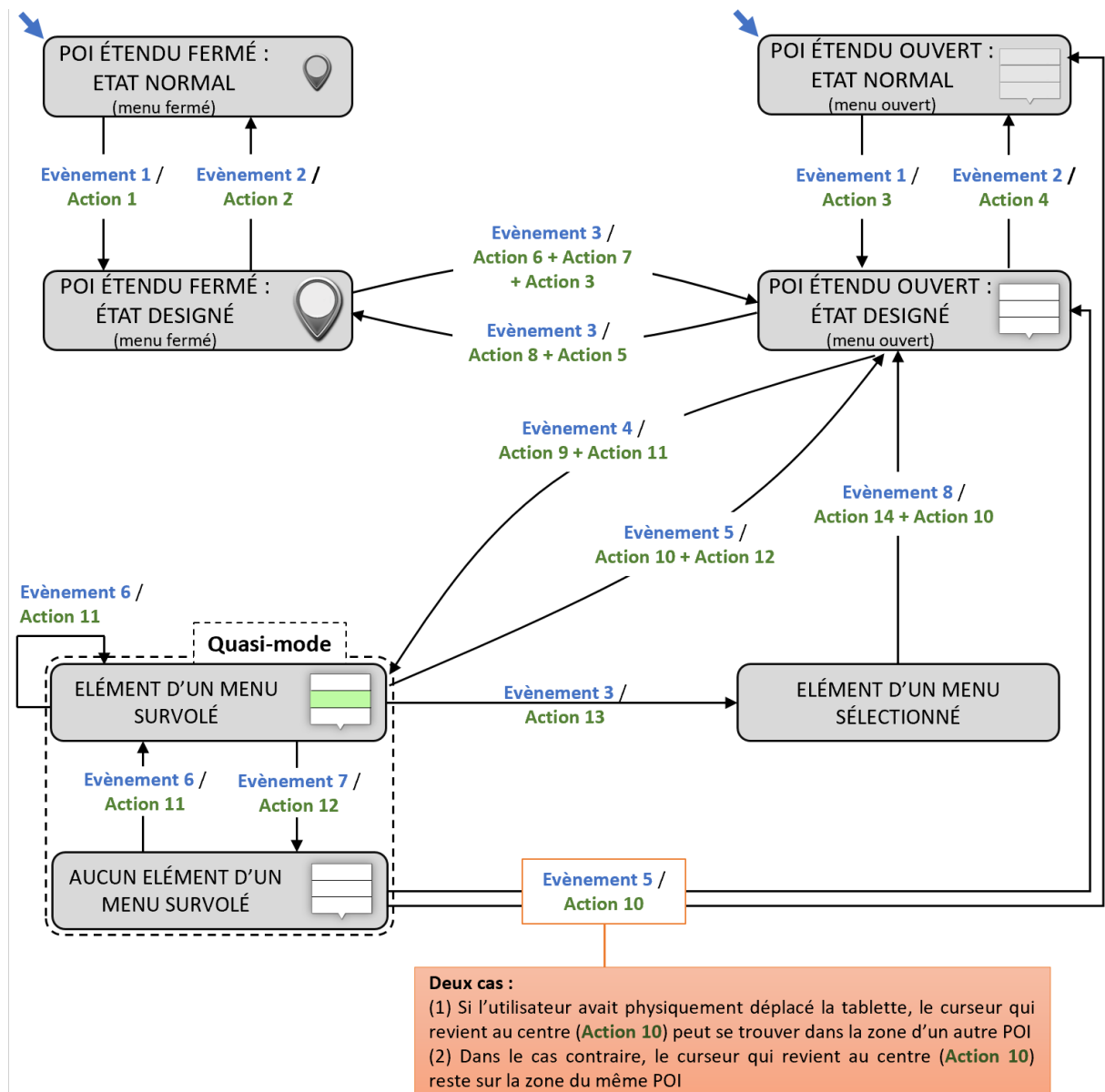
qu'à l'orienter vers l'élément de son choix. Une première solution de conception serait d'observer les mouvements de l'appareil mobile pour détecter s'il se dirige vers un menu. Si tel est le cas, le curseur sauterait alors automatiquement en milieu de mouvement pour raccourcir la distance de pointage. En pratique, cette solution ne fonctionne pas. En effet, nous faisons l'hypothèse qu'un déplacement de l'appareil mobile vers un menu signifie que l'utilisateur est intéressé par ce menu. Or, elle/il peut parfaitement être intéressé(e) par un POI se trouvant près de ce menu. Le curseur sauterait alors sur le menu et perturberait la tâche de l'utilisateur.

Une autre solution de conception que nous adoptons consiste à laisser le contrôle explicite à l'utilisateur. Une action explicite de sa part provoque le saut du curseur sur le menu ouvert du POI couramment sélectionné, si un tel menu existe (voir Figure 7.9). De plus, comme pour les menus déroulants en cascade étudiés par Ahlström et al. [3], le curseur a un comportement déterministe : sauter sur le menu le plus proche, s'il existe. Notre hypothèse est qu'une action délibérée pour déclencher le saut du curseur et le comportement déterministe du curseur limitent l'effet de désorientation. De cette façon, le saut du curseur ne surprendrait pas l'utilisateur car il est le déclencheur de cette action. Deux comportements restent donc à définir : la manipulation du curseur à l'intérieur d'un menu et la validation de l'élément numérique pointé par le curseur.

Nous avons choisi de séparer spatialement l'interaction permettant de manipuler le curseur et l'interaction permettant de valider une sélection (élément d'un menu ou POI). Pour cela, nous proposons une solution utilisant les deux pouces dans un contexte d'interaction bi-manuelle. En effet, la RA peut être impactée par les tremblements naturels des mains. Maintenir l'appareil mobile des deux mains permet de minimiser cet impact. De plus, l'interaction bi-manuelle est particulièrement adaptée à un contexte de pointage indirect avec un curseur à l'écran. Enfin, les tablettes sont couramment employées en RA industrielle pour leur taille d'écran et donc leur confort d'utilisation. Leur taille et leur poids implique de tenir ce type de dispositif mobile à deux mains pour plus de stabilité.

Lorsque l'utilisateur tient l'appareil mobile des deux mains, les pouces sont à l'avant et les doigts restants se trouvent à l'arrière de l'écran. Les deux pouces ont donc une zone fonctionnelle confortable pour l'interaction dans les coins inférieurs de l'écran [110]. Reposant sur l'espace de conception BiTouch pour l'interaction bimanuelle défini par Wagner et al. [110], nous avons conçu une interaction bi-manuelle comme suit : le pouce gauche est utilisé pour faire sauter le curseur et le manipuler, tandis que le pouce droit est dédié à la validation de la sélection d'un POI ou d'un élément dans un menu (voir Figure 7.9). Le bouton placé dans le coin inférieur droit de l'écran est donc nécessaire pour confirmer toute sélection (voir Figure 7.8 et Figure 7.9).

Lorsque le pouce gauche touche l'écran, deux cas sont possibles selon l'état du POI le plus proche : (1) si le POI n'est pas sélectionné (état fermé), aucun menu n'est affiché et le curseur reste au centre de l'écran. (2) Si le POI est déjà sélectionné (état ouvert),

**Utilisateur**

- Évènement 1 :** Entrer curseur dans la zone du POI
- Évènement 2 :** Sortir curseur de la zone du POI
- Évènement 3 :** Appuyer & relâcher le bouton de validation
- Évènement 4 :** Appuyer le pouce gauche sur l'écran
- Évènement 5 :** Relâcher le pouce gauche de l'écran
- Évènement 6 :** Déplacer le curseur dans le menu par mouvement physique ou déplacement (drag) de la main gauche
- Évènement 7 :** Déplacer le curseur hors du menu par mouvement physique ou déplacement (drag) de la main gauche
- Évènement 8 :** Action de fermer l'élément du menu sélectionné (action variable selon le contenu : sélection d'un bouton etc.)

Système

- Action 1 :** Mettre POI en évidence
- Action 2 :** Désactiver mise en évidence du POI
- Action 3 :** Mettre menu en évidence
- Action 4 :** Désactiver mise en évidence du menu
- Action 5 :** Afficher POI
- Action 6 :** Désafficher POI
- Action 7 :** Afficher menu
- Action 8 :** Désafficher menu
- Action 9 :** Faire sauter le curseur sur menu
- Action 10 :** Ramener le curseur au centre
- Action 11 :** Mettre élément du menu en évidence
- Action 12 :** Désactiver mise en évidence de l'élément du menu
- Action 13 :** Afficher contenu associé à l'élément du menu sélectionné
- Action 14 :** Désafficher contenu associé à l'élément du menu sélectionné

FIGURE 7.11 – États d'un POI et du menu associé : diagramme de transition d'états pour le cas de l'interaction bi-manuelle VJP et VJR. Les transitions sont annotées par les événements utilisateur qui les déclenchent et les actions système résultantes (ou sorties graphiques produites).

son menu est affiché, et le curseur saute automatiquement sur le premier élément de ce menu (Figure 7.9e). Le pouce droit est utilisé pour valider la sélection d'un élément dans un menu (Figure 7.9g). Ce même bouton est également utilisé pour sélectionner/ouvrir et dé-sélectionner/fermer un POI (Figure 7.9b). Par conséquent, l'action d'appuyer sur le bouton avec le pouce droit a une signification différente si le curseur se trouve à l'intérieur ou à l'extérieur d'un menu.

Comme un menu peut inclure plusieurs éléments, le curseur parcourant les éléments d'un menu peut se rapprocher d'autres POIs ouverts. Pour éviter un saut non désiré du curseur vers un autre menu ouvert adjacent, nous désactivons la technique d'expansion de cibles lorsque le curseur est à l'intérieur d'un menu. Elle sera réactivée une fois le curseur revenu au centre de l'écran.

Enfin, nous proposons deux techniques pour manipuler le curseur lorsqu'il a sauté sur un menu : (1) Pointage physique absolu : le curseur reste fixe sur l'écran et un élément dans le menu est sélectionné en déplaçant physiquement l'appareil (Figure 7.9f1). Nous appelons cette technique Voronoï Jumping-Physical (VJP). (2) Pointage tactile relatif : le curseur est fixe dans le monde physique (technique de pointage relatif [108]), et contrôlé grâce aux déplacements du pouce gauche sur l'écran (Figure 7.9f2). Nous appelons cette technique Voronoï Jumping-Relative (VJR).

Lors du couplage des deux techniques Jumping-Physical et Jumping-Relative avec la technique d'expansion de cibles (V), les utilisateurs peuvent (1) sélectionner/ouvrir un POI sans avoir à aligner le curseur avec la cible, (2) placer le curseur dans le menu en touchant l'écran du pouce gauche, et (3) effectuer un pointage physique ou un pointage relatif pour sélectionner un élément dans le menu. L'automate de la Figure 7.11 décrit l'ensemble de l'interaction bi-manuelle.

7.4 Conclusion

Dans ce chapitre, nous avons présenté notre contribution en termes de techniques de pointage pour la RA mobile. Notre étude envisage la tâche complète consistant à pointer un POI puis à interagir avec son menu. Une première contribution est d'appliquer une approche par expansion des POIs en RA mobile. Une telle approche se prête particulièrement bien sur des cibles numériques en RA mobile. Une deuxième contribution au delà du pointage d'une cible en RA mobile est de considérer l'interaction qui suit cette sélection. Dans notre cas, il s'agit de sélectionner un élément dans le menu du POI sélectionné. Nous nous sommes inspirés des travaux sur les menus en cascade exposés au chapitre précédent pour concevoir deux techniques qui facilitent la sélection d'un élément du menu correspondant à un POI.

Les techniques conçues résultantes (1) minimisent l'intrusion visuelle pour éviter de surcharger la zone centrale de l'écran, (2) ne brisent pas le lien qui unit les cibles numériques

à leur environnement physique, et (3) considèrent les informations contenues à l'intérieur des cibles. Ces techniques conçues pour l'aide au pointage en RA mobile répondent donc au trois critères énoncés au chapitre précédent (voir section 6.2).

Nous proposons dans le chapitre suivant une évaluation expérimentale des éléments de conception détaillés dans ce chapitre.

8

Exploration expérimentale des deux techniques de pointage pour la réalité augmentée mobile

Sommaire

8.1	Introduction	134
8.2	Plan d'expérience	135
8.2.1	Appareil et participants	135
8.2.2	Techniques de visualisation développées	135
8.2.3	Tâche expérimentale et environnement	135
8.2.4	Design expérimental et procédure	138
8.2.5	Hypothèses	139
8.2.6	Données collectées	139
8.3	Résultats	142
8.4	Discussion	143
8.5	Conclusion de l'expérience	146

8.1 Introduction

Ce chapitre est consacré à l'évaluation expérimentale des techniques d'interaction décrites dans le chapitre précédent. Nous décrivons une étude expérimentale qui compare les deux techniques Voronoi Jumping-Physical (VJP) et Voronoi Jumping-Relative (VJR) avec une technique d'expansion Voronoi (V) sans saut de curseur (pointage physique pour déplacer le curseur vers l'élément du menu, voir Figure 7.9c) et une technique de base (B), où le curseur reste fixe au centre de l'écran.

Le but de l'étude expérimentale est double. Le premier objectif est de quantifier les avantages de l'expansion de cibles pour pointer vers un POI en Réalité Augmentée (RA) mobile. Au-delà de l'optimisation du pointage uniquement, le deuxième objectif est de comparer les techniques conçues lors de l'exécution d'une tâche complète en sélectionnant d'abord un POI puis en interagissant avec le menu contenu au sein de ce POI.

Inspirée par le scénario 3 des cas d'utilisation industriels (voir section 2.3), l'étude considère

un environnement pseudo-industriel pour comparer les quatre techniques (VJP, VJR, V et B) lors de la sélection de POIs situés sur une machine et l'accès aux informations liées aux POIs.

8.2 Plan d'expérience

8.2.1 Appareil et participants

Les participants ont réalisé l'expérience sur un téléphone Lenovo phab2PRO intégrant la technologie Tango de Google. L'appareil est doté d'une combinaison de capteurs de profondeur et de suivi de mouvement, offrant à l'appareil une parfaite connaissance de l'espace (voir sous-section 2.4.3).

Le téléphone dispose également d'un écran de 6,4 pouces avec une résolution de 2560 x 1440 pixels qui a été pleinement utilisé dans cette étude expérimentale. Pour reproduire un environnement industriel, nous avons pris une photo d'un côté d'une machine de production de 9m x 3m. Nous l'avons imprimé sur une surface forex de 3mm de profondeur pour construire un panneau de 0,75m x 0,75m (voir Figure 8.1a). L'étude comprenait 12 volontaires non rémunérés d'un laboratoire universitaire en informatique (5 femmes, 7 hommes ; âgés de 23 à 41 ans, moyenne=29,6, écart-type=5,7). Bien que nos participants n'étaient pas des professionnels de la maintenance, l'objectif du contexte pseudo-industriel était de reproduire un arrière-plan (capture du flux vidéo) réaliste visuellement sur l'écran.

8.2.2 Techniques de visualisation développées

L'implémentation des trois techniques V, VJP et VJR suit le modèle de conception détaillé dans la chapitre 7.

L'apparence ("look") des POIs, des menus, du bouton de validation affiché dans le coin inférieur droit de l'écran ainsi que la technique de base et du curseur centré sur l'écran sont basés sur une application industrielle existante, l'application RA d'aide à la maintenance de Schneider (voir chapitre 2).

Les quatre techniques ont été développées en C# et affichées sur l'écran de l'appareil mobile à l'aide du moteur de rendu Unity3D.

8.2.3 Tâche expérimentale et environnement

La tâche expérimentale a été conçue pour reproduire le scénario 3 de nos cas d'usage industriels (voir section 2.3) : les participants ont été invités à (1) sélectionner/ouvrir des POIs situés sur une machine, et (2) sélectionner un élément du menu ouvert correspondant à ces POIs.

Pour cette étude expérimentale, nous avons utilisé une tâche de sélection de cibles 2D

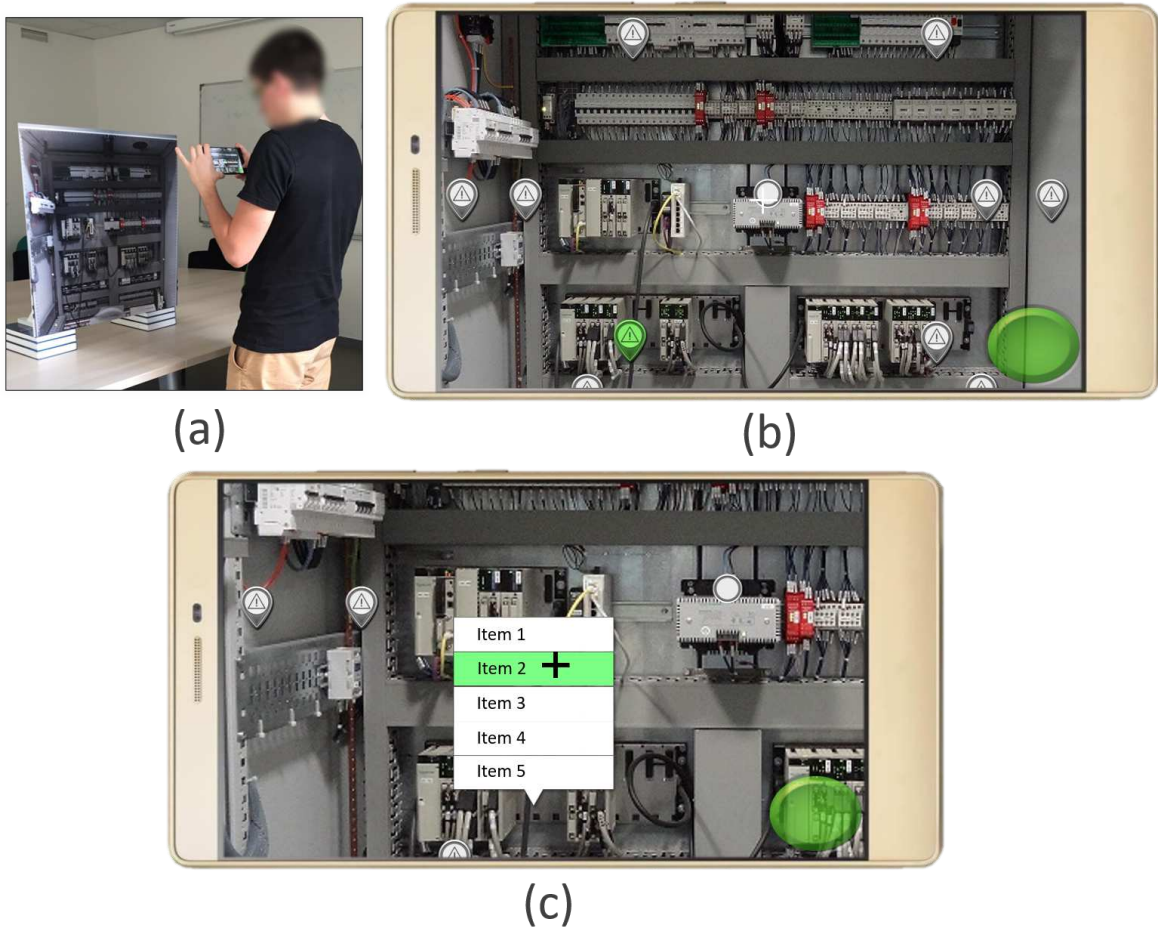


FIGURE 8.1 – (a) Un participant devant le modèle d'une machine de production. (b) Tâche de pointage pour sélectionner/ouvrir un POI. (c) Tâche de pointage pour sélectionner le deuxième élément d'un menu.

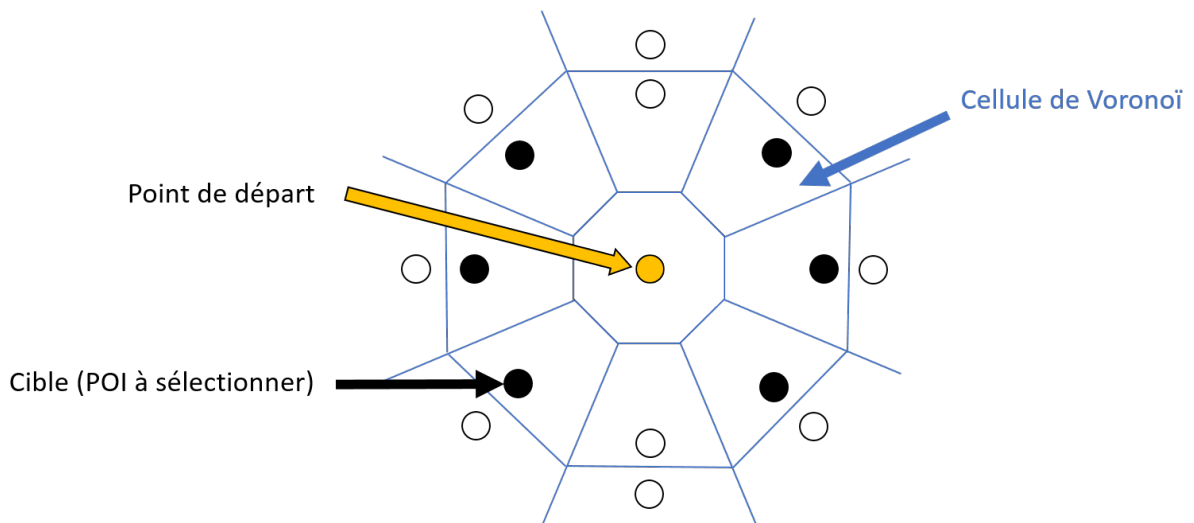


FIGURE 8.2 – Configuration des POIs en cercle et des cellules de Voronoi correspondantes.

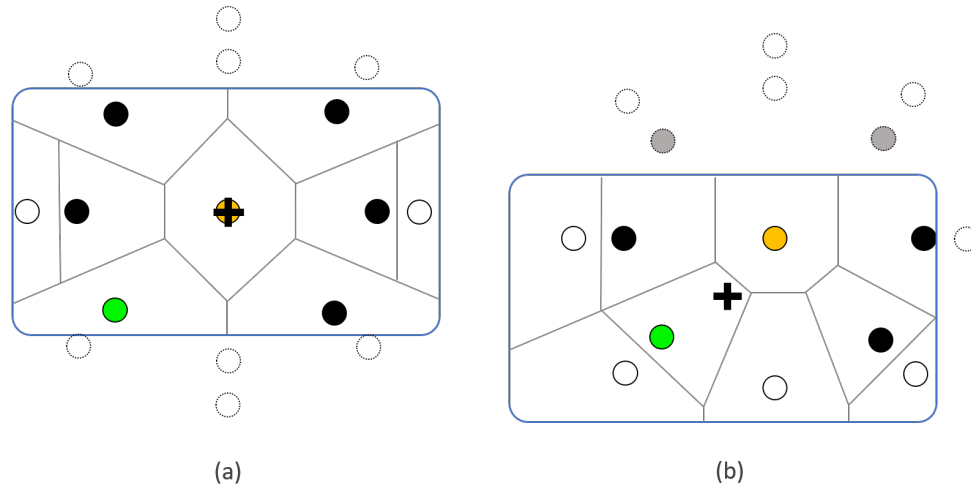


FIGURE 8.3 – Champ de vision de la caméra (rectangle bleu), partition de Voronoï (en gris), et POI à sélectionner (en vert) (a) 6 POIs du cercle interne sont visibles. (b) Déplacement de l'appareil mobile pour sélectionner le POI vert : le POI opposé sur le cercle devient hors-champ.

statiques, basée sur la norme ISO 9241-9. 8 POIs placés sur un cercle ont été présentés aux participants (voir Figure 8.1b). Pour limiter les zones étendues de ces POIs, 8 autres POIs ont été placés sur un cercle concentrique plus grand. Par conséquent, tous les POIs avaient la même taille étendue (voir Figure 8.2). Pour profiter pleinement des POIs élargis, les participants ont été invités à réaliser l'expérience à 40 cm de la machine (Figure 8.1a). Cela permet d'obtenir des cellules de Voronoï suffisamment étendues pour sélectionner/ouvrir un POI sans avoir le curseur placé près du POI.

La forme rectangulaire de l'appareil de l'écran a entraîné 2 problèmes : (1) les POIs supérieurs et inférieurs du cercle étaient placés en dehors de l'écran (voir Figure 8.3a), et (2) lorsqu'un POI du cercle était sélectionné, le POI opposé sur le cercle se trouvait toujours hors-champ (voir Figure 8.3b). Pour résoudre ce problème lié à la taille limitée de l'écran et à sa forme rectangulaire, nous avons (1) éliminé les POIs supérieurs et inférieurs du cercle de POIs à pointer, et (2) demandé aux participants de revenir à un point de départ entre chaque tâche de pointage. Le point de départ était placé au centre du cercle (voir Figure 8.2). Ce point de départ constituait une cible spéciale : sa zone étendue était calculée par la partition de Voronoï (voir Figure 8.2), mais sa sélection n'était validée que lorsque le curseur était aligné avec le point. Le curseur est au centre du cercle des POIs à pointer lorsqu'il est aligné avec le point de départ. Ainsi, nous garantissons que la prochaine cible à sélectionner sur le cercle soit obligatoirement visible à l'écran. De plus, considérer le point de départ dans la partition de Voronoï permettait de contraindre la taille de la zone étendue des POIs à sélectionner sur le cercle (voir Figure 8.3b) : lorsque le curseur arrive dans la cellule de Voronoï du POI à sélectionner, le POI est entouré de suffisamment de POIs pour avoir une zone étendue limitée, identique à celle des autres POIs à sélectionner sur le cercle.

Un POI à sélectionner sur l'écran était désigné par la couleur verte (voir Figure 8.1b).

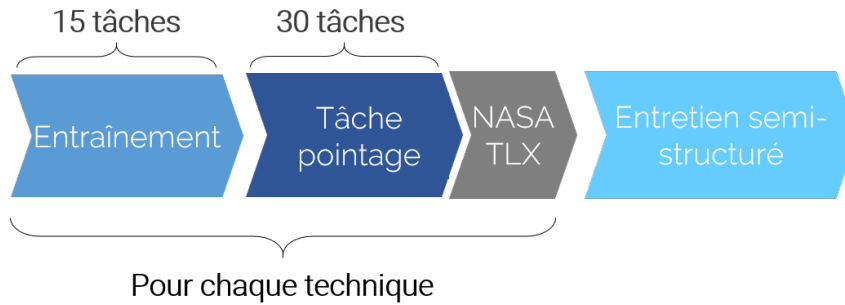


FIGURE 8.4 – Déroulement de l'expérience.

Après avoir sélectionné/ouvert un POI, les participants devaient sélectionner un élément donné du menu ouvert correspondant, également colorié en vert (voir Figure 8.1c). Ce menu s'ouvrait toujours au-dessus du POI sélectionné (voir Figure 8.1c). Comme expliqué au chapitre précédent, la sélection d'un POI et d'un élément de son menu est validée en appuyant sur un bouton "Valider" placé en bas à droite de l'écran. Nous avons figé la taille des menus à 5 éléments et l'élément du menu à sélectionner est le deuxième en partant du haut (voir Figure 8.1c). Une tâche de pointage ne se terminait que lorsque le POI demandé a été ouvert et que l'élément du menu correspondant a correctement été sélectionné. Dans le cas contraire, une erreur était comptabilisée, et ne permettait pas de valider la tâche tant que les participants n'avaient pas sélectionné le POI et l'élément du menu demandés.

8.2.4 Design expérimental et procédure

Design. L'étude est basée sur un design intra-sujets de 4 techniques de pointage (B, V, VJP, VJR). Nous avons divisé les participants en 4 groupes de 3 personnes. Chaque groupe a commencé l'expérience avec une technique différente. Chaque participant a effectué 120 tâches de pointage (4 techniques x 5 itérations x 6 POIs).

Procédure. Les participants ont d'abord reçu une brève explication sur les tâches expérimentales et les 4 techniques. En raison du nombre de techniques, ils ont été invités, pour chaque technique, à effectuer la session d'essai immédiatement après la session de formation [31] (voir Figure 8.4). Une session de formation consistait à réaliser 15 tâches de pointage (sélection/ouverture d'un POI + sélection d'un élément du menu). Pour chaque technique, l'étude expérimentale n'a commencé que lorsque les participants se sont sentis à l'aise avec la tâche. Sinon, ils devaient effectuer un autre bloc d'entraînement de 15 tâches de pointage. Ces longues séances de formation avant d'utiliser chaque technique ont été conçues pour permettre aux participants de pleinement se familiariser avec les techniques. Ceci est très important car les techniques sont de complexité différente mais aussi étroitement liées. Les longues sessions de formation ont ainsi été conçues pour atténuer le transfert de compétences entre les techniques. Cela nous a permis d'utiliser une conception expérimentale intra-sujet sans avoir besoin de contrebalancer pleinement les techniques entre les participants.

Les participants ont effectué la tâche à un endroit précis devant la machine marqué d'une croix blanche sur le sol. Les participants ont été invités à être aussi rapides et précis que possible. Après avoir terminé toutes les tâches de pointage avec une seule technique, les participants ont rempli un questionnaire Raw TLX [61] (voir Annexe B) et sont retournés à l'emplacement marqué pour commencer avec la technique suivante (voir Figure 8.4). Nous avons conclu l'étude par un entretien semi-structuré pour collecter des éléments subjectifs de comparaison entre les techniques. Ce questionnaire est décrit en Annexe C.

L'expérience complète avait une durée moyenne de 30 minutes.

8.2.5 Hypothèses

Nous avons formulé les hypothèses suivantes :

- **H1.** Les techniques V, VJP et VJR permettent un pointage plus rapide que la technique de référence (B) lors de la sélection/ouverture des POIs.
- **H2.** La technique VJP est perçue comme plus utilisable que la technique VJR : le déplacement physique de l'appareil vers un POI est la technique par défaut pour la première partie de la tâche (ouverture d'un POI). VJR implique de changer de mode d'interaction pour la deuxième partie de la tâche (sélection d'un élément dans un menu). Cette modification peut perturber l'utilisateur dans l'accomplissement de la tâche.

8.2.6 Données collectées

Pour vérifier nos hypothèses, nous avons mesuré le temps passé à (1) sélectionner/ouvrir un POI (tâche T1) (pour vérifier **H1**), et (2) sélectionner l'élément de menu (tâche T2). Le temps global est égal à $T1 + T2$. Nous avons également enregistré le nombre d'erreurs lors de la sélection d'un élément à l'écran : une erreur a été comptée lorsque les participants ont sélectionné/ouvert un POI incorrect ou sélectionné un élément incorrect dans un menu.

Nous avons également considéré des mesures subjectives pour vérifier **H2**. Après avoir terminé toutes les tâches de pointage pour une technique, les participants ont été invités à remplir un formulaire Raw TLX [61] (voir Annexe B), évalué sur une échelle de Likert à 5 valeurs [72]. À la fin de la session, les participants devaient évaluer, pour chaque technique, la fatigue mentale, physique et la facilité d'apprentissage. De même, les questions étaient à évaluer sur une échelle de Likert à 5 niveaux [72]. Enfin, les participants ont également été invités à classer les 4 techniques par ordre de préférence (voir Annexe C).

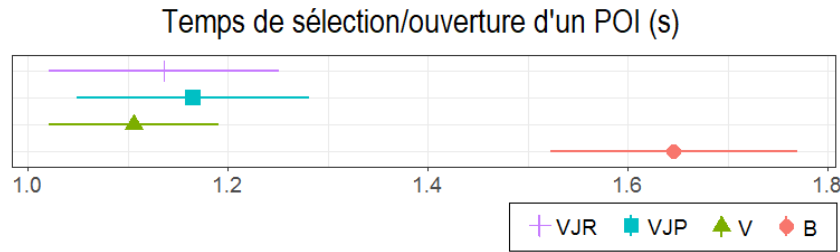


FIGURE 8.5 – Temps (s) de sélection/d'ouverture d'un POI pour toutes les techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.

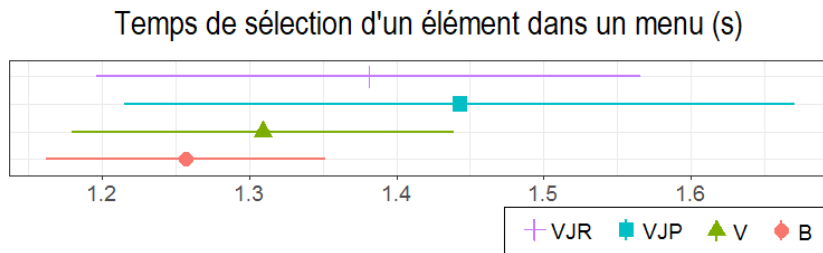


FIGURE 8.6 – Temps (s) de sélection d'un élément dans un menu pour toutes les techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.

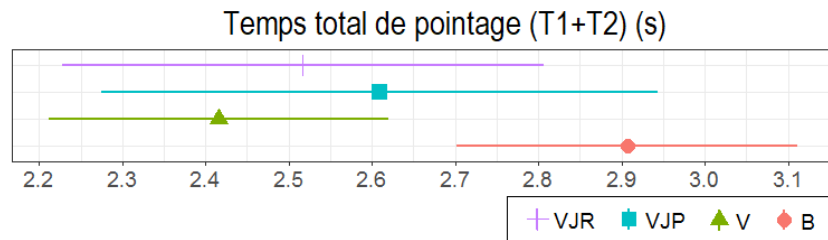


FIGURE 8.7 – Temps (s) total de complétion d'une tâche de pointage (T1 :sélection d'un POI + T2 :sélection d'un élément dans un menu) pour toutes les techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.

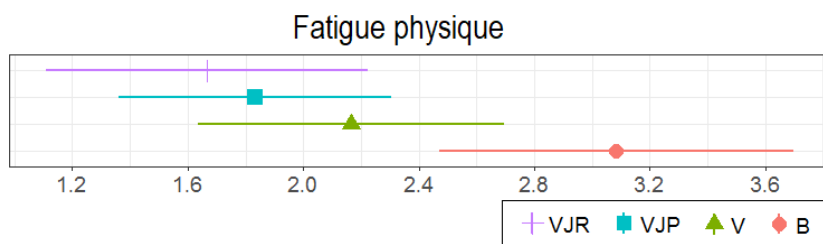


FIGURE 8.8 – Fatigue physique sur 5 (1 = très mauvais, 5 = très fatiguant) ressentie par les participants après utilisation des techniques. Les barres d'erreur indiquent des intervalles de confiance à 95%.

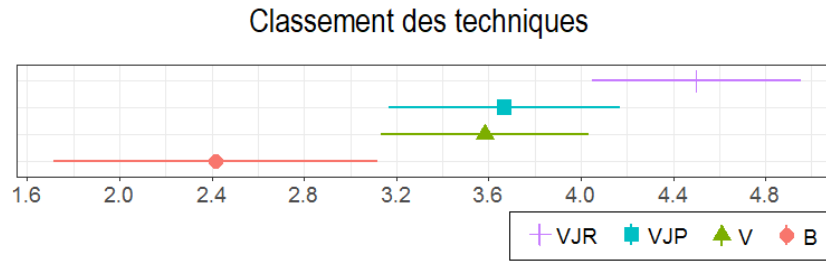


FIGURE 8.9 – Notes sur 5 (1 = très mauvais, 5 = très bien) données aux techniques. Les barres d’erreur indiquent des intervalles de confiance à 95%.

-	Temps sélection POI (s)	Temps sélection élément menu (s)	Temps global (s)
B	1.65 ; 1.6 (0.22)	1.26 ; 1.26 (0.17)	2.91 ; 2.93 (0.36)
V	1.11 ; 1.12 (0.15)	1.31 ; 1.2 (0.23)	2.42 ; 2.34 (0.36)
VJP	1.17 ; 1.13 (0.2)	1.45 ; 1.3 (0.4)	2.61 ; 2.43 (0.59)
VJR	1.14 ; 1.12 (0.2)	1.38 ; 1.35 (0.33)	2.52 ; 2.46 (0.51)

TABLE 8.1 – Moyenne, médiane (et écart-type) des temps (s) de sélection d’un POI, d’un élément dans un menu ainsi que du temps total pour accomplir la tâche de pointage pour les techniques B, V, VJP et VJR

-	Fatigue mentale	Fatigue physique	Facilité d’apprentissage	Classement
B	1.75 ; 1.5 (0.97)	3.08 ; 3 (1.08)	3.67 ; 5 (1.97)	2.42 ; 2 (1.24)
V	1.67 ; 1 (0.98)	2.17 ; 2 (0.94)	3.42 ; 4 (1.83)	3.58 ; 4 (0.79)
VJP	2.25 ; 2 (0.87)	1.83 ; 2 (0.83)	3.17 ; 3 (1.27)	3.67 ; 4 (0.89)
VJR	1.92 ; 1.5 (1.24)	1.67 ; 1 (0.98)	3.25 ; 3 (0.97)	4.5 ; 5 (0.8)

TABLE 8.2 – Moyenne, médiane (et écart-type) des notes sur une échelle de 5 valeurs (1 = très mauvais, 5 = très bien) données aux techniques pour la fatigue mentale, la fatigue physique, la facilité d’apprentissage, et le classement général

8.3 Résultats

8.3.0.1 Temps de sélection et taux d'erreur

Nous rapportons les résultats en considérant les effets de chaque technique sur le taux d'erreur et le temps nécessaire pour (1) sélectionner/ouvrir un POI et (2) sélectionner un élément d'un menu. Un test de Shapiro-Wilk a indiqué que le temps pour accomplir l'ensemble de la tâche (T1+T2) et le temps d'ouverture de POI suivent une distribution normale (voir Tableau 8.1). Une ANOVA à un facteur n'a pas trouvé d'effet important des techniques sur le temps total de complétion de la tâche ($F_{(3,44)} = 2.471, p = 0.07$). Cependant, il a montré une différence statistiquement significative des techniques sur le temps nécessaire pour sélectionner un POI ($F_{(3,44)} = 20.47, p < 0.001$). Nous avons effectué un test post-hoc à l'aide d'un test-t par paires et de corrections de Bonferroni : le résultat de ce test a révélé que statistiquement, les participants étaient significativement plus longs pour ouvrir les POIs avec la technique de référence (B) qu'avec V ($p < 0.001$), VJP ($p < 0.001$) et VJR ($p < 0.001$). Les données de temps de sélection d'un élément dans un menu ne suivent pas une distribution normale. Le test non-paramétrique de Kruskal-Wallis n'a pas révélé d'effet important des techniques sur le temps nécessaire pour sélectionner un élément dans un menu ($\chi^2_3 = 0.93, p = 0.817$).

Les données sur le taux d'erreur ne suivent pas une distribution normale. Le test de Kruskal-Wallis n'a pas révélé d'effet important des techniques sur le taux d'erreur, soit pour sélectionner un POI ($\chi^2_3 = 3.93, p = 0.26$), soit pour sélectionner un élément d'un menu ($\chi^2_3 = 6.34, p = 0.09$).

8.3.0.2 Préférences subjectives

Résultats Raw TLX. Un test de Friedman a révélé un effet des techniques sur l'exigence mentale ($\chi^2_3 = 8.2, p=0.042$), et la frustration ($\chi^2_3 = 7.93, p=0.047$) ressentie pendant la tâche d'exploration. Une analyse post-hoc avec comparaisons par paires et corrections de Bonferroni a révélé que la technique VJP a demandé plus d'effort mental que la technique V ($p=0.03$). Concernant la frustration, le test post-hoc n'a révélé aucune différence statistiquement significative entre les facteurs.

Aucune différence statistiquement significative n'a été constatée dans toutes les autres mesures : exigence physique ($\chi^2_3 = 5.2, p=0.16$), exigence temporelle ($\chi^2_3 = 1.78, p=0.62$), performance ($\chi^2_3 = 7.55, p=0.06$) et effort ($\chi^2_3 = 3.04, p=0.38$).

Résultats du questionnaire en fin de session. Le Tableau 8.2 présente les résultats de notre questionnaire. Un test non-paramétrique de Friedman a trouvé un fort effet des techniques sur la fatigue physique ($\chi^2_3 = 16.9, p < 0.001$), et sur la note globale ($\chi^2_3 = 16.1, p = 0.001$). Les comparaisons par paires avec corrections de Bonferroni ont révélé que la technique de référence a été (1) notée beaucoup plus fatigante physiquement que VJP ($p < 0.01$) et VJR ($p < 0.05$), et (2) notée comme la pire technique par rapport

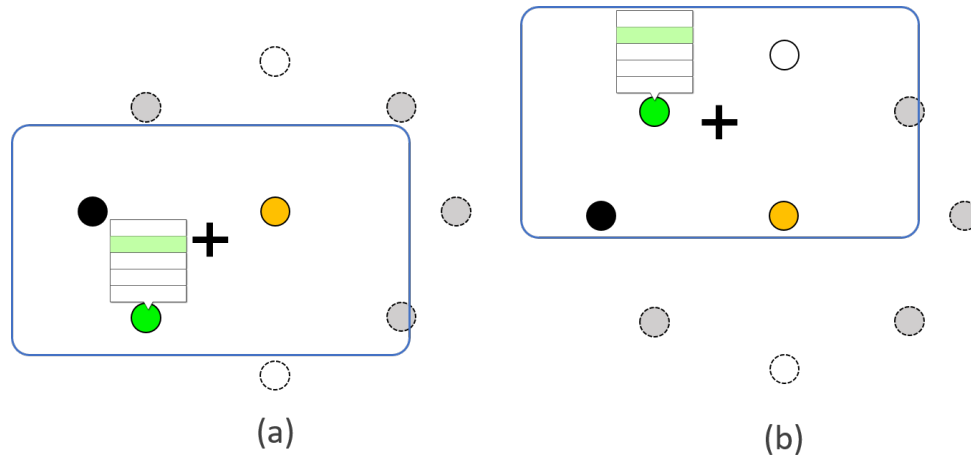


FIGURE 8.10 – (a) POI ouvert dans la section inférieure de l’écran (rectangle bleu) : courte distance entre le curseur et l’élément vert du menu. (b) POI ouvert dans la section supérieure de l’écran (rectangle bleu) : importante distance entre le curseur et l’élément vert du menu.

à V ($p < 0.01$) et VJR ($p < 0.01$), tandis que VJR a été la technique préférée par les participants. Aucune différence statistiquement significative n’a été trouvée pour toutes les autres mesures : fatigue mentale ($\chi_3^2 = 3.79, p = 0.28$) et facilité d’apprentissage ($\chi_3^2 = 2.32, p = 0.50$).

8.4 Discussion

Ouverture de POIs et sélection d’éléments dans les menus. Le temps de sélection d’un POI avec la technique de référence (B) est le plus long (voir Figure 8.5). Le temps nécessaire pour sélectionner/ouvrir les POI était 31% plus long avec la technique de base qu’avec toute autre technique. Cela confirme notre hypothèse **H1**. Même si nous n’avons pas trouvé de différences statistiquement significatives sur les taux d’erreur, tous les participants ont déclaré que la technique de base nécessitait une précision et une attention supplémentaires pour sélectionner correctement un POI à l’écran. Nos résultats confirment que les techniques d’expansion des cibles sont efficaces pour faciliter les tâches de pointage, et qu’elles peuvent être appliquées efficacement aux cibles de réalité augmentée. Bien que les techniques VJP et VJR réduisent la distance pour atteindre un élément d’un menu, nous n’avons pas trouvé de différence statistiquement significative entre toutes les techniques sur le temps nécessaire pour atteindre l’élément du menu (voir Figure 8.6).

Distance pour atteindre un élément d’un menu. Le deuxième élément à sélectionner était toujours situé dans la section supérieure du menu. Comme les menus sont toujours ouverts au-dessus des POIs, deux cas se présentent lors de l’utilisation de techniques d’expansion de cible (V, VJP, VJR) : (1) le POI ouvert est situé dans la partie inférieure de l’écran, et la distance entre le curseur et l’élément du menu à sélectionner est réduite, car le curseur est déjà proche de la partie supérieure du menu (voir Figure 8.10a). (2) Le POI ouvert se trouve dans la partie supérieure de l’écran et le chemin pour atteindre

l'élément du menu est plus long, car le curseur est situé sous le menu (voir Figure 8.10b).

Par rapport à VJP et VJR, la technique V est la seule qui permet aux participants de pointer directement sur l'élément d'un menu une fois le POI ouvert. Avec VJP et VJR, le chemin est indirect car le curseur saute d'abord en bas du menu, puis doit être déplacé vers l'élément souhaité. Par conséquent, la technique V est privilégiée (resp. désavantagée) avec les POIs situés sur la partie inférieure (resp. supérieure) de l'écran. En examinant uniquement les POIs de la partie inférieure de l'écran, le test non-paramétrique de Kruskal-Wallis a révélé un fort effet des techniques sur le temps de sélection d'un élément dans un menu ($\chi_3^2 = 8.42, p = 0.04$). Les comparaisons par paires avec corrections de Bonferroni ont montré que les participants ont pris plus de temps pour sélectionner un élément dans un menu avec VJP qu'avec V. Pour les POIs situés sur la partie supérieure de l'écran, nous n'avons pas trouvé d'effet fort des techniques sur le temps de sélection dans un menu : bien que les techniques VJP et VJR réduisent la distance pour atteindre un élément d'un menu à l'aide du saut de curseur, l'interaction et le changement de contexte induisent deux temps de réaction. Un premier temps de réaction est nécessaire lorsqu'un POI a été sélectionné/ouvert, afin de passer à une interaction à deux mains et de faire sauter le curseur. Une fois le saut de curseur effectué, un autre temps de réaction est nécessaire pour (1) repérer la nouvelle position du curseur sur l'écran et (2) comprendre le nouveau contexte dans lequel se trouve le curseur.

Changement de modalité d'interaction. Pour effectuer la tâche composée de la sélection d'un POI et de la sélection d'un élément dans un menu, les techniques VJP et VJR modifient toutes deux (1) le nombre de mains nécessaires pour effectuer la tâche, car les participants doivent également utiliser leur main gauche pour faire sauter le curseur sur le menu, et (2) le contexte visuel entourant le curseur lorsqu'il saute à un autre endroit de l'écran. En plus de ces changements, la technique VJR implique de basculer d'une modalité de pointage physique avec l'appareil (pour sélectionner/ouvrir un POI) à une modalité tactile (pour déplacer le curseur dans un menu à l'aide du pouce sur l'écran). Même en considérant ces changements d'interaction et de contexte, la technique VJR a été la mieux classée (voir Figure 8.9), contredisant notre hypothèse **H2**. 10 des 12 participants ont rapporté que VJR limitait la fatigue physique. Bien que nous n'ayons pas trouvé de différences statistiquement significatives entre VJR et VJP sur la fatigue physique, les participants ont déclaré qu'avec VJR, ils n'étaient plus obligés de déplacer l'appareil pour sélectionner un élément dans un menu. Cela est par conséquent cohérent avec la première partie de la tâche de pointage : une technique d'expansion de cible limite les mouvements physiques nécessaires pour sélectionner un POI, tandis que le pointage relatif limite les mouvements physiques en permettant au curseur d'être déplacé à l'aide du pouce à l'écran. 6 des 12 participants ont déclaré qu'ils préféraient déplacer leur pouce gauche avec VJR que d'effectuer des mouvements physiques avec VJP : alors que VJP utilise le pouce gauche uniquement pour faire sauter le curseur sur un menu, VJR profite du pouce gauche sur écran pour permettre le déplacement du curseur. Par conséquent, les participants ont trouvé cohérent de laisser le pouce gauche sur l'écran et de l'utiliser

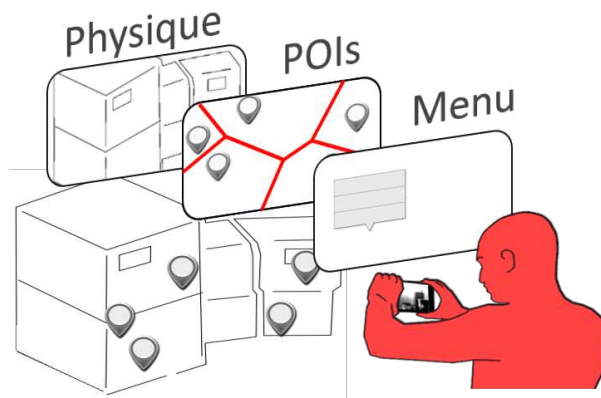


FIGURE 8.11 – Calques de réalité augmentée : monde physique - POIs - menus.

pour sélectionner un élément dans un menu.

Calques de réalité augmentée comme modèle mental. Que VJP ou VJR ait été utilisé, 4 participants ont indiqué que l'interaction avec le pouce gauche avait ajouté un "calque" pour les aider dans leur tâche de pointage : pendant la première partie de la tâche, les utilisateurs doivent sélectionner un POI. Ils parcourent un premier calque contenant tous les POIs (voir Figure 8.11). Une fois ouvert, un appui du pouce gauche sur l'écran fait sauter le curseur sur le menu correspondant. Ceci a été perçu comme un nouveau calque contenant uniquement le menu. Ce modèle mental les a aidé à ignorer le reste des objets numériques à l'écran. Une fois l'élément du menu validé, le curseur revient au centre de l'écran, de retour au calque contenant tous les POIs (voir Figure 8.11).

Tâche globale. Nous n'avons pas trouvé d'effet important de VJP et VJR sur le temps global pour terminer la tâche composée (voir Figure 8.7). Le gain obtenu en réduisant la distance entre le curseur et le menu est compensé avec le temps nécessaire aux utilisateurs pour traiter mentalement le changement de modalités et le changement de contexte autour du curseur. Cependant, les participants ont rapporté significativement moins de fatigue physique avec les techniques VJP et VJR qu'avec la technique de référence B (voir Figure 8.8). Enfin, VJR a obtenu la meilleure note globale (voir Figure 8.9) : les participants (1) ont trouvé plus facile de continuer à utiliser le pouce gauche pour interagir avec le curseur et (2) n'étaient plus tenus de déplacer physiquement l'appareil. Bien qu'aucune différence significative n'ait été trouvée quant à la facilité d'apprentissage des techniques, tous les participants ont indiqué que la phase d'apprentissage avec VJP et VJR nécessitait plus d'attention qu'avec les techniques V et B. En outre, 9 des 12 participants ont eu besoin de 2 blocs de formation pour apprendre à correctement gérer l'interaction à deux mains. Ainsi, les utilisateurs effectuant de nombreuses tâches de pointage, comme les opérateurs de maintenance, trouveront un avantage à consacrer un effort supplémentaire à l'apprentissage de la technique VJR, pour être efficaces à long terme tout en minimisant la fatigue physique.

8.5 Conclusion de l'expérience

Les techniques d'expansion de cibles facilitent le pointage en élargissant les tailles effectives des cibles. Bien que ces techniques ne soient pas nouvelles, leur application au pointage de POIs en RA est nouvelle. Contrairement aux nombreuses études sur l'expansion de cibles se concentrant uniquement sur l'optimisation du pointage, notre étude se focalise sur la tâche complète consistant à pointer vers un POI, puis à interagir avec son menu.

Les résultats montrent que : (1) les techniques d'expansion des cibles sont efficaces pour faciliter la sélection de POIs et s'adaptent bien aux applications RA mobiles. (2) Lorsque l'on considère une tâche composée d'une sélection de POI et d'une sélection d'un élément dans un menu, la combinaison du pointage physique avec le pointage relatif est préférable car elle minimise la fatigue physique. De plus l'étude souligne aussi l'importance de considérer une tâche complète comme ici l'interaction avec le menu obtenu après sélection.

Pour analyser en détail la différence entre les quatre techniques pour la sélection d'un élément de menu, il conviendrait de mener une autre étude expérimentale en faisant varier les éléments du menu à sélectionner. En effet l'étude menée a identifié des situations où certaines techniques sont favorisées/défavorisées par la position de l'élément du menu.

L'étude détaillée dans ce chapitre a été menée dans un contexte pseudo-industriel. De ce fait, la poursuite de ces travaux s'orientera aussi vers une évaluation de la technique VJR dans des contextes industriels professionnels afin de recueillir les retours des opérateurs de maintenance.

9

Sommaire

9.1 Contributions	147
9.1.1 Techniques de visualisation hors-champ en RA	147
9.1.2 Techniques de pointage en RA	148
9.1.3 Diffusion et brevet	149
9.2 Perspectives	150
9.2.1 Perspectives à court terme : extension des travaux effectués . .	150
9.2.2 Perspectives à long terme : ouverture vers d'autres travaux . .	153

9.1 Contributions

En réalité augmentée (RA), l'utilisateur est amené à sélectionner des cibles numériques qui regroupent des informations pertinentes sur le monde réel. Nos contributions sont dédiées aux deux étapes consécutives qui sont : explorer l'environnement pour ramener les cibles numériques dans le champ vision de la caméra puis sélectionner les cibles. Ainsi, nous avons étudié des techniques permettant (1) d'étendre le champ de vision de la caméra lorsque l'utilisateur explore l'environnement RA et (2) de faciliter la sélection des cibles et des informations qui y sont contenues lorsque celles-ci sont dans le champ de vision de la caméra.

9.1.1 Techniques de visualisation hors-champ en RA

Pour compenser le champ de vision restreint de la caméra du dispositif mobile utilisé pour explorer l'environnement RA, nous avons introduit trois nouvelles techniques de visualisation hors-champ : Halo3D, Halo3D* et Halo3D**. Ces trois techniques sont basées sur Halo [13], une visualisation qui affiche des arcs de cercles à l'écran pour indiquer la direction et la distance de chaque cible numérique hors-champ.

Nous avons tout d'abord proposé une adaptation de Halo [13] à la réalité augmentée mo-

bile : nous projetons une cible hors-champ sur le plan de l'écran de l'appareil mobile. Nous déterminons ensuite le rayon du cercle en calculant la distance qui sépare la projection de la cible du bord de l'écran le plus proche. A cette distance nous rajoutons un décalage constant afin que le cercle soit partiellement visible sur l'écran. Cependant l'affichage permanent à l'écran des halos correspondant à toutes les cibles numériques hors-champ conduit à une surcharge visuelle de l'interface graphique. Nous avons ainsi conçu trois méthodes d'agrégation des cercles, pour réduire la surcharge visuelle :

- Halo3D utilise une agrégation basée sur les centroïdes des projections des POIs hors-champ sur le plan de l'appareil mobile : l'agrégation proposée est uniquement basée sur l'affichage des indices vers les cibles, les cercles. Ainsi, deux cercles sont agrégés dès qu'ils se touchent à l'écran. Le centre du nouveau cercle d'agrégation affiché à l'écran est placé sur le centroïde des cibles hors-champ agrégées.
- Halo3D* reprend la technique précédente mais autorise les chevauchements de cercles jusqu'à un seuil.
- Halo3D** affiche la distribution spatiale des cibles hors-champ agrégées en affichant des courbes au lieu de cercles. Halo3D** a fait l'objet d'un dépôt de brevet par Schneider Electric.

Les avantages des trois techniques sont triples : (1) fournir des informations sur l'environnement hors-champ tout en limitant l'intrusion visuelle sur l'écran, (2) gérer une forte densité de cibles numériques, (3) indiquer l'amplitude du mouvement physique nécessaire pour ramener une cible numérique hors-champ dans le champ de vision de la caméra.

Nous avons expérimentalement étudié les avantages des techniques tout d'abord en les comparant avec une technique basée sur l'affichage de points sur la périphérie de l'écran, *AroundPlot**, et une technique basée sur des flèches, *Arrows2D*. La tâche expérimentale consistait à ramener des cibles dans le champ de vision de la caméra. Les résultats ont montré que les utilisateurs étaient 18% plus rapides avec Halo3D qu'avec *AroundPlot** et aussi rapide qu'*Arrows2D* tout en étant perçu comme 34% moins intrusif à l'écran. Nous avons ensuite étudié expérimentalement nos techniques en contexte pseudo-industriel. Les résultats ont révélé que les deux variantes de Halo3D, Halo3D* et Halo3D**, nécessitaient moins d'effort que *AroundPlot** pour localiser une cible numérique dans l'environnement et permettaient une compréhension efficace de l'environnement hors-champ.

9.1.2 Techniques de pointage en RA

Tandis que le pointage est largement étudié dans la communauté IHM, l'originalité de notre démarche de recherche est de mettre en contexte la tâche de pointage en RA : en effet, au-delà de la sélection d'une cible, nous avons aussi considéré une tâche complète consistant à pointer une cible puis à interagir avec les informations contenues à l'intérieur de ces cibles. Nous avons ainsi introduit deux nouvelles techniques de pointage bi-manuelle (Voronoi Physical Pointing - VJP, et Voronoï Relative Pointing - VJR) pour faciliter la sélection d'une cible numérique et des éléments qui y sont contenus.

Ces techniques sont basées sur (1) une technique d'expansion de cibles, pour faciliter leur sélection en allouant à chaque cible une zone active plus grande et (2) un saut de curseur, pour raccourcir la distance entre le curseur et les informations regroupées au sein de la cible numérique.

Nous avons étudié l'approche basée sur le partitionnement de Voronoï pour augmenter la taille des cibles dans le cas de la réalité augmentée mobile. Pour cela, nous avons considéré (1) un environnement 3D (points situés autour de l'utilisateur) et (2) l'aspect dynamique d'une tâche de pointage en RA (caméra en mouvement). Nous avons ensuite étudié comment ramener le curseur sur l'information contenue dans une cible. En particulier, faire sauter le curseur raccourcit la distance entre le curseur et les informations regroupées au sein de la cible numérique. Nos deux techniques VJP et VJR se distinguent par la façon de manipuler le curseur une fois qu'il a sauté sur la cible numérique :

- Voronoi Physical Pointing (VJP) : le saut de curseur est actif en posant le pouce gauche sur l'écran, et ne peut être déplacé qu'en bougeant physiquement l'appareil mobile vers l'information souhaitée.
- Voronoi Relative Pointing (VJR) : le saut de curseur est actif en posant le pouce gauche sur l'écran, et ne peut être déplacé qu'en faisant glisser le doigt sur l'écran.

Nos deux techniques d'aide au pointage VJP et VJR (1) évitent toute intrusion visuelle à l'écran, (2) ne brisent pas le lien qui unit les cibles numériques à leur environnement physique, et (3) considèrent la sélection des informations contenues à l'intérieur des cibles.

Nous avons mené une étude expérimentale en contexte pseudo-industriel. La tâche consistait à sélectionner des cibles numériques placées sur une machine, puis à cliquer sur un élément du menu correspondant à chaque cible sélectionnée. Les résultats expérimentaux ont montré que : (1) les techniques d'expansion des cibles sont efficaces pour faciliter la sélection de cibles numériques et s'adaptent bien aux applications RA mobiles. (2) Lorsque l'on considère une tâche composée d'une sélection d'une cible numérique et d'une sélection d'un élément dans un menu, la technique VJR est préférée car elle minimise la fatigue physique.

Les techniques d'assistance au pointage en RA mobile présentées dans ce manuscrit ont été intégrées dans les systèmes professionnels de maintenance de Schneider. Ces travaux mettent bien en avant la démarche de recherche adoptée qui part de besoins utilisateurs pour étudier des techniques d'interaction pour des tâches génériques. Les techniques d'interaction conçues ont fait l'objet d'études expérimentales contrôlées en laboratoire et en contexte pseudo industriel et ont été intégrées dans une application industrielle.

9.1.3 Diffusion et brevet

L'ensemble de ces contributions a donné lieu aux publications suivantes et à un brevet :

- Patrick Perea, Denis Morand, and Laurence Nigay. 2017. *Halo3D : a technique for visualizing off-screen points of interest in mobile augmented reality*. In Proceedings of the 29th Conference on l'Interaction Homme-Machine (IHM '17). Association for Computing Machinery, New York, NY, USA, 43–51.
- Patrick Perea, Denis Morand and Laurence Nigay. 2017. *[POSTER] Halo3D : A Technique for Visualizing Off-Screen Points of Interest in Mobile Augmented Reality*. IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), Nantes, 170-175.
- Patrick Perea, Denis Morand, and Laurence Nigay. 2019. *Spotlight on Off-Screen Points of Interest in Handheld Augmented Reality : Halo-based techniques*. In Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces (ISS '19). Association for Computing Machinery, New York, NY, USA, 43–54.
- Géraldine Fabre. Automne 2018. *Les Nouvelles Frontières de la Réalité Augmentée*. Magasine [H]auteurs, Université Grenoble Alpes, 10-11.
- Patrick Perea, Denis Morand, and Laurence Nigay. 2020. *Target Expansion in Context : the Case of Menu in Handheld Augmented Reality*. In Proceedings of the 2020 International Conference on Advanced Visual Interfaces (AVI '20). Association for Computing Machinery, New York, NY, USA (à paraître).

Au-delà des publications scientifiques et de l'article de vulgarisation, les travaux ont aussi donné lieu à un brevet déposé par Schneider Electric Industries SAS :

- Brevet Schneider Electric Industries SAS, déposé le 30/01/19, *Graphical User Interface For Indicating Off-screen Points of Interest*, N° :19305112.5 - 1221.

9.2 Perspectives

Nos travaux sur les techniques visualisation hors-champ et d'assistance au pointage en RA ouvrent la voie à plusieurs perspectives de recherche à court terme et à plus long terme.

9.2.1 Perspectives à court terme : extension des travaux effectués

Pour les techniques de visualisation hors-champ et de sélection de cibles, nous avons montré des différences de performances et de préférences lors d'études expérimentales contrôlées en laboratoire. D'autres études restent à mener pour compléter ces résultats. En particulier, il serait pertinent de mener des expériences en contexte avec des utilisateurs experts métiers à plus grande échelle que celles que nous avons menées avec quelques experts métiers. L'accès à un contexte applicatif dans le cadre de ces travaux a été très important pour obtenir des premiers retours lors de la conception. L'étape suivante serait de tester en contexte les techniques finales conçues.

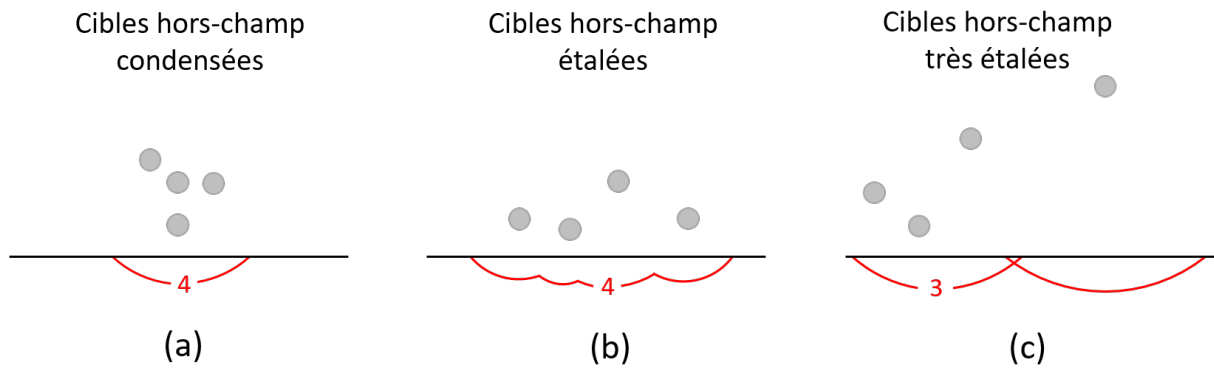


FIGURE 9.1 – Adaptation automatique des techniques de visualisation hors-champ en fonction de la disposition spatiale des cibles hors-champ. (a) Halo3D convient lorsque les cibles sont condensées autour d’une position. (b) Halo3D** s’adapte mieux lorsque les cibles d’un paquet s’étalent dans l’espace hors-champ. (c) Halo3D* est la technique la plus appropriée lorsque les cibles sont très dispersées dans l’espace hors-champ.

Au-delà de ces études expérimentales en contexte, nous identifions des études à court terme qui étendent directement nos travaux.

9.2.1.1 Techniques de visualisation hors-champ en RA

Nous avons déterminé qu’aucune des technique Halo3D, Halo3D* et Halo3D** n’est parfaite pour toutes les configurations de cibles numériques. Halo3D est la technique qui minimise le plus l’intrusion visuelle à l’écran : seuls des cercles sont utilisés et aucun chevauchement entre ces cercles n’est autorisé à l’écran. Cependant, cette technique n’affiche pas la répartition spatiale des cibles numériques et, dans un environnement très dense, peut induire en erreur si trop d’agrégations sont effectuées. Une solution est d’utiliser Halo3D*. Mais si les cibles numériques sont très étalées, Halo3D* peut afficher trop de chevauchements à l’écran et Halo3D** peut dessiner une seule courbe qui longent tous les bords de l’écran.

Dans ce contexte, une piste de recherche serait d’adapter automatiquement la technique de visualisation à l’environnement hors-champ. En particulier, une future étude poserait le problème de changement de visualisation et celui du transfert d’expertise d’une technique à l’autre. L’enjeu est de déterminer les paramètres de l’environnement qui seront déterminant pour le changement de techniques. Un premier paramètre est l’étendue spatiale des paquets (clusters) de cibles hors-champ : si les cibles sont regroupées dans un paquet condensé, il est possible d’utiliser Halo3D car la visualisation de la répartition spatiale n’est ici pas utile. En revanche dès que les cibles d’un même paquet hors-champ s’étalent dans l’espace, Halo3D* et Halo3D** sont plus adaptées car elles permettent l’affichage de l’étendue des paquets de cibles hors-champ (voir Figure 9.1). Un deuxième paramètre à considérer est le nombre de chevauchements visibles entre les halos à l’écran. Halo3D** autorise les chevauchements entre les cercles jusqu’à un certain seuil. Mais si le nombre de chevauchements est important, la surcharge visuelle diminuera la lisibilité de l’inter-

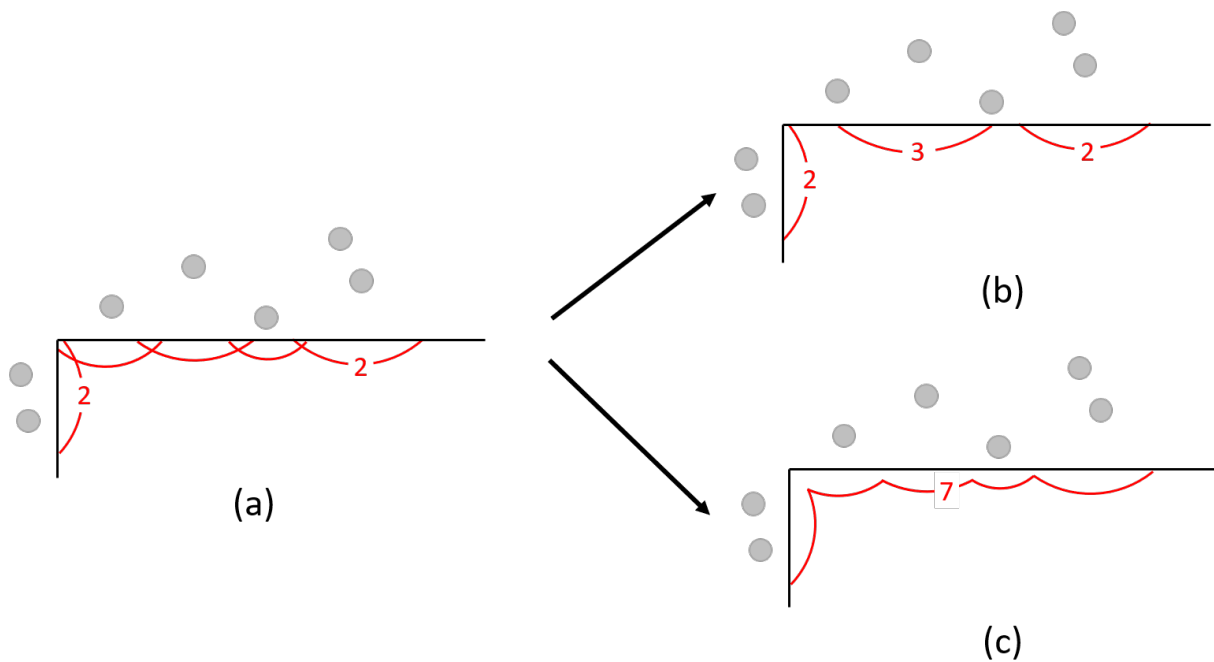


FIGURE 9.2 – Adaptation automatique des techniques de visualisation hors-champ en fonction du nombre de chevauchements visibles entre les cercles à l'écran. (a) Halo3D* ne convient pas lorsqu'un nombre important de chevauchements est présent à l'écran. Dans ce cas, deux techniques sont plus adaptées : Halo3D (b) ou Halo3D** (c).

face graphique (voir Figure 9.2). Halo3D ou Halo3D** seraient alors plus adaptés car ces techniques n'impliquent aucun chevauchement à l'écran. Une approche expérimentale pourrait permettre de démarrer cette étude de l'adaptation automatique de la visualisation. Ainsi, une première étude expérimentale consistera à explorer un environnement RA en adaptant automatiquement les techniques en fonction de l'étendue spatiale des paquets de cibles hors-champ. Puis une deuxième expérience étudiera l'adaptation automatique des techniques en fonction du nombre de chevauchements visibles à l'écran.

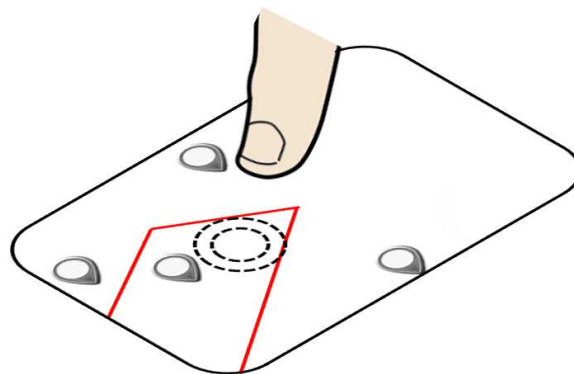


FIGURE 9.3 – Pointage tactile à l'intérieur d'une cellule Voronoï d'une cible : détection de la proximité du doigt au-dessus de la surface d'affichage.

9.2.1.2 Techniques de pointage en RA

Nous avons proposé des techniques visant à améliorer le pointage indirect, effectué avec un curseur sur l'écran. Mais le calcul dynamique du partitionnement de Voronoï pour agrandir les cibles serait également bénéfique pour un pointage tactile direct : l'utilisateur serait en mesure de sélectionner une cible en cliquant n'importe où dans la cellule de Voronoï, sans avoir à toucher directement sur le POI (voir Figure 9.3). Pour éviter l'affichage permanent à l'écran du partitionnement complet de Voronoï, une piste de recherche serait d'utiliser la détection de la proximité des doigts au-dessus de la surface d'affichage pour mettre en évidence soit la cellule de Voronoï la plus proche du doigt, soit la cible de la cellule de Voronoï la plus proche du doigt (voir Figure 7.6). Cette anticipation permettrait de (1) minimiser l'intrusion visuelle à l'écran (2) éviter l'occultation des cibles sous le doigt [88] tout en facilitant la sélection de cibles en allouant à chaque cible une zone active plus grande (voir Figure 9.3).

9.2.1.3 Combinaison des techniques de visualisation hors-champ et de pointage en RA

Un autre axe de recherche consisterait à étudier la sélection d'une cible en combinant les techniques de visualisation hors-champ avec les techniques d'assistance au pointage des cibles numériques. Pour cela nous considérons deux pistes de recherche : (1) Étudier l'utilisation successive des techniques de visualisation hors-champ (pour ramener une cible dans le champ de vision de la caméra), puis des techniques d'assistance au pointage (pour sélectionner la cible). L'objectif est ici de mesurer le bénéfice apporté par la combinaison des deux types de techniques sur une tâche de pointage d'une cible hors-champ. (2) Étudier l'utilisation simultanée des techniques de visualisation hors-champ et d'assistance au pointage pour sélectionner une cible hors-champ non visible sans la ramener dans le champ visuel de la caméra. Dans ce cas, le partitionnement de Voronoï serait appliqué aux cibles situées dans l'espace hors-champ pour optimiser leur sélection.

9.2.2 Perspectives à long terme : ouverture vers d'autres travaux

L'interaction en RA sur supports mobiles est centrale pour l'expansion de la RA et permettre un usage effectif de cette technologie dans de nombreux domaines applicatifs. Nous avons principalement étudié la recherche et sélection d'une cible numérique. Il s'agit là d'une tâche élémentaire et universelle dans toute interaction. Des tâches de plus haut niveau d'abstraction compléteraient ces travaux comme la création de cibles numériques en contexte, la manipulation de ces cibles numériques ou encore la gestion d'une procédure à suivre lors d'une maintenance.

Une autre piste de recherche qui nous semble particulièrement prometteuse est de considérer un casque de RA plutôt qu'un dispositif mobile tenu dans les mains. En effet, dans les domaines applicatifs comme la maintenance industrielle mais aussi la chirurgie augmentée,

libérer les mains des utilisateurs se justifient aisément. Par exemple dans notre domaine applicatif cible, le port d'un casque de RA n'est pas exclu (comme l'utilisation du casque Hololens pour la maintenance [69]). De plus, nous constatons actuellement dans le domaine de l'Interaction Homme-Machine un essor des travaux de recherche sur l'interaction avec un casque de RA, sans doute encouragé par la disponibilité du casque Hololens et plus récemment du casque Hololens 2. Le casque apporte une liberté physique car les mains ne sont plus nécessaires pour maintenir le support, contrairement aux appareils mobiles. Ce confort est au détriment du champ de vision restreint. Actuellement, les casques pour la RA offrent un champ de vision plus limité que celui offert par la caméra des téléphones ou tablettes mobiles : par exemple, le casque Hololens 2 [62] offre un champ visuel augmenté de 43 degrés horizontalement, tandis que la caméra arrière de l'iphone 6s (2015) offre un champ visuel horizontal de 58 degrés [4]. Dans les deux cas, ces champs de vision sont réduits par rapport aux capacités de la vision humaine. Un premier pas dans cet axe de recherche serait de traiter les questions de recherche de nos travaux sur un casque de RA. En particulier, nous avons proposé des techniques de visualisation hors-champ pour étendre la connaissance de l'utilisateur sur l'environnement hors-champ. La question est alors de déterminer si les bénéfices apportés par ces techniques sur supports mobiles restent valides dans les casques de RA. Des travaux existants ont par exemple cherché à exploiter les techniques de visualisation hors-champ Halo et Wedge dans un casque de RA [46, 52]. Il serait alors pertinent de comparer les résultats obtenus sur un support mobile (téléphone ou tablette) avec ceux que nous obtiendrions si les techniques proposées étaient affichées dans un casque.

Au-delà de la comparaison des résultats afin de comprendre si les techniques proposées sont utilisables dans un casque de RA, un autre axe concerne l'interaction : si une technique basée sur le halo est affichée dans un casque de RA, les indices vers les cibles peuvent aussi servir à indiquer le geste mid-air que l'utilisateur doit faire pour ramener momentanément la cible dans le champ visuel de l'utilisateur. Cette problématique peut s'étudier aussi sur un support mobile de type tablette mais prend tout son sens dans un casque puisque l'utilisateur n'a pas à tenir le support mobile. Au lieu de perdre son focus visuel de travail courant, les indices visuels indiqueront le geste ou l'ampleur du geste à effectuer pour jeter un coup d'œil par exemple sur la valeur d'un capteur qui n'est pas devant l'utilisateur [117, 8]. Ceci permettrait à la fois de guider vers la cible mais aussi d'indiquer le geste mid-air à réaliser pour ramener temporairement la cible dans le champ visuel. Ceci permettrait de faire une sélection sans regarder "eye-free" tout en évitant que l'utilisateur apprenne un langage gestuel.

A N N E X E S

A

System Usability Scale

Le System Usability Scale [24] est un questionnaire comportant 10 questions pour évaluer l'utilisabilité des différentes techniques comparées dans l'étude expérimentale 1 :

	Strongly disagree					Strongly agree
1. I think that I would like to use this system frequently	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
2. I found the system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
3. I thought the system was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
4. I think that I would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
5. I found the various functions in this system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
7. I would imagine that most people would learn to use this system very quickly	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
8. I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
9. I felt very confident using the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	1	2	3	4	5	

B

RAW TLX (Task Load Index)

Le Raw Task Load Index (RTLX) [61] est un questionnaire qui permet d'évaluer la charge de travail ressentie par les participants lors de l'utilisation des différentes techniques comparées dans les études expérimentales 2 et 3 :

Mental Demand	How mentally demanding was the task?
Very Low	Very High
Physical Demand	How physically demanding was the task?
Very Low	Very High
Temporal Demand	How hurried or rushed was the pace of the task?
Very Low	Very High
Performance	How successful were you in accomplishing what you were asked to do?
Perfect	Failure
Effort	How hard did you have to work to accomplish your level of performance?
Very Low	Very High
Frustration	How insecure, discouraged, irritated, stressed, and annoyed were you?
Very Low	Very High

C

Questionnaire comparatif des études expérimentales 1, 2 et 3

Nous présentons ici le questionnaire final comparatif des techniques comparées dans les études expérimentales 1, 2 et 3. Le questionnaire comprenait les questions ci-dessous pour chaque technique :

- J'ai trouvé la technique fatigantes mentalement (Note de 1 à 5)
- J'ai trouvé la technique fatigantes physiquement (Note de 1 à 5)
- J'ai trouvé la techniques facile à apprendre (Note de 1 à 5)
- Note générale de la technique (Note de 1 à 5)

A titre d'exemple, nous détaillons le questionnaire et les réponses fournies par un participant de l'étude expérimentale 3. Les 4 techniques de pointage comparées étaient : Baseline (B), Voronoï (V), Voronoï Jumping Physical (VJP) et Voronoï Jumping Relative (VJR). Ces 4 techniques ont été renommées dans le questionnaire pour plus de facilité pour les participants tel que B = "Align", V = "Distant Opening", VJP = "Fixed cursor jump" et VJR = "Mobile cursor jump".

I found the technique mentally tiring

Align *

	1	2	3	4	5	
Strongly disagree	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

Distant opening *

	1	2	3	4	5	
Strongly disagree	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

Fixed cursor jump *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

Mobile cursor jump *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

I found the technique physically tiring

Align *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Strongly agree

Distant opening *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

Fixed cursor jump *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

Mobile cursor jump *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

I found the technique easy to learn

Align *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Strongly agree

Distant opening *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Strongly agree

Fixed cursor jump *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

Mobile cursor jump *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

Overall note

Align *

	1	2	3	4	5	
Very bad	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very good

Distant opening *

	1	2	3	4	5	
Very bad	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very good

Fixed cursor jump *

	1	2	3	4	5	
Very bad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Very good

Mobile cursor jump *

	1	2	3	4	5	
Very bad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Very good

Bibliographie

- [1] Christopher AHLBERG et Ben SHNEIDERMAN. “Visual information seeking : Tight coupling of dynamic query filters with starfield displays”. In : *The Craft of Information Visualization*. Elsevier, 2003, p. 7-13.
- [2] David AHLSTRÖM. “Modeling and Improving Selection in Cascading Pull-down Menus Using Fitts’ Law, the Steering Law and Force Fields”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '05. Portland, Oregon, USA : Association for Computing Machinery, 2005, p. 61-70. ISBN : 1581139985. DOI : 10.1145/1054972.1054982. URL : <https://doi.org/10.1145/1054972.1054982>.
- [3] David AHLSTRÖM, Rainer ALEXANDROWICZ et Martin HITZ. “Improving Menu Interaction : A Comparison of Standard, Force Enhanced and Jumping Menus”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montréal, Québec, Canada : Association for Computing Machinery, 2006, p. 1067-1076. ISBN : 1595933727. DOI : 10.1145/1124772.1124932. URL : <https://doi.org/10.1145/1124772.1124932>.
- [4] *Apple Cameras*. <https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/Cameras/Cameras.html>. Accessed : 2020-04-07.
- [5] R. AZUMA et al. “Recent advances in augmented reality”. In : *IEEE Computer Graphics and Applications* 21.6 (nov. 2001), p. 34-47. ISSN : 1558-1756. DOI : 10.1109/38.963459.
- [6] Ronald T. AZUMA. “A Survey of Augmented Reality”. In : *Presence : Teleoperators and Virtual Environments* 6.4 (1997), p. 355-385. DOI : 10.1162/pres.1997.6.4.355.
- [7] Huidong BAI, Gun A. LEE et Mark BILLINGHURST. “Freeze View Touch and Finger Gesture Based Interaction Methods for Handheld Augmented Reality Interfaces”. In : *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*. IVCNZ '12. Dunedin, New Zealand : Association for Computing Machinery, 2012, p. 126-131. ISBN : 9781450314732. DOI : 10.1145/2425836.2425864. URL : <https://doi.org/10.1145/2425836.2425864>.

- [8] Charles BAILLY, François LEITNER et Laurence NIGAY. “Bring2Me : Bringing Virtual Widgets Back to the User’s Field of View in Mixed Reality”. In : *Proceedings of the 2020 AVI Conference on Advanced Visual Interfaces*. AVI ’20. New York, NY, USA : Association for Computing Machinery, 2020.
- [9] Ravin BALAKRISHNAN. “”Beating” Fitts’ Law : Virtual Enhancements for Pointing Facilitation”. In : *International Journal of Human-Computer Studies* 61.6 (déc. 2004), p. 857-874. ISSN : 1071-5819. DOI : 10.1016/j.ijhcs.2004.09.002. URL : <https://doi.org/10.1016/j.ijhcs.2004.09.002>.
- [10] Marc BALOUP, Thomas PIETRZAK et Géry CASIEZ. “RayCursor : A 3D Pointing Facilitation Technique Based on Raycasting”. In : *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. Glasgow, Scotland Uk : Association for Computing Machinery, 2019. ISBN : 9781450359702. DOI : 10.1145/3290605.3300331. URL : <https://doi.org/10.1145/3290605.3300331>.
- [11] Olivier BAU et Ivan POUPYREV. “REVEL : Tactile Feedback Technology for Augmented Reality”. In : *ACM Trans. Graph.* 31.4 (juil. 2012). ISSN : 0730-0301. DOI : 10.1145/2185520.2185585. URL : <https://doi.org/10.1145/2185520.2185585>.
- [12] Patrick BAUDISCH et al. “Drag-and-pop and drag-and-pick : Techniques for accessing remote screen content on touch-and pen-operated systems”. In : *Proceedings of INTERACT*. T. 3. 2003, p. 57-64.
- [13] Patrick BAUDISCH et Ruth ROSENHOLTZ. “Halo : a technique for visualizing off-screen objects”. In : *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2003, p. 481-488.
- [14] Patrick BAUDISCH et al. “Keeping Things in Context : A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’02. Minneapolis, Minnesota, USA : ACM, 2002, p. 259-266. ISBN : 1-58113-453-3. DOI : 10.1145/503376.503423. URL : <http://doi.acm.org/10.1145/503376.503423>.
- [15] Patrick BAUDISCH et al. “Starburst : A Target Expansion Algorithm for Non-Uniform Target Distributions”. In : *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI ’08. Napoli, Italy : Association for Computing Machinery, 2008, p. 129-137. ISBN : 9781605581415. DOI : 10.1145/1385569.1385592. URL : <https://doi.org/10.1145/1385569.1385592>.
- [16] David V BEARD et JOHN Q WALKER II. “Navigational techniques to improve the display of large two-dimensional spaces”. In : *Behaviour & Information Technology* 9.6 (1990), p. 451-466.
- [17] Hrvoje BENKO, Andrew D. WILSON et Patrick BAUDISCH. “Precise Selection Techniques for Multi-Touch Screens”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’06. Montréal, Québec, Canada : Association for Computing Machinery, 2006, p. 1263-1272. ISBN : 1595933727. DOI : 10.1145/1124772.1124963. URL : <https://doi.org/10.1145/1124772.1124963>.

- [18] Joanna BERGSTROM-LEHTOVIRTA et Antti OULASVIRTA. “Modeling the Functional Area of the Thumb on Mobile Touchscreen Surfaces”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada : Association for Computing Machinery, 2014, p. 1991-2000. ISBN : 9781450324731. DOI : 10.1145/2556288.2557354. URL : <https://doi.org/10.1145/2556288.2557354>.
- [19] M. BILLINGHURST et al. “A wearable spatial conferencing space”. In : *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215)*. Oct. 1998, p. 76-83. DOI : 10.1109/ISWC.1998.729532.
- [20] Mark BILLINGHURST, Raphael GRASSET et Julian LOOSER. “Designing augmented reality interfaces”. In : *ACM SIGGRAPH Computer Graphics* 39 (fév. 2005), p. 17-22. DOI : 10.1145/1057792.1057803.
- [21] Renaud BLANCH et Michaël ORTEGA. “Rake Cursor : Improving Pointing Performance with Concurrent Input Channels”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA : Association for Computing Machinery, 2009, p. 1415-1418. ISBN : 9781605582467. DOI : 10.1145/1518701.1518914. URL : <https://doi.org/10.1145/1518701.1518914>.
- [22] Ó. BLANCO-NOVOA et al. “A Practical Evaluation of Commercial Industrial Augmented Reality Systems in an Industry 4.0 Shipyard”. In : *IEEE Access* 6 (2018), p. 8201-8218. ISSN : 2169-3536. DOI : 10.1109/ACCESS.2018.2802699.
- [23] Sebastian BORING et al. “Touch Projector : Mobile Interaction through Video”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA : Association for Computing Machinery, 2010, p. 2287-2296. ISBN : 9781605589299. DOI : 10.1145/1753326.1753671. URL : <https://doi.org/10.1145/1753326.1753671>.
- [24] John BROOKE et al. “SUS-A quick and dirty usability scale”. In : *Usability evaluation in industry* 189.194 (1996), p. 4-7.
- [25] Stefano BURIGAT et Luca CHITTARO. “On the effectiveness of Overview+ Detail visualization on mobile devices”. In : *Personal and ubiquitous computing* 17.2 (2013), p. 371-385.
- [26] Stefano BURIGAT et Luca CHITTARO. “Visualizing references to off-screen content on mobile devices : A comparison of Arrows, Wedge, and Overview+ Detail”. In : *Interacting with Computers* 23.2 (2011), p. 156-166.
- [27] Stefano BURIGAT, Luca CHITTARO et Silvia GABRIELLI. “Visualizing locations of off-screen objects on mobile devices : a comparative evaluation of three approaches”. In : *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. ACM. 2006, p. 239-246.

- [28] Stefano BURIGAT, Luca CHITTARO et Andrea VIANELLO. “Dynamic visualization of large numbers of off-screen objects on mobile devices : an experimental comparison of wedge and overview+ detail”. In : *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*. ACM. 2012, p. 93-102.
- [29] David CARR, Catherine PLAISANT et Hiroaki HASEGAWA. “Designing a real-time telepathology workstation to mitigate communication delays”. In : *Interacting with Computers* 11.1 (1998), p. 33-52.
- [30] J. CASHION, C. WINGRAVE et J. J. LAVIOLA JR. “Dense and Dynamic 3D Selection for Game-Based Virtual Environments”. In : *IEEE Transactions on Visualization and Computer Graphics* 18.4 (avr. 2012), p. 634-642. ISSN : 2160-9306. DOI : 10.1109/TVCG.2012.40.
- [31] Géry CASIEZ et al. “RubberEdge : Reducing Clutching by Combining Position and Rate Control with Elastic Feedback”. In : *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*. UIST '07. Newport, Rhode Island, USA : Association for Computing Machinery, 2007, p. 129-138. ISBN : 9781595936790. DOI : 10.1145/1294211.1294234. URL : <https://doi.org/10.1145/1294211.1294234>.
- [32] T. P. CAUDELL et D. W. MIZELL. “Augmented reality : an application of heads-up display technology to manual manufacturing processes”. In : *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*. T. ii. Jan. 1992, 659-669 vol.2. DOI : 10.1109/HICSS.1992.183317.
- [33] Youli CHANG et al. “Understanding Users’ Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture”. In : *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea : Association for Computing Machinery, 2015, p. 1499-1508. ISBN : 9781450331456. DOI : 10.1145/2702123.2702425. URL : <https://doi.org/10.1145/2702123.2702425>.
- [34] Luca CHITTARO et Stefano BURIGAT. “3D location-pointing as a navigation aid in Virtual Environments”. In : *Proceedings of the working conference on Advanced visual interfaces*. ACM. 2004, p. 267-274.
- [35] Andy COCKBURN, Amy KARLSON et Benjamin B. BEDERSON. “A Review of Overview+Detail, Zooming, and Focus+Context Interfaces”. In : *ACM Comput. Surv.* 41.1 (jan. 2009), 2 :1-2 :31. ISSN : 0360-0300. DOI : 10.1145/1456650.1456652. URL : <http://doi.acm.org/10.1145/1456650.1456652>.
- [36] ERFW CROSSMAN et PJ GOODEVE. “Feedback control of hand-movement and Fitts’ law”. In : *The Quarterly Journal of Experimental Psychology Section A* 35.2 (1983), p. 251-278. DOI : 10.1080/14640748308402133.
- [37] *Driving the Digital Transformation with Augmented Reality*. <https://blog.se.com/machine-and-process-management/2018/01/16/driving-digital-transformation-augmented-reality/>. Accessed : 2020-03-19.

- [38] Mike EISSELE, Oliver SIEMONEIT et Thomas ERTL. “Transition of Mixed, Virtual, and Augmented Reality in Smart Production Environments - An Interdisciplinary View”. In : juil. 2006, p. 1-6. DOI : 10.1109/RAMECH.2006.252671.
- [39] Barrett ENS et al. “Characterizing User Performance with Assisted Direct Off-Screen Pointing”. In : *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. MobileHCI '11. Stockholm, Sweden : Association for Computing Machinery, 2011, p. 485-494. ISBN : 9781450305419. DOI : 10.1145/2037373.2037445. URL : <https://doi.org/10.1145/2037373.2037445>.
- [40] Steven FEINER, Blair MACINTYRE et Dorée SELIGMANN. “Knowledge-Based Augmented Reality”. In : *Commun. ACM* 36.7 (juil. 1993), p. 53-62. ISSN : 0001-0782. DOI : 10.1145/159544.159587. URL : <https://doi.org/10.1145/159544.159587>.
- [41] Paul M FITTS. “The information capacity of the human motor system in controlling the amplitude of movement.” In : *Journal of experimental psychology* 47.6 (1954), p. 381.
- [42] S FORTUNE. “A Sweepline Algorithm for Voronoi Diagrams”. In : *Proceedings of the Second Annual Symposium on Computational Geometry*. SCG '86. Yorktown Heights, New York, USA : Association for Computing Machinery, 1986, p. 313-322. ISBN : 0897911946. DOI : 10.1145/10515.10549. URL : <https://doi.org/10.1145/10515.10549>.
- [43] Tiago GONÇALVES et al. “Halodot : Visualization of the relevance of off-screen objects”. In : *Proceedings of SIACG* (2011), p. 117-120.
- [44] Tovi GROSSMAN et Ravin BALAKRISHNAN. “The Bubble Cursor : Enhancing Target Acquisition by Dynamic Resizing of the Cursor’s Activation Area”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '05. Portland, Oregon, USA : Association for Computing Machinery, 2005, p. 281-290. ISBN : 1581139985. DOI : 10.1145/1054972.1055012. URL : <https://doi.org/10.1145/1054972.1055012>.
- [45] Uwe GRUENEFELD, Dana HSIAO et Wilko HEUTEN. “EyeSeeX : Visualization of Out-of-View Objects on Small Field-of-View Augmented and Virtual Reality Devices”. In : *Proceedings of the 7th ACM International Symposium on Pervasive Displays*. ACM. 2018, p. 26.
- [46] Uwe GRUENEFELD et al. “Beyond Halo and Wedge : visualizing out-of-view objects on head-mounted virtual and augmented reality devices”. In : *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. 2018, p. 40.
- [47] Uwe GRUENEFELD et al. “EyeSee : beyond reality with Microsoft HoloLens”. In : *Proceedings of the 5th Symposium on Spatial User Interaction*. ACM. 2017, p. 148-148.

- [48] Uwe GRUENEFELD et al. “EyeSee360 : designing a visualization technique for out-of-view objects in head-mounted augmented reality”. In : *Proceedings of the 5th Symposium on Spatial User Interaction*. ACM. 2017, p. 109-118.
- [49] Uwe GRUENEFELD et al. “FlyingARrow : Pointing Towards Out-of-View Objects on Augmented Reality Devices”. In : *Proceedings of the 7th ACM International Symposium on Pervasive Displays*. ACM. 2018, p. 20.
- [50] Uwe GRUENEFELD et al. “MonocularAR : a radial light display to point towards out-of-view objects on augmented reality devices”. In : *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. ACM. 2018, p. 16-22.
- [51] Uwe GRUENEFELD et al. “PeriMR : a prototyping tool for head-mounted peripheral light displays in mixed reality”. In : *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. 2017, p. 51.
- [52] Uwe GRUENEFELD et al. “Visualizing out-of-view objects in head-mounted augmented reality”. In : *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. 2017, p. 81.
- [53] Yves GUIARD, Renaud BLANCH et Michel BEAUDOUIN-LAFON. “Object Pointing : A Complement to Bitmap Pointing in GUIs”. In : *Proceedings of Graphics Interface 2004*. GI '04. London, Ontario, Canada : Canadian Human-Computer Communications Society, 2004, p. 9-16. ISBN : 1568812272.
- [54] Maxime GUILLON. “Expansion de cibles pour le pointage et sélection : application à l’interaction à distance en chirurgie augmentée”. Mém. de mast. Université Grenoble Alpes, 2017.
- [55] Maxime GUILLON, François LEITNER et Laurence NIGAY. “Investigating Visual Feedforward for Target Expansion Techniques”. In : *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea : Association for Computing Machinery, 2015, p. 2777-2786. ISBN : 9781450331456. DOI : 10.1145/2702123.2702375. URL : <https://doi.org/10.1145/2702123.2702375>.
- [56] Maxime GUILLON, François LEITNER et Laurence NIGAY. “Target Expansion Lens : It is Not the More Visual Feedback the Better!” In : *Proceedings of the International Working Conference on Advanced Visual Interfaces*. AVI '16. Bari, Italy : Association for Computing Machinery, 2016, p. 52-59. ISBN : 9781450341318. DOI : 10.1145/2909132.2909265. URL : <https://doi.org/10.1145/2909132.2909265>.
- [57] Tyler J. GUNN et al. “An Evaluation of One-Handed Techniques for Multiple-Target Selection”. In : *CHI '09 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '09. Boston, MA, USA : Association for Computing Machinery, 2009, p. 4189-4194. ISBN : 9781605582474. DOI : 10.1145/1520340.1520638. URL : <https://doi.org/10.1145/1520340.1520638>.

- [58] Sean G GUSTAFSON et Pourang P IRANI. “Comparing visualizations for tracking off-screen moving targets”. In : *CHI'07 Extended Abstracts on Human Factors in Computing Systems*. ACM. 2007, p. 2399-2404.
- [59] Sean GUSTAFSON et al. “Wedge : Clutter-free Visualization of Off-screen Locations”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy : ACM, 2008, p. 787-796. ISBN : 978-1-60558-011-1. DOI : 10.1145/1357054.1357179. URL : <http://doi.acm.org/10.1145/1357054.1357179>.
- [60] Sinem GUVEN, Steven FEINER et Ohan ODA. “Mobile Augmented Reality Interaction Techniques for Authoring Situated Media On-Site”. In : *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*. ISMAR '06. USA : IEEE Computer Society, 2006, p. 235-236. ISBN : 1424406501. DOI : 10.1109/ISMAR.2006.297821. URL : <https://doi.org/10.1109/ISMAR.2006.297821>.
- [61] Sandra G HART et Lowell E STAVELAND. “Development of NASA-TLX (Task Load Index) : Results of empirical and theoretical research”. In : *Advances in psychology*. T. 52. Elsevier, 1988, p. 139-183.
- [62] *Hololens 2*. https://www.microsoft.com/en-us/hololens?icid=SSM_AS_Promo_Devices_HoloLens2. Accessed : 2020-04-07.
- [63] Kasper HORNBAEK, Benjamin B. BEDERSON et Catherine PLAISANT. “Navigation Patterns and Usability of Zoomable User Interfaces with and Without an Overview”. In : *ACM Trans. Comput.-Hum. Interact.* 9.4 (déc. 2002), p. 362-389. ISSN : 1073-0516. DOI : 10.1145/586081.586086. URL : <http://doi.acm.org/10.1145/586081.586086>.
- [64] Zahid HOSSAIN et al. “EdgeSplit : Facilitating the Selection of off-Screen Objects”. In : *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '12. San Francisco, California, USA : Association for Computing Machinery, 2012, p. 79-82. ISBN : 9781450311052. DOI : 10.1145/2371574.2371588. URL : <https://doi.org/10.1145/2371574.2371588>.
- [65] Hyungeun JO et al. “Aroundplot : Focus+ context interface for off-screen objects in 3D environments”. In : *Computers & Graphics* 35.4 (2011), p. 841-853.
- [66] Amy K. KARLSON et Benjamin B. BEDERSON. “One-Handed Touchscreen Input for Legacy Applications”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy : Association for Computing Machinery, 2008, p. 1399-1408. ISBN : 9781605580111. DOI : 10.1145/1357054.1357274. URL : <https://doi.org/10.1145/1357054.1357274>.

- [67] Masatomo KOBAYASHI et Takeo IGARASHI. “Considering the Direction of Cursor Movement for Efficient Traversal of Cascading Menus”. In : *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. UIST '03. Vancouver, Canada : Association for Computing Machinery, 2003, p. 91-94. ISBN : 1581136366. DOI : 10.1145/964696.964706. URL : <https://doi.org/10.1145/964696.964706>.
- [68] Masatomo KOBAYASHI et Takeo IGARASHI. “Ninja Cursors : Using Multiple Cursors to Assist Target Acquisition on Large Screens”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy : Association for Computing Machinery, 2008, p. 949-958. ISBN : 9781605580111. DOI : 10.1145/1357054.1357201. URL : <https://doi.org/10.1145/1357054.1357201>.
- [69] *La réalité augmentée au service de la maintenance industrielle*. <https://www.ictjournal.ch/news/2018-05-28/la-realite-augmentee-au-service-de-la-maintenance-industrielle>. Accessed : 2020-04-06.
- [70] Gun A. LEE et al. “Freeze-Set-Go Interaction Method for Handheld Mobile Augmented Reality Environments”. In : *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*. VRST '09. Kyoto, Japan : Association for Computing Machinery, 2009, p. 143-146. ISBN : 9781605588698. DOI : 10.1145/1643928.1643961. URL : <https://doi.org/10.1145/1643928.1643961>.
- [71] Wing Ho Andy LI et Hongbo FU. “BezelCursor : Bezel-Initiated Cursor for One-Handed Target Acquisition on Mobile Touch Screens”. In : *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications*. SA '13. Hong Kong, Hong Kong : Association for Computing Machinery, 2013. ISBN : 9781450326339. DOI : 10.1145/2543651.2543680. URL : <https://doi.org/10.1145/2543651.2543680>.
- [72] Rensis LIKERT. “A technique for the measurement of attitudes.” In : *Archives of psychology* (1932).
- [73] J MACKINLAY et al. “City lights : contextual views in minimal space”. In : *CHI'03 Extended Abstracts*. 2003, p. 838-839.
- [74] Michael J. MCGUFFIN et Ravin BALAKRISHNAN. “Fitts’ Law and Expanding Targets : Experimental Studies and Designs for User Interfaces”. In : *ACM Trans. Comput.-Hum. Interact.* 12.4 (déc. 2005), p. 388-422. ISSN : 1073-0516. DOI : 10.1145/1121112.1121115. URL : <https://doi.org/10.1145/1121112.1121115>.
- [75] David E MEYER et al. “Optimality in human motor performance : ideal control of rapid aimed movements.” In : *Psychological review* 95.3 (1988), p. 340. DOI : 10.1037/0033-295x.95.3.340.
- [76] Paul MILGRAM et Fumio KISHINO. “A Taxonomy of Mixed Reality Visual Displays”. In : *IEICE Trans. Information Systems* vol. E77-D, no. 12 (déc. 1994), p. 1321-1329.

- [77] Paul MILGRAM et al. “Augmented reality : A class of displays on the reality-virtuality continuum”. In : *Telem manipulator and Telepresence Technologies* 2351 (jan. 1994). DOI : 10.1117/12.197321.
- [78] Annette MOSSEL, Benjamin VENDITTI et Hannes KAUFMANN. “DrillSample : Precise Selection in Dense Handheld Augmented Reality Environments”. In : *Proceedings of the Virtual Reality International Conference : Laval Virtual. VRIC '13*. Laval, France : Association for Computing Machinery, 2013. ISBN : 9781450318754. DOI : 10.1145/2466816.2466827. URL : <https://doi.org/10.1145/2466816.2466827>.
- [79] Martez E. MOTT et Jacob O. WOBROCK. “Beating the Bubble : Using Kinematic Triggering in the Bubble Lens for Acquiring Small, Dense Targets”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '14*. Toronto, Ontario, Canada : Association for Computing Machinery, 2014, p. 733-742. ISBN : 9781450324731. DOI : 10.1145/2556288.2557410. URL : <https://doi.org/10.1145/2556288.2557410>.
- [80] *Mounting and Unmounting TeSys GV Manual Starter to DIN Rails — Schneider Electric Support*. <https://www.se.com/us/en/faqs/FA109994/>. Accessed : 2020-05-04.
- [81] Heiko MÜLLER et al. “Sparkle : An Ambient Light Display for Dynamic Off-screen Points of Interest”. In : *Proceedings of the 8th Nordic Conference on Human-Computer Interaction : Fun, Fast, Foundational. NordiCHI '14*. Helsinki, Finland : ACM, 2014, p. 51-60. ISBN : 978-1-4503-2542-4. DOI : 10.1145/2639189.2639205. URL : <http://doi.acm.org/10.1145/2639189.2639205>.
- [82] Alessandro MULLONI, Andreas DÜNSER et Dieter SCHMALSTIEG. “Zooming Interfaces for Augmented Reality Browsers”. In : *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services. MobileHCI '10*. Lisbon, Portugal : ACM, 2010, p. 161-170. ISBN : 978-1-60558-835-3. DOI : 10.1145/1851600.1851629. URL : <http://doi.acm.org/10.1145/1851600.1851629>.
- [83] Fabio NARDUCCI. “Mixed Reality System for Industrial Environment : an Evaluation Study”. In : *CAAI Transactions on Intelligence Technology* 2 (mar. 2019). DOI : 10.1049/trit.2017.0017.
- [84] N. NAVAB. “Developing killer apps for industrial augmented reality”. In : *IEEE Computer Graphics and Applications* 24.3 (mai 2004), p. 16-20. ISSN : 1558-1756. DOI : 10.1109/MCG.2004.1297006.
- [85] K. A. NGUYEN et Z. LUO. “On assessing the positioning accuracy of Google Tango in challenging indoor environments”. In : *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Sept. 2017, p. 1-8. DOI : 10.1109/IPIN.2017.8115933.

- [86] Haruo NOMA, Tsutomu MIYASATO et Fumio KISHINO. “A Palmtop Display for Dextrous Manipulation with Haptic Sensation”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '96. Vancouver, British Columbia, Canada : Association for Computing Machinery, 1996, p. 126-133. ISBN : 0897917774. DOI : 10.1145/238386.238454. URL : <https://doi.org/10.1145/238386.238454>.
- [87] C PLAISANT, D CARR et Ben SHNEIDERMAN. “Image Browser : Taxonomy Guidelines and Informal Specification”. In : *IEEE Software* (1995).
- [88] R. L. POTTER, L. J. WELDON et B. SHNEIDERMAN. “Improving the Accuracy of Touch Screens : An Experimental Evaluation of Three Strategies”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '88. Washington, D.C., USA : Association for Computing Machinery, 1988, p. 27-32. ISBN : 0201142376. DOI : 10.1145/57167.57171. URL : <https://doi.org/10.1145/57167.57171>.
- [89] R. L. POTTER, L. J. WELDON et B. SHNEIDERMAN. “Improving the Accuracy of Touch Screens : An Experimental Evaluation of Three Strategies”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '88. Washington, D.C., USA : Association for Computing Machinery, 1988, p. 27-32. ISBN : 0201142376. DOI : 10.1145/57167.57171. URL : <https://doi.org/10.1145/57167.57171>.
- [90] R. L. POTTER, L. J. WELDON et B. SHNEIDERMAN. “Improving the Accuracy of Touch Screens : An Experimental Evaluation of Three Strategies”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '88. Washington, D.C., USA : Association for Computing Machinery, 1988, p. 27-32. ISBN : 0201142376. DOI : 10.1145/57167.57171. URL : <https://doi.org/10.1145/57167.57171>.
- [91] Iulian RADU, Blair MACINTYRE et Stella LOURENCO. “Comparing Children’s Crosshair and Finger Interactions in Handheld Augmented Reality : Relationships Between Usability and Child Development”. In : *Proceedings of the The 15th International Conference on Interaction Design and Children*. IDC '16. Manchester, United Kingdom : Association for Computing Machinery, 2016, p. 288-298. ISBN : 9781450343138. DOI : 10.1145/2930674.2930726. URL : <https://doi.org/10.1145/2930674.2930726>.
- [92] Rafael ROBERTO et al. “Evaluation of Motion Tracking and Depth Sensing Accuracy of the Tango Tablet”. In : sept. 2016, p. 231-234. DOI : 10.1109/ISMAR-Adjunct.2016.0082.
- [93] Michael ROHS et Antti OULASVIRTA. “Target Acquisition with Camera Phones When Used as Magic Lenses”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy : Association for Computing Machinery, 2008, p. 1409-1418. ISBN : 9781605580111. DOI : 10.1145/1357054.1357275. URL : <https://doi.org/10.1145/1357054.1357275>.

- [94] Michael ROHS et Antti OULASVIRTA. “Target Acquisition with Camera Phones When Used as Magic Lenses”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy : Association for Computing Machinery, 2008, p. 1409-1418. ISBN : 9781605580111. DOI : 10.1145/1357054.1357275. URL : <https://doi.org/10.1145/1357054.1357275>.
- [95] Michael ROHS, Antti OULASVIRTA et Tiia SUOMALAINEN. “Interaction with Magic Lenses : Real-World Validation of a Fitts’ Law Model”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada : Association for Computing Machinery, 2011, p. 2725-2728. ISBN : 9781450302289. DOI : 10.1145/1978942.1979343. URL : <https://doi.org/10.1145/1978942.1979343>.
- [96] Anne ROUDAUT, Stéphane HUOT et Eric LECOLINET. “TapTap and MagStick : Improving One-Handed Target Acquisition on Small Touch-Screens”. In : *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '08. Napoli, Italy : Association for Computing Machinery, 2008, p. 146-153. ISBN : 9781605581415. DOI : 10.1145/1385569.1385594. URL : <https://doi.org/10.1145/1385569.1385594>.
- [97] Daniel SEGOVIA et al. “Augmented Reality as a Tool for Production and Quality Monitoring”. In : *Procedia Computer Science* 75 (déc. 2015), p. 291-300. DOI : 10.1016/j.procs.2015.12.250.
- [98] Allison B SEKULER et Richard F MURRAY. “Amodal completion : a case study”. In : *From fragments to objects : Segmentation and grouping in vision* 130 (2001), p. 265.
- [99] Ben SHNEIDERMAN. *Designing the User Interface : Strategies for Effective Human-Computer Interaction*. 3rd. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1997. ISBN : 0201694972.
- [100] Caroline SHOURABOURA et Pavel BLEHER. “Placement of applications in computing clouds using Voronoi diagrams”. In : *J. Internet Services and Applications* 2 (déc. 2011), p. 229-241. DOI : 10.1007/s13174-011-0037-8.
- [101] R William SOUKOREFF et I Scott MACKENZIE. “Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts’ law research in HCI”. In : *International journal of human-computer studies* 61.6 (2004), p. 751-789.
- [102] Qingkun SU et al. “2D-Dragger : Unified Touch-Based Target Acquisition with Constant Effective Width”. In : *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '16. Florence, Italy : Association for Computing Machinery, 2016, p. 170-179. ISBN : 9781450344081. DOI : 10.1145/2935334.2935339. URL : <https://doi.org/10.1145/2935334.2935339>.

- [103] Xiaojun SU, Oscar Kin-Chung AU et Rynson W.H. LAU. “The Implicit Fan Cursor : A Velocity Dependent Area Cursor”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada : Association for Computing Machinery, 2014, p. 753-762. ISBN : 9781450324731. DOI : 10.1145/2556288.2557095. URL : <https://doi.org/10.1145/2556288.2557095>.
- [104] Can TELKENAROGLU et Tolga CAPIN. “Dual-Finger 3D Interaction Techniques for Mobile Devices”. In : *Personal Ubiquitous Comput.* 17.7 (oct. 2013), p. 1551-1572. ISSN : 1617-4909. DOI : 10.1007/s00779-012-0594-2. URL : <https://doi.org/10.1007/s00779-012-0594-2>.
- [105] Matthias TRAPP et al. “Strategies for visualizing points-of-interest of 3D virtual environments on mobile devices”. In : *Proceedings of the sixth international symposium on LBS & TeleCartography*. Citeseer. 2009.
- [106] Theophanis TSANDILAS et m. c. schraefel m. c. “Bubbling Menus : A Selective Mechanism for Accessing Hierarchical Drop-down Menus”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA : Association for Computing Machinery, 2007, p. 1195-1204. ISBN : 9781595935939. DOI : 10.1145/1240624.1240806. URL : <https://doi.org/10.1145/1240624.1240806>.
- [107] Thomas VINCENT, Laurence NIGAY et Takeshi KURATA. “Handheld Augmented Reality : Effect of Registration Jitter on Cursor-Based Pointing Techniques”. In : *Proceedings of the 25th Conference on l'Interaction Homme-Machine*. IHM '13. Talence, France : Association for Computing Machinery, 2013, p. 1-6. ISBN : 9781450324076. DOI : 10.1145/2534903.2534905. URL : <https://doi.org/10.1145/2534903.2534905>.
- [108] Thomas VINCENT, Laurence NIGAY et Takeshi KURATA. “Precise Pointing Techniques for Handheld Augmented Reality”. In : t. 8117. Sept. 2013. DOI : 10.1007/978-3-642-40483-2_9.
- [109] Daniel VOGEL et Patrick BAUDISCH. “Shift : A Technique for Operating Pen-Based Interfaces Using Touch”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA : Association for Computing Machinery, 2007, p. 657-666. ISBN : 9781595935939. DOI : 10.1145/1240624.1240727. URL : <https://doi.org/10.1145/1240624.1240727>.
- [110] Julie WAGNER, Stéphane HUOT et Wendy MACKAY. “BiTouch and BiPad : Designing Bimanual Interaction for Hand-Held Tablets”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA : Association for Computing Machinery, 2012, p. 2317-2326. ISBN : 9781450310154. DOI : 10.1145/2207676.2208391. URL : <https://doi.org/10.1145/2207676.2208391>.

- [111] Michelle Q WANG BALDONADO, Allison WOODRUFF et Allan KUCHINSKY. “Guidelines for using multiple views in information visualization”. In : *Proceedings of the working conference on Advanced visual interfaces*. ACM. 2000, p. 110-119.
- [112] Colin WARE. “Information Visualization : Perception for Design : Second Edition”. In : Morgan Kaufmann, avr. 2004.
- [113] C WINGRAVE et D BOWMAN. “Baseline factors for raycasting selection”. In : *Proceedings of HCI International*. Citeseer. 2005, p. 61-68.
- [114] Jacob O. WOBBROCK et al. “The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada : Association for Computing Machinery, 2011, p. 143-146. ISBN : 9781450302289. DOI : 10.1145/1978942.1978963. URL : <https://doi.org/10.1145/1978942.1978963>.
- [115] Jacob O. WOBBROCK et al. “The Angle Mouse : Target-Agnostic Dynamic Gain Adjustment Based on Angular Deviation”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA : Association for Computing Machinery, 2009, p. 1401-1410. ISBN : 9781605582467. DOI : 10.1145/1518701.1518912. URL : <https://doi.org/10.1145/1518701.1518912>.
- [116] Daisuke YAMAMOTO, Shotaro OZEKI et Naohisa TAKAHASHI. “Focus+Glue+Context : An Improved Fisheye Approach for Web Map Services”. In : *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '09. Seattle, Washington : ACM, 2009, p. 101-110. ISBN : 978-1-60558-649-6. DOI : 10.1145/1653771.1653788. URL : <http://doi.acm.org/10.1145/1653771.1653788>.
- [117] Yukang YAN et al. “Eyes-Free Target Acquisition in Interaction Space around the Body for Virtual Reality”. In : *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada : Association for Computing Machinery, 2018. ISBN : 9781450356206. DOI : 10.1145/3173574.3173616. URL : <https://doi.org/10.1145/3173574.3173616>.
- [118] Xing-Dong YANG et al. “TouchCuts and TouchZoom : Enhanced Target Selection for Touch Displays Using Finger Proximity Sensing”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada : Association for Computing Machinery, 2011, p. 2585-2594. ISBN : 9781450302289. DOI : 10.1145/1978942.1979319. URL : <https://doi.org/10.1145/1978942.1979319>.
- [119] Koji YATANI et al. “Escape : A Target Selection Technique Using Visually-Cued Gestures”. In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy : Association for Computing Machinery, 2008, p. 285-294. ISBN : 9781605580111. DOI : 10.1145/1357054.1357104. URL : <https://doi.org/10.1145/1357054.1357104>.

- [120] Chun YU et al. “The Satellite Cursor : Achieving MAGIC Pointing without Gaze Tracking Using Multiple Cursors”. In : *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*. UIST '10. New York, New York, USA : Association for Computing Machinery, 2010, p. 163-172. ISBN : 9781450302715. DOI : 10.1145/1866029.1866056. URL : <https://doi.org/10.1145/1866029.1866056>.