

Titre: Développement d'un système en réalité augmentée d'assistance aux opérateurs de machinerie lourde dans une cour à bois
Title: Development of a system for heavy machinery operators in a wood yard using augmented reality

Auteur: Goulven Kermarec
Author:

Date: 2019

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Kermarec, G. (2019). Développement d'un système en réalité augmentée d'assistance aux opérateurs de machinerie lourde dans une cour à bois [Master's thesis, Polytechnique Montréal]. PolyPublie. <https://publications.polymtl.ca/3873/>
Citation:

Document en libre accès dans PolyPublie

Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/3873/>
PolyPublie URL:

Directeurs de recherche: Benoît Ozell
Advisors:

Programme: Génie informatique
Program:

UNIVERSITÉ DE MONTRÉAL

DÉVELOPPEMENT D'UN SYSTÈME EN RÉALITÉ AUGMENTÉE D'ASSISTANCE
AUX OPÉRATEURS DE MACHINERIE LOURDE DANS UNE COUR À BOIS

GOULVEN KERMAREC

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
MAI 2019

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

DÉVELOPPEMENT D'UN SYSTÈME EN RÉALITÉ AUGMENTÉE D'ASSISTANCE
AUX OPÉRATEURS DE MACHINERIE LOURDE DANS UNE COUR À BOIS

présenté par : KERMAREC Goulven

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

Mme BELLAÏCHE Martine, Ph. D., présidente

M. OZELL Benoît, Ph. D., membre et directeur de recherche

M. DAGENAIS Michel, Ph. D., membre

DÉDICACE

*À Lara,
pour qui l'aventure ne fait que commencer.*

REMERCIEMENTS

Je tiens à remercier Benoît Ozell pour sa supervision enrichissante lors de cette maîtrise, et je remercie aussi Dany Naser Addin pour son aide et son travail acharné lors du développement du système d'inventaire.

Je remercie aussi FPInnovations ainsi que Francis Charette pour leur soutien et leurs conseils tout au long du projet. J'aimerais aussi dire un grand merci au Groupe Crête et son président Sébastien Crête qui nous a laissés visiter une de ses cours à bois et qui a accepté de répondre à nos nombreuses questions

Je remercie particulièrement Juliette Tibayrenc pour son soutien moral et logistique à travers la dure période d'écriture. Je remercie mes parents pour leur suivi assidu et pour les relectures pertinentes de mon mémoire.

Et enfin, merci à Alexandra Elbakyan de rendre la science plus ouverte et égalitaire.

RÉSUMÉ

De nos jours, la Réalité augmentée (RA) prend une place de plus en plus prépondérante dans l'industrie en fournissant de l'assistance contextuelle à un niveau jamais atteint précédemment. En particulier, dans le domaine de la foresterie, les cours à bois sont des zones de stockage de paquets de bois, où évoluent des engins de machinerie lourde réalisant la manutention. Dans notre étude, on s'attache à développer une application en RA capable d'assister les opérateurs des machines évoluant dans la cour.

On utilise pour cela le visiocasque HoloLens de Microsoft. Ce visiocasque possède un dispositif interne pour mesurer sa position basé sur une centrale inertielle et des caméras analysant l'environnement. Mais nos tests ont montré que ce dispositif était incapable de fournir une position fiable à l'intérieur d'un véhicule en mouvement. En conséquence nous avons conçu un système capable de mesurer la position et l'orientation d'un visiocasque à l'intérieur d'un véhicule évoluant en extérieur avec la précision requise pour obtenir une qualité d'expérience satisfaisante en RA. Nous n'avons pas pu mettre en pratique ce système car nous n'avions pas accès au matériel haute précision nécessaire.

Nous avons cependant développé un prototype applicatif fonctionnant en intérieur en utilisant les informations du système de positionnement embarqué de l'HoloLens afin de mettre en pratique et valider les solutions de design d'interface que nous avons conçues. En parallèle, nous avons aussi mené une réflexion pour déterminer comment instrumenter les chargeurs à fourche servant à la manutention dans la cour à bois afin qu'ils puissent s'interfacer avec le reste du système d'assistance aux opérateurs.

La manutention dans une cour à bois étant rarement la tâche d'un seul chargeur, nous avons ainsi conçu et mis en pratique une architecture réseau permettant d'interconnecter tous les chargeurs instrumentés de la cour avec un serveur d'inventaire qui était développé en parallèle avec notre étude. De plus, nous nous sommes assurés de la robustesse du système face à des connexions instables telles que rencontrées en utilisant des liens sans-fil à bord de véhicules en mouvement.

Nous avons donc conçu un système complet d'assistance en RA aux opérateurs de machinerie lourde dans une cour à bois, et nous en avons mis en pratique l'interface ainsi que l'ensemble de l'architecture réseautique.

ABSTRACT

Nowadays, augmented reality (AR) has taken a central role in industrial settings by providing contextual help at a level never reached before. For example, in the domain of forestry, timber yards are storage areas for wood stacks. In these yards, one can find heavy equipment tasked with handling material. In our study, we aim to develop an AR application that can assist the operators working in the timber yard.

For this purpose, we elected to use the HoloLens head-mounted display (HMD) sold by Microsoft. The HMD has an embedded positioning device that uses an inertial measurement unit (IMU) and environment sensing cameras. However, our results showed this device is not able to provide a reliable position when used inside a moving vehicle. Thus we designed an attitude measurement system able to locate and give the orientation of an HMD sitting inside a vehicle driving outdoors, with enough precision to provide a satisfying experience in AR. However, we were not able to fully implement this system due to lacking access the high performance equipment required.

Nonetheless, we developed an application prototype designed to work indoors using the embedded HoloLens positioning system. We did so in order to implement and validate the user interface we designed. We also lead a reflection in parallel, to determine a way to instrument the fork-loaders used for handling in the timber yard. That way, they would be able to interface with the rest of the operator assistance system.

Handling in a timber yard is rarely – if ever – a one-loader task. Thus we designed and implemented a network architecture allowing for connections between the instrumented loaders and an inventory server that was developed in parallel of our project. Moreover, we made sure the system was robust against unstable connections, as those can be common when using a wireless link in a moving vehicle.

In summary, we designed an entire system to assist heavy equipment operators in a timber yard, and we implemented the interface and user experience, as well as the whole network architecture.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	x
LISTE DES FIGURES	xi
LISTE DES SIGLES ET ABRÉVIATIONS	xii
CHAPITRE 1 INTRODUCTION	1
1.1 Contexte	1
1.1.1 Cour à bois	1
1.2 Plan du mémoire	4
CHAPITRE 2 REVUE DE LITTÉRATURE	5
2.1 État de l'art	5
2.2 Positionnement	6
2.2.1 Positionnement par satellite	6
2.2.2 Positionnement par stations de base	9
2.2.3 Positionnement par lidar	11
2.2.4 Le positionnement de l'HoloLens	11
2.3 Orientation	12
2.3.1 Centrale inertielle (IMU)	13
2.3.2 Antennes Global Navigation Satellite System (GNSS) multiples . . .	14
2.4 Fusion de capteurs	15
2.4.1 Filtre de Kalman	15
2.4.2 GNSS + Inertial Measurement Unit (IMU)	17
2.4.3 Capteurs et prédictions secondaires	18
2.4.4 Vision	18

2.4.5 Lidar + IMU	18
2.4.6 GNSS + IMU + lidar	19
2.5 Instrumentation d'un chargeur	19
2.5.1 Balance de fourche	19
2.5.2 Estimation de pose dans la cabine	20
2.6 Principes de design	23
2.6.1 Interface homme-machine (IHM)	23
2.6.2 Design visuel	24
2.6.3 Qualité de l'expérience	25
2.7 Objectifs de recherche	27
 CHAPITRE 3 MÉTHODOLOGIE	28
3.1 Conception	28
3.1.1 Cour à bois	28
3.1.2 Visiocasque	29
3.1.3 Technologies de positionnement	32
3.1.4 Interconnexion entre les acteurs	35
3.1.5 Chargeur et instrumentation	37
3.1.6 Interface Utilisateur	41
3.2 Expérimentation	45
3.2.1 Cartographie tridimensionnelle	45
3.2.2 Affichage en plein soleil	46
3.2.3 Suivi complet dans un véhicule	47
3.2.4 Suivi en rotation dans un véhicule	47
3.3 Conclusion	48
 CHAPITRE 4 RÉSULTATS	49
4.1 Vérification des capacités de l'HoloLens	49
4.1.1 Cartographie tridimensionnelle	49
4.1.2 Affichage en plein soleil	49
4.1.3 Suivi complet dans un véhicule	51
4.1.4 Suivi en rotation dans un véhicule	51
4.2 Prototype	52
4.2.1 Interface en Réalité augmentée (RA)	52
4.2.2 Communication réseau	62
4.2.3 Multi-utilisateur et système d'inventaire	62
4.2.4 Optimisation graphique	64

4.3 Conclusion	72
CHAPITRE 5 DISCUSSION	73
5.1 Vérification des capacités de l'HoloLens	73
5.1.1 Cartographie tridimensionnelle	73
5.1.2 Affichage en plein soleil	73
5.1.3 Suivi complet dans un véhicule	74
5.1.4 Suivi en rotation dans un véhicule	74
5.2 Réflexions sur l'application prototype développée et ses limitations	75
5.2.1 Modélisation du problème	75
5.2.2 Communication	76
5.2.3 Infographie	76
5.2.4 Visiocasque	77
5.2.5 Positionnement	77
5.2.6 Chargeurs	78
5.2.7 Interface utilisateur	78
CHAPITRE 6 CONCLUSION	80
6.1 Synthèse des travaux	80
6.2 Travaux futurs	80
6.2.1 Environnement alternatif	81
6.2.2 Contrôle à distance	82
RÉFÉRENCES	83

LISTE DES TABLEAUX

Tableau 4.1	Temps d'exécution des différentes étapes du dessin de texte avec et sans Champ de distance signé (SDF)	66
Tableau 4.2	Comparaison des temps d'exécution sur le Graphics Processing Unit (GPU) des méthodes de dessin instancée et naïve	68
Tableau 4.3	Comparaison des performances de différents nuanceurs de pixels	71

LISTE DES FIGURES

Figure 1.1	La cour à bois du Groupe Crête à Saint-Faustin-Lac-Carré. On voit de multiples rangées de paquets de bois non rabotés. Sur la droite de l'image on voit l'une des usines de rabotage. À gauche on distingue aussi trois des séchoirs.	1
Figure 3.1	Plusieurs rangées dans la cour à bois fini. On remarque que la rangée la plus haute compte sept étages. La photo est prise de l'arrière des rangées et on voit que la rangée la plus à gauche est quasiment vide. . .	30
Figure 3.2	Zone d'acceptabilité du couple erreur en translation et erreur angulaire. La courbe semble ici linéaire car on a $\tan \theta \simeq \theta$ quand $\theta \rightarrow 0$. . .	34
Figure 4.1	Avec cette scène en arrière-plan, les objets virtuels seraient visibles dans la zone 1 mais quasiment invisibles dans la zone 2, en plein soleil. . .	50
Figure 4.2	Vue de profil d'une rangée schématisée. Lorsque le paquet (1) est présent, le chargeur peut ramasser tous les paquets soulignés en vert, pour autant qu'ils soient sur le haut de la pile et accessibles.	55
Figure 4.3	Un paquet ramassable dans la rangée active est mis en valeur lorsque le regard de l'utilisateur se porte dessus.	56
Figure 4.4	Le paquet ramassé est placé sous le curseur pour ne pas gêner le pointage par l'utilisateur. On voit aussi le paquet de prévisualisation en «fil de fer».	57
Figure 4.5	Indication visuelle que la pile visée est pleine.	58
Figure 4.6	Menu circulaire montrant quatre options. On voit que l'option « exploration » est sélectionnée grâce à l'arrière-plan coloré. Le cercle central permet d'annuler la sélection	59
Figure 4.7	Frise chronologique du travail sur le GPU et le Central Processing Unit (CPU) pour les variantes avec présentation bloquante (1) ou non (2)	70

LISTE DES SIGLES ET ABRÉVIATIONS

API	Application Programming Interface
BDD	Base de données
CPU	Central Processing Unit
GPS	Differential Global Positioning System
FOG	Fiber Optic Gyroscope
GLONASS	GLObal NAVigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
GSM	Global System for Mobile communications
HF	Haute fréquence
HPU	Holographic Processing Unit
IGS	International GNSS Service
IHM	Interface homme-machine
IMU	Inertial Measurement Unit
IR	Infrarouge
LAI	Local Area Identifier
LIRV	Laboratoire d'infographie et de réalité virtuelle
MEMS	Microelectromechanical Systems
PPP	Precise Point Positioning
RA	Réalité augmentée
RSS	Received Signal Strength
RTK	Real-Time Kinematic
RV	Réalité virtuelle
SDF	Champ de distance signé
SLAM	Simultaneous Localization And Mapping
SSR	State Space Representation
TDoA	Time Difference of Arrival
ToA	Time of Arrival
ToF	Time of Flight
VRS	Virtual Reference Station

CHAPITRE 1 INTRODUCTION

Ce projet s'intéresse à la problématique liée à la création d'une interface de Réalité augmentée (RA) en milieu extérieur. Plus particulièrement, nous nous sommes placés dans le cadre de la logistique d'une cour à bois, avec comme but d'aider les opérateurs de machinerie lourde. Pour ce faire on utilise un visiocasque de RA existant : l'HoloLens de Microsoft. Les problèmes à résoudre pour obtenir un système fonctionnel sont multiples : beaucoup des systèmes de positionnement habituellement utilisés pour la RA ne sont pas viables en extérieur, il faut donc déterminer une technologie adaptée à partir de la littérature et de tests établissant les limites du matériel utilisé. Il faut aussi concevoir une interface et des interactions en RA adaptées à un contexte industriel.

1.1 Contexte

1.1.1 Cour à bois



Figure 1.1 La cour à bois du Groupe Crête à Saint-Faustin-Lac-Carré. On voit de multiples rangées de paquets de bois non rabotés. Sur la droite de l'image on voit l'une des usines de rabotage. À gauche on distingue aussi trois des séchoirs.

On a mentionné précédemment que notre projet a pour but d'être utilisé dans une cour à bois de scierie. Afin d'ancrer nos recherches dans le réel, nous avons pu nous baser sur le fonctionnement d'une cour à bois de la société Crête, visible sur la figure 1.1, et ce grâce à

l'aide de notre partenaire FPInnovations. De plus, nous avons restreint le cadre de l'étude à la seule cour de bois fini initialement, afin de ne pas introduire de complexité inutile lors de la modélisation du problème.

En effet, une cour à bois est généralement divisée en plusieurs zones. On a tout d'abord une zone de stockage des billes de bois, qui sont la matière première sur laquelle l'entreprise travaille. Ces billes sont ensuite sciées dans l'une des scieries de la cour pour produire des morceaux avec les dimensions approximatives du produit fini. Ces morceaux sont alors regroupés par lots et stockés dans une autre zone de la cour. Ces lots sont désignés paquets de bois humide, l'étape suivante étant en effet celle du séchage. Les paquets sont placés par lots sur des chariots, et sont ensuite introduits dans l'un des séchoirs de la cour. Trois de ces séchoirs sont visibles sur la gauche de la figure 1.1 avec des chariots de bois placés devant.

Cette étape a un rôle double : le séchage participe à la fois à modifier les propriétés mécaniques du bois, mais aussi à éliminer d'éventuels insectes présents dans le matériau, qui risqueraient de porter atteinte à l'intégrité structurelle du bois dans le futur. Cette étape a d'ailleurs une importance particulière en matière de traçabilité, l'entreprise devant être en mesure de prouver à ses clients que le bois vendu était séché correctement. En conséquence, chaque lot de séchage reçoit un numéro qui suivra chaque paquet jusqu'à la vente.



(a) Des paquets de bois séché, marqués avec le numéro du lot de séchage.



(b) Un chargeur à fourche est en train de charger des paquets de bois sur un camion de livraison.

Après le séchage, les paquets sont stockés par lot de séchage dans une troisième zone de la cour. On peut voir un exemple de ces paquets sur la figure 1.2a, et on notera le numéro de lot de séchage peint sur les paquets. La dernière étape de préparation du bois est alors le rabotage, où les morceaux de chaque paquet sont amenés à leur dimensions finales, puis rassemblés à nouveau et emballés pour être protégés des intempéries. À l'issue de ce processus, les paquets de bois fini sont placés dans une dernière zone de la cour : la cour à bois fini, celle qui nous intéresse pour cette étude. Dans cette zone, les paquets de bois sont disposés par

rangées en attente pour être chargés sur des camions (ou parfois des trains), comme montré sur la figure 1.2b, afin d'être livrés aux différents clients.

C'est ce flux entre l'entrée des billes de bois et la sortie du bois fini qui maintient l'équilibre de la cour, tandis qu'à plus petite échelle dans la cour de bois fini, l'équilibre doit être maintenu entre la sortie du rabotage et la livraison. De plus, le bois est un matériau qui vieillit, ainsi, afin d'éviter que certains paquets ne passent trop de temps stockés dans la cour par faute d'être accessibles, les rangées sont organisées en deux états :

- en cours de remplissage, où seuls des paquets neufs en provenance du rabotage sont ajoutés ;
- en cours de vidage, où les paquets sont alors retirés graduellement de la rangée pour être vendus.

Ce qu'il est important de retenir, c'est que le passage de l'état «vidage» à l'état «remplissage» ne se fait uniquement lorsque la rangée est vide. On s'assure ainsi qu'aucun paquet ne vieillit trop longtemps au fond d'une rangée.

1.2 Plan du mémoire

Dans le chapitre suivant, on effectue un revue de la littérature pertinente à notre sujet : on commence par une description de l'état de l'art puis on présente les technologies nécessaires au développement de notre système de RA en extérieur ainsi que des principes de design d'interfaces. On présente aussi les objectifs qu'on souhaite atteindre par la réalisation de cette étude.

Au chapitre 3, On commence par décrire en détail la conception des composants de notre application, puis dans un deuxième temps, on expose plusieurs expériences avec pour but de déterminer la capacité du visiocasque HoloLens à être utilisé dans un véhicule en extérieur.

On rapporte ensuite dans un premier temps au chapitre 4 les résultats des expériences menées sur l'HoloLens et décrites au chapitre 3, puis dans un deuxième temps, on décrit le développement d'un prototype de notre application.

Enfin, on discutera dans le chapitre 5 des résultats obtenus au chapitre précédent ainsi que des limites de notre étude, et on conclura le mémoire en fournissant de possibles ouvertures pour des travaux futurs au chapitre 6.

CHAPITRE 2 REVUE DE LITTÉRATURE

Pour ce projet, un des problèmes majeurs qui se présentent est celui de la mise en place d'un système de RA en extérieur. En conséquence, dans cette revue de littérature, on va tout d'abord s'intéresser à l'état de l'art en matière de systèmes de RA développés pour fonctionner dans un environnement extérieur. On passera ensuite en revue les technologies les plus pertinentes pour permettre la création d'un tel système. On s'intéressera ainsi aux différentes méthodes de positionnement, d'orientation, ainsi que la fusion de ces différentes mesures.

2.1 État de l'art

En comparaison de l'intérêt croissant apporté à la RA d'intérieur, le problème de la RA en extérieur reste relativement peu exploré. Cependant on peut retenir certaines contributions importantes à l'état de l'art. Un précurseur des expériences en RA d'extérieur est le projet ARQuake développé à partir de l'année 2000. ARQuake propose une adaptation du jeu de tir à la première personne Quake I en RA (Close et al., 2000). Dans cette version, le joueur porte un visiocasque transmissif qui ajoute des monstres virtuels dans l'environnement réel. Pour le suivi en position, le système utilise un GPS standard avec assistance DGPS, de plus, pour la détermination de l'orientation, un compas numérique est utilisé. De l'avis des auteurs eux-mêmes, un système d'acquisition de pose aussi rudimentaire que celui-ci ne permet pas d'obtenir une expérience très immersive pour le joueur (Thomas et al., 2002). En 2009, le Wearable Computer Lab, responsable pour le projet ARQuake, a présenté un nouveau système de RA, fournissant cette fois de la visualisation pour la conception architecturale (Thomas, 2009). Le positionnement a été considérablement amélioré, cependant l'ancrage des objets virtuels dans le monde réel reste toujours problématique, et ce malgré l'utilisation d'un Global Positioning System (GPS) précis à moins d'un mètre (Differential Global Positioning System (DGPS)) et d'une centrale inertielle plus précise.

Récemment, la RA a aussi été proposée pour augmenter l'expérience des touristes sur des sites d'importance culturelle (Caggianese et al., 2014). Pour compenser la mesure de position de faible qualité fournie par le système GPS standard, un système optique est utilisé en complément. Ce dernier se base sur une connaissance préalable de l'environnement, cette solution est donc adaptée à des sites culturels dont le paysage est en général fixé.

Malheureusement, un domaine dont l'intérêt pour la RA d'extérieur va croissant est celui

des technologies militaires. Dans leur article de 2014, Menozzi et al. décrivent un système de RA destiné à fournir des informations tactiques à un soldat (Menozzi et al., 2014). Cependant, l'aspect le plus intéressant de cet article est que grâce à la combinaison des données d'une centrale inertielle, d'un compas, d'un GPS, et en particulier d'algorithmes de vision par ordinateur, le système est capable d'atteindre une précision d'orientation de 0,01 rad en azimut. La partie optique du système se base sur la reconnaissance de la ligne d'horizon ainsi que de certains points remarquables pour obtenir l'information d'orientation. L'algorithme semble robuste aux occlusions partielles, mais demande tout de même un horizon visible pour fonctionner.

Enfin, en 2018, une entreprise commune entre Dura Vermeer et Recreate a produit un système de RA, utilisant le visiocasque HoloLens de Microsoft, capable d'afficher un rendu d'un bâtiment futur sur un site de construction (Xsens, 2018). Le projet promet une précision égale à celle d'un arporage professionnel. Ceci est confirmé par l'utilisation d'un appareil de positionnement Global Navigation Satellite System (GNSS)-Real-Time Kinematic (RTK), dont la précision est de quelques centimètres. Une centrale inertielle de haute qualité couplée à un magnétomètre est aussi utilisée pour fournir l'information d'orientation. Cependant, devant l'absence de publication académique, il est difficile de juger de la qualité de l'expérience.

2.2 Positionnement

Un problème majeur pour être capable d'afficher un environnement en RA est de pouvoir connaître avec précision la position et l'orientation du visiocasque dans le monde réel, afin de pouvoir lui faire correspondre un monde virtuel. Le degré de précision du positionnement (et de l'orientation) est le facteur majeur qui décide la qualité de la superposition du monde réel et du monde virtuel. La plupart des systèmes de RA sont conçus pour fonctionner dans un environnement intérieur contrôlé, or dans notre cas, nous devons être capables d'obtenir une position et orientation suffisamment précise dans des environnements extérieurs non préparés. Cela exclut donc de fait une partie non négligeable des technologies habituellement utilisées dans le domaine de la RA. Nous explorons ci-dessous l'état de l'art des technologies de positionnement.

2.2.1 Positionnement par satellite

Technologie de localisation disponible virtuellement partout sur terre, le positionnement par satellite s'est rapidement établi comme système de référence pour obtenir une information de localisation en extérieur. La première version de ce système, la constellation GPS reste

la plus présente dans l'imaginaire collectif, mais depuis le lancement du projet en 1973, la technologie a connu de nombreuses évolutions. À commencer par le lancement de multiples autres constellations de satellites comme Galileo (européen), GLObal NAvigation Satellite System (GLONASS) (russe) ou BeiDou-2 (chinois) pour ne citer seulement les GNSS globaux.

GNSS/GPS

Tous les systèmes GNSS sont basés sur une constellation de satellites qui émettent en permanence leur position et leur vitesse. Les récepteurs sur terre peuvent ensuite se baser sur ces signaux pour déterminer leur position par trilateration (FANG, 1986; Yassin et Rachid, 2015). Ce n'est cependant que la méthode la plus basique de détermination, et nous verrons dans les paragraphes suivants comment il est possible d'améliorer les résultats de cette méthode. Cela dit, le problème majeur de ces systèmes est que la qualité de leur mesure se dégrade de façon substantielle en intérieur et dans les canyons urbains. En effet, le signal GNSS est respectivement atténué au point de ne plus être exploitable, ou bien réfléchi de plusieurs manières différentes vers le récepteur, amenant ce dernier à calculer un *pseudo-range* erroné pour le satellite en question. Ce problème est connu sous le nom de *multipath* dans la littérature. Cependant, il est pertinent de noter que la plupart des cours à bois se situent en général loin des centres urbains et ne sont pas concernées par ce problème (Gouvernement du Canada, 2010).

RTK

Les systèmes GNSS classiques effectuent la mesure de distance à chacun des satellites utilisés en superposant une séquence pseudo aléatoire reçue avec la même séquence générée localement. La distance est déduite du décalage temporel entre les deux séquences et de la vitesse de propagation du signal. Un système RTK se base sur le même principe de correspondance, mais ignore les données de la séquence et tente directement de se caler sur la phase du signal porteur. Par exemple, pour le signal L1, qui est une des signaux émis par les satellites du système GPS, la longueur d'onde est de 19 cm, sachant que la mesure de phase permet d'obtenir une précision inférieure à longueur d'onde, on entrevoit déjà la performance possible de ce système. Cependant, à l'inverse de l'appairage avec une séquence pseudo aléatoire, il y a une ambiguïté égale à un multiple de la longueur d'onde lors de la mesure de la phase du signal. La résolution de cette ambiguïté demande l'application de méthodes statistiques (Verhagen et Teunissen, 2006). De plus, des perturbations au niveau de l'orbite des satellites, de leur horloge, de la transmission à travers l'ionosphère et la troposphère, dégradent la précision théorique obtenable par la mesure de la phase du signal porteur. Pour mesurer et compenser

ces erreurs, une installation RTK doit être composée d'au moins deux récepteurs : le récepteur mobile qui effectue les mesures recherchées que l'on nommera *rover* et le récepteur fixe, en communication constante avec le rover, en lui fournissant les coefficients de correction, déterminés de façon différentielle.

Network-RTK

Un désavantage majeur d'un système RTK standard est le besoin d'avoir une station de base à moins de 20 km du rover pour obtenir une précision acceptable pour les mesures. Cependant, à partir des observations de multiples stations de base, il est possible de modéliser les différentes erreurs du signal, et ce en utilisant moins de stations que si elles étaient toutes indépendantes. Ainsi, si pour couvrir une zone de 10 000 km² il aurait fallu environ 30 stations de base individuelles, ce nombre est réduit à 5 à 10 stations en combinant les observations (Wanninger, 2008). C'est là le concept primaire du network-RTK. Pour fournir des corrections de qualité, le réseau de référence suis en processus en trois étapes. Tout d'abord, toutes les stations du réseau doivent résoudre l'ambiguïté de leur signal reçu. Cette étape est rendue difficile par l'éloignement des stations (environ 50 à 100 km) et par le besoin de travailler en temps réel. Ensuite, à partir des signaux désambiguïsés, il est possible pour le réseau de calculer les différents coefficients de correction requis. Ces coefficients sont calculés indépendamment les uns des autres. Tous les coefficients n'ayant pas la même durée de validité, il est donc possible de les mettre à jour séparément. Dans certains systèmes Network-RTK, le traitement du réseau s'arrête à ce point et ces données sont transmises au rover qui réalise localement les calculs de l'étape suivante. Dans cette étape, les corrections pour une Virtual Reference Station (VRS) située aux coordonnées approximatives du rover sont calculées. Cette VRS remplace la station de base d'un système RTK standard.

Le ministère des Ressources Naturelles du Canada liste quatre réseaux RTK en activité dans le pays (Gouvernement du Canada, 2018). Cependant la couverture de ces réseaux est limitée aux seuls abords des centres de population. Ainsi, seule la partie sud du Québec est couverte par des réseaux RTK.

Precise Point Positioning (PPP)

Un problème majeur des systèmes RTK standards est qu'ils nécessitent une station de base fixe géographiquement proche pour fonctionner de par leur nature différentielle. Au contraire, la technologie PPP effectue une mesure absolue avec une précision équivalente aux méthodes RTK (PPP standard obtient une précision horizontale de 46 mm et verticale de 72 mm (Choy et al., 2017)). Pour obtenir une telle précision, la technologie PPP se base aussi sur la mesure

de la phase du signal porteur diffusé par les satellites des différentes constellations GNSS. Sans mesure différentielle disponible pour annuler ou mesurer les erreurs du milieu, PPP utilise des corrections de haute précision sur les orbites et les horloges des satellites fournies par exemple par l'International GNSS Service (IGS).

La technologie PPP souffre tout de même d'un problème majeur : le temps de convergence de la solution est extrêmement long. Ainsi, il faut environ 20 minutes pour obtenir une précision inférieure à 20 cm et 1 h pour une précision horizontale de 5 cm. De plus le temps de convergence est aussi long pour trouver la première solution (« démarrage à froid ») que pour des solutions subséquentes lorsque le signal satellite a été perdu. En cause : l'ambigüité de la phase du signal porteur qui n'est plus entière comme dans le cas de RTK mais réelle. La partie réelle est introduite par des biais matériels des satellites et du récepteur (Ces biais s'annulent dans le cas d'une mesure différentielle comme avec RTK). Un paramètre majeur de l'incertitude de mesure est l'état de l'ionosphère. Cet état varie en fonction du temps et de la position géographique, ce qui explique d'ailleurs en grande partie la limite de 20 km d'éloignement entre la base et le rover pour les systèmes RTK. Cependant, lorsque les variations de l'ionosphère sont lentes, il est possible après une perte de signal, de dériver les paramètres de correction afin de rétablir une solution bien rapidement que normalement possible avec du PPP standard. Il est aussi possible d'utiliser des données de correction ionosphériques fournies par un réseau dense de stations de mesure. Mais on perd alors l'avantage de la couverture globale du système PPP puisqu'on devient dépendant de l'existence de données de correction dans la zone en question (Choy et al., 2017). De plus, en utilisant simultanément les satellites de plusieurs constellations GNSS (aussi appelé multi-GNSS), il est possible d'améliorer à la fois le temps de convergence et la précision de la solution PPP d'environ 20 %.

PPP-RTK

Il apparaît que les technologies RTK et PPP sont très similaires, tout en apportant différents avantages au domaine du positionnement global. Naturellement, des systèmes hybrides ont été développés ces dernières années. On obtient alors des systèmes qui ont la rapidité d'acquisition de solution et la précision de la technologie RTK (quand des stations de référence sont disponibles), mais aussi la disponibilité globale du PPP.

2.2.2 Positionnement par stations de base

Un autre type de technologie de positionnement popularisée par la démocratisation des terminaux personnels mobiles se base sur les signaux de données émis par les stations de base Global System for Mobile communications (GSM)/2G/3G/4G et WiFi. Les mêmes principes

peuvent aussi s'appliquer à des balises ad hoc radio Haute fréquence (HF) ou encore à ultrasons. Quatre méthodes distinctes de traiter ces signaux se démarquent (Yassin et Rachid, 2015). Dans tous les cas, les positions des stations de base sont connues.

Identification des cellules

Cette première technique se base simplement sur la connaissance des stations visibles par le récepteur pour déterminer sa position. Dans le cas du réseau mobile, chaque station émet son Local Area Identifier (LAI), ce qui permet au récepteur de calculer une position grossière sans autre source de données. Si on veut une précision un peu plus élevée, ou bien faire fonctionner le système avec un réseau WiFi, il faut se baser soit sur des modèles de propagation radio, soit des empreintes de signaux mesurées pour établir la position du récepteur par rapport aux stations de base. Cette technologie de positionnement n'offre par contre qu'une précision de l'ordre de 200 m à 1 km (Yassin et Rachid, 2015).

Temps de trajet

Un autre type de positionnement se base sur la même méthode que celle des GNSS : la mesure du temps de trajet du signal depuis les stations de base. La technique Time of Arrival (ToA) mesure le temps de trajet absolu entre chaque émetteur et le récepteur. La connaissance du temps de trajet et de la vitesse de propagation du signal permet de déduire l'éloignement de chaque émetteur et donc de calculer par trilateration la position du récepteur. Cependant cette technique est extrêmement sensible à la synchronisation des horloges des stations de base *et* du récepteur.

Pour pallier ce problème, une autre technique a été développée : Time Difference of Arrival (TDoA). Avec cette méthode, ce n'est plus le temps de trajet absolu du signal entre émetteur et récepteur qui est mesuré, mais la différence de temps de trajet entre des signaux envoyés simultanément par toutes les stations de base. Ces mesures permettent aussi de calculer la position du récepteur et ont l'avantage de ne pas demander de synchronisation d'horloge du côté du récepteur. De plus la méthode n'est pas sensible aux décalages d'horloge entre les émetteurs, elle l'est par contre à un biais de synchronisation (Gholami et al., 2013). Ces techniques et leurs variantes fournissent une précision de 50 à 500 m quand utilisées sur un réseau mobile (Yassin et Rachid, 2015).

Triangulation

Souvent utilisé à tort pour désigner la trilateration, le terme triangulation désigne une méthode de positionnement basée sur la mesure des angles des triangles formés par des points connus et le point dont on cherche à déterminer la position. Dans le contexte de la localisation dans un réseau mobile ou WiFi, les stations de base sont équipées d'antennes directionnelles qui permettent de mesurer l'azimut depuis lequel le signal du terminal mobile est reçu. En combinant au moins trois de ces mesures, on peut déterminer par triangulation la position du terminal. Le problème évident de ce système est la nécessité de mettre en place des antennes directionnelles au niveau de tous les émetteurs, cela peut représenter un coût prohibitif, en particulier dans la mise en place d'un réseau WiFi privé.

Received Signal Strength (RSS)

Les méthodes de type RSS sont celles les plus couramment utilisées dans le contexte de réseaux WiFi. Ces méthodes se basent sur la puissance du signal reçu pour estimer la distance à chacun des émetteurs visibles. Selon la technique utilisée, l'estimation de la distance se fera en se basant soit sur des modèles théoriques de propagation des ondes, soit sur des mesures empiriques de puissances reçues associées à une base de données de positions.

2.2.3 Positionnement par lidar

Le lidar est un appareil de mesure utilisant un faisceau laser balayé pour obtenir un nuage de points représentant l'environnement par une mesure du temps de retour du signal. Cet instrument est habituellement utilisé pour des tâches de cartographie, mais les algorithmes Simultaneous Localization And Mapping (SLAM) permettent de dériver un déplacement relatif des observations. De plus, si on connaît déjà la topologie de l'environnement, il est même possible d'extraire une position absolue des mesures en faisant correspondre par des méthodes statistiques les données mesurées avec la cartographie existante. En utilisant un algorithme SLAM, il est possible d'atteindre une précision de positionnement décimétrique. Cependant, cette précision se dégrade avec le temps, en particulier dans des conditions où les repères se font rares (Qian et al., 2016).

2.2.4 Le positionnement de l'HoloLens

Le visiocasque HoloLens intègre de nombreux senseurs :

- 4 caméras d'environnement, pointées sur l'avant et les côtés de l'appareil.

- Une centrale inertielle (ou Inertial Measurement Unit (IMU)) composée d'un accéléromètre, un gyroscope et un magnétomètre.
- Une caméra Time of Flight (ToF) mesurant la profondeur sur un champ de vision de 120° par 120° , avec une faible consommation énergétique.
- Une caméra pointée droit devant de 2,4 MP
- Quatre microphones
- Un capteur de lumière ambiante

Seules les caméras d'environnement et l'IMU participent à l'acquisition de la position et de l'orientation de l'HoloLens. Le traitement des données de mesures brutes est effectué par un processeur séparé et spécialisé (appelé Holographic Processing Unit (HPU) par Microsoft). Microsoft indique que cette méthode permet une accélération non négligeable par rapport à des algorithmes purement logiciels (Colaner, 2016).

La façon exacte dont le HPU combine les données des différents senseurs pour obtenir une information de pose est un secret industriel de Microsoft. Cependant, en se basant sur le fonctionnement de méthodes existantes, on peut avancer quelques hypothèses éclairées. Ainsi, il est probable que les caméras d'environnement alimentent un algorithme de type flux optique permettant de mesurer les mouvements du décor par rapport au visiocasque. Ce traitement est sûrement appuyé par l'IMU afin de gagner en précision, tandis que la dérive de l'IMU doit être en permanence corrigée par les données des caméras. De plus, on constate à l'utilisation que l'HoloLens est capable de conserver un système de coordonnées stable dans chaque environnement où il est utilisé, on peut donc supposer que les caméras sont aussi utilisées pour identifier chaque environnement dans lequel le visiocasque est utilisé. On constate cependant que le temps nécessaire pour identifier un environnement particulier est comparable à celui nécessaire à l'HoloLens pour réaliser le *Spatial Mapping* de la pièce, cet aspect joue donc peut-être aussi un rôle dans l'identification de chaque espace.

2.3 Orientation

L'autre problème majeur qu'il nous faut résoudre pour obtenir une superposition adéquate des mondes réel et virtuel, est celui de la mesure de l'orientation. L'orientation désigne la mesure de la position angulaire en 3 dimensions. En général, on mesure trois angles : le roulis (*roll*), le tangage (*pitch*) et le cap (*yaw* ou *heading*) parfois aussi appelé lacet. Il est à noter que les méthodes de mesures de ces trois quantités (présentées ci-après) ne sont pas forcément égales en matière de précision.

2.3.1 Centrale inertielle (IMU)

Les IMU, ou centrales inertielles en français, sont des systèmes combinant plusieurs appareils de mesure pour obtenir l'orientation de l'appareil. L'accélération propre du système est aussi mesurée par l'IMU, ce qui lui permet de calculer une position relative. Cependant, la qualité de cette estimation se dégrade rapidement à cause de l'accumulation des erreurs d'intégration. Une centrale inertielle comprend généralement les appareils de mesure suivants sur chacun des trois axes :

- gyromètres ou gyroscopes (respectivement fournissant la vitesse et la position angulaire.
Souvent ces deux informations sont fournies par le même appareil)
- accéléromètres
- magnétomètres

Chacune de ces familles de capteurs encapsule un éventail de technologies dont les caractéristiques ainsi que le prix varient de façon substantielle.

Microelectromechanical Systems (MEMS)

La technologie que l'on retrouve maintenant le plus fréquemment dans les IMU de basse et moyenne gamme est celle des MEMS qui permet d'intégrer des systèmes mécaniques miniaturisés avec le même format que les circuits intégrés classiques. Les processus de fabrication des MEMS ont d'ailleurs évolué des méthodes de manufacture des puces de silicium. Ces méthodes sont fondamentalement bidimensionnelles, pour obtenir une mesure sur chacun des trois axes x , y , et z , deux capteurs identiques sont positionnés à 90° sur le *wafer* tandis que le dernier doit être conçu différemment pour effectuer une mesure sur l'axe normal à la surface du *wafer*. Comparativement aux autres technologies présentées ici, les capteurs MEMS ont un coût relativement faible de fabrication ce qui les rend très attractifs pour les applications grand public, ainsi la plupart des téléphones intelligents embarquent un accéléromètre et un magnétomètre trois axes qui réalisent un capteur d'attitude.

Les gyroscopes fabriqués en utilisant cette technologie sont des gyroscopes vibrants : ces gyroscopes possèdent une masse placée en résonance (un cylindre, un diapason, une demi-sphère ou encore une roue), lorsque l'appareil est mis en rotation, la force de Coriolis dévie la trajectoire de la masse, et c'est cette déviation qui est mesurée.

Quant aux accéléromètres, ils sont souvent considérés comme étant les dispositifs MEMS les plus simples, en effet un accéléromètre est modélisé par une masse retenue par un ressort amorti dont on mesure la déformation. En pratique ce ressort prend la forme d'une simple poutre avec une masse connue au bout.

Enfin, les magnétomètres MEMS mesurent l'effet de la force de Lorentz sur un conducteur déformable. On rappelle qu'en présence d'un courant et d'un champ magnétique, une force est appliquée sur le conducteur dans la direction orthogonale au plan formé par le conducteur et la direction du champ magnétique. La déformation peut ensuite être mesurée par différents moyens :

- En mesurant la variation de potentiel causé par la déformation de piézo résistances.
- En mesurant la modification de la fréquence de résonance mécanique de la partie mobile.
- Par mesure optique directe de la déformation du mobile.

Il faut aussi noter que ces systèmes sont sensibles à des variations de température puisque ces dernières modifient les caractéristiques mécaniques des matériaux auxquels la précision des mesures est liée.

Fiber Optic Gyroscope (FOG)

Une autre technologie de gyroscope plutôt répandue de nos jours est celle des gyroscopes FOG. La mesure de vitesse angulaire dans ces gyroscopes est réalisée grâce à l'effet Sagnac : un décalage temporel est mesurable entre deux signaux lumineux tournant en sens inverse sur la circonférence d'un disque en rotation par rapport à un référentiel galiléen. Dans le cas de la technologie FOG, le signal est propagé dans une fibre optique enroulée sur elle-même un grand nombre de fois afin d'accentuer l'effet. Le signal est envoyé aux deux bouts de la fibre par l'intermédiaire d'un séparateur optique, puis la différence de temps de trajet est mesurée par interférométrie.

La précision qu'il est possible d'obtenir avec cette technique est à plusieurs ordres de grandeur au-dessus des gyroscopes MEMS. Cette technologie à l'avantage de ne comporter aucune pièce mobile et de n'être donc que peu sensible aux vibrations, accélérations et chocs.

Cependant, leur prix reste encore de nos jours relativement élevé, de l'ordre de la dizaine à centaine de milliers de dollars.

2.3.2 Antennes GNSS multiples

On a mentionné précédemment dans la sous-section 2.2.1 l'existence de récepteurs GNSS capables d'une précision centimétrique, voire millimétrique. Cette précision a permis le développement d'une nouvelle catégorie de compas basée sur l'utilisation de plusieurs antennes GNSS-RTK fixées sur un même support. En fixant ces antennes à différents points connus et mesurés sur un véhicule, il est possible de calculer l'orientation du véhicule à partir des

différences de position constatées entre tous les récepteurs. Cette méthode est par exemple utilisée dans le milieu maritime ou à bord d'avions.

En fonction du nombre d'antennes utilisées, il est possible d'accéder à tout ou seulement une partie des angles de rotation du véhicule. Ainsi, avec uniquement deux antennes installées, on perd la composante de rotation colinéaire à l'axe formé entre les deux antennes. On notera cependant que si ces dernières ne sont pas installées verticalement, il est possible d'obtenir l'angle de lacet qui, on le rappelle, est l'angle dont la précision de mesure laisse le plus à désirer avec les méthodes magnéto-inertielles. Avec l'ajout d'un troisième récepteur, le système devient capable d'une mesure selon les trois axes de tangage, roulis, et lacet.

La précision atteignable par cette méthode est de l'ordre de $0,1^\circ$ selon l'article (Giorgi et al., 2010). De multiples modèles commerciaux vantent de la même façon des précisions en matière de cap entre $0,1^\circ$ et $0,3^\circ$ pour un séparation d'un mètre entre les antennes. De plus, cet arrangement des antennes avec une géométrie connue permet aussi d'accélérer grandement la résolution de l'ambiguïté entière de phase.

2.4 Fusion de capteurs

Comme il est apparu au cours des paragraphes précédents, toutes les techniques de mesure de la position ou de l'attitude mentionnées ont des inconvénients, des situations où la qualité de la mesure devient grandement dégradée. On pense par exemple à la perte d'acquisition des signaux satellite pour un récepteur GNSS. Cependant, tous les systèmes explorés ne sont pas sensibles aux mêmes perturbations. Il est donc possible de mettre à profit plusieurs méthodes de mesure différentes simultanément afin qu'elles se renforcent mutuellement.

2.4.1 Filtre de Kalman

Le filtre de Kalman, dont la théorie a été publiée en 1960 par Rudolf E. Kálmán (Kalman, 1960), est une méthode statistique, qui à partir de mesures de précisions variables ainsi que bruitées, d'une modélisation de l'évolution physique du système, et facultativement des entrées de contrôle, est capable de donner une meilleure estimation du vecteur d'état que celle fournie directement par les capteurs. De plus le filtre de Kalman modélise les corrélations entre les différentes mesures, ce qui le rend extrêmement adapté à la fusion de différents capteurs, par exemple une centrale inertielle et un récepteur GNSS.

Avant de décrire le fonctionnement du filtre de Kalman original, on pose les notations suivantes :

- x : vecteur d'état du système.

- \hat{x}_k : estimateur de la prédiction du vecteur d'état à l'instant k .
- \hat{x}'_k : estimateur mis à jour du vecteur d'état.
- P_k : matrice de covariance de l'état
- Q_k : incertitude externe, l'incertitude causée par des facteurs externes non maîtrisés.
- F_k : matrice de prédiction, modélise le comportement du système à partir de son état précédent.
- z_k : vecteur de mesure, regroupant les données des capteurs.
- H_k : matrice d'observation, décrit la transformation entre les mesures des senseurs et les valeurs physiques réelles.
- K_k : gain de Kalman, définit l'importance relative donnée à l'état courant contre l'état mesuré.
- R_k : matrice de covariance du bruit de mesure.
- u_k : vecteur de contrôle, contient les différentes commandes envoyées au système.
- B_k : matrice de contrôle, représente la loi de commande du système.

Le filtre de Kalman est récursif et n'a donc uniquement besoin des informations d'état et de mesure de l'itération précédente, ce qui le rend très avantageux en matière de consommation mémoire lors de l'implémentation algorithmique. Pour chaque itération du filtre de Kalman, on distingue deux phases : la phase de prédiction et la phase de mise à jour.

Phase de prédiction

Durant cette phase, on se base sur la connaissance de l'état précédent \hat{x}_{k-1} , la modélisation de l'évolution du système F_k , et si ces données existent, les entrées de commande du système $B_k u_k$. On fait aussi une prédiction quant à la covariance du vecteur d'état P_k , en prenant en compte les perturbations extérieures qui ne sont pas modélisées Q_k .

$$\hat{x}_k = F_k \hat{x}_{k-1} + B_k u_k$$

$$P_k = F_k P_{k-1} F_k^T + Q_k$$

Phase de mise à jour

Ensuite, durant la phase de mise à jour, on combine l'état prédit \hat{x}_k avec les mesures des capteurs z_k en pondérant avec le gain de Kalman K_k . Le gain de Kalman détermine pour

chaque élément du vecteur d'état et chaque mesure l'importance respective à leur accorder en fonction des incertitudes de chaque variable, la combinaison de Q_k et R_k .

$$K_k = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$$

$$\hat{x}'_k = \hat{x}_k + K_k(z_k - H_k \hat{x}_k)$$

$$P'_k = P_k - K_k H_k P_k$$

Filtre de Kalman étendu

Un des problèmes majeurs du filtre de Kalman est qu'il ne fonctionne qu'avec des systèmes dont le modèle est strictement linéaire. Or il se trouve que la majorité des systèmes réels ne sont pas linéaires. Pour palier ce problème, le filtre de Kalman étendu utilise les matrices jacobiniennes des fonctions non-linéaires (qui doivent toutefois être différentiables) (calculées à chaque itération) qui modélisent le système, ce qui fournit une approximation linéaire du premier ordre pour les matrices F_k et H_k . Par contre cette linéarisation locale a la malheureuse conséquence que la convergence du filtre n'est plus garantie.

2.4.2 GNSS + IMU

Une association de technologies très couramment utilisée pour la navigation en environnement extérieur est celle d'un récepteur GNSS combiné avec une centrale inertuelle. En effet, un récepteur GNSS peut fournir une position absolue d'une assez grande précision, en particulier lorsqu'il est compatible avec la technologie RTK, par contre il n'est pas capable de mesurer son attitude et est sensible à la perte du signal satellite. À l'inverse, une centrale inertuelle est capable de fournir des mesures de variation de son attitude et vitesse avec une grande précision, sans dépendance à un signal extérieur. Par contre la connaissance de la position et de l'attitude est seulement relative au point de départ et souffre d'erreurs d'intégration exponentielles au cours du temps.

La fusion des mesures de ces deux appareils permet à la fois de compenser la perte du signal GNSS pendant une courte durée de temps (quelques minutes) en utilisant la position intégrée par l'IMU, mais aussi de recalier régulièrement la centrale inertuelle en utilisant les données de position et de vitesse provenant du système GNSS. En particulier, on est capable de compenser les biais du gyroscope ainsi que de détecter les perturbations affectant le magnétomètre.

La fusion de ces mesures est en général réalisé avec un filtre de Kalman ou un de ses filtres

dérivés.

2.4.3 Capteurs et prédictions secondaires

Lorsqu'on a une connaissance du véhicule auquel le système de navigation est destiné, il est possible de s'appuyer des prédictions propres au comportement du véhicule ainsi que sur des capteurs embarqués afin d'améliorer la précision atteignable par l'algorithme de fusion de capteurs. En effet, en ajoutant des contraintes au modèle, on diminue l'incertitude sur le résultat calculé. Ainsi, dans le cas d'un véhicule terrestre il est possible de supposer que son déplacement peut être déduit à partir de la vitesse de rotation des roues ainsi que l'orientation des roues directrices. Cependant, cette hypothèse peut s'avérer hasardeuse dans le cas où l'adhérence des roues est dégradée, par exemple lorsque le terrain est boueux, glacé ou enneigé. Il faut dans ce cas comparer le risque de perte d'adhérence avec la perte de précision engendrée par cet évènement.

2.4.4 Vision

Même avec l'aide des données d'un récepteur GNSS, il reste une imprécision inhérente à la centrale inertielle, en particulier au niveau de la mesure du cap. Le problème est que le récepteur ne mesure pas cette valeur et ne peut fournir qu'une correction indirecte des biais de l'IMU. L'idéal serait d'avoir une autre mesure directe d'attitude, ce que certains algorithmes de vision par ordinateur sont capables de fournir. En utilisant une caméra fixée au châssis du véhicule dont on veut connaître l'attitude, on peut extraire à intervalle régulier des *features* de l'image. En comparant le mouvement de ces *features* entre deux images, il est possible de calculer, en connaissant la perspective, la rotation et la translation que le véhicule a subies entre les deux images. Cependant cette technique de mesure est sensible à des mouvements parasites dans le champ de la caméra.

2.4.5 Lidar + IMU

Contrairement à l'IMU dont la performance reste relativement constante quel que soit l'environnement (à l'exclusion des perturbations de champ magnétique), un lidar a besoin de points remarquables dans son environnement pour pouvoir être utilisé pour une mesure de positionnement ou d'attitude, par l'intermédiaire d'un algorithme de SLAM. Par contre ce type d'algorithme est capable de donner une position localement absolue, par rapport à l'environnement, au contraire de l'IMU qui ne travaille qu'à l'estime. La combinaison des deux technologies par l'intermédiaire d'un filtre de Kalman permet d'augmenter drastiquement la

précision, d'autant plus que les mesures d'une IMU sont en général plus précises lorsqu'elle est en mouvement, tandis qu'un lidar a une excellente précision en mode statique, les deux appareils étant donc complémentaires. Il a été constaté que même en utilisant des appareils de faible qualité autant pour l'IMU que pour le lidar, il est néanmoins possible d'obtenir une précision centimétrique pour le positionnement de points d'intérêt de l'environnement, ce qui correspond donc à la précision envisageable pour le positionnement d'éléments dans l'environnement virtuel (Tang et al., 2015).

2.4.6 GNSS + IMU + lidar

Les solutions lidar + IMU sont en général mises en place en intérieur, mais lorsqu'on a affaire à un environnement extérieur, il est possible d'ajouter au système un récepteur GNSS, fournissant une position absolue dans le référentiel mondial, et offrant des corrections lorsque le lidar manque de points d'intérêt à détecter pour exécuter son algorithme de SLAM. On peut aussi concevoir le problème d'un autre point de vue : si l'on rajoute un lidar à un système GNSS + IMU, on gagne l'annulation de la dérive angulaire, grâce au lidar qui se recèle sur l'environnement. De plus cette correction fonctionne aussi à l'arrêt, à l'inverse de celle qu'il est possible d'obtenir avec un récepteur GNSS qui se dégrade de façon notable lorsque la vitesse est nulle. Ainsi, il a été montré (Qian et al., 2016) que dans un environnement contenant à la fois des zones de mauvaise réception satellite et des zones avec peu de points d'intérêt (l'avantage étant que le plus souvent ces caractéristiques sont mutuellement exclusives), on est capable d'obtenir une précision de 0.13 m de positionnement des éléments de l'environnement.

2.5 Instrumentation d'un chargeur

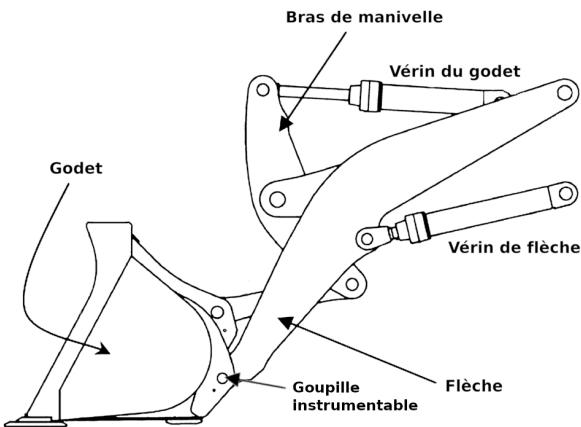
Notre projet se place dans un contexte industriel, où il doit assister l'opération de véhicules de manutention. On cherche donc à obtenir des informations sur l'état courant de ces véhicules. On a déjà mentionné à la sous-section 2.4.3 les possibilités apportées par des mesures odométriques sur le véhicule pour aider au positionnement, mais dans un cadre de manutention, on s'intéresse aussi à connaître la présence ou non d'un chargement, et si possible des informations sur ce dernier.

2.5.1 Balance de fourche

Le standard de l'industrie pour obtenir des informations sur un chargement porté en fourche est d'utiliser un système nommé balance de fourche. Le principe de base consiste à déterminer l'effort appliqué par le poids du chargement aux différents éléments du bras de levage du

chargeur. On distingue alors généralement deux approches différentes d'instrumentations pour effectuer cette mesure.

Un premier type de système se base sur la cinématique du bras de levage ainsi que les caractéristiques du circuit hydraulique pour convertir la pression mesurée dans ce dernier pour évaluer l'effort subi par les vérins de levage, qui est lui-même corrélé au poids supporté par le bras de levage (Hindman, 2008). Sur le schéma de la figure 2.1a, ces vérins sont les vérins de flèche, le vérin du godet étant lui-même supporté par la flèche. Dans la thèse citée précédemment, la précision obtenue quant à la mesure de la masse du chargement est d'environ $\pm 1\%$ de la masse du chargement.



(a) Mecanisme avant des chargeurs



(b) Un chargeur à fourche

Un autre type de système utilise quant à lui des goupilles instrumentées à l'intérieur du mécanisme du bras de levage. L'instrumentation des goupilles (simples pièces mécaniques, sans parties mobiles) est réalisée à l'aide de jauge de contraintes, qui mesurent la déformation du matériau sous l'effort. Une fois encore, la connaissance de la géométrie du bras permet de faire la corrélation entre l'effort mesuré dans la goupille et le poids subi par la fourche. Par exemple, il est possible d'instrumenter les goupilles de l'articulation de la fourche elle-même, comme montré sur la figure 2.1a, court-circuitant ainsi une grande partie des incertitudes et difficultés de modélisation créées par le reste du mécanisme (A. F. Kheiralla et al., 2012).

2.5.2 Estimation de pose dans la cabine

Beaucoup des technologies de positionnement et orientation présentées ci-dessus ne sont pas compatibles avec une utilisation solidaire d'un visiocasque porté sur la tête, à l'intérieur d'une cabine. En effet, un lidar où encore une antenne GNSS-RTK sont des objets trop lourds et encombrants pour être portés sur la tête, sans mentionner les perturbations introduites dans les mesures par l'utilisation à l'intérieur d'une cabine d'engin de machinerie lourde.

En conséquence, en complément des méthodes présentées ci-dessus, on s'intéresse aussi à des technologies d'estimation de pose à l'intérieur d'un volume défini. À l'inverse des solutions dédiées à la RA en milieu extérieur, il existe de nombreuses solutions commerciales abouties venant de recherche sur la Réalité virtuelle (RV) pour effectuer le suivi d'un visiocasque dans un volume prédéfini. Elles sont en général séparées en deux catégories : celles qui utilisent un suivi *inside-out*, où les capteurs actifs sont situés sur le visiocasque et utilisent des informations extérieures pour se positionner ; et celles qui font un suivi *outside-in*, le visiocasque ne contenant que de l'équipement passif et le positionnement étant mesuré par des capteurs extérieurs. Par exemple, le système de suivi intégré à l'HoloLens rentre dans la catégorie *inside-out* avec ses caméras d'environnement.

Balises Infrarouge (IR)

De nombreux systèmes de suivi sont basés sur des balises lumineuses. En général, pour éviter les désagréments, ces systèmes utilisent des longueurs d'onde du domaine infrarouge, invisibles à l'œil humain. De nombreux systèmes commerciaux, en particulier dans le domaine de la RV, utilisent ce type de technologie. Il en existe des versions avec un visiocasque passif : c'est le cas de l'Oculus Rift qui utilise un ensemble de marqueurs IR placés sur le casque et deux caméras installées dans la salle, chargées de positionner le visiocasque. La répartition des balises sur le visiocasque étant connue, il est possible à partir de l'image de ces balises capturée par la caméra de calculer la position et l'orientation de la caméra par rapport au visiocasque. En général on utilise plusieurs caméras pour obtenir de la redondance et une meilleure couverture de l'espace. Ce système est capable de fournir une précision de positionnement centimétrique (Borrego et al., 2018).

Une autre technologie utilisant des balises IR est le système à balayage laser. Dans ce système, les balises sont installées autour de l'espace de suivi, ces balises effectuent chacun des balayages de la zone selon les axes verticaux et horizontaux avec des lasers IR. Le visiocasque possède quand un à lui un ensemble de récepteurs infrarouges et est capable de mesurer les décalages de temps entre la réception d'un signal lumineux par chaque senseur. La connaissance de ces décalages temporels, couplée avec la connaissance de la géométrie du casque et des vitesses de balayages des lasers, permet d'obtenir la position de chacun des senseurs, et donc la pose complète du visiocasque (Islam et al., 2016). SteamVR™Tracking, le système de suivi commercial développé par Valve utilise cette technique. Les mesures de précision effectuées sur cette implémentation particulière varient selon les études de 0,2 mm et 0,02° à 1 cm de précision (Borrego et al., 2018; Oliver Kreylos, 2016; Ng et al., 2017). Il est aussi notable que des modules de suivi multi-usage pour ce système sont vendus par Valve et que

de la documentation est fournie pour aider au développement d'autres modules customisés. Ainsi une équipe a pu concevoir un module de suivi pour des lunettes stéréoscopiques (Ng et al., 2017).

Suivi optique

De nombreuses applications en RA utilisent pour se positionner dans le monde une ou plusieurs caméras en général montées sur l'appareil. Le principe est de déterminer les caractéristiques du mouvement entre deux images capturées successivement. Pour cela on détermine des points d'intérêts présents dans les deux images. Les différences de position de ces points sur les images permettent alors de déterminer le mouvement de la caméra par rapport à ces points-là. Cette méthode s'appuie sur l'hypothèse qu'une majorité du décor est fixe. Or comme on l'a vu avec l'HoloLens, l'environnement à bord d'un véhicule en mouvement n'offre pas cette garantie, ce qui rend cette technique beaucoup moins attractive pour notre utilisation.

IMU et fusion de senseurs

Toutes les méthodes présentées dans les paragraphes précédents souffrent d'un problème majeur : leur taux de rafraîchissement, qui atteint par exemple seulement 60 Hz avec toutes les techniques existantes utilisant des balises IR. Bien que ce taux de rafraîchissement soit égal à celui de l'affichage d'un visiocasque, dans le pire des cas, si l'on accède à l'information de position juste avant sa prochaine mise à jour, on obtient une mesure datant de la précédente itération.

Pour cette raison, un grand nombre de systèmes de suivi, comme par exemple ceux des visiocasques de RV Oculus Rift et HTC Vive, utilisent une IMU comme source principale pour la mesure de position et d'orientation. Cette dernière est capable de réaliser un échantillonnage jusqu'à 1000 Hz dans le cas du système SteamVR™Tracking. Cependant on a vu dans la sous-section 2.3.1 que les mesures de positions fournies par une IMU souffrent de double erreur d'intégration avec des erreurs de plusieurs mètres par seconde. De plus, une IMU ne peut fournir qu'une mesure relative de position.

Afin de contrer ce phénomène, on fusionne les mesures à haute fréquence mais relatives et à déviation rapide de l'IMU avec celles d'un autre système de suivi absolu à plus basse fréquence. Ainsi, la double erreur d'intégration de la centrale inertielle ne peut jamais s'accumuler qu'entre deux mises à jour du système de suivi absolu.

2.6 Principes de design

2.6.1 Interface homme-machine (IHM)

Le design d'interface est une discipline très subjective. Cependant, de nombreux principes existent pour guider le processus. Ainsi, Ören et Yilmaz exposent plusieurs principes bien connus et développent aussi leur propre ensemble de directives pour la conception d'IHM (Ören et Yilmaz, 2005). On décrit ces quatre directives dans les paragraphes qui suivent :

Utilisabilité C'est ce qu'on désigne généralement comme le caractère intuitif (ou non) d'une interface, bien que le terme consacré dans le domaine du développement d'IHM soit la « familiarité » (Raskin, 1994). Les facteurs contribuant à améliorer l'utilisabilité d'une interface sont sa simplicité, dans le sens que chaque action est atteignable en un minimum d'actions de la part de l'utilisateur et que l'information présentée n'est pas ambiguë ; son apprentissage aisément, en se basant sur des conventions existantes pour fournir un environnement déjà partiellement familier à l'utilisateur ; sa charge mentale minimale pour l'utilisateur, qui ne doit avoir à mémoriser que le minimum d'informations.

Il va sans dire que tout cela ne doit pas se faire au détriment de la fonctionnalité de l'application. Il y a un équilibre à trouver entre simplicité et fonctionnalité.

Communication claire On peut étendre l'idée de fournir un environnement familier pour faciliter l'apprentissage au reste de l'interface : celle-ci devrait de façon générale respecter du mieux qu'elle peut les normes culturelles et esthétiques locales. Ces normes se retrouvent en particulier dans le design visuel de l'application que l'on explorera plus en détail dans un paragraphe suivant. De plus, le langage utilisé par l'application devrait respecter les utilisateurs ainsi que le vocabulaire spécifique du domaine.

Un autre aspect important d'une communication claire est d'être capable de fournir la bonne information au moment opportun, voire de proposer des conseils contextuels à l'utilisateur.

Robustesse L'interface devrait protéger l'application sous-jacente, ainsi que l'utilisateur. L'IHM devrait filtrer les entrées erronées par l'utilisateur, et tolérer les erreurs qui ne peuvent être empêchées. On cherchera à faire en sorte de prévenir l'utilisateur dans le cas où des actions dangereuses sont possibles.

De plus, afin d'éviter de placer inutilement l'utilisateur dans une situation où il risquerait d'être induit en erreur, le design de l'interface doit être cohérent, prévisible : Un même élément, une même action, doivent avoir une signification, un effet similaire dans des situations

différentes.

Capacité d'évolution Enfin, l'interface doit être capable d'évoluer, autant pour s'adapter aux préférences de différents utilisateurs, que pour être portée vers d'autres plateformes quand le matériel doit évoluer. Autant que possible, la maintenance devrait être facilitée par l'IHM. On retiendra cependant que ces directives existent surtout à titre indicatif, et qu'en fonction de l'application développée, certains points ne seront pas applicables, tandis que d'autres prendront une importance toute particulière. Ainsi, dans le cas d'un programme en RA, le design visuel prendra une importance prépondérante.

2.6.2 Design visuel

Le design visuel d'une application a le rôle crucial de suppléer au texte seul pour fournir de l'information à l'utilisateur sur l'état actuel du programme ainsi que ses potentielles transitions. D'ailleurs, toujours dans le même article (Ören et Yilmaz, 2005), les auteurs présentent une liste de principes de design recommandés par IBM spécialement dédiée au design visuel.

Il est ainsi proposé d'aborder le design de façon soustractive plutôt qu'additive : au lieu de la méthode habituelle consistant à ajouter des éléments à l'interface jusqu'à obtenir un résultat satisfaisant, on préfèrera retirer des éléments jusqu'à seul le nécessaire persiste. Cette approche est particulièrement adaptée à la conception d'une interface pour un visiocasque autonome tel que l'HoloLens où chaque élément d'interface dessiné représente un drain potentiel de ses ressources.

La liste insiste aussi sur les concepts de hiérarchie visuelle et de thème : les éléments importants doivent être mis en valeur par rapport aux autres éléments de l'interface, formant ainsi une hiérarchie. La mise en valeur peut être faite en jouant sur la couleur, la taille, l'alignement, ou même le caractère curviligne ou rectiligne de l'élément en question. C'est aussi dans ce cadre qu'on pourra mettre en pratique les normes esthétiques et culturelles dans le design afin de maîtriser l'information communiquée par le design.

Ensuite, l'idée d'affordance est introduite. L'affordance étant la façon dont un objet (ou un élément d'interface) suggère par son existence la manière de l'utiliser (Gaver, 1991). Dans le cadre du design d'IHM, on souhaite que les éléments de l'interface rendent l'utilisation de l'application évidente à un utilisateur. On revient ainsi au concept évoqué plus tôt d'interface intuitive.

Par ailleurs, les interfaces en RA ont la particularité d'exister dans un monde tridimensionnel présent tout autour de l'utilisateur, quand bien même le champ de vision de ce dernier est

réduit à une seule fraction de la combinaison de ces mondes réels et virtuels. En conséquence de quoi, il peut parfois être nécessaire de guider le regard de l'utilisateur vers un élément situé hors de son champ de vision. Dans l'article (Schwerdtfeger et Klinker, 2008), les auteurs comparent plusieurs méthodes pour attirer et guider le regard en RA.

Enfin, un autre facteur à garder en tête lors de la conception d'une interface en RA est que les méthodes de pointage utilisées (mouvement de la tête dans le cas de l'HoloLens) peuvent être relativement difficiles à utiliser. En vertu de la loi de Fitts, les tailles des cibles d'interaction doivent donc être choisies en conséquence (Fitts, 1954; Kytö et al., 2018).

2.6.3 Qualité de l'expérience

Pour appuyer la conception de notre application, ainsi que pour juger de la qualité de l'expérience qu'elle fournit, on a besoin d'un cadre conceptuel sur lequel se baser. Dans leur article, Roberts et al. (Roberts et al., 2013) présentent cinq critères permettant de décrire quantitativement la qualité de l'expérience visuelle d'une application en RA, en particulier concernant le placement d'objets virtuels en superposition du monde réel. Les critères sont les suivants :

Sautillement Le sautillement est un mouvement à haute fréquence (perceptible sur une échelle de temps de l'ordre de la seconde ou moins) l'objet virtuel par rapport à l'environnement. Ce phénomène est généralement une conséquence de bruit dans la mesure de pose.

Dérive La dérive est un mouvement lent, perceptible sur des durées allant de la minute à quelques heures. Il peut être causé par un glissement du référentiel absolu de certains des capteurs de position ou d'orientation utilisés.

Traînage Le traînage, mesuré en secondes (ou plus généralement en millisecondes au vu des durées concernées), désigne le décalage temporel entre un mouvement du visiocasque et la réaction correspondante de l'objet virtuel affiché. Visuellement, l'objet virtuel semble «traîner» derrière l'objet réel correspondant. Cela peut par exemple s'expliquer par un délai trop long de traitement entre l'obtention des informations de pose et l'affichage de la scène dans le visiocasque.

Rebond Le rebond est un mouvement vertical de l'objet virtuel par rapport au monde réel qui apparaît durant la marche de l'utilisateur. Ce point particulier est moins pertinent dans notre cas puisque le visiocasque est prévu pour être utilisé par un utilisateur assis dans un

véhicule. Cependant, il est possible que des cahots lors du déplacement du véhicule puissent générer du rebond dans la scène virtuelle.

Précision La précision quant à elle est caractérisée par l'erreur de positionnement constante d'un objet virtuel par rapport au monde réel. Elle peut en général être corrélée directement avec la précision des capteurs de positionnement et d'orientation.

Comme le soulignent Roberts et al., on cherche en général à minimiser tous ces critères simultanément. Cependant, ils sont en général interdépendants et une amélioration sur l'un aura généralement une conséquence néfaste sur un autre. Par exemple, on peut essayer de réduire le sautillement se basant sur plusieurs mesures successives afin de diminuer l'influence du bruit, mais on risque alors d'aggraver le traînage. On est donc face à un compromis.

2.7 Objectifs de recherche

Au cours de ce projet de recherche, on souhaite compléter les objectifs suivants :

- (O1) Analyser les caractéristiques du visiocasque HoloLens afin de déterminer avec précision les limites de certaines capacités, en particulier en matière de possibilités de positionnement.
- (O2) Proposer et développer une solution d'affichage en RA dans un véhicule de manutention en extérieur :
 - (a) mettre en place une solution de positionnement du visiocasque dans un véhicule de manutention en extérieur ;
 - (b) développer en parallèle la détermination de l'orientation du visiocasque afin d'avoir une solution d'estimation de pose complète ;
 - (c) à partir du positionnement et de l'orientation, être capable d'afficher un objet virtuel fixe dans le référentiel du monde réel.
- (O3) Proposer et développer une interface en RA pour l'application fournissant des informations contextuelles aux opérateurs de machinerie lourde.
- (O4) Concevoir un système de détection du ramassage ou du dépôt des paquets de bois par les chargeurs.
- (O5) Étudier et mettre en place l'architecture réseautique pour la mise en relation de tous les nœuds du système global de la cour à bois.
- (O6) Intégrer l'application avec un système d'inventaire centralisé de la cour à bois avec un support de plusieurs visiocasques actifs simultanément.

CHAPITRE 3 MÉTHODOLOGIE

Dans ce chapitre, nous présenterons tout d'abord les étapes de la conception du système que nous souhaitons développer, puis dans une deuxième partie, nous décrirons le protocole expérimental que nous utiliserons pour réaliser nos tests sur le visiocasque HoloLens.

3.1 Conception

Afin de développer un système complexe comme celui proposé dans notre étude, on se doit de séparer sa conception suivant les sous-systèmes majeurs. Pour chacun d'entre eux, on décrira les problèmes à surmonter, puis on proposera une ou plusieurs solutions permettant de les résoudre. On discutera aussi des différents aspects d'éventuels compromis s'il y a lieu.

On distingue les parties suivantes : la cour à bois, qui est en fait l'environnement dans lequel notre application est destinée à évoluer ; le visiocasque, élément majeur d'interaction physique avec l'utilisateur ; les systèmes de positionnement et d'orientation, items de premier plan dans le fonctionnement d'un système de RA ; les chargeurs et leur instrumentation, ce sont ces véhicules de machinerie lourde qui fournissent un support à toute la partie mobile de l'application ; la réseautique, qui regroupe les problèmes de connectivité entre les acteurs mobiles et fixes ; et enfin l'interface utilisateur, dont le rôle est de transmettre les informations du système à l'utilisateur et d'en retour recevoir les ordres de ce dernier.

3.1.1 Cour à bois

On a déjà évoqué le fonctionnement de la cour à bois dans la sous-section 1.1.1, Cependant on veut en préciser ici quelques aspects. En particulier, à la fin de l'étape de rabotage, les paquets de bois sont emballés dans une bâche protectrice et gagnent simultanément une étiquette (visible sur la figure 3.1b) contenant les informations suivantes :

- Grade du produit
- Taille des morceaux contenus
- Essence du bois (*p.e.* pin ou épinette)
- Nombre de morceaux
- Numéro de lot de séchage
- Date

- Informations relatives au traitement de séchage

On constate toujours sur la Figure 3.1a que l'étiquette est illisible à moins de s'en approcher à une très courte distance. Les informations que celles-ci contiennent sont donc virtuellement inaccessibles aux opératrices des chargeurs, qui, par la géométrie de leur engin, ne sont que très rarement à une distance suffisamment faible des paquets. Grâce à l'utilisation de la RA dans ce projet, on souhaite rendre ces informations plus accessibles.



(a) Un paquet de bois fini, situé à la sortie d'une des usines de rabotage. L'étiquette informative est mise en valeur.



(b) Détail d'une étiquette sur un paquet de bois. On distingue par exemple la taille des morceaux, le grade *Premium* du bois, ou encore le numéro de lot de séchage.

De plus, une fois ces paquets sortis de l'étape de rabotage, ils sont empilés et organisés par rangées dans la cour à bois, comme montré sur la figure 3.1. La hauteur maximale d'empilement est dictée par la géométrie des chargeurs. Il est d'ailleurs à noter qu'une pratique courante est pour un chargeur de soulever deux paquets déjà empilés afin d'obtenir un étage supplémentaire sur la pile de paquets. On a constaté que dans la cour où nous avons pu faire nos observations, la longueur maximale des rangées est d'environ 30 m.

3.1.2 Visiocasque

Pour fournir toutes les informations visuelles à l'opératrice, on souhaite utiliser un visiocasque de RA. Il existe deux types principaux de visiocasques pour la RA :

- visiocasque à transmission directe ;
- visiocasque fermé.



Figure 3.1 Plusieurs rangées dans la cour à bois fini. On remarque que la rangée la plus haute compte sept étages. La photo est prise de l'arrière des rangées et on voit que la rangée la plus à gauche est quasiment vide.

Dans le premier cas, l'utilisateur voit le monde réel directement à travers un écran semi-transparent sur lequel peuvent aussi être projetés des éléments du monde virtuel. Le second type de visiocasque quant à lui, isole totalement l'utilisateur du monde extérieur et utilise des écrans pour la totalité de l'affichage, les informations venant du monde réel étant capturées par des caméras. Ce type de casque a l'avantage d'être capable de fournir une excellente intégration entre les informations du monde réel et virtuel.

Cependant il souffre aussi d'un énorme défaut dans un contexte industriel : en cas de panne quelconque, l'opératrice est totalement aveuglée jusqu'à ce que la panne se résorbe ou bien qu'elle retire le visiocasque. Ce n'est pas un mode d'erreur acceptable lorsqu'on conduit un engin de machinerie lourde. À l'inverse, avec un visiocasque à transmission directe, la plupart des pannes réduisent le visiocasque à une simple visière transparente. Même dans le cas où l'affichage deviendrait complètement opaque, seule une partie du champ de vision de l'utilisateur est couvert.

Idéalement, on souhaiterait avoir un visiocasque capable de connaître son attitude et position de façon autonome. Cependant, l'état de l'art des visiocasques en matière de positionnement

n'est peut-être pas suffisant pour répondre aux critères de cette étude. Pour le déterminer, on fait donc le choix d'utiliser un visiocasque embarquant un système de positionnement dont on pourra juger des capacités.

Au moment du début de cette étude, le seul visiocasque de RA répondant à ces critères et étant disponible commercialement était l'HoloLens de Microsoft. De plus le Laboratoire d'infographie et de réalité virtuelle (LIRV) et son partenaire FPIInnovation avaient déjà accès à ce matériel ce qui a permis de cimenter ce choix.

Comme expliqué à la sous-section 2.2.4, l'HoloLens utilise un système de positionnement totalement contenu dans le visiocasque pour fournir une mesure de position et d'orientation de haute qualité dans un environnement intérieur et statique. Une partie de notre travail sera donc de déterminer les limites de ce matériel une fois placé à l'intérieur d'un véhicule en mouvement.

Pour développer une application en réalité augmentée sur l'HoloLens, Microsoft propose deux approches :

- utiliser le moteur Unity, qui possède des interfaces avec les API de l'HoloLens
- développer en utilisant directement les API de l'HoloLens et l'API graphique Direct3D 11.

Ces deux approches diffèrent radicalement, il serait donc logistiquement ardu de les explorer toutes deux dans un temps raisonnable. Il y a en conséquence une réflexion à avoir sur les mérites respectifs de ces deux solutions. Le processus décisionnel est aussi influencé par des conclusions empiriques obtenues d'un usage de ces technologies pré-datant l'étude actuelle.

On a ainsi constaté que malgré une relative facilité pour le prototypage, la complexité et l'instabilité de l'environnement de développement Unity ainsi que les performances décevantes de l'application par rapport à sa complexité apparente ne jouent pas en la faveur de cet engin. Le haut niveau d'abstraction apporté par Unity cache au développeur les informations nécessaires pour prendre de bonnes décisions de conception, en particulier en matière de performance. Par exemple, lorsqu'on souhaite charger et afficher de nombreux éléments géométriques identiques, l'utilisation de méthodes naïves mises en valeur par Unity n'aboutit pas à une performance satisfaisante.

À l'inverse, l'approche à bas niveau rendue accessible par l'utilisation de Direct3D permet un contrôle aigu des facteurs affectant la performance. On maîtrise complètement l'implémentation de la chaîne d'affichage et on peut se limiter au strict nécessaire. Étant donné qu'on souhaite concevoir une application fonctionnant sur un système disposant d'une autonomie limitée tel que l'HoloLens, la recherche de performance est cruciale pour diminuer la consom-

mation d'énergie. La performance d'une application est aussi un facteur majeur d'ergonomie, en effet on a mis en lumière dans la sous-section 2.6.3 que le traînage généré par un délai de latence dans l'application pouvait dégrader l'expérience d'utilisation.

On propose donc de coder l'application en C++ en utilisant l'API Direct3D 11, pour donner la priorité à la performance et au gain de compétence avec ces technologies.

3.1.3 Technologies de positionnement

Un défi majeur à relever pour toutes les technologies de RA est d'arriver à superposer des objets virtuels avec le monde réel avec suffisamment de précision pour que les utilisateurs aient l'impression que l'objet en question est rattaché au monde réel. Ce problème est majoritairement résolu dans le cas d'un environnement intérieur, mais — comme évoqué dans la section 2.1 — dans un environnement extérieur, l'état de l'art laisse à désirer. Dans la partie qui suit, nous allons nous intéresser au principal problème auquel on doit faire face lorsqu'on développe un système de positionnement pour de la RA en extérieur : la précision du système en question. On exposera ensuite nos recommandations quant aux solutions adaptées pour résoudre ce problème.

Nous voulons ici déterminer la précision minimale nécessaire en matière de position et orientation. Pour cela, il nous faut déterminer le niveau d'erreur acceptable pour notre expérience de RA. Dans la cour à bois, la plus petite dimension des paquets de bois est d'environ un mètre. Comme évoqué à la sous-section 3.1.1, on a mesuré que la profondeur des rangées est d'environ 30 m. On double cette valeur pour prendre en compte le fait que l'opératrice doit être capable de voir les informations sans être parfaitement en face de la rangée. Cette marge couvre aussi le fait que dans les piles les plus hautes il peut y avoir des paquets 3 à 5 m au-dessus de la ligne de vue de l'opératrice. Pour récapituler, on a donc :

- $d_{\max} = 60 \text{ m}$: la distance maximale entre le visiocasque et un paquet de bois sur lequel on veut superposer de l'information ;
- $l_{\min} = 1 \text{ m}$: la largeur minimale d'un paquet de bois dans la cour.

On s'intéresse à l'erreur de positionnement d'un objet virtuel situé à d_{\max} de l'opératrice causée par le cumul d'une erreur de positionnement du visiocasque, ainsi que par une erreur de la mesure angulaire du cap. On pose déjà que l'erreur linéaire ε_α créée à d_{\max} par une erreur angulaire α_e est :

$$\varepsilon_\alpha = \tan(\alpha_e) \cdot d_{\max} \quad (3.1)$$

On pose ensuite ε_l comme l'erreur de positionnement en translation du visiocasque qui est

transférée directement à l'objet virtuel. L'objectif ici est d'être capable de pointer un objet réel avec un objet virtuel de taille nulle (on peut imaginer la pointe d'une flèche), ce qui a pour conséquence de nous permettre de considérer toute valeur de ε_α dans l'intervalle $[-\frac{l_{\min}}{2}, \frac{l_{\min}}{2}]$ comme valide. On doit donc vérifier l'inégalité suivante :

$$\frac{l_{\min}}{2} - \varepsilon_l - \tan(\alpha_e) \cdot d_{\max} > 0 \quad (3.2)$$

On trace sur la Figure 3.2 la zone couverte par cette inégalité. On remarque qu'au vu des technologies mentionnées dans les parties 2.2, 2.3 et 2.4, le critère d'erreur angulaire est bien plus contraignant, en particulier au niveau de la mesure du cap qui souffre en général de plus d'imprécision que celles du tangage ou du roulis.

Enfin, il ne faut pas perdre de vue que si l'on sépare notre système de positionnement en deux sous-systèmes : le suivi du chargeur et le suivi du visiocasque dans le chargeur, les erreurs de ces deux sous-systèmes s'additionnent. On a alors :

$$\varepsilon_l = \varepsilon_{\text{chargeur}} + \varepsilon_{\text{visiocasque}} \quad (3.3)$$

$$\alpha_e = \alpha_{\text{chargeur}} + \alpha_{\text{visiocasque}} \quad (3.4)$$

Avec $\varepsilon_{\text{chargeur}}$, α_{chargeur} , $\varepsilon_{\text{visiocasque}}$, et $\alpha_{\text{visiocasque}}$ étant respectivement les erreurs linéaires et angulaires des sous-systèmes de positionnement du chargeur et du visiocasque dans le chargeur.

Suivi du chargeur

À partir de la limite $l_{\min}/2 = 0,5 \text{ m}$ on peut immédiatement éliminer nombre de méthodes de positionnement. Tout d'abord, à l'exception des technologies RTK et PPP, les systèmes GNSS sont incapables de fournir une précision inférieure à 1 m. Ensuite, on a déterminé dans la sous-section 2.2.2 que les méthodes de positionnement basées sur des stations de base WiFi ou GSM ont une précision limitée à quelques dizaines de mètres. On peut donc éliminer directement ce type de systèmes de la liste des technologies viables pour notre application.

À partir de l'équation 3.2, on détermine que l'erreur angulaire maximale acceptable dans le cas idéal où $\varepsilon_l = 0$ est $\alpha_e = 0,01 \text{ rad} = 0,57^\circ$. Il s'avère que la plupart des IMU MEMS commerciales ne sont pas capables de fournir une mesure absolue de cap avec une précision suffisante, à cause des effets combinés des perturbations dans la mesure du champ magnétique terrestre et des biais du gyroscope. Ainsi, si l'on regarde seulement les chiffres pour chacun des systèmes considérés en isolation, les deux principales technologies candidates sont le GNSS

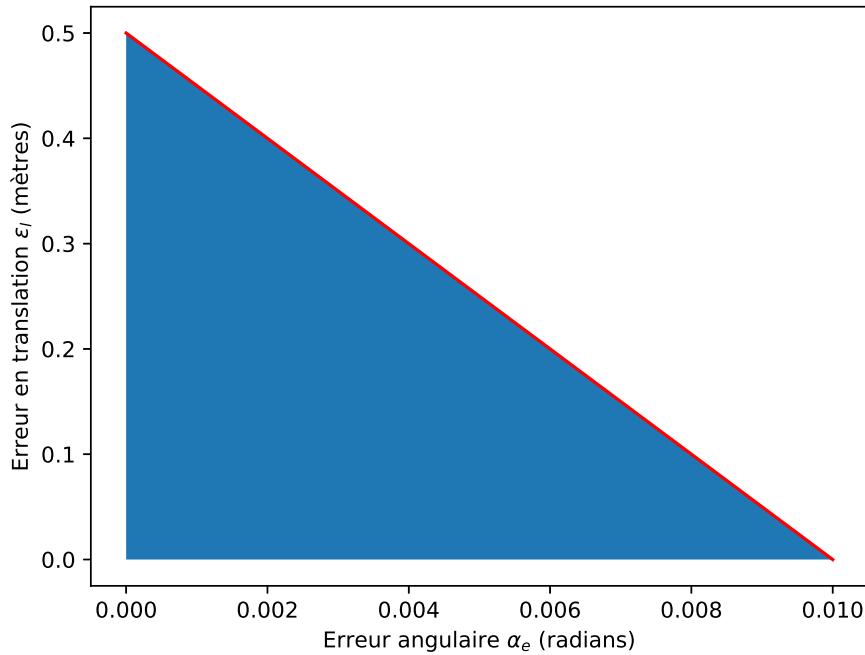


Figure 3.2 Zone d'acceptabilité du couple erreur en translation et erreur angulaire. La courbe semble ici linéaire car on a $\tan \theta \simeq \theta$ quand $\theta \rightarrow 0$

haute précision et le lidar, en particulier si l'on utilise plusieurs antennes GNSS afin d'obtenir des informations d'orientation, comme expliqué dans la sous-section 2.3.2.

Cependant, il ressort de la section 2.4 qu'on peut grandement améliorer la précision des mesures en fusionnant les informations venant des plusieurs capteurs différents à l'aide d'un filtre de Kalman ou une de ses variantes. Avec cette vision, on peut regarder quel est l'apport de chaque capteur. Ainsi, une IMU va fournir des informations de position et d'orientation avec une très grande fréquence, et une assez grande précision, mais qui se dégradent extrêmement rapidement par double intégration de l'erreur. À l'inverse, on a vu qu'un GNSS-RTK peut fournir une position absolue, bien qu'à plus basse fréquence, et sans information d'orientation lorsqu'on utilise qu'un seul récepteur. Quant au lidar, il pourra fournir des données de position et d'orientation précises à l'aide d'un algorithme de SLAM, mais cette méthode n'est pas robuste à une perte d'acquisition.

Ainsi, on a vu dans la sous-section 2.4.6 qu'en combinant un récepteur GNSS, une IMU et un lidar, on obtient une précision plus qu'acceptable. Cependant on se retrouve face à un problème de coût, chacun de ces équipements ayant déjà seuls un prix non négligeable. De plus, dans le contexte d'un montage sur un chargeur à fourche, le lidar risque d'être perturbé

par les éléments mobiles du véhicule ainsi que son chargement. À l'inverse, pour fournir une information d'orientation absolue, un double récepteur GNSS ne serait pas gêné par ce genre de problèmes. La conclusion de cette réflexion sur les technologies de positionnement en extérieur reste tout de même qu'on est face à un compromis, on recommande d'utiliser des technologies complémentaires, mais la disponibilité de ces dernières au moment de l'implémentation jouera un rôle majeur pour effectuer un choix.

Suivi du visiocasque dans la cabine du chargeur

On vient de proposer des technologies capables de fournir des mesures de position et d'orientation avec la précision attendue pour notre application en RA. Cependant il apparaît très vite que la plupart de ces solutions sont peu adaptées à une installation solidaire du visiocasque qui pèserait sur la tête de l'utilisateur. De plus, une installation à l'intérieur de la cabine perturberait les mesures des systèmes dépendant de signaux extérieurs, comme le GNSS ou le lidar. Pour ces raisons, on sépare le système de localisation en deux : une partie mesurant l'état du véhicule dans le référentiel monde, puis une deuxième partie mesurant l'état du visiocasque dans le repère du véhicule.

En se basant sur les informations de précision exposées dans la sous-section 2.5.2, on discerne immédiatement que les systèmes à base de balises IR assistées par IMU offrent une excellente précision par rapport aux autres technologies. Cette supériorité est d'ailleurs confirmée par le fait que ce type de technologie est utilisé dans le système de suivi de deux des modèles de visiocasque de RV les plus populaires.

On ajoute à cela le fait que le système Lighthouse développé par la société Valve est disponible à l'utilisation par tous pour effectuer le suivi de n'importe quel objet dans un espace contrôlé, tel que par exemple un environnement CAVE, comme dans l'article (Ng et al., 2017), ou bien à l'intérieur de la cabine d'un chargeur.

Le seul problème potentiel avec cette méthode est que le signal IR pourrait être perturbé par la lumière du soleil. Il faudra donc effectuer des essais afin de confirmer que le système est robuste à des conditions de grand ensoleillement.

3.1.4 Interconnexion entre les acteurs

Notre application est destinée à fonctionner avec plusieurs chargeurs opérant en même temps dans la cour à bois. Pour atteindre ce but, il faut que la connaissance de l'environnement, en particulier l'état de la cour à bois, soit commune à tous les acteurs du système. Pour ce faire, on doit établir une communication entre tous les acteurs, comme décrit à la sous-section 3.1.5,

et à travers cette dernière synchroniser l'information.

Architecture

Il est possible d'aborder cette problématique de deux façons différentes :

- avec un système de communication pair-à-pair ;
- avec une architecture client-serveur.

Le principal intérêt d'une architecture pair-à-pair est qu'elle permet de se passer d'un serveur central en répartissant le traitement de l'information sur tous les acteurs du système. Un système conçu de cette façon est en général assez robuste à la défaillance d'un des membres puisqu' aucun d'entre eux n'est un point unique de défaillance. Par contre, comme chaque noeud doit être synchronisé avec tous les autres, le besoin en bande passante est élevé. De plus ces systèmes doivent maintenir la cohérence des données de façon distribuée, en gérant les éventuels conflits entre les noeuds, augmentant ainsi la complexité de l'ensemble.

À l'inverse, en utilisant une architecture client-serveur, on allège la charge sur le réseau, surtout du côté des clients, puisque chacun d'entre eux ne communique qu'avec le serveur. C'est d'autant plus important qu'on a vu à la sous-section 3.1.5 que les chargeurs pouvaient être connectés par des réseaux sans fil de faible débit. Un autre avantage d'utiliser un serveur distinct est qu'il peut servir de source de données pour d'autres applications de l'entreprise, par exemple une application web permettant de visualiser l'état de l'inventaire en temps réel.

Dans le cas de notre projet, un système d'inventaire centralisé est développé en parallèle dans notre laboratoire par Dany Naser Addin. En plus des avantages mentionnés ci-dessus, ce facteur conforte notre décision d'utiliser un système de communication centralisé pour notre application.

Priorités

Pour l'implémentation du système de communication, nous définissons des priorités quant aux capacités de ce dernier.

1. *Impact minimal sur la performance* Un point crucial pour conserver une bonne qualité d'expérience pendant l'utilisation de l'application est qu'il faut conserver une bonne performance en tout temps pour l'affichage (sous-section 2.6.2). Par conséquent, il est inacceptable que le processus de communication ralentisse le processus de rendu de l'application.

2. *Robustesse* L'ensemble du système (serveur central et visiocasques) doit être résistant à une perte de connexion au reste du réseau. En effet, malgré toutes les précautions que l'on peut prendre, la connectivité à bord d'un véhicule en mouvement n'est jamais garantie à 100%. On peut même conjecturer que ce genre d'évènement pourra être courant, en particulier dans les cours à bois disposant d'une mauvaise couverture sans fil. Cependant, il n'est pas envisageable de stopper toutes les activités de manutention à chaque fois qu'un nœud du système perd sa connexion. On veut donc que chaque visiocasque soit capable de travailler en isolation le temps de retrouver une connexion, et que la synchronisation subséquente n'entraîne aucun problème de cohérence dans le système d'inventaire.
3. *Facilité d'intégration* Notre application a vocation à être utilisé en milieu industriel. On peut spéculer qu'un nombre non négligeable d'entreprises souhaitant intégrer notre système possèdent déjà un réseau interne. Ainsi notre but est d'utiliser des technologies les plus universelles et fiables possibles, et en apportant qu'une importance secondaire à la performance en matière de réseautique.

De nos jours, le protocole TCP/IP a une place prépondérante dans les télécommunications, et est sous-jacent à la plupart des réseaux informatiques. Il répond donc bien à notre objectif d'universalité. De plus ce protocole possède un mécanisme de détection et correction d'erreur ce qui est un autre point positif du point de vue de la fiabilité. Par comparaison, l'utilisation du protocole UDP, en particulier sur des liens de mauvaise qualité comme on peut en rencontrer hors des grandes agglomérations nécessiterait l'implémentation d'un système de gestion d'erreur au niveau de la couche applicative ce qui n'est pas souhaitable pour conserver une application simple.

De la même façon, pour le protocole de couche applicative, on propose d'utiliser le protocole HTTP. Ce protocole est loin d'être le plus efficace en termes de performances, mais il a l'avantage d'être universellement utilisé, étant à la base du *World Wide Web*. Les réseaux d'entreprise sont donc généralement déjà prêts à gérer ce genre de trafic, ce qui n'est pas le cas de protocoles moins répandus.

3.1.5 Chargeur et instrumentation

Les chargeurs à fourche fournissent le moyen principal de déplacement des paquets dans la cour à bois, et sont donc de ce fait les principaux agents de modification de l'état de la cour. C'est aux opérateurs et opératrices de ces engins que l'on souhaite fournir une assistance à l'aide de la RA. Pour ce faire, plusieurs problèmes technologiques sont à résoudre qui demanderont l'installation de matériel sur les chargeurs :

Instrumentation de la fourche

Enfin, pour compléter les fonctionnalités de la partie embarquée du système, on a besoin d'un moyen de savoir si le chargeur transporte ou non un paquet de bois à un instant donné. Pour cela, trois approches majeures se distinguent : La première et de loin la plus simple à implémenter serait de demander à l'opératrice d'informer le système lorsqu'elle dépose ou ramasse un paquet de bois. Cependant cette solution ajoute une charge mentale supplémentaire non négligeable à l'opératrice alors que l'optique de ce projet est justement de diminuer cette charge, en l'état, cette solution n'est donc pas satisfaisante.

Afin de libérer l'opératrice de la tâche de désigner elle-même les paquets en cours de manutention au système, il faut détecter de façon automatique le ramassage des paquets de bois. Pour cela, on propose les deux approches suivantes :

- la détection du poids du paquet
- la détection de la forme du paquet

Pour la première méthode, des produits existants sur le marché peuvent être intégrés dans des engins de machinerie lourde pour mesurer le poids porté par la fourche d'un chargeur. Ainsi on a mentionné dans la sous-section 2.5.1 les différents systèmes de balances de fourche, qui se basent sur des mesures de pression dans le circuit hydraulique du chargeur ou bien sur des mesures de charge de certaines pièces du mécanisme de levage. L'avantage de cette méthode est que l'on a une mesure fiable. Cependant l'installation du système sur le véhicule est relativement intrusive.

Il est possible d'explorer d'autres techniques moins standards dans l'industrie : ainsi on pourrait par exemple utiliser de simples capteurs de distance voire des capteurs de contact pour détecter la présence d'un objet solide sur la fourche. Ces capteurs étant situés au plus près de la zone de levage de la fourche, ils courrent un plus grand risque d'être endommagés.

La seconde méthode quant à elle se base sur une caméra et un algorithme de vision par ordinateur pour détecter la présence d'un paquet de bois sur la fourche à partir de sa forme. Cette solution a l'avantage d'être moins intrusive à l'installation sur le chargeur. Elle permet de plus aussi de faire la différence entre un paquet de bois et un autre type de chargement que l'opératrice pourrait être éventuellement amenée à transporter, une distinction que le capteur de charge n'est pas par sa nature pas capable de faire. Cependant la détection de forme a aussi ses problèmes : tout d'abord, cette méthode est très sensible aux conditions environnementales, qui sont justement très variées sur une cour à bois (jour, nuit, soleil, pluie, neige, etc.). De plus, à cause du problème mentionné précédemment ainsi que la géométrie du chargeur, un algorithme fiable serait extrêmement complexe à développer. Une solution alternative pour

avoir une reconnaissance de forme plus fiable et plus simple à développer serait d'utiliser un lidar, qui est en effet capable de fournir un nuage de point correspondant à la forme du paquet, quelles que soient les conditions climatiques. Mais le coût prohibitif d'une telle technologie limite l'utilisation réaliste de cette méthode au cas où le lidar est aussi incorporé au système de positionnement.

En conclusion, la solution la plus robuste dans notre cas est d'utiliser une balance de fourche pour détecter la présence ou non de paquets de bois sur la fourche du chargeur. C'est donc celle-ci qu'on privilégiera.

Réseautique

On a déterminé ci-dessus que notre système était composé de nombreuses parties discrètes, en particulier sur la plateforme que représente le chargeur, mais aussi comme on le verra à la sous-section 3.1.4, à l'extérieur du véhicule. Afin d'obtenir un système fonctionnel, il est nécessaire que la communication entre les différents composants se fasse correctement. Les capacités réseautiques du visiocasque HoloLens sont relativement limitées : l'HoloLens peut seulement se comporter en client WiFi et non en point d'accès, et son port USB n'est utilisable que pour l'alimentation électrique ainsi que pour accéder à l'interface de débogage. De plus, les autres éléments du système peuvent utiliser un vaste ensemble d'interfaces de communication. Ainsi, certaines antennes GPS fournissent une interface WiFi, tandis que d'autres permettent de recevoir les données via une connexion USB ou RS232.

Ainsi, malgré le caractère autonome du visiocasque en matière de traitement de données, il est en pratique incapable de communiquer directement avec tous les autres appareils présents à bord du chargeur. De plus, sa puissance de calcul est limitée par ses capacités de dissipation thermique et par l'autonomie de sa batterie. On a donc besoin de mettre en place un système servant d'intermédiaire entre tous les modules embarqués et effectuant certains prétraitements lourds.

On propose donc de mettre en place un sous-réseau dédié à la communication intra véhiculaire dans chacun des chargeurs. Les besoins pour ce réseau sont décrits ci-dessous :

- Connectivité WiFi pour pouvoir communiquer avec le visiocasque et potentiellement d'autres équipements.
- Interfaces physiques correspondants aux besoins des différents capteurs (par exemple : USB, RS232, ou autres connecteurs moins répandus).
- Capacité de prétraitement des données brutes (par exemple pour effectuer de l'analyse d'image, ou exécuter un algorithme de SLAM sur la sortie d'un lidar).

- Latence faible entre l'acquisition des données de position et d'orientation et leur réception par le visiocasque.
- Connectivité avec le reste du réseau à l'extérieur du véhicule.

Pour répondre à ces besoins, on distingue deux extrêmes : une approche complètement modulaire et une approche monolithique.

Approche modulaire Quasiment tous les items énumérés ci-dessus correspondent à un module particulier du réseau. Ainsi, la connectivité WiFi peut être fournie par un point d'accès sans-fil, qui en général dans un réseau de cette taille est intégré avec le routeur. Pour agréger les données des différents capteurs, il est possible d'utiliser des cartes d'acquisitions spécialisées en fonction de la connectique rencontrée. Cependant, des connecteurs standards tels que l'USB ou le RS232 sont présents dans la majorité des cas, limitant le besoin pour de multiples interfaces d'acquisition. Quant au prétraitement des données acquises, un ordinateur, ou même un nano-ordinateur, serait en mesure d'exécuter un grand nombre des algorithmes utilisés.

Un problème majeur émerge tout de même de cette architecture : en séparant chaque module physiquement, on introduit de la latence entre chacun des composants. En particulier, dans le pire des cas on introduit des étapes de communication réseau entre un des capteurs, l'ordinateur, le routeur, le point d'accès sans-fil, et enfin le visiocasque. Cependant, les transmissions sur un réseau câblé local offrent une latence inférieure à la milliseconde.

Approche monolithique On remarque que la plupart des modules cités ci-dessus (routeur, point d'accès, interface USB, etc.), sont généralement présents dans un ordinateur portable moderne. Seul le connecteur RS232 est usuellement absent, et la fonctionnalité peut-être très facilement obtenue en utilisant un adaptateur USB. En remplaçant la majorité des modules du réseau par une unique machine, on élimine ainsi les latences de communication entre l'ordinateur, le routeur et le point d'accès. Les seules latences de communications restantes étant entre les capteurs, l'ordinateur et le visiocasque.

La latence dans une transmission WiFi est aussi de l'ordre de la milliseconde, mais celle-ci peut être facilement dégradée par des transmissions concurrentes sur le même canal radio (Zhang et al., 2018). Il se trouve qu'une cour à bois est en général un environnement peu perturbé en matière d'accès aux canaux WiFi, et on a en général un plus grand contrôle sur les émissions parasites que dans un contexte plus urbain. De plus, à l'inverse du contexte expérimental utilisé par Zhang et al. (Zhang et al., 2018), nous ne voulons pas utiliser le réseau WiFi pour transmettre des informations visuelles, mais seulement des informations

additionnelles de pose. La quantité d'information ainsi que les restrictions quant à la latence sont donc moindres.

Pour la communication avec l'extérieur du véhicule, plusieurs possibilités s'offrent à nous. En effet, pour établir une connexion Internet suffisamment stable entre les chargeurs et le serveur central (qu'on a évoqué à la sous-section 3.1.4), on a besoin d'un système de communication sans fil, supportant l'établissement d'une connexion IP, couvrant l'ensemble de la cour, et qui n'est pas prohibitivement cher et complexe à mettre en place. Nous avons retenu deux possibilités majeures qui s'alignent avec ces requis. Tout d'abord on peut utiliser le réseau mobile existant (3G ou 4G/LTE), l'avantage étant que si une station de base existe près de la cour à bois, cette dernière sera probablement couverte dans sa totalité. Par contre, le désavantage de cette solution est la perte de contrôle sur une partie de notre infrastructure en déplaçant la responsabilité vers un opérateur de téléphonie mobile. Cette solution nécessite aussi d'ajouter au réseau embarqué un modem sans fil afin de pouvoir communiquer avec la station de base. En fonction des technologies utilisées, ce dernier pourra être intégré à l'ordinateur, connecté à l'ordinateur, voire connecté au routeur.

L'autre méthode consisterait à utiliser une connexion WiFi, ce qui d'un côté simplifie le système embarqué puisque des émetteurs/récepteurs WiFi sont déjà à bord, mais qui implique d'un autre côté de déployer une couverture WiFi sur toute la cour. Sachant que la portée que l'on peut attendre d'un point d'accès WiFi est de quelques dizaines de mètres, il est nécessaire d'installer de multiples points d'accès pour obtenir la couverture attendue. Cependant, il y a la possibilité bien réelle que la cour soit déjà équipée et qu'on puisse ainsi bénéficier d'une installation existante.

3.1.6 Interface Utilisateur

Dans une application en RA, on peut séparer l'interface utilisateur (aussi appelée expérience utilisateur) en deux aspects : la perception et l'interaction.

Perception

On nomme ici perception tous les aspects de l'expérience qui concernent l'information voyageant de l'application vers de l'utilisateur. Dans le cas de cette application, on a majoritairement un retour d'information sous forme visuelle. Il faut noter aussi que le visiocasque peut émettre de l'audio. Cependant cette capacité est à relativiser au regard de l'environnement bruyant d'une cour à bois et de la cabine d'un engin de machinerie lourde.

Dans les aspects visuels de l'interface, on peut distinguer encore deux types principaux d'ob-

jets : les objets fixes dans le monde réel, et ceux qui sont fixes dans le référentiel du visiocasque. Cette dichotomie n'est cependant pas absolue et il existe des possibilités intermédiaires. Ainsi on peut avoir de l'information orientée de façon absolue par rapport au monde réel, mais dont la position suit celle de l'utilisateur.

En règle générale, les objets placés dans le monde réel vont fournir une information circonstancielle ainsi que positionnelle. Dans le cas de la cour à bois, on peut imaginer désigner un paquet de bois particulier. Les capacités d'affichage stéréoscopique du casque permettent alors à l'opératrice de localiser de façon naturelle le paquet de bois dans l'espace réel.

À l'inverse, en plaçant de l'information fixe dans le champ de vision, elle est alors en permanence à disposition de l'opératrice. Ce type d'affichage est donc plus adapté pour afficher des paramètres qui doivent être régulièrement consultés et qui doivent donc être constamment accessibles dans le champ de vision. On peut penser à une horloge, des indicateurs d'état (batterie, connectivité, vitesse), l'avancement de la tâche en cours, etc.

On mentionnait précédemment la solution hybride consistant à placer des objets fixés en rotation dans des repères différents de celui du visiocasque tout en suivant en position ce dernier. Il est ainsi possible d'utiliser les axes du référentiel du véhicule afin de créer un cockpit virtuel, en fixant des objets virtuels par rapport à l'intérieur de la cabine du chargeur. Cette configuration est particulièrement utile pour afficher des informations pertinentes en tout temps, mais qui sont trop volumineuses et gênantes pour être dans le champ de vision en permanence. Par exemple une liste complète de tâches, ou le détail d'un chargement, ou encore une carte de la cour à bois. En utilisant les axes du repère monde on peut aussi fournir un compas augmenté, pour aider à l'identification directionnelle de points d'intérêt.

Interaction

L'autre facette majeure de l'expérience en RA est la façon dont on interagit avec le système. Cet aspect est hautement dépendant des périphériques d'entrée proposés par le visiocasque ainsi que le reste du système. En effet, l'écosystème actuel en RA n'est pas standardisé et chaque fabricant tente de se démarquer en proposant des modes d'interaction originaux. Contrairement à la partie affichage d'un système de RA, il n'est donc pas possible de généraliser le modèle d'interaction.

Méthodes d'entrée Le visiocasque HoloLens propose trois éléments d'interaction intégrés : dans un premier lieu, l'orientation de la tête permet à l'utilisateur de pointer une cible. Ensuite l'HoloLens est capable de reconnaître un geste particulier de la main et de le décliner ensuite en clic, maintien, ou manipulation. De plus un autre geste est défini pour l'usage

propre du système d'exploitation du visiocasque. Enfin, l'HoloLens possède un système de reconnaissance vocale, permettant à l'utilisateur d'effectuer des actions logicielles sans avoir besoin d'utiliser ses mains.

Le casque est en outre capable de se connecter à des périphériques Bluetooth, ce qui décuple les possibilités quant aux méthodes d'entrée plus classiques (clavier, joystick, souris, etc.).

Interactions avec l'interface La combinaison la plus utile de ces entrées est celle du pointage par le regard avec le geste de clic. Rien qu'à partir de cette brique simple, on a déjà accès à un grand nombre de possibilités : en effet, l'utilisateur peut pointer vers n'importe quel objet virtuel, voire réel, et utiliser le clic pour déclencher une action contextuelle. On peut de plus découpler le nombre d'actions disponibles par ce clic en proposant un menu de sélection pour différentes options en réponse au geste initial. Cependant, il faut aussi prendre en compte le fait que l'ajout d'une deuxième étape rend plus fastidieux l'accès à toutes les options. On se trouve alors face à un compromis entre être capable de fournir un accès rapide à une seule action, ou bien compliquer l'accès à une multitude de possibilités.

Il est cependant possible d'accéder au meilleur des deux mondes en ajoutant le geste de clic et maintien dans le modèle d'interaction. On peut alors affecter l'action la plus couramment utilisée dans un contexte particulier au clic simple, afin d'en simplifier l'utilisation, et le reste des options restent accessibles avec le geste alternatif.

Avec cette solution, il ne faut cependant pas négliger le fait qu'on reste limité quant au nombre d'options accessibles par la plage de mouvements de tête confortables pour l'opératrice. De plus, plus la cible à pointer (pour effectuer une sélection) est petite, plus la tâche de pointage devient ardue pour un humain. On se référera à la loi de Fitts, comme indiqué dans la sous-section 2.6.2.

Dans cette situation, si le nombre d'options est suffisamment réduit, on pourrait alternativement imaginer d'utiliser le dernier geste reconnu par l'HoloLens : la manipulation, qui permet de détecter un déplacement de la main de l'utilisateur après un clic et maintien. En effet, en plus de son usage premier dédié à modifier la position d'un objet sur de courtes distances, on peut s'intéresser à la direction du geste, et ainsi facilement discriminer deux ou plus options, en gardant cependant à l'esprit que l'on reste limité par la précision angulaire dont est capable la main de l'utilisateur.

Enfin, dans une interface suffisamment complexe, on a régulièrement besoin d'entrer des informations textuelles dans le système. Par défaut, le système d'exploitation de l'HoloLens propose un clavier virtuel utilisable en visant les touches avec un réticule et en les sélectionnant une par une avec le geste de clic. Ce mode d'interaction illustre parfaitement le

problème mentionné ci-dessus, en effet ce clavier virtuel possède des touches relativement petites, il demande une assez grande concentration sur les mouvements de la tête pour être utilisé. La «frappe» avec ce clavier est extrêmement fastidieuse. On a mentionné précédemment la reconnaissance vocale ainsi que les claviers sans-fil, qui sont des solutions beaucoup plus adaptées à ce problème. La reconnaissance vocale a l'avantage (non négligeable dans un véhicule) de ne pas demander de mouvement de la part de l'utilisateur. Cependant la technologie souffre encore d'un manque de précision et est notoire pour être biaisée en faveur de locuteurs mâles anglophones (Rodger et Pendharkar, 2004; Carty, 2011; Tatman, 2016). À l'inverse, un clavier ne peut pas être utilisé pendant la conduite d'un véhicule, mais garantit en général une entrée correcte des informations.

3.2 Expérimentation

On a évoqué dans la sous-section 3.1.2 le besoin de connaître les limites des capacités du visiocasque HoloLens dans les conditions de notre étude. En effet, on cherche à utiliser l’HoloLens dans des conditions pour lesquelles il n’a pas été spécifiquement conçu. On peut se baser sur les spécifications fournies par Microsoft pour juger de la réponse de ce matériel à nos besoins, cependant certaines interrogations persistent. Ainsi dans cette partie nous testerons le comportement de l’HoloLens dans des situations simulant les situations qui pourront être rencontrées en utilisation réelle :

- Comment se comporte le système de cartographie tridimensionnelle ? En particulier face à des surfaces semi-transparentes comme une vitre ou un grillage ?
- Le visiocasque est-il toujours utilisable dans un environnement ensoleillé ?
- Est-il capable de déterminer sa pose à l’intérieur d’un véhicule ? Et si oui, par rapport à quel repère ?
- Est-il capable de déterminer son orientation à l’intérieur d’un véhicule ? Et si oui, par rapport à quel repère ?

Des réponses à ces questions découlent plusieurs décisions de conception pour le reste du système, en particulier pour ce qui est du système de détermination de pose.

3.2.1 Cartographie tridimensionnelle

On s’intéresse dans un premier lieu aux possibilités de cartographie tridimensionnelle de l’HoloLens promises par Microsoft. La documentation fournie indique que la caméra de profondeur possède une portée comprise entre 0,8 m et 3,1 m (Yoon Park et al., 2018). Cependant on souhaite comparer ces chiffres avec le comportement en conditions réelles du visiocasque. De plus on veut savoir comment l’environnement affecte la cartographie tridimensionnelle.

Protocole

Pour toute la durée de l’expérience, un observateur porte le visiocasque sur la tête pour collecter les résultats. L’HoloLens enregistre des espaces qui contiennent des informations de cartographie 3D. Afin d’éviter toute interférence, on prendra soin de conduire les expériences dans des zones inconnues de l’HoloLens ou de supprimer les données existantes.

Pour le premier test, on se place dans une zone où seul le sol lisse est à portée théorique de la caméra de profondeur. À partir de cette position de départ, deux caractéristiques vont pouvoir être vérifiées :

- comment le système de positionnement de l’HoloLens se comporte en l’absence de repères pour la caméra de profondeur (ou autrement dit, est-ce que celle-ci joue un rôle vital dans le positionnement)
- est-ce que les capacités de cartographie tridimensionnelles mesurées correspondent aux valeurs théoriques.

Ensuite, pour le deuxième test, on se place à proximité d’une zone cartographiable derrière une vitre. On souhaite vérifier comment la présence de verre affecte les mesures de la caméra de profondeur. On espère déterminer si cette dernière répond à l’un des comportements suivants :

- le verre est considéré comme transparent par le capteur et n’est pas détecté ;
- la caméra est au contraire capable de détecter le verre et les vitres sont alors opaques du point de vue de la cartographie tridimensionnelle ;
- ou bien la vitre est détectée de façon intermittente.

L’observation de ces comportement se fera par l’intermédiaire de l’observation de la génération du maillage de cartographie tridimensionnelle. Enfin, dans un troisième temps, on réalise le même test que précédemment en remplaçant la paroi en verre par un grillage.

3.2.2 Affichage en plein soleil

Dans ce projet, l’HoloLens devra être utilisé depuis la cabine d’un véhicule de machinerie lourde en regardant vers l’extérieur. Or, de par sa nature, son système d’affichage est en compétition directe avec l’intensité lumineuse émanant de l’environnement. On souhaite donc vérifier que le visiocasque reste utilisable dans les conditions de luminosité les plus hostiles.

Protocole

On choisit un jour d’ensoleillement maximal pour réaliser l’expérience afin d’obtenir un environnement représentatif des pires conditions pour l’Hololens. On prépare une scène virtuelle contenant des objets de couleurs et intensités lumineuses variées. En particulier, on s’intéresse aux deux extrêmes : noir et blanc.

On a un test à réaliser :

- observer le contraste obtenu entre des objets noirs et blancs dans un environnement d’ensoleillement maximal.

3.2.3 Suivi complet dans un véhicule

Un des objectifs majeurs de cette étude est d'être capable de fournir un environnement de RA dans un véhicule, potentiellement en mouvement. Pour ce faire, il serait idéal de pouvoir s'appuyer sur le système de localisation de l'HoloLens pour construire l'expérience.

On cherche donc à évaluer les performances du système de localisation intégré au visiocasque à l'intérieur d'un véhicule. Trois hypothèses se démarquent :

- l'HoloLens est capable de se positionner avec fiabilité par rapport à l'environnement extérieur. Il s'agirait du cas idéal.
- Le visiocasque obtient une position fiable par rapport à l'intérieur du véhicule. Dans ce cas, la position et l'orientation du véhicule doivent être obtenues par des moyens externes.
- Enfin, le système n'est pas capable d'obtenir une position ou bien alterne entre les référentiels du monde et du véhicule, ce qui rend les données fournies inexploitables.

Pour ce test, on reproduit les conditions proches de celles rencontrées dans la cabine d'un chargeur en se plaçant à l'intérieur d'un véhicule de tourisme. En effet, dans les deux cas, l'HoloLens est confronté à un décor intérieur proche et fixe, de nombreuses surfaces vitrées, un paysage en arrière-plan dans un référentiel différent que le premier plan, ainsi que la qualité non-galiléenne du référentiel du véhicule.

On propose trois expériences pour analyser le comportement du visiocasque dans des situations de complexité croissante.

- On commence le test à l'arrêt. On laisse le temps au système de repérage de s'initialiser. On observe la qualité de la localisation en se basant sur les critères exposés à la sous-section 2.6.3.
- Pour l'étape suivante, le véhicule est mis en mouvement rectiligne. En se référant aux hypothèses ci-dessus, on vérifie à quel référentiel la scène virtuelle reste attachée : celui du monde ou celui du véhicule, ou bien encore si le suivi en position devient instable.
- On ajoute des rotations dans le déplacement du véhicule. On veut alors observer simultanément ce qu'il advient du suivi en position ainsi que du suivi en rotation.

3.2.4 Suivi en rotation dans un véhicule

L'HoloLens fournit une interface de positionnement alternative à la pose complète : il est possible de ne recevoir que des informations d'orientation, en gardant le repère centré sur le visiocasque. Ce mode est intéressant pour réaliser des applications qui ont uniquement besoin

de placer des objets virtuels dans le repère de l'utilisateur, mais aussi lorsqu'on possède une mesure de position plus fiable que celle fournie par l'HoloLens. Ainsi, si le test de suivi complet se révèle être un échec, on pourrait toujours espérer utiliser la mesure d'orientation fournie par le visiocasque.

Une fois encore, on s'attend à l'un des trois résultats suivants :

- l'HoloLens est capable de s'orienter avec fiabilité par rapport à l'environnement extérieur. Il s'agirait du cas idéal.
- Le visiocasque obtient une mesure d'orientation fiable par rapport à l'intérieur du véhicule. Dans ce cas, l'orientation du véhicule doit être obtenue par des moyens externes.
- Enfin, le système n'est pas capable d'obtenir une orientation stable. Le référentiel fourni est alors inutilisable.

De la même façon que pour le test de suivi complet, on utilise un véhicule de tourisme pour simuler l'environnement à l'intérieur d'un chargeur à fourche.

On propose d'observer le comportement du visiocasque dans les cas suivants :

- Le véhicule est à l'arrêt. On démarre le visiocasque en lui donnant le temps nécessaire pour son initialisation. On vérifie que le repère est fixé en rotation.
- On fait avancer le véhicule sur une trajectoire rectiligne. On confirme encore que le repère reste stable en rotation.
- On modifie la trajectoire du véhicule pour inclure des virages. On vérifie alors si le repère suit les rotations du référentiel du véhicule ou bien reste fixé sur le référentiel du monde.

3.3 Conclusion

Pour récapituler, dans ce chapitre, nous avons décrit les éléments constituant le système que nous souhaitons développer. Puis nous avons présenté le protocole expérimental pour les quatre tests que nous voulons effectuer sur le visiocasque HoloLens.

CHAPITRE 4 RÉSULTATS

Dans ce chapitre, on s'appliquera tout d'abord à vérifier les diverses capacités du visiocasque HoloLens qui vont influer sur le développement du reste du projet. On présentera ainsi les résultats des expériences décrites dans la section 3.2. Ensuite, dans une deuxième partie, on présentera le prototype que nous avons réalisé pour mettre en pratique les éléments du système décrits dans la partie 3.1.

4.1 Vérification des capacités de l'HoloLens

4.1.1 Cartographie tridimensionnelle

On constate dans un premier temps en balayant en rotation la scène, que l'HoloLens génère un maillage circulaire au sol tout autour de sa position, à une distance correspondant à celle annoncée par la documentation. De plus on constate que le système de positionnement du visiocasque ne semble pas perturbé par l'absence de repères dans la cartographie tridimensionnelle. Sans pouvoir conclure que ces données ne sont pas utilisées pour la détermination de la position du visiocasque, on peut tout de même affirmer que ce capteur n'est pas nécessaire de ce point de vue ci.

Lorsqu'on se place à proximité d'une vitre et qu'on observe la génération du maillage de cartographie, on constate que la vitre n'est absolument pas détectée et est donc considérée comme absente par l'HoloLens.

Si on répète l'expérience en regardant cette fois-ci à travers un grillage, les résultats sont plus mitigés. En effet, si le grillage est peu dense, il sera considéré comme transparent en l'HoloLens cartographiera les surfaces situées derrière. Par contre lorsqu'on a un grillage plus dense, la cartographie devient perturbée.

4.1.2 Affichage en plein soleil

Dès le démarrage de l'HoloLens à l'extérieur, on constate que le faible contraste rend difficile la navigation dans les menus afin de démarrer notre application de test. En plein soleil, on observe que seuls les objets blancs ou de couleur très claire peuvent être distingués, et ce même en poussant la luminosité du visiocasque à son réglage maximum.

En effet, avec la technologie d'écran transmissif qu'utilise l'HoloLens, la couleur noire est représentée par l'absence d'émission de lumière par le dispositif d'affichage, tandis que la

couleur blanche correspond à la luminosité maximale (Il en découle par ailleurs que des objets noirs ou transparents sont fonctionnellement identiques lorsque superposés au monde réel). Cependant lorsque la luminosité ambiante dépasse grandement celle que peut fournir l'HoloLens, le contraste s'effondre, car la quantité de lumière que l'HoloLens peut envoyer vers l'œil devient négligeable.

On constate une amélioration de la lisibilité pour les objets placés devant un arrière-plan situé à l'ombre dans le monde réel comme expliqué sur la figure 4.1. L'œil recevant moins de luminosité de cette direction précise, l'HoloLens peut alors reprendre le dessus au niveau du contraste.

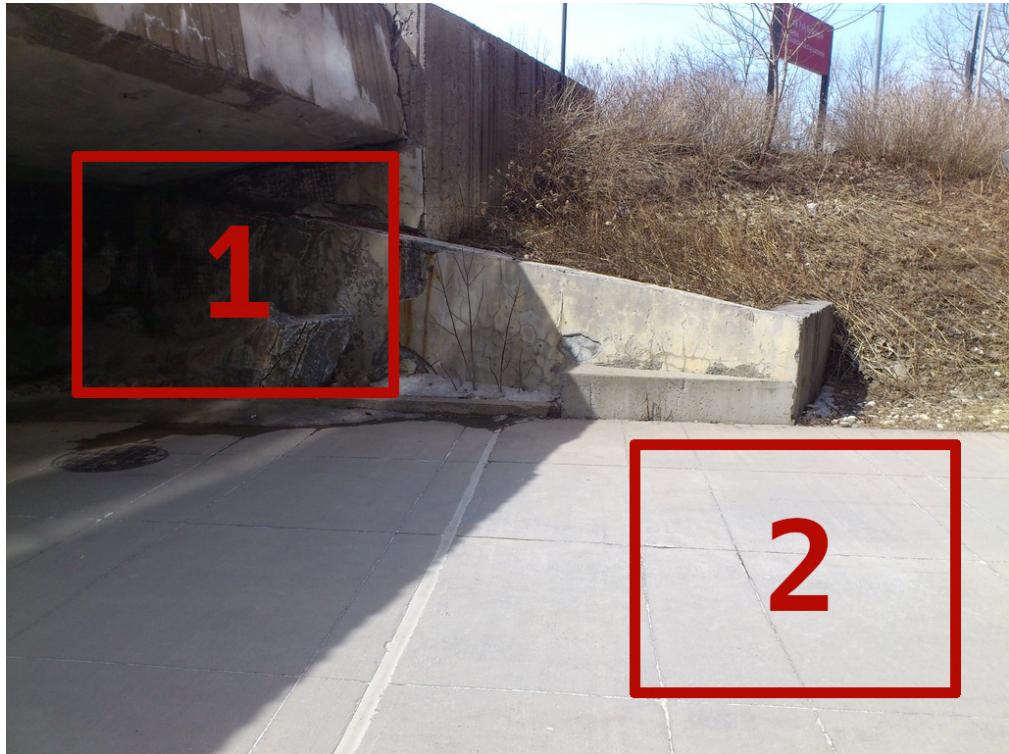


Figure 4.1 Avec cette scène en arrière-plan, les objets virtuels seraient visibles dans la zone 1 mais quasiment invisibles dans la zone 2, en plein soleil.

Il est important de noter aussi que ce phénomène de perte de contraste n'est pas visible sur le système de capture vidéo en RA interne de l'HoloLens. En conséquence, il est impossible de juger de la viabilité de ce visiocasque dans un environnement en se basant uniquement sur une vidéo embarquée.

4.1.3 Suivi complet dans un véhicule

On commence par initialiser le visiocasque à l'intérieur du véhicule. Le suivi est correct, ce qui n'est pas étonnant puisque la situation est proche du mode d'utilisation idéal de l'HoloLens. On constate cependant que la qualité de détection des gestes est diminuée par rapport à une utilisation dans des conditions normales. La détection des gestes échoue environ une fois sur deux.

À partir de cet état initial, on met le véhicule en mouvement à basse vitesse (5–10 km/h), en suivant une ligne droite. Pendant cette phase, le repère virtuel généré par l'HoloLens reste lié au référentiel du véhicule. On constate déjà cependant un peu de *dérive* et de *sautillement* de la part des objets virtuels, de l'ordre de quelques centimètres quant à leur position à l'intérieur du véhicule.

On ajoute ensuite des virages dans la trajectoire du véhicule. Dès qu'on entre dans cette phase des tests, le suivi devient très instable, le référentiel alternant entre celui lié au véhicule et celui lié au monde, résultant en une *dérive* et un *sautillement* allant jusqu'à plusieurs mètres. Ces erreurs de positionnement vont jusqu'à régulièrement désactiver le fonctionnement normal des applications le temps de trouver de nouveaux repères.

On constate de plus que si on reprend une trajectoire rectiligne après coup, le visiocasque semble alors privilégier le référentiel monde contre le référentiel du véhicule.

4.1.4 Suivi en rotation dans un véhicule

À la suite du test précédent, on vérifie maintenant les capacités du système de suivi en rotation seul. Lorsqu'on met le véhicule en mouvement rectiligne uniforme, le repère virtuel reste bien aligné avec les axes du référentiel réel. De plus, on n'observe pas du tout le glissement et les vibrations mentionnées à l'expérience précédente, en effet, le repère est fixé en position au visiocasque lui-même.

Lorsque le véhicule effectue des virages, le repère reste généralement aligné avec les axes du référentiel monde. Cependant, on a observé que dans le cas d'un virage effectué à moyenne vitesse (50–70 km/h) en conservant le même rayon de courbure pendant suffisamment long-temps, l'HoloLens semble finir par «perdre» sa référence vers le haut et l'ensemble de la scène commence à pencher, une *dérive* angulaire de plusieurs dizaines de degrés se développe et va en s'aggravant. Le problème se résorbe lorsqu'on revient à une trajectoire rectiligne, mais l'orientation initiale du repère n'est pas conservée.

4.2 Prototype

Afin de démontrer la faisabilité des concepts discutés à la section 3.1, un prototype a été réalisé. Il met en pratique les aspects de réseautique, d'interaction ainsi que les aspects visuels de l'application. Cependant, la difficulté que nous avons rencontrée à accéder au matériel haute performance nécessaire à l'implémentation de notre solution de positionnement en extérieur, ainsi qu'à l'interfaçage avec un chargeur à fourche, nous a empêché d'aller plus loin que la phase de conception pour ces objectifs.

4.2.1 Interface en RA

Plus tôt, nous avions annoncé l'objectif (O3) visant à développer une interface en RA pour le système. Dans cette sous-section, on s'attellera à décrire tous les éléments que nous avons mis en place pour réaliser cette interface.

Modèle d'interaction

Dans le cadre du développement de notre prototype, comme mentionné ci-dessus, nous n'avons pas accès à un véhicule de machinerie lourde ni à un système de positionnement d'extérieur. En conséquence, toutes les actions normalement réalisées par le chargeur sur de vrais paquets de bois doivent être simulées.

Afin de simuler les actions normalement réalisées par le chargeur, on définit un ensemble d'actions simples modélisant la manutention dans la cour :

- Ramasser un paquet
- Transporter un paquet
- Déposer un paquet

De plus, il faut aussi modéliser la dynamique de la cour à bois fini. En l'occurrence cette dernière peut être réduite à trois types d'entités :

- des entrées, qui modélisent chacune la sortie d'une des usines de rabotage ;
- des sorties, qui modélisent un camion ou wagon en cours de chargement ;
- et enfin des zones de stockage pour les paquets, zones que l'on appellera *rangées*.

À partir de ces éléments, on peut définir un exemple simplifié de scénario d'usage.

- Un nouveau paquet est créé dans la zone d'entrée.
- L'opératrice ramasse le paquet.

- L'opératrice se déplace avec le paquet jusqu'à proximité d'une rangée.
- L'opératrice dépose le paquet dans la rangée.
- L'opératrice se déplace vers une autre rangée.
- L'opératrice ramasse un paquet dans cette rangée.
- L'opératrice transporte le paquet jusqu'à la zone de sortie correspondante.
- L'opératrice place le paquet dans la zone de sortie.

Ce scénario est intéressant en première approche, mais il laisse de côté certains aspects importants de la réalité du travail de manutention dans la cour à bois. En effet, les opérateurs ne se contentent pas simplement de déplacer des blocs comme bon leur semble, la cour possède une organisation propre. On a mentionné par exemple dans la sous-section 1.1.1 que le remplissage et le vidage des rangées devaient répondre à une logique précise pour éviter l'oubli de vieux paquets au fond des rangées.

La manutention dans la cour de bois fini est donc séparée en deux activités : on a d'un côté la prise en charge des paquets neufs produits par les usines de rabotage, qui viennent compléter les rangées en cours de remplissage, et on a de l'autre des véhicules de livraison qui prennent leur chargement dans les rangées en cours de vidage.

De plus, les rangées sont classées selon le type de paquet qu'elles contiennent. Ainsi, dans une rangée on ne devrait trouver qu'un seul type de produit, défini par ses dimensions, son essence, son grade, etc. En particulier, les dimensions des paquets revêtent une importance majeure par rapport à l'empilement des paquets dans une rangée.

On cherche donc à définir un modèle d'interaction qui prend aussi en compte ces contraintes pour reproduire avec le plus de fidélité possible la réalité du travail des opérateurs et ainsi contextualiser l'aide fournie par notre application.

En fonction de l'autonomie que l'on souhaite donner aux opérateurs, on a développé deux approches. La première, qui fournit une assistance légère à des opérateurs possédant une grande autonomie se contente de réagir aux actions de ces derniers. Ainsi en se référant au scénario ci-dessus, si l'opératrice décide de ramasser un paquet au rabotage, seulement alors l'application limitera les rangées accessibles à celles étant en cours de remplissage et pouvant accepter un paquet du type de celui ramassé. De même, une fois que l'opératrice a déposé son paquet, c'est toutes les rangées en cours de vidage qui deviennent accessibles. Dans cette optique d'autonomie, dans cette approche on fournit aussi à l'opératrice la possibilité d'ajouter elle-même de nouvelles rangées si nécessaire. Le prototype fournit seulement des tests d'intégrité simples.

Au contraire, la deuxième approche qu'on propose transfère beaucoup plus de contrôle vers une autorité centrale, dans notre cas le serveur du système d'inventaire. Ici, les deux activités de manutention — remplissage et vidage des rangées — sont complètement séparées et implémentées comme deux modes distincts, chacun avec deux états :

- remplissage des rangées :
 - L'opératrice ne transporte aucun paquet de bois, seules les zones contenant des paquets à la sortie des usines de rabotage sont actives.
 - L'opératrice transporte un paquet, seules les rangées en cours de remplissage du type correspondant sont actives.
- vidage des rangées :
 - L'opératrice ne transporte aucun paquet, les seules rangées actives sont celles dont le type de paquet correspond à celui qui doit être chargé sur un des camions de livraison.
 - L'opératrice transporte un paquet, seule la zone de livraison correspondante est alors active.

Le terme de rangées «actives» est volontairement vague, on précisera dans la sous-section suivante les différentes manières qu'on a de signifier cet état à travers l'interface en RA.

Dans le cadre d'un prototype purement virtuel, on se heurte additionnellement à l'absence de lois physiques intrinsèques. Ainsi, une implémentation naïve d'une rangée permettrait à l'utilisateur de déposer un paquet à n'importe quelle position dans la rangée, et ce même lorsqu'il aurait été physiquement impossible à un chargeur d'atteindre une telle position de dépôt. Une amélioration du modèle qui a donc été implémentée a été d'ajouter une logique de contrôle lors de la prise et du dépôt des paquets. Les règles suivantes ont été définies :

- Seul le paquet situé sur le dessus d'une pile est accessible.
- Les piles de paquets ne peuvent dépasser la hauteur maximale théorique d'accès par un chargeur, telle que définie sur la figure 4.2
- Les paquets masqués par d'autres paquets plus proches de l'avant de la rangée ne sont pas accessibles.

Ces règles ne sont que le minimum nécessaire afin d'obtenir un environnement physiquement cohérent au regard de cet aspect de l'application.

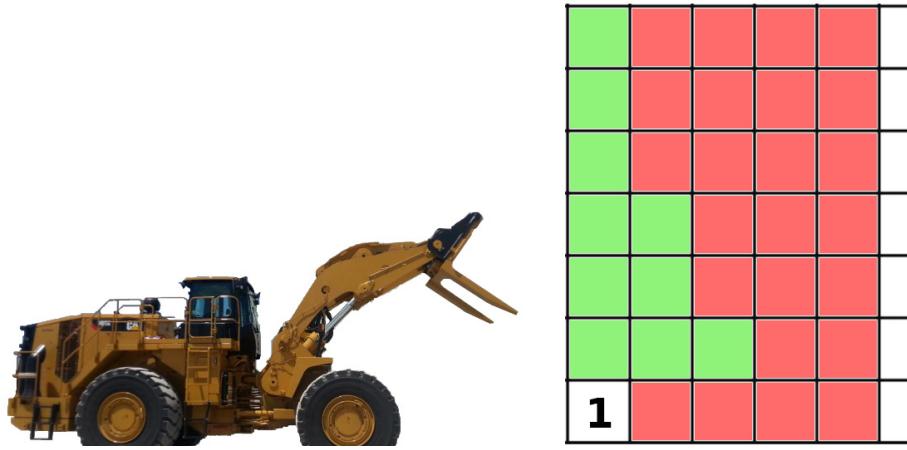


Figure 4.2 Vue de profil d'une rangée schématisée. Lorsque le paquet (1) est présent, le chargeur peut ramasser tous les paquets soulignés en vert, pour autant qu'ils soient sur le haut de la pile et accessibles.

Langage visuel

On souhaite ici décrire un langage visuel pour notre application dont le but est de fournir une compréhension intuitive de l'interface à l'utilisateur. Le concept «d'expérience intuitive» étant subjectif et propre à chaque utilisateur, on ne peut qu'essayer d'améliorer cet aspect. Cependant on peut largement s'appuyer sur les concepts explicités dans la sous-section 2.6.2, comme l'affordance ou le respect des normes culturelles locales.

Interactions avec l'environnement Sur la Figure 4.3, on observe comment l'application signifie à l'utilisateur que l'objet qu'il pointe à ce moment précis (par l'intermédiaire de l'alignement avec le curseur) a un intérêt : l'objet prend une teinte différente de celle des autres objets, afin de souligner son statut spécial. Lors du processus d'apprentissage par l'utilisateur, ce changement d'état a pour but d'attirer son attention et si possible de le pousser à effectuer une action. Il s'agit du principe d'affordance qu'on décrit dans la sous-section 2.6.2. En effet, comme l'utilisateur n'a à sa disposition qu'un nombre très restreint d'actions comprises par le visiocasque (*Tap, Maintien, Manipulation*), assez rapidement la bonne action sera utilisée, en l'occurrence : *Tap*, qui est d'ailleurs l'action la plus simple à réaliser.

Par ailleurs, si on a une situation où les éléments d'intérêt risquent d'être compliqués à localiser pour l'utilisateur, on se référera à la sous-section 2.6.2, et en particulier au paragraphe sur

le travail de Schwerdtfeger et Klinker concernant le guidage du regard en réalité augmentée.

Pour implémenter l'indicateur, on effectue un lancer de rayon depuis la caméra (ou plutôt le point entre les deux caméras) et on teste l'intersection avec les paquets actuellement ramassables. Lorsque l'opérateur ne transporte pas de paquet, l'action *Tap* effectuée tout en

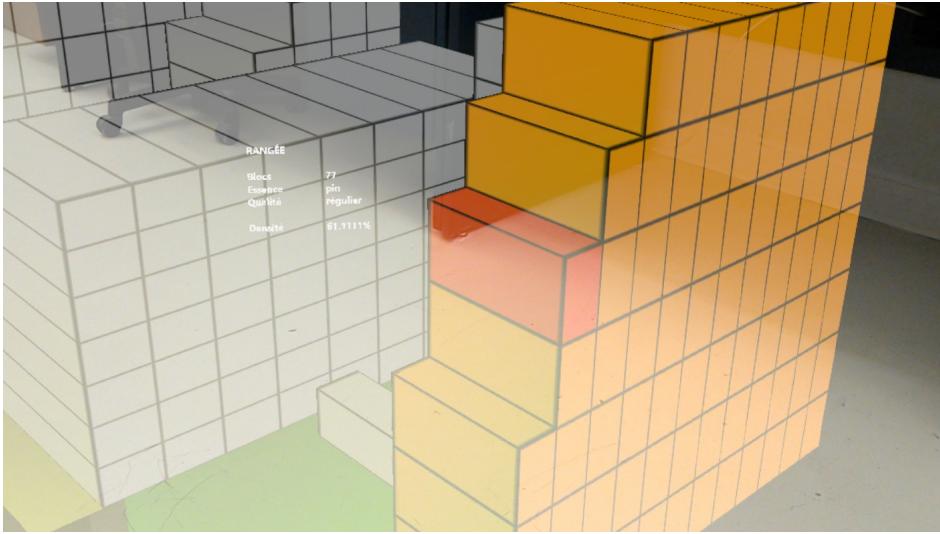


Figure 4.3 Un paquet ramassable dans la rangée active est mis en valeur lorsque le regard de l'utilisateur se porte dessus.

pointant un paquet dans une rangée active ou dans la zone d'entrée déclenche le ramassage du paquet. Comme on le voit sur la Figure 4.4, le paquet est alors «ramassé» par l'utilisateur et stocké légèrement sous le centre du regard (et donc sous le curseur), à une distance minimale de 2 m, pour ne pas empêcher l'utilisateur de voir ce qu'il vise.

On constate aussi, toujours sur la figure 4.4, qu'une fois qu'on transporte un paquet de bois les rangées ne réagissent plus de la même façon au pointage par l'utilisateur. Une prévisualisation en mode «fil de fer» est maintenant affichée pour signifier l'emplacement hypothétique que pourrait prendre le paquet que l'on transporte dans la rangée.

Cependant, dans certains cas, l'emplacement pointé par l'utilisateur n'est pas un endroit de dépôt normal. En règle générale, on se contente de ne rien afficher pour signifier l'absence de possibilité d'interaction. Mais on fait une exception dans un cas particulier : lorsque l'utilisateur cible une pile de paquets trop haute. En effet, comme on l'observe sur la figure 4.5, tout semble indiquer qu'on devrait pouvoir déposer un paquet à cet endroit précis. Pour cette raison, on ajoute un plan rouge, couleur d'interdiction, pour signifier l'impossibilité d'interagir.

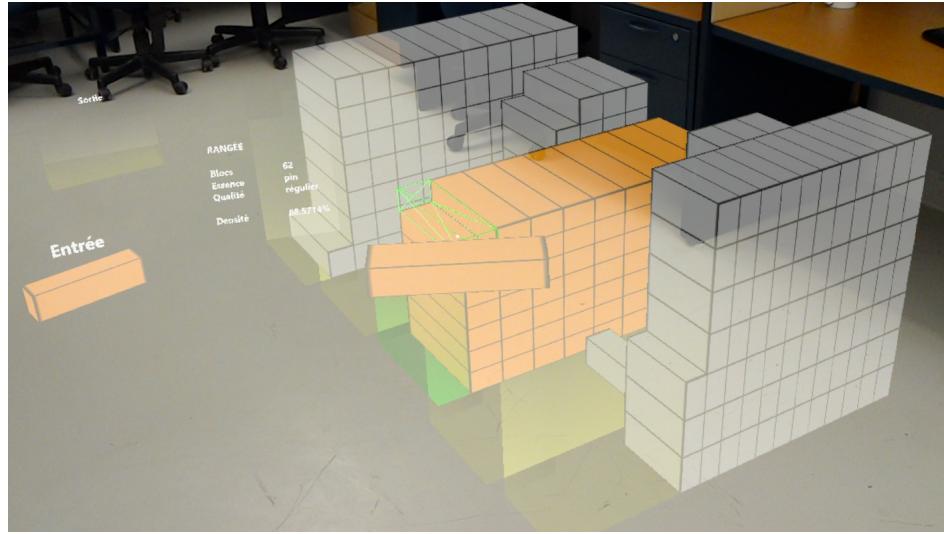


Figure 4.4 Le paquet ramassé est placé sous le curseur pour ne pas gêner le pointage par l'utilisateur. On voit aussi le paquet de prévisualisation en «fil de fer».

Curseur On a mentionné plus haut l'existence d'un curseur. La fonction première de ce dernier est de donner un repère à l'utilisateur afin de l'aider pour le processus de visée (par exemple pour effectuer une sélection). Le curseur est donc par définition en tout temps au milieu du champ de vision de l'utilisateur. En conséquence, on choisit un curseur de petite taille, blanc. On n'occupe donc qu'une faible portion du champ de vision, et la couleur blanche n'attire pas l'attention de façon exagérée tout en étant la couleur la plus résistante à la perte de contraste dans l'afficheur du visiocasque.

Cependant, en vertu de sa position au centre du champ de vision et donc au centre de l'attention, le curseur peut aussi servir lui-même à fournir des informations à l'utilisateur. Étant situé au centre du champ de vision, on ne peut ajouter qu'une faible quantité d'information avant de surcharger l'attention de l'utilisateur. Ainsi, une fonctionnalité intéressante est l'indication de détection de la main : lorsque l'utilisateur place une de ses mains en position pour effectuer un geste reconnu par l'HoloLens, le curseur change d'aspect. Nous avons choisi d'utiliser le même symbolisme que l'interface native : le curseur devient un cercle, comme montré sur la figure 4.6a. On participe ainsi à conserver une interface cohérente et prévisible, comme le conseillent les principes de la sous-section 2.6.1. Le choix de montrer cette information sur le curseur est d'autant plus pertinent que cette indication sera justement utile lorsqu'on cherche à sélectionner un élément de l'interface.

Un autre type d'information qu'il est pertinent de placer dans le champ de vision sont les problèmes survenant dans l'application, voir dans le reste du système. On ne peut évidemment pas afficher un long message d'erreur explicatif, mais on peut utiliser des icônes uni-

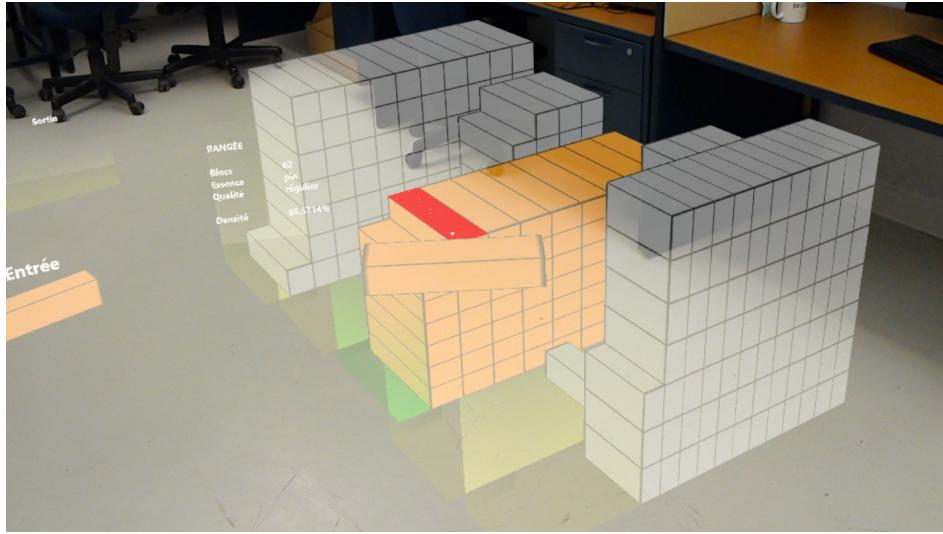
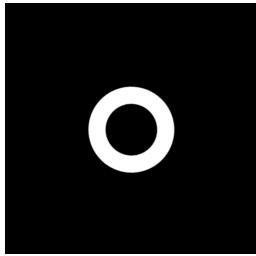
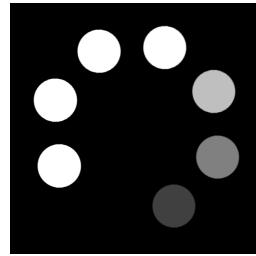


Figure 4.5 Indication visuelle que la pile visée est pleine.

versellement comprises pour signifier qu'un problème mérite l'attention de l'utilisateur. Par exemple, dans notre cas il est important d'avertir l'utilisateur d'une perte de connexion au réseau, même si le problème n'est pas bloquant (on discute à la sous-section 4.2.3 de la robustesse de notre application face aux pertes de connexion). Nous faisons donc le choix d'afficher l'icône présentée sur la figure 4.6c à proximité du curseur. On ne remplace pas le curseur, car cela pourrait être interprété comme une perte de fonctionnalité complète, le curseur étant l'intermédiaire au travers duquel l'utilisateur interagit. Le symbole réutilise l'iconographie bien connue des quarts de cercles concentriques pour signifier qu'on s'intéresse au réseau, tandis que le point d'exclamation et la couleur rouge indiquent un problème.



(a) Curseur indiquant que la main de l'utilisateur a été détectée.



(b) Curseur indiquant qu'une ressource est en cours de chargement.



(c) Indicateur de perte de connexion ajouté à côté du curseur

Enfin, il existe aussi des cas où l'on veut signifier que l'interaction est temporairement indisponible. Un exemple notable concerne les temps de chargement. En effet, lorsque l'application est en train de travailler pour préparer l'interface, par exemple suite à un changement de mode de fonctionnement, il faut signifier à l'utilisateur qu'une tentative d'interaction sera

inutile. Pour communiquer cette information, on remplace cette fois-ci le curseur lui-même. À sa place, on affiche le symbole montré sur la figure 4.6b, en rotation autour de son centre. On se base ici sur des normes culturelles pour faire passer le message, la symbolique du cercle en rotation pour signifier un chargement ayant été largement popularisée par les sites de vidéos en ligne ainsi que les interfaces de téléphones intelligents.

Menu Jusqu'ici, on a seulement décrit la communication du système vers l'utilisateur et les interactions de l'utilisateur avec l'environnement. Cependant, dans l'état actuel des choses, l'utilisateur n'a aucun moyen d'interagir avec le système lui-même. On a ainsi mentionné au début de la sous-section que le système pouvait être amené à fonctionner dans deux modes différents. Dans notre prototype, on veut fournir la possibilité à l'utilisateur de basculer entre ces deux modes. De même, si d'autres options sont ajoutées, on veut pouvoir y accéder interactivement.

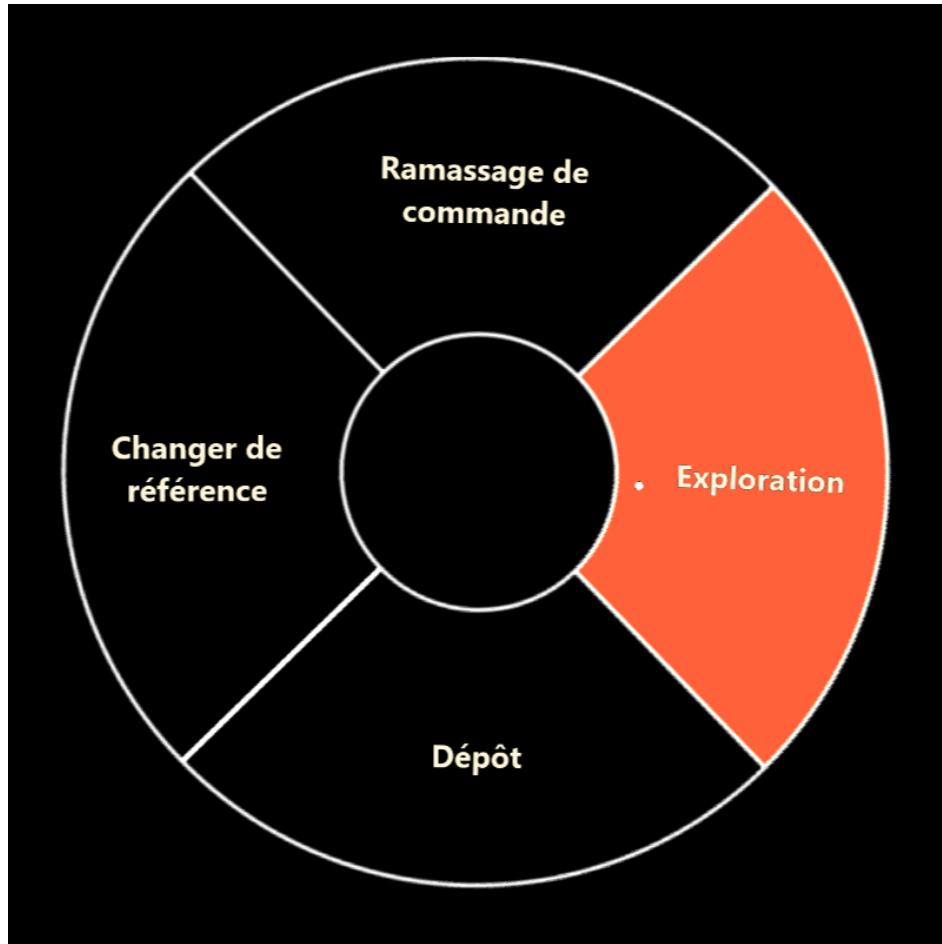


Figure 4.6 Menu circulaire montrant quatre options. On voit que l'option « exploration » est sélectionnée grâce à l'arrière-plan coloré. Le cercle central permet d'annuler la sélection

Afin de répondre à ce besoin, nous avons développé un menu circulaire (figure 4.6) donnant accès à l'utilisateur aux options décrites précédemment. Le choix d'un menu circulaire a été principalement guidé par la loi de Fitts (sous-section 2.6.2) et le fait qu'un visiocasque reste un dispositif de pointage relativement médiocre, en particulier par rapport à une souris. En conséquence, les menus sous forme de listes tel qu'on les rencontre souvent dans une interface d'ordinateur sont peu adaptés, car difficiles à pointer avec un visiocasque. À l'inverse, le menu circulaire, popularisé par les jeux vidéos où il permet une sélection extrêmement rapide, base sa sélection sur autour d'un cercle. La cible à atteindre pour chaque option représente donc un quadrant avec une ouverture d'un certain angle. La taille de la cible est donc petite près du centre, mais sa taille angulaire dans le champ de vision de l'utilisateur tend vers la valeur de l'intervalle angulaire occupé par cette option, ce qui pour un menu non dégénéré représente une taille confortable.

L'apparition du menu est déclenchée par le geste *Maintien*. Le menu est alors créé centré dans l'axe du curseur. Il est placé dans le référentiel positionnel du visiocasque afin de suivre le mouvement de l'utilisateur, par contre sa rotation est fixée dans le monde, puisque c'est justement la rotation du visiocasque qui va permettre à l'utilisateur de pointer une option. Pour signifier qu'une option est pointée, on la met en valeur en ajoutant une couleur vive en fond du quadrant correspondant. Une fois que l'utilisateur a pointé l'option qui lui convient, la sélection se fait simplement en relâchant le geste *Maintien*.

On définit aussi une zone centrale sans effet afin de permettre à l'utilisateur d'annuler sa sélection. On évite des sélections intempestives en cas d'ouverture intempestive du menu, la zone centrale étant la zone sélectionnée par défaut.

Informations textuelles Parfois, une simple icône n'est pas suffisante pour communiquer toute l'information dont a besoin l'utilisateur. On doit alors exprimer l'information de façon explicite sous forme de texte. On s'appuie pour cela sur les principes de communication exposés dans la sous-section 2.6.1. On doit alors s'attarder sur le choix de la langue utilisée afin de s'adapter aux utilisateurs potentiels. Dans le cas de notre prototype, le public visé était uniquement francophone, mais pour un déploiement dans l'ensemble du Canada, la possibilité de changer entre la langue française et anglaise serait indispensable.

On distingue trois types de messages qu'on peut souhaiter afficher :

1. *Informations globales de faible importance* Ce sont des informations concernant l'ensemble du système et pas un élément en particulier, et dont la connaissance ne demande pas à l'utilisateur une réponse immédiate. L'exemple qui a été implémenté dans notre application prototype est celui de l'affichage des statistiques globales de la cour : nombre

de rangée et nombre total de paquets de bois. Les statistiques en question étaient affichées sur un panneau placé en bordure de la cour à bois simulée.

Ainsi, si l'utilisateur souhaite consulter l'information, elle est disponible simplement en tournant le regard vers l'emplacement du panneau. Mais autrement, ce dernier se trouve généralement hors du champ de vision de l'utilisateur, en évitant donc de le distraire.

2. *Informations globales de haute importance* À l'inverse, certaines informations concernant l'état actuel du système sont trop importantes pour être placées dans un contexte : l'information arrivera à l'utilisateur uniquement si celui-ci initie l'action de la consulter. On peut penser par exemple à des messages annonçant des problèmes dans le système. On a mentionné plus haut la possibilité d'utiliser une iconographie simple pour communiquer sur des erreurs courantes comme une perte de connexion. Mais cette méthode ne s'étend pas bien à des erreurs rares : la rareté du problème rend par définition l'utilisateur peu familier avec l'iconographie, et en plus, la conception des symboles adaptés pour tous les cas d'erreur possible, aussi obscurs soient-ils, représente un travail considérable pour un gain quasi-nul.

La solution consiste donc à afficher un message d'erreur (media beaucoup plus facile à créer pour chaque type d'erreur) fixé dans le champ de vision de l'utilisateur. De la même façon que pour les curseurs évoqués plus tôt, on ajuste la position du texte en fonction de la sévérité du problème. Ainsi, une alerte non bloquante pourra être affichée dans un coin de l'image, tandis qu'une erreur fatale empêchant le bon fonctionnement de l'application sera affiché en plein centre du champ de vision, en éliminant le curseur.

3. *Informations contextuelles* Enfin, on a des informations qui dépendent de l'action que l'utilisateur est en train de réaliser, ou de quel élément il pointe actuellement. Dans notre application prototype, on va ainsi afficher un panneau flottant contenant des informations contextuelles lorsque l'utilisateur porte son curseur vers une rangée. On affiche alors le nombre de paquets contenus, la capacité de la rangée en matière de nombre de paquets supportés, les dimensions des paquets contenus, etc.

Comme on ne fait qu'informer l'utilisateur, on ne veut pas diminuer sa capacité à se servir de l'application, pour cette raison, on fait attention à ne pas placer le texte au milieu du champ de vision mais décalé pour laisser le curseur dégagé à l'écran. On peut observer ce comportement sur la figure 4.3 : on voit que le texte est placé au-dessus et à gauche du curseur (le curseur est toujours au centre de l'écran).

Enfin, puisqu'on utilise un visiocasque stéréoscopique, la distance apparente a de l'importance. Dans le cas d'un affichage contextuel, on a donc fait attention à afficher le

panneau d'information à une distance proche de l'élément pointé afin de ne pas forcer l'utilisateur à changer d'accommodation pour lire le message.

4.2.2 Communication réseau

On présente ci-après les choix qui ont été faits pour mettre en place la communication réseau dans le prototype d'application RA que nous avons développé. On propose donc des éléments de réponse à l'objectif (O5). Il est important de noter que le travail présenté dans cette sous-section ainsi que dans la suivante a été réalisé avec l'aide inestimable de Dany Naser Addin.

Technologie

Le prototype que nous avons développé fonctionne en autonomie, sans l'infrastructure de support prévue à bord du chargeur. Les seules interfaces de communication auxquelles l'HoloLens a accès sont donc les interfaces WiFi et Bluetooth. Bien que les deux technologies soient capables de se connecter à un réseau TCP/IP, la communication Bluetooth est traditionnellement utilisée pour se connecter avec des périphériques. Par contraste, le protocole WiFi est de nos jours majoritairement utilisé pour connecter des équipements à des réseaux TCP/IP et en particulier Internet.

On veut être capable de contacter facilement des hôtes aussi bien sur un réseau local que sur Internet. En conséquence, on choisit d'utiliser le WiFi pour nos communications réseau, ce qui de plus nous permet de conserver l'interface Bluetooth totalement disponible pour connecter d'éventuels périphériques.

4.2.3 Multi-utilisateur et système d'inventaire

Cohérence du modèle

En ayant fait le choix d'utiliser un serveur central pour gérer le système d'inventaire et la communication entre les chargeurs, on doit donc se baser entièrement sur le serveur pour maintenir la cohérence du modèle logique : du point de vue du système, la vérité est sur le serveur.

On a mentionné à la sous-section 3.1.4 le besoin de s'assurer de la robustesse du système en cas de perte de connexion d'un des acteurs. Il peut ainsi facilement arriver qu'un chargeur perde sa connexion, mais continue à travailler et donc à ramasser, déplacer et déposer des paquets de bois. On a alors une disparité entre le monde réel et le modèle du système d'inventaire.

Par exemple, si l'on prend un scénario fictif où deux chargeurs (chargeur 1 et chargeur 2) travaillent à vider la même rangée :

- Chargeur (1) perd sa connexion.
- (1) ramasse un paquet à l'avant de la rangée.
- Chargeur (2) se présente ensuite pour ramasser un autre paquet.
- L'interface RA de (2) indique alors toujours l'emplacement du paquet déjà ramassé par (1).
- Ensuite, lorsque (2) ramasse le premier paquet accessible sur la pile, le serveur central peut comprendre cette action de deux façons différentes :
 - si l'instrumentation du chargeur ne détecte que la position horizontale du paquet ramassé, le système d'inventaire va alors être induit en erreur quant à quel paquet a été effectivement ramassé ;
 - si par contre le chargeur est capable de connaître la position en 3D du paquet, le serveur va être amené à penser que le chargeur a ramassé deux paquets d'un coup, voir qu'il a ramassé un paquet situé sous un autre paquet, ce qui est physiquement impossible.

À la suite de scénarios comme celui-ci, ce que l'on souhaite être capable d'obtenir, c'est une cohérence éventuelle de l'inventaire par rapport à l'état réel de la cour, par exemple lorsque le chargeur en question se reconnecte.

Pour cela nous proposons que les chargeurs communiquent au serveur par transactions horodatées de l'heure de la réalisation de l'action. Ainsi, lorsqu'un visiocasque perd sa connexion, il peut continuer à stocker les transactions correspondant à ses actions, et lorsque finalement il se reconnecte, toutes ces transactions peuvent alors être transmises au serveur.

Le serveur peut alors rejouer dans l'ordre toutes les transactions plus récentes que la première transaction envoyée par le chargeur se reconnectant, et ainsi arriver à un état cohérent.

Fils d'exécution

Lors de la définition de la partie réseautique du système, à la sous-section 3.1.4, l'un des principes majeurs qu'on a mis en avant est que la communication réseau ne devait en aucun cas affecter en mal les performances du pipeline de rendu. Or on sait que les temps de transmission des informations sur des réseaux, même locaux, sont supérieurs de plusieurs ordres de grandeur aux temps de traitement des informations au sein d'une seule machine. Dans le cas d'une application graphique, on doit en plus être capable de générer une nouvelle

image 60 fois par seconde. On a donc moins de 17 ms de temps de traitement. Il n'est pas concevable d'amputer cette durée du temps nécessaire pour effectuer les transactions réseau.

C'est pour cette raison que nous avons implémenté, avec l'aide de Dany Naser Addin, toute la logique de communication dans un fil d'exécution séparé, ce qui est par ailleurs une pratique standard en matière de développement logiciel. Dans notre application, le rôle majeur du fil de communication est de recevoir du serveur les mises à jour du modèle de la cour à bois, et de renvoyer au serveur les modifications effectuées par l'utilisateur.

Cependant, en particulier au démarrage de l'application ou lors d'un changement de mode, il peut y avoir un nombre conséquent d'objets à mettre à jour. Afin de ne pas risquer de bloquer le fil de communication pendant une trop longue durée, on ajoute donc un troisième fil d'exécution, dont la responsabilité est de mettre à jour le modèle de la cour à bois utilisé par le fil de rendu avec les données fournies par le fil de communication.

Pour être sûr que le fil d'affichage ne puisse pas être bloqué par le travail du fil de mise à jour sur la structure de données partagée, c'est le fil de rendu qui se charge d'autoriser le fil de mise à jour à travailler en débloquant un mutex. Ainsi, le fil d'affichage peut contrôler la durée qui est dédiée à la mise à jour des données pendant chaque intervalle de rendu et ainsi éviter un blocage.

4.2.4 Optimisation graphique

Dans une application en RA, l'expérience d'un utilisateur est grandement influencée par la performance du rendu. On doit donc porter une attention particulière aux différents aspects du programme pouvant contribuer à dégrader cette performance. De plus, l'HoloLens, bien qu'embarquant un processeur graphique récent, reste un terminal mobile, limité par sa batterie. En matière d'infographie, on fait donc face à des restrictions plus sévères que pour le développement d'une application pour ordinateur personnel. Dans cette partie, nous décrirons les gains de performances apportés par différentes optimisations.

Affichage du texte

Le simple fait d'afficher du texte dans une application est en fait une opération relativement complexe. En effet, la plupart des bibliothèques se chargeant de dessiner du texte correctement typographié ne sont pas fournissent pas d'interface pour dessiner directement sur le Graphics Processing Unit (GPU). Elles permettent cependant d'obtenir de nombreuses informations sur la façon d'effectuer le rendu du côté du Central Processing Unit (CPU). On distingue ensuite plusieurs manières d'effectuer le rendu tridimensionnel à partir de ces informations :

- tesseler les formes des lettres afin de pouvoir envoyer leur géométrie vers le GPU ;
- utiliser une texture d’atlas contenant un ensemble de caractères et rendre le texte sur un quadrilatère ;
- faire le rendu de texte dans une texture et envoyer cette texture vers le GPU pour un rendu direct ;
- toujours rendre le texte côté CPU mais effectuer un prétraitement en calculant un Champ de distance signé (SDF) afin de diminuer la taille de la texture finale(Green, 2007).

En raison de la disponibilité d'une librairie de rendu de texte côté CPU sur la plateforme de l'HoloLens, nous nous sommes concentrés sur les deux dernières solutions, à savoir : le rendu dans une texture, et le prétraitement en SDF de cette texture. Le fonctionnement est relativement simple et est le suivant :

1. Le texte que l'on souhaite dessiner est passé à la librairie Direct2D afin de générer une texture contenant un rendu en 2D de ce texte.
2. Ensuite la texture est chargée sur le GPU.
3. Lorsque l'on souhaite afficher ce texte dans le monde, il suffit de dessiner un quadrilatère à la position souhaitée.
4. Le programme de nuanceur utilisera alors la texture précédemment chargée en mémoire pour dessiner le texte sur le quadrilatère.

Dans le cas du prétraitement en SDF, on ajoute après le point 2 les étapes suivantes :

- On utilise le GPU pour générer un SDF à partir de la texture originale.
- La texture SDF est à son tour mise en mémoire sur le GPU, dans une texture de taille bien inférieure à celle de la texture originale. L'implémentation de Microsoft utilise une texture originale de 2048 pixels et une texture SDF de 256×256 pixels.

Il semble au premier abord que la méthode basée sur les SDF demande plus de temps d'exécution puisque le prétraitement est simplement absent de la première méthode. Cependant, la méthode SDF permet un meilleur compromis entre la qualité du texte et l'empreinte mémoire sur le GPU. Dans le cas d'un texte qui reste identique durant toute la durée de vie de l'application, le coût du prétraitement devient même négligeable.

Mais lorsque, comme dans notre cas, on souhaite avoir du texte changeant rapidement (par exemple pour afficher des quantités variables), il semble que le temps de calcul du SDF devienne problématique et impacte de façon visible la fluidité de l'application.

Profilage Pour s'en assurer, on réalise un test comparatif. Pour cela, deux versions d'une application HoloLens simplifiée n'affichant qu'un panneau de texte sont développées.

- La première utilise l'implémentation de référence de Microsoft qui dessine initialement le texte dans une texture de 2048×2048 pixels. Cette texture sert ensuite d'entrée pour l'étape de génération du SDF dans une texture de 256×256 pixels, qui sera ensuite elle-même utilisée au moment du dessin à l'écran.
- La deuxième version élimine l'étape de génération du SDF. En conséquence, pour conserver une qualité visuelle équivalente, il a été déterminé empiriquement qu'une taille de 512×512 pixels était appropriée pour la texture contenant le texte.

On utilise l'outil *Visual Studio Graphics Diagnostics* afin de capturer des images générées par chacune des versions de l'application. Cet outil est ensuite capable de mesurer le temps d'exécution des appels individuels au pilote graphique. Les résultats pour chacune des étapes du dessin de texte sont compilés dans le tableau 4.1.

Tableau 4.1 Temps d'exécution des différentes étapes du dessin de texte avec et sans SDF

Étape	Temps d'exécution (ms)	
	Texte SDF	Texte normal
Dessin du texte par D2D	0,17	0,056
Calcul du SDF	35,34	N/A
Dessin à l'écran	7,77	0,58

Ainsi, on observe que le calcul du champ de distance demande à lui seul 35,34 ms, soit environ le temps de rendu complet de deux images pour un taux de rafraîchissement de 60 images par seconde. Ce résultat seul discrédite la solution avec SDF pour n'importe quelle utilisation demandant des mises à jour plus qu'occasionnelles. Qui plus est, toujours dans le tableau 4.1, on observe aussi une différence d'un ordre de grandeur pour les temps de dessin à l'écran (7,77 ms en utilisant le champ de distance contre 0,58 ms sans). Ce résultat pourrait s'expliquer par la plus grande complexité du nuanceur de pixels de la méthode avec SDF, mais il semble aussi que la performance exécrable de l'étape de génération du SDF ait eu une influence sur le reste du pipeline graphique.

Enfin, la méthode avec SDF permet de diminuer la consommation de mémoire de texture sur le GPU car La texture contenant le texte haute résolution peut être libérée une fois le SDF généré. Cependant notre application ne fait que peu usage de la mémoire du GPU. Les gains potentiels apportés par cette méthode sur ce plan ne sont donc pas intéressants.

Dessin instancé

Contrairement à l'application finale, notre prototype est conçu pour fonctionner avec une cour à bois totalement virtuelle. Cela a comme conséquence que le prototype doit aussi fournir une représentation virtuelle de la cour et ne peut pas se contenter d'augmenter seulement les points d'intérêts.

Dans l'application, on choisit de représenter les paquets de bois comme des parallélépipèdes rectangles. En conséquence, le rendu de la cour consiste à dessiner un nombre élevé de fois la même géométrie simple.

En première approche, la solution naïve consiste à effectuer un appel de dessin à l'interface de programmation Direct3D pour chaque paquet, en prenant soin de modifier la matrice modèle entre chaque appel pour définir la position de chaque paquet.

Cependant on s'aperçoit que lorsque le nombre de paquets à dessiner augmente, le nombre d'appels au pilote graphique croît aussi linéairement. En effet, pour chaque paquet, une opération complète de dessin est effectuée par le pilote.

Pour remédier à ce problème, Direct3D 11 (de la même façon qu'OpenGL 4) expose le concept d'instance au programmeur. Cette fonctionnalité permet, en un seul appel à l'interface de programmation, de déclencher un nombre prédéfini de rendus de la même géométrie (à partir d'un ou plusieurs tampons contenant cette dernière), chaque rendu étant considéré comme une instance. Il est de plus possible de fournir d'autres tampons non pas indexés sur les différents sommets, mais sur le numéro de l'instance courante (ou un multiple de ce nombre). En pratique, ce que cela signifie, c'est qu'on peut stocker dans un tampon les positions, orientations et échelles de tous les objets de même géométrie que l'on souhaite afficher, et le rendu est ensuite fait avec un unique appel au pilote graphique.

Profilage intuitivement, il paraît raisonnable de s'attendre à ce qu'une réduction des appels au pilote graphique pour le même travail effectif améliore la performance. Afin de vérifier cette hypothèse, on propose de développer deux versions d'une application HoloLens simplifiée pour afficher uniquement plusieurs centaines de parallélépipèdes rectangles.

- La première version utilise la méthode naïve consistant à dessiner chaque objet indépendamment, avec une matrice de modélisation dédiée. On notera cependant que dû à la façon dont le rendu stéréoscopique est fait pour l'HoloLens, on dessine tout de même deux instances pour chaque objet : une pour l'œil gauche et une pour l'œil droit.
- La deuxième version utilise quant à elle la méthode de rendu instancé décrite ci-dessus afin de dessiner l'ensemble des objets en un seul appel au pilote graphique. Plutôt

que d'utiliser une matrice de modélisation pour positionner les objets dans le monde, le tampon d'instance contient les coordonnées cartésiennes de chaque objet, ainsi que leur taille et leur position angulaire. On remarque qu'on pourrait directement stocker une matrice 4×4 de modélisation pour chaque objet. Cependant, comme on peut avoir à en stocker plusieurs milliers, il peut être intéressant de réduire l'espace occupé en ne stockant que les composantes nécessaires.

On utilise une fois encore l'outil de diagnostic graphique de *Visual Studio* afin de déterminer les temps d'exécution sur le GPU de chaque appel de dessin au pilote graphique pour les deux versions de l'application de test. Les résultats de cette analyse sont présentés dans le tableau 4.2.

Tableau 4.2 Comparaison des temps d'exécution sur le GPU des méthodes de dessin instancée et naïve

Méthode	Appels de dessin au pilote				
	Nombre d'appels	Instances/appel	Durée moyenne/appel (ms)	Durée totale (ms)	Durée relative clôture 1×1
Instanciée	1	952	6,24	6,24	5%
Naïve	463	2	0,0324	15,0	28%

On confirme immédiatement que la méthode instanciée permet effectivement d'obtenir un unique appel au pilote graphique. On remarque aussi les instances utilisées par le dessin stéréoscopique lorsqu'on utilise la méthode naïve. On observe aussi que même si chaque appel au pilote pour afficher un objet ne prend qu'en moyenne 32,4 µs contre les 6,24 ms de l'appel instancié, ces 463 appels prennent finalement plus de deux fois plus temps avec un temps total d'exécution de 15,0 ms.

Si l'on pousse le raisonnement un peu plus loin, on peut déduire que les deux méthodes deviennent équivalentes en termes de performance lorsqu'on dessine environ 200 objets. Cependant cette conclusion est fragilisée par le fait que la performance de la méthode instanciée est majoritairement bornée par son nuanceur de pixels. En effet, on observe sur le tableau 4.2 qu'en éliminant l'étape du nuanceur de pixels (clôture 1×1), on accélère l'appel d'un facteur 20. Il en découle que la performance de cet appel est étroitement liée au nombre de pixels rendu, donc à l'espace occupé à l'écran par les objets. On observe aussi que le gain avec la clôture 1×1 est moins important avec la méthode naïve ce qui implique que d'autres facteurs ralentissent le rendu. On peut supposer que les changements d'état du GPU entre chaque appel en font partie.

Présentation non bloquante

Il est un principe généralement admis en infographie que les points de synchronisation entre le CPU et le GPU sont détrimentaux à la performance. En effet, lorsque cela arrive, le CPU envoie une tâche au GPU, le CPU est alors bloqué jusqu'à la réception du résultat. De l'autre côté, lorsque le GPU envoie enfin son résultat vers le CPU, il doit alors attendre que ce dernier lui envoie une autre tâche. Au contraire, lorsque les deux entités fonctionnent de manière asynchrone, il est possible de les faire travailler en parallèle sans aucun temps d'inactivité. En pratique ce n'est cependant pas toujours le cas, par exemple si la charge de travail est suffisamment légère pour que le rendu à l'écran soit fait en moins d' $\frac{1}{60}$ e de seconde (60 images par seconde étant le taux de rafraîchissement recommandé et aussi le taux de rafraîchissement des écrans (*VSync*) de l'HoloLens).

Dans le cas de l'HoloLens, la fonction `PresentUsingCurrentPrediction()` permet de déclencher la présentation à l'écran de l'image actuellement calculée sur le GPU. Elle possède deux comportements configurables : elle peut soit entendre que l'image actuelle soit terminée avant de rendre la main, ou bien elle termine immédiatement après avoir effectué l'appel de présentation au GPU. Dans un mode d'attente de la fin du rendu, cette fonction est donc un exemple de point de synchronisation entre GPU et CPU.

Profilage On propose donc vérifier l'influence de ce choix de comportement sur la performance d'une application HoloLens. On réutilise l'application ayant servi à l'expérience sur le dessin instancé afin d'avoir une charge de travail réaliste lors du test. Un on développe encore deux versions dont la seule différence est le comportement de la fonction de présentation :

1. attendre la fin du rendu de l'image courante ;
2. terminer immédiatement.

On utilise l'outil de mesure d'utilisation GPU de *Visual Studio* pour obtenir une frise chronologique du travail sur le GPU et le CPU pour chacune des versions. Les résultats apparaissent sur la figure 4.7.

On observe immédiatement que la version non bloquante (2) est capable de présenter (en rouge) une nouvelle image au même rythme que le rafraîchissement de l'écran (en bleu), tandis que la version bloquante (1) n'atteint que la moitié de cette performance, avec une présentation pour deux intervalles de rafraîchissement.

La documentation fournie par Microsoft sur le sujet est succincte, mais on peut tenter d'avancer quelques suppositions pour expliquer cet écart conséquent de performance. On sait que

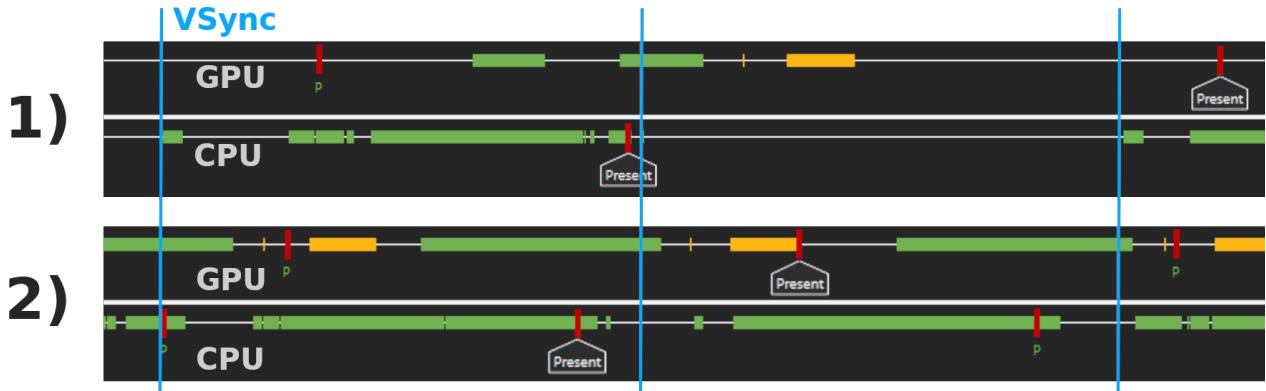


Figure 4.7 Frise chronologique du travail sur le GPU et le CPU pour les variantes avec présentation bloquante (1) ou non (2)

la fonction de présentation en (1) est censée attendre que le rendu de l'image courante soit terminée. Or on voit qu'au moment de l'appel de présentation sur le CPU, le GPU est toujours en train de travailler. Lorsqu'arrive l'intervalle de rafraîchissement, le travail sur le GPU n'est toujours pas terminé. Un peu plus tard dans cet intervalle, le GPU finit de travailler, cependant on ne voit aucune reprise d'activité du côté du CPU avant le début de l'intervalle de rafraîchissement suivant. Une fois cette limite passée, on observe une reprise du travail sur le CPU et la réalisation de l'appel de présentation sur le GPU peu de temps après. Partant de là, nous supposons qu'en mode bloquant, la fonction de présentation ne peut terminer que sur les intervalles de rafraîchissement, ce qui a pour conséquence de bloquer la présentation pour un intervalle supplémentaire si le travail GPU n'est pas terminé à temps.

En conclusion, La configuration de la fonction de présentation en mode non bloquant est vitale pour pouvoir obtenir une bonne performance avec une application non triviale.

Nuanceur de pixels

Une particularité du GPU de l'HoloLens, c'est que l'étape du nuanceur de pixels devient limitante quant à la puissance de calcul bien plus rapidement que sur le GPU d'un ordinateur de bureau. En particulier, deux facteurs liés au nuanceur de pixels peuvent avoir un impact négatif sur la performance : le nombre de pixels écrits par le nuanceur, et la complexité du nuanceur.

Profilage Pour vérifier l'impact de la complexité du nuanceur de pixels dans le cas de notre application, on a implémenté quatre nuanceurs différents servant au dessin des paquets

de bois. En effet, notre application prototype doit être capable d'afficher un nombre potentiellement élevé de paquets à l'utilisateur, ce sont donc des éléments sensibles en termes de performance. Les quatre nuanceurs sont les suivants :

- un nuanceur fournissant un aplat de couleur uni, il s'agit du code le plus simple qu'on puisse fournir à un nuanceur ;
- un nuanceur générant une texture procédurale simple (figure 4.3) ;
- un nuanceur appliquant une texture déjà chargée en mémoire ;
- un nuanceur générant une texture procédurale (une texture de bois détaillée).

Une fois encore, on mesure le temps d'exécution de l'appel d'affichage sur le GPU avec chacun des quatre nuanceurs. La mesure est réalisée avec les outils de profilage de *Visual Studio*. De la même façon qu'à l'expérience sur le dessin instancé, on mesure aussi l'accélération possible en éliminant l'étape du nuanceur de pixels (clôture 1×1). Cette mesure permet d'estimer à quel point le nuanceur de pixels représente un goulet d'étranglement dans le pipeline graphique. Le tableau 4.3 récapitule les mesures.

Tableau 4.3 Comparaison des performances de différents nuanceurs de pixels

Méthode	Appels de dessin au pilote	
	Durée de l'appel (ms)	Durée relative clôture 1×1
Couleur unie	3,94	8%
Texture procédurale simple	5,08	6%
Texture en mémoire	5,32	6%
Texture procédurale complexe	77,73	0%

On observe finalement assez peu de variation entre les trois nuanceurs simples. Le nuanceur affichant simplement une couleur, semble être le plus rapide, ce qui n'est pas surprenant, mais il n'est aussi par très utile : la couleur unie efface toute information de perspective.

Le nuanceur procédural simple exécute un nombre conséquent d'instructions par rapport à celui affichant une texture en mémoire, et pourtant leurs performances sont comparables. On peut donc raisonnablement supposer que dans le cas du GPU de l'HoloLens, l'échantillonnage d'une texture a un coût non négligeable lors de l'exécution.

Cependant, le résultat le plus notable de l'expérience est le temps d'exécution du nuanceur procédural complexe. Ce dernier comprend plusieurs centaines d'instructions et demande pour sa seule exécution quasiment cinq intervalles de rafraîchissement. On observe de plus

qu'en court-circuitant le nuanceur de pixels on améliore la performance du pipeline graphique d'un facteur 200 au minimum, ce qui confirme s'il le fallait le rôle central du nuanceur de pixels.

Ainsi, il est possible d'utiliser un nuanceur relativement complexe (avec une texture procédurale par exemple), mais seulement si cet effet n'est présent que sur une très petite partie de l'écran. Au contraire, les paquets et les rangées de bois qui peuvent facilement occuper une grande partie du champ de vision doivent avoir des nuanceurs simples (aplats de couleur, plaquage de texture échantillonnée ou procédurale simple).

Il faut aussi noter que l'ajout d'éléments transparents peut entraîner l'écriture multiple de certains pixels lorsque les éléments en question sont superposés. L'utilisation de transparence est donc un compromis entre la performance et les besoins du design d'interface.

Dans notre application, nous avons choisi d'utiliser de la transparence dans la représentation des rangées. En effet, la transparence permet de communiquer le fait que les rangées ne sont pas des objets concrets, mais des entités conceptuelles, délimitant une certaine zone de la cour. Afin de limiter l'impact en termes de performance, le dégradé de couleur est calculé dans le nuanceur de sommets et l'interpolation est faite directement dans le pipeline du GPU, le nuanceur de pixels se contente d'afficher la couleur reçue sans modification.

Enfin, lorsqu'on dessine des objets opaques, on active l'élimination des faces arrières (*backface culling*) afin une fois encore d'éviter la réécriture de pixels.

4.3 Conclusion

En résumé, lors des tests sur l'HoloLens, nous avons pu constater plusieurs faiblesses du visiocasque quant à son utilisation en extérieur dans un véhicule, son système de suivi intégré n'étant pas adapté à la tâche et ses écrans trop faibles pour garantir un affichage clair dans toutes les conditions d'ensoleillement.

Nous avons aussi présenté le prototype que nous avons développé qui implémente l'interface de RA, la communication réseau, l'intégration avec le système d'inventaire et le multi-utilisateur que nous avons décrits dans le chapitre 3.

CHAPITRE 5 DISCUSSION

Au cours de ce chapitre, nous discuterons dans une première partie des résultats des tests sur le visiocasque Hololens que nous avons présentés dans la partie 4.1 ainsi que ce que ces résultats impliquent quant au reste de l'étude. Puis, dans la deuxième partie, nous proposerons une réflexion sur les aspects du systèmes ayant été implantés dans notre prototype ainsi que sur certains sous-systèmes n'ayant pas pu l'être.

5.1 Vérification des capacités de l'HoloLens

5.1.1 Cartographie tridimensionnelle

Ainsi, les résultats présentés dans la sous-section 4.1.1 nous permettent de constater que le système de cartographie tridimensionnelle a une limite de portée très nette à environ 3 m du visiocasque. Les conditions expérimentales de ce test — un sol lisse ne présentant aucun point d'intérêt pour la caméra 3D — permettent de suggérer que le capteur de profondeur réalisant la cartographie tridimensionnelle n'est pas nécessaire à l'obtention d'une mesure de position précise du visiocasque. Il est impossible à ce stade de disqualifier ce capteur de l'ensemble des capteurs servant à fournir une information de position et d'orientation, mais les résultats obtenus pointent dans cette direction.

Lors des tests, on constate que le verre est considéré comme transparent, ainsi que les grillages fins. En effet la caméra 3D ToF opère en infrarouge. Le verre est donc probablement transparent à la longueur d'onde utilisée. Quant au grillage, on peut supposer qu'un grillage fin est à la limite de la résolution de la caméra 3D et n'est donc pas correctement détecté.

5.1.2 Affichage en plein soleil

Les résultats présentés à la sous-section 4.1.2 montrent que le visiocasque peut devenir inutilisable lorsque les conditions de luminosité deviennent défavorables en extérieur, par exemple lors d'un jour de grand ensoleillement. À cela s'ajoute le fait que les objets réels que l'on souhaite augmenter — les paquets de bois — ont un emballage blanc vif, qui est le pire arrière-plan dans des conditions d'éclairage défavorables. Ce facteur devra être pris en compte lors du design de l'interface finale, par exemple en décalant l'information au-dessus d'un arrière-plan plus sombre. Cependant on peut aussi conjecturer qu'un simple filtre diminuant la luminosité en amont des écrans du visiocasque pourrait grandement améliorer la situation pour un coût faible. On note d'ailleurs que la visière de l'HoloLens est déjà légèrement teintée, il s'agirait

simplement d'accentuer cette teinte.

5.1.3 Suivi complet dans un véhicule

On constate dans les résultats de l'expérience de suivi complet dans un véhicule, présentés dans la sous-section 4.1.3, que l'HoloLens est incapable de réconcilier les informations de l'IMU, qui fournissent une orientation fixée dans le référentiel monde ; les informations contradictoires mesurées par les caméras d'environnement qui enregistrent simultanément le paysage extérieur se déplaçant de façon cohérente par rapport aux données de l'IMU et l'intérieur du véhicule restant fixe par rapport au visiocasque ; et enfin potentiellement les informations de la caméra de profondeur. Cette dernière pourrait en effet être capable de fournir une information de position et d'orientation stable dans le référentiel du véhicule (les éléments du paysage sont hors de portée de détection), le fait que le résultat final soit aussi mauvais laisse à penser une fois de plus que la cartographie tridimensionnelle ne participe pas à la détermination de la pose du visiocasque.

On observe en outre durant l'expérience que la détection des gestes à l'intérieur d'un véhicule est dégradée. Une hypothèse pour expliquer ce phénomène est que la proximité des éléments de l'habitacle perturbe l'algorithme de détection de la main et des gestes qui utilise les données de profondeur fournie par la caméra ToF intégrée. En effet, la différence de profondeur du point de vue de la caméra entre la main de l'utilisateur et le reste de l'environnement est moins marquée que dans le cas d'une utilisation en situation idéale, telle que se tenir debout au milieu d'une pièce.

En conclusion, le système de positionnement intégré à l'HoloLens n'est pas capable de déterminer sa position et son orientation de façon stable lorsqu'utilisé dans un véhicule en mouvement. Par conséquent l'HoloLens ne peut pas être utilisé en autonomie, sans matériel externe d'acquisition de position, dans notre projet.

5.1.4 Suivi en rotation dans un véhicule

Les résultats présentés à la sous-section 4.1.4 permettent de conclure que le système de suivi d'orientation de l'HoloLens peut fournir des informations pertinentes, mais n'est pas suffisamment robuste pour être utilisé en isolation. Cependant, ce système possède deux avantages majeurs par rapport à tous ceux que l'on a pu évoquer dans la sous-section 3.1.3 :

- Étant intégré à l'intérieur même du visiocasque, la latence entre la réalisation d'une mesure d'orientation et sa disponibilité pour le système de rendu est extrêmement faible, ce qui, comme on l'a vu dans la sous-section 2.6.3, est un facteur contribuant à

la qualité de l'expérience utilisateur, une grande latence générant un effet de traînage des objets virtuels.

- Les données provenant de ce système sont profondément intégrées avec l'interface de programmation de l'HoloLens, et en particulier avec le système de rendu qui utilise ces données pour faire une prédiction de l'orientation au moment du rendu à l'écran afin encore une fois de diminuer le traînage des objets virtuels.

On notera aussi que le cas observé de perte de référence gravitationnelle causée par un virage continu est un cas extrême qui n'est normalement pas rencontré dans le cadre de la conduite d'engins de machinerie lourde, ces derniers évoluant à basse vitesse. Mais on remarque tout de même que dans le cas où le problème apparaît, il ne se résorbe pas de lui-même et demande l'intervention d'une source externe pour corriger l'erreur de cap qui résulte de l'évènement. Ce qui confirme que dans le cadre d'un système industriel qui se veut robuste, ce comportement n'est pas acceptable et demande l'utilisation de systèmes de suivi d'orientation redondants.

En conclusion, le système de suivi en rotation intégré à l'HoloLens pourra être utilisé en complément d'autres méthodes de détermination de l'orientation afin d'améliorer la qualité de l'expérience utilisateur, en particulier pour éviter des problèmes de traînage.

5.2 Réflexions sur l'application prototype développée et ses limitations

Le prototype que nous avons conçu et développé au cours de cette étude avait pour but de mettre en place certains concepts nécessaires au fonctionnement de l'application grandeur nature complète, ou application finale, qui est la version hypothétique de l'application implémentant tous les objectifs de façon concrète, avec un système de positionnement extérieur fonctionnel et une intégration complète de l'instrumentation des chargeurs. Notre application prototype avait aussi pour but de simuler certains autres composants de façon à fournir des pistes pour le futur de ces derniers, mais aussi pour créer une expérience cohérente lors de l'utilisation du prototype malgré l'absence de certains composants concrets.

On discutera donc ici des enseignements gagnés à travers le développement du prototype, des limitations de cet outil et de ces composants, mais aussi des réflexions pour faire évoluer le prototype dans la direction de l'application finale.

5.2.1 Modélisation du problème

La modélisation de la cour à bois que l'on a définie au début de l'étude est restée robuste pendant tout le développement du prototype. En effet, les éléments virtuels se portaient

extrêmement bien à une simple abstraction de contenant et contenus.

Cependant, ce modèle risque de montrer ses limites lorsqu'on l'utilisera pour décrire le monde physique : bien que la cour à bois soit effectivement organisée avec des rangées contenant des paquets, la position de ces paquets ne peu plus être simplement réduite à un numéro de pile et un étage. En effet, dans le monde réel, on doit prendre en compte des marges quant à la taille des paquets, mais aussi le fait que la dépose par une opératrice n'est jamais parfaite, et un paquet réel ne sera donc jamais aligné exactement avec la grille théorique d'une rangée du modèle.

Le modèle de représentation de la cour (qui commande aussi le schéma de stockage du système d'inventaire) doit donc être capable de stocker la position précise des paquets dans une rangée en plus de leur rang.

5.2.2 Communication

Le module de communication de notre prototype a atteint ses objectifs en matière de robustesse et d'absence d'impact sur la performance. Cependant, on a peut-être poussé l'exercice trop loin. En effet, en voulant à tout prix éviter de perturber le fil d'exécution du rendu, nous avons sévèrement limités les capacités du fil de communication.

Par conséquent, l'expérience utilisateur s'est tout de même trouvée impactée par une latence visible dans la transmission des informations entre les visiocasques lors des démonstrations avec plusieurs visiocasque en réseau. À l'avenir, il faudra donc ramener le compromis entre performance de rendu et performance de la communication un peu plus en direction de la communication.

On notera tout de même que dans l'application grandeur nature, une latence de quelques secondes lors de la mise à jour des informations n'est pas obligatoirement gênante quand on la compare aux temps typiques de déplacement dans une cour à bois relativement étendue.

5.2.3 Infographie

Tout le travail d'optimisation graphique et de détermination des forces et faiblesses du pipeline graphique de l'HoloLens est directement transposable au développement de l'application grandeur nature. Cette dernière sera même sûrement moins sollicitée en matière de rendu graphique que le prototype. En effet, l'application prototype devait dessiner un environnement complet (tous les paquets de la cour) afin de fournir du contexte à l'utilisateur. À l'inverse, dans le cas de l'utilisation dans une vraie cour à bois, l'environnement existe dans le monde réel, il est donc inutile de le dupliquer dans le monde virtuel.

On pourra donc mettre à contribution ces ressources inutilisées par le GPU pour améliorer la qualité du rendu de certains éléments, ou bien tout simplement garder ces ressources inutilisées afin de ménager la batterie du visiocasque.

5.2.4 Visiocasque

Pendant le développement de l'application prototype, l'utilisation de l'HoloLens avec son système de positionnement autonome nous a permis de faire abstraction du système de positionnement utilisé dans l'application grande nature. Par contre, comme on l'a souligné dans la sous-section 5.1, les capacités de positionnement de l'HoloLens deviennent inutilisables à l'intérieur d'un véhicule. De plus, nous ne sommes pas non plus intéressés par la cartographie tridimensionnelle que fournit l'HoloLens, car la portée de ce capteur ne dépasse guère l'extérieur de la cabine du chargeur.

La seule fonctionnalité intéressante du visiocasque HoloLens dans notre cas est donc son système de positionnement angulaire. Or de nombreux visiocasques à écrans transmissifs actuellement sur le marché possèdent aussi cette fonctionnalité. Beaucoup de ces visiocasques possèdent moins de fonctionnalités que l'HoloLens et sont donc plus simples et surtout plus légers, ce qui est un facteur majeur à prendre en compte quand l'opératrice devra porter le visiocasque à longueur de journée.

Afin de pouvoir construire sur le travail déjà accompli en termes de code, on cherchera à utiliser un visiocasque supportant Direct3D 11 afin de pouvoir réutiliser le maximum de code du pipeline graphique. Direct3D étant une interface de programmation exclusive aux produits Microsoft, on retrouvera cette compatibilité majoritairement avec les visiocasques non autonomes, pour lesquels le rendu est calculé sur un ordinateur séparé.

5.2.5 Positionnement

La détermination de la position ainsi que l'orientation du visiocasque à l'intérieur du véhicule et dans le monde étaient l'un des objectifs majeurs (O2) de notre étude. En raison de difficultés logistiques dans l'obtention du matériel nécessaire (récepteur GNSS-RTK, IMU haute qualité, lidar), nous n'avons pas pu tester en conditions réelles les propositions de conception que nous avions émises dans la sous-section 3.1.3.

Cependant, la revue de littérature poussée que nous avons réalisée nous donne un cadre solide sur lequel nous appuyer pour énoncer nos propositions de technologies à utiliser pour mettre en place de la RA dans un véhicule en extérieur.

Certaines questions restent tout de même en suspens. Par exemple il est extrêmement difficile,

sans faire de test en conditions réelles, de savoir s'il est possible d'intégrer des sources de positionnement externes à l'HoloLens (ou tout autre visiocasque) et d'obtenir une expérience en RA de qualité acceptable. Il pourra par exemple être nécessaire de développer nos propres algorithmes de prédiction afin de contrer les phénomènes de traînage créés par la latence inhérente à un système en réseau (voir la sous-section 3.1.5).

Enfin, un problème auquel nous nous sommes heurtés au cours de cette étude et qui risque d'être un frein majeur à des recherches subséquentes sur le sujet est le prix de tous les appareils dont on a besoin pour atteindre la précision requise. Ainsi la majorité des équipements recommandés dans la sous-section 3.1.3 coûtent plus de 1000\$ chacun.

5.2.6 Chargeurs

Dans notre application prototype, nous simulons le ramassage et le dépôt des paquets de bois par un chargeur en utilisant un simple *Tap*. Cette méthode est simple et est parfaitement adaptée à un environnement de test et de démonstration tel que notre prototype. Par contre cette façon de faire n'est pas viable lorsqu'on veut introduire les vrais chargeurs dans le système. Comme expliqué dans la sous-section 3.1.5, la détection du ramassage et du dépôt doit être automatique afin de ne pas ajouter à la charge mentale de l'opératrice.

Durant notre étude, nous n'avons pas eu accès à un chargeur à fourche pour pouvoir mettre en pratique et tester nos propositions d'instrumentation. À partir des données de précision présentées dans la sous-section 2.5.1, on peut tout de même garantir qu'une balance de fourche sera capable de détecter le ramassage et dépôt d'un paquet de bois par le chargeur à fourche.

Il reste cependant à établir quelle précision il est possible d'espérer quant à la détermination de la position de dépôt d'un paquet par un chargeur. En effet, on peut supposer que le paquet n'est pas toujours parfaitement en butée sur la fourche lors du ramassage, ou bien le paquet peut glisser lors du dépôt. Les erreurs sur cette mesure sont ensuite propagées au système d'inventaire puis à la représentation virtuelle des paquets.

5.2.7 Interface utilisateur

Le design de notre interface s'appuie sur des principes de conception bien établis. Cela dit, le design reste une discipline subjective. Notre approche aurait bénéficié d'une étude comparative entre différentes interfaces afin de s'assurer de choisir le design supérieur.

Pour passer du prototype à l'application grandeur nature, plusieurs aspects de l'interface doivent changer. En effet, comme on l'a mentionné plus tôt, l'environnement réel remplace

les rangées et paquets virtuels de notre prototype. De plus, les paquets ne sont plus ramassés ou déposés à travers l'interface en RA. Il faut donc modifier les comportements des différentes mises en valeur de paquets qu'on a développées. Mais on peut aussi imaginer conserver le même symbolisme, ainsi ou pourrait utiliser un contour en « fil de fer » pour indiquer l'endroit où une opératrice devrait déposer un paquet. De même, la mise en évidence colorée des paquets sous le curseur pourrait être recyclée pour permettre de sélectionner des paquets afin d'obtenir des informations supplémentaires à leur sujet.

On notera aussi qu'en fonction de la précision de placement des objets virtuels que notre système de positionnement est capable d'atteindre, on devra varier le design des éléments visuels de mise en valeur des paquets. Ainsi, si on a une grande précision (jusqu'à 10 cm), l'utilisation d'une boîte englobante est envisageable. Par contre si la précision est plus proche de 1 m, le minimum de précision qu'on considère comme acceptable, alors des éléments visuels basés sur le pointage tels que des flèches seraient plus adaptés.

Un autre changement important à noter est que l'opératrice utilisant l'application grandeur nature est assise dans la cabine d'un chargeur. Cet environnement offre un référentiel de plus que l'on peut utiliser dans notre interface : on peut introduire le concept de cockpit virtuel dans notre design. On peut placer des éléments virtuels à des endroits fixes dans la cabine du chargeur. Cette disposition est rendue possible par le fractionnement du positionnement entre la localisation du chargeur dans le monde, et celle du visiocasque à l'intérieur du chargeur.

CHAPITRE 6 CONCLUSION

6.1 Synthèse des travaux

Notre étude avait pour but de développer un système d'assistance aux opérateurs de machine-lourde dans une cour à bois. Nous avons finalement développé un prototype qui n'atteint qu'une partie des objectifs spécifiés au départ (objectifs (O1), (O3), (O5), et (O6)).

Eu égard aux objectifs (O2) et (O4), nous n'avons malheureusement pas pu les concrétiser faute d'avoir eu facilement accès au matériel adéquat, mais nous proposons tout de même une étude théorique complète de ces problèmes pouvant être utilisée pour justifier des choix d'instrumentation dans de futures études sur les technologies de positionnement d'un véhicule en extérieur à des fins de RA ou aux possibilités d'instrumentation d'un chargeur à fourche.

Les expériences conduites dans le cadre de l'objectif (O1) ont permis de mettre en lumière certaines faiblesses du visiocasque HoloLens lorsqu'utilisé dans un véhicule dans un contexte extérieur. Ainsi le système de positionnement absolu intégré de l'HoloLens n'est pas fiable lorsqu'utilisé dans un véhicule en mouvement. De plus, les écrans de ce visiocasque peuvent souffrir d'une perte de contraste handicapante dans des conditions de grand ensoleillement.

Notre étude pose donc des bases solides sur lesquelles il sera possible de s'appuyer pour développer une solution complète d'assistance en RA dans une cour à bois, ou même pour développer des systèmes de RA véhiculaires d'extérieur variés.

6.2 Travaux futurs

Dans notre interface de RA, nous nous sommes limités à fournir à l'opératrice les informations voulues par le cahier des charges initial, c'est à dire les informations individuelles pour chaque paquet de bois, reprenant le contenu de l'étiquette papier qu'on souhaitait remplacer.

Cependant, la puissance apportée par l'utilisation de la RA et d'un système d'inventaire centralisé permettrait d'aller beaucoup plus loin en matière d'assistance. Ainsi, des membres de l'industrie forestière ont émis un intérêt pour les fonctionnalités suivantes : la possibilité de fournir un guidage aux opérateurs dans la cour, en particulier pour les opérateurs en cours de formation ; la capacité d'optimiser algorithmiquement les chargements des camions de livraison des paquets de bois ainsi que l'organisation de la cour elle-même et de ses rangées ; et aussi le potentiel de pouvoir détecter des erreurs de dépôt des paquets de bois par les opérateurs.

De plus notre travail, bien que dirigé explicitement à la foresterie, développe des solutions qui pourraient être pertinentes dans beaucoup d'autres domaines de l'industrie. En effet, la possibilité d'accéder à de la RA de qualité dans un véhicule est un atout pour de nombreux secteurs de l'industrie.

6.2.1 Environnement alternatif

Dans cette sous-section, on explore ainsi quelques utilisations possibles de notre technologie dans d'autres industries, ainsi que les différences remarquables de ces environnements.

Manutention en entrepôts de plein air

Tout d'abord, notre solution est très facilement adaptable à d'autres cours d'entreposage, pour autant qu'elles soient dans une configuration où l'on est capable d'obtenir une position suffisamment précise pour pouvoir superposer correctement nos objets virtuels de RA avec le monde réel.

Il faudrait aussi adapter la logique de la gestion d'inventaire aux matériaux étant manipulés. Par exemple, notre application est basée sur le principe de ne pas enfermer un paquet de bois au fond d'une rangée, car il risquerait de trop vieillir. Mais d'autres industries ont probablement d'autres requis quant au stockage de leurs produits. Les possibilités sont presque infinies.

Industrie minière

À l'inverse de l'environnement de la cour à bois, une mine souterraine ne reçoit pas de lumière du jour. On évite alors le problème de saturation lumineuse et de perte de contraste des visiocasques transmissifs tels que l'HoloLens.

Par contre, un désavantage majeur de l'environnement souterrain est l'absence de signal GNSS. En conséquence, on basera alors notre système de positionnement sur un algorithme de SLAM avec un lidar, fusionné avec une IMU. On peut supposer que le lidar sera capable d'atteindre une bonne précision de positionnement grâce à un environnement dont les limites sont en général assez restreintes.

Assistance aux opérations de déneigement

Les conducteurs d'engins de déneigement pourraient aussi grandement profiter d'une assistance en RA indiquant la position exacte des éléments cachés sous la neige. À l'inverse d'une

utilisation dans les mines, le système est ici potentiellement confronté à une illumination forte, d'autant plus que la réverbération sur la neige accentue le problème. De la même façon qu'on l'a proposé dans la sous-section 5.1, la mise en place de filtres pour réduire l'intensité lumineuse serait recommandée.

Par contre, ce scénario a l'avantage d'avoir accès aux signaux GNSS, ce qui simplifie l'acquisition d'une position précise.

6.2.2 Contrôle à distance

Une autre idée qui a été proposée lors d'une rencontre avec des responsables d'une cour à bois Crête, serait d'utiliser notre technologie pour effectuer un contrôle distant des chargeurs dans la cour. On remarque en effet que beaucoup de l'équipement qu'on installe sur les chargeurs facilite une utilisation distante : entre le GNSS, l'instrumentation de la fourche et la mise en place d'une connectivité. Il ne reste qu'à installer des contrôles informatisés sur un chargeur et ajouter des caméras pour relayer le flux vidéo vers une opératrice distante. Il serait même possible pour les opératrices d'utiliser des visiocasques de RV afin d'embarquer dans un cockpit virtuel immersif.

RÉFÉRENCES

- A. F. Kheiralla, Abbas Eltigani, Mohamed Musa, Ahmed Osman Elzaki, et Mohamed Hamed, “Design and Development and Calibration of an On-board Weighing System for an Industrial Wheel Loader”, dans *International Conference on Trends in Industrial and Mechanical Engineering (ICTIME'2012)*, Dubai, Mars 2012.
- A. Borrego, J. Latorre, M. Alcañiz, et R. Llorens, “Comparison of Oculus Rift and HTC Vive : Feasibility for Virtual Reality-Based Exploration, Navigation, Exergaming, and Rehabilitation”, *Games for Health Journal*, vol. 7, no. 3, pp. 151–156, Juin 2018. DOI : [10.1089/g4h.2017.0114](https://doi.org/10.1089/g4h.2017.0114)
- G. Caggianese, P. Neroni, et L. Gallo, “Natural Interaction and Wearable Augmented Reality for the Enjoyment of the Cultural Heritage in Outdoor Conditions”, dans *Augmented and Virtual Reality*, série Lecture Notes in Computer Science, L. T. De Paolis et A. Mongelli, éds. Springer International Publishing, 2014, pp. 267–282.
- S. S. Carty, “Many Cars Tone Deaf To Women’s Voices”, <https://www.autoblog.com/2011/05/31/women-voice-command-systems/>, Mai 2011, (visité le 2019-04-02).
- S. Choy, S. Bisnath, et C. Rizos, “Uncovering common misconceptions in GNSS Precise Point Positioning and its future prospect”, *GPS Solutions*, vol. 21, no. 1, pp. 13–22, Jan. 2017. DOI : [10.1007/s10291-016-0545-x](https://doi.org/10.1007/s10291-016-0545-x)
- B. Close, J. Donoghue, J. Squires, P. D. Bondi, M. Morris, W. Piekarski, B. Thomas, B. Thomas, et U. E. Au, “ARQuake : An Outdoor/Indoor Augmented Reality First Person Application”, dans *In 4th Int'l Symposium on Wearable Computers*, 2000, pp. 139–146.
- S. Colaner, “What’s Inside Microsoft’s HoloLens And How It Works”, https://www.tomshardware.com/news/microsoft-hololens-components-hpu-28nm_32546.html, Août 2016, (visité le 2018-10-25).
- B. T. FANG, “Trilateration and extension to Global Positioning System navigation”, *Journal of Guidance, Control, and Dynamics*, vol. 9, no. 6, pp. 715–717, 1986. DOI : [10.2514/3.20169](https://doi.org/10.2514/3.20169)
- P. M. Fitts, “The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement”, *Journal of experimental psychology. General*, vol. 46, no. 6, pp.

381–391, Juin 1954. DOI : 10.1037/h0055392

W. W. Gaver, “Technology affordances”, dans *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Reaching through Technology - CHI '91*. New Orleans, Louisiana, USA : ACM Press, 1991, pp. 79–84. DOI : 10.1145/108844.108856

M. R. Gholami, S. Gezici, et E. G. Strom, “TDOA Based Positioning in the Presence of Unknown Clock Skew”, *IEEE Transactions on Communications*, vol. 61, no. 6, pp. 2522–2534, Juin 2013. DOI : 10.1109/TCOMM.2013.032013.120381

G. Giorgi, P. J. Teunissen, S. Verhagen, et P. J. Buist, “Testing a new multivariate GNSS carrier phase attitude determination method for remote sensing platforms”, *Advances in Space Research*, vol. 46, no. 2, pp. 118–129, Juil. 2010. DOI : 10.1016/j.asr.2010.02.023

Gouvernement du Canada, “Les Réseaux RTK”, <https://webapp.geod.nrcan.gc.ca/geod/data-donnees/rtk.php?locale=fr>, Jan. 2018, (visité le 2018-10-31).

—, “Scieries - Open Government Portal”, <https://ouvert.canada.ca/data/fr/dataset/db48db70-8893-11e0-a1b3-6cf049291510>, Déc. 2010, (visité le 2019-03-12).

C. Green, “Improved Alpha-tested Magnification for Vector Textures and Special Effects”, dans *ACM SIGGRAPH 2007 Courses*, série SIGGRAPH '07. New York, NY, USA : ACM, 2007, pp. 9–18. DOI : 10.1145/1281500.1281665

J. J. Hindman, “Dynamic payload estimation in four wheel drive loaders”, Thèse de doctorat, University of Saskatchewan, Sep. 2008.

S. Islam, B. Ionescu, C. Gadea, et D. Ionescu, “Full-body tracking using a sensor array system and laser-based sweeps”, dans *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, Mars 2016, pp. 71–80. DOI : 10.1109/3DUI.2016.7460034

R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems”, *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mars 1960. DOI : 10.1115/1.3662552

M. Kytö, B. Ens, T. Piumsomboon, G. A. Lee, et M. Billinghurst, “Pinpointing : Precise Head- and Eye-Based Target Selection for Augmented Reality”, dans *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. Montreal QC, Canada : ACM Press, 2018, pp. 1–14. DOI : 10.1145/3173574.3173655

A. Menozzi, B. Clipp, E. Wenger, J. Heinly, E. Dunn, H. Towles, J. Frahm, et G. Welch, “Development of vision-aided navigation for a wearable outdoor augmented reality system”,

dans *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, Mai 2014, pp. 460–472. DOI : 10.1109/PLANS.2014.6851442

A. K. T. Ng, L. K. Y. Chan, et H. Y. K. Lau, “A low-cost lighthouse-based virtual reality head tracking system”, dans *2017 International Conference on 3D Immersion (IC3D)*, Déc. 2017, pp. 1–5. DOI : 10.1109/IC3D.2017.8251910

Oliver Kreylos, “Lighthouse tracking examined”, <http://doc-ok.org/?p=1478>, Mai 2016, (visité le 2019-04-04).

T. Ören et L. Yilmaz, “Quality Principles for the Ergonomics of Human-Computer Interfaces of Modeling and Simulation Software”, dans *Proceedings of SIMCHI'05*, New Orleans, Louisiana, USA, Jan. 2005, pp. 5–11.

C. Qian, H. Liu, J. Tang, Y. Chen, H. Kaartinen, A. Kukko, L. Zhu, X. Liang, L. Chen, et J. Hyppä, “An Integrated GNSS/INS/LiDAR-SLAM Positioning Method for Highly Accurate Forest Stem Mapping”, *Remote Sensing*, vol. 9, p. 3, Déc. 2016. DOI : 10.3390/rs9010003

J. Raskin, “Viewpoint : Intuitive Equals Familiar”, *Commun. ACM*, vol. 37, no. 9, pp. 17–18, Sep. 1994. DOI : 10.1145/182987.584629

D. Roberts, A. Menozzi, J. Cook, T. Sherrill, S. Snarski, P. Russler, B. Clipp, R. Karl, E. Wenger, M. Bennett, J. Mauger, W. Church, H. Towles, S. MacCabe, J. Webb, J. Lupo, J.-M. Frahm, E. Dunn, C. Leslie, et G. Welch, “Testing and evaluation of a wearable augmented reality system for natural outdoor environments”, dans *Head- and Helmet-Mounted Displays XVIII : Design and Applications*, vol. 8735, Mai 2013, p. 87350A. DOI : 10.1117/12.2015621

J. A. Rodger et P. C. Pendharkar, “A field study of the impact of gender and user’s technical experience on the performance of voice-activated medical tracking application”, *International Journal of Human-Computer Studies*, vol. 60, no. 5, pp. 529–544, Mai 2004. DOI : 10.1016/j.ijhcs.2003.09.005

B. Schwerdtfeger et G. Klinker, “Supporting order picking with Augmented Reality”, dans *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, Sep. 2008, pp. 91–94. DOI : 10.1109/ISMAR.2008.4637331

J. Tang, Y. Chen, X. Niu, L. Wang, L. Chen, J. Liu, C. Shi, et J. Hyppä, “LiDAR Scan Matching Aided Inertial Navigation System in GNSS-Denied Environments”, *Sensors (Ba-*

sel, Switzerland), vol. 15, no. 7, pp. 16 710–16 728, Juil. 2015. DOI : 10.3390/s150716710

R. Tatman, “Google’s speech recognition has a gender bias”, <https://makingnoiseandhearingthings.com/2016/07/12/googles-speech-recognition-has-a-gender-bias/>, Juil. 2016, (visité le 2019-04-02).

B. Thomas, N. Krul, B. Close, et W. Piekarski, “Usability and Playability Issues for AR-Quake”, dans *In Int’l Workshop on Entertainment Computing*, 2002, pp. 455–462.

B. H. Thomas, “Augmented Reality Visualisation Facilitating The Architectural Process”, dans *Mixed Reality In Architecture, Design And Construction*, X. Wang et M. A. Schnabel, éds. Dordrecht : Springer Netherlands, 2009, pp. 105–118.

S. Verhagen et P. J. Teunissen, “New Global Navigation Satellite System Ambiguity Resolution Method Compared to Existing Approaches”, *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 4, pp. 981–991, Juil. 2006. DOI : 10.2514/1.15905

L. Wanninger, “Introduction to Network RTK”, <http://www.wasoft.de/e/iagwg451/intro/introduction.html>, Juin 2008, (visité le 2018-10-23).

Xsens, “New mixed-reality holographic projector for construction sites - News”, <https://www.xsens.com/news/new-mixed-reality-holographic-projector-for-construction-sites/>, Juil. 2018, (visité le 2019-04-02).

M. Yassin et E. Rachid, “A survey of positioning techniques and location based services in wireless networks”, dans *2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, Fév. 2015, pp. 1–5. DOI : 10.1109/SPICES.2015.7091420

Yoon Park, Nick Schonning, Matt Zeller, rwinj, et Brandon Bray, “Spatial mapping design - Mixed Reality”, <https://docs.microsoft.com/en-us/windows/mixed-reality/spatial-mapping-design>, Mars 2018, (visité le 2019-02-01).

H. Zhang, A. Elmokashfi, et P. Mohapatra, “WiFi and Multiple Interfaces : Adequate for Virtual Reality ?” dans *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*. Singapore, Singapore : IEEE, Déc. 2018, pp. 220–227. DOI : 10.1109/PADSW.2018.8644610