Simulation et expérimentations pour l'évaluation de la navigation de robots dans la foule

Fabien Grzeskowiak

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

THESE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Info

Par
Fabien GRZESKOWIAK

Simulation et expérimentations pour l'évaluation de la navigation de robots dans la foule

Thèse présentée et soutenue à Irisa, le 08/06/2021
Unité de recherche : UMR IRISA 6074

Rapporteurs avant soutenance :

SPALANZANI Anne	Maître de conférences à l'Université de Grenoble - Alpes
SOUERES Philippe	Directeur de recherche CNRS au LAAS

Composition du Jury :

Examinateurs :	CARLSON Tom	Professeur à UCL
	MARCHAND Eric	Professeur à l'Université de Rennes 1
	SPALANZANI Anne	Maître de conférences à l'Université de Grenoble - Alpes
	SOUERES Philippe	Directeur de recherche CNRS au LAAS
Dir. de thèse :	PETTRE Julien	Directeur de recherche à l'INRIA Rennes
Co-encadrants de thèse :	BABEL Marie	Maître de conférences HDR, IRISA UMR6074, INSA Rennes

Figure 1 – Un robot parmi une foule simulée.

Le projet CrowdBot est un projet collaboratif entre 7 partenaires de 4 pays européens, financé par la Commission européenne (EU H2020). Ce projet soulève des questions d'ordre éthique et technique à la fois, dans le but d'accroître significativement la capacité des robots, autonomes ou semi-autonomes, à naviguer dans des environnements peuplés d'humains, en foule. Le projet n'est pas orienté autour d'un robot spécifique, mais cherche au contraire à dresser des conclusions d'ordre plus général. Ainsi, les solutions robotiques proposées sont expérimentées sur divers robots afin de vérifier d'une part les performances des systèmes et de valider la généricité des solutions. D'abord, nous utilisons le robot Pepper [Pandey and Gelin, 2018b], un robot humanoïde social qu'on peut retrouver dans des salons grand publics par exemple, mais qui peine encore à naviguer efficacement dans des foules denses. Un autre robot utilisé dans le projet est un fauteuil roulant semi-autonome, un système similaire à celui utilisé par K. Narayanan et al. [2016], qui s'appuie sur un algorithme de contrôle partagé du fauteuil, c'est à dire qui utilise à la fois les entrées utilisateur (pouvant provenir d'un joystick par exemple) et une composante de navigation autonome servant d'assistance au conducteur. Un autre robot semi-autonome est utilisé dans le projet : le robot Qolo présenté par Granados et al. [2018], un robot à destination des personnes à mobilité réduite leur permettant de se déplacer tout en se tenant debout, maintenu par un harnais. Enfin, le projet CrowdBot a pour objectif de permettre à un dernier type de robot, Cuybot, de naviguer dans la foule. Ce robot répondra spécifiquement

aux préconisations en matière de conception issues du projet CrowdBot. Encore au stade de développement, ces robots pourraient pourtant révolutionner la vie de leurs utilisateurs lorsqu'ils seront capables de naviguer efficacement et en sécurité dans la foule.

Dans le cadre du projet, il convient donc de développer des outils de simulation adéquats. Le but de ces outils de simulation est double :

— permettre au robot de prédire l'évolution de la foule autour de lui et d'y naviguer de manière plus sûre et plus efficace ;

— tester *in silico* les capacités de navigation du robot, ou l'entraîner à le faire dans une boucle d'apprentissage.

C'est dans ce deuxième objectif que cette thèse s'inscrit.

Table des matières

Chapitre

1 Introduction

Sommaire

1.1 Des robots parmi la foule

Les robots intègrent progressivement l'espace public, et sont amenés à circuler parmi les gens dans une très grande proximité. En 1999, Burgard et al. [1999] présentent un robot-guide qui cartographie, localise et évite ses voisins. Burgard revendique alors la définition d'"un nouveau moyen d'interaction entre l'homme et le robot" et "des robots pour fonctionner en toute sécurité, de manière fiable et à grande vitesse dans des environnements hautement dynamiques". En 2003, Siegwart et al. [2003] présentent, à l'exposition nationale suisse Expo02, Robox, un robot social autonome, qui interagit et navigue de façon autonome. On sait cependant que de telles machines peuvent être vite dépassées par la densité de personnes. En effet, elles sont rapidement confrontées à la difficulté de percevoir

Figure 1.1 – Le robot social Pepper interagissant avec des personnes

et d'interpréter la présence d'humains à partir des données issues de leurs capteurs embarqués et à la difficulté de prédire leur comportement, ainsi qu'à la difficulté de naviguer dans cet environnement complexe. Par ailleurs, ces machines ne tiennent pas compte des codes sociaux pour se déplacer afin d'adopter un comportement acceptable. Enfin, les robots peuvent souffrir du problème de "Freezing", c'est à dire qu'ils vont se figer dans la foule : les algorithmes qui guident leurs comportements saturent de toutes les contraintes que leur impose la présence d'humains en mouvement autour d'eux, la solution d'un mouvement parfaitement sûr n'existant tout simplement pas.

La robotique industrielle, confrontée à la question de la cohabitation de l'humain et du robot, est ainsi à l'origine de l'émergence des "Cobots", des robots collaboratifs qui partagent leur espace de travail avec un opérateur. Par exemple, Fiore et al. [2016] présentent en 2016 un système de supervision pour les robots collaboratifs : leur algorithme calcule d'abord une tâche spécifie pour le robot et prévoie une tâche pour l'humain, puis dans un second temps, le système de supervision contrôle le robot et surveille les actions de l'humain. Si ce dernier n'effectue pas l'action prédite, alors le système s'adapte et calcule un nouveau plan d'action.

Cependant, Javaheri et al. [2019] montrent que la perception grand public sur le concept du "robot" a radicalement changé en dix ans. Ils notent ainsi la diminution de l'importance de certains sujets, tels que l'"efficacité des robots" ou les "Robots autonomes", alors que l'essor autour des "Robots sociaux" et des "Robots de service" illustre l'évolution de l'opinion publique et des médias sur les robots, passant d'une machine à usage exclusivement industriel vers un gadget plus domestique, convivial, social et polyvalent.

Aujourd'hui, les robots intelligents sont devenus communs dans les ménages, aidant aux tâches, pouvant les soutenir dans leur quotidien, voire surveiller leur santé et leur bien-être. Les robots font désormais partie de la vie quotidienne. Pour autant, alors que l'opinion des gens évolue, la technologie doit suivre. En particulier, la robotique mobile est sujet à de grandes attentes et tente de passer à l'étape suivante : concevoir des robots se déplaçant aisément dans la foule. L'accroissement de la recherche sur cette thématique est illustré par le nombre croissant de publications sur ce sujet, comme l'illustre la figure 1.2.

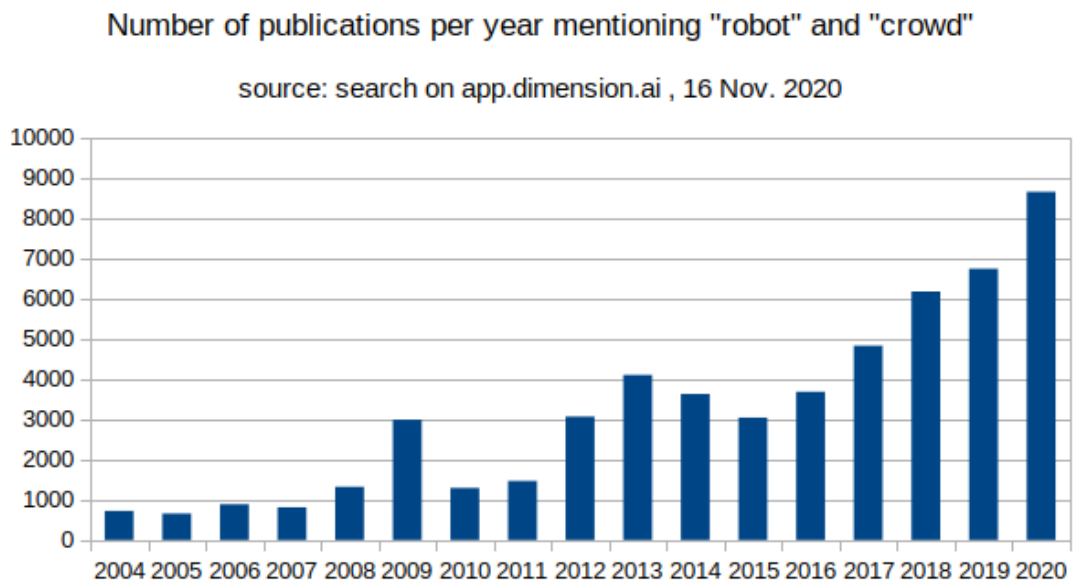


Figure 1.2 – Nombre de publications par an mentionnant les deux termes "robot" et "foule", selon le site web app.dimension.ai le 16 novembre 2020

Dans ce contexte, il est donc primordial que la robotique intègre des **modèles de comportement humains**, pour mieux capter, prédire et interpréter l'environnement du robot. Également, il est essentiel que les **métriques** qui caractérisent les performances du robot évoluent, car la notion d'efficacité dans un environnement de travail vide de vie humaine ne peut pas se transposer quand le robot évolue à très grande proximité des humains. Il convient a minima d'intégrer des notions de sécurité par rapport aux risques d'accident et de perturbations que la présence du robot peut générer. Il convient aussi de trouver les méthodes d'évaluation de ces nouveaux critères de performance. En outre, l'expérimentation incluant des sujets humains est inévitable, mais soulève des questions éthiques et logistiques difficiles. La simulation est alors un recours mais soulève toutefois la question de la modélisation du comportement humain, et du niveau de fidélité

qu'elle peut offrir. Cette thèse tente donc de répondre aux besoins de la robotique mobile en termes de **simulation** et d'**évaluation en environnement de foule**.

1.2 La robotique mobile en environnement de foule

Ici, nous abordons les différents aspects que recouvre la question de la navigation de robots dans des environnements de foule. Si cette thèse n'aborde pas ces problématiques en particulier, il nous parait indispensable de les décrire pour comprendre les pré-requis d'outils de simulations que nous cherchons à proposer dans ce travail.

Piloter un robot mobile dans un environnement inconnu nécessite de cartographier efficacement l'environnement tout en localisant le robot dans celui-ci. Ce problème généralement appelé SLAM (pour Simultaneous localization And Mapping) a été abordé tout d'abord par Sebastian Thrun *et al.* avec le robot STANLEY [Thrun et al., 2005, Thrun, 2002, Thrun et al., 2006]. Ce problème bien que complexe n'est alors exprimé que pour des situations d'environnements fixes dans le temps. Cependant, le problème de la navigation de robots mobiles en milieu inconnu reste ouvert dans les environnements dynamiques. En effet, les solveurs SLAM s'appuient sur des techniques probabilistes qui nécessitent une bonne qualité de détection de l'environnement. Par exemple, un robot entouré de personnes ne peut pas détecter correctement les murs de l'endroit où il se déplace, car ceux-ci sont occultés : la localisation est donc difficilement réalisable. Par ailleurs, si on ne prend en compte que des obstacles pseudo-statiques, c'est à dire ne bougeant pas ou peu et ne faisant pas partie de l'environnement connu du robot, il faut alors développer des méthodes d'évitement local. Par exemple, Fox et al. [1997] proposent une méthode de contrôle classique dans des environnements statiques prenant en compte la dynamique du robot, en travaillant dans l'espace de vitesse. Des techniques de planification de chemin voient alors le jour comme avec LaValle and Kuffner [2001] qui traitent de l'approche Rapidly-exploring Random Tree (RRT) pour la planification du mouvement en utilisant l'exploration rapide d'arbres aléatoires.

De plus, le problème de l'évitement local devient aussi critique : lorsque la trajectoire planifiée d'un robot entre en conflit avec un obstacle dynamique dans son voisinage, l'évitement local permet de modifier localement cette dernière pour éviter la collision et revenir dès que possible à la trajectoire planifiée. Cette situation se répétant très souvent dans le cadre d'une navigation dans une foule,

il s'agit alors d'éviter un nouveau calcul de planification globale à chaque perturbation locale. Pour y parvenir, des techniques originales utilisant les vitesses des obstacles voient le jour. Fiorini and Shiller [1998] présentent un algorithme de navigation de robot mobile utilisant des notions de "vitesses atteignables" correspondant aux caractéristiques mécaniques du robot, qui sont couplées aux "vitesses d'évitement" pour donner un ensemble de vitesses possibles. Ces méthodes nouvelles s'avèrent toutefois insuffisantes car dans une foule, les êtres humains ont des comportements complexes qui ne sont que grossièrement approximés par ces méthodes. Par exemple, on peut calculer une carte dynamique à la façon de Saarinen et al. [2013] qui décrivent une méthode de carte d'occupation en environnement dynamique (principalement pour les véhicules autonomes dans le cas de Saarinen et al. [2013] mais qui peut s'étendre à une foule) : chaque point de la carte a alors une propriété dynamique qui correspond à la vitesse et l'orientation moyenne des objets (ici des voitures) à cet endroit de la carte. Cependant, cela reste une approximation. En outre, en ce qui concerne l'évitement local d'humains par un robot dans un environnement de foule, il convient de se pencher sur le comportement de la foule dans ce contexte. On parle de comportement collectif, quand bien même ces comportements ne sont pas guidés par un but commun, mais plutôt par des motivations individuelles soumises à des contraintes locales imposées par les voisins.

Ainsi, une notion importante relative au comportement collectif de l'humain est celle de la proxémie. Elle caractérise le rapport entre l'espace qui entoure une personne et les relations interpersonnelles, et a été présentée par l'anthropologue Hall [1966] (figure 1.3). L'un des principaux concepts de la proxémie est la distance physique qui s'établit entre des personnes prises dans une interaction. Hall a remarqué que ces distances varient selon les cultures (plus faibles en Afrique qu'au Japon par exemple) et l'endroit où se déroule l'interaction (plus faibles dans un ascenseur que dans une grande rue piétonne).

Cette notion réapparaît comme concept clé pour la recherche dans le domaine des interactions humain-robot (HRI). Par exemple, Joosse et al. [2014] montrent que les préférences culturelles, notamment entre les États-Unis, la Chine, et l'Argentine, en matière de proximité entre humains sont les mêmes si l'interaction s'effectue avec un robot plutôt qu'un humain. Les pionniers de la robotique bio-inspirée travaillaient ainsi dans l'idée de créer une machine capable d'interagir avec son environnement, voire avec d'autres robots similaires. La figure 1.4 montre les "tortues", des robots construits par Walter en 1948 [Walter, 1950]. Ces "tortues" sont équipées de cellules photosensibles qui agissent directement sur le mécanisme permettant à la tortue de se déplacer. Elles sont attirées par la lumière la plus

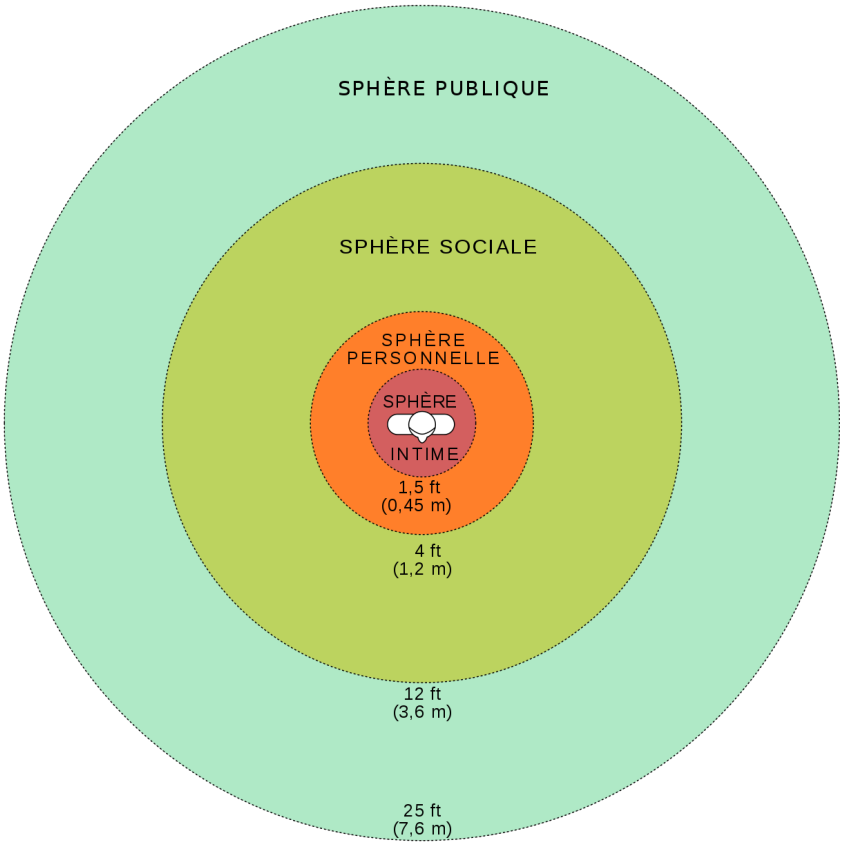


Figure 1.3 – Diagramme des sphères proxémiques chez des sujets de la classe moyenne de la côte nord-est du continent américain, d'après Hall.

intense de leur environnement. Lorsque qu'un obstacle est sur son chemin, la pression exercée par le contact de l'obstacle sur la carapace déclenche un court-circuit sur la cellule photosensible. La "tortue" se déplace alors de façon aléatoire jusqu'à ce qu'elle parvienne à se dégager de l'obstacle. Il s'agit d'un des premiers robots capable d'interagir avec son environnement. Aujourd'hui, on parle de robotique sociale pour travailler sur l'interaction entre les machines et les humains [Kruse et al., 2012] [Narayanan et al., 2015] [Paulin et al., 2018].

De nos jours, la notion de robotique sociale évoque davantage une interaction entre un robot et un humain. Dans ce cadre, Kruse et al. [2013] proposent une enquête sur la navigation *"human-aware"*, c'est à dire consciente de la présence d'humains dans l'environnement. Kruse et al. [2012] présentent un planificateur basé sur la notion de coûts sociaux, c'est à dire une formalisation mathématique de "l'inconfort" basé sur le concept de proxémie de Hall [Hall, 1966], sous la forme de champs de potentiels similaires à la figure 1.3. Le robot utilise alors les coûts sociaux pour se déplacer en interaction avec les humains, rendant le mouvement

Figure 1.4 – Précurseurs de la robotique sociale : Les tortues de Walter, Elmer et Elsie, "dansent" l'une autour de l'autre.

du robot moins gênant, et donc plus lisible par les humains. Cette méthode est ensuite évaluée par Kruse et al. [2014] au moyen de questionnaires : l'étude montre que le comportement du robot réel est similaire aux observations faites en simulation, et que les participants jugent le comportement du robot moins déroutant avec le modèle de coût sociaux. Narayanan et al. [2015] proposent une stratégie de navigation permettant à un robot de rejoindre un groupe de personnes interagissant entre elles, en tenant compte de l'espace d'interaction du groupe pour le répartir de façon équitable. Cette stratégie est utile pour les robots sociaux et d'assistance lorsqu'ils doivent rejoindre et interagir avec un groupe de personnes. Narayanan et al. [2016] montre l'efficacité de cette méthode dans un monde réel et dynamique, dans divers scénarios. Chi et al. [2016] proposent à leur tour une technique de navigation parmi les humains, basée sur la méthode RiskRRT. Paulin et al. [2018] proposent quant à eux une façon de prendre en compte vers qui ou quoi les humains portent leur attention pour éviter de les interrompre, et ainsi générer des trajectoires acceptables.

Pour valider les propositions, des expériences avec des humains et des robots sont alors nécessaires. Par exemple, Dongqing Shi et al. [2008] présentent un pro-

jet de segway intelligent capable de détecter les humains environnants et d'agir en conséquence. Ils suggèrent la mise en place d'expériences grandeur nature avec des participants, pour en déduire des contraintes de vitesses des robots garantissant la sécurité. De façon plus spécifique, Trautman and Krause [2010] proposent de résoudre le problème du "Freezing robot" en considérant que les agents dynamiques, des êtres humains, participent conjointement à l'évitement des collisions avec la machine. Leur modèle est donc moins prudent mais permet de débloquer le robot en cas de besoin. Trautman et al. [2013] améliorent ensuite son modèle pour l'adapter à une foule plus dense, puis réalisent des expériences dans une cafétéria d'un campus universitaire pour évaluer la méthode en la confrontant à la réalité d'une foule sur laquelle Trautman n'a pas le contrôle [Trautman et al., 2015]. Mavrogiannis et al. [2019] conduisent eux aussi des expériences grandeur nature avec un robot mobile et des humains, et avec un grand nombre de participants à l'expérience.

Enfin, le nombre toujours croissant d'expériences impliquant des personnes permet d'accéder à une grande quantité de données. Ces données peuvent être utilisées pour l'apprentissage machine ("machine learning") pour la robotique sociale. Chen et al. [2019] utilisent l'apprentissage par renforcement profond ("deep learning") pour modéliser les interactions entre humains et les interactions entre robots et humains dans la foule, donnant ainsi un modèle d'interaction entre la foule et le robot. Long et al. [2018] proposent un outil de "deep learning" pour les situations impliquant plusieurs robots, qui est ensuite adapté à un environnement de foule [Fan et al., 2018]. Vasquez et al. [2014] utilisent une approche d'apprentissage par renforcement inverse pour retrouver le modèle de l'interaction sociale entre les personnes d'une foule, en utilisant des composantes comme la densité, la vitesse des personnes, la proxémie. L'objectif est alors d'utiliser ensuite ce modèle d'interaction pour permettre à un robot d'anticiper les mouvements d'une foule dans un aéroport.

1.3 La simulation de robots mobiles

Les outils de simulation sont très répandus en robotique : on simule le robot et son environnement, via des objets en 3 dimensions soumis au lois de la physique par un moteur physique. Cependant, ces simulations manquent de souvent de réalisme. En effet, Ivaldi et al. [2014] ont interrogé les chercheurs en robotique sur les outils qu'ils utilisent, et leurs attentes vis à vis de ces outils pour l'avenir : leur Graal consiste en la simulation au plus proche de la réalité. A défaut d'avoir un niveau de réalisme satisfaisant pour l'évaluation, la simulation est utili-

sée pour faire de la preuve de concept. L'évaluation est quand à elle faite en milieu expérimental qui, dans le cas de la robotique mobile dans la foule, fait appel à des personnes volontaires. Par exemple, Dongqing Shi et al. [2008] et Chi et al. [2016] proposent des méthodes qui sont testées en simulation puis sont évaluées en cas réel, et qui, malgré tout, induisent un risque non nul pour les participants à l'expérience. A l'inverse, la sécurité des participants aux expériences pourrait être garantie avec une simulation de haute qualité. En fait, la question de la sécurité lors d'interactions humain-robot (HRI) est de plus en plus posée. De Santis et al. [2008] dressent ainsi un portrait de la recherche en HRI, en mettant l'accent sur les interactions physiques. Les notions clés de la discussion sont la sécurité, c'est à dire la mise en place de systèmes garantissant la sécurité des personnes environnantes, et la fiabilité, c'est à dire l'absence de défaillance des systèmes de sécurité. Plus tard, Vasic and Billard [2013] réalisent une enquête sur le problème de la sécurité en robotique, dans différents domaines comme dans l'industrie, la robotique mobile de service, ou les véhicules autonomes. Il convient donc de discuter de la sécurité des participants impliqués dans des expériences notamment avec des robots mobiles.

En effet, des expériences impliquant un grand nombre de personnes sont parfois nécessaires pour valider certains algorithmes en cas réel, mais leur mise en place peut s'avérer complexe, et il est rare de trouver un nombre conséquent de participants. Peu d'études font mention d'expérimentations à large échelle. Par exemple, Mavrogiannis et al. [2019] conduisent des expériences grandeur nature avec un robot mobile et de nombreux volontaires humains, visant à étudier les interactions entre un robot et des personnes se déplaçant a proximité, et montrent notamment que l'accélération des participants est plus faible lorsque le robot est autonome que lorsqu'il est téléopéré par une personne. Ce travail souligne à la fois le manque de qualité des outils de simulation d'une part, et la rareté des données collectées dans le cas d'expériences grandeur nature d'autre part. Idéalement, il faut donc réunir en un seul outil la capacité de fournir des données nombreuses et de garantir la fidélité de ces résultats. Afin d'atteindre cet objectif, le niveau de réalisme actuel des simulateurs est un obstacle qu'il faut franchir.

1.4 Notre approche pour la simulation d'environnements de foule pour la robotique

Cette thèse vise à proposer un nouvel outil pour l'étude d'interactions robot-foule, dont le principe est illustré par la figure 1.5. L'originalité principale de notre

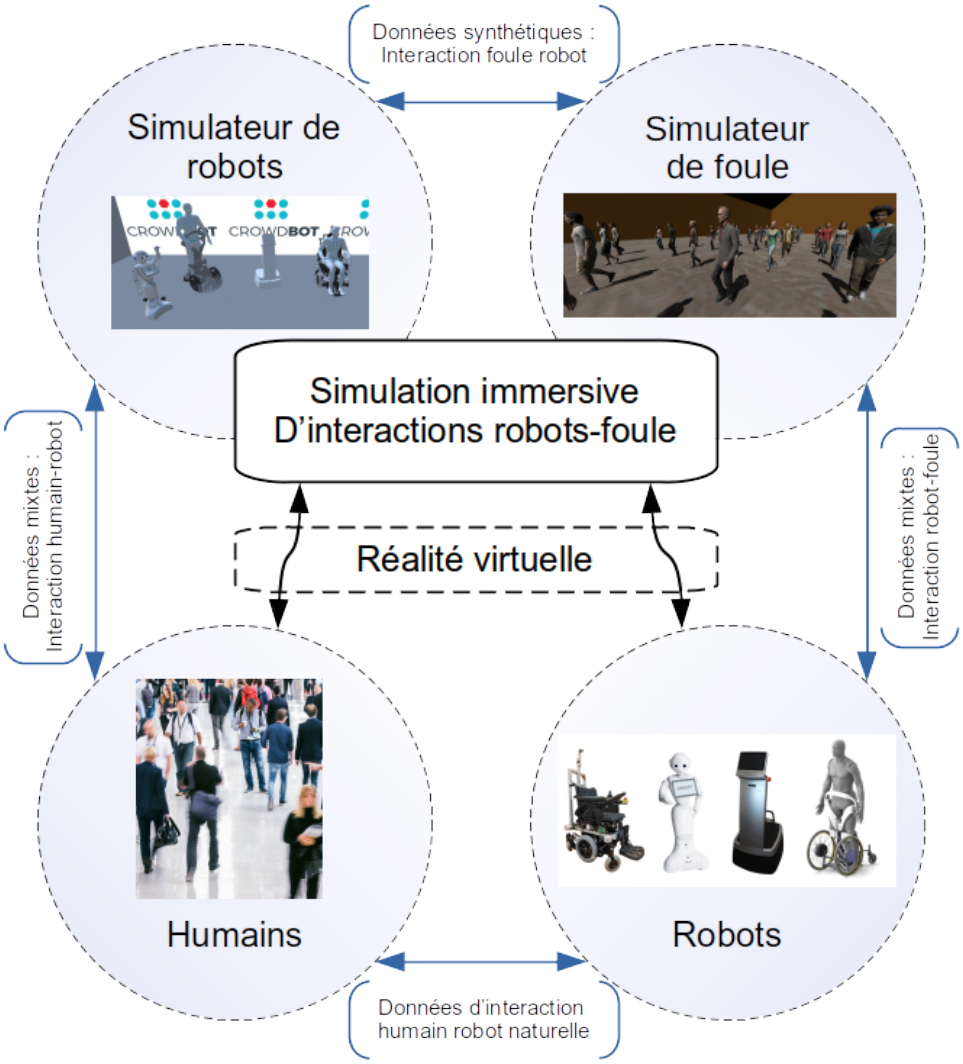


Figure 1.5 – Outils de simulation de la thèse

approche est de proposer un outil de simulation immersif : on laisse la possibilité pour un humain ou un robot de prendre part à la simulation. Par le truchement des technologies de réalité virtuelle, nous pouvons rendre cette simulation perceptible à un humain (par l'utilisation de périphériques pour le retour sensoriel, tels que des casques de réalité virtuelle pour la vue) ou un robot (par simulation des données captées). Nous pouvons enfin transposer les actions de ces humains ou de ces robots en mesurant par exemple leur mouvements dans la simulation, via l'avatar qui les incarne. Ainsi, l'outil que nous proposons repose sur 4 piliers et permet la génération de données synthétiques ou mixtes pour l'étude des interactions robot-foule :

1. **la simulation du robot**, dont le rôle est de reproduire les propriétés in-

trinsèques d'un robot : ses propriétés mécanique, son fonctionnement dynamique, ses capteurs. On utilise alors un simulateur de robot pour imiter les attributs physiques d'un robot réel ;

2. **la simulation de foule**, dont le rôle est de créer un environnement dynamique similaire à une foule autour du robot simulé. La simulation de foule permet de créer des scénarios complexes en donnant des comportements différents aux agents de la foule ;
3. **un outil d'immersion du sujet humain** qui permet une interaction à l'échelle 1, incarnée, entre un (ou plusieurs) sujet(s) humain(s) et la scène numérique qui comporte foule et robots virtuels, et qui permet l'enregistrement des comportement du sujet ;
4. **un outil d'immersion du robot** qui permet d'immerger un robot dans un environnement complexe et maîtrisé comme celui du simulateur de foule. Immerger un robot permet de garder les propriétés intrinsèques (physiques) du robot sans avoir besoin de les simuler, tout en offrant un environnement contrôlé au robot.

Nos outils de simulation s'appuient naturellement sur des travaux précédents et des standards actuels en matière de robotique mobile et de simulation. En effet, comme le soulignent Craighead et al. [2007], il n'est pas nécessaire aujourd'hui de faire un simulateur de robot à partir de rien, car des outils existent. Ils dressent ainsi une liste d'outils de simulation qui sont en fait une réutilisation d'outils commerciaux utilisés dans les jeux vidéos, mais adaptés à la simulation de robot. Notre outil de simulation s'adresse à la communauté robotique par le biais d'interfaces standard du domaine. En particulier, nous nous appuyons sur la communauté qui utilise et développe le projet open source ROS (`ros.org`), et qui propose un ensemble d'outils pour le développement de projets robotiques.

Le second domaine sur lequel notre simulateur s'appuie est celui de la simulation de foule. En effet, ce domaine s'est particulièrement développé ces dernières années, sur la même tendance que la Figure 1.2, avec une augmentation significative du nombre de publications chaque année dans le domaine. Par conséquent, la demande d'outils innovants dans le domaine de la simulation foule est forte. Notre approche consiste à utiliser des moyens d'animation et de contrôle de foule, notamment des algorithmes de navigation multi-agents, pour les intégrer dans notre simulateur, créant ainsi un outil innovant permettant la simulation conjointe d'une foule et de robots.

Enfin, notre outil se repose sur des systèmes issus du domaine de l'immersion en réalité virtuelle. En effet, afin de créer des environnements d'expérimentations

standard et sans risques, tout en restant réalistes, nous nous sommes tournés vers la réalité virtuelle. Si cette dernière a déjà fait ses preuves en matière de d'interactions entre humains, notre approche procure quant à elle le moyen de réaliser une telle interaction entre un robot et plusieurs êtres humains, ou encore entre un être humain au milieu d'une foule.

La rencontre entre ces domaines (robotique, simulation de foule, et réalité virtuelle) est une nouveauté à part entière : nous explorons dans cette thèse les nouveaux problèmes que cela soulève et les usages potentiels d'un tel rapprochement des technologies.

1.5 Contributions

Les contributions de cette thèse s'alignent sur l'objectif de développer un environnement de simulation pour l'étude d'interactions robot-foule :

1. un algorithme de simulation de foule généraliste, capable d'uniformiser les algorithmes existants et d'en reproduire les résultats, afin d'étendre le domaine de validité de la simulation. Cette contribution a fait l'objet d'une collaboration interne à l'Inria de Rennes, entre plusieurs doctorants et chercheurs travaillant sur divers sujet de simulation de foule. L'idée a été de développer un outil de simulation de foule utilisant un principe général d'optimisation de fonction de coût. Cet outil permet donc d'utiliser ou d'imiter un grand nombre de techniques de navigation de foule, et de développer ses propres fonctions de coût. Ma contribution a été de concevoir et de développer la librairie de simulation de foule.
2. l'utilisation de l'outil de simulation en tant que banc d'essai. L'idée est de fournir à la communauté robotique un cadre nouveau d'évaluation, avec des scénarios imposés mettant en scène d'une foule de densité et de comportements variables, ainsi que des robots mobiles simulés proposés aux utilisateurs.
3. l'utilisation de la Réalité Virtuelle pour l'immersion commune d'un robot et d'une personne dans un monde virtuel, pour l'étude des interactions robot-foule. En effet, nous proposons l'idée d'immerger un robot mobile réel dans un monde virtuel, afin d'y faire des études d'interactions homme-robot. Nous avons donc commencé le développement de l'outil de simulation dans l'optique de créer des expériences en réalité virtuelle. Nous avons mis en place une expérience dans le but de faire la preuve de concept, qui n'a pas révélé de biais qui pourraient remettre en cause le principe, montrant ainsi

que l'utilisation de la réalité virtuelle est adaptée à l'étude d'interaction en humains et robots.

4. la réalisation d'une expérience en Réalité Virtuelle visant à l'utilisation de brassards haptiques pour simuler des contacts en réalité virtuelle. L'idée est d'évaluer si l'utilisation de tels bracelet influe le comportement des gens lorsqu'ils se déplacent dans une foule virtuelle. J'ai mis en place les outils nécessaires à l'expérience et l'ai ensuite conduite auprès de volontaires.
5. la mise en place d'un simulateur haptique de fauteuil électrique. L'idée était de concevoir une plateforme robotique innovante restituant les sensations d'un vrai fauteuil roulant électrique, afin d'être utilisé en réalité virtuelle, réduisant ainsi le mal des simulateurs. Ce projet a été fait en collaboration avec le projet européen ADAPT, en particulier avec l'INSA de Rennes.

L'ensemble de ces contributions s'articulent toutes autour du même projet libre de droit qu'est le simulateur.

1.6 Plan de thèse

Cette thèse se présente comme suit : dans un premier temps, le chapitre 2 présente l'état de l'art dans chacun des domaines de recherche sur lesquels cette thèse s'appuie : la simulation en robotique, la simulation de foule, et la réalité virtuelle. Ensuite, le chapitre 3 décrit l'outil de simulation robotique développé pour réaliser nos expériences, qui repose sur différents outils : la simulation robotique, la simulation de foule, l'analyse d'interaction foule-robot, et la Réalité Virtuelle pour l'interaction humain-robot. Ensuite, le chapitre 4 décrit notre simulateur de robot. Puis, le chapitre 5 expose en détail notre approche de la simulation de foule. Ensuite, le chapitre 6 décrit un banc d'essai permettant l'analyse de d'interactions robot-foule, et présente des résultats générés par ce banc d'essai pour des scénarios d'exemple. Puis, le chapitre 7 aborde la question de l'utilisation de la Réalité Virtuelle pour l'immersion de robots dans un monde virtuel auprès d'êtres humains. Le chapitre 8 aborde quand à lui la question de l'utilisation d'outils haptiques pour les humains dans le contexte d'expériences en Réalité Virtuelle où les contacts (virtuels) sont inévitables. Enfin, le chapitre 9 présente les résultats d'une collaboration autour d'un simulateur mécanique de fauteuil roulant électrique utilisant la Réalité Virtuelle. Finalement, une conclusion générale et une perspective sont présentées au chapitre 10

Chapitre

2 État de l'art

Sommaire

Le chapitre 1 introduisant cette thèse indique (en particulier à travers la figure 1.5) que notre approche vis-à-vis de la simulation d'interactions robot-foule est à l'interface de trois domaines, que sont :

1. la simulation de robots mobiles dans des environnements dynamiques ;
2. la simulation de foule ;
3. la réalité virtuelle par l'immersion de sujets et de robots dans cette simulation.

Ainsi, ce chapitre se compose de trois parties couvrant chacune de ces aspects. Dans une dernière partie, nous donnons notre analyse de cet état de l'art pour situer notre approche.

2.1 Simulation de foule

La simulation de foule consiste à calculer le déplacement de nombreuses personnes, une foule, dans de but de reproduire ou de prédire des comportement de foules réelles. Dans le cadre de cette thèse, le rôle du simulateur de foule est de simuler le comportement de personnes nombreuses autour du robot. Pour simuler les données issues des capteurs du robot, il est important que la simulation permette de calculer la posture de personnages, qu'ils soient seuls ou en nombre, a minima dans la zone perçue par le robot. On distingue donc deux problématiques associées :

— la simulation du déplacement global des personnes dans l'environnement qui relève à proprement parler de la simulation de foule ;

— le calcul des postures des personnages en fonction de cette trajectoire d'animation.

Cet état de l'art porte principalement sur la première problématique, la deuxième sera réalisée avec des méthodes existantes de l'état de l'art. On distingue ainsi plusieurs type d'approches : l'approche dite macroscopique, l'approche dite microscopique, et l'approche basée données.

2.1.1 Les approches macroscopiques pour une représentation globale

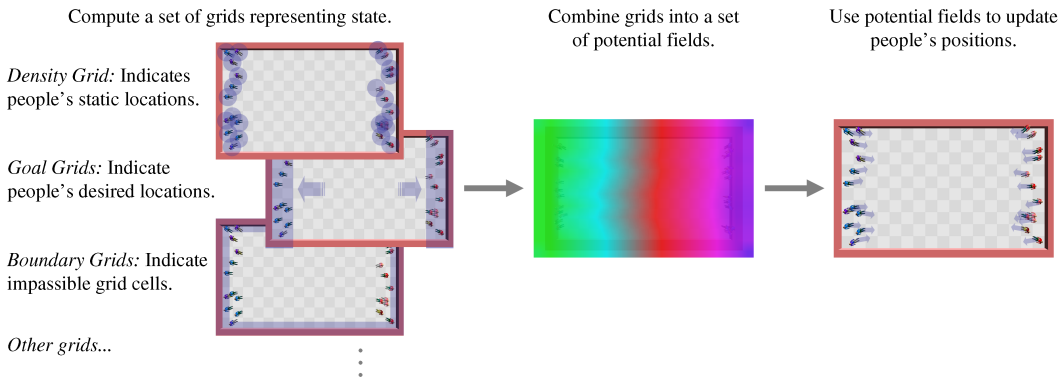


Figure 2.1 – Description de l'algorithme de Treuille et al. [2006] (image issue de l'article)

Les approches macroscopiques considèrent la foule comme un écoulement et s'intéressent tout particulièrement aux phénomènes de grande échelle, comme la formation de ligne, ou la vitesse d'écoulement selon l'environnement (goulot d'étranglement, couloir...). Les approches macroscopiques tentent quant à elle de reproduire ces phénomènes visibles à l'échelle globale en considérant que la foule se comporte comme un ensemble d'éléments entre lesquels le passage est continu, dénommé *continuum*, comme défini par Hughes [2002] (Figure 2.1), Hughes [2003], et qui répond aux lois de la physique. Plus précisément, un *continuum* est "un corps dont la matière est continuellement répartie et remplit toute la région de l'espace qu'il occupe" d'après Gan [2012]. Shimizu et al. [2003] et Pimenta et al. [2008] appliquent les lois de la dynamique des fluides pour simuler une foule : le mouvement des particules du fluide (correspondant aux piétons) est ainsi entraîné par les lois de la dynamique des fluides.

Jin et al. [2008], Chenney [2004], Treuille et al. [2006], et Patil et al. [2011], eux, définissent des champs de vitesses afin de mettre en mouvement la foule sans collision avec les obstacles présents dans l'environnement. Ces champs de vitesse sont soit créés manuellement, soit extraits d'une vidéo, soit calculés mathématiquement. Ces approches sont très intéressantes pour simuler des foules très denses en raison de leur complexité : les coûts de calcul associés dépendent en effet de la résolution de l'espace choisi, et non pas de la densité de la foule. Cependant, elles conduisent également à la création de certains artefacts tels que l'interpénétration entre la représentation visuelle des individus de la foule, qu'on appelle par la suite "agent". Ces méthodes présentent toutes une particularité : elles ne permettent pas de donner un état (position, vitesse) de chaque agent dans la foule, individuellement, mais on déplace les particules sur les champs de vitesse ou de densité. Les différentes particules-agents sont alors soumises à des paramètres globaux identiques pour tous.

Certaines approches donnent quant à elle à plus de libertés aux individus de la foule simulée. Narain et al. [2009] proposent une approche hybride : dans les régions à forte densité de population, leurs algorithmes utilisent les principes de la dynamique des fluides avec des agents représentés comme des éléments granulaires incompressibles. Pour les régions à plus faible densité, la vitesse des agents est alors obtenue en interpolant la vitesse d'un modèle de dynamique des fluides et la vitesse de l'agent préféré afin d'atteindre leur objectif. Dans le monde la robotique, Kerr and Spears [2005] utilisent la dynamique des gaz comme principe de base pour le contrôle d'un essaim de robots. Les robots de l'essaim se comportent tous de façon similaire, et ont pour objectif de couvrir un maximum de terrain. Mais dans le cas d'une foule, chaque agent a son comportement bien à lui et un but qui lui est propre. Lorsqu'il s'agit de faire naviguer un seul robot dans une foule, il convient donc de prendre en compte les individualités des agents.

2.1.2 Les méthodes basées sur des données données

Le concept sous-jacent à ces approches est de partir de données réelles, afin de non seulement d'en préserver le réalisme mais aussi de les adapter à des structures différentes (géométries d'environnements, densité). Pour cela, il faut ajuster les trajectoires des agents simulés à ces structures différentes. Il y a trois grands principes :

1. préserver les interactions : on enregistre les portions de trajectoires avec leur voisinage en créant des patchs. Un patch de foule est un élément d'animation de foule réutilisable dans une zone fixée de l'environnement de simulation,

avec des agents entrant et sortant de la zone à des moments donnés. En assemblant plusieurs patchs ensemble comme des pièces de puzzle et en faisant tourner l'animation en boucle dans chaque patch, l'illusion d'un mouvement de foule infini peut être créée ;

2. préserver les trajectoires entières, et les déformer pour les ajuster en résolvant les collision résiduelles. Pour ce faire, on déforme le plan sur lequel ces trajectoires sont définies ;

3. Préserver les caractéristiques grâce à l'apprentissage machine. On donne en entrée d'un réseau de neurones l'ensemble des données réelles. Le réseau est ensuite capable de générer des trajectoires réalistes inédites.

Ces approches ont pour intérêt de préserver le réalisme des trajectoires. Cependant, elles ont le défaut d'avoir des trajectoires figées, et n'ont pas de réaction par rapport à des éléments dynamiques nouveaux, comme un robot.

Ces méthodes proposent en fait soit de reproduire exactement un comportement réel à suivre pour une situation donnée (i.e le positionnement des agents voisins), proche d'une situation enregistrée, soit de créer des patchs de trajectoires, reliées par des transitions. Lee et al. [2007], Lerner et al. [2007] simulent ainsi des agents en cherchant une correspondance entre la situation simulée et les données enregistrées, et copient le comportement correspondant à cette situation réelle. Yersin et al. [2009] et Kim et al. [2012] assemblent des patchs de foule créés via des données réelles. Charalambous and Chrysanthou [2014] proposent un graphe de perception-action où les noeuds du graphe correspondent à des états similaire de l'agent et permettent de déduire l'action à effectuer ensuite. Lai et al. [2005] et Kovar et al. [2008] présentent des graphes utilisant des mouvements réels et des transitions auto-générées. Ju et al. [2010] proposent une méthode pour mélanger des données de foules pour créer des transitions entre différentes données réelles. En outre, d'autres méthodes proposent de simuler la foule en déformant les données réelles pour générer de nouvelles trajectoires. Kwon et al. [2008] proposent de modifier des trajectoires réelles qui conservent certaines propriétés comme le voisinage ou la trajectoire "haute fréquence", mais construisent la trajectoire sur le long terme. Kim et al. [2014] présentent une méthode de déformation des trajectoires en imposant des contraintes. Jordao et al. [2014] sculptent les patchs de foule pour créer les mouvements souhaités en étirant l'environnement.

Enfin, avec l'essor de l'intelligence artificielle, les données de trajectoires sont utilisées pour alimenter des réseaux de neurones artificiels. Gupta et al. [2018], Amirian et al. [2019a,b] définissent ainsi des réseaux de neurones artificiels génératifs (GAN). Utilisant des trajectoires réelles, ils apprennent à leur réseau à

générer de nouvelles trajectoires crédibles, mais entièrement synthétiques. Zhong et al. [2016] définissent deux réseaux : l'un apprenant les comportements locaux, l'autre pour les comportements à long terme. Ces réseaux neuronaux sont entraînés sur des vidéos. Zhao et al. [2020] entraînent également 2 réseaux avec des données de foules. Enfin, Zhao et al. [2013] utilisent un réseau de neurones classifiant pour créer des *clusters* (i.e. des catégories) de comportement de foule.

Finalement, ces méthodes utilisant des données sont très utiles dans le domaine de l'animation, car elle proposent des méthodes qui permettent d'obtenir des rendus de foule dense pour un temps de calcul réduit. Cependant, ce type de méthodes peut souffrir du manque de données disponibles, et manquent de diversité dans les comportements des agents.

2.2 La simulation de foule microscopique : évitement local entre agents distincts

Les approches microscopiques se distingue des approches macroscopiques et des approches basées sur des données par l'usage d'un raisonnement centré sur les *agents*. Le principe est de définir une loi de contrôle par agent, permettant d'aller à l'objectif tout en évitant les obstacles locaux. Contrairement aux approches macroscopiques, les approches microscopiques visent à faire émerger, par des calculs de trajectoires individuelles, des comportement de foule. C'est-à-dire qu'à partir de la seule modélisation individuelle du comportement, on cherche à déduire le comportement collectif. En particulier, on cherche à reproduire certains *motifs* de comportement de foule, comme la formation de files dans des flux bidirectionnels, ou de circulation en accordéon dans des flux unidirectionnels.

Les approches microscopiques peuvent se décomposer en plusieurs sous-catégories : les méthodes utilisant des forces, les méthodes utilisant des contraintes de vitesse, et les méthodes utilisant une vision synthétique comme moyen principal de perception de l'environnement.

2.2.1 Évitement par calcul de force

Helbing and Molnár [1995] (SFM) proposent un des premiers modèles de simulation microscopique de foule. Faisant l'analogie avec les lois de l'électromagnétique, chaque agent de la foule est une particule soumise à des forces : une force d'attraction vers le but, et des forces de répulsion par les obstacles environnants et autres agents, dépendant des positions et vitesses relatives des agents. Cette

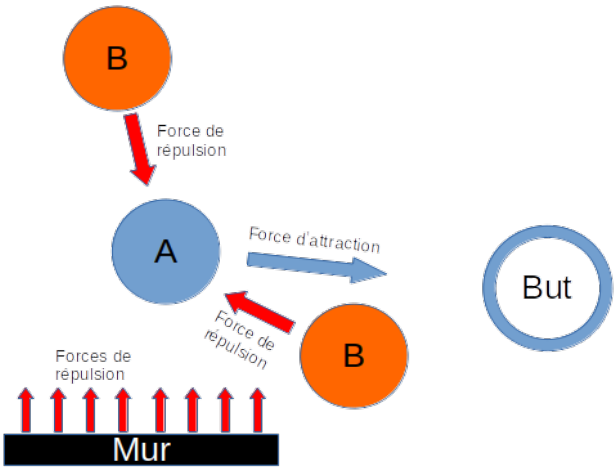


Figure 2.2 – Description des algorithmes basés sur des forces. L'agent **A** est attiré par son **But**, et est repoussé par les agents environnants **B** et les obstacles statiques comme les **murs**.

méthode pionnière a été source d'inspiration de nombreuses autres méthodes. La figure 2.2 est un schéma représentatif du fonctionnement des méthodes de simulation utilisant des forces.

Karamouzas et al. [2009] proposent une méthode similaire à la méthode SFM mais la définition de la force repose sur la notion de *"time to collision"* (TTC) pour chaque agent, correspondant au temps restant avant la collision avec un autre agent. Plus tard, Karamouzas et al. [2014] introduisent la méthode *"Powerlaw"* qui minimise l'énergie nécessaire à l'interaction entre agents pour l'évitement. Cette énergie est exprimée en fonction de la mesure TTC. Cette méthode est considérée plus "humaine" car la force utilise des paramètres calculés a partir de données de trajectoires réelles. Zanlungo et al. [2011] proposent une autre méthode similaire à SFM mais la force utilise la notion de time to collision (TTC) globale, à la différence de SFM qui n'utilise que les positions relatives des agents. Cette méthode est plus proche de SFM que de celle de Karamouzas et al. [2009] dans son implémentation. Enfin, Reynolds [1987, 1999] propose quant à lui une méthode de simulation multi-agents inspiré du vol des étourneaux. Il édicte trois règles pour que les agents, appelés *boids*, effectuent des mouvements de groupe. Ces règles sont des expressions mathématiques permettant de garder la cohésion de groupe (les *boids* sont attirés vers leurs voisins), puis l'alignement (les *boids* s'orientent dans la même direction que leurs voisins), et enfin de procéder à l'évitement (les *boids* se repoussent lorsqu'ils sont trop proches). Ces trois règles peuvent êtres représentées elles aussi par des forces.

2.2.2 Évitement par contrainte de vitesse

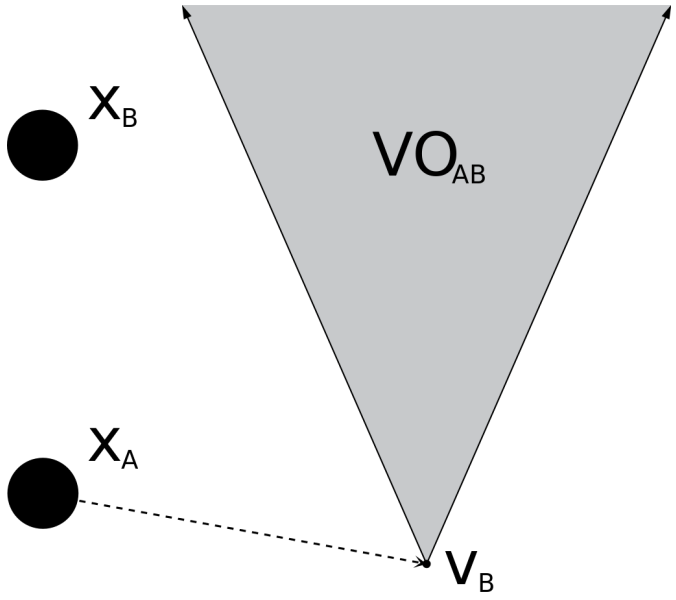


Figure 2.3 – Les vitesses amenant à des collisions (communément appelées "velocity obstacles") VO_{AB} pour un agent **A** de position x_A et de vitesse v_A, induites par un autre agent **B** de position x_B et de vitesse v_B

La faiblesse des méthodes basées sur les forces vient du fait qu'elles *n'anticipent pas* les états futurs du voisinage en fonction des états courants des agents. Pour introduire une certaine prédiction, l'idée des méthodes basées sur des contraintes de vitesses est de prendre en compte l'ensemble des vitesses atteignables, c'est à dire celles qui ne provoquerons pas de collisions avec les agents voisins, et de choisir judicieusement parmi ces vitesses. La figure 2.3 montre la forme des vitesses atteignables dans l'espace des vitesses pour un cas simple (un seul agent-obstacle). Paris et al. [2007] proposent alors le premier algorithme qui utilise des contraintes de vitesse pour l'évitement dans une simulation de foule. Cette méthode calcule les vitesses non-atteignables induites par les agents proches, et choisit une vitesse atteignable via une fonction de coût. Peu après, Berg et al. [2008] popularisent la méthode de contrainte des vitesses (Reciprocal Velocity Obstacles - RVO), et l'utilisent pour contrôler un essaim de robots. Cette méthode considère que les agents à éviter évitent en retour et partagent l'effort d'évitement. La méthode RVO est améliorée avec le temps [2011b], pour devenir une méthode très prisée dans de nombreux domaines, comme l'animation, les jeux vidéos, et bien sûr la robotique. Pour éviter les effets d'un évitement purement local, Bruneau and Pettré [2017] proposent une planification de trajectoire moyen terme en explicitant l'action à

effectuer par rapport à un agent, comme contourner par la droite ou la gauche, ou accélérer ou ralentir, et ce pour plusieurs évitements successifs. Cette méthode permet la navigation dans la foule dense de manière plus convaincante car des bénéfices à plus long terme sont recherchés dans les stratégies d'évitement. Karamouzas and Overmars [2010] proposent une méthode par échantillonnage avec un coût dépendant de la mesure TTC, de la différence entre la vitesse sélectionnée et la vitesse préférée de l'agent, de la différence entre la vitesse sélectionnée et la vitesse courante. L'éventail des vitesses possibles dépend du TTC calculé avec la vitesse préférée de l'agent. Cette méthode est plus réaliste que RVO ([2008]) car les formules pour le calcul des coûts comprennent des paramètres choisis pour correspondre à des données expérimentales de trajectoires réelles. Moussaïd et al. [2011] utilisent également une méthode par échantillonnage, très semblable à Karamouzas and Overmars [2010]. Avec cette méthode l'ensemble des vitesses atteignables est indépendant de l'état de l'agent, et les coûts vitesses ne dépendent pas de la différence par rapport à la vitesse courante. Cette méthode est plus simple, et donc moins coûteuse en calcul. Il est toutefois difficile de dire quelle méthode est la plus *réaliste* : bien que Moussaïd et al. [2011] comparent leurs simulations avec des données empiriques, la méthode elle-même n'est pas basée sur des observations du monde réel, à la différence de Karamouzas and Overmars [2010]. Guy et al. [2010] proposent une méthode qui calcule également les vitesses atteignables à la manière de van den Berg et al. [2011b]. La vitesse sélectionnée ensuite dépend d'une minimisation d'énergie selon le principe du moindre effort. Cette méthode est donc considérée plus réaliste que la méthode RVO car le principe du moindre effort est naturel dans la démarche humaine. Enfin, Pettré et al. [2009] quant à eux ne considèrent que les agents avec qui il y a interaction, et restreignent le nombre d'agents maximum à 7, afin de rendre le modèle plus réaliste. Ils introduisent alors la notion de "Minimum Predicted Distance" (MPD) pour définir l'interaction, qui remplace donc la mesure TTC. La mesure MPD correspond au temps restant avant d'atteindre le point où la distance entre deux agents est minimum. L'interaction n'est donc plus uniquement la conséquence d'une collision anticipée.

2.2.3 Évitement local basé sur une vision synthétique

Les algorithmes calculant les vitesses atteignables ont le défaut d'être omniscients : pour chaque agent, le calcul de trajectoire est effectué avec la pleine connaissance de l'état du voisinage de l'agent (il n'y a aucune incertitude quant à savoir quels agents sont proches, où ils sont, à quelle vitesse ils vont). Or, lorsqu'un humain se déplace, il le fait avec des informations partielles sur son environne-

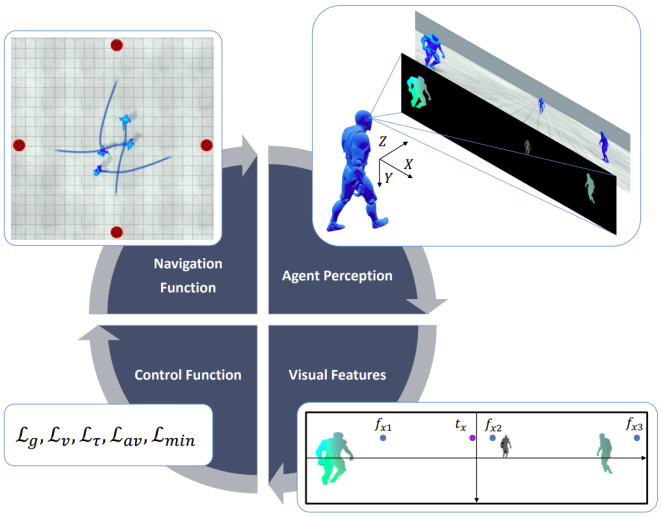


Figure 2.4 – Routine de contrôle des agents de l'algorithme de López et al. [2019b] (figure extraite de cet article). Les agents calculent des caractéristiques visuelles, utilisées dans des fonctions de coût, qui, combinées, donnent une fonction de navigation.

ment : celles que peuvent lui fournir ses propres sens. D'autres articles précédemment cités (Moussaïd et al. [2011]) utilisent alors le terme de *"vison-based"* pour parler d'un champ de vision, correspondant en fait à une zone dans laquelle sont sélectionnés les agents voisins. Ces algorithmes sont purement géométriques et ne tiennent pas compte du rôle de la perception dans la navigation humaine : les humains marchent pourtant en conséquence de ce qu'ils voient. Les algorithmes présentés dans cette catégorie ont donc pour principe que les actions d'évitement local sont la conséquence d'une perception visuelle de l'environnement : le sens de la vue est en effet le sens principal utilisé pour la marche.

Ainsi, c'est entre autres le flux optique qui est utilisé. Le flux optique fait référence au mouvement des objets tels qu'il apparaissent au yeux d'un observateur. Bien qu'on ne parle pas encore de foule, Warren et al. [2001], Warren [2006], Warren and Fajen [2008] utilisent le flux optique comme donnée d'entrée. Le flux optique est alors utilisé comme poids dans la variation d'orientation de l'agent. Plus tard, Ondřej et al. [2010] introduisent l'idée de travailler pour la première fois directement sur l'image : les agents-obstacles sont perçus sur une rétine virtuelle sous forme de cône en trois dimensions et introduisent la mesure *"Time To Closest Approach"* (TTCA). Leur algorithme calcule une fonction de coût basée sur la TTCA globale de l'ensemble des pixels de la rétine, et non pas pour chaque agent. Un agent plus proche prend plus de place sur la rétine, et a donc plus d'influence sur l'évitement. Dutra et al. [2017] améliorent ensuite le principe en proposant une optimisation de gradient pour la fonction de coût, rendant le mouvement plus

lisse. Par la suite, López et al. [2019a,b] utilisent le flux optique pour calculer un risque de collision par pixel. Ce risque est ensuite utilisé dans une fonction de coût qui est optimisée pour donner une vitesse à l'agent. La routine de cette méthode est décrite dans la figure 2.4.

Ces méthodes basées sur une vision synthétique ont la particularité de ne prendre en compte que les agents qui sont réellement vus, et ne prennent pas en compte les agents hors du champ de vision ni les agents occultés par d'autres agents. Une contrainte d'occultation qui rappelle le problème d'occultation des obstacles que rencontre un robot mobile en milieu dynamique.

Conclusion

Nous avons vu dans cette section qu'il existe trois catégories d'approches pour la simulation de foule. La première, l'approche macroscopique, s'intéresse principalement au mouvement de la foule dans son ensemble, mais ne permet pas de tenir compte des caractères individuels des comportements collectifs. Dans le cadre de nos travaux, nous avons le besoin de détailler les trajectoires individuelles, et de pouvoir jouer sur les caractéristiques individuelles des humains dans leurs trajectoires à proximité d'un robot. Dans ce sens, ces méthodes ne paraissent pas les plus appropriées. Les simulations basées données ont quant à elle l'avantage de proposer des résultats réalistes, dans le sens où elles reproduisent des comportements observés. Cependant, la question de la variété et de la quantité des données est cruciale, et il semble difficile de traiter de situations nouvelles, ou décrites concrètement, par les seules données à disposition. Enfin, les approches microscopiques reposent sur l'idée de simuler le déplacement de chaque individu par quelques règles qui régissent les interactions locales, et semblent à même de pouvoir s'adapter à notre cadre applicatif.

2.3 Simulation de robots

Les outils de simulation en robotique sont des outils précieux et indispensables. Un roboticien est généralement amené à développer un certain nombre d'éléments comme un algorithme, un mécanisme, un capteur, ou encore une disposition particulière de ces éléments. Nous appelons ici ces éléments des *concepts*. Les outils de simulation jouent alors un rôle important dans :

— la preuve de concept et la conception : la simulation est utilisée pour donner un aperçu rapide des conséquences de l'implémentation d'un concept et permet au roboticien d'ajuster sa conception rapidement ;

— la génération de base de données : la simulation permet de mettre en scène les concepts du roboticien dans des disposition diverses rapidement, et d'obtenir facilement un grand nombre d'informations utiles au roboticien pour mieux comprendre les implication de sa conception ;

— l'analyse de données : la simulation fourni des outils pour évaluer le bon fonctionnement, ou non, d'un concept.

Ces éléments permettent finalement la validation de concept par le roboticien, sans avoir besoin d'un robot à disposition.

Dans le cas de robots évoluant à proximité d'humains, de tels outils deviennent d'autant plus importants s'ils permettent, grâce à des méthodes d'analyse poussées, d'évaluer le niveau de sûreté du concept testé. Les données récoltées avec des tels outils sont également utiles pour les méthodes basée données, notamment les algorithmes basés sur des réseaux de neurones. La communauté des roboticiens répond donc à ces besoins en proposant une multitude de projets, souvent libres de droit et donc facile d'accès. Cette section est un aperçu de l'état actuel des outils de simulation en robotique et de leurs applications.

2.3.1 Études comparatives sur les outils de simulation en robotique

Afin de faire le point sur les différents outils judicieux pour la communauté robotique, il existe des études comparatives. Staranowicz and Mariottini [2011] présentent ainsi une enquête donnant une vue d'ensemble et une comparaison entre les outils logiciels robotiques commerciaux et libres de droit les plus populaires pour la simulation et l'interface avec des robots réels. Une étude de cas est présentée, montrant la polyvalence du portage d'un concept d'une simulation d'un robot mobile à un robot réel. De façon similaire, Castillo-Pizarro et al. [2010] comparent des outils de simulation libres de droits et commerciaux, sur des critères de simplicité, de flexibilité, ou de champs d'application. Ils réalisent une étude sur un scénario de simulation avec un robot mobile comparant trois outils de simulation en termes de précision de la trajectoire à réaliser, de précision de la simulation et de temps de calcul. Alors que Staranowicz and Mariottini [2011], Castillo-Pizarro et al. [2010] listent des outils à usage général ou dédiés à la robotique, Craighead et al. [2007] mettent en avant une autre approche. Ils listent des outils de simulation qui sont en fait des réutilisation d'outils commerciaux, notamment utilisés dans les jeux vidéos. Ces moteurs de jeux vidéos ont en effet été réutilisés pour la simulation robotique. Leur conclusion est qu'il n'est pas nécessaire de concevoir un simulateur robotique à partir de rien, car les outils existent déjà. Depuis ces

études (qui datent d'avant 2011), il existe peu d'articles sur l'état actuel des outils. Ivaldi et al. [2014] proposent une enquête concernant les outils de simulations sous forme de sondage adressé aux roboticiens. Ils mettent en avant le fait que ceux-ci cherchent des outils répondant à des besoins bien spécifiques notamment concernant les moteurs physiques, et que les outils modulaires qui supportent plusieurs solutions sont donc plus populaires dans la communauté. Ce sondage aillant eu lieu en 2014, on constate que les outils de simulation mentionnés par Ivaldi et al. [2014] qui sont toujours fortement actifs aujourd'hui sont les outils modulaires tels que *Gazebo* ou *V-rep*, alors que des outils plus spécifiques ont aujourd'hui disparu comme le logiciel *HumanS*. Ces enquêtes peuvent cependant être considérées comme obsolètes compte tenu de la date de leurs publications, il convient donc de faire un point sur l'état actuel des choses.

2.3.2 Les outils de simulation

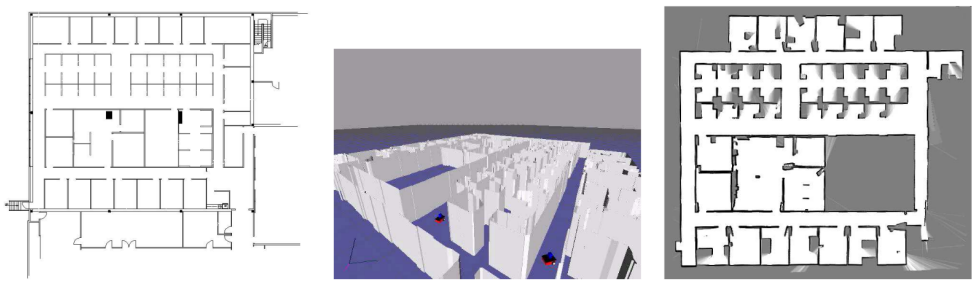


Figure 2.5 – Exemple d'utilisation du logiciel Gazebo [Koenig and Howard] : une carte 2D dessinée à la main, puis l'extrusion 3D créée à partir de la carte 2D dans laquelle circule une simulation d'un robot mobile Pioneer, et enfin la carte de l'environnement générée par laser dans gazebo.

Cette sous-section vise à présenter un aperçu des outils tels qu'ils sont disponibles aujourd'hui.

Gerkey et al. [2003] présentent un outil de simulation 2D pour la simulation multi-robot. Bien que dépassé aujourd'hui, il est avant tout à l'origine d'un projet plus grand, nommé Gazebo et présenté par Koenig and Howard. Gazebo est un logiciel de simulation 3D conçu pour reproduire aussi fidèlement que possible l'environnement dynamique d'un (ou plusieurs) robot mobile. Ce simulateur est développé en étroite collaboration avec la suite logiciel ROS, très populaire dans la communauté robotique. La particularité de ce simulateur est son utilisation libre de droit, et son développement qui repose sur la communauté robotique. Cette particularité fait de Gazebo un outil à la fois complexe, complet, et en perpétuelle

évolution. Gazebo permet de transférer des programmes de commande à plusieurs robots mobiles réels disponibles dans le commerce. Gazebo permet de définir et de modifier un robot entier, même si plusieurs robots différents partagent le même environnement. Pour chaque objet, il est possible de définir un certain nombre de propriétés, telles que la forme, la couleur, la texture, la masse, la friction, etc. Enfin, il est possible d'équiper chaque robot avec un grand nombre de capteurs et d'actionneurs disponibles. Michel [2004] présentent le logiciel libre Webots, qui, comme Gazebo, est un logiciel de simulation de robotique mobile offrant un environnement de prototypage rapide pour la modélisation, la programmation et la simulation de robots. C'est un logiciel libre depuis 2018 seulement. Il est moins complexe que Gazebo ce qui facilite sa prise en main mais peut être limitant à la longue. Par exemple, il n'a qu'un seul moteur physique. Rohmer et al. [2013] présentent le logiciel V-rep. Ce simulateur commercial offrait, à sa sortie, une plus large palette de techniques de programmation et des modèles de simulation et des contrôleurs plus facilement portables sur de vrais robot que ses concurrents Gazebo et Webots. V-rep serait donc plus facile à utiliser, mais demeure plus fermé que Gazebo et Webots.

Echeverria et al. [2011] présentent quant à eux le logiciel MORSE. La particularité de ce logiciel est de se baser sur le moteur de rendu graphique et de modélisation 3d dynamique Blender. Les auteurs cherchaient à donner à leurs robots des environnements plus réalistes que ceux proposés par Gazebo, Webots, ou V-rep, encore limités de ce point de vue en 2011. Leur idée est donc de s'appuyer sur la composante "moteur de jeu vidéo" du logiciel Blender.

Le moteur de jeu vidéo Unity est également utilisé comme outil de simulation. Konrad [2019] compare ainsi l'utilisation de Unity par rapport à Gazebo. Les simulations effectuées par Konrad [2019] dans Unity montraient des comportements réalistes des robots mobiles simulés. La mise en place de la simulation dans Unity était plus détaillée et plus proche de la réalité que la simulation dans Gazebo. L'une des principales raisons de ce résultat est l'absence de contrôleur de roues dans le simulateur Gazebo. Les bancs d'essai ont montré un comportement similaire à la réalité dans les simulations de Unity. Dans l'ensemble, Unity a été jugé adapté à la simulation de robots mobiles, si l'approximation des propriétés des robots est acceptable.

Enfin, Dosovitskiy et al. [2017] présente le logiciel CARLA. Plus récent que les précédents simulateurs cités, CARLA à été conçu sur la base du moteur de jeu vidéo ultra réaliste Unreal Engine. CARLA a été conçu pour répondre à un besoin spécifique : créer un outil de simulation dédié au développement des véhicules autonomes. Le simulateur propose des environnements urbains photo-réalistes, et

est conçu pour fonctionner facilement avec des réseaux de neurones.

Conclusion

On retrouve dans cet aperçu les outils les plus mentionnés dans les enquêtes présentées à la section précédente : les outils les plus modulaires sont ceux qui ont duré dans le temps. Les outils de simulation plus récents sont des outils dérivés de logiciels aillant un tout autre usage à l'origine, comme la conception de jeux vidéos ou la modélisation et rendu 3D ou encore l'animation.

Concernant le simulateur lui-même, Ivaldi et al. [2014] listent notamment, pour les utilisateurs d'outils de simulation, les critères de choix par ordre d'importance :

1. la simulation est très proche de la réalité ;
2. l'outil est libre de droit ;
3. le portage au robot réel est immédiat (même code) ;
4. l'outil est léger et rapide ;
5. l'outil est personnalisable ;
6. les corps ne s'interpénètrent pas. Cela dépend qualité du moteur physique.

Finalement, les utilisateurs cherchent avant tout à tester leur outil avant une évaluation grandeur nature sur robot réel. Dans notre cas des simulation foule-robot, les évaluations grandeur nature peuvent être particulièrement ardues. Nous considérons donc la qualité du moteur physique comme un critère hautement important. Fraichard and Levesy [2020] prétendent que les hypothèses fortes des outils de simulation de foule induisent un risque lors du portage sur robot réel. On peut étendre cette réflexion à l'ensemble de la simulation : l'environnement 3D, la physique, le rendu photo réaliste...

Cela soulève la question de la sécurité lors d'interactions humains-robot (HRI). Par exemple, Dongqing Shi et al. [2008] et Chi et al. [2016] présentent de méthodes de navigation pour robots mobiles testées en simulation puis évaluées en cas réel. Cependant, la simulation souffre de ses hypothèses simplificatrices, et *ne peux pas garantir* la sécurité des participants à l'expérience en cas réel. La sécurité des participants aux expériences *pourrait* être pourtant garantie avec une simulation de haute qualité. Mavrogiannis et al. [2019] font des expériences grandeur nature avec un robot mobile et des humains, avec "un grand nombre de participants à l'expérience". Ce papier souligne le manque de qualité des outils de simulation d'une part, et la rareté des données collectées et du nombre de participants impliqués dans le cas d'expériences grandeur nature (la plupart du temps, quelques participants). Cette rareté s'explique par les questions **éthiques** soulevées par

la présence de machine autonomes près d'humains. Il est difficile de garantir la sécurité des participants pour ce type d'expérience : en effet la question de la sécurité en HRI est de plus en plus présente. Par exemple, De Santis et al. [2008] dressent un portrait de la recherche en HRI, avec un focus sur les interactions physiques, et fait ressortir des notions maîtresses de la discussion : la sécurité (lié à la conception du robot) et la fiabilité (lié à la mise en oeuvre de la conception). Vasic and Billard [2013] réalisent une enquête sur le problème de la sécurité en robotique, dans différents domaines comme dans l'industrie, la robotique mobile de service, ou les véhicules autonomes. L'objectif est de déterminer les sources de dangers potentiels dans ces différents contextes pour éviter d'éventuels accidents.

Comment alors garantir sécurité et réalisme des expériences ?

Il faut donc de proposer des outils de simulation qui offrent davantage de réalisme. Notre approche consiste donc à immerger nos participants et nos robots dans un monde où les collisions n'ont pas de conséquences. Pour ce faire, nous utilisons la réalité virtuelle.

2.4 Réalité virtuelle

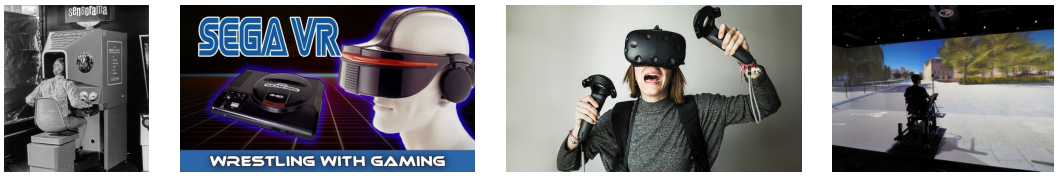
Fuchs [1996] propose une définition de la réalité virtuelle (RV) :

> *"La finalité de la réalité virtuelle est de permettre à une personne (ou à plusieurs) une activité* ***sensori-motrice*** *et cognitive dans un monde artificiel, créé numériquement, qui peut être imaginaire, symbolique ou une simulation de certains aspects du monde réel."*

L'utilisation de la RV est populaire dans le monde du jeu vidéo, mais elle a également un intérêt pour la recherche. Les scientifiques s'intéressant aux comportements humains utilisent la RV pour manipuler et contrôler l'ensemble de l'environnement expérimental (le monde artificiel), ce qui est impossible dans le monde physique. Tarr and Warren [2002] montre l'intérêt de la RV dans les années 2000 car elle présente un bon compromis entre contrôle et similitude de l'environnement. Par exemple, lors de l'étude des interactions avec les piétons, il est possible de contrôler parfaitement toutes les actions des piétons dans le temps, leurs comportements, leurs attributs (par exemple, le sexe, la taille, la morphologie, les vêtements...).

2.4.1 L'utilisation de la réalité virtuelle pour l'étude des mécanismes impliqués dans la locomotion humain

La réalité virtuelle, c'est d'abord la conception de systèmes immersifs. Les premiers casques de RV (HMD pour "Head Mounted Display") ont commencé à apparaître au début des années 1990 (Figure 2.6-b) et sont maintenant devenus courants et abordables pour le grand public, en particulier pour leur application dans les jeux vidéo (Figure 2.6-c). Il existe actuellement plus d'une douzaine de HMD sur le marché (par exemple, Vive, Occulus, Valve Index...), chacun ayant ses propres avantages et inconvénients. Outre les HMD, il existe une autre technologie de réalité virtuelle de type salle immersive appelée CAVE ("Cave Automatic Virtual Environment"), qui permet d'immerger une personne grâce à l'utilisation de lunettes et de murs en 3D sur lesquels sont projetées des vidéos en 3D (Figure 2.6-d), comme au cinéma.



a) b) c) d)

Figure 2.6 – De gauche à droite, l'évolution des systèmes de réalité virtuelle avec : Sensorama conçu par Heilig [1962] (a) en 1956, SegaVR (b) en 1991, HTC Vive (c) en 2019 le CAVE Immersia (d) à Rennes

Par ailleurs, la science de la réalité virtuelle, c'est aussi la création d'outils logiciels spécialisés. Par exemple, Cirio et al. [2013a] proposent un outil pour comparer des trajectoires réelles et virtuelles. Lin et al. [2016] proposent un système de réalité virtuelle incluant un système de suivi des mains permettant l'interaction avec les mains dans l'environnement virtuel, en temps réel. Moussaïd et al. [2016] développent quant à eux un outil de réalité virtuelle destiné à l'étude des scénarios à très haute densité de foule, comme lors de situations d'évacuations.

Grâce à ces outils, les scientifiques cherchant à déterminer les mécanismes en oeuvre dans la locomotion humaine ont pu utiliser la réalité virtuelle. Par exemple, Warren et al. [2001] utilisent la RV pour démontrer que le flux optique est utilisé dans la locomotion humaine, en modifiant le flux optique virtuel lors d'une expérience de locomotion en VR.

Un certain nombre d'expériences consécutives ont été menées en réalité virtuelle pour étudier les mécanismes en oeuvre dans la locomotion dans diverses situations. Sheik-Nainar and Kaber [2016] étudient l'effet du flux optique sur la

posture de participants marchant sur un tapis roulant en utilisant la RV en comparaison avec la posture sur un tapis roulant conventionnel (sans RV), et une marche conventionnelle. Chou et al. [2009] utilisent la réalité virtuelle pour étudier les différences de locomotion entre des participants d'âges différents. Fink et al. [2007] analysent si la locomotion est similaire en VR et en réel en comparant des trajet comprenant des obstacles. Agethen et al. [2018b] présentent une expérience analysant la disparité comportementale entre la locomotion humaine effectuée sans aucun équipement et en RV (avec un casque de réalité virtuelle). Aravind et al. [2015] proposent une expérience avec une tâche de navigation utilisant la réalité virtuelle pour révéler les performances d'évitement des obstacles chez les personnes présentant des troubles visuospatiaux.

Alors que la VR montrait ses atouts, il a fallu déterminer si elle était viable pour l'étude du comportement humain. Des expériences ont donc été menées pour déterminer les biais causés par l'utilisation de la RV. Par exemple, Armbrüster et al. [2008] ont étudié la perception de la profondeur dans les environnements virtuels et ont montré que les personnes immergées en RV *sous-estiment* les distances, tout en restant dans le bon ordre de grandeur. Rebenitsch and Owen [2016] passent en revue les méthodes, théories, et connaissances en lien avec le mal des simulateur (*"cybersickness"* en anglais), rendant malade certains utilisateur de systèmes de RV.

Bruneau [2016] consacre sa thèse a l'amélioration des algorithmes de simulation de foule dans des situations complexes, et cherche notamment à comprendre comment les êtres humains naviguent en évitant les collisions avec plusieurs personnes. Pour cela, des expériences ont été menées avec l'aide de la réalité virtuelle avec des participants qui devaient naviguer dans un environnement peuplé de plusieurs individus. Par exemple, Bruneau et al. [2015] proposent une expérience pour analyser ce qui pousse une personne amenée à aller vers un objectif se trouvant derrière un groupe de personnes, à choisir entre contourner ou passer à travers ce groupe. Enfin, Bühler and Lamontagne [2018] proposent une étude dont les objectifs sont d'une part d'examiner les stratégies de contournement en réponse aux piétons qui s'approchent à partir de différentes directions dans l'environnement virtuel par rapport à l'environnement physique, et d'autre part de déterminer les effets de l'entraînement répété à effectuer cette tâche sur les stratégies de contournement.

Figure 2.7 – Utilisation d'un CAVE pour l'immersion en réalité virtuelle d'un utilisateur de fauteuil roulant à assistance anti-collision

2.4.2 Applications en robotique

La réalité virtuelle présente également un intérêt pour la robotique. En particulier, elle est utile lorsqu'un robot et un humain sont amenés à coopérer dans une tâche. Son pouvoir de contrôle de l'environnement en fait un outil idéal pour l'entraînement des personnes destinées à interagir avec des robots. Par exemple, Devigne et al. [2017] présentent une plateforme d'immersion en réalité virtuelle (Figure 2.7) pour l'apprentissage à la conduite d'un fauteuil roulant électrique semi-autonome, c'est-à-dire équipé d'une assistance à la conduite anti-collision. La RV peut être vue également comme un organe de sécurité entre une personne et un robot collaboratif. Par exemple, Hernoux et al. [2015] utilise la VR comme interface de sécurité entre un cobot (robot collaboratif) et un opérateur.

La RV a également été suggérée comme outil d'étude pour le domaine de l'interaction humain robot. Des études comparatives ont été menées, comme Li et al. [2019], Wainer et al. [2007] qui comparent l'interaction humain robot en RV par rapport à une interaction réelle. Il n'y a cependant pas d'interaction physique ni d'interactions entre trajectoires : l'humain et le robot sont fixes.

Conclusion

On a vu dans cette section que diverses technologies permettant l'immersion dans des environnements virtuels ont vu le jour et se sont améliorés avec les années (Figure 2.6). On a vu que les scientifiques cherchant à comprendre les mécanismes de la locomotion humaine se sont intéressés à cette technologie. Ces

chercheurs ont montré avec le temps que la technologie peut être biaisée mais fiable pour mener des expériences de locomotion. Par ailleurs, la communauté de roboticiens a accueilli la réalité virtuelle comme une nouvelle interface humain-robot, permettant la collaboration entre un humain et un robot en toute sécurité.

2.5 Analyse de l'état de l'art - Conclusion

Notre approche relative à la simulation d'interaction robot-foule est à l'interface de trois domaines, montrés dans la Figure 2.8, que sont :

1. la simulation de robots mobiles dans des environnements dynamiques ;
2. la simulation de foule ;
3. l'interaction par la réalité virtuelle par l'immersion de sujets et de robots dans cette simulation.

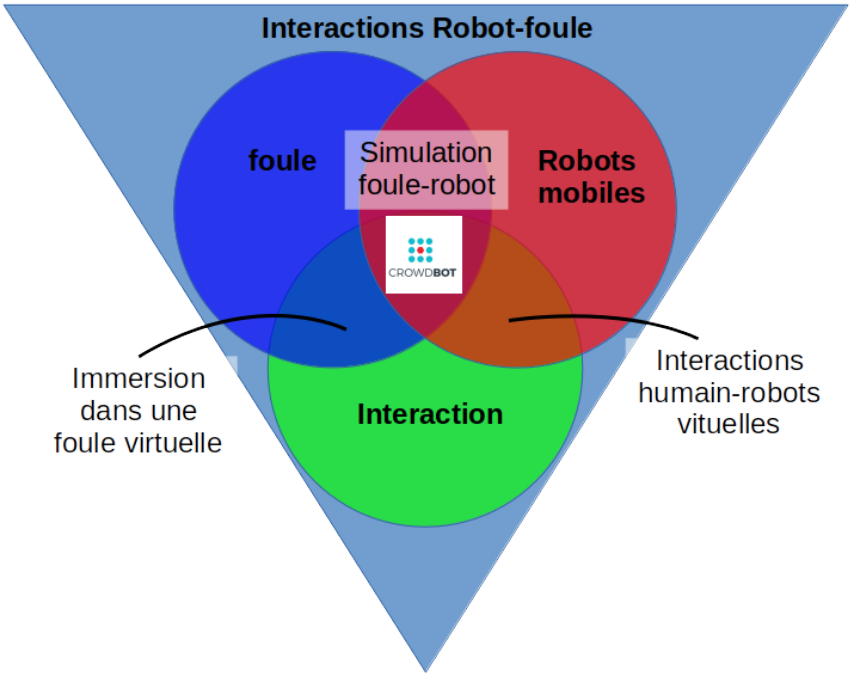


Figure 2.8 – Les trois domaines d'études pour notre approche de la simulation d'interaction robot-foule

Nous avons passé en revue dans la section 2.1 le domaine de la simulation de foule. Nous avons vu qu'il était possible de classer les algorithmes de simulation de foule selon les catégories suivantes ;

— simulation macroscopique ;

— simulation basée donnée et basée réseau de neurones ;

— simulation microscopique.

Nous avons vu que la simulation macroscopique s'intéressait principalement au mouvement global de la foule, et par conséquent, cette approche n'est pas

adaptée à notre problème d'interaction robot-foule, nécessairement centrée sur le robot. Nous avons également vu que les simulations basées données proposent des résultats réalistes : par exemple, il existe des jeux de données de trajectoires de foule issues d'enregistrements vidéos comme celui utilisé dans Pellegrini et al. [2012] ou encore Lerner et al. [2007]. Le dépôt `https://github.com/crowdbotp/OpenTraj` répertorie des jeux de données publiques de trajectoires de foule, et standardise le format des données pour être utilisables par un simulateur comme le nôtre. Malheureusement, ces approches basée sur des données sont inefficaces pour prendre en compte un élément dynamique inconnu de la foule, à savoir le robot. Nous avons vu qu'il existe une approche plus adaptée à notre cadre applicatif : les approches microscopiques qui sont orientées autour de l'interaction locale.

Cependant, la simulation de foule microscopique souffre d'un manque d'outils capable de les unifier. Fraichard and Levesy [2020] stipulent que les article présentant des algorithmes de simulation de foule microscopiques font des hypothèses fortes qui ne sont pas satisfaisantes pour des cas réels et qui sont donc problématiques d'un point de vue sécurité. En effet, la foule *simulée* est bien souvent homogène (les paramètres sont les mêmes chez tous les agents), la foule est bien souvent omnisciente (les agents connaissent la position de tout les autres agents à chaque instant), les collisions entre agents sont souvent mal gérée ou traitées. Par ailleurs, ces algorithmes sont souvent présentés comme indépendants des autres algorithmes, donnant l'illusion qu'ils se suffisent à eux seuls en toutes circonstances. Cependant, certains algorithmes échouent dans des situations où d'autres réussissent. La combinaison d'algorithmes pourrait être un atout pour des simulation robustes. Nous proposerons un moyen de rassembler ces algorithmes autour d'un principe unificateur dans le chapitre 5.

Dans la section 2.3, nous avons passé en revue les outils de simulation robotiques actuels. Nous avons vu qu'il y avait d'un côté les besoins des roboticiens : des outils de simulation de simulation réalistes, libres et sans contraintes de portage sur robot réel. De l'autre, les outils proposés à la communauté des roboticiens : soit des logiciels hyper spécialisés qui ont fini par disparaître, et des logiciels à usage général mais souffrant d'approximations dans la simulation. Aujourd'hui, les logiciels les plus utilisés sont ceux qui ont su toucher l'ensemble de la communauté, mais qui sont parfois insuffisants. Lorsqu'il y a un besoin particulier, comme pour Dosovitskiy et al. [2017] qui avait besoin d'un environnement photo-réaliste, de nouveaux logiciels sont développés. Ces nouveaux logiciels utilisent et détournent des logiciels servant à la conception de jeux vidéo ou d'animation. Notre approche utilisant la réalité virtuelle pour l'interaction, il fallait que notre outil de simulation supporte cette technologie. Le chapitre 3 montre comment nous simulons nos

robots.

Enfin, dans la section 2.4, nous avons donné un aperçu de la réalité virtuelle, en axant sur les études du comportement humain en VR d'une façon générale et sur les mécanismes de la locomotion en particulier. Nous avons vu que cet outil est maintenant répandu et ses biais sont bien connus. Nous avons ensuite présenté des exemples d'utilisation de la réalité virtuelle pour la robotique. Elle se présente comme un outil d'interface lorsque l'humain et le robot doivent coopérer dans une tâche commune. Cependant, il n'y a pas aujourd'hui d'immersion de robots mobiles. En effet, si nous avons bien répertorié des expériences impliquant des robots qui sont capables de se déplacer, mais ils étaient fixes dans ces expériences. Pour notre part, notre problème porte sur l'interaction robot-foule, et dans ce contexte, le robot se déplace forcément. Le chapitre 7 montre notre approche pour immerger un robot et un humain ensemble dans un monde virtuel.

Figure 2.9 – Un doctorant cherchant à rejoindre le robot Pepper dans son monde virtuel

Chapitre

3 Simulateur robot-foule

Sommaire

Dans ce chapitre, nous présentons une vue d'ensemble d'un simulateur robot-foule pour permettre l'évaluation *in silico* des capacités d'un robot à se mouvoir parmi la foule. Nous avons vu dans la section 2.3 que la simulation est un outil très communément utilisé en robotique. Idéalement, notre outil de simulation reproduit avec fidélité le comportement du robot ainsi que l'environnement dans lequel il évolue. Son rôle est d'offrir à la communauté robotique un outil innovant : la simulation d'une foule qui peuple l'environnement virtuel et qui réagit en fonction des actions du robot simulé. Les scénarios d'usage d'un tel outil sont multiples :

— permettre le prototypage rapide d'expérimentations robotiques en présence d'un grand nombre de personnes ;

— permettre l'évaluation détaillée des expériences de simulation ;

— fournir un environnement propice au développement d'algorithmes nécessitant beaucoup de données, comme les algorithmes basés sur de l'apprentissage profond.

Ce simulateur vise ainsi à combler un manque dans la simulation robotique aujourd'hui. En effet, la plupart des logiciels de simulation (voir section 2.3) simulent un environnement statique, voire qui incorporent quelques objets dont le mouvement est soumis aux lois de la physique. Néanmoins, pour reproduire de façon réaliste les situations réelles d'évolution d'un robot dans son environnement, il est nécessaire de pouvoir simuler des environnements dynamiques, en particulier des environnements peuplés d'humains. Dans ce domaine, s'il existe des simulateurs qui incluent une simulation de piétons, cette simulation, bien que fonctionnelle, demeure secondaire. Par exemple, CARLA [Dosovitskiy et al., 2017] est conçu pour le développement des véhicules autonomes : il est donc nécessaire d'inclure la simulation de quelques piétons, mais la création d'une foule en tant que telle ne fait pas partie des scénarios envisagés avec cet outil. Ainsi, dans CARLA, les humains simulés se déplacent grâce à l'algorithme ORCA [van den Berg et al., 2011a] mais ne se déplacent pas en nombre suffisant pour créer des scénarios de foule intéressants comme des flux s'entrecroisant.

Notre contexte est donc très différent de celui des véhicules autonomes. Nous cherchons à pouvoir simuler un environnement de foule dense pour les robots qui y sont destinés. A cette fin, nous avons conçu un simulateur capable de générer des scénarios de robots naviguant dans une foule dense. La foule devient alors un élément central de la simulation au même titre que le robot lui même. Ce simulateur permet donc d'évaluer le comportement du robot dans un environnement aussi réaliste que possible, tout en donnant des informations qui ne peuvent être calculées qu'en simulation, comme les informations liées aux collisions, l'intensité des contacts, ou encore la position précise de chaque agent de la foule. Finalement, ce simulateur est idéal pour le développement de nouveaux concepts de robotique mobile prenant la foule en compte.

Le simulateur robot-foule proposé s'appuie sur le schéma 3.1 présentant de manière générale les grandes composantes du simulateur, ainsi que les relations entre ces composantes. Ces grandes composantes du simulateur robot-foule ont besoin de communiquer, d'obtenir des informations relatives au scénario, l'environnement, le déroulement de la simulation. Pour ce faire, nous définissons deux

types d'informations : les paramètres et les états. Les paramètres constituent le point de départ de chaque élément. On y retrouve l'ensemble des positions des murs, des agents à l'instant initial, des robots, des caméras. Les paramètres sont stockés dans des fichiers *XML* que nous appelons "fichiers de configuration". Les états représentent les informations dynamiques de la simulation. On y retrouve les vitesses, les positions, les buts (*i.e* les objectifs à atteindre en se déplaçant), les rapport sur les collisions, ou encore les données des capteurs. Les états ne sont en principe disponibles qu'à travers le canal de communication de ROS (détails dans la section 3.2), à l'exception de la position du robot qui est communiquée en interne au simulateur de foule. Par défaut, les états ne sont ni calculés ni enregistrés : il faut définir dans le simulateur quels états sont nécessaires et quels sont les états à enregistrer.

Le simulateur de foule et le simulateur de robot sont similaires au sens où les leur but est de définir des vitesses pour le temps de simulation suivant. Cependant, il se peut que les vitesses choisies soient physiquement impossibles. C'est le rôle du moteur physique qui calcule les collisions lorsque toutes les vitesses ont été calculées. Il a le dernier mot sur l'état réel des agents et des robots, et impose donc des contraintes. Nous utilisons le moteur physique par défaut de Unity, *PhysX*, par souci de simplicité. Cependant, nous avons fait la preuve de concept d'utiliser un autre moteur physique, *Bullet*, de façon externe au simulateur : notre solution est donc générique.

3.1 Simulateurs de foule

3.1.1 Généralités

Les objectifs de notre simulateur sont de :

1. générer des trajectoires de piétons qui évoluent dans l'environnement de simulation ;
2. faire en sorte que ces trajectoires simulées soient le reflet des trajectoires réelles, c'est à dire que l'ensemble des propriétés de ces trajectoires simulées soient cohérentes avec des trajectoires réelles ;
3. assurer la variété des trajectoires simulées : les agents peuvent avoir des comportements très différents.

Le problème, c'est que nous allons générer ces trajectoires avec des modèles de simulation microscopique dont le domaine de validité est *limité*. En effet, ils ont été conçus pour un certain contexte :

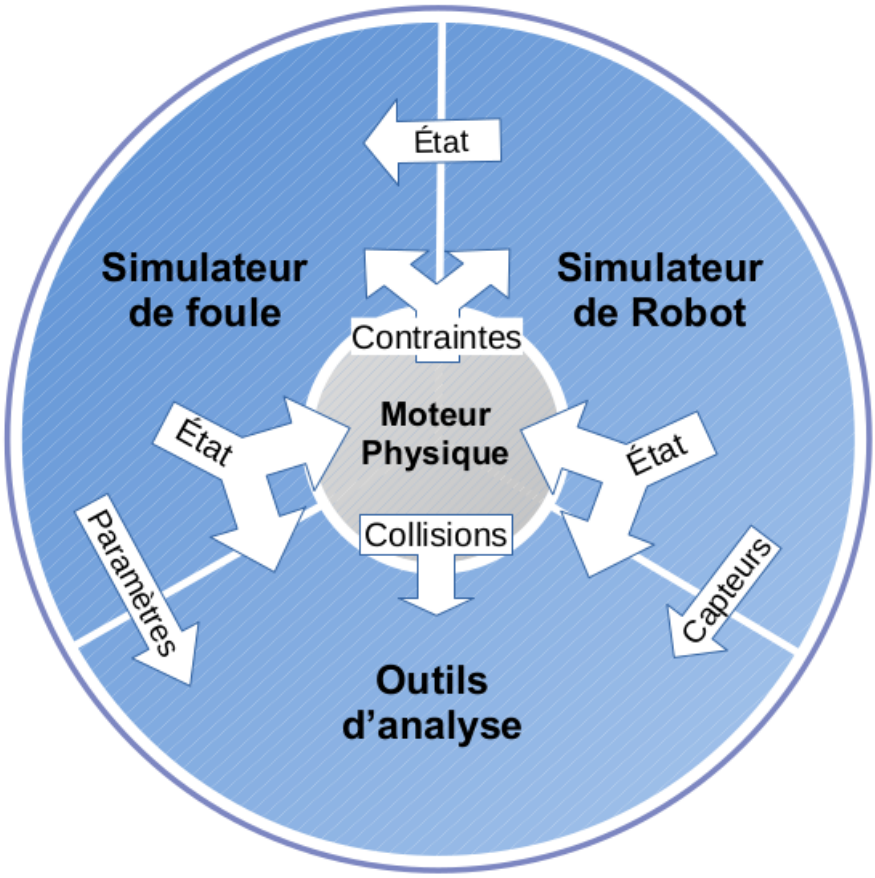


Figure 3.1 – Schéma général du interactions internes du simulateur. Les briques logicielles essentielles sont le simulateur de foule, le simulateur de robot, et l'outil d'analyse.

— Ces modèles ont un niveau de densité typique. La densité est faible pour les approche basées vision, moyenne pour les approches basée contraintes de vitesses, et plus élevées pour les approches basées sur des forces. Ces approches sont toutes présentées dans la section 2.2).

— Ces modèles sont adaptés à un petit nombre de comportements. Typiquement, l'évitement de collision entre agents indépendants (individuels) est la norme. Quelques modèles traitent du suivi ou des déplacements en groupe.

— La question des paramètres de simulation est aussi cruciale, car il existe un lien entre les paramètres utilisés et la nature des résultats produits. Ces paramètres sont autant liés à l'individu qu'il faut simuler (e.g., adulte pressé, ou personne âgée en promenade) qu'à la situation (situation normale, ou évacuation).

Finalement, notre simulateur vise à générer une foule *réaliste* autour du robot : des trajectoires variées sont générées, qui combinées, montrent des propriétés in-

trinsèques à des trajectoires réelles. On comprend qu'utiliser *un modèle unique* de simulation ne permettra pas d'assurer ce réalisme car il restreint trop le domaine de validité de notre simulateur. Ainsi, nous suggérons dans ce travail de généraliser les modèles existants sous la forme d'une unique *structure* qui va permettre de varier les algorithmes de simulation employés dans notre simulateur.

3.1.2 La standardisation des entrées et sorties

Afin d'être transférable à des situations réelles, l'outil de simulation se veut à la fois polyvalent et avec des entrées/sorties standard. Le simulateur robot-foule s'adresse à des utilisateurs possédant des notions de simulation de foule très différentes. Il faut donc standardiser les entrées et sorties de l'outil, quels que soient les paramètres internes au simulateur, afin de pouvoir l'intégrer de manière harmonieuse dans les outils de simulation robot-foule. L'outil de simulation de foule que nous proposons, décrit plus en détail au chapitre 5, est intégré au simulateur selon le schéma 3.1. Le simulateur ainsi reçoit l'état du ou des robots présents dans l'environnement. On définit l'état à un instant donné de la simulation par la position, la vitesse, et le but d'un agent à cet instant. Cette définition est valable pour les robots et les agents de la foule.

Le simulateur reçoit également les contraintes issues du moteur physique. Ces contraintes sont de deux sortes : statiques ou dynamiques. Les contraintes statiques correspondent à l'agencement de l'environnement dans lequel les agents évoluent. On fournit alors au simulateur de foule les informations relatives à l'environnement par deux moyens au choix : on peut donner, selon la complexité de l'environnement, soit la position des murs et des pylônes qui indique alors au simulateur la zone où les agents de la foule ne peuvent pas se déplacer, soit un maillage de navigation correspondant à la zone où les agents peuvent se déplacer. Le simulateur de foule reçoit également des contraintes dynamiques, qui sont calculées au cours de la simulation lors des contacts entre un agent simulé et un robot ou un objet de l'environnement. Cette contrainte vient empêcher les aberrations physique telles que notamment l'interpénétration entre objets 3D. Le moteur physique a donc le dernier mot pour définir l'état de la foule à un instant donné.

Les sorties du simulateur de foule sont apparentées à un état de la foule : la position, la vitesse, et le but de chaque agent à chaque instant. Cette information est fournie à l'outil d'analyse. Les paramètres du simulateur, standardisés autant que possible comme décrit dans le chapitre 5, sont également fournis à l'outil d'analyse. Pour pouvoir utiliser ces entrées/sorties, les outils de simulation de foule utilisés sont nécessairement des algorithmes microscopiques car il est indispensable

de connaître l'état de chaque agent à chaque instant.

3.1.3 La généricité des simulateurs de foule microscopiques

L simulateur de foule utilise deux niveaux de navigation : la navigation globale et la navigation locale.

La navigation globale permet de donner un objectif long terme pour l'agent. L'agent prendra une vitesse souhaitée, à chaque instant de la simulation, calculée en fonction de ce but à long terme, et se déplacera alors avec une animation de marche ou de course. Par exemple, on peut choisir d'utiliser une loi de contrôle pour donner une vitesse constante à l'agent sans changement d'orientation, ou encore on peut choisir de donner un point ou une liste de points de l'espace navigable comme buts successifs à l'agent. On peut aussi choisir d'utiliser une loi de contrôle qui définit une vitesse dirigée directement vers le but, ou alors une vitesse qui suit un chemin défini par un algorithme de recherche de chemin (A* par exemple) rejoignant le (ou les) but(s). Il est également possible d'utiliser une loi qui contrôle la vitesse de l'agent par un dispositif externe, tel n Joystick ou un clavier. Enfin, il est possible de donner pour objectif de rester sur place. Il existe alors des animations pour rendre l'agent plus réaliste, comme téléphoner, parler, ou simplement appliquer des micro-mouvements des bras et du bassin.

La navigation locale permet de calculer une vitesse pour l'agent qui soit sans risques à partir d'une vitesse souhaitée et des obstacles et agents environnants. On peut, si on le souhaite, utiliser alors des algorithmes populaires tels que RVO Berg et al. [2008] ou encore SFM Helbing and Molnár [1995]. Mais si on y regarde de plus près, les simulateurs de foule microscopiques tels qu'ils sont décrits dans la section 2.1 peuvent se modéliser sous la forme *générique* suivante :

$$S_i(t) = \begin{pmatrix} x_i(t) \\ v_i(t) \\ g_i(t) \end{pmatrix}, \tag{3.1}$$

$$\begin{pmatrix} S_1(t+1) \\ S_2(t+1) \\ \vdots \\ S_n(t+1) \end{pmatrix} = f((S_1(t), \ldots, S_n(t)), p_0, \ldots, p_m). \tag{3.2}$$

L'équation 3.1 définit l'état noté S d'un agent i de la foule à un instant t. On y note $x_i(t)$, $v_i(t)$ et $g_i(t)$ respectivement la position, la vitesse, et le but, de l'agent i à l'instant t, exprimés dans le repère fixe du monde.

L'équation 3.2 définit la fonction f qui donne les états à l'instant suivant $t+1$ pour chacun des n agents de la foule en fonction l'état à l'instant t de chacun des agents de la foules, ainsi que d'un certain nombre de paramètres fixés qui définissent les réglages du simulateur permettant de montrer des comportements de foules différents (par exemple, le rayon du cercle représentant les agents).

Cependant, on peut aller plus loin en montrant que ces algorithmes peuvent être reformulés sous la forme d'une fonction de coût C à optimiser dans l'espace des vitesses. On peut alors utiliser la formulation mathématique suivante :

$$C(S_{\text{optim}}) = G[C(S)|S \in P], \tag{3.3}$$

où C est une fonction de coût pour un état donné, G est une méthode d'optimisation, P est l'ensemble des état dont les vitesses sont admissibles et qui tiennent compte des autres agents de la foule, et S_{optim} est l'état pour lequel la fonction de coût C est minimale.

On défini alors l'état suivant d'un agent comme ceci :

$$S_n(t+1) = S_n(t) + S_{optim}(t), \tag{3.4}$$

où $S_n(t+1)$ est l'état de l'agent n à l'instant suivant $t+1$.

Au final, notre outil de simulation de foule utilise ce principe pour créer des foules hybrides. Par exemple, on peut imaginer que la moitié d'une foule utilise RVO et que l'autre utilise SFM. On peut même imaginer que les agents puissent utiliser une méthode basée vision comme Dutra et al. [2017] pour les agents éloignés, et RVO ou SFM pour les agents avec qui la collision est imminente. Ce principe de généricité des simulateurs de foule est à la base de la contribution décrite au chapitre 5.

3.2 Simulateur de robot

3.2.1 Généralités

Le deuxième élément présent sur la figure 3.1 est le simulateur de robots. Il est pensé pour fonctionner avec l'outil standard en robotique ROS [1]). ROS fournit de nombreux outils qui passent nécessairement par le système de communication de ROS. C'est donc par ce système que l'ensemble des données du simulateur robotique circulent. Le simulateur robotique prend comme entrées :

1. voir https://ros.org

— l'état de la foule correspondant à la position et à la vitesse de chaque agent pour chaque pas de temps ;

— la description de l'environnement statique sous la forme d'une liste de polygones $[x_1, y_1, x_2, y_2, ..., x_n, y_n]$;

— une horloge. Le simulateur robotique est en charge du déroulement de la simulation : il fait avancer le temps de la simulation générale, en utilisant une horloge interne fournie par Unity, ou alors une horloge externe qui est alors fournie par le biais de ROS ;

— l'état nouveau du robot, c'est à dire la vitesse et la position de l'ensemble des éléments mobiles du robot.

Les actions à effectuer dépendent ensuite du robot lui-même. On trouve dans le simulateur des robots mobiles avec des propriétés mécaniques très différentes [2]. Le simulateur prend en compte l'inertie du robot et ses différentes articulations grâce à une définition basée sur le standard *URDF*[3]. Un soin particulier a été apporté pour respecter les spécificités des différentes bases mobiles : certains robots se déplacent avec deux roues motrices en montage différentiel, alors que d'autres utilisent des roues holonomes leur permettant un déplacement latéral instantané. De ces spécificités découlent des contrôles différents du robot pour une commande donnée :

— contrôle sans dynamique : la position de commande est alors celle du robot dans le simulateur ;

— contrôle dynamique très simple très simple de type point-masse : on applique une force en un point unique pour atteindre la vitesse désirée ;

— contrôle dynamique avancéé : on utilise directement les moteurs virtuels en leur appliquant un couple qui génère le mouvement des roues. Il faut dans ce cas définir une chaîne de contrôle, par exemple basée sur le principe du PID.

On définit également pour notre robot des sorties issues des capteurs : les données calculées par les capteurs virtuels respectent les formats standards de messages ROS. Seul le simulateur de foule récupère les informations d'état (vitesse et position) du robot directement, sans passer par ROS, pour des raisons de simplifications techniques. Le simulateur de foule considère en fait le robot

2. L'ensemble des robots actuellement supportés par le simulateur sont disponible sous license MIT sur https://github.com/FabienGrzeskowiakInria/CrowdBot_robots.

3. décrit sur http://wiki.ros.org/urdf

comme un agent non réactif de la foule. Dans ce cadre, l'outil de simulation *Gazebo* [Koenig and Howard] est très populaire et répond déjà à la description faite de notre simulateur, mais il lui manque une fonctionnalité maîtresse pour notre contexte : notre simulateur de robot est en effet combiné avec des outils puissants de simulation de foule et d'analyse de la foule.

L'architecture présentée figure 3.2 correspond à une architecture standard pour un robot destiné à naviguer au sein d'une foule. Cette architecture décrit les différents modules à développer pour parvenir à cet objectif, et montre leurs interconnexions. La conception du simulateur de robot s'appuie sur deux principes. Premièrement, les interconnexions présentes sur l'architecture de la figure 3.2 reposent sur des outils standards de communication en robotique qu'il convient d'utiliser dans le simulateur à des fins d'harmonisation. Deuxièmement, le simulateur doit se comporter comme un robot mobile pour lequel cette architecture est destinée. Il doit disposer des mêmes entrées et sorties, et doit être aussi réaliste que possible concernant la simulation du robot (en terme de mouvements du robot et de données de capteurs) . Ainsi, les outils développés sur le simulateur doivent être transférables aux robots mobiles réels sans efforts.

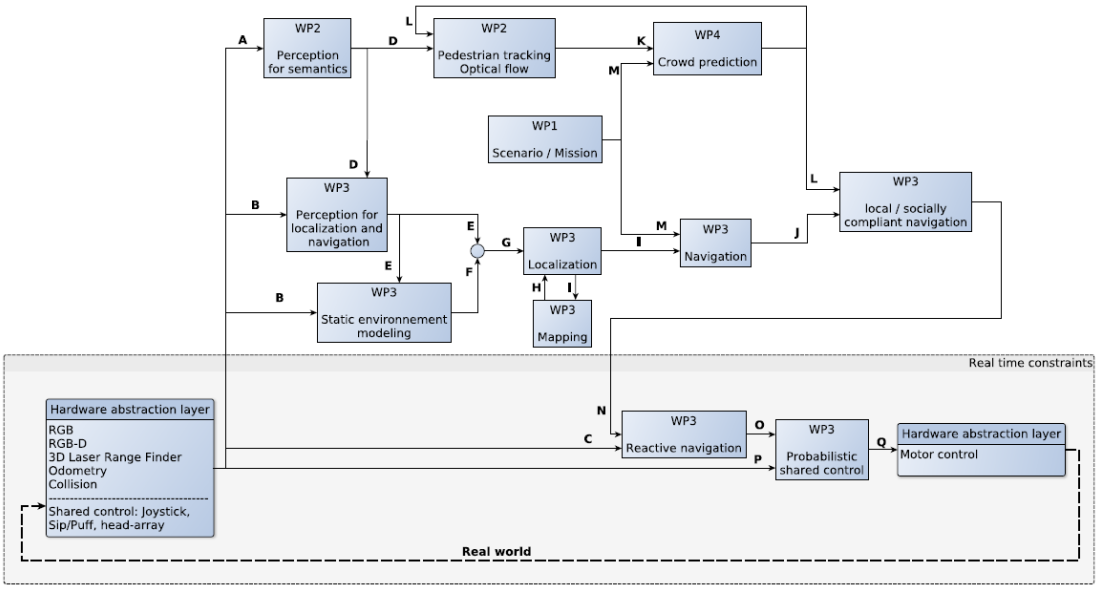


Figure 3.2 – Schéma de l'architecture système d'un robot mobile destiné à naviguer dans la foule

3.2.2 ROS : La standardisation des entrées et sorties

Comme indiqué précédemment, nous avons choisi de nous appuyer sur l'outil ROS. ROS (Robot Operating System) est un environnement de développement flexible permettant d'écrire des logiciels pour des applications robotiques. Il s'agit d'un ensemble d'outils, de bibliothèques et de conventions qui visent à simplifier la tâche de création de comportements complexes et robustes pour les robots sur une grande variété de plates-formes robotiques. ROS a été imaginé pour encourager le développement de logiciels de robotique en collaboration. Par exemple, un laboratoire peut développer un module de détection des personnes environnantes, et pourrait contribuer à ROS en rendant public leur outil. Un autre groupe peut développer un module de navigation, et utiliser le module du premier groupe, ou un autre déjà existant. Nous avons donc choisi d'utiliser des entrées et sorties utilisant les messages standards de ROS. Le simulateur robot-foule a d'abord été conçu avec l'idée d'être utilisable a travers ROS : les entrées et sorties du simulateur sont donc les mêmes que si le robot était réel et évoluait dans un environnement réel. La façon dont les données issues des capteurs du robot virtuel est la même que pour le robot réel.

Pour autant, pour rendre encore plus générique notre simulateur, nous avons réfléchi à la manière de nous affranchir d'un outil logiciel externe tel que ROS. En particulier, le simulateur peut être utilisé pour la génération de données sur un serveur, ou pour de l'apprentissage pas à pas (Q learning). Dans ce contexte, un couche de séparation entre le simulateur et ROS a été introduite. Cette couche intermédiaire permet d'utiliser le simulateur avec ou sans ROS.

3.2.3 Simulation physique et réalisme du comportement

Nos robots sont composés de deux attributs :

— leur chaîne cinématique : composée de membres et d'articulations ;

— leurs formes et dimensions.

La tâche du robot va conditionner ces attributs. Par exemple, la conception d'un robot social va être conçu pour avoir une forme "agréable", alors que pour un robot d'usine, l'apparence est moins importante donc la forme est surtout la conséquence des besoins fonctionnels.

Ces deux attributs sont à modéliser par le simulateur. On définit alors mathématiquement la chaîne cinématique : un membre est considéré comme un corps rigide non déformable, les articulations sont parfaites et sans frottements, et elles définissent les contraintes entre les membres.

Cependant, dans la réalité, il existe des frottements et du jeu dans les articulations, et les membres sont des corps déformables dont le comportement au stress d'un membre dépend de son matériau et de sa forme. Il existe des méthodes permettant de simuler de façon plus réaliste la mécanique du robot en utilisant des modèles déformables ou en complexifiant la définition des articulations en ajoutant des contraintes modélisant des frottements par exemple. Cependant, lorsqu'on parle de simulation, on parle surtout de compromis entre réalisme et faisabilité. En effet, l'utilisation d'hypothèses simplificatrices (corps rigide, articulations parfaites, monde dans un plan 2D...) permet soit de simplifier les calculs à effectuer par la machine, et accélère ainsi le processus de simulation, soit de rendre la simulation possible. En effet, la simulation requiert une expression mathématique simplifiée pour représenter un phénomène physique complexe. Par exemple, les modèles de frottements souvent utilisés sont souvent trop simples et ne peuvent tenir compte de toutes les forces de frottements possibles. Pourtant, utiliser ces modèles, bien qu'imparfaits, rend la simulation plus réaliste. Pour notre simulateur, le choix a été fait de garder les modèles de corps rigides, et de définir la chaîne cinématique du robot et sa modélisation 3D au plus proche de la réalité. Par exemple, les robots virtuels utilisent des roues et des moteurs virtuels pour se déplacer. Le corps du robot n'est pas juste une simple masse sans articulation à déplacer en poussant avec des forces virtuelles, mais bien un assemblage de membres et d'articulations. Pour les agents de la foule, le chaîne cinématique est conservée et les membres sont approximés par des cylindres indéformables, afin d'alléger la simulation en termes de calculs. Davantage de détails concernant la simulation robotique sont donnés dans le chapitre 4.

3.3 Outil d'analyse

Le dernier élément du simulateur robot-foule est l'outil d'analyse. La simulation hybride (robot-foule) a besoin d'un cadre d'évaluation permettant de conclure sur la qualité de l'algorithme a évaluer. Pour ce faire, il convient d'imaginer des métriques permettant de classer les algorithmes. Notre proposition dans cette thèse est de définir des métriques qui se répartissent sur les catégories suivantes : l'efficacité du chemin emprunté, l'effet sur la foule, et le risque lié aux collisions. Ces catégories sont détaillées dans le chapitre 6 qui décrit la création et l'usage d'un banc d'essai pour la navigation de robots dans la foule. L'outil d'analyse est indépendant du simulateur temps réel, car il ne prend que des données, et n'envoie rien au simulateur. L'ensemble des données disponibles sont récupérées par l'outil d'analyse via ROS :

— les états à chaque instant (les positions et vitesses) de chaque agent et de chaque élément mobile des robots ;

— les résultats des collisions ;

— les données des capteurs du robot ;

— la forme de l'environnement ;

— les scénarios (densité, état initial, but de la foule).

L'ensemble de ces données sont alors traitées, transformées et mises en forme pour être visualisées afin de comparer des situations similaires.

En vue de populariser nos méthodes et motiver la recherche dans le domaine de la robotique de foule, nous avons décidé d'organiser un défi à destination des chercheurs en robotique mobile sur la navigation des robots dans des environnements à forte densité de population, basé sur la plateforme proposée de simulation robot-foule. Bien sûr, les résultats de la simulation ne peuvent pas être directement transférés aux environnements réels, mais cela va déclencher un cercle vertueux : l'environnement de simulation va stimuler la recherche sur la navigation dans les foules. En retour, cette activité de recherche nécessitera un environnement de simulation toujours plus réaliste.

3.4 Immersion dans la simulation

Le simulateur tel qu'il est décrit dans la figure 3.1 souffre toujours d'un manque de réalisme lors des interactions de proximité. En effet, les agents simulés réagissent bien à la présence du robot virtuel lorsque l'interaction est facilement prévisible, mais dans une situation de foule dense, ces algorithmes sont contraints d'éviter le robot alors que la collision avec celui-ci est proche. Il faut donc trouver un meilleur modèle pour ces situations de quasi-collision.

> "The best model for a cat is another, or preferably the same cat." - Rosenblueth, A., & Wiener, N. (1945). The role of models in science.

L'idée est donc d'introduire la réalité virtuelle pour plonger l'utilisateur dans le simulateur afin de créer des situations de quasi collision aussi réalistes que possible. Ce module immersif s'intègre dans le simulateur comme décrit sur la figure 3.3. Le module immersif ne reçoit comme information que des données sensorielles : la vue, par le biais d'une caméra virtuelle, l'ouïe, par le biais de sons spatialisés du simulateur, et le toucher, par le biais du moteur physique et d'outils d'interaction physique. Le principe de ce module est décrit dans les chapitres 7

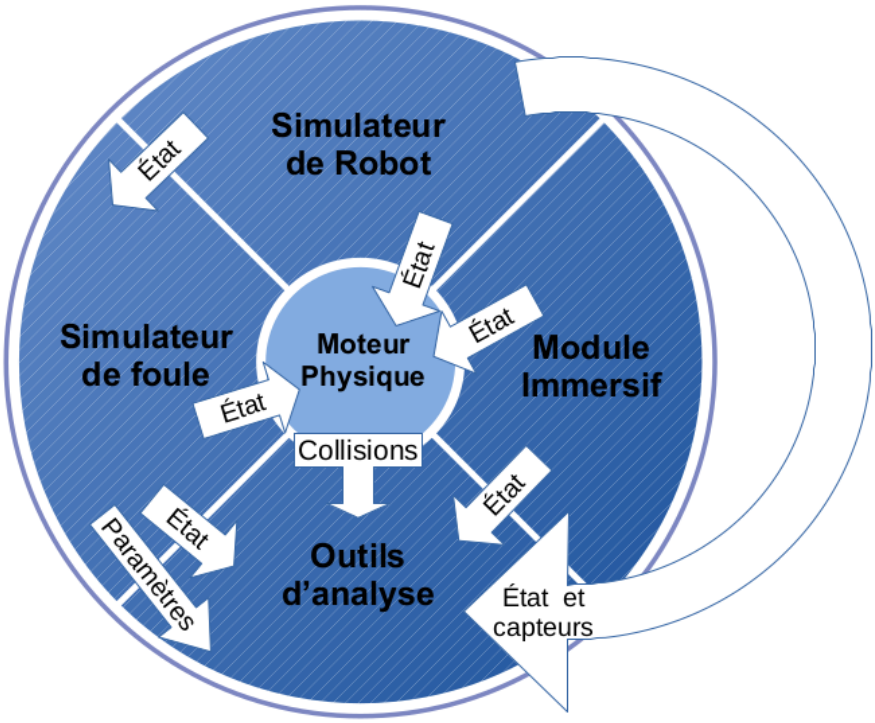


Figure 3.3 – Schéma augmenté des interactions internes du simulateur. On retrouve les modules essentiels au simulateur décrit dans la figure 3.1. On ajoute un module immersif permettant à une personne de se plonger dans le simulateur grâce à des outils de réalité virtuelle ou de réalité augmentée. Le module reçoit ainsi l'image d'une caméra virtuelle du simulateur pour l'outil de réalité virtuelle (casque, CAVE...)

et 8. Ce chapitre montre également l'évaluation de ce module. En effet, il nous a fallu valider le concept, identifier les biais dû à l'immersion, afin de réaliser des expériences avec la maîtrise de notre outil.

3.5 Conclusion

Dans ce chapitre nous avons décrit l'architecture d'un simulateur de robot-foule doté d'un module d'analyse. Cet outil vise à combler un manque en simulation robotique : le réalisme de la simulation de foule. Nous en avons détaillé les composantes principales, comment elle s'articulent entre elles et communiquent. L'outil a pour objectif de capitaliser à la fois sur les standards de communication existant, tout en créant des standards là où ils n'existent pas (notamment pour la foule). Le simulateur robot-foule propose une immersion réaliste du robot dans une foule, et des interactions humain-robot aussi vraies que possible grâce à un module d'immersion en réalité virtuelle. Enfin, le simulateur propose des

outils adaptés à l'analyse de la navigation de robot dans la foule. Un défi majeur concerne néanmoins la transférabilité des résultats des tests aux simulations : le succès ou l'échec de la navigation d'un robot dans un environnement de simulation est-il lié à la réalité ? Les mêmes conclusions seraient-elles tirées dans des tests réels ?

Chapitre

4 Simulation de robots

Sommaire

Ce chapitre est centré sur la description de la simulation des robots eux-mêmes. Si ce chapitre ne présente pas de contribution scientifique, il décrit en détails le partie du simulateur présenté quant à lui au chapitre 3 pour ce qui concerne la simulation de robots parmi la foule. Nous apportons ici des précisions sur les modèles des robots que nous employons ainsi que des techniques utilisée pour la simulations de leurs capteurs.

4.1 Standardisation des modèles de robot

Simuler un robot consiste à modéliser principalement deux catégories d'éléments :

— la mécanique et la cinématique du robot : c'est la définition mécanique du robot qui lui permet d'interagir avec son environnement ;

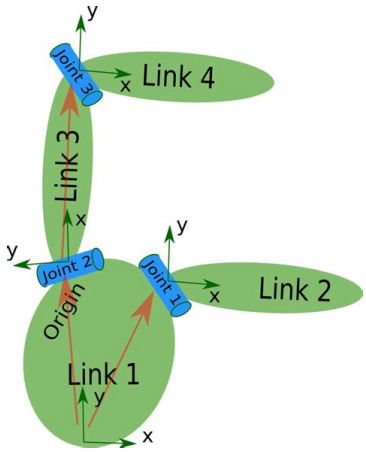


Figure 4.1 – Standard *URDF* : Illustration de la description cinématique d'un robot

— les capteurs du robot : ils sont le moyen pour le robot d'acquérir des informations sur son environnement.

Ces éléments forment ensemble un modèle de robot complet pour notre simulation.

Dans cette thèse, nous restons génériques en termes de robots. En effet, nous considérons que la navigation de robots dans la foule est un problème de robotique mobile d'ordre général qui doit être agnostique au robot utilisé, et nous ne nous contraignons pas à un modèle de robot en particulier. Pour autant, nous nous restreignons au cas des robots mobiles et pour cela, les conditions suivantes doivent être respectées :

— être un robot mobile : sa base n'est donc pas fixe ;

— être un robot capable de se repérer dans l'espace grâce à des capteurs embarqués ;

— être un robot capable de détecter des obstacles environnants grâce à des capteurs embarqués.

Nous utilisons ainsi ce principe générique pour modéliser différents robots pouvant entrer dans le cadre de cette thèse. Pour ce faire, nous utilisons des fichiers *URDF* pour *Unified Robot Description Format*. Le standard *URDF* est une spécification XML permettant de décrire un robot. La description *URDF* d'un robot se compose alors d'un ensemble d'éléments de liaison (ou membres) et d'un ensemble d'éléments d'articulation reliant les liaisons entre elles, comme l'illustre la figure 4.1.

La spécification *URDF* couvre :

— la représentation visuelle du robot ;

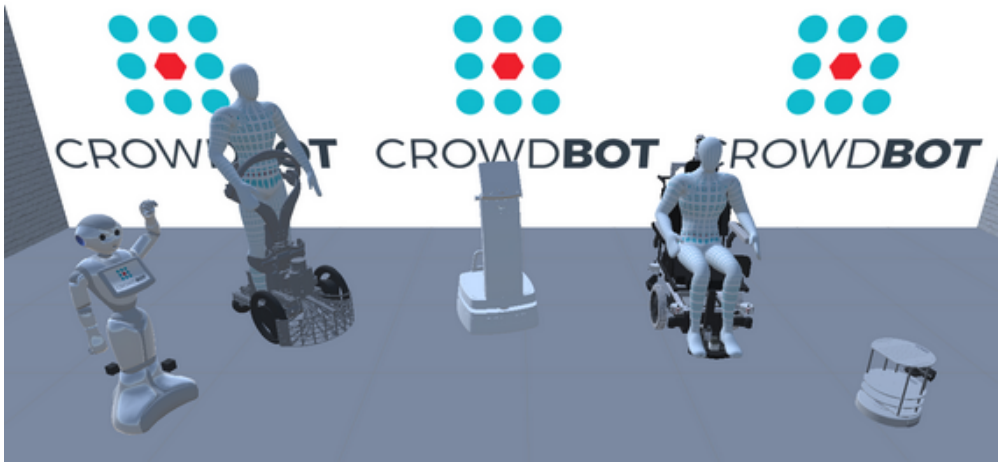


Figure 4.2 – Les différents robots implémentés dans le simulateur. De gauche à droite : **Pepper**, un robot humanoïde qui doit naviguer dans une foule dense tout en s'approchant activement des personnes pour les aider, **Qolo**, un dispositif qui combine des roues motorisées et un exosquelette passif, **CuyBot**, un robot a usage général dédié à un environnement de foule compacte, **Smart Wheelchair**, un fauteuil roulant semi-autonome, qui doit adapter sa trajectoire aux mouvements inattendus des personnes à proximité, et **TurtleBot2** un robot mobile libre de droit conçu par ROS.

— le modèle de collision du robot ;

— la description cinématique du robot.

La limitation principale du standard *URDF* est qu'il ne peut décrire que des robots reposant sur une structure en arbre, ce qui exclut les robots parallèles. Ce n'est pas un problème pour nous : les robots présentés dans cette thèse ont tous une structure en arbre. De plus, la spécification suppose que le robot est constitué de membres rigides reliés par des articulations, les éléments flexibles ne sont donc pas pris en charge. Dans notre cas, ce n'est pas bloquant car nous approchons les membres du robot à des corps rigides. Enfin, cette description implique une bonne connaissance du robot lui même pour obtenir des résultats cohérents en simulation.

4.1.1 Représentation visuelle

La représentation 3D d'un robot, que l'on appelle maillage, est définie dans un fichier qui doit être divisé en plusieurs parties, correspondant aux différents parties du corps du robot.

La figure 4.2 montre les modèles 3D que nous utilisons actuellement dans le simulateur. Pour concevoir ces fichiers de maillage, nous devons utiliser des logiciels de conception assistée par ordinateur (CAO).

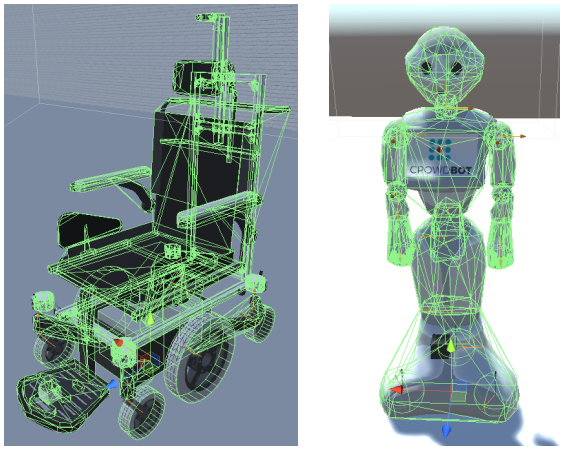


Figure 4.3 – Illustration des maillages de collision pour le fauteuil roulant et pour le robot Pepper.

4.1.2 Modèles de collision

Afin de simuler les interactions physiques (contacts, impacts...) entre les robots et l'environnement, nous utilisons un moteur physique. En particulier, nous utilisons une représentation mathématique de l'objet, appelée *Collider* : le *Collider* s'appuie alors soit sur le même maillage que celui utilisé pour la visualisation (voir section 4.1.1), soit sur un maillage simple qui est composé de formes de base (cube, sphère, cylindre, capsule, plan...). Dans ce deuxième cas, la simulation est beaucoup plus rapide, mais la précision est moindre. Pour nos modèles montrés en figure 4.2, nous avons divisé les maillages visuels en parties élémentaires et utilisé le maillage de ces parties pour définir les éléments *Collider*. Lorsque cela est possible, nous utilisons une forme de base à la place du maillage. Par exemple, le modèle de collision du fauteuil roulant, représenté en vert sur la figure 4.3, est composé de plusieurs *Collider* définis par des maillages (par exemple le maillage de l'accoudoir, le maillage du siège). Les *Collider* des roues utilisent une forme cylindrique de base. Tout ces maillages constituent le modèle de collision du fauteuil roulant. Pour le robot Pepper, nous avons utilisé les maillages de chaque membre du robot, et utilisé des sphères pour les roues.

4.1.3 Description cinématique

Le standard *URDF* décrit les articulations et les membres associé à un robot. Nous donnons ici les principes de définition de ces éléments.

Membres

Pour chaque membre on définit des propriétés mécaniques. Ces propriétés sont utilisées dans le simulateur par l'intermédiaire d'un élément du moteur physique appelé *rigid body* associé au membre. Pour un *rigid body*, on définit sa masse, son frottement linéaire et angulaire, ou encore sa matrice d'inertie. On peut également choisir des options liées au moteur physique, comme le choix entre un comportement dynamique (réagissant aux forces) ou cinématique (ne réagissant pas aux forces).

Articulations

Un articulation connecte deux membres. Elles sont donc nécessaires à la conception du modèle du robot et doivent être bien définies pour la stabilité du modèle.

Nous pouvons définir différents types d'articulations entre 2 membres :

— *Revolute* : une articulation *charnière* tourne le long de son axe et a une plage de rotation limitée spécifiée par les limites supérieure et inférieure.

— *Continuous* : une articulation *continue* tourne autour de l'axe et n'a pas de limite supérieure ni inférieure.

— *Prismatic* : une articulation *coulissante* glisse le long de l'axe et la portée est limitée par les limites supérieure et inférieure.

— *Fixed* : ce n'est pas vraiment une articulation car elle est *fixe* et ne peut pas bouger ; tous les degrés de liberté sont verrouillés.

— *Floating* : Cette articulation *flottante* permet le mouvement des 6 degrés de liberté.

— *Planar* : Cette articulation *planaire* permet le mouvement dans un plan perpendiculaire à l'axe.

Chaque type d'articulation a un grande quantité d'éléments de configuration. Par exemple, pour une articulation charnière comme sur la figure 4.4, nous devons définir les membres connectés et leurs masses, l'axe de rotation, les limites angulaires, l'usage de ressorts ou de moteurs dans l'articulation, ou encore un couple maximum avant casse de l'articulation.

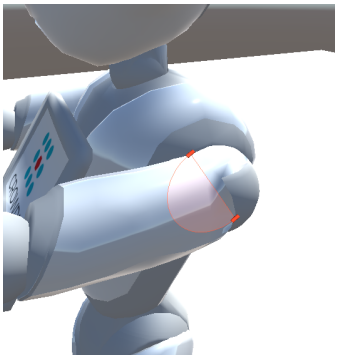


Figure 4.4 – Articulation charnière sur le modèle du robot Pepper.

4.2 Modèle de perception du robot

Nous séparons le côté mécanique du côté perception dans la simulation. Pour intégrer nos capteurs dans le modèle URDF du robot, nous définissons simplement un *rigid body* associé au capteur, constituant alors un membre du robot relié par une articulation *fixe* au membre du robot auquel il est associé, ou à la base du robot par défaut. Ce membre capteur est associé à un script générant les données du capteur.

4.2.1 Simulation de capteurs

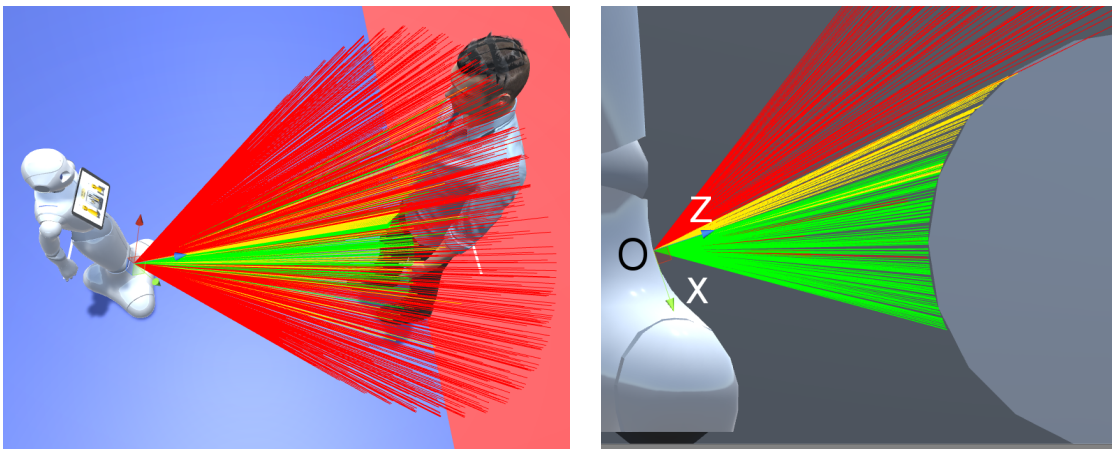
La simulation de capteurs consiste à reproduire le flot de données généré par un capteur réel, reproduisant le même format et fréquence de données, ainsi que ses imprécisions Pour y parvenir, nous devons avoir une bonne connaissance des capteurs du robot, en utilisant la fiche technique ou en testant les capteurs dans de nombreuses situations réelles. Nos capteurs simulés offrent la possibilité de régler les paramètres habituels en fonction des propriétés des capteurs réels, comme la distance de détection maximale et minimale, l'angle d'incidence maximal requis pour la détection, le champ de vision, la fréquence de mise à jour des mesures, ou encore la précision de la mesure...

Par ailleurs, afin de fournir une simulation réaliste des capteurs, nous avons la possibilité d'ajouter différents types de bruit à n'importe quel type de capteur. Nous avons ainsi implémenté un générateur de bruit gaussien, un générateur de décalage et un générateur de pics, qui sont configurables et peuvent être ajoutés à la sortie de n'importe quel type de capteurs décrits ci-dessous.

Les données des capteurs du robot sont générées grâce aux éléments 3D de l'environnement de simulation : la scène, et les avatars humains. Nous avons implémenté des simulation pour capteurs de proximité infrarouge (IR), à ultrasons

(US), des capteurs laser 2D (LIDAR) et une caméra de profondeur (RGB-D). L'implémentation des capteurs simulés utilise les capacités de *Unity* en termes de moteur physique et de calcul (GPU).

Pour les US, l'IR et le LIDAR, les capteurs simulés émettent un nombre prédéfini de rayons dans une zone de détection donnée. Ces rayons sont générés par le moteur physique qui renvoie un point de collision lorsqu'un rayon frappe un *Collider* (voir section 4.1.2. Les rayons sont lancés de manière uniforme dans la zone de détection, en fonction du nombre prédéfini de rayons. La figure 4.5 donne une illustration de la simlulation d'un capteur US. Pour un résultat plus réaliste, nous proposons de simuler le fait que ces capteurs ne détectent pas les obstacles si l'angle d'incidence est trop élevé, en donnant la possibilité de spécifier un angle d'incidence maximal. Si le rayon frappe un objet sur sa trajectoire, et que l'angle d'incidence est inférieur à cette valeur, alors l'objet est détecté et la distance entre le capteur et le point de mire est enregistrée.



a) b)

Figure 4.5 – Illustration de la simulation de capteur US par projection de rayons. a) Le robot Pepper est équipé d'un capteur US, la zone de detection a la forme d'une larme. Des rayons sont projetés en partant du capteur pour couvrir cette zone. b) Les rayons montrés en vert ont détecté un obstacle, les rayons en jaune ont détecté un obstacle avec un angle d'incidence trop élevé (ces rayons sont donc ignorés), et les rayons rouges n'ont pas détecté d'obstacles

Les capteurs US ont une zone de détection spécifique, que nous proposons d'approcher par une fonction de larme centrée sur le capteur. Le capteur US simulé renvoie la plus petite valeur observée de l'ensemble des rayons. Nous proposons d'approximer la zone de détection d'un capteur IR par un cône centré sur le capteur. Il est alors possible de configurer la distance maximale de détection, l'angle du cône et le niveau d'échantillonnage.

Nous proposons de simuler les capteurs LIDAR en lançant des rayons selon une

sphère partielle pour laquelle il est possible de configurer la résolution angulaire, les angles de départ et d'arrivée, ainsi que la portée minimale et maximale du LIDAR.

Les caméras RGB sont simulées quant à elle à l'aide de simples images extraites de *Unity*. La composante de profondeur de la caméra RGB-D est simulée à l'aide de *shaders Unity*, qui sont généralement chargés de rendre les textures et ont été détournés ici pour cet usage. Nous les utilisons pour générer une texture de profondeur, qui nous donne une image en niveau de gris où chaque valeur de pixel correspond à la valeur de profondeur. De plus, la caméra simulée est hautement configurable (*e.g.* champ de vision, résolution...). Cette méthode nous permet surtout de déporter les calculs sur la carte graphique de l'ordinateur afin de gagner en vitesse de simulation.

4.3 Conclusion

Dans ce chapitre, nous avons détaillé nos techniques de simulation de robots, dont en particulier nos techniques de simulation de capteurs de robots. La simulation de ces capteurs présente différents avantages, dont celle de pouvoir tester les techniques de localisation et de cartographie dans des environnements bondés, mais aussi les techniques de détection et de suivi de personne par exemple, sur la base de données capteurs synthétiques. Notons que la synthèse de données capteurs incorporant la présence d'humain peut présenter un très grand intérêt pour la création de jeu de données d'entraînement pour une utilisation pour entraîner des systèmes à différentes tâches.

Chapitre

5 Unification d'algorithmes de simulation de foule

Sommaire

La simulation de foule microscopique consiste à simuler le déplacement d'une foule d'agents en contrôlant les trajectoires de chaque individu indépendamment. Ce contrôle individuel tient compte des objectifs de chaque agent, sa destination, ainsi que des interactions locales entre individus, comme par exemple les évitements de collision. De nouveaux algorithmes microscopiques pour la simulation

de foule apparaissent régulièrement dans la littérature, chacun ayant ses propres avantages et inconvénients, et reposant sur des principes différents. De plus, le paramétrage et les détails d'implémentation peuvent avoir un impact important sur le comportement général du système chaotique qu'est la foule simulée. Il est donc pratiquement impossible de comparer les algorithmes au niveau conceptuel, ou d'apporter des affirmations générales sur la qualité d'une méthode. Par ailleurs, il peut être intéressant de combiner plusieurs simulateurs de foule afin de tirer le meilleur parti des forces de chaque algorithme, en créant des algorithmes hybrides. En effet, chaque algorithme a son propre domaine de validité, comme par exemple la plage de densité sur lequel il fonctionne, ou la nature des interactions locales qu'il est capable de reproduire. Apparaît alors un besoin évident d'un cadre qui utilise un principe unique pour reproduire l'essence du plus grand nombre possible d'algorithmes. Un tel cadre serait utile pour comprendre de manière plus approfondie les différences entre les algorithmes, pour les comparer de façon honnête et pour orienter les recherches futures, comme par exemple des simulations de foules hybrides utilisant plusieurs algorithmes différents pour le comportement d'une foule ou pour le comportement d'un agent qui pourrait ainsi s'adapter à différentes situations.

5.1 Contributions

Nous présentons ici deux contributions [1].

Premièrement, nous montrons que de nombreux algorithmes pour la navigation locale peuvent être reformulés en tant que méthodes qui *optimisent une fonction de coût dans un espace de vitesses*. Cela signifie concrètement que l'on peut traduire l'objectif de chaque agent (destination, direction souhaitée de déplacement, vitesse désirée de déplacement, etc...) ainsi que ses contraintes locales (évitement de collision, déplacement avec un groupe, etc...) sous la forme d'une fonction qui associe un coût à chacune des valeurs de vitesses atteignables. Le principe de cette fonction est d'obtenir des coûts minimaux aux vitesses qui satisfait aux objectifs et contraintes. Cette fonction est directement utile au calcul du mouvement de l'agent, en ajustant sa vitesse de déplacement pour qu'elle atteigne un coût mini-

1. Fait l'objet d'une publication à la conférence I3D 2020 : *van Toll, W., Grzeskowiak, F., Gandía, A. L., Amirian, J., Berton, F., Bruneau, J., ... & Pettré, J. (2020, May). Generalized Microscropic Crowd Simulation using Costs in Velocity Space. In Symposium on Interactive 3D Graphics and Games (pp. 1-9).*

mum, que ce minimum soit global ou local.

Nous étudions donc ici la possibilité de reproduire une majorité d'algorithmes existants en établissant :

1. une fonction de coût qui traduit le principe utilisé pour le contrôle des agents ;
2. une méthode d'optimisation appliquée à la fonction de navigation pour calculer un vecteur d'accélération pour l'agent.

Ainsi, on obtient un cadre général qui peut reproduire de nombreux algorithmes de navigation locale en faisant certains choix pour ces deux éléments.

Deuxièmement, nous présentons un logiciel nommé *UMANS*[2], qui utilise ces principes pour intégrer diverses méthodes de navigation locale dans un seul système. Nous montrons que *UMANS* peut en effet reproduire de nombreux algorithmes microscopiques existants. Le logiciel permet de comparer facilement les algorithmes et les réglages de paramètres dans des scénarios spécifiés par l'utilisateur. Par rapport à d'autres logiciels de simulation de foule [Curtis et al., 2016, Kielar et al., 2016, van Toll et al., 2015, Singh et al., 2009], nous nous concentrons uniquement ici sur la navigation locale. On peut définir la navigation locale comme étant une fonction qui ajuste localement une navigation globale (par exemple établie par un planificateur de chemin) pour prendre en compte d'éventuels changements dans l'environnement (comme des obstacles dynamiques). Ainsi, dans le cadre de la simulation de foule, à l'instar de la robotique mobile, la navigation locale consiste à adapter le déplacement de l'agent (ou du robot) à la présence d'agents voisins. Par conséquent, notre travail pourrait être considéré comme une composante d'un système plus vaste. En ramenant les fonctions de navigation locales différentes à une définition, nous cherchons à comprendre véritablement les différences entre les algorithmes microscopiques, en exprimant ces algorithmes dans des termes similaires et en unifiant autant d'autres paramètres que possible.

5.2 Principe unificateur des algorithmes de navigation de foule microscopiques

Cette section décrit notre outil de simulation de foule microscopique. D'abord, on définit la boucle de simulation, suivi du schéma général suivi pour la navigation

2. *UMANS* est conçu comme une plate-forme partagée, et son code source est disponible en ligne : https://project.inria.fr/crowdscience/project/ocsr/umans/.

locale à travers l'utilisation de fonctions de coût et de méthodes d'optimisation. Comme il est d'usage dans le domaine de la simulation de foule, nous approximons chaque agent par un disque d'un rayon donné. Les agents peuvent être ajoutés à la simulation ou supprimés au fil du temps. Nous supposerons que l'environnement est un plan 2D sans limites et sans obstacles. Notons que tous les concepts peuvent être étendus pour traiter les obstacles polygonaux et les limites de l'environnement, sans changement fondamental.

5.2.1 Boucle de simulation

Nous utilisons une boucle de simulation avec des pas de temps de longueur fixe Δt. La simulation s'exécute en temps réel si chaque pas de temps prend au maximum Δt secondes réelles pour être calculée. Sur l'ordinateur, un pas de temps se compose des étapes suivantes :

1. Créer un hachage spatial contenant toutes les positions des agents, pour faciliter les calculs de voisinage.
2. Pour chaque agent A_j, trouver tous les agents à moins de r_N mètres de A_j, où r_N est un paramètre de la simulation. Ce sont les agents voisins que A_j prendra en compte lors de la navigation locale (à l'étape 4).
3. Pour chaque agent A_j, calculer une vitesse préférée $\mathbf{v}_{\text{pref}}$. Dans notre implémentation, c'est la vitesse qui permettrait à A_j d'atteindre directement son objectif $\mathbf{g}$ à une vitesse préférée s_{pref}. Cette vitesse pourrait être étendue à une vitesse qui suit une trajectoire globale, mais dans ce document, nous nous concentrons délibérément sur l'aspect local uniquement.
4. Pour chaque agent A_j, effectuer un algorithme de navigation locale au choix de l'utilisateur. Ceci induit un vecteur d'accélération $\mathbf{a}$, prescrivant un changement de vitesse. Nous parlerons de cette étape plus en détail dans la section 5.2.2.
5. Pour chaque agent A_j, calculer un vecteur de force de contact $\mathbf{F}_c$ résultant des collisions qui ont lieu à cet instant de la simulation. Cela permet de traiter des collisions résiduelles et ainsi éviter que des agents se superposent.
6. Pour chaque agent A_j, mettre à jour la vitesse et la position via l'intégration d'Euler :

$$\mathbf{v} := \mathit{clamp}(\mathbf{v} + \mathit{clamp}(\mathbf{a}, a_{\max}) \cdot \Delta t, s_{\max}) + \mathbf{F}_c/m \cdot \Delta t,$$

$$\mathbf{p} := \mathbf{p} + \mathbf{v} \cdot \Delta t,$$

où m est la masse de l'agent, s_{max} et a_{max} représentent sa vitesse et son accélération maximales, et $clamp(\mathbf{x}, L)$ fixe un vecteur $\mathbf{x}$ à une longueur maximale L. Il est à noter que, techniquement, d'autres schémas d'intégration pourraient également être utilisés. Nous avons choisi la méthode Euler car c'est le choix le plus courant dans le domaine de la simulation de foule.

5.2.2 Le principe unificateur pour les algorithmes de navigation locale

Nous décrivons maintenant en détails l'étape 4 de la boucle de simulation : la navigation locale. Les deux principales composantes de cette étape sont les *fonctions de coût* et les *méthodes d'optimisation*.

5.2.2.1 Fonctions de coût

Pour un agent A_j, à un instant donné de la simulation, l'objectif d'un algorithme de navigation locale est de calculer comment A_j devrait contrôler sa vitesse pour l'étape de simulation suivante. L'idée est que l'agent doit rester proche de sa vitesse préférée tout en respectant les règles locales, telles que l'évitement des collisions avec les obstacles proches et les autres agents. Pour ce faire, on définit *l'espace de vitesse* $\mathcal{V} \subseteq \mathbb{R}^2$ l'ensemble de toutes les vitesses atteignables que les agents peuvent avoir. Les deux axes de $\mathcal{V}$ sont les coordonnées x et y des vitesses. Ainsi, $\mathcal{V}$ peut être pensé comme un disque autour de $(0, 0)$ avec un rayon de la vitesse maximale $s_{\max}$.

Ensuite, on définit une *fonction de coût* $\mathcal{C} : \mathcal{V} \to \mathbb{R}$ comme une fonction qui attribue à chaque vitesse possible $\mathbf{v}' \in \mathcal{V}$ un coût scalaire $\mathcal{C}(\mathbf{v}')$. Plus ce coût est faible, plus il est pertinent de choisir $\mathbf{v}'$ comme la prochaine vitesse de l'agent. Le coût d'une vitesse pour un agent A_j peut être calculé sur la base de différentes informations, comme la position actuelle de l'agent et sa vitesse courante, sa vitesse préférée $\mathbf{v}_{\text{pref}}$ la position de son objectif de navigation $\mathbf{g}$, et les positions et vitesses des autres agents ou des obstacles autour de A_j.

Ces informations nécessitent l'usage de plusieurs métriques qui sont souvent utilisées dans une fonction de coût $\mathcal{C}$:

Distance *dist*(A_j, A_k) : la distance euclidienne actuelle en mètres entre les agents A_j et A_k.

Temps avant collision (Time to collision) *ttc*$(\mathbf{v}', A_j, A_k)$: le temps en secondes après lequel l'agent A_j entrera en collision avec un autre agent A_k,

en supposant que A_j a pour vitesse $\mathbf{v}'$ et que A_k utilise sa vitesse actuelle telle qu'elle est observée par A_j.

Distance avant collision (Distance to collision) $\boldsymbol{dc}(\mathbf{v}', A_j, A_k)$ **:** la distance jusqu'au point où A_j et A_k entreront en collision selon les mêmes hypothèses, et qui s'écrit
$ttc(\mathbf{v}', A_j, A_k) \cdot ||\mathbf{v}'||$.

Temps de point d'approche minimale (Time to closest approach) $\boldsymbol{ttca}(\mathbf{v}', A_j, A_k)$ **:** le moment où la distance entre A_j et A_k sera la plus petite. Ce temps est égal à $ttc(\mathbf{v}', A_j, A_k)$ si les agents sont censés entrer en collision.

Distance au point d'approche minimale (Distance of closest approach) $\boldsymbol{dca}(\mathbf{v}', A_j, A_k)$ **:** la plus petite distance prédite entre A_j et A_k si leurs vitesses sont constantes, c'est-à-dire leur distance après $ttca(\mathbf{v}', A_j, A_k)$ secondes. Cette distance est égale à zéro si les agents entrent en collision.

Les équations exactes de ces mesures se trouvent dans la littérature, et notre logiciel en propose une implémentation. Chaque métrique peut également être définie pour un obstacle voisin (statique) au lieu d'un agent voisin. De plus, pour chaque concept, nous pouvons définir une version qui prend le temps ou la distance minimum entre tous les agents voisins et les obstacles. Nous les indiquerons en utilisant des lettres majuscules : $TTC(\mathbf{v}', A_j)$, $DC(\mathbf{v}', A_j)$, $TTCA(\mathbf{v}', A_j)$, et $DCA(\mathbf{v}', A_j)$.

Il convient de noter que la fonction de navigation n'est pas nécessairement régulière, et son gradient $\nabla\mathcal{C}$ peut ne pas être bien défini. Par exemple, si $\mathcal{C}$utilise TTC ou DC, le coût de deux vitesses similaires $\mathbf{v}'$ et $\mathbf{v}''$ peuvent être très différents si une vitesse provoque une collision alors que l'autre l'évite. En revanche, $TTCA$ et DCA changent progressivement, c'est pourquoi elles sont préférées par les algorithmes qui s'appuient sur le gradient de $\mathcal{C}$.

5.2.2.2 Méthodes d'optimisation

La vitesse de l'agent est continuellement adaptée pour minimiser la valeur de $\mathcal{C}$. Il existe plusieurs façons d'optimiser la prochaine vitesse de l'agent minimisant $\mathcal{C}$. Dans la littérature, nous identifions deux options :

La descente de gradient : en partant de la vitesse actuelle $\mathbf{v}$de l'agent, on se déplace dans la direction opposée du gradient $\nabla\mathcal{C}$. Il en résulte un vecteur d'accélération $\mathbf{a} = -\nabla\mathcal{C}(\mathbf{v})$. Naturellement, cette approche nécessite que $\nabla\mathcal{C}(\mathbf{v})$ soit bien définie et calculable.

L'optimisation globale : on trouve une vitesse $\mathbf{v}^* \in \mathcal{V}$ avec un coût minimal. La recherche peut être limitée à un sous-ensemble de $\mathcal{V}' \subseteq \mathcal{V}$, en donnant à

toutes les vitesses en dehors de $\mathcal{V}'$ un coût infini. Finalement, on convertit la vitesse résultante $\mathbf{v}^*$ en une accélération selon

$$\mathbf{a} = (\mathbf{v}^* - \mathbf{v}) / \max(\tau, \Delta t),$$

où τ est un temps de relaxation en secondes limitant le changement de vitesse de l'agent. Certaines méthodes permettent aux agents d'utiliser immédiatement la vitesse $\mathbf{v}^*$. Quand c'est le cas, alors on fixe $\tau = 0$.

Selon la fonction de coût, l'optimisation globale peut ne pas avoir de solution analytique. Certaines implémentations se rapprochent donc de la vitesse optimale par échantillonnage de multiples vitesses "candidates" (à partir du sous-ensemble pertinent $\mathcal{V}'$) et en choisissant celui qui a le coût le plus bas.

5.3 Application du principe unificateur à des algorithmes existants

Cette section montre comment les principaux algorithmes de l'état de l'art peuvent s'exprimer à l'aide d'une fonction de coût $\mathcal{C}$et d'une méthode d'optimisation locale. Nous proposons donc notre propre implémentation de ces algorithmes en suivant notre principe unificateur, même lorsque ces algorithmes sont *a priori* basés sur des principes différents. Le tableau 5.1 donne, pour ces principaux algorithmes, une fonction de coût et une méthode d'optimisation avec lesquels nous pouvons reproduire exactement ou de manière approximée le comportement de ces algorithmes.

5.3.1 Forces

Les méthodes de navigation locale utilisant des forces [Helbing and Molnár, 1995, Reynolds, 1999, Karamouzas et al., 2009, 2014] considèrent les agents comme des particules qui exercent des forces pour se repousser entre elles. De plus, chaque agent est mû par une force l'attirant vers son but. Ainsi, à tout moment de la simulation, chaque agent A_j est sujet à un certain nombre de forces $\{\mathbf{F}_1, \ldots, \mathbf{F}_k\}$ qui résultent en une force $\mathbf{F}_{\text{total}}$. De cette force, on en déduit une accélération $\mathbf{a}$ selon $\mathbf{a} = \mathbf{F}_{\text{total}}/m$ où m est la masse de l'agent (deuxième loi de Newton).

Dans notre implémentation, on peut se contenter d'utiliser la méthode d'optimisation par gradient. En effet, ce principe peux être imité en définissant une fonction de coût pour laquelle le gradient a la propriété suivante : $\nabla\mathcal{C}(\mathbf{v}) = -\mathbf{F}_{\text{total}}/m$. En théorie, seul le gradient de la fonction de coût est utile pour choisir la vitesse

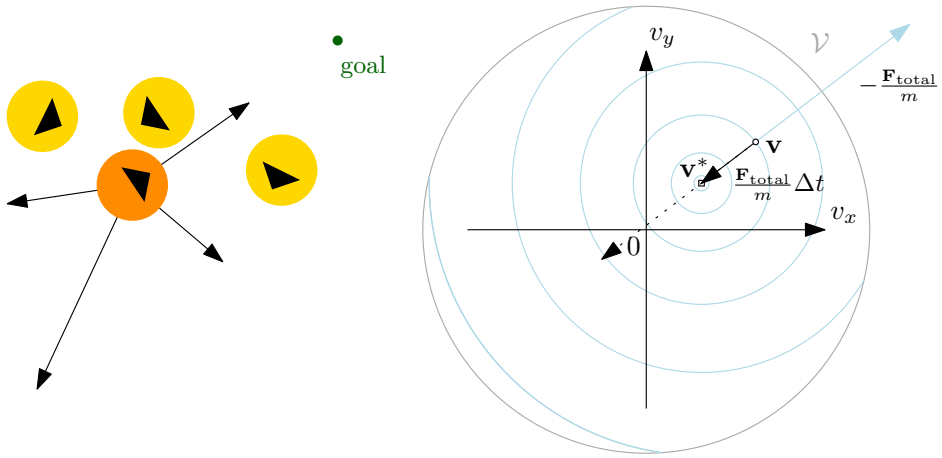


Figure 5.1 – Application du principe unificateur aux méthodes de navigation utilisant des forces. Gauche : un agent est soumis à des forces exercées par d'autres agents, et par son objectif (*goal*). Droite : Le coût $\mathcal{C}(\mathbf{v}')$ dépend de la distance entre $\mathbf{v}'$ et la vitesse $\mathbf{v}^*$ obtenue par application de la deuxième loi de Newton.

à l'instant suivant, et il n'est pas nécessaire d'aller plus loin, à moins de vouloir utiliser une méthode d'optimisation différente. Dans ce cas, il est alors utile de définir une fonction $\mathcal{C}$ qui ait du sens pour les autres vitesses également.

Nous proposons donc la forme suivante. Soit $\mathbf{v}^* = \mathbf{v} + \mathbf{F}_{\text{total}}/m \cdot \Delta t$ la vitesse que l'agent aurait si on le soumettait à $\mathbf{F}_{\text{total}}$ à un instant de simulation donné. Alors, pour toute vitesse $\mathbf{v}' \in \mathcal{V}$, on définit la fonction de coût par

$$\mathcal{C}(\mathbf{v}') = \frac{1}{2\Delta t} \left\| \mathbf{v}' - \mathbf{v}^* \right\|^2, \quad \nabla\mathcal{C}(\mathbf{v}') = \frac{1}{\Delta t}(\mathbf{v}' - \mathbf{v}^*).$$

Autrement dit, le coût dépend de la distance à $\mathbf{v}^*$, et le gradient s'éloigne de $\mathbf{v}^*$avec une amplitude telle que la condition $\nabla\mathcal{C}(\mathbf{v}) = -\mathbf{F}_{\text{total}}/m$ est satisfaite. La figure 5.1 illustre le phénomène.

5.3.2 Les algorithmes sélectionnant parmi des échantillons de vitesses

La catégorie d'algorithmes principaux qui ont pour principe de sélectionner une vitesse parmi un échantillon de vitesses [Karamouzas and Overmars, 2010, Moussaïd et al., 2011] définissent explicitement une fonction de coût $\mathcal{C}$ et approximent le minimum de cette fonction. En pratique, ces algorithmes testent un nombre fixé de vitesses sélectionnées aléatoirement dans l'espace des vitesses $\mathcal{V}$, et limitées par un "angle de vue" devant l'agent.

Pour obtenir un mouvement fluide, ces algorithmes peuvent utiliser un temps de relaxation $\tau > 0$, ou bien utiliser la différence entre la vitesse candidate $\mathbf{v}'$(i.e.

celle sélectionnée par l'algorithme) et la vitesse courante $\mathbf{v}$dans leur fonction de coût.

La figure 5.2 montre un exemple d'algorithme échantillonnant dans un cône de 180 degrés de façon régulière.

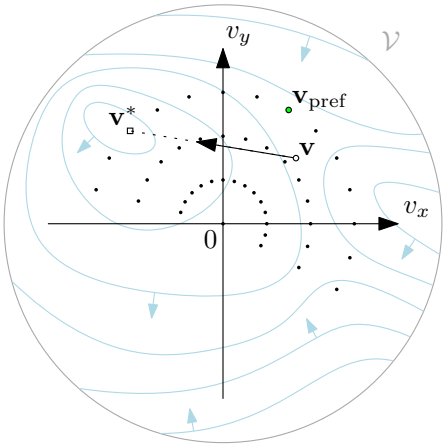


Figure 5.2 – Application du principe unificateur pour une méthode par échantillonnage des vitesses typique. En bleu clair, on représente le gradient et la valeur de la fonction coût (arbitraire sur ce schéma d'illustration).

Le tableau 5.1 précise les détails d'implémentation pour les algorithmes Karamouzas and Overmars [2010], Moussaïd et al. [2011]. Par exemple, Karamouzas and Overmars [2010] n'utilisent pas les vitesses admissibles $\mathcal{V}$ mais un sous-ensemble dépendant de $TTC(\mathbf{v}_{\text{pref}})$. Pour Karamouzas and Overmars [2010], la méthode sélectionne quant à elle la vitesse en deux temps : d'abord, l'angle optimal est sélectionné, puis on ajuste l'amplitude de la vitesse choisie à s_{pref}. Cette amplitude est réduite si une collision est imminente. Nous reproduisons ce comportement en deux temps en définissant une fonction de coût unique qui se présente de la façon suivante :

— $K(\mathbf{d})$ est le coût pour une vitesse $\mathbf{v}$ de direction $\mathbf{d}$ d'amplitude s_{pref}. Autrement dit, $\mathbf{v} = \mathbf{d} \cdot s_{\text{pref}}$.

— $S(\mathbf{d})$ est la vitesse optimale dans la direction $\mathbf{d}$ telle qu'elle est définie dans l'article original, calculée à partir de la distance à la première collision $DC(\mathbf{d} \cdot s_{\text{pref}}, A_j)$.

— Le coût d'une vitesse candidate $C(\mathbf{v}')$ utilise $K(\frac{\mathbf{v}'}{\|\mathbf{v}'\|})$, i.e. le coût directionnel de $\mathbf{v}'$tel qu'il est défini dans l'article original. On multiplie ce coût par un facteur de mise à l'échelle dépendant de $S(\frac{\mathbf{v}'}{\|\mathbf{v}'\|})$ pour garantir le fait que le minimum global de $\mathcal{C}$soit bien obtenu pour la vitesse qui aurait été choisie si on avait utilisé directement la méthode décrite dans l'article.

5.3.3 Les algorithmes basés sur le calcul des vitesses admissibles

Les algorithmes basés sur le calcul des vitesses admissibles (*"Velocity obstacles") en anglais* comme ceux de Berg et al. [2008], van den Berg et al. [2011b], Guy et al. [2010] définissent $\mathcal{V}_{\text{obs}}$ l'ensemble des vitesses pour l'agent A_j qui conduiraient à une collision dans le futur proche. Ces algorithmes choisissent ensuite une vitesse admissible dans l'ensemble des vitesses $\mathcal{V}$ privé du sous-ensemble $\mathcal{V}_{\text{obs}}$, autrement dit $\mathbf{v}^* \in \mathcal{V} \setminus \mathcal{V}_{\text{obs}}$. En général, cette vitesse est choisie pour être la plus proche possible de la vitesse préférée $\mathbf{v}_{\text{pref}}$. D'autres critères peuvent être utilisés comme c'est le cas pour Guy et al. [2010] qui utilisent le principe de moindre effort pour sélectionner la nouvelle vitesse de l'agent $\mathbf{v}^*$. Pour appliquer le principe unificateur à cette catégorie d'algorithmes, on introduit la notion d'ensemble de vitesses non-admissibles $\mathcal{V}_{\text{obs}}$ en attribuant un coût infini pour ces vitesses. La figure 5.3 montre un exemple d'application du principe unificateur à un algorithme de type "velocity obstacle".

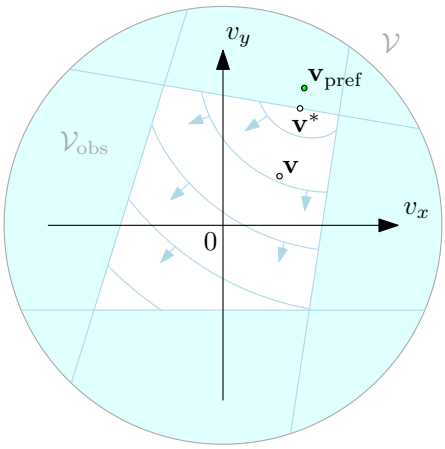


Figure 5.3 – Application du principe d'unification à un algorithme utilisant l'ensemble des vitesses admissibles. La méthode défini un espace des vitesses interdites $\mathcal{V}_{\text{obs}}$ (en bleu clair) et une fonction de coût. On donne un coût infini aux vitesses dans la zone interdite. La vitesse optimale $\mathbf{v}^*$, c'est à dire la plus proche de $\mathbf{v}_{\text{pref}}$ peut être calculée analytiquement.

En outre, certaines de ces méthodes utilisent le principe de réciprocité. Elles considèrent que lors d'une potentielle collision entre deux agents A_j et A_k, l'effort à fournir pour l'évitement est partagé entre les deux agents. Le précurseur du principe, RVO pour "Reciprocal Velocity Obstacles" Berg et al. [2008], effectue une moyenne entre la vitesse courante $\mathbf{v}$ et la vitesse optimale $\mathbf{v}^*$L'algorithme ORCA pour "Optimal Reciprocal Collision Avoidance" van den Berg et al. [2011b] en améliore le principe : pour chaque agent du voisinage A_k, on définit pour A_j un

demi-plan dans l'espace des vitesses, qui contient toutes les vitesses d'évitement dites admissibles pour A_j. Le calcul de la frontière de ce demi-plan est fait avec le principe que les efforts d'évitement sont partagés de façon équitable. On sélectionne alors la vitesse la plus proche de $\mathbf{v}_{\text{pref}}$ qui appartient à tout les demi-plans ORCA générés par les agents voisins. Dans le cas de ORCA, il est possible que tout l'espace des vitesses soit interdit, notamment dans le cas de foules denses. Dans ce cas, ORCA cherche l'option "la moins mauvaise" par le biais d'une fonction de coût. Cela peut être par exemple la vitesse qui limite le plus l'énergie à l'impact. Nous implémentons dans notre logiciel cette solution de repli en spécifiant une seconde fonction de coût $\mathcal{C}_2$ qui se déclenche dès que la fonction de coût primaire $\mathcal{C}_1$ ne trouve pas de vitesse ayant un coût fini. Bien qu'il soit possible d'avoir une unique fonction de coût en fusionnant $\mathcal{C}_1$ et $\mathcal{C}_2$, la solution la plus convaincante pour imiter ORCA reste de bien séparer ces deux fonctions.

Notre logiciel permet ainsi aux utilisateur d'implémenter leur propres fonctions de coût ainsi que leurs propres méthodes d'optimisation. La solution d'optimisation par échantillonnage reste toujours présente par défaut ou en cas d'échec de la fonction d'optimisation de l'utilisateur, afin de rester générique et de permettre aux utilisateur de tester des fonctions de coût sans nécessairement implémenter de méthode d'optimisation.

5.4 Les algorithmes basés sur des descentes de gradient

Certains algorithmes calculent la nouvelle vitesse des agents avec le gradient d'une fonction de coût de manière intrinsèque Ondřej et al. [2010], Dutra et al. [2017], López et al. [2019a]. Ces algorithmes utilisent des images synthétiques représentant le champ de vision de l'agent ou son flux optique, définissent un coût par pixel, et somment les gradient de ces coûts par pixel pour obtenir une nouvelle vitesse. Toutefois, la fonction de coût $\mathcal{C}$ est trop complexe pour être calculée à chaque pas de temps (les simulation ne seraient pas temps-réel). Le gradient est donc calculé analytiquement et utilisé directement dans la méthode d'optimisation par pas de gradients, sans calculer les résultats de la fonction de coût directement.

Pour adapter ces algorithmes à notre méthode, il faut d'abord changer de système de coordonnées. Ces algorithmes étant basés sur la vision d'un agents, il est logique de travailler en coordonnées polaires (vitesse linéaire s et vitesse angulaire θ exprimées dans le repère de l'agent A_j). Cependant, notre méthode

utilise des coordonnées euclidiennes (vitesses en x et vitesse en y, dans le repère global de l'environnement). En conséquence, la fonction de navigation $\mathcal{C}$ n'est pas modifiée mais la forme analytique de son gradient $\nabla\mathcal{C}$ est différente. En effet, en coordonnées polaires, le gradient est de la forme

$$\nabla_{s,\theta}\mathcal{C} = \left(\frac{\partial\mathcal{C}}{\partial\theta}, \frac{\partial\mathcal{C}}{\partial s}\right). \tag{5.1}$$

Nous transformons cette équation en coordonnées Euclidiennes dans le repère agent, et obtenons

$$\nabla_{\mathbf{v}'}\mathcal{C} = \left(s\sin\frac{\partial\mathcal{C}}{\partial\theta}, s\left(1-\cos\frac{\partial\mathcal{C}}{\partial\theta}\right) + \frac{\partial\mathcal{C}}{\partial s}\right). \tag{5.2}$$

Finalement, par le biais d'une matrice de passage, nous utilisons la forme en coordonnées cartésiennes dans le repère global définie par

$$\nabla\mathcal{C}(\mathbf{v}') = \frac{1}{\|\mathbf{v}'\|}\begin{bmatrix} v'_y & v'_x \\ -v'_x & v'_y \end{bmatrix}\nabla_{\mathbf{v}'}\mathcal{C}(\mathbf{v}'), \tag{5.3}$$

où v'_x et v'_y sont les coordonnées ce $\mathbf{v}'$dans le repère global. Pour un gradient défini en coordonnées polaires (rapidité et angle), cette conversion donne le gradient $\nabla\mathcal{C}$ en coordonnées cartésiennes.

Une spécificité de l'ensemble de ces algorithmes est d'utiliser la synthèse d'image pour évaluer la fonction $\mathcal{C}$. Cela signifie que $\mathcal{C}$n'est pas définie uniquement sur la base de l'état de l'agent et de ses voisins, mais que ses termes sont évalués pour un ensemble de points de l'environnement "visibles" de l'agent. C'est pour cette raison qu'on dit que ces approches sont "basées vision" dans la section 2.2.

Notre implémentation approxime l'influence de cette synthèse sur le comportement sensori-moteur des agents, rendant le code plus portable et plus générique pour être utilisé avec d'autre méthodes d'optimisation (comme l'échantillonnage).

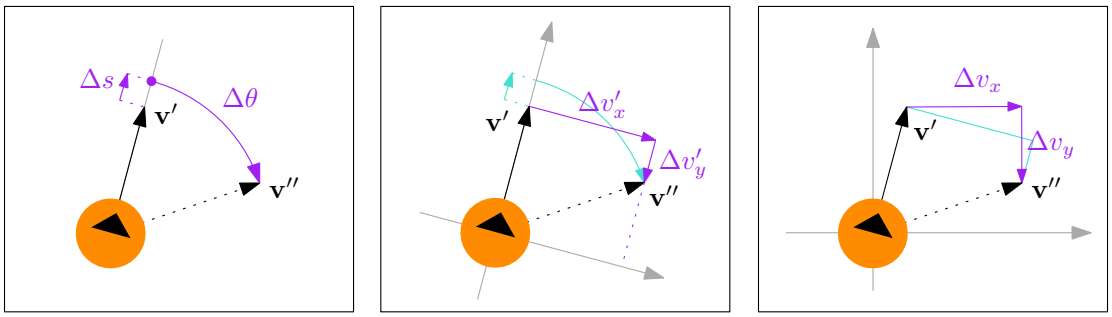


Figure 5.4 – Les trois systèmes de coordonnées : angle et amplitude par rapport à $\mathbf{v}'$ (gauche), coordonnées cartésiennes par rapport à $\mathbf{v}'$ (centre), et coordonnées cartésiennes globales (droite) affectant la forme du gradient.

Table 5.1 – Aperçu des algorithmes de navigation locale et de leur traduction dans notre cadre. Dans les fonctions de coût, nous avons omis toute constante non pertinente, nous avons renommé les paramètres de poids en w_a, w_b, w_c, w_d, et les seuils de temps/distance à $t_{min}, t_{max}, d_{max}$. Pour plus de clarté, ces poids et seuils sont représentés en bleu. En outre, n est le nombre de voisins qu'un agent considère, $\angle$ représente l'angle (en radians) entre deux vecteurs, et $\hat{\mathbf{x}}$ désigne la version normalisée d'un vecteur $\mathbf{x}$.

Algorithme	**Fonction de coût $\mathcal{C}$ (+ fonction de repli $\mathcal{C}_2$)**	**Méthode d'optimisation**
Helbing and Molnár (SF), Karamouzas et al., Karamouzas et al. (PowerLaw)	$\mathcal{C}(\mathbf{v}') = 1/(2\Delta t) \cdot \|\mathbf{v}' - \mathbf{v}^*\|^2$, où $\mathbf{v}^* = \mathbf{v} + \mathbf{F}_{total}/m \cdot \Delta t$, et $\mathbf{F}_{total}$ diffèrent selon l'algorithme	Pas de Gradient (∇C est explicite)
Berg et al. (RVO)	$\mathcal{C}(\mathbf{v}') = w_a / TTC(2\mathbf{v}' - \mathbf{v}, A_j) + \|\mathbf{v}' - \mathbf{v}_{\text{pref}}\|$	'Optimisation Globale' : échantillonnage aléatoire dans $\{\mathbf{v}' \mid \|\mathbf{v}' - \mathbf{v}\| / \Delta t \leq a_{max}\}$, $\tau = 0$
Guy et al. (PLEdestrians)	$\mathcal{C}(\mathbf{v}') = \begin{cases} \infty, \textit{ if } TTC(\mathbf{v}', A_j) < t_{min} \\ t_{max}(w_a + w_b \|\mathbf{v}'\|^2) + 2\|\mathbf{g} - \mathbf{p} - t_{max} \cdot \mathbf{v}'\| \sqrt{w_a w_b}, \textit{ otherwise} \end{cases}$	Optimisation Globale, $\tau = 0$
Karamouzas and Overmars	$\mathcal{C}(\mathbf{v}') = w_a \cdot (1 - \cos\angle(\mathbf{v}', \mathbf{v}_{\text{pref}})) + w_b \cdot \|\|\mathbf{v}'\| - \|\mathbf{v}\|\| / s_{max}$ $+ w_c \cdot \|\mathbf{v}' - \mathbf{v}_{\text{pref}}\| / s_{max} + w_d \cdot TTC(\mathbf{v}', A_j)/t_{max}$	'Optimisation Globale' : Échantillonnage régulier dans $range(TTC(\mathbf{v}_{\text{pref}}))$, $\tau = 0$
Moussaïd et al.	$\mathcal{C}(\mathbf{v}') = K(\hat{\mathbf{v}}') \cdot \left(1 + (\frac{S(\hat{\mathbf{v}}') - \|\mathbf{v}'\|}{S(\hat{\mathbf{v}}')})^2\right)$, $K(\mathbf{d}) = 1 + {d_{max}}^2 + DC^*(\mathbf{d})^2 - 2d_{max} \cdot DC^*(\mathbf{d}) \cdot \cos\angle(\mathbf{d}, \mathbf{v}_{\text{pref}})$, $S(\mathbf{d}) = \min(s_{\text{pref}}, DC^*(\mathbf{d})/\tau)$, $DC^*(\mathbf{d}) = \min(d_{max}, DC(\mathbf{d} \cdot s_{\text{pref}}, A_j))$	'Optimisation Globale' : Échantillonnage régulier dans $\{\mathbf{v}' \mid \angle(\mathbf{v}', \mathbf{v}_{\text{pref}}) > \theta_{max}$ $\wedge \|\mathbf{v}'\| \leq s_{\text{pref}}\}$, $\tau = 0.5$
van den Berg et al. (ORCA)	$\mathcal{C}(\mathbf{v}') = \begin{cases} \infty, \textit{ si } \min_{k=0}^{n-1}(\mathbf{v}' - (\mathbf{v} + \frac{1}{2}\mathbf{u}_{jk})) \bullet \mathbf{n}_{jk} < 0, \\ \|\mathbf{v}' - \mathbf{v}_{\text{pref}}\|, \textit{ sinon} \end{cases}$ $\mathcal{C}_2(\mathbf{v}') = \max_{k=0}^{n-1}(\mathbf{v}' - (\mathbf{v} + \frac{1}{2}\mathbf{u}_{jk})) \bullet \mathbf{n}_{jk}$, où $\mathbf{u}_{jk}$ et $\mathbf{n}_{jk}$ definissent les demi-plans ORCA$^{t_{max}}$ pour A_j and A_k	Optimisation globale, $\tau = 0$
Dutra et al.	$\mathcal{C}(\mathbf{v}') \approx \sum_{k=0}^{n-1}\left(W_k \cdot e^{f_1(\mathbf{v}', A_k)}\right) \cdot \sum_{k=0}^{n-1} W_k + \left(1 - \frac{1}{2}(e^{f_2(\mathbf{v}')} + e^{f_3(\mathbf{v}')})\right)$, $W_k = \frac{1}{dist(A_j, A_k)^2}$, $f_1(\mathbf{v}', A_k) = -\frac{1}{2}\left((ttca(\mathbf{v}', A_j, A_k)/w_a)^2 + (dca(\mathbf{v}', A_j, A_k)/w_b)^2\right)$, $f_2(\mathbf{v}') = -\frac{1}{2}(\angle(\mathbf{v}', \mathbf{v}_{\text{pref}})/w_c)^2$, $\quad f_3(\mathbf{v}') = -\frac{1}{2}((\|\mathbf{v}'\| - \|\mathbf{v}_{\text{pref}}\|)/w_d)^2$	Pas de Gradient (∇C est explicite)

Notre méthode consiste ainsi à calculer une moyenne pondérée pour chaque agent voisin A_k plutôt que pour chaque pixel, où le poids associé à l'agent A_k vaut $1/dist(A_k, A_j)^2$. Cette méthode est imparfaite, car elle ne tient pas compte du nombre de pixels occupés par l'agent A_k, notamment en cas d'occultation de l'agent A_k par d'autres agents ou obstacles de l'environnement. Cependant, ce poids simule l'importance qu'ont les agents proches par rapport aux agents plus éloignés. Enfin, nous avons trouvé empiriquement que multiplier le coût total (et donc le gradient de l'équation 5.3) par un facteur $S = 20$ donne des comportements de foule satisfaisants.

5.5 Le logiciel UMANS

Nous avons implémenté l'algorithme précédemment décrit sous la forme d'un logiciel libre de droits qui s'intitule UMANS pour *"Unified Microscopic Agent*

Navigation Simulator"[3].

Figure 5.5 – Logo du logiciel UMANS

Les sources du logiciel sont publiques. Nous créons des versions correspondant à des étapes stables du développement du logiciel. On retrouve sur le dépôt public une version utilisable en lignes de commandes (Linux, Windows). Cette version utilise des fichiers de configuration pour définir les fonctions de coût à utiliser, les agents présents dans la simulation ou encore la forme de l'environnement. De nombreuses configurations pré-définies sont disponibles comme point de départ à l'utilisation du logiciel. Le logiciel peut être également utilisé en mode graphique par le biais du *framework* **Qt**. La figure 5.6 donne un aperçu de l'interface graphique de UMANS (décembre 2020).

Enfin, le logiciel est également disponible en version "librairie" (compilée pour Windows et Linux), qui se présente comme un ensemble de fonctions permettant de d'embarquer une simulation de foule générée par logiciel sur n'importe quelle plateforme (par exemple Unity3D, en langage C#).

5.6 Conclusion

Dans ce chapitre, nous avons détaillé une approche relative à la simulation de foule et qui consiste à contrôler des agents pour que leur vitesse courante optimise continûment une fonction de coût qui capture leurs objectifs et leurs interactions avec les agents voisins. Nous avons expliqué comment cette approche peut unifier un ensemble d'algorithmes existants qui reposent sur un principe équivalent. Leur reproduction est possible en construisant une fonction de coût adéquat, et en jouant sur la méthode d'optimisation employée.

Dans le cadre plus général de notre simulation robot-foule, cette approche présente plusieurs avantages. On peut basculer facilement d'une méthode à une autre pour simuler le comportement de la foule. On peut même mixer plusieurs

3. disponible à l'adresse https://project.inria.fr/crowdscience/project/ocsr/umans/

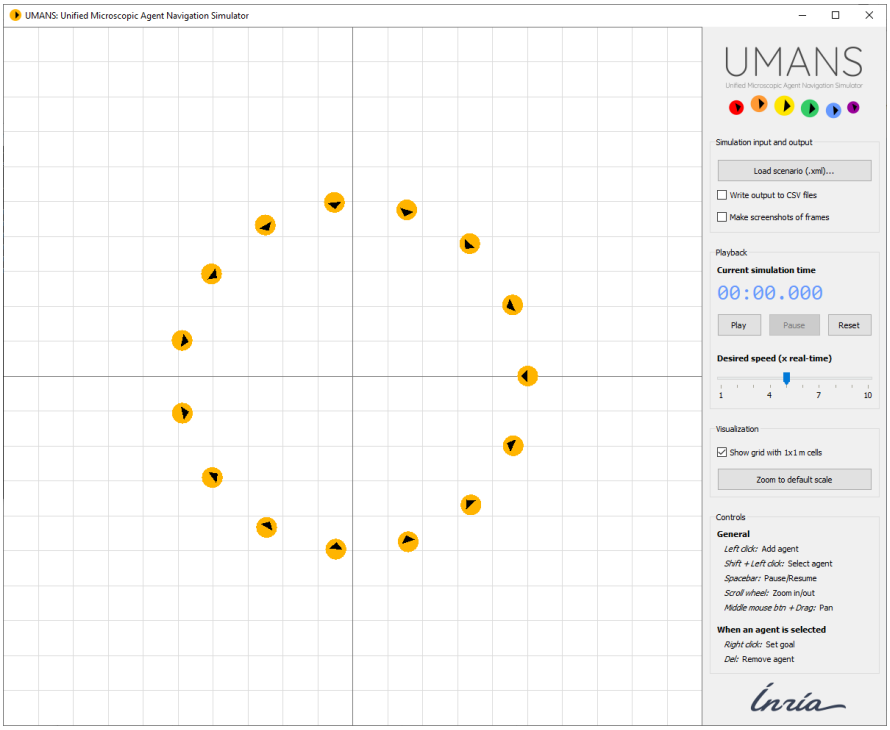
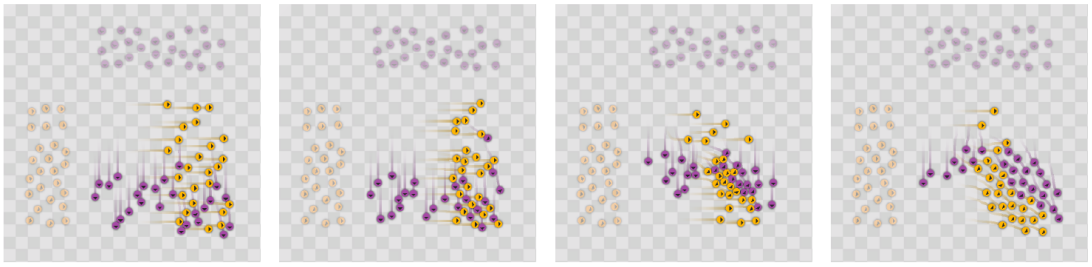


Figure 5.6 – Aperçu du logiciel UMANS en version graphique

comportements (ici, plusieurs fonctions de coût) simultanément. Cela évite de faire des choix arbitraires sur "le bon simulateur" de foule à utiliser. Cela permet également d'étendre le domaine de validité de la simulation en combinant plusieurs méthodes. Enfin, cela évite de mettre une barrière technique au changement de méthode de simulation.

Dans le chapitre suivant nous abordons plus directement l'utilisation de cette simulation dans le cadre de l'*évaluation* de capacités de robots à naviguer dans la foule



Chapitre 6

Évaluation à travers la simulation de la navigation de robot parmi la foule

Sommaire

Dans le chapitre 4, nous avons posé les bases d'un simulateur robot-foule qui repose sur deux grands éléments :

— un simulateur de robot, qui s'appuie sur le système ROS [1] pour permettre d'utiliser de fonctionnalités de ROS comme les outils de communication et d'assurer une utilisation facile par la communauté robotique ;

— une simulation de foule d'autre part, reposant sur Unity [2] pour ce qui concerne l'animation d'humains virtuels et incorporant un algorithme de simulation de foule pour piloter leurs comportements (i.e. trajectoires de déplacement).

Dans ce chapitre, nous explorons la mise en oeuvre de ces briques logicielles pour permettre l'évaluation de la navigation de robot dans la foule via un *banc d'essai* . Notre travail aborde deux aspects de ce problème :

1. définir des scénarios de navigation de référence ;
2. introduire des métriques pour évaluer les niveaux d'efficacité et de sécurité.

Le *banc d'essai* consiste donc à doter la communauté d'un outil de simulation qui peut proposer des scénarios communs pour éprouver les techniques de navigation de robots sur leurs capacités à faire évoluer un robot parmi la foule (virtuelle ici). Cela passe par la mise à disposition d'un ensemble de scénarios intégré à un outil de simulation qui permet de tester, d'évaluer et de comparer différentes techniques de navigation de robots adaptées à la navigation de foules d'humains. Ce chapitre explore donc notre solution [3] permettant de fournir à la communauté un outil de référence pour la navigation dans les foules.

6.1 Scénarios pour le *banc d'essai*

L'objectif du *banc d'essai* est de comparer les performances de diverses techniques de navigation de robots dans des situations standardisées, que nous appelons des scénarios. Le mouvement d'une foule simulée peut être infiniment variée, et dépend des conditions initiales (l'état initial des agents) et des paramètres (conditionnant le comportement dynamique des agents). Pour la simulation, il

1. ROS est une suite de logiciels dédiés à la robotique. ROS permet l'abstraction matérielle, le contrôle des périphériques de bas niveau, l'implémentation de fonctionnalités couramment utilisées, ou encore le passage de messages entre processus.

2. Unity est un moteur de jeu vidéos multiplateforme très répandu.

3. Notre outil fait l'objet d'une publication à conférence ICRA 2021 : *Fabien Grzeskowiak*, David Julian Gonon, Daniel Dugas, Diego Paez-Granados, Jen Jen Chung, Juan Nieto, Roland Siegwart, Aude Billard, Marie Babel, Julien Pettre. "Crowd against the machine : A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd"*

est nécessaire de *standardiser* les scénarios. Pour le *banc d'essai* , nous avons déterminé quatre variables :

1. *L'environnement :* c'est l'ensemble des obstacles statiques dans la simulation qui contraignent le robot et la foule.
2. *La dynamique de la foule :* cela correspond aux paramètres globaux de la foule définissant le comportement de la foule dans son ensemble.
3. *L'évitement local :* cela correspond aux paramètres individuels des agents de la foule, donnant à chaque agent son comportement individuel lors de l'évitement d'un autre agent ou du robot.
4. *Le robot :* c'est l'ensemble des caractéristiques du robot, comme son objectif, mais aussi ses capteurs et sa cinématique.

Le table 6.1 donne un exemple de paramètres possibles pour ces différentes variables.

Table 6.1 – Tableau donnant en exemple les conditions imposées pour des scénarios. Ces conditions donnent 162 scénarios différents

Variable	sous-variable	Paramètres	Valeurs possibles des paramètres
Environnement	Couloir	Dimensions	$50m * 10m$
	Obstacles	Type, localisation	Murs frontière
Dynamique de foule	Flux	Direction	Avec, Contre
		Densité (p/m^2)	0.125, 0.25, 0.5
		Vitesse (m/s) (moyenne, σ)	lent (0.8, 0.03), rapide (1.6,0.03), disparate (1.2,0.1)
Évitement local	RVO (UMANS) [van den Berg et al., 2011b]	temps d'horizon	5 secondes
	TtcaDca (UMANS) [Dutra et al., 2017]	Portée	2m, 5m, 10m
		Coefficient de coût	1, 20, 50
Robot	TurtleBot2	Objectif	traverser le couloir
		capteurs	2 Lidar (vers l'avant et l'arrière)
		Vitesse max	$1m/s$

Le reste de cette section détaille les variables des scénarios.

6.1.1 Environnement

Le robot évolue dans un couloir, il commence d'un côté du couloir et sa tâche consiste à atteindre l'autre côté. Le robot est libre de se déplacer dans ce couloir mais ne peut pas franchir les frontières délimitées par les quatre murs. En dehors des agents de la foule, les seuls obstacles sont donc ces quatre murs de délimitation. Le robot doit alors traverser le couloir, de la zone de début à la zone de fin, en évitant les différentes personnes/obstacles qui s'y trouvent. Les images des figures 6.1 et 6.2, extraites du simulateur, montrent un exemple de scénario.

Les paramètres de l'environnement sont les dimensions du couloir, ainsi que la taille et la localisation des zones de départ et d'arrivée. Nous avons fixé ces paramètres pour standardiser les scénarios :

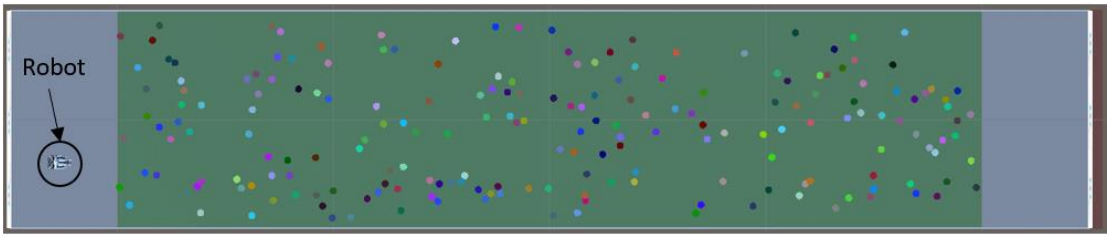


Figure 6.1 – État initial d'un scénario du *banc d'essai* avec 200 piétons, le robot se déplace avec le flux (de gauche à droite). Les cercles représentent les positions initiales des agents de la foule, toutes dans la zone verte.

— le couloir mesure 50 mètres de long et de 10 mètres de large ;

— la zone de départ du robot est définie par les 5 premiers mètres du couloir ;

— la zone de départ de la foule est définie par les 40 mètres situés juste après la zone de départ du robot ;

— les 5 derniers mètres correspondent à la zone d'arrivée du robot.

Densité constante garantie :

Nous introduisons ici un mécanisme pour maintenir l'activité et la densité de la foule constante. Quand un agent de la foule franchit un des murs au bout du couloir, il disparaît pour réapparaître instantanément à l'autre bout du couloir. Par exemple, les agents de la foule atteignant le mur de droite disparaissent instantanément pour réapparaître au même niveau près du mur de gauche. Ce mécanisme permet de garantir une densité constante dans le couloir et nous permet d'avoir un flux de foule continu.

6.1.2 Dynamique de la foule

Les quatre paramètres pour établir la dynamique de foule sont :

— la direction du flux,

— la densité,

— la vitesse moyenne,

— l'écart-type de la vitesse.

Pour contrôler la dynamique de la foule, il faut donner pour chaque agent une direction à suivre, et une vitesse. Pour ce faire, chaque agent de la foule a un but, correspondant à un point *en dehors* du couloir, de façon à ce que le but ne puisse jamais être atteint (les agents disparaissent en atteignant un des murs du bout du

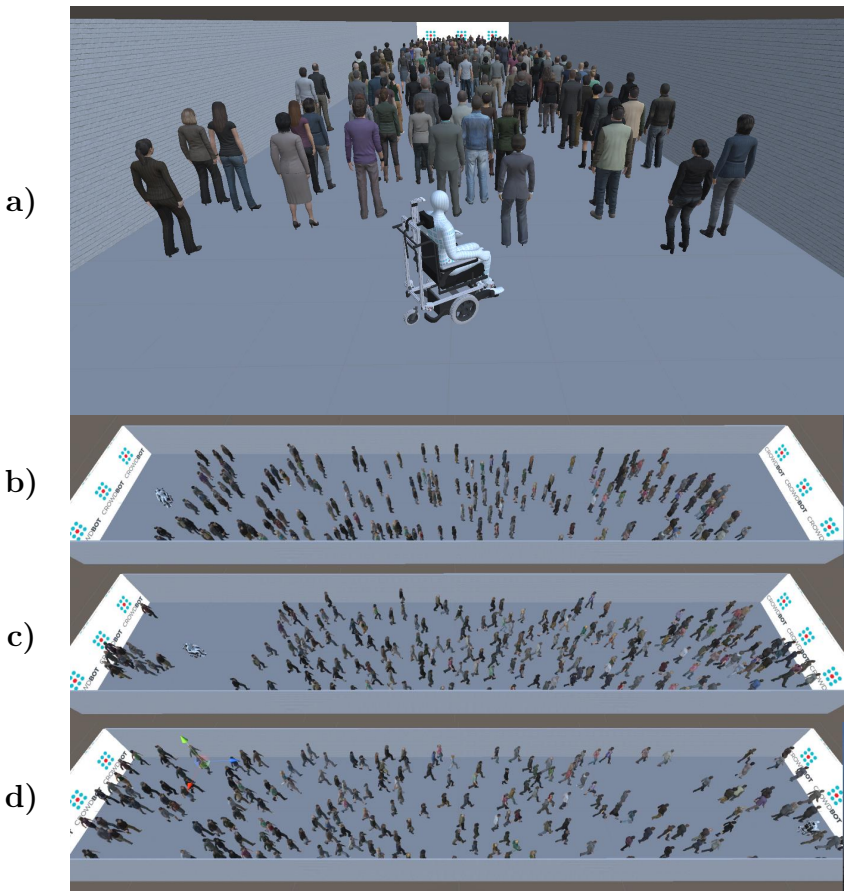


Figure 6.2 – Étapes typiques d'un scénario du *banc d'essai* . De haut en bas : a) Première étape de la simulation : le robot est à sa position initiale, b) Première étape de la simulation, vue de dessus, c) Étape intermédiaire : le robot et la foule interagissent, d) Dernière étape de la simulation : le robot a atteint son but.

couloir). Ce but est choisi au hasard, dans le sens de la largeur du couloir, mais tient compte du paramètre de direction du flux défini par l'utilisateur afin que la foule ait le comportement attendu : par exemple, si toute la foule doit aller dans une même direction, disons vers la droite, alors les buts des agents sont tous choisis à la droite du couloir. Le nombre d'agents dans le couloir est défini en fonction du paramètre de densité (la dimension du couloir étant fixée), tandis que la vitesse de marche de chaque agent est définie en fonction du paramètre de vitesse. Chaque paramètre est choisi en fonction du tableau figure 6.1. Pour chaque agent dans un scénario donné, la position de départ, l'objectif et d'autres paramètres aléatoires tels que les paramètres de vitesse, ont été générés une fois pour toutes et sont fixés pour chaque scénario du *banc d'essai* . Ainsi, les utilisateurs auront exactement les mêmes scénarios. Cela permet de faire une étude *comparative* sur une base commune.

Figure 6.3 – Modèle de simulation du robot TurtleBot2

Les deux possibilités de direction du flux de la foule sont "avec" (le robot et la foule se déplacent dans la même direction) ou "contre" (le robot et la foule ont des directions opposées).

La vitesse moyenne est une moyenne de la vitesse de tous les agents dans la simulation. L'écart-type est une évaluation de la dispersion des vitesses autour de la valeur moyenne : un grand écart-type donnera des agents aux vitesses disparates.

Dans l'exemple donné dans le tableau 6.1, les trois possibilités de vitesse sont les suivantes :

— Lente : toute la foule marche lentement avec une vitesse moyenne de 0,8 m/s et un écart-type de 0,03.

— Rapide : toute la foule marche vite avec une vitesse moyenne de 1,6 m/s et un écart type de 0,03.

— Dispersée : chaque agent de la foule a une vitesse de marche différente avec une vitesse moyenne de 1,2 m/s et un écart type de 0,1.

6.1.3 Simulateur de foule pour l'évitement local

Nous utilisons UMANS, le logiciel de simulation de foule décrit en détails dans le chapitre 5. En particulier, nous utilisons la capacité qu'offre ce logiciel de *reproduire* plusieurs algorithmes existants, et éventuellement de les mélanger dans un unique scénario pour créer un comportement de foule original.

6.1.4 Le robot implémenté pour le *banc d'essai*

Le robot par défaut du *banc d'essai* est TurtleBot2 (voir figure 6.3). C'est un robot libre de droit classique dans la communauté robotique. Il a été développé

par les créateurs de ROS. Il s'agit d'un kit de robot personnel à faible coût, avec un logiciel libre de droit. TurtleBot2 permet de construire facilement un robot capable de se déplacer, de voir en 3D, et qui est raisonnablement puissant.

La modélisation du robot de la figure 6.3 provient d'un package de ROS qui fournit un modèle 3D complet de TurtleBot pour la simulation et la visualisation.

Le TurtleBot de notre simulateur a une vitesse maximale de $1ms^{-1}$ et une accélération maximale de $0.5ms^{-2}$. Il est contrôlable par une commande de vitesse linéaire et angulaire. La simulation fournit l'emplacement de la foule, l'odométrie du robot et les obstacles statiques. Ces données peuvent être utilisées comme entrée pour la navigation du robot. Le robot virtuel est également équipé d'un ensemble de capteurs virtuels qui offrent la possibilité de tester la robustesse de la navigation du robot à des données limitées :

— douze capteurs à ultrasons, tout autour du robot ;

— deux LIDAR : un devant et un derrière ;

— une caméra RDBG vers l'avant ;

— un système d'odométrie pour se repérer dans l'environnement.

6.2 Outils d'évaluation

L'objectif principal du banc d'essai proposé est de générer un rapport qui donne, pour chaque métrique disponible, un score. Ces scores peuvent être utilisés comme une évaluation objective d'un algorithme donné. Chaque scénario complété générera un fichier CSV avec les scores des métriques, et un rosbag (fichier d'enregistrement pour ROS) avec les données brutes de la simulation (capteurs, position du robot et des agents...). Il incombe aux utilisateurs d'analyser les résultats et de proposer leur propre interprétation.

6.2.1 Métriques d'évaluation

Nous présentons ici sept métriques qui peuvent être divisées en trois catégories couvrant les thèmes de la sécurité de la foule et de l'efficacité du robot à effectuer sa tâche, qui cherchent à mesurer :

— l'efficacité du trajet,

— l'effet sur le flux de la foule,

— la proximité avec la foule.

Efficacité du trajet

Les métriques de la catégorie "efficacité de la trajectoire" donnent des éléments pour comparer la situation où le robot est seul à la configuration où le robot est entouré par une foule. Pour ces deux configurations, nous calculons les temps en seconde T et T_{foule} mis par le robot pour atteindre l'objectif et les comparons :

$$\text{Efficacité}_T = \frac{T}{T_{\text{foule}}}$$

, les longueurs en mètre L et L_{foule} de la trajectoire du robot :

$$\text{Efficacité}_L = \frac{L}{L_{\text{foule}}}$$

et la variation de la vitesse en m/s^2 (linéaire et angulaire) du robot J et J_{foule} pendant tout le scénario.

$$\text{Efficacité}_J = \frac{J}{J_{\text{foule}}}$$

Effets sur le flux de la foule

Les métriques relatives aux "effets sur le flux de la foule" comparent la vitesse en m/s des voisins du robot (dans un rayon de 1 m) V_{Voisins} à la vitesse en m/s de la foule entière V_{Tous} selon

$$NBR_{\text{vitesse}} = \frac{V_{\text{Voisins}}}{V_{\text{Tous}}}.$$

Ainsi, plus les voisins sont lents, plus la métrique est petite. Une seconde métrique,

$$NBR_{\text{reac}} = \frac{\omega_{\text{Tous}}}{\omega_{\text{Voisins}}},$$

nous comparons la vitesse angulaire en rad/s de la foule entière ω_{Tous} à la vitesse angulaire en rad/s des voisins du robot ω_{Voisins}. Si les voisins tournent plus que le reste de la foule, NBR_{reac} sera petit.

Proximité avec la foule

La première des métriques traitant de la "proximité de la foule" est la métrique de proximité Prox. définie par

$$\text{Prox} = 1 - \frac{1}{t_{\text{final}}} \sum_{t=0}^{t_{\text{final}}} \frac{d_{\min}(t)}{R},$$

où d_{min} est la distance en mètre à l'agent le plus proche, et R la portée en mètre. Dans notre cas, nous considérons qu'il y a au moins une personne qui se trouve dans un rayon de 5 mètres à proximité du robot ($R = 5$).

La dernière métrique de proximité est la métrique de collision, qui est définie par

$$\text{Col} = 1 - \frac{T_{\text{Collision}}}{T_{\text{Scénario}}}$$

où $T_{\text{Scénario}}$ est le temps total du scénario en secondes, et $T_{\text{Collision}}$ est le cumul des instants de collision, en seconde. Un score de 1 signifie qu'il n'y a pas de collision du tout.

6.2.2 Évaluation des collisions

La simulation donne accès a des informations précises sur les collisions, telles que leur position, la force de l'impact. Nous évaluons les collisions sur la base de ces rapports indiquant quelle partie du corps de l'agent est entrée en collision avec le robot. A partir de cette information, nous calculons sa masse de référence m_{ref}, ce qui correspond approximativement à la contribution de l'inertie humaine à l'impact. La masse de référence de chaque partie du corps en collision m_{ref} est calculée comme étant la masse du segment correspondant de la partie d'un modèle de corps humain de référence. Ce modèle représente le corps humain en une chaîne verticale articulée de quatre corps rigides : les pieds, les jambes, les cuisses et le reste du corps. La longueur et les propriétés inertielles de ces segments sont la somme des valeurs de ces 4 corps rigides pour des adultes moyens (selon Winter [2009]). En suivant l'approche de Khatib [1995], nous calculons pour notre modèle la masse correspondante m_{ref}, à savoir : environ 4 kg pour les pieds, 13 kg pour le bas des jambes, et 24 kg pour le haut des jambes. Par ailleurs, en supposant que la masse du robot est de $m_{\text{rob}} = 20\text{kg}$, nous calculons l'énergie cinétique en joule ΔE qu'une collision absorberait (et éventuellement restituerait) comme égale à

$$\Delta E = \frac{\mu v_{\text{rel}}^2}{2}, \text{ où } \mu = (m_{\text{ref}}^{-1} + m_{\text{rob}}^{-1})^{-1}$$

est la masse réduite et v_{rel} est la vitesse en m/s relative des centres de masse des deux agents l'un par rapport à l'autre. Comme ΔE quantifie directement la déformation potentielle des corps, il fournit une mesure de la gravité des collisions Rossi et al. [2015].

6.3 Expérimentations avec le *banc d'essai*

Cette section présente notre protocole d'évaluation et les choix fait en termes de paramètres pour un exemple d'évaluation.

6.3.1 Techniques de navigation de robot et hypothèses

Pour notre robot, nous avons sélectionné les 3 techniques de navigation suivantes :

1. une méthode "BASELINE", qui consiste à faire aller le robot droit vers le but en ignorant la foule ;
2. la méthode d'évitement par fenêtres dynamiques (DWA) Fox et al. [1997] ;
3. la méthode des obstacles à vitesse réciproque (RVO). van den Berg et al. [2011b].

La raison de ce choix est que le robot va a priori être plus ou moins adapté à naviguer dans la foule qui l'entoure selon les méthodes. En effet, le robot va respectivement

1. ignorer les agents de la foule : efficace mais dangereux,
2. les considérer comme des obstacles statiques : moins dangereux mais peu efficace,
3. prédire le mouvement futur à court terme des agents : Plus adapté que les autres méthodes.

Avec ce choix, qui consiste à *illustrer* le fonctionnement de notre outil, celui-cidevrait démontrer sa capacité à révéler les avantages et les inconvénients de l'efficacité et de la sécurité de la navigation du robot par rapport aux agents de la foule. Si notre outil d'évaluation est adéquat pour être utilisé comme outil de référence, alors l'évaluation doit satisfaire les hypothèses suivantes :

$H1$: La méthode BASELINE devrait obtenir le meilleur score en termes d'efficacité, puisque le robot ignore la foule. Cependant, elle devrait être la plus mauvaise en termes de sécurité, puisqu'elle doit se rapprocher des agents de la foule et provoquer de nombreuses collisions avec eux, sans même essayer de les éviter, maximisant ainsi les forces de collision avec la foule.

$H2$: La méthode DWA, par rapport à la méthode BASELINE, devrait montrer un nombre de collisions plus faible, et donc un niveau de sécurité plus élevé. Comme la méthode considère les obstacles comme demeurant statiques, elle devrait montrer des résultats assez médiocres en termes d'efficacité (selon

les indicateurs des métriques du paragraphe 6.2.1), car elle peut placer le robot sur des trajectoires d'évitement inutiles ou tardives.

*H*3 : La méthode RVO devrait montrer le meilleur compromis entre efficacité et sécurité, car elle est capable de prédire le mouvement des agents et attend la contribution des agents dans l'exécution de l'évitement.

6.3.2 Paramètres des scénarios pour notre exemple d'évaluation

Nous décrivons ici l'ensemble des paramètres pour notre exemple d'évaluation. Si l'on combine tous ces paramètres, on obtient 100 scénarios différents.

Robot : Notre exemple utilise un seul robot, le TurtleBot2, décrit dans le chapitre 4. C'est un robot mobile classique et bien connu dans la communauté robotique ROS. Pour effectuer sa tâche (i.e. traverser le couloir décrit à la section 6.1.1), nous imposons une limite de temps de 180 secondes pour éviter une simulation infinie pour les situations où la navigation ne trouve pas de solution.

Flux de foule : Notre exemple d'évaluation présente cinq flux de foule :

- 1D+ : La foule se dirige dans la même direction que le robot,
- 1D- : la foule se dirige dans la direction opposée au robot,
- 1Dx : La moitié de la foule se dirige dans la direction du robot (1D+), l'autre moité dans la direction opposée (1D-). Les flux se croisent.
- 2D : la foule se dirige dans la direction perpendiculaire à celle du robot,
- 2Dx : : La foule se divise en deux flux opposés, perpendiculaires à la direction du robot.

Simulation de foule : Nous avons sélectionné deux algorithmes de simulation de foule qui sont assez courants et facilement configurables : "RVO" (également utilisé par le robot), et "Forces sociales". Ils sont décrits dans la section 2.2. Nous avons choisi de définir les paramètres de RVO avec un horizon temporel de 0,5s (i.e. la foule ne réagit que lorsqu'elle est proche du robot) ou avec un horizon temporel de 1,5s, ce qui permet d'anticiper davantage les interactions. Forces sociales est utilisé avec des paramètres par défaut de UMANS, correspondant à ceux suggérés par Helbing and Molnár [1995].

Réactivité : Notre exemple présente à la fois les scénarios où la foule est réactive au robot, et ceux dans lesquels la foule n'est pas réactive à la présence du robot.

Densité : Nous avons sélectionné quatre niveaux de densité de la foule : 50 $(0,1pm^{-2})$, 100 $(0,2pm^{-2})$, 200 $(0,4pm^{-2})$ ou 350 $(0,7pm^{-2})$ agents dans le couloir.

6.4 Résultats sur un exemple

Nous représentons les sept métriques décrites dans la section 6.2.1 sur deux cartes radar présentées dans la Fig. 6.4. Le tableau 6.2 donne l'écart type pour chaque métrique des cartes radar.

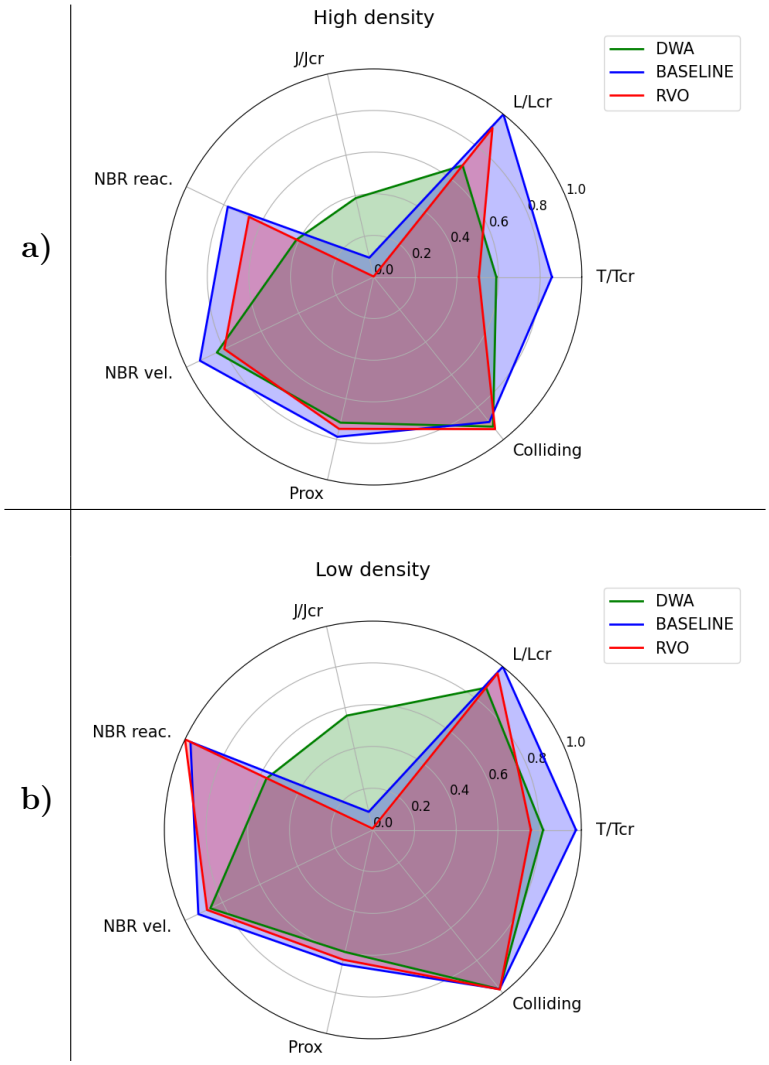


Figure 6.4 – Cartes radar pour l'exemple d'évaluation du *banc d'essai* . (a) Nous calculons les sept métriques décrites dans la section 6.2.1 pour 20 scénarios : 4 niveaux de densité, 5 flux de foule (voir Section 6.3.2). (b) De même, nous calculons la même carte radar pour 5 scénarios : seulement ceux avec une faible densité (0,1pm^{-2})

Table 6.2 – Écarts-type pour les cartes radar à haute et basse densité

Méthode de nav.	Ecart-type pour chaque métrique						
	$\frac{T}{T_{foule}}$	$\frac{L}{L_{foule}}$	$\frac{J}{J_{foule}}$	NBR_{reac}	NBR_{vel}	$Prox$	Col
BASELINE haute densité	0.190	0.00	0.238	1.022	0.070	0.132	0.132
BASELINE basse densité	0.022	0.00	0.244	1.402	0.084	0.164	0.027
DWA haute densité	0.257	0.225	0.289	0.488	0.089	0.175	0.117
DWA basse densité	0.171	0.123	0.280	0.807	0.096	0.171	0.040
RVO haute densité	0.236	0.100	0.015	1.052	0.129	0.141	0.092
RVO basse densité	0.145	0.067	0.029	1.863	0.125	0.149	0.031

6.4.1 Résultats de l'évaluation des collisions

La figure 6.5 montre combien de collisions se sont produites avec des valeurs particulières d'énergie ΔE au cours de toutes les simulations, et pour chaque méthode de navigation du robot. Pour normaliser par le temps simulé, nous définissons le taux de collision f_c en s^{-1} d'une méthode de navigation donnée par

$$f_c = N_c/T$$

et son taux d'énergie Q en J/s par

$$Q = \Sigma\Delta E/T,$$

où N_c, T et $\Sigma\Delta E$ désignent respectivement le nombre total de collisions, le temps écoulé et l'absorption d'énergie dans les simulations correspondantes. Pour la BASELINE, DWA et RVO, nous mesurons respectivement $f_c = 0{,}624\ s^{-1}$, $0{,}610 s^{-1}$, $0{,}311 s^{-1}$ et $Q = 2{,}568$ J/s, 2,285 J/s, 0,901 J/s.

6.5 Discussion

Cette section reprend chacun des résultats et en donne une interprétation à deux aspects : l'efficacité de la trajectoire et la sécurité des foules.

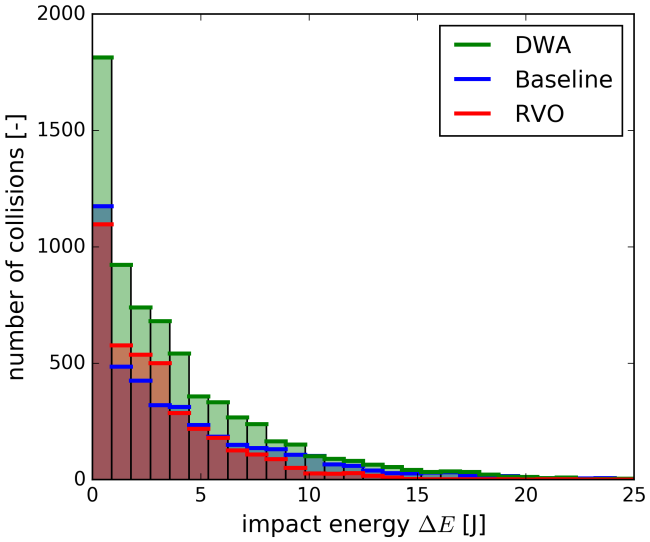


Figure 6.5 – Les histogrammes comptent les collisions pour chaque méthode de navigation robot sur l'ensemble des simulations, en les discriminant par l'absorption d'énergie estimée.

6.5.1 Efficacité de la trajectoire

L'efficacité de la trajectoire peut être évaluée à l'aide de la carte radar de la figure 6.4. La métrique L/L_{foule} est un indicateur de la longueur du trajet. Naturellement, le score de la méthode BASELINE est de 1, car la méthode prendra le chemin le plus court possible dans le couloir (ce qui valide $H1$). La méthode RVO est beaucoup plus directe que la méthode DWA dans les foules à faible densité et à forte densité, ce qui montre la tendance qu'a la méthode DWA à prendre des chemins moins directs que la méthode RVO pour atteindre l'objectif (ce qui valide $H2$ et $H3$). Cette idée est renforcée par la métrique J/J_{foule} qui est un indicateur de la variation de la vitesse linéaire et de rotation. En effet, les méthodes BASELINE et RVO ont beaucoup de mouvements erratiques (démarrage et arrêt), tandis que la méthode DWA prend en compte la dynamique du robot et sanctionne la différence entre la vitesse actuelle et la vitesse souhaitée. La métrique T/T_{foule}, qui est un indicateur du temps nécessaire pour atteindre l'objectif, montre que la méthode la plus directe, BASELINE, est la plus rapide dans les scénarios de faible et de forte densité (en accord avec $H1$). Les deux méthodes RVO et DWA sont beaucoup plus lentes que la méthode BASELINE dans les scénarios de haute densité. En effet, ces scénarios conduisent à des situations où la seule solution pour le robot est d'arrêter totalement sa progression voire même de faire demi-tour, en faisant face au point de départ plutôt qu'au but, afin d'éviter les collisions.

6.5.2 Navigation sécurisante

La sécurité de la foule peut être appréciée grâce à la métrique de proximité, la métrique de collision, les métriques liées aux vitesses angulaires et linéaires du voisinage $\mathrm{NBR}_{\mathrm{reac}}$ et $\mathrm{NBR}_{\mathrm{vel}}$ et les histogrammes de l'énergie d'impact de la figure 6.5.

Tout d'abord, il est intéressant de noter que, sur la carte radar de la figure 6.4 pour les foules de faible densité, la métrique de collision a une valeur maximale pour chacune des méthodes, ce qui signifie qu'en faible densité (0,1 pm^{-2}), la foule est parfaitement capable d'éviter le robot sans qu'il n'utilise de méthode spécifique. Les paramètres des algorithmes de simulation de foule choisis sont *typiques* des articles originaux de ces algorithmes, et représentent donc des comportement réalistes. Dans les scénarios à haute densité, la métrique pour la méthode BASELINE a une valeur inférieure à celle des deux autres méthodes DWA et RVO : le robot n'essaie pas d'éviter la foule, donc le risque de collision augmente (ce qui valide l'hypothèse $H1$). De plus, dans les scénarios à haute densité, la méthode RVO est plus performante que la méthode DWA en ce qui concerne la métrique de collision (ce qui valide les hypothèses $H2$ et $H3$ concernant les performances de DWA et RVO). La réaction de la foule au robot peut être évaluée grâce aux métriques $\mathrm{NBR}_{\mathrm{reac}}$ et $\mathrm{NBR}_{\mathrm{vel}}$. Les cartes radar de la figure 6.4 montrent que dans les foules à forte densité, le voisinage du robot est plus susceptible de changer d'orientation que de vitesse. En faible densité, la métrique $\mathrm{NBR}_{\mathrm{reac}}$ montre que les voisins du robot changent d'orientation beaucoup plus souvent que le reste de la foule. Nous pensons que le mouvement du robot généré par la méthode RVO et la méthode BASELINE génèrent des trajectoires plus linéaires (par définition) que la méthode DWA (qui valide l'hypothèse $H3$). Les trajectoires linéaires sont plus faciles à prédire pour le simulateur de foule que nous avons utilisé, notamment pour RVO qui fais une prédiction en considérant une vitesse constante des agents.

La métrique Prox. est naturellement plus élevée dans les foules à forte densité que dans les foules à faible densité, mais la méthode de navigation du robot ne semble pas affecter cette métrique. Les histogrammes de l'énergie d'impact de la figure 6.5 montrent que les 3 techniques considérées génèrent une distribution similaire des énergies d'impact et donc également des forces d'impact lors des collisions. Il est également visible que la méthode DWA entraîne le plus grand nombre de collisions à tous les niveaux d'énergie. Cependant, un nombre total de collisions plus élevé résulte de simulations plus longues, car le taux de collision f_c pour la méthode DWA se situe entre les deux autres méthodes (confirmant le

classement donné par la métrique de collision).

Néanmoins, la méthode BASELINE cause des collisions à hautes énergies plus souvent que la méthode RVO, tandis que la méthode RVO donne des collisions plus souvent que la méthode BASELINE sur les basses énergies. Ainsi, RVO s'avère un bon compromis entre le nombre de collisions et l'énergie de l'impact (ce qui valide $H1$ et $H3$). Enfin, les taux de collision et d'absorption d'énergie f_c et Q sont systématiquement mauvais pour la méthode BASELINE, meilleurs pour la méthode DWA, et bien meilleurs pour la méthode RVO. Cette constatation valide $H1$, $H2$ et $H3$ et indique que la méthode RVO conduit au plus petit nombre de collisions et au plus faible risque de collisions par unité de temps.

6.6 Conclusion

Dans ce chapitre, nous avons détaillé notre méthode pour standardiser des scénarios, et avons détaillé des métriques d'évaluation. Nous avons ensuite illustré ces métrique sur des scénarios dont les résultats étaient prévisibles. En obtenant des résultats cohérents, nous avons démontré le potentiel de notre simulateur à être utilisé comme outil de référence pour comparer différentes techniques de navigation de robot mobile dans la foule. Forts des premières expérimentations menées, nous pensons que la solution que nous proposons est intéressante pour la communauté et ouvre de nouvelles perspectives dans l'évaluation des capacités de navigation des robots dans la foule. La simulation de foule est en effet utile pour créer un environnement de test synthétique. Elle peut être un outil préliminaire et complémentaire aux tests réels qui évaluent l'évolution d'un robot dans un environnement qui est représentatif à bien des égards d'une foule réelle. A minima, les tests basés sur la simulation permettent d'explorer automatiquement un grand nombre de paramètres de scénario. Ils facilitent les comparaisons et sont, sans aucun doute, sûrs. De plus, la simulation présente l'intérêt majeur de permettre de définir facilement un ensemble de *scénarios standard*, qui sont définis par la configuration de la foule et la tâche du robot. Ces scénarios peuvent être reproduits avec précision d'une série de tests à l'autre, et peuvent être élaborés et partagés à l'échelle de toute une communauté. Au final, notre simulateur est le candidat idéal pour évaluer et comparer équitablement le comportement des robots dans une foule.

Bien que l'évaluation basée sur la simulation offre de précieuses informations, elle ne peut remplacer la validité des tests réels car, si la simulation de robots ne peut déjà être considérée comme parfaitement précise, la simulation du comportement humain est encore plus complexe. Il est impossible de simuler tous les

comportements qu'une foule peut manifester, en particulier ses réactions face à un robot. Le chapitre suivant explore donc l'utilisation d'outils d'immersion dans la simulation pour des humains et des robots, dans le but d'accroître significativement la fidélité de nos résultats par rapport à la réalité, au prix cependant de l'utilisation de dispositifs conséquents de réalité virtuelle.

Chapitre

7 Interactions humain-robot en réalité virtuelle

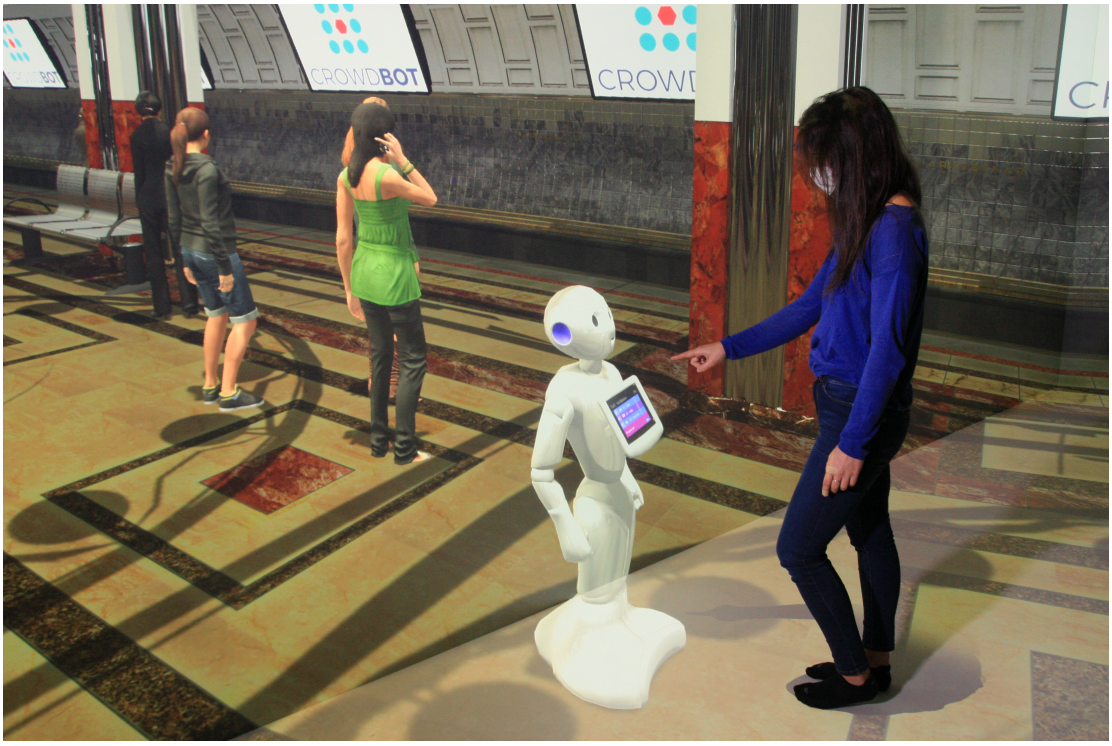


Figure 7.1 – Illustration d'interaction humain-robot virtuelle. Le robot Pepper est virtuel alors que la personne est dans un CAVE ("Cave Automatic Virtual Environnement"), un outil d'immersion en réalité virtuelle

Sommaire

7.1 Introduction : pourquoi utiliser la réalité virtuelle pour l'interaction humain-robot ?

On a vu dans les chapitres précédents que la simulation est un outil largement utilisé pour valider des concepts en robotique. On cherche par ce biais à évaluer le bon fonctionnement de ces concepts. Cela peut se traduire par la notion *d'efficacité* du concept : la tâche est-elle accomplie, dans un temps acceptable, sans compromettre l'intégrité du robot et de son environnement ?

Cependant, les robots naviguant au sein d'une foule sont confrontés à un nouveau problème : la présence de l'humain dans l'équation pose en effet la problématique de la *sécurité* pour nos robots, en plus de leur efficacité.

Pour s'assurer de la qualité de nos algorithmes, il nous faut mener des expérimentations poussées prenant en compte les interactions entre robots et humains. Il faut en particulier s'intéresser aux interaction proches, celles qui amènent inévitablement une collision (même anodine) entre le robot et l'humain. Comme nous l'avons vu dans le chapitre 6, la simulation offre la possibilité de tester de nombreuses configurations, mais reste limitée du fait des simplifications nécessaires. En particulier, l'interaction entre un robot et un agent d'une foule simulée est peu convaincante. En effet, les agents virtuels sont mus par des animations simples telles que la marche, la course, ou restent dans une position statique. Ces animations sont insuffisantes pour convaincre de leur réalisme lors d'une interaction

proche entre un agent simulé et un robot. L'approche théorique, *i.e* la simulation, pour étudier ces interactions proches est donc vouée à l'échec.

Il est ainsi nécessaire d'évaluer ces interactions avec des expérimentations impliquant de vraies personnes, réagissant à la présence du robot. Cependant, une telle étude pose de nouveaux problèmes. Tout d'abord, il y a des questions purement techniques comme le contrôle des nombreux paramètres d'interaction : par exemple, un environnement de gare est plus ou moins peuplé, les gens sont plus ou moins pressés selon l'heure, et un simple changement de luminosité peut affecter l'interaction. Ce contrôle des paramètres est assuré en simulation. Les expérimentations impliquant de vraies personnes posent également des questions éthiques : certaines des configurations les plus intéressantes à étudier, offrant des interactions proches, présentent des risques potentiels de collision entre les humains et les robots. La simulation, c'est-à-dire l'utilisation d'algorithmes pour reproduire l'environnement et les comportements des robots, atténue partiellement ce problème car elle offre la possibilité d'échantillonner et de normaliser un très grand nombre de configurations. La modélisation du comportement humain reste toutefois un problème largement ouvert.

Dans ce chapitre, notre idée est de concilier les avantages des outils de simulation, offrant efficacité et sûreté, avec ceux des tests empiriques impliquant des robots et des humains réels, offrant des conditions réalistes. Dans un premier temps, nous présentons notre solution pour concilier simulation et expérimentation : un principe innovant d'immersion combinée de robot réel et d'une personne en réalité virtuelle. A notre connaissance, si l'idée d'utiliser la réalité virtuelle (RV) pour l'étude des interactions homme-robot (HRI) a déjà été envisagée, l'utilisation d'un robot mobile est innovant et, pour la première fois, un robot mobile est immergé avec un utilisateur humain en vue de provoquer une interaction naturelle. Ainsi, la nouveauté de notre travail réside dans le fait que les stimulus sensoriels de l'homme **et** du robot sont simulées, préservant ainsi tous les mécanismes de décision et d'action pour chacun d'entre eux. Pour le robot, cela signifie qu'il réagit à la situation en utilisant l'électronique, les algorithmes, le contrôle et les moteurs embarqués : ses capacités réelles sont préservées et reproduites dans le monde virtuel par la capture du mouvement réel résultant. La section 7.2 détaille la mise en place de ce principe pour des expérimentations.

7.2 Immersion de robots et d'humains

L'étude des interactions humain robots peut se faire de traditionnellement de 3 façons différentes : la simulation, les expérimentations contrôlées en laboratoire,

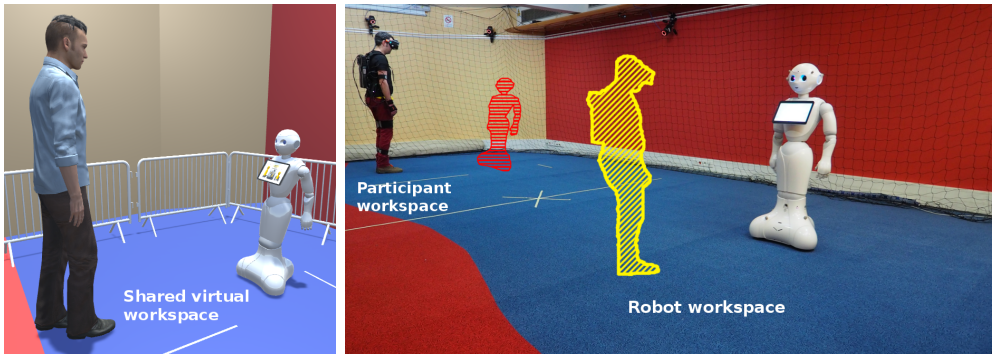


Figure 7.2 – *A gauche :* les avatars d'un robot réel et d'un vrai participant se font face dans un environnement virtuel partagé. *A droite :* les mouvements des avatars sont dirigés respectivement par la capture des mouvements d'un vrai robot et d'un humain qui sont physiquement séparés pour éviter la collision.

	Réalisme	Sécurité	Contrôle	Mesurabilité
Simulation		✓✓✓	✓✓✓	✓✓✓
Réalité virtuelle	✓	✓✓	✓✓	✓✓✓
Laboratoire	✓✓	✓	✓	✓✓
Conditions réelles	✓✓✓			✓

Table 7.1 – Tableau comparatif des différentes méthodes d'études d'interactions humain-robot

et les expérimentations en situations réelles. Nous proposons une autre façon d'expérimenter en interaction humain robot à travers la réalité virtuelle. Le tableau 7.1 compare les différentes méthodes sur 4 critères : le réalisme de l'interaction, la garantie de la sécurité pour les humains participant aux interactions, le contrôle des paramètre de l'expérimentation, et enfin la facilité pour l'expérimentateur de prendre des mesures.

Nous souhaitons obtenir le meilleur compromis, garantissant le meilleur niveau de sécurité possible pour des expériences impliquant des personnes, sans perturber l'interaction. A cette fin, nous utilisons la RV pour plonger des robots et des humains dans le même espace de mouvement virtuel, dans lequel ils peuvent interagir, alors qu'en réalité, ils se déplacent (physiquement) dans des espaces de mouvements réels bien distincts et séparés (figure 7.2). Cette méthode enlève tout risque de blessure lié à une éventuelle collision entre l'humain et le robot. Bien que la RV en robotique ait déjà été utilisée, l'intérêt est souvent porté sur le point de vue utilisateur, et le robot est purement virtuel. Par exemple, dans Hernoux et al. [2015], la RV est utilisée comme interface pour le contrôle de robots collaboratifs

industriels. Dans Devigne et al. [2017], la RV est utilisée pour enseigner aux utilisateurs comment conduire un fauteuil roulant électrique en toute sécurité. Dans ces deux exemples, l'attention est portée sur l'utilisateur. Il n'y pas vraiment de robot, mais plutôt une interface pour la réalité virtuelle. Par ailleurs, il n'est jamais question de robotique *mobile* en interaction avec un humain. La nouveauté présentée dans cette thèse est de plonger le robot mobile lui même dans le monde virtuel. Seules les entrées sensorielles du robot et de l'humain sont donc simulées, préservant ainsi les mécanismes de décision et d'action pour chacun. Cela veut dire que le robot répond à la situation en utilisant son matériel embarqué, son électronique, ses algorithmes, ses moteurs : ses vraies capacités sont ainsi préservées et son mouvement est copié par l'avatar dans le monde virtuel en utilisant de la capture de mouvement.

Nous avons donc pour objectif d'explorer la capacité d'une telle plateforme à générer des situations d'interaction réalistes. En d'autres termes, nous souhaitons nous assurer que la simulation des entrées sensorielles que nous générons pour le robot et l'humain sont de qualité suffisante pour permettre à un robot et à un humain d'interagir de façon similaire à la façon dont ils interagiraient dans la vraie vie. C'est un problème classique en RV lorsqu'il s'agit d'étudier l'immersion d'un humain, mais la question doit se poser lorsqu'il s'agit d'un robot qui est plongé en RV. Il s'agit bien sûr d'un objectif à long terme, et nous proposons de réaliser une étude pilote pour tenter d'observer chez le robot des biais de perception similaires à ceux qu'on observe lorsque le sujet d'étude est un humain.

Nous présentons donc ici deux contributions :

— nous donnons une description technique de la plateforme de RV, adaptée spécialement à l'étude d'interactions humain-robot lors de tâches de navigation ;

— nous proposons une évaluation pilote pour cette plateforme, principalement pour mesurer la qualité de perception du robot par les humains en RV et inversement. A cette fin, nous observons l'influence de cette perception mutuelle sur l'interaction.

7.3 Description technique de la plateforme de RV

Notre objectif technique, illustré par la figure 7.3, est de créer un espace virtuel partagé (VW pour *"Virtual Workspace"*) où l'avatar d'un humain et d'un robot

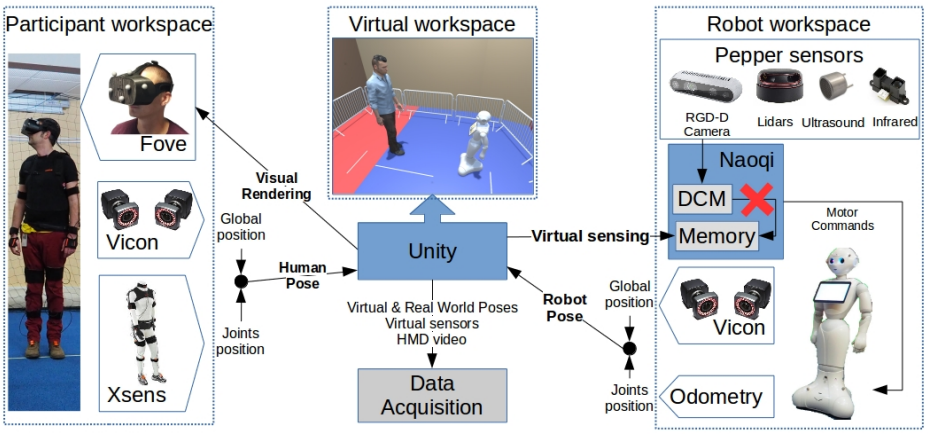


Figure 7.3 – Schéma de la plateforme. Une scène virtuelle en 3 dimensions (*virtual workspace*) est la réplique virtuelle des espaces séparés où l'humain et le robot bougent. Les participants sont matérialisés dans l'espace virtuel grâce au couplage de plusieurs systèmes de capture de mouvement. Le participant perçoit son environnement grâce à un casque de réalité virtuelle (*FOVE HMD*). Le robot *Pepper* peut bouger dans son espace dédié. Un robot virtuel imite le robot réel dans l'environnement virtuel grâce à un couplage entre ses capteurs odometriques et un système de capture de mouvement. Le robot virtuel capte à son tour l'environnement virtuel grâce à des capteurs simulés. Ces données capteur simulées remplacent ensuite les capteurs réels dans le processus de contrôle du robot réel.

peuvent interagir par l'intermédiaire de la VR. Nous avons besoin d'une représentation précise et convaincante de l'humain, notamment des membres inférieurs et supérieurs, en réalité virtuelle afin de donner un meilleur sentiment *d'incarnation* de l'avatar. Une telle représentation est également nécessaire afin de permettre aux capteurs virtuels de générer des données plus réalistes. A l'inverse, le robot virtuel doit bouger en mimant le robot réel, pour donner à l'humain une perception réaliste. Pour contrôler les avatars, nous utilisons la capture de mouvement. L'humain perçoit alors le VW et le robot virtuel, ainsi que le mouvement de son propre avatar par le biais d'un rendu graphique du VW affiché sur les écrans du casque de réalité virtuelle. Le robot perçoit à son tour le VW et et l'avatar de l'humain par le biais de ses capteurs virtuels qui outrepassent les capteurs réels.

7.3.1 Équipement

Pour concrétiser l'idée de cette plateforme, il est nécessaire d'utiliser un certain nombre d'outils, visibles sur la figure 7.3 :

— un moteur de jeu *Unity*. Le moteur de jeu génère le VW et fournit les outils et *plugins* permettant d'utiliser les différents composants. Plus exactement,

nous avons utilisé l'outil de simulation présenté dans le chapitre 3.

— un robot mobile : ici nous utilisons le robot *Pepper* [Pandey and Gelin, 2018a]. C'est un robot humanoïde avec une base mobile omnidirectionnelles conçu par *Softbank Robotics* pour l'interaction avec les humains. Il utilise un processeur *Intel* Atom E3845. Pour détecter son environnement, il est équipé d'un capteur Lidar, de capteurs à ultrasons (US) et de capteurs infrarouges (IR), de caméras RGB et RGB-D. Il est programmé pour s'arrêter s'il est trop proche d'obstacles. La distance d'arrêt dépend de la vitesse du robot (0.30 m pour une vitesse de 0.35 m/s) [1].

— un système de capture de mouvement global pour la position de la base : on utilise le système *Vicon* [2]. C'est un système capable de suivre le mouvement d'objets rigides à l'aide de caméras infrarouges et de marqueurs réfléchissants. Avec le logiciel *Tracker 3*, nous pouvons calculer une nouvelle position toutes les 1.5ms.

— un système de capture du mouvement relatif par rapport à la base : nous utilisons le costume *Xsens* [3] [Schepers et al., 2018] montré sur la figure 7.4. Le costume se compose de 18 centrales inertielles (IMU pour *"Inertial Measument Units"* connectées ensemble pour estimer la position relative des membres du corps. Le système Xsens permet de suivre les mouvements humains en temps réel, de manière simple, fiable et précise. Les capteurs IMU ont été installés sur les participants à l'aide de combinaisons et de sangles adaptées, tandis que le suivi du corps a été assuré par le logiciel *Xsens MVN Animate* et transmis à *Unity* en temps réel.

— un système de suivi des mouvements du robot : nous avons utilisé les propres capteurs *odométriques* de *Pepper*.

— un système de RV : nous avons utilisé le casque de RV *Fove* [4]. Il intègre un écran WQHD OLED (2560 X 1440), qui spécifie une fréquence d'images de 70 images par seconde, et un champ de vision allant jusqu'à 100 degrés. Le système de suivi *Fove* combine un capteur IMU et un suivi basé sur l'infrarouge. Il intègre des possibilités de suivi du regard même si elles ne sont pas utilisées dans notre solution actuelle.

1. voir http://doc.aldebaran.com/2-5/naoqi/sensors/dcm.html
2. www.vicon.com
3. www.xsens.com
4. www.getfove.com

Figure 7.4 – Le costume *Xsens* permettant la capture des mouvements du corps avec ses 18 capteurs IMU.

Comme le montre la figure 7.2, le système utilise trois espaces de travail différents. Le robot et l'humain ont leur propre espace de travail dans le monde réel, et restent donc physiquement séparés. Toutes leurs actions, effectuées physiquement dans leurs espaces de travail respectifs, sont traduites dans le VW. Ils perçoivent également tous deux le VW, au lieu de leurs espaces de travail respectifs.

La VW est une scène 3D qui peut être modélisée et visualisée à l'aide de *Unity* via l'outil décrit au chapitre 3. La figure 7.3 montre la VW conçue spécifiquement pour l'expérience présentée dans la section 7.4. Les obstacles présents dans l'espace de travail, tels que les murs, sont présents également dans le VW, de façon a ce qu'ils puissent être perçus et évités par les participants et les robots.

7.3.2 Perception des actions de l'humain et du robot :

Afin de reproduire les actions de l'homme et du robot dans le VW, nous avons besoin du mouvement de tout le corps. Pour ce faire, le mouvement de l'homme est suivi par le système *Vicon* et la combinaison *Xsens*, tandis que le mouvement du robot est suivi par le système *Vicon* et ses propres capteurs odométriques, comme le montre la figure 7.3.

Afin de suivre un objet avec le système *Vicon*, nous collons des marqueurs réfléchissants tout autour de l'objet à détecter par les caméras. Ensuite, nous les associons dans le logiciel, en créant une forme spécifique pour cet objet. Ce système est sensible aux symétries dans la forme, et nécessite d'avoir suffisamment de marqueurs pour être visibles par les caméras afin de détecter l'objet. Le robot a 5 marqueurs réfléchissants sur sa base et 5 sur sa tête tandis que 7 marqueurs ont été placés sur le casque RV *FOVE* pour suivre l'humain. Ces marqueurs peuvent être vus sur le casque RV *FOVE* sur la figure 7.3. Le système *Vicon* est sujet à la

perte de suivi qui peut être causée par l'occultation des marqueurs, la latence du réseau, de mauvaises conditions lumineuses... Même si ces pertes sont très brèves, puisque nous avons besoin d'une capture de mouvement en temps réel et précise, elles peuvent conduire à un comportement irréaliste des avatars et provoquer par exemple des saccades. Cela peut rompre l'immersion pour le participant humain, inclure des biais ou, pire encore, donner le mal des simulateurs [Rebenitsch and Owen, 2016]. C'est pourquoi un système de suivi supplémentaire est utilisé pour le robot et l'humain.

Pour le robot, le système de suivi *Vicon* est combiné aux capteurs odométriques du robot. Les *capteurs d'odométrie* estiment la position globale du robot dans l'environnement, et la position des articulations par rapport à la position globale. Le *capteur d'odométrie* de la base du robot est moins précis que le système de suivi *Vicon* (ou les capteurs des articulations) en raison des frottements mais ne souffre pas de perte de données. Au final, le mouvement du robot est estimé à partir des données des capteurs odométriques et sa position globale est périodiquement calibrée à l'aide des données *Vicon* afin d'éviter toute dérive sur la position globale. Cette estimation n'est utilisée que pour reproduire la pose réelle du robot dans l'environnement virtuel. Malgré tout, le robot estime sa pose et sa position globale en utilisant uniquement ses propres capteurs, afin de conserver les imperfections de l'odométrie du robot. En d'autres termes, les informations précises sur la position du robot sont nécessaires pour simuler les données correctement détectées en fonction de la configuration de la position du capteur.

Le participant quant à lui est équipé d'une combinaison de suivi de mouvement *Xsens*. La combinaison *Xsens* estime la position globale du pelvis du participant dans l'environnement, et la position des membres par rapport à la position globale du pelvis. L'IMU intégrée accumule les erreurs au fil du temps et la position globale estimée de la combinaison est sujette à une dérive notable après quelques minutes d'utilisation. Les positions des membres ne souffrent pas de cette dérive car elles sont situées par rapport à la position globale. Comme pour le robot, un étalonnage périodique de la combinaison *Xsens* est donc effectué avec le système *Vicon* pour éviter toute dérive sur la position globale.

7.3.3 Comment l'humain et le robot perçoivent le monde virtuel

D'abord, l'humain perçoit le VW via le sens de la vue et celui du toucher. La vision est gérée grâce à un casque de réalité de virtuelle (HMD) *Fove*. Alors que l'avatar humain est contrôlé par les systèmes de suivi que nous avons décrit

section 7.3.2, la rotation de la caméra, par laquelle l'humain perçoit le VW, est contrôlée par les capteurs IMU du *Fove*. L'utilisation des capteurs du *Fove* permet d'être plus précis dans le calibrage et réduit les problèmes de latence. Cette solution améliore l'expérience pour l'humain et diminue le mal des simulateurs. En outre, afin de donner à l'humain un plus grand sentiment de présence en RV, l'environnement 3D reproduit avec précision l'environnement réel et lui est superposé. Ainsi, lorsque l'humain touche virtuellement un mur virtuel, il touche également le mur réel.

Par ailleurs, le robot réel perçoit la VW à partir de capteurs simulés. Nous devons donc être en mesure de remplacer les capteurs intégrés au robot, ce qui pourrait ne pas être possible avec un robot commercial. En général, pour le robot *Pepper*, le logiciel *Naoqi*[5] est utilisé pour manipuler le robot. Le contrôleur bas niveau de ce robot est appelé *DCM*, qui fait partie de *Naoqi*, et manipule les dispositifs électroniques en fonction des commandes, puis met à jour la mémoire avec les valeurs des capteurs. Cependant, le *DCM* ne nous permet pas d'arrêter les vrais capteurs de fonctionner : nous avons donc résolu ce problème en travaillant au niveau de la mémoire même du robot, en écrasant la mémoire des vrais capteurs du robot dès que le DCM mettait à jour ses valeurs. Nous utilisons alors comme données de remplacement celles issues de capteurs simulés. La description de notre simulation de capteurs fait l'objet de la section 4.2.

7.4 Évaluation du système d'immersion humain-robot en RV

Cette section porte sur la mise en œuvre du système décrit en section 7.3.

Nous avons l'intention d'utiliser notre plateforme de RV pour étudier les interactions humain-robot. À cette fin, nous cherchons à évaluer la capacité de notre plateforme à mener ces études. En particulier, nous voulons explorer les comportements des robots et des humains en comparant une interaction simple réalisée à la fois en RV et dans la réalité.

Notre expérience est cependant pilote : nous n'avons pas l'intention d'explorer de nouveaux aspects de la perception et du comportement en RV. Nous souhaitons faire la preuve que ce concept peut être intéressant à explorer pour des expériences de HRI, mais qu'il convient de concevoir une expérience préliminaire. Cette ex-

5. http://doc.aldebaran.com/2-5/naoqi/sensors/dcm.html

périence est conçue pour éviter tout risque de collision entre l'humain et le robot. Nous voulons d'abord évaluer les interactions à distance qui se produisent dans notre plateforme. Il n'y aura donc pas de collisions (réelles ou virtuelles) dans cette expérience. Notre souhait est de constater si, dans une situation simple, notre système de RV ne biaise pas l'interaction homme robot.

Dans ce nouveau type de situation d'interactions entre un robot et un humain, nous devons vérifier que les biais relatifs à la perception humaine sont similaires à ceux qui ont été décrits dans la littérature. En effet, il est courant dans le domaine de la RV de comparer des situations similaires en RV et en conditions réelles pour évaluer un biais chez l'humain. De telles comparaisons ont été faites pour de nombreuses études différentes : locomotion vers un but [Agethen et al., 2018a, Cirio et al., 2013b], évitement des collisions [Agethen et al., 2018a, Bühler and Lamontagne, 2018, Fink et al., 2007, Olivier et al., 2017a], espace personnel [Gérin-Lajoie et al., 2008, Moussaïd et al., 2016], etc. La section 2.4 donne l'état de l'art sur la RV utilisée pour l'analyse du comportement humain. Toutes ces études montrent que, s'il existe certaines différences quantitatives, comme une vitesse de marche plus lente ou un espace personnel plus grand, les humains présentent des schémas et des comportements similaires en RV. Nous rapportons également nos observations concernant la détection du robot, mais la littérature n'offre pas ce type d'études pour les robots (les robots virtuels étant systématiquement simulés).

Notre évaluation est basée sur l'expérience de Ducourant et al. [2005] qui aborde la question de la coordination interpersonnelle lorsqu'une paire de participants marchent de concert. Dans notre cas, nous remplaçons l'un des humains par un robot. Nous mettons le robot et l'humain face à face et nous donnons l'instruction, alternativement au robot et à l'humain, d'être le leader, et d'effectuer un mouvement de va-et-vient. L'autre est le suiveur, et doit garder une distance constante avec le leader. La tâche est davantage précisée dans la section suivante 7.4.1. Nos hypothèses sont les suivantes :

— **H1** : dans des conditions virtuelles et réelles, les trajectoires du participant et du robot seront fortement corrélées dans le temps.

— **H2** : la condition virtuelle induira un temps de réaction du robot et du participant plus élevé que la condition réelle.

— **H3** : dans la condition virtuelle, la vitesse de l'humain sera inférieure à celle de la condition réelle, car c'est un fait communément observé [Fink et al., 2007].

7.4.1 Protocole expérimental

Nous présentons ici notre protocole expérimental.

Population

Pour cette étude pilote, 7 participants, non rémunérés, recrutés via des listes de diffusion internes parmi les étudiants et le personnel de l'institut, se sont portés volontaires pour l'expérience. (2F, 5M ; âge : moyen $= 27,2 \pm 3,89$, min=21, max=42). Ils étaient tous naïfs quant à l'objectif de l'expérience, avaient une vision normale ou corrigée à la normale et ont donné leur consentement écrit et éclairé. L'étude était conforme à la déclaration d'Helsinki, et a été approuvée par le comité d'éthique local. En raison des problèmes liés aux systèmes de suivi qui sont apparus pour 2 participants, seules les données de 5 participants ont finalement pu être utilisées pour l'analyse.

Conditions

Les tâches, détaillées ci-dessous, ont été effectuées dans deux conditions : réelles et virtuelles. La condition par laquelle un participant a commencé a été choisie au hasard.

— Conditions réelles : c'est notre situation de référence. Le robot et l'humain partagent le même espace de travail physique. Il est à noter que la pose de la tête du participant a été suivie par le port de lunettes avec des marqueurs réfléchissants. Ces lunettes sont visibles sur la figure 7.5 colonne "condition réelle".

— Condition virtuelle : le participant est immergé avec le robot dans une copie virtuelle de l'espace de travail du participant. Ils sont physiquement séparés, dans leurs propres espaces de travail réels respectifs, comme indiqué dans la figure 7.2.

Tâches

Nous proposons 2 tâches d'interaction inspirées de Ducourant et al. [2005]. Nous définissons la distance interpersonnelle comme étant la distance euclidienne entre le participant et le robot. Les participants ont été invités à se déplacer en coordination avec le robot, et ont toujours commencé par la tâche 1 :

— Tâche 1 : Le robot dirige l'interaction. Il a été demandé aux participants de rester devant le robot et de maintenir la distance de départ ($1,5\ m$) avec le

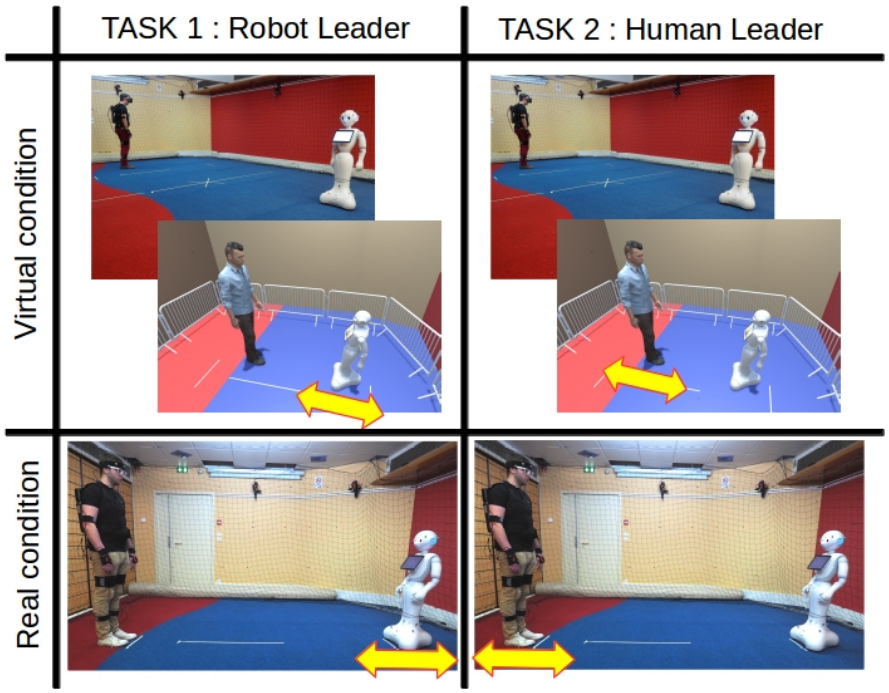


Figure 7.5 – Tâches et conditions : le *leader* se déplace en avant et en arrière avec un mouvement sinusoïdal. Le suiveur essaye de rester devant le suiveur et de maintenir une distance interpersonnelle constante entre le leader et le suiveur. Le leader est soit le robot (tâche 1), soit le participant (tâche 2), et la tâche est exécutée en RV ou en condition réelle

robot entre celui-ci et eux. Nous avons appliqué une commande de vitesse sinusoïdale au robot avec une fréquence de $0,15\ Hz$ et une amplitude de $0,12\ m/s$, le faisant avancer et reculer, comme indiqué dans la figure 7.5, pendant une minute.

— Tâche 2 : le participant dirige l'interaction et il lui est demandé d'avoir un comportement similaire à celui du robot de la tâche 1, comme indiqué dans la figure 7.5. Le participant doit avancer et reculer dans un mouvement périodique tout en faisant face au robot à tout moment. Le robot, à l'aide de ses capteurs réels ou virtuels, essaye de maintenir une distance interpersonnelle constante de $1,5\ m$ pendant une minute, et essaye de rester face à face avec le participant. La détection du sujet humain a été faite à l'aide du réseau neuronal de détection d'objets en temps réel tiny−YOLO [Redmon et al., 2016]. Le résultat retourné par le réseau a été utilisé pour le contrôle de l'orientation. La distance à l'humain a été donnée par des capteurs de proximité (US et LIDAR).

Déroulement de l'expérience

L'environnement, représenté dans la figure 7.2, était un espace vide formé par un rectangle de 8m x 5m, entouré de murs et d'un filet de protection. La pièce était séparée en deux carrés de 4m x 5m avec un marquage au sol, délimitant l'espace de travail du robot et l'espace de travail des participants. Les participants ont commencé soit en condition virtuelle, soit en condition réelle, alternativement. Au début de la condition virtuelle, nous avons suggéré aux participants de se déplacer et de sentir la correspondance entre les murs réels de la pièce et les murs virtuels de la VW. Après un entraînement d'une minute à effectuer la tâche, les participants ont effectué deux essais d'une minute chacun pour chacune de ces conditions. Au total, chaque participant a effectué 8 essais d'une minute (2 Conditions $\times$ 2 Tâches $\times$ 2 répétitions). Un recalibrage du système de suivi a été effectué entre chaque essai.

Données collectées

Au cours de l'expérience, les trajectoires du participant et du robot ont été enregistrées, ainsi que les capteurs du robot. L'acquisition des données a utilisé le système *Vicon* en condition réelle, et la combinaison *Xsens* et l'odométrie interne du robot en condition virtuelle. Les données ont été acquises avec une fréquence de 120 Hz. Toutefois, puisque la communication réseau repose sur le *ROS* qui ne garantit pas les contraintes dures en temps réel, il existe un risque que la fréquence ne soit pas précisément respectée par le système.

7.4.2 Résultats

Nous avons échantillonné les données avec une fréquence fixe de 100 Hz et les avons filtrées avec un filtre linéaire *Forward/Backward*, sur chacun des enregistrements de 60 secondes. Nous avons choisi de limiter l'analyse à un axe correspondant à la direction préférentielle de l'interaction. Nous avons normalisé les données dans le temps : nous avons supprimé les 5 premières secondes de chaque enregistrement qui correspondent au temps pendant lequel le robot se stabilise dans la tâche 2. Les statistiques prennent en compte toutes les valeurs de chaque essai (5 participants par tâche, avec 2x55 secondes d'enregistrement pour chaque condition et chaque tâche).

Comme pour Ducourant et al. [2005], nous proposons d'analyser les similitudes et les retards entre les positions du leader et du suiveur (qui correspondent au robot ou à l'humain selon la tâche) grâce à une corrélation croisée partielle dans

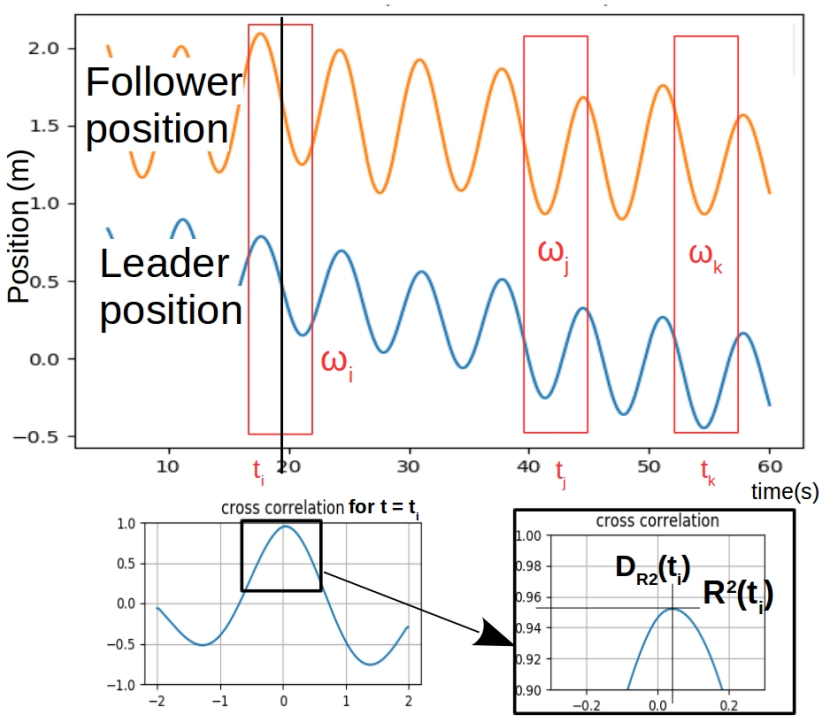


Figure 7.6 – Calcul de la corrélation croisée et observation du retard sur le temps. Exemple d'essai : pour un temps donné t_i, on définit une fenêtre temporelle ω_i centrée sur t_i. Nous effectuons une corrélation croisée normalisée entre la position de leader et la position de suiveur. Le retard D_{R^2} pour t_i et la corrélation R^2 pour t_i sont respectivement l'ordonnée et l'abscisse du point de valeur maximale de la fonction donnée par la corrélation croisée normalisée. Le processus est répété pour chaque étape temporelle de l'essai, ce qui nous donne un ensemble de valeurs affichées sur Figure 7.7.

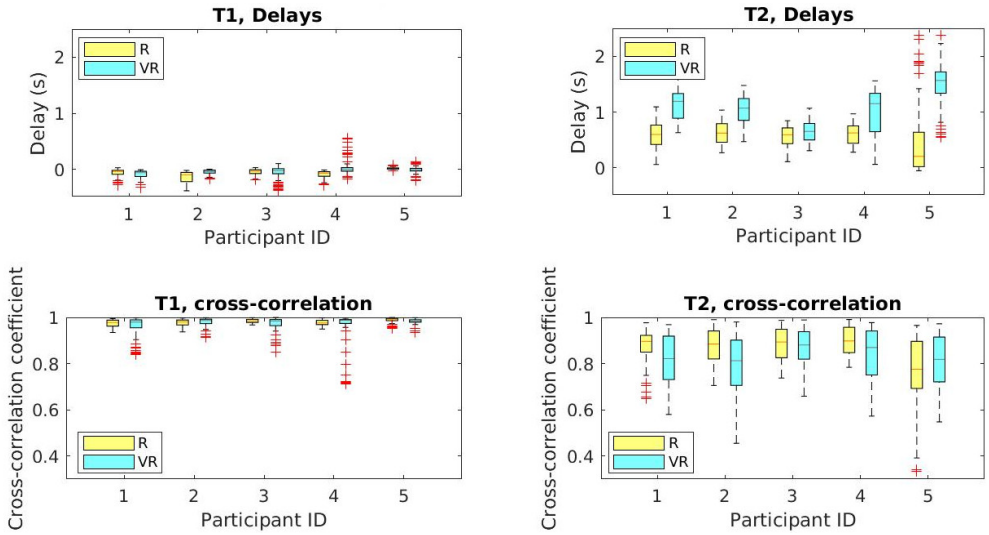


Figure 7.7 – Diagrammes-boîtes des retards et des corrélations croisées pour chaque participant, pour chaque tâche et chaque condition

	Robot velocity		Human velocity	
	Real	VR	Real	VR
T1	-0.006±0.012	0.008±0.034	-0.002±0.013	0.007±0.024
	-0.336 ; 0.349	-0.432 ; 0.400	-0.336 ;0.369	-0.472 ;0.477
T2	0.006±0.086	0±0.059	0.001±0.045 ;	0.005±0.052
	-0.422 ; 0.373	-0.563 ; 0.496	-0.270 ; 0.363	-0.421 ; 0.360

Table 7.2 – Moyenne et écart-type (première ligne d'une tâche), ainsi que l'amplitude (deuxième ligne d'une tâche), des vitesses du robot et de l'humain pour chaque condition et chaque tâche.

le temps pour chaque essai (c'est-à-dire effectué sur une fenêtre temporelle mobile). Notre fenêtre temporelle est longue de $10s$, et se déplace par pas de $2,5ms$. Figure 7.6 détaille notre calcul sur un exemple d'essai. Plus précisément, nous estimons la variation du coefficient de corrélation de Pearson dans le temps sur la base d'une corrélation croisée normalisée, ainsi que la variation du délai entre les deux signaux. Ce processus aboutit à une fonction de coefficient de Pearson ainsi qu'à une fonction de retard dans le temps. Leur distribution est tracée pour chaque participant et chaque condition dans la figure 7.7.

Nous utilisons un test de Lilliefors [1967] sur les scores et les retards de corrélation croisée, qui a rejeté l'hypothèse de normalité pour chaque condition et tâche ($p < 0,05$ pour chaque test). Nous utilisons ensuite un test de rang signé Wilcoxon entre les deux conditions, pour chaque tâche et pour chacune de ces variables. Ces tests ont rejeté l'hypothèse nulle de la médiane zéro ($p < 0,05$), sauf pour les données de corrélation croisée sur la tâche 1 ($p = 0,4620$).

Pour chaque tâche et condition, nous avons calculé la distance interpersonnelle, à chaque pas de temps pour chaque épreuve de chaque participant, qui a une valeur de [avg :1.39 ; std :0.05] m et [avg :1.03 ; std :0.06] m pour la tâche 1 en condition réelle et en condition virtuelle respectivement, et [avg :1.73 ; std :0.17] m et [avg :1.74 ; std :0.12] m pour la tâche 2 en condition réelle et en condition virtuelle respectivement.

Nous avons également calculé la vitesse du robot et du participant, résumée dans le tableau 7.2.

7.4.3 Analyse

Nous examinons ici les résultats présentés dans la section 7.4.2.

Tout d'abord, nous remarquons une différence de vitesse du robot entre les conditions réelles et virtuelles. Le robot va plus vite en RV. Cela est dû à l'erreur cumulée par ses capteurs d'odométrie qui est expliquée dans le paragraphe 7.3.2.

Influence de la RV dans l'interaction humain-robot

Les résultats montrent des similitudes évidentes entre les conditions réelles et virtuelles. Il existe toutefois des différences quantitatives entre les conditions réelles et virtuelles.

Tout d'abord, le test de classement signé par Wilcoxon sur les délais entre les deux conditions rejette l'hypothèse d'égalité. De plus, on peut observer que les retards sur la figure 7.7 sont légèrement plus élevés en RV qu'en condition réelle sur la tâche 2. Nous supposons que le système décrit dans la figure 7.3 a une latence causée par le réseau. Ainsi, notre hypothèse **H2** est vérifiée. Les retards sur la tâche 1 sont proches de 0 pour les deux conditions, ce qui signifie que le participant est capable de réaliser à temps que le robot a changé de vitesse. Comme les participants ont pu entendre le mouvement du robot (le robot et le participant se trouvent dans la même pièce), nous supposons qu'en RV ils ont parfois anticipé le mouvement du robot, ce qui explique les valeurs positives de la tâche 1 pour le participant 4. Dans l'expérience de Ducourant et al. [2005], où le leader et le suiveur sont tous deux des humains, l'auteur suggère qu'au lieu d'anticiper, les suiveurs ont tendance à être plus réactifs. Dans notre cas, lorsque le robot était le suiveur, il était également réactif.

Cependant, lorsque le robot était le leader, il se déplaçait assez lentement et avec un mouvement prévisible : l'humain pouvait donc parfaitement anticiper le mouvement du robot efficacement au lieu d'être réactif. Contrairement à Ducourant et al. [2005] qui observe que les vitesses sont plus élevées lorsque le leader marche en arrière et le suiveur en avant, nous n'observons pas cela dans notre situation. Nous pensons que la vitesse maximale faible est en cause ici, et qu'il était facile pour les humains de suivre le robot. Cependant, cela pourrait être dû aussi à la manière dont le robot est contrôlé : pour des vitesses plus élevées, différents contrôleurs pourraient avoir une influence différente sur les vitesses d'avance et de recul du robot et de l'homme, en tant que leader ou suiveur. On remarque également que la distance interpersonnelle moyenne pendant la tâche 1 montre que la variable est plus petite dans la réalité virtuelle qu'en condition réelle. Il est possible que les humains éprouvent une peur d'une éventuelle collision dans le scénario réel, peur qui est atténuée dans la RV. De plus, lorsque le robot était en tête, la distance de sécurité entre le robot et l'humain n'a jamais été atteinte : l'humain semble avoir un espace personnel plus grand que notre robot. Par conséquent, les humains n'ont presque jamais été surpris par des arrêts brutaux, ce qui renforce l'idée que l'humain peut anticiper les mouvements du robot.

La figure 7.7 montre en outre un score élevé de corrélation croisée pour chaque

condition pour la tâche 1, et les données de corrélation croisée pour la tâche 1 ont passé le test de classement signé Wilcoxon. Cela vérifie notre hypothèse **H1** pour cette tâche, mais pas pour la seconde. Nous expliquons les scores inférieurs de corrélation croisée pour la tâche 2 par le fait que les participants avaient des rythmes différents entre eux, alors que le robot se déplaçait toujours à la même vitesse dans la tâche 1 avec chaque participant. De plus, les participants étaient attentifs aux mouvements du robot pendant la tâche 2 : ils attendaient le robot parce qu'il se déplaçait lentement, alors que le robot de la tâche 1 était toujours en mouvement. Nous observons ce comportement dans les deux conditions, ce biais n'est donc pas dû à la réalité virtuelle.

De plus, la vitesse des participants pour les tâches 1 et 2 est centrée sur zéro, ce qui diffère des résultats de Ducourant et al. [2005], et a des valeurs maximales plus élevées en RV, ce qui contredit notre hypothèse **H3** et Fink et al. [2007]. Cependant, le robot a **surestimé** sa vitesse en RV : nous ne pouvons donc pas évaluer clairement le biais ici. Pour la seconde tâche, la faible vitesse du robot ($< 0,35m/s$) a contraint le participant à se déplacer dans les deux conditions, comme expliqué dans le paragraphe précédent. Nous pensons que la vitesse de marche par défaut est plus petite en RV mais dans notre expérience, les participants n'atteignent même pas leur vitesse de marche par défaut. L'hypothèse **H3** n'est donc pas validée dans ce cas précis.

Enfin, dans l'expérience de Ducourant et al. [2005], les trajectoires ont été corrélées, les sujets étaient donc totalement impliqués dans une interaction bidirectionnelle avec des influences mutuelles. Nous observons cela lorsque l'expérience implique un robot : l'humain et le robot s'observaient et s'imitaient l'un l'autre . Même avec un contrôleur aussi simple pour le robot, toute l'interaction a été préservée grâce à l'humain.

7.5 Conclusion

Dans ce chapitre, nous présentons une extension de notre simulation et détaillons une nouvelle plateforme de RV conçue pour étudier les interactions entre les robots et les humains. Sa principale innovation est d'immerger à la fois un humain et un robot dans un environnement virtuel partagé. La perception visuelle du robot et celle de l'homme sont toutes deux synthétiques, mais le robot et l'homme sont capables d'effectuer des actions comme dans des conditions réelles. C'est une méthode sûre pour étudier les interactions entre l'homme et le robot, et elle dépasse le niveau de réalisme qui peut être atteint dans un environnement de simulation pure. En particulier, la réponse humaine au comportement du robot

serait difficile à prévoir en se basant uniquement sur des algorithmes.

Dans ce travail, nous nous sommes penchés sur les biais de perception introduits par l'utilisation des technologies de RV. Leurs effets sur les interactions dans notre étude pilote, une simple tâche de coordination effectuée entre un humain et un robot, sont limités. Ainsi, il semble que la RV pourrait être utilisée pour réaliser des expériences HRI, mais les résultats quantitatifs doivent être interprétés avec précaution, car de légères différences peuvent être trouvées par rapport aux expériences réelles.

Nous pensons que notre plateforme comble ainsi un manque entre la simulation et les expériences en conditions réelles. Elle offre des conditions plus réalistes que la simulation, au prix de la mise en place de plus de matériel et du recrutement de participants qui prend du temps, ce qui signifie moins de données que si nous utilisions la simple simulation. Elle offre également plus de contrôle, de reproductibilité des expériences et de sécurité que les expériences en conditions réelles, au prix d'une configuration complexe qui génère des données moins réalistes que les expériences en conditions réelles.

Nous pensons que notre plateforme offre toutefois de nouvelles capacités pour les expériences de RV sur les HRI, nous permettant de développer directement de nouvelles méthodologies pour étudier les HRI.

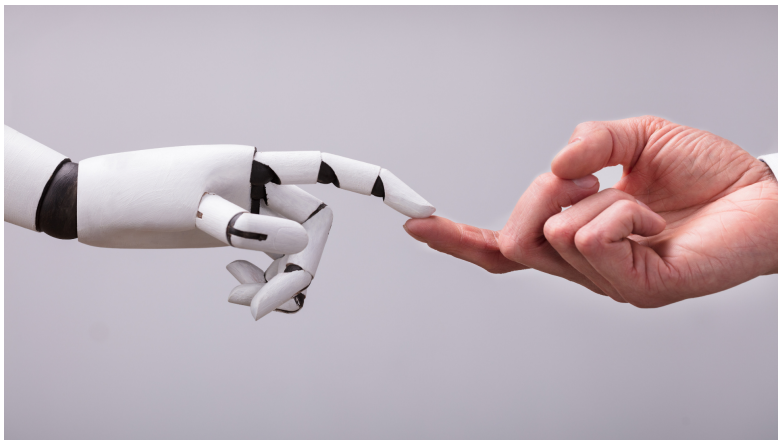
Il semble que la latence, induite par le système décrit dans la section 7.3.1, pourrait être une source majeure de perturbation pour le participant, et peut provoquer le mal des simulateurs. La fréquence des mises à jour des configurations dans le monde de la RV, tant pour l'homme que pour le robot, était de 20,6 Hz pendant l'expérience présentée. Cette fréquence est trop faible pour pouvoir assurer une expérience fluide.

En outre, la vitesse du robot a été limitée (<0,35 m/s), ce qui a limité la portée des résultats. Certains participants ont déclaré avoir entendu le robot réel se déplacer pendant qu'ils étaient en réalité virtuelle et ont utilisé ce son du robot réel se déplaçant dans son propre espace de travail, pour synchroniser leur mouvement dans la RV. Certains participants ont également été perturbés par leur avatar, qui visuellement ne correspondait pas parfaitement à leur propre corps. De plus, le HMD *Fove* que nous avons utilisé a un champ de vision limité, ce qui a un impact sur l'immersion des participants. En ce qui concerne l'expérience pilote, le nombre de participants est faible (5). Cependant, il est suffisant pour faire une preuve de concept et pour identifier les moyens d'amélioration. Enfin, les tâches proposées imposent des contraintes de vitesse aux participants, et induisent un manque de variation car la commande sinusoïdale du robot est toujours la même pendant toute l'expérience. Cela limite la variété des interactions, donc la portée

des résultats.

Aller plus loin dans l'immersion...

L'interaction humain-robot présentée dans ce chapitre se limite à une interaction à distance. Cependant, notre objectif est bien d'étudier les interactions proches, lorsque les risques de collision sont élevés. Lors d'une collision, l'humain *touche* le robot. Ainsi, étudier une telle interaction en VR nécessite de prendre en compte le fait que les robots et les humain se touchent réellement. Le chapitre suivant explore la question de restituer les contacts liés à une interaction en RV.



Chapitre 8

Restitution des contacts lors d'interactions virtuelles proches

Sommaire

Dans ce chapitre, nous explorons des méthodes de rendu haptique pour augmenter l'immersion des participants lors des expériences en RV. On l'a vu, la RV offre des possibilités intéressantes pour l'étude des comportements humains, notamment à traveres la capacité de contrôler avec précision les conditions expérimentales, de les reproduire sur différents sujets et essais, et d'accéder facilement aux informations environnementales (voir tableau 7.1).

Cependant, l'immersion peut être gâchée par des stimulus du monde réel incohérents avec le monde virtuel. Par exemple, un son provenant du monde réel peut priver le participant de son immersion dans le monde virtuel, comme une porte qui claque, des bruits de pas. Un autre exemple : on a vu dans le chapitre 7 que l'utilisation d'un avatar pour notre humain servait à la perception de l'interaction pour notre robot. Mais l'avatar influe également sur le sentiment de présence de la personne dans le monde virtuel. Cet avatar doit donc être adapté à la personne, car dans le cas contraire, la personne perçoit une incohérence ("je ne suis pas dans mon propre corps") et n'est plus correctement plongée dans le monde virtuel.

Enfin, un autre sens impliqué dans l'immersion est celui du toucher. En effet, si on considère que la personne possède un avatar en RV qui lui correspond bien, elle pourrait être tentée de toucher des objets virtuels. A l'instant du contact, il y a une incohérence entre le sens du toucher et celui de la vue : la personne se voit toucher un objet mais ne le sent pas. Lors d'une interaction proche homme-robot, il y a pourtant de grands risques de contacts. Si on souhaite étudier une telle interaction en réalité virtuelle, comme vu au le chapitre 7, alors il faut explorer la question du rendu de ces contacts pour l'humain. Dans ce contexte, l'approche présentée dans ce chapitre propose d'utiliser des brassards haptiques dans une expérience de navigation dans une foule virtuelle pour augmenter l'immersion des participants.

8.1 Immersion d'humains dans la foule

On vu dans les chapitres 2 et 7 l'intérêt d'utiliser la RV pour l'étude des comportements humains. Cependant, les comportements d'une personne peuvent être

Figure 8.1 – Notre objectif est de comprendre si et dans quelle mesure le fait de fournir un rendu haptique des collisions lors de la navigation dans une foule virtuelle (à droite) permet aux utilisateurs de se comporter de manière plus réaliste. Chaque fois qu'une collision se produit (au centre), les brassards portés sur les bras vibrent localement pour rendre compte de ce contact (à gauche). Nous avons réalisé une expérience avec 23 participants, et utilisé des mesures subjectives et objectives relatives à la planification de la trajectoire des utilisateurs, les mouvements du corps, l'énergie cinétique, la présence et l'incarnation.

différents en VR par rapport à la réalité. A cet égard, des efforts significatifs ont été déployés dans l'évaluation du réalisme des comportements en RV [Cirio et al., 2013c, Olivier et al., 2017b], dans la compréhension des informations visuelles requises [Lynch et al., 2017] ou dans le développement d'environnements et de personnages de RV hautement réalistes [Achenbach et al., 2017].

Cependant, malgré ces efforts, la plupart des expériences de RV manquent encore de toute sensation haptique, d'une importance pourtant capitale lorsqu'on étudie le comportement et les interactions des foules. Par exemple, si nous sommes incapables de rendre la sensation de se heurter à des personnages virtuels en naviguant dans un environnement bondé, les participants pourraient cesser d'éviter les collisions, ce qui conduit à collecter des données qui ne saisissent pas bien comment les humains se comportent réellement.

C'est pourquoi les études du comportement collectif en RV sont souvent limitées à des cas ne prenant en compte que les interactions à distance [Rio et al., 2018, Rio and Warren, 2014], afin d'éviter d'avoir besoin de retour haptique.

Ce chapitre explore donc le rôle des interactions de contact (collisions) pendant la navigation dans une foule virtuelle dense, comme le montre la figure 8.1. Pour ce faire, nous utilisons un ensemble d'interfaces haptiques portables capables de fournir des sensations vibrotactiles de contact aux bras de l'utilisateur. En d'autres termes, des brassards vibrants.

Notre objectif est d'étudier si, et dans quelle mesure, le rendu des contacts influence le comportement de l'utilisateur dans ce contexte. Il s'agit aussi de limiter l'apparition de certains artefacts bien connus, comme lorsque l'avatar virtuel

de l'utilisateur inter-pénètre d'autres personnages virtuels. Nous avons mené une expérience (N=23) où les participants étaient équipés de quatre interfaces haptiques portables (deux sur chaque bras), et étaient invités à naviguer dans une gare virtuelle très fréquentée.

Notre évaluation repose sur des mesures objectives liées au comportement de l'utilisateur par rapport à la foule, ainsi que des mesures subjectives liées au sentiment de présence et d'incarnation de l'utilisateur. Nous avons d'abord réalisé l'expérience sans rendu haptique des contacts, puis avec un rendu haptique, et enfin une fois de plus sans rendu haptique. Ce protocole expérimental nous permet d'enregistrer la différence de comportement de l'utilisateur lors de l'activation du retour haptique ainsi que la persistance d'un éventuel effet post-activation.

8.2 Résumé de l'expérience

L'objectif de cette étude est d'étudier l'effet du rendu haptique des collisions sur le comportement des participants pendant la navigation à travers une foule statique en RV. Pour explorer cette question, nous avons immergé les participants dans une gare virtuelle et leur avons demandé d'effectuer une tâche de navigation qui consistait à se déplacer à travers une foule de personnages virtuels. Dans certaines conditions, les collisions avec les personnages virtuels ont été restituées aux participants à l'aide de 4 dispositifs haptiques vibrotactiles portables (avec des brassards). Notre hypothèse générale est que le rendu haptique modifie le comportement des participants en leur donnant un retour d'information sur les collisions virtuelles. En outre, nous pensons que même après avoir supprimé le rendu haptique, un effet secondaire persiste sur le comportement des participants.

Afin d'immerger les participants dans l'environnement virtuel et d'étudier les effets potentiels du rendu haptique tout en naviguant dans des groupes de personnages, nous avons utilisé les dispositifs suivants, qui sont résumés dans la figure 8.2 :

— **Suivi du mouvement** : pour enregistrer les mouvements du corps des participants, ainsi que pour animer leur avatar dans la scène, nous avons utilisé le système *Xsens* décrit dans le chapitre 7 (section 7.3.1).

— **HMD** : pour immerger les participants dans l'environnement virtuel, nous avons choisi d'utiliser un casque de réalité virtuelle Pimax [1], notamment en

1. https://www.pimax.com/

raison du large champ de vision offert dans ces situations de proximité avec d'autres personnages (spécifications : 90 Hz, 200° fov, 2560 × 1440 de résolution), avec 4 stations de base SteamVR 2.0, offrant une zone de suivi d'environ 10×10 m. Cette configuration a permis aux participants de marcher physiquement dans l'espace réel, tandis que leurs mouvements de marche étaient reproduits par leur avatar adapté à leur taille dans l'environnement virtuel.

— **Rendu haptique** : pour le rendu des collisions entre les participants et les personnages virtuels, nous avons équipé les participants de quatre brassards haptiques (un sur chaque bras et chaque avant-bras)[Scheggi et al., 2016]. Chaque brassard est composé de quatre moteurs vibrotactiles dont la fréquence de vibration se situe entre 80 et 280 Hz et qui sont contrôlés indépendamment. Les moteurs sont positionnés de manière uniforme sur une bande de tissu élastique (figure 8.3). Une carte électronique contrôle le matériel. Elle comprend un Arduino Mini Pro à 3, 3 V, une batterie Li-on à 3, 7 V et une antenne Bluetooth 2.1 pour la communication sans fil avec la station de contrôle externe.

— **Ordinateur sac à dos** : pour permettre aux participants de se déplacer librement dans l'environnement, ils ont été équipés d'un ordinateur sac à dos MSI VR One, qui gère l'expérience. Tous les appareils étaient connectés directement à cet ordinateur (spécifications : NVidia GTX 1070, processeur Intel Core i7-7820HK, 32 GB RAM).

8.2.0.1 Rendu haptique

Le rendu haptique nécessite que les collisions soient détectées dans l'environnement RV. Comme des dispositifs haptiques étaient portés aux bras des participants, nous avons détecté les collisions entre leur avatar (animé à l'aide du système de capture de mouvement Xsens) et la foule virtuelle. À cette fin, nous avons segmenté le bras de chaque avatar en trois parties (bras, avant-bras et main), et attaché à chaque segment un *"capsule Collider"* Unity qui signalait les collisions avec d'autres objets de la scène (voir la figure 8.4).

À la détection d'une collision, c'est-à-dire si l'un des six segments de l'avatar entre en collision avec la géométrie d'un personnage de foule virtuelle, l'un des quatre dispositifs haptiques est activé. Plus précisément, les *Collider* de l'avant-bras virtuel gauche (resp. droit) et de la main ont activé le brassard situé sur l'avant-bras gauche (resp. droit) du participant, tandis que les *Collider* du bras virtuel gauche (resp. droit) ont activé le brassard situé sur le bras gauche (resp.

Figure 8.2 – Dispositifs portés par les participants pendant l'expérience.

droit) du participant. En termes de vibrations, chaque vibromoteur d'un brassard était piloté par un seul paramètre appelé *taux vibrotactile*, qui contrôlait à la fois l'amplitude et la fréquence des vibrations. Au cours de l'expérience, tous les moteurs d'un brassard activé ont donc été contrôlés à l'aide du même *taux vibratoire*, qui variait selon un profil d'onde sinusoïdale de 10 Hz par période. La variation du *taux vibratoire* a entraîné une fréquence de vibration de l'ordre de [57–126] Hz. Bien que ces moteurs puissent vibrer jusqu'à 255 Hz, nous avons décidé de limiter leur fréquence après que des participants à une pré-étude aient déclaré que l'intensité de la vibration, proportionnelle à la fréquence, était trop forte.

La communication avec les brassards était effectuée à 4 Hz, ce qui signifie que les collisions d'une durée inférieure à 250 ms n'étaient pas transmises aux participants, et qu'il y avait un délai maximum de 250 ms pour activer (resp. arrêter) les brassards après qu'une collision ait été détectée (resp. terminée).

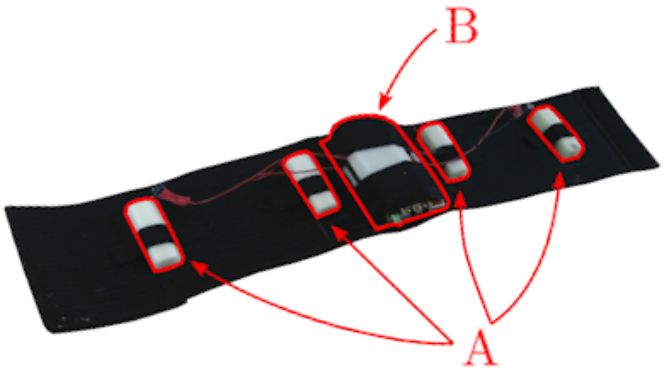
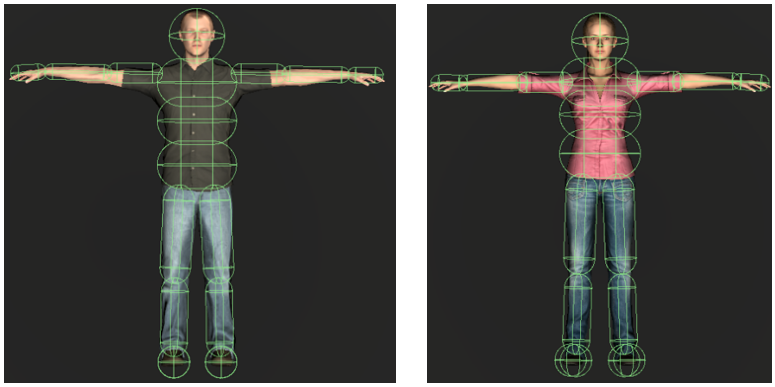


Figure 8.3 – Brassards vibrotactiles, composés de quatre moteur vibrants (A). L'électronique est protégée par un boîte imprimée en 3D (B). [Scheggi et al., 2016].

8.2.1 Environnement & Tâche

Les participants ont été immergés dans une reproduction numérique de la station de métro *"Mayakovskaya"* à Moscou, au milieu d'une foule virtuelle statique (voir la figure 8.5). Au total, huit configurations différentes de la scène ont été préparées à l'avance et utilisées dans le cadre de l'expérience. Une configuration est définie par la position exacte de chaque personnage de la foule dans la station virtuelle. Dans chaque configuration, la foule a formé une forme carrée, et les positions des personnages ont suivi une distribution de Poisson résultant en une densité de $1,47 \pm 0,06$ personne/m^2. Une telle distribution combinée à un tel niveau de densité garantit qu'un écart de $0,60$ m en moyenne existe entre chaque personnage. La foule est constituée de personnages virtuels animés par diverses animations d'attente (simplement de petits mouvements sur place). Dans chaque configuration, les personnages suivaient potentiellement deux types de comportement : soit l'attente (orientés de façon à faire face au tableau affichant les horaires des trains, en bougeant légèrement le haut du corps) soit l'appel téléphonique (avec une orientation aléatoire). Nous avons utilisé plusieurs clips d'animation pour chacun des deux comportements, afin d'éviter que le même clip d'animation ne soit utilisé par deux personnages virtuels différents.

Au début de chaque essai, les participants se tenaient d'abord à un coin de la foule carrée, incarnés dans un avatar correspondant à leur genre (voir la figure 8.4). Ils devaient traverser la foule de manière à atteindre le tableau affichant les horaires des trains, et lire à haute voix le numéro de voie du prochain train affiché au tableau avant de revenir à leur position initiale. Ils marchaient naturellement dans une pièce réelle, et leur corps était localisé afin que leur avatar agisse de la même manière. Cette tâche exigeait des participants qu'ils atteignent le coin opposé de l'espace afin de lire les informations au tableau, tout en les obligeant à



a) b)

Figure 8.4 – Les avatars masculins (a) et féminins (b) utilisés pour représenter les participants dans l'environnement virtuel. Pour les deux avatars, la capsule autour de chaque segment représente le solide utilisé pour calculer les collisions, appelé *Collider*.

se déplacer dans la foule virtuelle. De plus, l'écran n'affichait les informations sur le train que lorsque les participants étaient à moins de 2 m de celui-ci (c'est-à-dire lorsqu'ils atteignaient la zone verte affichée dans la figure 8.5.b). En outre, nous avons fourni l'instruction suivante aux participants avant l'expérience : "*Marchez dans la gare virtuelle comme si vous étiez dans une vraie gare*".

8.2.2 Protocole expérimental

À leur arrivée, les participants ont été invités à remplir un formulaire de consentement, après leur avoir présenté la tâche à accomplir. Ils ont ensuite été équipés des équipements montrés sur la figure 8.2. Le calibrage du système de capture de mouvement Xsens a ensuite été effectué pour assurer la qualité de la capture, ainsi que pour redimensionner l'avatar aux dimensions des participants. Une fois prêts, les participants ont alors effectué un essai d'entraînement au cours duquel ils ont pu explorer l'environnement virtuel et se familiariser avec la tâche.

L'expérience a ensuite consisté en 3 blocs de 8 essais, où les blocs ont été présentés pour tous les participants dans l'ordre suivant : *NoHaptic1*, *Haptic*, et *NoHaptic2*. Le bloc *Haptic* correspondait à l'exécution de la tâche avec un rendu haptique des contacts, tandis que les blocs *NoHaptic* n'impliquaient aucun rendu haptique des contacts. L'expérience a donc consisté à réaliser d'abord un bloc sans rendu haptique, afin de collecter une base de référence des réactions des participants. L'objectif du bloc *Haptic* était ensuite d'étudier si l'introduction du rendu haptique influençait leur comportement lorsqu'ils naviguaient dans une foule, tandis que l'objectif du bloc *NoHaptic2* (sans rendu haptique) était de

a) b)

Figure 8.5 – Des aperçus de l'environnement sous deux points de vue différents. Les participants sont partis de la croix bleue sur le sol, et ont été invités à atteindre le tableau d'affichage. La figure (b) montre un exemple de trajectoire représentée par une ligne pointillée rouge. L'écran n'a affiché les informations sur le train que lorsque les participants ont atteint la zone verte.

mesurer un éventuel apprentissage. Dans chaque essai, les participants ont effectué une fois la tâche. Chaque bloc était composé de 8 essais, correspondant aux 8 configurations de foule, réalisés dans un ordre aléatoire. À la fin de chaque bloc, les participants ont été invités à répondre aux questionnaires *Incarnation* et *Présence* (Tables 8.2, 8.3, 8.4 & 8.5) tout en restant dans l'environnement virtuel (pour des raisons pratiques, afin de conserver une calibration correcte des équipements entre les blocs). Enfin, à la fin de l'expérience, les participants ont rempli un questionnaire démographique.

8.2.3 Participants

Vingt-trois participants non rémunérés, recrutés via des listes de diffusion internes à l'Inria parmi les étudiants et le personnel, se sont portés volontaires pour l'expérience (8F, 15M ; âge : avg=26 ± 6, min=18, max=43). Ils étaient tous naïfs quant à l'objectif de l'expérience, avaient une vision normale ou corrigée, et ont donné leur consentement écrit et éclairé. L'étude était conforme à la déclaration d'Helsinki, et a été approuvée par le comité d'éthique interne de l'Inria (COERLE).

8.2.4 Données collectées

Au cours de l'expérience, nous avons enregistré à 45 Hz les trajectoires des participants, ainsi que la position et l'orientation des membres de leur corps dans l'environnement virtuel à l'aide des capteurs *Xsens* et *Unity*. Nous avons également

enregistré les positions des corps évoluant au fil du temps de chaque personnage de la foule virtuelle. Ensuite, nous avons pu rejouer hors ligne l'ensemble des essais afin de calculer des opérations complexes telles que le volume de chaque collision.

8.2.5 Hypothèses

Nos hypothèses sont les suivantes :

$H1$: Le rendu haptique ne changera pas le chemin suivi par les participants à travers la foule. En effet, les piétons se fient principalement à la vision pour contrôler leur locomotion [Patla, 1997, Warren, 1998]. Or nous avons reproduit chaque configuration de la foule sur les 3 blocs, ce qui donne des informations visuelles identiques pour la navigation des participants. Par conséquent, le chemin suivi sera similaire dans les trois blocs de l'expérience (*NoHaptic1*, *Haptic* et *NoHaptic2*).

$H2$: Le rendu haptique des collisions fera prendre conscience aux participants des collisions et influencera leur mouvement corporel pendant la navigation à travers la foule. C'est pourquoi, en ce qui concerne les blocs *NoHaptic1* et *Haptic* de l'expérience, nous nous attendons aux hypothèses suivantes :

- $H2_1$: Les participants navigueront dans la foule plus prudemment dans le bloc *Haptic* afin d'éviter les collisions. Il y aura plus de mouvements d'évitement locaux (par exemple, plus de rotations des épaules) et une différence de vitesse entre les participants.
- $H2_2$: Avec ces changements sur les mouvements locaux du corps des participants, il y aura à la fois moins de collisions, et de plus petits volumes d'interpénétration lorsqu'une collision se produit.

$H3$: Nous nous attendons à quelques effets secondaires dus au rendu haptique, c'est-à-dire que nous attendons des participants qu'ils restent plus conscients et attentifs aux collisions même après avoir désactivé le rendu haptique. Nous nous attendons donc à ce que $H2_1$ et $H2_2$ restent vrais dans le bloc *NoHaptic2*.

$H4$: Le rendu haptique améliorera le sens de la présence et le sens de l'incarnation des participants à la RV, car ils deviendront plus conscients des dimensions de leur corps virtuel dans l'espace par rapport aux personnages virtuels voisins.

8.3 Outils d'analyse

Cette section présente les variables utilisées pour évaluer nos hypothèses.

8.3.1 Trajectoires

Pour étudier $H1$, nous avons comparé les trajectoires des participants à travers la foule virtuelle. À cette fin, nous avons décomposé l'environnement en cellules basées sur une triangulation de Delaunay Chew [1989], dont les sommets étaient les personnages de la foule. Une trajectoire est alors représentée comme une séquence de cellules traversées. Un exemple est présenté dans la figure 8.6, où la trajectoire affichée correspond à la séquence de cellules suivante : $C_{15}C_{18}C_{13}C_{31}C_5C_{30}C_2C_{34}C_4$.

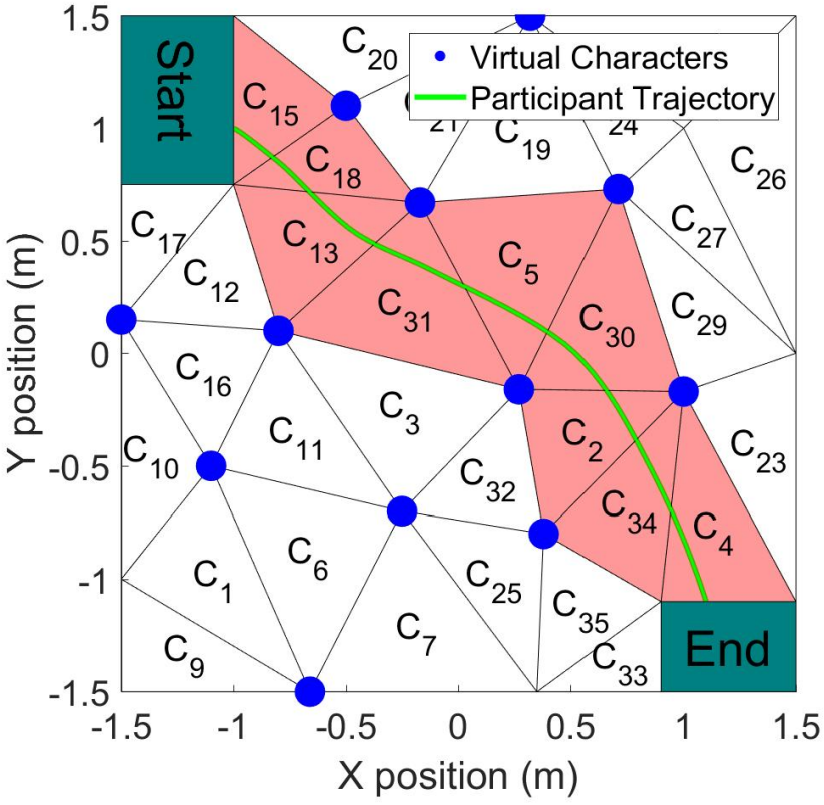


Figure 8.6 – Illustration de la trajectoire d'un participant dans une foule et de la décomposition de l'environnement en cellules à l'aide de la triangulation de Delaunay [Chew, 1989].

Avec cette représentation, la comparaison n'est possible que lorsque la configuration des personnages de la foule est identique, ce qui est une des raisons

pour lesquelles nous avons veillé à répéter les mêmes configurations à travers les 3 blocs étudiés (cf. Section 8.2). En d'autres termes, nous avons d'abord regroupé les trajectoires par configuration de foule, puis comparé l'ensemble des trajectoires effectuées dans la même configuration de foule dans différentes conditions.

La comparaison était basée sur le coefficient de similarité de Dice (DSC) Sørensen [1948]. Le DSC calcule la similarité entre deux ensembles A et B selon

$$DSC(A, B) = \frac{2|A \cap B|}{|A| + |B|}. \tag{8.1}$$

Puisque nos trajectoires sont des ensembles de cellules traversées, deux trajectoires traversant le même ensemble de cellules seront similaires à 100%. La similarité diminuera avec le nombre de cellules différentes traversées par le participant (se produisant dans une trajectoire et non dans l'autre).

8.3.2 Mouvements du corps

Pour naviguer dans des environnements encombrés, comme ceux étudiés dans cette expérience, les participants doivent se faufiler avec leur corps à travers la foule. Cette section présente les données qui seront utilisées pour analyser les mouvements du corps lors de la navigation à travers la foule virtuelle pour étudier $H2_1$.

Rotation des épaules

Tourner les épaules est une stratégie communément observée pour se faufiler à travers des ouvertures étroites [Wilmut and Barnett, 2010], c'est-à-dire, dans notre cas, pour se placer entre deux personnages proches. Pour évaluer l'effet du rendu haptique sur l'émergence de tels comportements, nous avons mesuré l'orientation de l'épaule à certains points critiques du parcours. Ces points critiques correspondent aux points de passage entre les limites des cellules de Delaunay (cf. Section 8.3.1) et les trajectoires du participant.

Plus précisément, nous avons calculé l'angle α_{SA} entre l'axe épaule-épaule du participant et le segment reliant les deux personnages virtuels considérés (figure 8.7). Cet angle fournit des informations sur l'orientation des épaules, et donc du tronc, aux endroits les plus étroits du chemin suivi par les participants lorsque ceux-ci passent entre deux personnages. Plus cet angle est grand, plus les participants sont prudents – en essayant de réduire leur largeur au maximum – lorsqu'ils traversent l'ouverture entre les deux personnages virtuels.

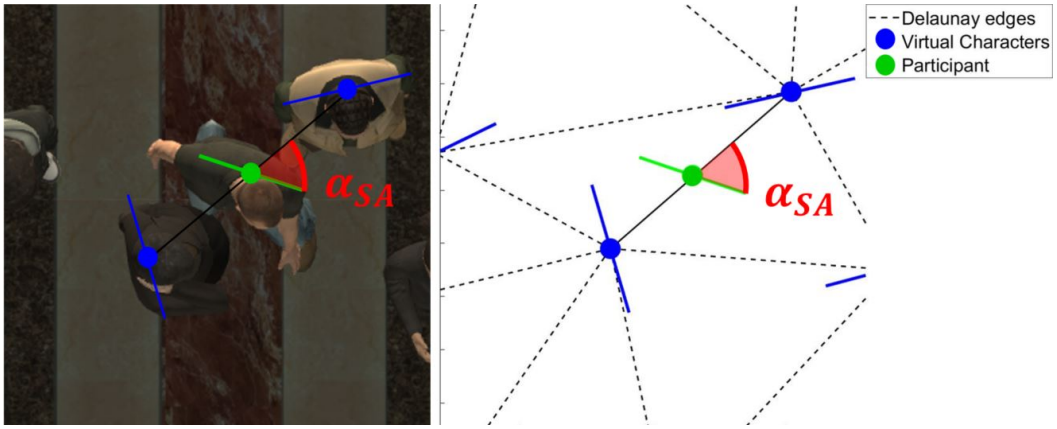


Figure 8.7 – Rotation des épaules. L'angle $\alpha_{SA} \in [0, 90]°$ est défini entre l'axe épaule à épaule des participants et le segment reliant les deux personnages virtuels. A gauche : vue de dessus de la scène. A droite : diagramme avec les triangles de Delaunay, les personnages virtuels et le participant.

Vitesse de marche

Au-delà de l'analyse posturale présentée dans la section précédente, nous nous intéressons également à la vitesse de marche pour analyser si les participants ont effectué la tâche de mouvement différemment selon les conditions. Pour évaluer ce paramètre uniquement pendant la navigation, nous avons supprimé les parties des essais où les participants étaient pour la plupart statiques (par exemple, le temps pendant lequel ils lisaient le tableau). À cette fin, nous avons calculé la distance minimale entre le participant et l'écran, qui correspond au moment où les participants s'arrêtent pour lire l'information. Nous avons ensuite supprimé toutes les images lorsque la position du participant était à moins d'un pas de cette position (choisie comme étant $0,74$ m pour les hommes et $0,67$ m pour les femmes Cho et al. [2004]).

8.3.3 Collisions

Une collision correspond au *contact détecté* entre une partie quelconque du corps virtuel du participant et une partie quelconque du maillage d'un personnage virtuel. Nous identifions une collision par la paire participant-virtuel ainsi que par le temps initial. Cela signifie que nous classifions séparément les collisions avec différents personnages, même si elles se produisent au même moment. Cela signifie également que nous pouvons détecter plusieurs collisions avec le même personnage mais avec des temps initiaux différents. La détection commence au premier contact de l'un des membres du personnage concerné et de la géométrie

a) b) c)

Figure 8.8 – Calcul de volume par itération de volumes élémentaires (voxels) de dimensions décroissantes. (a) En partant de l'AABB autour des géométries sélectionnées, le premier espace de voxel avec 8 voxels (cubes verts) est créé autour des géométries. (b) Lors de l'itération suivante, seuls les voxels qui se croisent sont conservés, puis subdivisés en 8 cubes chacun. (c) Le processus est appliqué de manière itérative jusqu'à atteindre la taille minimale de subdivision, où le volume final d'interpénétration est affiché en violet.

du participant, et elle dure jusqu'à ce qu'il n'y ait plus de contact détecté entre les deux mailles respectives. L'ensemble du schéma de calcul des collisions est résumé dans la figure 8.9. Pour analyser les collisions, nous avons sélectionné deux valeurs principales d'intérêt : le *nombre de collisions* et le *volume maximal d'interpénétration* entre le participant et le personnage virtuel lors d'une collision.

— *Nombre de collisions.* On compte toute collision dont le volume d'interpénétration est supérieur à 10^{-6} m^3 et qui dure plus de 10 ms.

— *Volume maximal d'interpénétration.* Le volume maximal d'interpénétration entre l'avatar d'un participant et un personnage virtuel lors d'une collision est calculé à chaque pas de temps par la *voxélisation* de l'intersection de leurs mailles respectives, selon la procédure suivante. Pour chaque 10 ms, le calcul commence à partir des maillages (modèles 3D) des deux personnages impliqués. Autour de ceux-ci, on construit un AABB (Axis Aligned Bounding Box), qui est ensuite subdivisé itérativement en octant où, à chacune de ces itérations en octant, seuls les voxels en collision sont conservés. On s'arrête lorsque la taille du voxel cible est atteinte. Dans notre analyse, elle a été fixée à un cube de largeur $0,01$ m. Ce processus est illustré dans la figure 8.8. À la fin, nous recueillons tous les volumes calculés à chaque intervalle de temps de 10 ms et nous en extrayons le maximum.

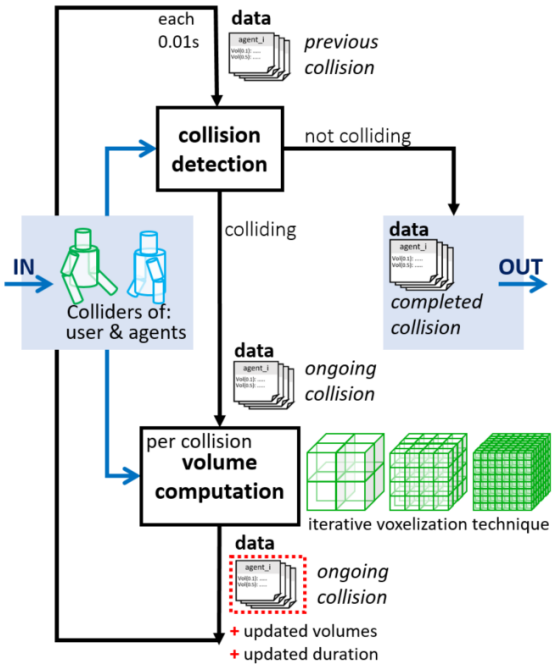


Figure 8.9 – Schéma de boucle d'itération représentant une étape de la détection de collision, dans lequel nous détectons s'il y a une collision (nouvelle ou en cours) et calculons son volume. Nous ajoutons cette information aux données de la collision. Lorsque la collision est terminée, nous enregistrons les données.

8.3.4 Présence et Incarnation

Conformément à $H4$, nous avons recherché des différences dans les sentiments de présence et d'incarnation des utilisateurs, en comparant la perception subjective enregistrée avec et sans rendu haptique. Les participants ont répondu aux deux questionnaires à la fin de chaque bloc (Incarnation puis Présence), en répondant à chaque question sur une échelle de Likert à 7 points.

L'utilisation d'un dispositif haptique est généralement censée augmenter l'immersion de l'utilisateur dans le monde virtuel [Krogmeier et al., 2019], car elle ajoute un nouveau retour sensoriel, même si elle ne conduit pas toujours à une augmentation du réalisme perçu [Slater, 2003]. Pour cette raison, nous avons mesuré la présence en utilisant le questionnaire Slater-Usoh-Steed (SUS) [Usoh et al., 2000] (Tableau 8.5). Chaque utilisateur a répondu à la série de 6 questions, résumées dans le tableau 8.5, à la fin de chaque bloc.

Nous avons procédé de façon similaire à la Présence pour l'Incarnation. Nous nous sommes ainsi concentrés sur la comparaison du sens de l'incarnation entre différents blocs pour étudier l'influence du rendu haptique sur la perception du corps virtuel. Nous avons mesuré l'incarnation sur la base du questionnaire de Roth et Latoschik [Roth and Latoschik, 2019]. Les participants ont répondu aux

questionnaires sur l'incarnation en même temps que ceux sur la Présence.

8.3.5 Analyses statistiques

Notre objectif est de comprendre si et dans quelle mesure les utilisateurs modifient leur comportement dans chaque bloc expérimental.

Pour ce faire, nous avons analysé les différences entre les blocs pour toutes les variables dans la section précédente.

Pour toutes les variables dépendantes, nous avons fixé le niveau de signification à $p = 0,05$. Tout d'abord, un test de Shapiro-Wilk a été effectué pour évaluer si la distribution de nos données suivait une distribution normale. Si la distribution ne suit pas une loi normale, un test de Friedman est alors effectué pour évaluer l'effet de la condition sur ces variables. Des comparaisons post-hoc sont ensuite réalisées à l'aide d'un test de rang signé Wilcoxon avec correction de Bonferroni. En revanche, si la distribution suit une loi normale, une analyse de variance à sens unique (ANOVA) avec mesures répétées est effectuée. Des ajustements de Greenhouse-Geisser aux degrés de liberté sont appliqués si les données violent l'hypothèse de sphéricité. Des tests post-hoc Bonferroni sont utilisés pour analyser tout effet significatif entre les groupes.

8.4 Résultats

Cette section présente les résultats de notre expérience, à commencer par l'étude de $H1$ sur les trajectoires formées par les participants à travers la foule virtuelle. Nous explorons ensuite $H2_1$ et $H2_2$ par rapport à l'analyse des mouvements du corps. Enfin, nous rapportons les résultats sur la métrique des collisions afin d'évaluer $H3$, pour finir avec les réponses aux questionnaires sur la présence et l'incarnation relatifs à $H4$.

8.4.1 Analyse des trajectoires

Le tableau 8.1 montre les résultats de la mesure de similarité de Dice entre toutes les paires de blocs possibles. La similarité varie de $84,7\%$ (blocs *Nohaptic1* vs. *Haptic*) à $88,5\%$ (blocs *Haptic* vs. *NoHaptic2*). Le score est plus élevé pour les blocs *Haptic* vs. *Nohaptic2* $(88,6 \pm 4.1\%)$ et pour les blocs *Nohaptic1* vs. *Nohaptic2* $(85,9 \pm 4,0\%)$.

Comme il est difficile de déterminer à partir de ces données uniquement si le niveau de similarité obtenu est dû à la variété naturelle des comportements

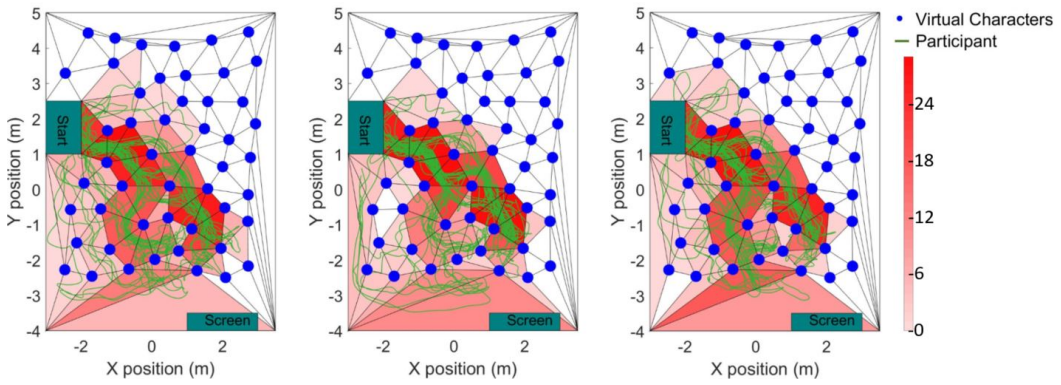


Figure 8.10 – Trajectoires des participants et triangulation de Delaunay pour l'essai T_6 pour les blocs *NoHaptic1* (à gauche), *Haptic* (au milieu) et *NoHaptic2* (à droite). La barre de couleur représente le nombre de fois que les participants ont marché sur un triangle.

humains, ou à la différence des conditions explorées dans chaque bloc, nous proposons de mesurer la similarité entre les chemins appartenant à un même bloc. Ainsi, pour chaque bloc et chaque configuration, nous avons divisé aléatoirement les trajectoires en deux sous-ensembles et calculé le score de similarité de Dice entre eux. Nous avons répété ce processus 30 fois (ce qui change la façon dont les trajectoires sont divisées en 2 sous-ensembles). En effectuant ce processus et en calculant la similarité sur les 3 blocs, nous avons obtenu des mesures de "similarité intra-bloc" de 90. La valeur moyenne obtenue est de $81,2 \pm 3,3$, ce qui peut être comparé avec les scores de "similarité inter-blocs" présentés dans le tableau 8.1.

Nos résultats montrent qu'il n'y a pas de différence statistique entre la mesure de similarité intra-bloc et la mesure de similarité entre blocs *Nohaptic1* et *Haptic* ($p > 0,05$). Il existe cependant une différence significative entre les mesures intra-bloc et les mesures inter-bloc *Haptique* vs. *Nohaptique2* ($p < 0.01$), ainsi qu'entre les mesures intra-bloc et les mesures inter-bloc *Nohaptique1* vs. *Nohaptique2* ($p < 0,05$), où les mesures de similarité intra-bloc sont toujours plus faibles. Étant donné que les mesures de similarité entre les paires de blocs sont soit aussi similaires, soit plus similaires que les similarités intra-bloc, nous pouvons conclure que les participants ont choisi leur chemin à travers la foule de manière similaire, indépendamment de la condition de bloc, qui valide $H1$. Pour illustrer la similarité des chemins de navigation, la figure 8.10 affiche toutes les trajectoires des participants et les triangles utilisés pour calculer les similarités de Dice (pour la configuration spécifique T_6 choisie pour l'exemple).

Table 8.1 – Mesures de similarités (Dice) des trajectoires de chaque participant entre chaque blocs (*NoHaptic1*, *Haptic*, *NoHaptic2*) pour chaque essai.

Essais	Blocs		
	NoHaptic1 vs. *Haptic*	*Haptic* vs. *NoHaptic2*	*NoHaptic1* vs. *NoHaptic2*
T_1	84.0%	88.6%	85.0%
T_2	88.4%	93.8%	88.3%
T_3	78.1%	93.2%	79.4%
T_4	91.9%	88.7%	90.7%
T_5	88.4%	90.2%	85.3%
T_6	82.8%	85.8%	91.0%
T_7	78.8%	81.6%	82.0%
T_8	85.0%	85.9%	85.3%
$\boldsymbol{T_{all}}$	$\mathbf{84.7 \pm 4.8\%}$	$\mathbf{88.6 \pm 4.1\%}$	$\mathbf{85.9 \pm 4.0\%}$

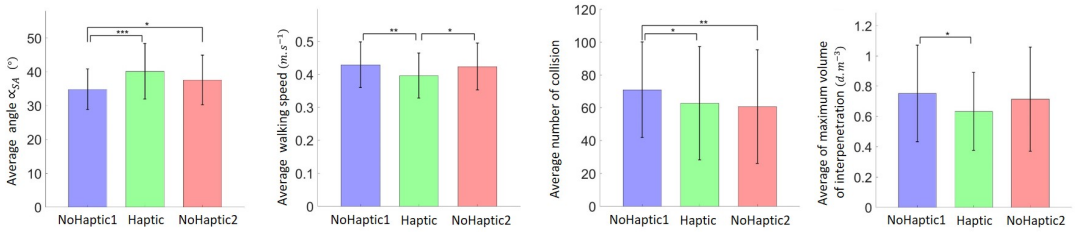
8.4.2 Mouvements du corps

Rotation des épaules

L'amplitude moyenne des rotations de l'épaule α_{SA}, illustrée dans la figure 8.11.a, est significativement différente dans chaque bloc ($F(2, 44) = 13.0, p < 0.001, \eta_p^2 = 0.37$). En particulier, il est significativement plus élevé dans le bloc avec rendu haptique ($40,1 \pm 8,2°$), que dans le premier bloc sans rendu haptique ($34,3 \pm 6,0°$). Nous rappelons qu'un angle α_{SA} plus élevé signifie que les participants ont effectué une rotation plus importante pour se faufiler entre les personnages virtuels, validant ainsi l'hypothèse $H2_1$. De plus, il est aussi significativement plus élevé dans le bloc *NoHaptic2* ($38.7 \pm 3.7°$) que dans le bloc *NoHaptic1*, ce qui suggère que les participants ont continué à tourner davantage leurs épaules même après que le rendu haptique ait été désactivé, ce qui justifie $H3$.

Vitesse de marche

Nous avons trouvé un effet du rendu haptique ($F(1, 56, 34, 2) = 7, 14, p = 0, 005, \eta_p^2 = 0, 245$) sur la vitesse moyenne de marche des participants (Figure 8.11. b), où la vitesse de marche moyenne des participants est en moyenne significativement plus faible dans le bloc *Haptic* ($0, 40 \pm 0, 1 m.s^{-1}$) que dans le bloc *NoHaptic1* ($0.43 \pm 0.1 m.s^{-1}$) et dans le bloc *NoHaptic2* ($0.42 \pm 0.1 m.s^{-1}$). Ce résultat soutient donc également l'hypothèse $H2_1$.



a) b) c) d)

Figure 8.11 – Principales différences significatives entre les trois blocs de l'expérience (*NoHaptic1*, *Haptic* et *NoHaptic2*) : a) amplitude des rotations de l'épaule (α_{SA}), b) vitesse de marche, c) nombre de collisions par essai, d) volume d'interpénétration. Les barres d'erreur représentent l'écart-type de la moyenne.

8.4.3 Collisions

Figures 8.11.c 8.11.d illustrent les résultats concernant les caractéristiques des collisions, c'est-à-dire le nombre de collisions ainsi que le volume d'interpénétration.

Le nombre moyen de collisions par essai a été nettement influencé par le rendu haptique ($F(2,44) = 7,13, p = 0,002, \eta_p^2 = 0,25$). L'analyse post-hoc a montré que le nombre de collisions était plus élevé pendant le bloc *NoHaptic1* ($71 \pm 29,2$) que pendant le bloc *Haptic* ($62,8 \pm 34,6$, $p = 0,018$) et les blocs *NoHaptic2* ($60,7 \pm 34,6$, $p = 0,002$), ce qui montre que les participants ont fait en moyenne plus de collisions au début de l'expérience, avant d'utiliser le rendu haptique.

Le volume moyen d'interpénétration a également été influencé par le bloc ($F(2,44) = 4.35, p = 0.019, \eta_p^2 = 0.16$), où l'analyse post-hoc a montré que ce volume était plus faible ($p = 0.016$) dans le bloc *Haptic* (0.6 ± 0.3 dm^{-3}) que pendant le bloc *NoHaptic1*. (0.8 ± 0.3 dm^{-3}).

Ces résultats valident notre hypothèse $H2_2$, qui affirme que le rendu haptique réduit la gravité des collisions entre les participants et les personnages virtuels. De plus, comme le nombre de collisions est plus élevé pendant le bloc *NoHaptic1* que pendant le bloc *NoHaptic2*, cela permet également de valider $H3$ sur les éventuelles séquelles du rendu haptique.

8.4.4 Présence et Incarnation

Les notes moyennes des participants et toutes les questions relatives à **l'Incarnation** sont indiquées dans les tableaux 8.2, 8.3 et 8.4. Nous n'avons pas trouvé d'effet significatif des blocs pour *Agency* ($p = 0,438$), *Change* ($p = 0,085$) et *Ownership* ($p = 0.753$). En outre, le tableau 8.5 montre les questions et les notes

moyennes des participants pour la **présence**, pour lesquelles nous n'avons pas non plus trouvé d'effet significatif des blocs ($p = 0.222$). Ces résultats ne valident donc pas l'hypothèse $H4$, ce qui suggère que le rendu haptique n'améliore pas le sens de la présence ou le sens de l'incarnation des participants dans la RV.

Table 8.2 – Questionnaire d'*Agencement* : notes moyennes des participants pour les trois blocs.

Questions	**Blocs**		
	NoHaptic1	*Haptic*	*NoHaptic2*
Les mouvements du corps virtuel semblaient être mes mouvements. – J'avais l'impression de contrôler les mouvements du corps virtuel. – J'avais l'impression de provoquer les mouvements du corps virtuel. – Les mouvements du corps virtuel étaient synchronisés avec mes propres mouvements.	6.1 ± 0.9	6.0 ± 0.8	5.9 ± 0.7

8.5 Discussion

L'objectif principal de cette étude était d'évaluer l'effet du rendu haptique des collisions sur le comportement des participants lorsqu'ils naviguent dans un environnement dense. À cette fin, nous avons conçu une expérience dans laquelle les participants devaient atteindre un objectif en marchant physiquement dans une gare virtuelle peuplée d'une foule dense. Les participants ont été équipés de brassards vibrotactiles situés sur leurs bras et ont effectué cette tâche en suivant 3 blocs : *NoHaptic1*, *Haptic* et *NoHaptic2*, pour lesquels le rendu haptique des collisions avec des personnages virtuels n'était d'abord pas présent, puis présent, et enfin absent à nouveau, respectivement pour chaque bloc.

8.5.1 Trajectoires

Dans la section 8.4.1, l'analyse de la mesure de similarité de Dice a montré que le rendu haptique ne changeait pas la façon dont les participants choisissaient leur chemin à travers la foule, comme l'indique l'hypothèse $H1$. Nous avons même

Table 8.3 – Questionnaire de *Changement* : notes moyennes des participants pour les trois blocs.

Questions	**Blocs**		
	NoHaptic1	*Haptic*	*NoHaptic2*
J'avais l'impression que la forme ou l'apparence de mon propre corps avait changé. – J'ai eu l'impression que le poids de mon propre corps avait changé. – J'ai eu l'impression que la taille (hauteur) de mon propre corps avait changé. – J'ai eu l'impression que la largeur de mon propre corps avait changé.	3.6 ± 1.3	3.8 ± 1.5	3.3 ± 1.5

Table 8.4 – Questionnaire de *Propriété* : notes moyennes des participants pour les trois blocs.

Questions	**Blocks**		
	NoHaptic1	*Haptic*	*NoHaptic2*
J'avais l'impression que le corps virtuel était mon corps. – C'était comme si les membres virtuels de mon corps étaient les miens. – Le corps virtuel semblait être comme un corps physique. J'avais l'impression que le corps virtuel m'appartenait.	4.9 ± 1.4	5.1 ± 1.2	5.0 ± 1.2

constaté que les chemins à travers les blocs étaient "plus similaires" qu'à l'intérieur d'un même bloc. Ceci peut provenir de la façon dont nous composons les ensembles dont nous comparons la similarité, où nous supposons que les chemins sont indépendants des participants. En effet, la mesure de la similarité intra-bloc nous a obligé à diviser un ensemble de trajectoires appartenant à la même configuration de bloc et de foule, ce qui a permis de comparer les trajectoires effectuées par différents participants. En revanche, l'analyse inter-blocs a pris en compte des ensembles qui ont été divisés selon les conditions de rendu haptique, comparant ainsi les trajectoires effectuées par le même groupe de 23 participants.

Malgré cette limitation dans notre analyse, nous considérons que les chemins sont similaires d'un bloc à l'autre. On peut décrire le mouvement humain comme une trajectoire résultant d'une boucle perception-action [Warren, 1998]. Selon les tâches, la boucle est multimodale, c'est-à-dire que différents sens sont utilisés

Table 8.5 – Questionnaire de Slater-Usoh-Steed (SUS) [Usoh et al., 2000] et les évaluations moyennes des participants pour les trois blocs.

Questions	**Blocks**		
	NoHaptic1	*Haptic*	*NoHaptic2*
J'avais l'impression d'être là, dans la gare. Il y a eu des moments pendant l'expérience où la gare était la réalité pour moi. – La gare me semble être plutôt... (Des images que j'ai vu $<->$ Un endroit que j'ai visité) – J'avais un sens plus fort d'... (Être ailleurs $<->$ Être dans une gare) – Je pense à la gare comme à un endroit similaire aux autres endroits où j'ai été aujourd'hui – Pendant l'expérience, j'ai souvent pensé que J'étais vraiment debout dans la gare.	5.2 ± 0.9	5.2 ± 1.2	5.0 ± 1.1

pour contrôler le mouvement. Cependant, dans le contexte de la marche, plusieurs études [Patla, 1997, Warren, 1998] ont montré que la vision est l'entrée perceptuelle la plus utilisée pour naviguer vers le but. Ces affirmations sont valables dans notre cas, malgré une différence majeure avec les travaux précédents concernant la densité plus élevée des obstacles. Néanmoins, en supposant que le retour haptique puisse affecter le choix du chemin emprunté, il aurait été probable que certains participants fassent marche arrière après une collision, ce qui n'a pas été observé.

8.5.2 Comportement d'évitement

Dans cette expérience, nous avons démontré que le rendu haptique avait un effet sur les rotations des épaules, ce qui valide l'hypothèse $H2_1$. En particulier, les participants ont davantage tourné leurs épaules en traversant les espaces entre les personnages virtuels pendant le bloc *Haptic* que pendant le bloc *NoHaptic1*. Ce résultat est cohérent avec les observations de Mestre et al. [2016] avec des participants passant par une porte virtuelle à moitié ouverte avec ou sans rendu haptique. Plus généralement, rappelons que le tronc humain est le plus souvent plus large le long de l'axe transversal que le long de l'axe antéro-postérieur. Ainsi, plus les participants tournent leurs épaules, plus le volume balayé par les mouvements de leur corps est faible. Nos résultats suggèrent donc que les participants auraient pu essayer de minimiser le risque de collision avec des personnages vir-

tuels davantage dans la condition où ils ont expérimenté le rendu haptique que dans le premier bloc de l'expérience. La vitesse plus lente observée dans le bloc *Haptic* révèle également que les participants se déplaçaient plus prudemment. Le fait d'être plus prudent a effectivement entraîné moins de collisions comme prévu dans l'hypothèse $H2_2$. Les résultats présentés montrent que le nombre moyen de collisions ainsi que le volume moyen d'interpénétration étaient significativement plus faibles dans le bloc *Haptic* que dans le bloc *NoHaptic1*. En outre, cette observation est conforme aux études précédentes [Louison et al., 2018] où le retour haptique a réduit le nombre de collisions avec un objet statique.

8.5.3 Effet post-activation du rendu haptique

Bien qu'il y ait eu moins de collisions et plus de rotations d'épaule observées dans le bloc *Haptic* par rapport au bloc *NoHaptic1*, il n'y avait aucune différence entre les blocs *Haptic* et *NoHaptic2*. Cela confirme l'hypothèse $H3$ sur les éventuels effets secondaires d'apprentissage du rendu haptique, comme une certaine prudence conservée entre les blocs. Cependant, un tel effet secondaire n'a pas influencé de la même manière toutes les mesures, comme la vitesse de marche qui a encore augmenté dans le bloc *NoHaptic2*. Une explication possible pourrait être la familiarité des participants avec l'environnement virtuel ainsi qu'avec leur corps virtuel qui a pu augmenter d'un bloc à l'autre. Par exemple, les participants peuvent avoir été capables de se déplacer plus rapidement et d'éviter de meilleures collisions avec les personnages virtuels du dernier bloc (*NoHaptic2*). Un autre point à souligner est que les participants, au début du bloc *Haptic*, ne savaient pas que les contacts allaient maintenant déclencher une sensation haptique vibrotactile. Pour cette raison, on peut s'attendre à une courte phase d'apprentissage au début du bloc, où les participants apprennent à gérer les nouvelles collisions haptiques. Compte tenu de ce point, on peut s'attendre à ce que l'effet de fournir des sensations haptiques de collisions soit encore plus fort que celui enregistré. Toutefois, pour parvenir à une conclusion plus définitive sur le rôle de l'effet haptique, il faudrait ajouter un groupe témoin sans rendu haptique tout au long des trois blocs de l'expérience, ce qui pourrait être étudié dans des travaux futurs.

Ces résultats peuvent également ouvrir des perspectives concernant la conception de nouvelles expériences, y compris les tâches d'amorçage haptique. Dans une étude récente, Krum et al. [2018] ont montré que l'amorçage haptique de la collision n'avait aucun effet sur les proxémies des participants et plus précisément sur les distances entre eux et les personnages virtuels. Il est important de noter que

dans cette article la tâche était différente : elle comprenait une interaction avec un personnage virtuel et il n'y avait aucun risque de collision puisque le personnage virtuel ne s'approchait jamais très près du participant. Il serait donc intéressant de réévaluer cette influence lorsque l'espace intime est franchi par un personnage virtuel.

8.5.4 Incarnation et présence

Contrairement à notre hypothèse $H4$, nous n'avons pas trouvé de changement significatif en termes de perception des sens d'incarnation et de présence de l'utilisateur lors de l'expérience du retour haptique. Ce résultat est assez surprenant, car nous avons trouvé des effets significatifs dans d'autres mesures, ce qui suggère que les participants ont agi différemment lorsqu'ils ont eu des sensations haptiques de contact. Ce résultat pourrait s'expliquer par le fait que les utilisateurs ont déjà enregistré des niveaux élevés d'incarnation et de présence sans ressentir de retour haptique dans la première condition (*NoHaptic1*), ce qui laisse peu de place à l'amélioration dans la condition *Haptic*. Une autre possibilité est que le retour vibrotactile ne soit pas adapté pour rendre des collisions dans des foules, bien qu'il existe plusieurs exemples de ce type de retour utilisé pour rendre des événements similaires [Bimbo et al., 2017, Devigne et al., 2020]. Enfin, une dernière explication pourrait être l'emplacement et le nombre de nos appareils haptiques. L'utilisation d'un plus grand nombre de brassards répartis dans le corps pourrait permettre de mieux restituer les sensations de contact de la cible.

8.5.5 Limitations

Notre étude avait quelques limites. Comme expliqué ci-dessus, nous avons utilisé un nombre limité de dispositifs de rendu haptique situés uniquement sur les bras des participants. Il est tout à fait possible que l'utilisation d'un plus grand nombre de dispositifs, y compris positionnés sur les jambes et les hanches, aurait entraîné des effets plus importants. Cependant, notre solution a tout de même révélé des effets significatifs, et la question de la nature, du nombre et de l'emplacement des dispositifs haptiques nécessiterait probablement une étude entièrement dédiée. Une autre question connexe est celle de *qualité* des sensations haptiques fournies. Nos dispositifs sont très faciles à porter et à transporter, mais ne peuvent fournir que des sensations haptiques vibrotactiles, et leur fréquence et leur amplitude sont indissociable. D'autres solutions comme l'usage d'exosquelettes de bras ou de corps entier pourraient fournir des sensations de force plus localisées. Toutefois, ces dispositifs sont beaucoup plus encombrants et coûteux que ceux

utilisés pour ce travail, ce qui limite considérablement leur applicabilité et leur disponibilité.

Une deuxième limitation concerne le comportement des personnages virtuels présents dans la foule. En effet, ils ne réagissent pas aux collisions, comme l'ont remarqué certains participants dans leurs commentaires. Il serait donc nécessaire de disposer d'une technique d'animation capable de réagir aux collisions comme, par exemple, le personnage virtuel faisant un pas dans la direction opposée à la collision. Nous pourrions également déclencher des réactions verbales pour exprimer que les personnages virtuels sont gênés par les collisions. L'ajout de tels comportements virtuels combinés à un retour haptique pourrait améliorer l'immersion et le sentiment de présence des participants.

Enfin, un dernier point concerne les nombreux dispositifs (brassards, MSI VR one, HMD, X-Sens, etc.) que les participants doivent porter pendant une durée significative. Le port de ces équipements peut avoir un effet sur les mouvements des participants ainsi que sur leur confort. Dans notre cas, l'expérience était relativement courte et ne durait que 15 à 20 minutes. Cependant, des durées d'immersion plus longues pourraient nécessiter l'utilisation de solutions HMD sans fil à la place, même si cela signifie aujourd'hui une diminution du champ de vision.

8.6 Conclusion

Dans ce chapitre, nous avons conçu une expérience pour évaluer les effets, ainsi que les effets secondaires, du rendu haptique sur une tâche de mouvement dans un environnement très fréquenté. Les participants ont effectué une tâche de navigation orientée vers un objectif à travers une foule virtuelle dense. Des dispositifs haptiques portatifs leur fournissaient un retour vibrotactile à chaque fois qu'une collision avec leurs bras se produisait. Les résultats ont montré que fournir un retour d'information haptique a un impact sur la façon dont les participants se déplacent dans la foule virtuelle. Lors de l'expérimentation, ils étaient en effet plus prudents quant aux collisions qu'ils provoquaient avec les personnages virtuels, mais ils ne changeaient pas leurs trajectoires globales. Nous avons également démontré la présence d'un effet secondaire du retour haptique, puisque les changements dans leurs mouvements sont restés observables après la désactivation du retour haptique. Enfin, de manière assez surprenante, nous n'avons pas remarqué d'impact du rendu haptique sur la perception de la présence et de l'incarnation. Ces résultats montrent que l'information visuelle est probablement le principal sens utilisé pour la navigation dans une foule dense. Cependant, une combinaison de retour visuel et haptique améliore le réalisme général de l'expérience, car les

participants ont un comportement plus réaliste : ils sont plus prudents lorsqu'il s'agit de ne pas toucher les personnages virtuels. C'est pourquoi nous suggérons d'utiliser le rendu haptique pour étudier le comportement humain et les interactions de locomotion qui peuvent conduire à des contacts.

Chapitre

9 Simulateur de fauteuil

Sommaire

Ce chapitre présente une méthode permettant d'améliorer l'immersion dans un contexte précis : celui de la conduite de fauteuil roulant électrique à l'aide d'un simulateur dédié. En effet, la RV peut donner au participant des sensations de malaise proche du mal de mer, qu'on appelle le mal des simulateur (*"Cyber sickness"* en anglais). Le mal des simulateur apparaît lorsque la perception visuelle donne des images conflictuelles par rapport à la proprioception du participant (i.e. la perception de son propre corps) et par rapport au système vestibulaire. Ce que l'on voit diffère de ce que l'on ressent. L'effet est le même lorsque l'utilisateur se retrouve virtuellement embarqué dans un véhicule. Notre approche décrit ici

la conception d'une plateforme mécanique restituant des sensations similaires à celles d'un fauteuil électrique pour l'immersion en RV d'utilisateurs à des fins d'entraînement et de pédagogie. Cette plateforme permet de synchroniser le retour visuel et la perception du système vestibulaire pour une meilleure immersion. [1]

9.1 Introduction

Les personnes en situation de handicap peuvent avoir du mal à conduire un fauteuil roulant électrique en raison d'une altération de leurs capacités de perception. Il a été démontré, notamment par Massengale et al. [2005], que de bonnes capacités visuo-spatiales et cognitives sont nécessaires pour conduire un fauteuil roulant électrique en toute sécurité. Si la formation et la pratique répétée par le biais de séances d'ergothérapie peuvent être suffisantes pour certaines personnes pour maîtriser leur fauteuil roulant, cette formation conventionnelle peut être insuffisante pour d'autres. Certaines personnes peuvent avoir en effet besoin d'une formation dans des situations spécifiques telles que la conduite dans un environnement urbain ou dans des endroits bondés. Cependant, ces formations posent des problèmes de sécurité et sont difficiles à mettre en œuvre. En conséquence, les thérapeutes peuvent décider de ne pas prescrire de fauteuil roulant électrique pour des raisons de sécurité (par exemple, si les performances de conduite sont supérieures à un niveau de risque subjectif) Mortenson et al. [2013]. Dans ce contexte, la simulation de conduite en fauteuil roulant peut constituer un outil de formation supplémentaire. En effet, les effets bénéfiques de la formation en simulation sont reconnus depuis longtemps dans des applications telles que l'aviation avec des simulateurs de vol pour former les pilotes Hays et al. [1992]. En outre, la simulation permet de naviguer en toute sécurité en fauteuil roulant dans des scénarios contrôlés et reproductibles. Abellard et al. [2010].

On a vu dans les chapitres précédents l'intérêt qu'offre la RV pour l'immersion de personnes dans un environnement contrôlé en toute sécurité. Cette technologie est donc parfaite pour la formation et la réadaptaton. [Inman et al., 1997, Howard, 2017, Dascal et al., 2017]. Une étude récente sur les simulateurs de fauteuils roulants basés sur la RV menée par Arlati et al. [2019] a identifié des projets de

1. fait l'objet d'une publication pour la conférence ICORR 2019 : *Guillaume Vailland, Fabien Grzeskowiak, Louise Devigne, Yoren Gaffary, Bastien Fraudet, Emilie Leblong, Florian Nouviale, François Pasteau, Ronan Le Breton, Sylvain Guegan, Valérie Gouranton, Bruno Arnaldi, Marie Babel, "User-centered design of a multisensory power wheelchair simulator : towards training and rehabilitation applications"*

Figure 9.1 – Le simulateur de fauteuil roulant électrique PWS (pour *"Power Wheelchair Simulator"* : plateforme mécanique et scène 3D transmise à travers un HMD pour une expérience immersive dans la Ville de Rennes.

simulateurs de fauteuils roulants, s'intéressant pour la plupart à la simulation de fauteuils roulants électriques. Cet examen a particulièrement mis en évidence les différences entre les simulateurs proposés :

— ils diffèrent dans leur conception et leur configuration ;

— ils diffèrent en termes de méthodologie d'évaluation et de programmes de formation, car la plupart des simulateurs ont été initialement conçus pour des applications de formation à la conduite.

La plupart des simulateurs basés sur la RV utilisent des affichages visuels en 3D tels que des dispositifs montés sur la tête (HMD) ou des salles immersives pour augmenter le sentiment de présence (défini comme le sentiment de l'utilisateur d'"être là" dans la simulation [Slater and Wilbur, 1997]). Certains des simulateurs de conduite en fauteuil roulant basés sur la RV tentent d'atteindre un niveau de *présence* plus élevé en fournissant un retour haptique et/ou des indices de mouvement en plus du retour visuel, comme nous le faisons dans le chapitre 8. Le principal inconvénient de ces simulateurs est qu'ils ne sont pas conformes aux exi-

gences et aux besoins cliniques et aux attentes des utilisateurs. En effet, l'affichage visuel doit être adapté aux besoins de l'utilisateur, mais ces simulateurs reposent souvent sur une technologie d'affichage visuel spécifique, avec laquelle il est difficile de satisfaire l'utilisateur. En outre, ils ne permettent pas un transfert facile des usagers sur le système ni ne comportent des sièges adaptés aux handicaps car ils n'intègrent pas les composants standard des fauteuils roulants.

C'est dans ce contexte qu'un consortium transverse à l'INSA de Rennes, dans le cadre du projet Interreg France (Manche) Angleterre ADAPT, travaille et propose un simulateur multisensoriel de fauteuil roulant électrique (PWS pour "Powered Wheelchair Simulator") conçu en étroite collaboration avec des ergothérapeutes et des utilisateurs de fauteuils roulants. L'INSA de Rennes nous a alors proposé de collaborer pour la définition et la mise en place de cet outil.

Le simulateur ainsi conçu comprend une plate-forme fournissant des retours visuels, auditifs et haptiques, ainsi que des signaux de mouvement. La conception modulaire proposée permet de s'interfacer facilement avec n'importe quel affichage visuel (*e.g.* HMD, salles immersives CAVE voir figure 2.7, écran unique). Elle permet également d'intégrer facilement des modules complémentaires et d'adapter la configuration aux besoins individuels des utilisateurs. Le PWS proposé vise à fournir une expérience réaliste de conduite en fauteuil roulant pour la formation et le transfert de compétences dans des situations réelles.

Ce projet est mené par notre équipe pluridisciplinaire à Rennes qui regroupe plusieurs centres de recherche comme l'Inria, l'IRISA, l'IETR, le LGCGM et le centre de médecine physique et de réadaptation du Pôle Saint Hélier. Grâce à cette pluralité de compétences, tous nos choix de conception ont bénéficié de l'expérience d'experts qualifiés. En particulier, des informaticiens et des mécaniciens ont travaillé en étroite collaboration avec le personnel clinique pour mieux faire correspondre les besoins et les contraintes des thérapeutes et des utilisateurs de fauteuils roulants.

9.2 Pré-requis

Nous proposons une conception multisensorielle de PWS fournissant des retours visuels, auditifs, haptiques adaptés et des indications de mouvement cohérents avec la navigation en fauteuil réel. La conception du simulateur est soumise à des contraintes déduites des besoins ciblés de l'utilisateur mais aussi de la technologie RV elle-même.

9.2.1 Les besoins des utilisateurs ciblés

La conception du PWS proposé doit tenir compte de l'expertise de l'utilisateur final. Les besoins de l'utilisateur final ont été formulés grâce aux thérapeutes et aux patients du centre de médecine physique et de réadaptation du Pôle Saint Hélier. Cette sous-section présente donc les contraintes basées sur les recommandations des ergothérapeutes, kinésithérapeutes et médecins, mais aussi et surtout des utilisateurs de fauteuils roulants.

Facteur d'assise et de forme

Compte tenu de la complexité de l'état postural associé à certaines pathologies, un large éventail d'ajustements des sièges est nécessaire. Pour éviter d'avoir une assise mal adaptée (et donc inutilisable pour des personnes en situation de handicap), les sièges doivent être réglables et permettant l'inclinaison du dossier. En outre, un large éventail de supports de positionnement doit être disponibles sur le siège en fonction des besoins spécifiques de l'utilisateur, du type de handicap et de la posture. De plus, le siège doit comporter des accessoires tels que des supports latéraux pour le tronc, des guides pour les hanches, un appui-tête, des supports latéraux pour les cuisses, des accoudoirs et un bloc d'abduction. Une ceinture devrait également être présente pour assurer le positionnement du participant pendant la simulation. Une bonne assise nécessite également des repose-pieds. La largeur et la hauteur du siège de la plate-forme doivent également être conformes aux normes des fauteuils roulants. La largeur de la plate-forme ne doit pas être sensiblement plus grande que la largeur du siège afin qu'elle puisse être perçue par ses utilisateurs à une plate-forme de type fauteuil roulant électrique.

Contrôleur

La pluralité des handicaps moteurs implique également la nécessité d'une grande variété de commandes : depuis les commandes proportionnelles (par exemple, des joysticks, ou des commandes au menton) jusqu'aux commandes tout-ou-rien (par exemple, contacteurs de tête, contacteurs "aspire-souffle"). De plus, diverses adaptations des entrées devraient être disponibles pour les personnes ayant une faible force des membres supérieurs ou des difficultés de préhension (par exemple, poignée en T, poignée en U, balle en mousse).

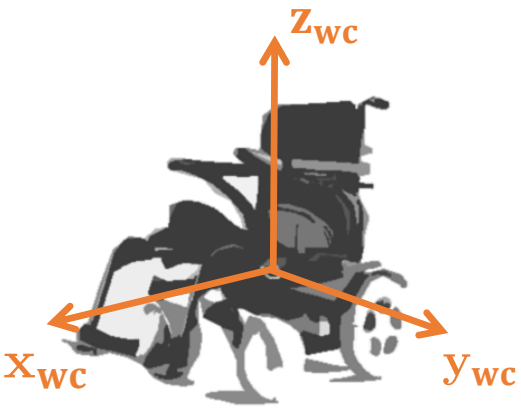


Figure 9.2 – Axes de mouvement d'un fauteuil roulant électrique (wc).

	Direction $\mathbf{x_{wc}}$	Direction $\mathbf{y_{wc}}$	Direction $\mathbf{z_{wc}}$
Accélérations en translation	$2\,\mathrm{m.s^{-2}}$	n.a.	$15\,\mathrm{m.s^{-2}}$
Accélérations en rotation	$2\,\mathrm{rad.s^{-2}}$	$10\,\mathrm{rad.s^{-2}}$	$5\,\mathrm{rad.s^{-2}}$

Table 9.1 – Valeurs cinématiques maximales d'un fauteuil roulant électrique. Le PWS doit pouvoir reproduire ces propriétés.

Du fauteuil roulant électrique à la cinématique

L'objectif du simulateur est donc de fournir une expérience de conduite virtuelle similaire à celle obtenue avec un fauteuil roulant électrique réel. Le comportement du simulateur doit donc reproduire avec précision le comportement dynamique d'un fauteuil roulant électrique. Nous avons donc identifié les principaux paramètres cinématiques du mouvement du fauteuil roulant électrique, c'est-à-dire les valeurs des accélérations ressenties par un utilisateur (voir la figure 9.2) et le Tableau 9.1).

Ces valeurs ont été obtenues expérimentalement grâce à un ensemble de scénarios de conduite représentant des cas classiques d'utilisation en intérieur et en extérieur. Ces scénarios comprenaient notamment la montée de trottoir et de rampe, la navigation sur des surfaces rugueuses, le passage de porte...

Les valeurs recueillies correspondent à une navigation naturelle et sûre d'un utilisateur régulier de fauteuil roulant, et servent de référence. Les dynamiques survenant lors d'événements accidentels, tels que les collisions et les bosses, ne doivent pas être entièrement reproduites par le retour d'information sur le mouvement afin d'atténuer les risques pour les utilisateurs. Ces situations particulières doivent être traitées par un retour d'information haptique et auditif adapté.

Réalité virtuelle

La technologie d'affichage visuel dépend des besoins spécifiques de chaque utilisateur. Par conséquent, un simulateur ne doit pas dépendre d'un dispositif ou d'une technologie d'affichage spécifique et doit être compatible avec divers affichages visuels immersifs tels que les HMD et les salles immersives.

9.2.2 Contraintes de conception liées à la réalité virtuelle

Comme le PWS proposé repose sur la technologie de la RV, les contraintes liées à la RV telles que le sentiment de présence et le mal des simulateur doivent également être prises en compte pour la conception du PWS.

Sentiment de présence (SoP)

Le SoP est un phénomène subjectif qui a été défini par Slater and Wilbur [1997] comme le sentiment de l'utilisateur d'"être là" dans la simulation. Il dépend de divers facteurs tels que l'immersion, les interactions sociales et le comportement des composants de la scène 3D (par exemple des objets, des avatars). Plus le SoP est important, plus le comportement de l'utilisateur sera réaliste. À cet égard, le SoP est un facteur critique que notre PWS doit maximiser.

Mal des simulateurs

L'immersion dans un environnement virtuel peut induire des malaises souvent caractérisés par le mal des transports (*e.g.* nausées, maux de tête, vertiges) [Rebenitsch and Owen, 2016]. Le mal des simulateurs résulte de conflits de perception entre les systèmes visuel, vestibulaire et proprioceptif. En effet, lorsqu'ils se déplacent dans l'environnement virtuel mais qu'ils ne reçoivent pas de retour physique ou qu'ils sont mal configurés, les utilisateurs obtiennent des informations contradictoires potentiellement désagréables. Le mal des simulateurs n'affecte pas malgré tout tout le monde de la même manière. En outre, les utilisateurs ciblés pouvant présenter des troubles visuels, visuo-spatiaux ou cognitifs, ils sont susceptibles d'être plus sensibles à ces symptômes. Ces symptômes peuvent alors altérer considérablement le confort de simulation et peuvent donc avoir un impact négatif sur l'efficacité de la formation sur simulateur. Il est donc essentiel de concevoir un système minimisant le mal des simulateurs en réduisant les décalages entre les informations visuelles, vestibulaires et proprioceptives.

9.3 Conception de la plateforme mécanique de simulation

Le consortium ADAPT de l'INSA a présenté dans des travaux précédents un premier concept de simulateur de fauteuil roulant [Devigne et al., 2017]. Ce simulateur consistait en un fauteuil roulant électrique standard fixé sur la salle d'immersion (CAVE) *Immersia* à Rennes. Ce premier simulateur répondait aux contraintes liées à l'évaluation clinique présentées dans la section 9.2.1 car il utilisait directement un fauteuil roulant électrique standard. Dans ce cadre, l'utilisateur était assis sur un fauteuil roulant demeurant immobile et immergé dans l'environnement virtuel grâce à un rendu de 3D. Bien que ce premier simulateur ait généré un assez bon sentiment de présence chez les utilisateurs, la plupart des personnes qui l'ont testé ont ressenti le mal des simulateurs.

Sur la base de cette première version du simulateur, nous proposons ici une nouvelle conception de simulateur qui vise à répondre aux contraintes liées à la RV en maximisant le sentiment de présence tout en s'attaquant au problème du mal des simulateurs.

Vers un simulateur multisensoriel

Alors que la plupart des simulateurs de RV existants n'utilisent que des techniques de rendu visuel 3D pour immerger l'utilisateur, l'immersion n'est pas limitée au seul sens visuel.

Nous proposons ici un simulateur multisensoriel innovant qui intègre une plateforme de mouvement visant à reproduire les indices de mouvement du fauteuil roulant et les interactions physiques avec l'environnement virtuel, ainsi que les modalités de retours visuels et auditifs.

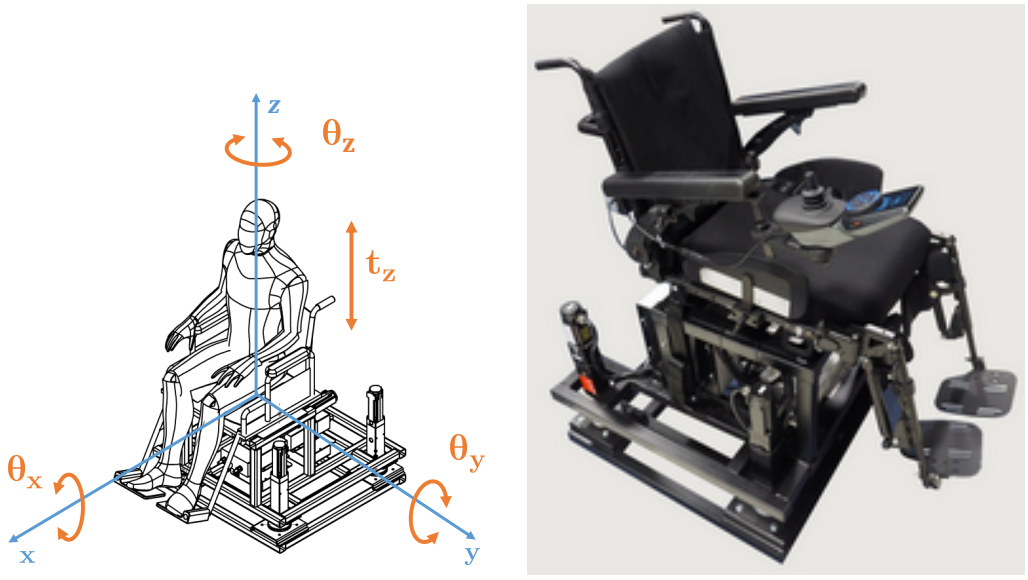
9.3.1 Plateforme mécanique

Nous avons conçu une plateforme mécanique générant des sensations de mouvement afin de simuler des accélérations, ainsi qu'un retour haptique afin de reproduire des interactions avec l'environnement virtuel telles que des bosses, des vibrations et des collisions.

Parmi les simulateurs de fauteuils roulants avec plateforme de mouvement existants, différents types de simulateurs peuvent être identifiés, en fonction du nombre de degrés de liberté (DoF) utilisés. En majorité, les simulateurs utilisent classiquement 3 degrés de liberté ou 6 degrés de liberté (plateforme Stewart). Si

les simulateurs à 6 DoF atteignent des performances en hautes fréquences, les simulateurs à 3 DoF se sont avérés suffisants pour les applications de formation en termes de rendu d'accélération. Pour autant, les simulateurs 6 DoF demeurent encombrants (donc non compatibles avec notre besoin), onéreux, et sont peu ergonomiques.

Nous proposons donc d'ajouter un plateau tournant sur un simulateur à 3 DoF, pour aboutir à un simulateur innovant à 4 DoF, capable d'atteindre les exigences cinématiques énoncées dans la section 9.2.1. La figure 9.3 illustre la solution proposée, basée sur un mécanisme parallèle utilisant 4 actionneurs linéaires (3 DoF) et un plateau tournant sur le dessus (4 DoF). Cette conception est le fruit d'un compromis entre l'amplitude de mouvement souhaitée et la compacité. La cinématique de la plate-forme a été conçue en fonction du comportement réel d'un fauteuil roulant électrique [Baudry et al., 2018].



a) b)

Figure 9.3 – a) Illustration des 4 degrés de liberté du simulateur (en orange) par rapport au châssis fixe (en bleu). b) PWS : Plateforme équippée d'un joystick de fauteuil roulant électrique standard

Un siège de fauteuil roulant *SALSA M2* de *Sunrise Medical* est ensuite fixé sur le plateau tournant afin que l'environnement de l'utilisateur ne diffère pas de celui d'un fauteuil roulant électrique commercial, comme le montre la figure 9.4) : l'expérience de simulation est donc conforme à l'expérience de la vie quotidienne.

L'empâtement du PWS est similaire à celui d'un fauteuil roulant électrique, et la plateforme est légère afin d'être compatible avec les salles immersives. Par exemple, la plateforme doit être plus légère que $100kg/m^2$ pour répondre aux spé-

cifications de *Immersia*. Ce choix facilitera également l'utilisation et le transport. La plate-forme a été fabriquée par CL Corporation [2] à partir des spécifications fournies.

a) b)

Figure 9.4 – PWS installé à *Immersia* : a) scène 3D du centre du Pôle Saint Hélier à Rennes. b) scène 3D de Rennes (fournie par *Rennes Metropole*).

9.3.2 Retour visuel

Le retour visuel permet de percevoir les mouvements du fauteuil roulant pour chacun des 6 DoF correspondant aux sensations de mouvement suggérées par la plateforme de mouvement.

Un fauteuil roulant virtuel est inclus dans les scènes 3D et se trouve à proximité de la plateforme de mouvement. Par exemple, la représentation 3D du siège du fauteuil roulant dans l'environnement virtuel correspond à la position exacte du siège de la plateforme de mouvement.

En outre, la complexité de l'environnement virtuel est adaptée aux déficiences des utilisateurs et aux programmes de formation. Par exemple, comme certains utilisateurs peuvent ne pas être capables de traiter de multiples stimuli visuels, les scènes 3D proposées doivent être simplifiées.

9.3.3 Retour auditif

Un retour auditif est fourni à l'utilisateur afin d'augmenter l'expérience d'immersion et le sentiment de présence. Cela comprend non seulement le son virtuel

2. http://www.clcorporation.com/

rendu sur les haut-parleurs pour les sons ambiants, mais aussi des effets sonores spécifiques provenant directement de la plate-forme elle-même. En particulier, le son produit par les freins du fauteuil roulant est un repère auditif important pour les utilisateurs : ce son est le premier retour d'information perçu par les utilisateurs lorsqu'un fauteuil roulant commence à bouger.

9.3.4 Architecture logicielle

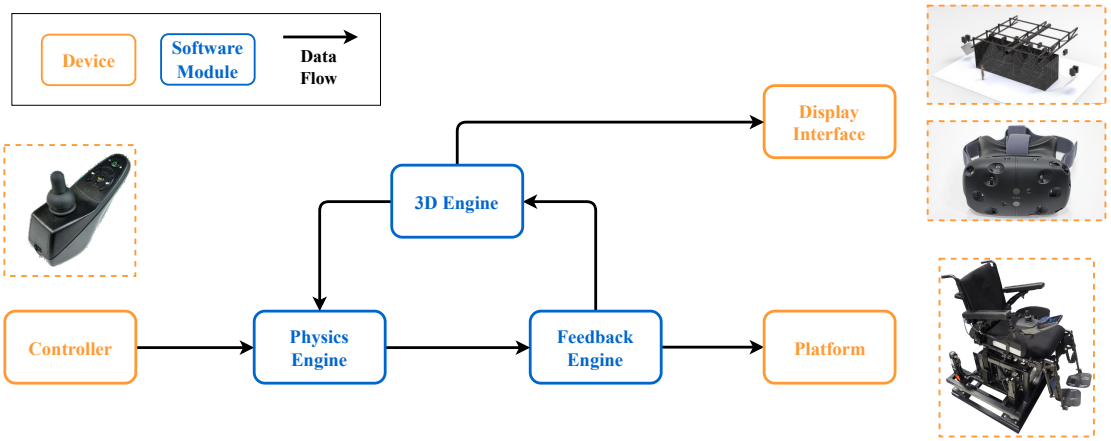


Figure 9.5 – Schéma de l'architecture logicielle du PWS montrant les différents dispositifs utilisés

La conception de PWS repose sur une architecture très proche de celle présentée au chapitre 3 et est composée des modules suivants :

— le *moteur 3D* (ou "3D engine") gère l'environnement virtuel et son affichage ;

— le *moteur physique* (ou "physics engine") met à jour la position, la vitesse et l'accélération du fauteuil roulant virtuel dans l'environnement virtuel. Ces valeurs sont calculées à partir des données du contrôleur d'entrée, des valeurs de position, de vitesse et d'accélération précédentes envoyées par le *moteur 3D*. Il intègre également les lois de la physique (*e.g.* gravité) ;

— le *moteur de retour haptique* (ou "feedback engine") calcule ensuite le mouvement à appliquer. Le mouvement est divisé en un mouvement mécanique de la plate-forme et en un retour visuel vers le *moteur 3D*.

Ces modules sont indépendants et dédiés à des tâches spécifiques distinctes. Cet arrangement modulaire permet à PWS d'être un simulateur polyvalent et adaptable. Chaque module peut être développé ou remplacé simultanément et indépendamment. L'architecture présentée ici ne décrit que les modules nécessaires. Plusieurs étapes peuvent être ajoutées sans incidence sur le processus global.

9.4 Discussion

Cette première mise en œuvre de simulateur est opérationnelle. En particulier, le PWS a été testé avec succès avec différents dispositifs d'affichage (HMD, grandes salles immersives CAVE, écrans simples), des scènes 3D (intérieur/extérieur, basse/haute résolution graphique) et des dispositifs d'entrée (joystick standard, contacteurs de tête).

La plateforme mécanique ainsi que l'architecture logicielle du simulateur ont été conçus pour être polyvalents et adaptables non seulement aux besoins de chaque utilisateur, mais aussi à l'environnement expérimental du centre de réadaptation et au matériel disponible.

La configuration proposée répond donc aux contraintes centrées sur l'utilisateur précédemment énoncées. La conformité du système avec les programmes de formation et de réadaptation à la conduite de fauteuils roulants doit maintenant être évaluée par une évaluation clinique auprès des utilisateurs de fauteuils roulants. En particulier, l'impact des différentes modalités de rétroaction fournies par le simulateur sur le sentiment de présence et le mal des simulateurs doit être évalué. Un premier essai clinique a été réalisé en janvier 2020. 32 utilisateurs réguliers de fauteuil roulant électrique ont testé le simulateur et ont pu confronter leur expérience virtuelle avec le réel. Pour ce faire, un circuit a été conçu pour une navigation en condition "réelle", et a été reproduit en virtuel, de façon à observer le comportement des usagers dans les deux modalités. L'expérience a été jugée très positive par les volontaires, et les performances de conduite sont identiques en réel et en virtuel.

Il apparaît toutefois que la navigation dans un environnement vide d'humains reste perturbant pour les participants. Il convient donc de réfléchir à interconnecter cet outil avec le simulateur de foule présenté au chapitre 5.

9.5 Conclusion

Dans ce chapitre, nous avons présenté une conception centrée sur l'utilisateur d'un simulateur de fauteuil roulant électrique multisensoriel. Ce simulateur s'appuie sur la RV pour immerger l'utilisateur dans une expérience de conduite simulée en fauteuil roulant. L'originalité du simulateur repose sur la combinaison de repères visuels, auditifs et de mouvement fournis par une structure mécanique innovante à 4 degrés de liberté. La conception proposée du simulateur est polyvalente et adaptable à la diversité des déficiences et des besoins. En effet, il est directement compatible avec tout type de dispositif d'affichage 3D prenant en

charge Unity3D, tel que le HMD, la salle immersive ou l'affichage sur écran. En outre, les composants standard des fauteuils roulants fixés sur la structure mécanique permettent à la plateforme d'être conforme aux exigences cliniques. En effet, le simulateur proposé est directement compatible avec les besoins de l'utilisateur de fauteuil roulant en matière de siège et de contrôle.

Finalement, le simulateur est compatible avec l'ensemble des outils présentés dans les chapitres précédents. En particulier, en environnement de foule peut être créé pour les utilisateurs du PWS.

Chapitre

10 Conclusion et perspectives

Sommaire

Cette thèse s'inscrit dans le contexte de la robotique mobile dans un environnement de foule. Ce type d'environnement reste problématique pour les roboticiens aujourd'hui :

— il faut résoudre les problèmes *techniques* liés à la présence d'une foule : le robot doit pouvoir se repérer dans ce milieu hautement dynamique, et doit pouvoir détecter les personnes environnantes et comprendre leurs intentions pour atteindre son but ;

— une foule partageant un espace avec un robot potentiellement dangereux soulève des problèmes éthiques liés à la sécurité des humains ;

— les simulations de robots dans la foule font systématiquement des hypothèses fortes sur le comportement de la foule simulée, hypothèses qui ne tiennent pas pour le cas d'une foule réelle.

L'objectif principal de cette thèse était **l'évaluation par la simulation de la navigation de robots mobiles dans la foule**. Dans cette optique, nous avons conçu un outil de simulation modulaire permettant de créer des interactions robot-foule hautement réalistes. Dans cette thèse, nous avons d'abord décrit l'architecture de cet outil visant à combler un manque en simulation robotique : le réalisme de la simulation de foule. Nous en avons détaillé les éléments principaux et comment ils s'articulent et communiquent entre eux. L'outil est lui-même composé d'outils standards, comme ROS, ou alors propose des outils de standardisation, notamment pour la foule avec un principe unificateur. Ainsi, le simulateur est à la fois capable d'immerger un robot dans une foule simulée et réaliste, et d'immerger un humain dans la simulation grâce à un module d'immersion en Réalité Virtuelle. Enfin, le simulateur propose un banc d'essai permettant l'analyse de la navigation de robot dans la foule. Les contributions de cette thèse sont centrées sur ces éléments fondamentaux du simulateur robot-foule.

10.1 Contributions

10.1.1 Simulation de foule

Nous avons d'abord détaillé notre approche de la simulation de foule microscopique dans le chapitre 5. Elle consiste à déplacer des agents en leur attribuant une vitesse résultant d'une optimisation continue d'une fonction de coût. Cette fonction de coût prend en compte leurs buts et les interactions avec les agents voisins. Nous avons expliqué comment cette approche peut unifier un ensemble d'algorithmes existants qui reposent sur un principe équivalent. Il nous est possible d'imiter ces algorithmes en trouvant, pour chacun d'entre eux, la bonne fonction de coût, couplée à la bonne méthode d'optimisation. L'intérêt d'une telle approche pour notre simulateur robot-foule est qu'elle permet d'utiliser de nombreuses méthodes de simulation de foule différentes de façon transparente pour l'utilisateur : on peut facilement passer d'une méthode à une autre pour simuler le comportement de la foule. On peut également utiliser une foule hybride, composée de plusieurs agents régis par différentes fonctions de coût s'activant dans des situations différentes. Avec cette approche, il n'y a donc plus de question de choisir arbitrairement un simulateur de foule, car ils sont tous disponible dans le simulateur.

10.1.2 Évaluation d'interaction robot-foule en simulation

Nous avons présenté une méthode d'analyse d'interaction robot-foule dans le chapitre 6. Nous avons montré que notre outil pouvait être utile pour comparer différentes techniques de navigation de robot mobile dans la foule. Il permet en effet d'explorer automatiquement un grand nombre de scénarios, et permet ainsi la comparaisons immédiate des performances des algorithmes de navigation de robots selon les scénarios. De plus, nous proposons des scénarios standard, qui sont définis par la configuration de la foule et la tâche du robot. Ces scénarios peuvent ainsi être partagés à l'échelle de toute une communauté. Au final, notre simulateur est le candidat idéal pour évaluer et comparer équitablement le comportement des robots dans une foule.

10.1.3 Immersion en réalité virtuelle pour le robot et l'humain

Nous avons ensuite présenté dans le chapitre 7 comment aller plus loin pour créer des expériences impliquant de vraies personnes, expériences qui soit à la fois sûres pour les participants et réalistes en termes de résultats obtenus, grâce à la réalité virtuelle. En effet, si l'évaluation basée sur la simulation offre de précieuses informations, notamment grâce au banc d'essai, les résultats issus de simulation peuvent souffrir des hypothèses fortes intrinsèques à la simulation. En particulier, les comportements qu'une foule peut présenter ne peuvent pas être pleinement appréciés par des algorithmes. Notre solution consiste donc en un module de réalité virtuelle pour notre simulateur robot-foule. La contribution majeure ici est d'immerger à la fois un humain et un robot dans un environnement virtuel partagé. C'est une méthode sûre pour étudier les interactions entre l'homme et le robot, et elle dépasse le niveau de réalisme qui peut être atteint dans un environnement de simulation pure. Nous avons montré via une expérience impliquant un humain et un robot que notre plateforme pouvait servir à l'étude des interactions humain-robot. Notre outil comble ainsi un manque entre la simulation et les expériences en conditions réelles.

10.1.4 Bracelets haptiques pour les contacts entre humain et une foule en réalité virtuelle

Nous avons ensuite présenté dans le chapitre 8 l'usage de brassards haptiques en réalité virtuelle visant à augmenter le sentiment de présence des participants

navigant au sein d'une foule virtuelle. Pour ce faire, nous avons présenté une expérience visant identifier les effets, ainsi que les effets d'apprentissage, du rendu haptique sur une tâche de mouvement dans un environnement très fréquenté. Les résultats ont montré que le fait de fournir un retour d'information haptique avait un impact sur la façon dont les participants se déplaçaient dans la foule virtuelle. Ils étaient plus prudents quant aux collisions qu'ils provoquaient avec les personnages virtuels, sans toutefois changer leurs trajectoires globales. Nous avons également démontré la présence d'un effet secondaire d'apprentissage du retour haptique, puisque les changements observés dans leurs mouvements du fait de ce retour haptique sont restés observables, même après la désactivation du retour haptique. Cette expérience nous permet d'en savoir davantage sur les biais comportement des participants aux expériences en réalité virtuelle. Finalement, nous suggérons que l'ajout d'un retour haptique est indiqué pour étudier le comportement humain en réalité virtuelle si il y a des contacts avec l'environnement.

10.1.5 Simulateur mécanique de fauteuil roulant électrique

Finalement, le chapitre 9 présente une collaboration avec le consortium ADAPT de l'INSA de Rennes. Il s'agit d'une conception centrée sur l'utilisateur d'un simulateur de fauteuil roulant électrique multisensoriel, utilisant la réalité virtuelle et un retour auditif, avec lesquels se synchronisent les mouvements d'une structure mécanique innovante à 4 degrés de liberté. Cette structure mécanique est compatible avec le simulateur robot-foule présenté dans cette thèse.

10.2 Perspectives

Cette section détaille quelles perspectives j'ai pour ce simulateur robot-foule, qui s'articulent autour de deux axes majeurs : la simulation pure, et les expériences en réalité virtuelle.

10.2.1 Le simulateur robot-foule

Nous percevons le simulateur robot-foule, associé à l'outil d'analyse proposé, comme un moyen unique de générer une grande quantité de données. Il sera com-

plètement libre de droit et d'accès [1] pour la communauté.

De notre côté, nous comptons d'abord améliorer globalement l'expérience utilisateur afin de permettre au plus grand nombre de s'approprier l'outil. Cela passe d'abord par la stabilisation des fonctionnalités actuelles et par la mise en place d'outils graphiques pour la paramétrisation de la simulation, mais aussi l'ajout de fonctionnalités nouvelles. Par exemple, le simulateur doit être capable de reproduire des trajectoires de foules pré-enregistrées. Si cela est déjà possible pour des trajectoires issues du simulateur, nous souhaitons en plus pouvoir utiliser des jeux de données de foule issues de situations réelles. Par exemple, il existe des jeux de données de trajectoires de foule issues d'enregistrements vidéos comme celui utilisé dans Pellegrini et al. [2012] ou encore Lerner et al. [2007]. Les jeux de données publiques disponibles en ligne sont la plupart du temps formatés différemment et nécessitent donc une intégration au cas par cas si on souhaite utiliser plusieurs jeux de données. Cependant, une intégration plus directe est possible grâce à un outil intermédiaire : le dépôt `https://github.com/crowdbotp/OpenTraj` répertorie des jeux de données publiques de trajectoires de foule, et permet le chargement des trajectoire de manière uniforme et nous souhaitons l'intégrer dans le simulateur robot-foule.

Nous souhaitons également à court terme étendre les mesures grâce auxquelles nous évaluons le comportement des robots via l'outil d'analyse. Par exemple, nous voudrions mesurer les situations de quasi-collision, correspondant aux instants précédents la collision où l'évitement est de l'ordre du réflexe, pour évaluer plus en profondeur les aspects de sécurité. D'autres éléments que les seules trajectoires devraient en outre être pris en compte. Par exemple, la simulation peut approfondir la question de la détection de la foule, en comparant par exemple les résultats d'analyse de la navigation dans des scénarios comme ceux présentés au chapitre 6 dans différentes configurations de capteurs. Une autre direction consisterait à étendre le nombre de scénarios pour l'analyse que nous envisageons. La force de la simulation est son potentiel à explorer de nombreuses situations pour le robot, et elle est encore inexploitée dans cette thèse. La conception des scénarios les plus intéressants devrait cependant se faire à l'échelle de la communauté.

A plus long terme, nous comptons utiliser cet outil dans le cadre de collaborations à venir.

— Notre idée est de générer une grande quantité de données de capteurs LI-

1. Ce simulateur robot-foule est en partie disponible sur le site `http://crowdbot.eu/data-sets-softwares/`.

DAR issus de la simulation d'un robot dans une foule virtuelle. Ces données pourrons alors servir pour l'apprentissage profond d'un réseau de neurones servant à détecter les personnes environnantes.

— Sur le même principe, mais en ajoutant des données de caméras RGBD, nous pouvons utiliser le simulateur comme outil d'apprentissage pour entraîner un réseau de neurones permettant de contrôler directement le robot en fonction des capteurs (sans modèle intermédiaire).

— Combinant les deux principes, nous souhaitons utiliser notre simulateur pour faire apprendre à un réseau de neurones contrôlant le mouvement du robot à *détecter une personne* et la *suivre* dans la foule.

10.2.2 Interactions humain-robot en réalité virtuelle

Le module de réalité virtuelle présenté dans le chapitre 7 a d'abord été utilisé via une expérience pilote. Cela a permis de mettre en avant quelques problèmes techniques lors de la mise en place du système et de mettre en lumière les limitations de la solution. Le système décrit dans le chapitre 7 est complexe et coûteux. Il existe certaines contraintes, comme une zone de suivi limitée, des capacités en temps réel limitées. Chaque sous-système est sujet à des défauts potentiels, et a des exigences système qui rendent l'ensemble de la plate-forme difficile à maintenir. Par exemple, une limitation vient du fait que nous avons utilisé uniquement l'odométrie du robot pour estimer son mouvement pendant l'expérience. Or l'odométrie est sujette à des dérives et à une surestimation de la vitesse. Pour résoudre ce problème, nous aimerions utiliser un filtre de Kalman utilisant à la fois les données du système de suivi *Vicon* et l'odométrie du robot pour avoir un suivi plus précis sans aucune perte. Un autre exemple : la latence est une source majeure de perturbation pour le participant, et peut provoquer le mal des simulateurs. En outre, la vitesse du robot a été limitée à un maximum (<0,35 m/s), ce qui a limité la portée des résultats. Certains participants ont déclaré de plus avoir entendu le robot réel se déplacer pendant qu'ils étaient en réalité virtuelle et ont utilisé ce son du robot réel se déplaçant dans son propre espace de travail pour synchroniser leur mouvement dans le monde virtuel. Certains participants ont également été perturbés par leur avatar, qui visuellement ne correspondait pas parfaitement à leur propre corps. De plus, le HMD *Fove* que nous avons utilisé a un champ de vision limité, ce qui a un impact sur l'immersion des participants. Enfin, en ce qui concerne l'expérience pilote, le nombre de participants est faible (5).

À court terme, nous devons donc prendre en compte les questions soulevées par cette étude pour valider correctement la plate-forme. Pour ce faire, nous de-

vons d'abord traiter les questions techniques. Une première optimisation (post-expérience) a montré que la fréquence d'acquisition des données peut atteindre au moins 30Hz (20,6Hz pendant les expériences). Concernant le bruit que fait le robot, qui a pu perturber l'interaction, la solution à long terme consistera à mettre en place une deuxième salle équipée d'un système de suivi, synchronisé avec le suivi de la première salle, afin de réaliser les expériences dans deux salles distinctes. Pour le court terme, nous avons amélioré le système au chapitre 8 : le participant est désormais équipé d'un casque pour couvrir le son ambiant et pour déplacer la source sonore à la position virtuelle du robot. Nous avons en outre utilisé une salle de suivi plus grande et un HMD avec un plus grand champ de vision, et des brassard haptiques. Par ailleurs, les avatars étaient plus adaptés aux participants, et les participants étaient plus nombreux. À long terme, nous envisageons d'ajouter des systèmes de suivi Xsens pour permettre la participation de plusieurs utilisateurs autour du robot.

Concernant les expériences elles mêmes, nous souhaitons mener une série d'expériences pour évaluer les biais restants en tenant compte des nouvelles tâches d'interaction. En particulier, nous devrions étudier les cas dans lesquels le robot effectue un mouvement en utilisant une combinaison de différentes sinusoïdes avec des fréquences et des déphasages différents, ainsi que le cas où l'humain effectue un mouvement totalement libre, afin de comparer notre système plus profondément avec Ducourant et al. [2005]. En outre, donner plus de liberté à l'humain nous permettra d'analyser l'interaction avec d'autres métriques telles que l'efficacité de la trajectoire, l'orientation des épaules ou les secousses (*jerk*) dans le mouvement. De plus, de par sa conception, le participant est à l'abri des contacts avec le robot et le simulateur permet de contourner les règles de sécurité du robot et d'éviter les problèmes classiques des interactions homme-robot tels que le problème de robots trop prudents s'arrêtant bien avant la collision (*"Freeze robot problem"* en anglais) et responsable de potentiels blocages de la foule.

Nous voulons donc utiliser cette plateforme dans les études HRI qui présentent des risques de collisions. Par exemple, nous voulons étudier le comportement des mouvements humains avant les collisions avec un robot, et nous voulons comparer différents algorithmes de navigation en rendant compte des collisions virtuelles. De plus, il serait intéressant de travailler sur des scénarios de croisement (avec ou sans évitement de collision). Nous aimerions également réaliser des expériences avec différents robots, par exemple avec un fauteuil roulant électrique. Nous souhaiterions également étudier l'effet d'arrêts brutaux du robot sur le comportement d'une personne qui suit ce robot. Enfin, cette plateforme pourrait être utilisée pour valider des algorithmes basés sur l'apprentissage de réseaux de neurones qui ont

été entraînés en simulation, et devenir le pont entre la simulation et la réalité.

Pour nos futurs travaux, nous souhaitons peupler nos environnements virtuels de personnages virtuels plus interactifs et réactifs. C'est un aspect crucial car il semble nécessaire d'améliorer encore le sentiment de présence des participants. En outre, l'ajout de personnages réactifs pourrait accroître l'effet du rendu haptique, car on pourrait s'attendre à des réactions plus fortes des participants si les personnages virtuels réagissent également après une collision. Une analyse plus détaillée qui évalue le mouvement avant et après le rendu d'une collision et la réaction d'un personnage virtuel serait alors également pertinente pour l'étude. Nous prévoyons également d'utiliser à l'avenir des dispositifs haptiques portables plus convaincants afin de fournir une sensation de collision plus réaliste tout en gardant l'ensemble du système compact et facile à porter : on peut ainsi penser par exemple à des dispositifs d'étirement de la peau tel que les dispositifs présentés par Chinello et al. [2017] ou de tapotement de l'épaule et du haut du bras.

Enfin, la dernière partie de nos perspectives concerne la collaboration avec le consortium ADAPT de l'INSA de Rennes à propos du simulateur mécanique de fauteuil roulant électrique. L'amélioration de la plateforme elle même est du ressort du consortium. Cependant, en ce qui concerne la restitution visuelle et l'environnement virtuel, nous voulons peupler les scènes 3D d'entités dynamiques telles qu'une foule, des véhicules, etc. Nous voulons également explorer d'autres cas d'utilisation. Ainsi, par exemple, sur la base des travaux de l'INSA de Rennes sur la robotique assistée, nous pourrions envisager d'ajouter des capteurs virtuels et un système d'évitement des obstacles dans le simulateur Devigne et al. [2017] et bénéficier ainsi complètement des avantages du simulateur robot-foule présenté dans cette thèse.

10.3 Ouverture

Aujourd'hui, la robotique mobile, et au sens plus large, les réseaux de systèmes autonomes et connectés doivent répondre à 4 contraintes : la sécurité, le respect de la vie privée, l'efficacité à effectuer sa tâche, et la robustesse aux cyber-attaques [Le Lann, 2019]. Se concentrer sur une de ces contraintes uniquement, c'est négliger toutes les autres. La conception de systèmes tels que des robots mobiles voués à interagir avec les humains doit aujourd'hui consister à trouver le compromis entre ces éléments. Cependant, cette affaire n'est pas purement technique, mais bien sociologique. Sommes-nous prêts à donner des informations sur notre vie privée pour améliorer la sécurité ? Par exemple, accepte-t-on d'être filmés par une voiture autonome afin d'être correctement détecté par celle-ci pour qu'elle freine devant

nous ? Un autre exemple, une personne en situation de handicap acceptera-t-elle d'avoir un fauteuil intelligent lui permettant de naviguer dans la foule, mais qui soit fragile face aux cyber-attaques ? La collaboration entre roboticiens, juristes, psychologues, et experts dans la sécurité semble nécessaire pour répondre à ces nouveaux besoins.

D'un point de vue robotique en tout cas, nous espérons que cette thèse contribuera à trouver le compromis entre sécurité et efficacité pour nos robots mobiles.

Contributions

1. **Fabien Grzeskowiak**, David Julian Gonon, Daniel Dugas, Diego Paez-Granados, Jen Jen Chung, Juan Nieto, Roland Siegwart, Aude Billard, Marie Babel, Julien Pettre« Crowd against the machine : A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd »
2. **F. Grzeskowiak**, M. Babel, J. Bruneau and J. Pettre, "Toward Virtual Reality-based Evaluation of Robot Navigation among People," 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Atlanta, GA, USA, 2020, pp. 766-774, doi : 10.1109/VR46266.2020.00100.
3. Wouter van Toll, **Fabien Grzeskowiak**, Axel López Gandía, Javad Amirian, Florian Berton, Julien Bruneau, Beatriz Cabrero Daniel, Alberto Jovane, and Julien Pettré. 2020. Generalized Microscropic Crowd Simulation using Costs in Velocity Space. In Symposium on Interactive 3D Graphics and Games (I3D '20). Association for Computing Machinery, New York, NY, USA, Article 6, 1–9. DOI :https ://doi.org/10.1145/3384382.3384532
4. Berton, F., **Grzeskowiak, F.**, Bonneau, A., Jovane, A., Aggravi, M., Hoyet, L., Olivier, A.H., Pacchierotti, C. and Pettré, J., 2020. Crowd Navigation in VR : exploring haptic rendering of collisions. IEEE Transactions on Visualization and Computer Graphics.
5. Vailland, G., **Grzeskowiak, F.**, Devigne, L., Gaffary, Y., Fraudet, B., Leblong, E., ... Babel, M. (2019, June). User-centered design of a multisensory power wheelchair simulator : towards training and rehabilitation applications. In 2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR) (pp. 77-82). IEEE.

Table des figures

Liste des tableaux

Bibliographie

P. Abellard, I. Randria, A. Abellard, M. B. Khelifa, and P. Ramanantsizehena. Electric wheelchair navigation simulators : why, when, how ? *Mechatronic Systems Applications*, pages 161–168, 2010.

J. Achenbach, T. Waltemate, M. E. Latoschik, and M. Botsch. Fast generation of realistic virtual humans. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, pages 1–10, 2017.

P. Agethen, V. S. Sekar, F. Gaisbauer, T. Pfeiffer, M. Otto, and E. Rukzio. Behavior analysis of human locomotion in the real world and virtual reality for the manufacturing industry. *ACM Transactions on Applied Perception (TAP)*, 15(3) :20, 2018a.

P. Agethen, V. S. Sekar, F. Gaisbauer, T. Pfeiffer, M. Otto, and E. Rukzio. Behavior Analysis of Human Locomotion in the Real World and Virtual Reality for the Manufacturing Industry. *ACM Transactions on Applied Perception*, 15 (3) :20 :1–20 :19, July 2018b. ISSN 1544-3558. doi : 10.1145/3230648. URL https://doi.org/10.1145/3230648.

J. Amirian, J.-B. Hayet, and J. Pettre. Social Ways : Learning Multi-Modal Distributions of Pedestrian Trajectories With GANs. pages 0–0, 2019a. URL https://openaccess.thecvf.com/content_CVPRW_2019/html/Precognition/Amirian_Social_Ways_Learning_Multi-Modal_Distributions_of_Pedestrian_Trajectories_With_GANs_CVPRW_2019_paper.html.

J. Amirian, W. van Toll, J.-B. Hayet, and J. Pettré. Data-Driven Crowd Simulation with Generative Adversarial Networks. In *Proceedings of the 32nd International Conference on Computer Animation and Social Agents - CASA '19*, pages 7–10, Paris, France, 2019b. ACM Press. ISBN 978-1-4503-7159-9. doi : 10.1145/3328756.3328769. URL http://dl.acm.org/citation.cfm?doid=3328756.3328769.

G. Aravind, A. Darekar, J. Fung, and A. Lamontagne. Virtual Reality-Based Navigation Task to Reveal Obstacle Avoidance Performance in Individuals With

Visuospatial Neglect. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(2) :179–188, Mar. 2015. ISSN 1558-0210. doi : 10.1109/TNSRE.2014.2369812. Conference Name : IEEE Transactions on Neural Systems and Rehabilitation Engineering.

S. Arlati, V. Colombo, G. Ferrigno, R. Sacchetti, and M. Sacco. Virtual reality-based wheelchair simulators : A scoping review. *Assistive Technology*, 2019.

C. Armbrüster, M. Wolter, T. Kuhlen, W. Spijkers, and B. Fimm. Depth Perception in Virtual Reality : Distance Estimations in Peri- and Extrapersonal Space. *CyberPsychology & Behavior*, 11(1) :9–15, Feb. 2008. ISSN 1094-9313. doi : 10.1089/cpb.2007.9935. URL https://www.liebertpub.com/doi/10.1089/cpb.2007.9935. Publisher : Mary Ann Liebert, Inc., publishers.

A. Baudry, S. Guegan, and M. Babel. Taking caster wheel behavior into account in the kinematics of powered wheelchairs. *Modelling, measurement and control C*, 79(4) :168–172, 2018.

J. v. d. Berg, Ming Lin, and D. Manocha. Reciprocal Velocity Obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935, May 2008. doi : 10.1109/ROBOT.2008.4543489. ISSN : 1050-4729.

J. Bimbo, C. Pacchierotti, M. Aggravi, N. Tsagarakis, and D. Prattichizzo. Teleoperation in cluttered environments using wearable haptic feedback. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3401–3408. IEEE, 2017.

J. Bruneau. *Studying and modeling complex interactions for crowd simulation.* PhD thesis, Rennes 1, 2016.

J. Bruneau and J. Pettré. EACS : Effective Avoidance Combination Strategy. *Computer Graphics Forum*, 36(8) :108–122, 2017. ISSN 1467-8659. doi : https://doi.org/10.1111/cgf.13066. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13066. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13066.

J. Bruneau, A. Olivier, and J. Pettré. Going Through, Going Around : A Study on Individual Avoidance of Groups. *IEEE Transactions on Visualization and Computer Graphics*, 21(4) :520–528, Apr. 2015. ISSN 1941-0506. doi : 10.1109/TVCG.2015.2391862. Conference Name : IEEE Transactions on Visualization and Computer Graphics.

M. A. Bühler and A. Lamontagne. Circumvention of pedestrians while walking in virtual and physical environments. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2018.

W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1) :3–55, Oct. 1999. ISSN 0004-3702. doi : 10.1016/S0004-3702(99)00070-3. URL http://www.sciencedirect.com/science/article/pii/S0004370299000703.

M. A. Bühler and A. Lamontagne. Circumvention of Pedestrians While Walking in Virtual and Physical Environments. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(9) :1813–1822, Sept. 2018. ISSN 1558-0210. doi : 10.1109/TNSRE.2018.2865907. Conference Name : IEEE Transactions on Neural Systems and Rehabilitation Engineering.

P. Castillo-Pizarro, T. V. Arredondo, and M. Torres-Torriti. Introductory Survey to Open-Source Mobile Robot Simulation Software. In *2010 Latin American Robotics Symposium and Intelligent Robotics Meeting*, pages 150–155, Oct. 2010. doi : 10.1109/LARS.2010.19.

P. Charalambous and Y. Chrysanthou. The PAG Crowd : A Graph Based Approach for Efficient Data-Driven Crowd Simulation. *Computer Graphics Forum*, 33(8) :95–108, 2014. ISSN 1467-8659. doi : https://doi.org/10.1111/cgf.12403. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12403. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12403.

C. Chen, Y. Liu, S. Kreiss, and A. Alahi. Crowd-Robot Interaction : Crowd-aware Robot Navigation with Attention-based Deep Reinforcement Learning. *arXiv :1809.08835 [cs]*, Feb. 2019. URL http://arxiv.org/abs/1809.08835. arXiv : 1809.08835.

S. Chenney. Flow tiles. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '04, pages 233–242, Goslar, DEU, Aug. 2004. Eurographics Association. ISBN 978-3-905673-14-2. doi : 10.1145/1028523.1028553. URL https://doi.org/10.1145/1028523.1028553.

L. P. Chew. Constrained delaunay triangulations. *Algorithmica*, 4(1) :97–108, 1989.

W. Chi, H. Kono, Y. Tamura, A. Yamashita, H. Asama, and M. Q. Meng. A human-friendly robot navigation algorithm using the risk-RRT approach. In

2016 IEEE International Conference on Real-time Computing and Robotics (RCAR), pages 227–232, June 2016. doi : 10.1109/RCAR.2016.7784030.

F. Chinello, C. Pacchierotti, J. Bimbo, N. G. Tsagarakis, and D. Prattichizzo. Design and evaluation of a wearable skin stretch device for haptic guidance. *IEEE Robotics and Automation Letters*, 3(1) :524–531, 2017.

S. Cho, J. Park, and O. Kwon. Gender differences in three dimensional gait analysis data from 98 healthy korean adults. *Clinical biomechanics*, 19(2) : 145–152, 2004.

Y.-h. Chou, R. C. Wagenaar, E. Saltzman, J. E. Giphart, D. Young, R. Davidsdottir, and A. Cronin-Golomb. Effects of Optic Flow Speed and Lateral Flow Asymmetry on Locomotion in Younger and Older Adults : A Virtual Reality Study. *The Journals of Gerontology : Series B*, 64B(2) :222–231, Mar. 2009. ISSN 1079-5014. doi : 10.1093/geronb/gbp003. URL https://academic.oup.com/psychsocgerontology/article/64B/2/222/552215. Publisher : Oxford Academic.

G. Cirio, A. Olivier, M. Marchal, and J. Pettré. Kinematic Evaluation of Virtual Walking Trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 19(4) :671–680, Apr. 2013a. ISSN 1941-0506. doi : 10.1109/TVCG.2013.34. Conference Name : IEEE Transactions on Visualization and Computer Graphics.

G. Cirio, A.-H. Olivier, M. Marchal, and J. Pettré. Kinematic evaluation of virtual walking trajectories. *IEEE transactions on visualization and computer graphics*, 19(4) :671–680, 2013b.

G. Cirio, A.-H. Olivier, M. Marchal, and J. Pettre. Kinematic evaluation of virtual walking trajectories. *IEEE transactions on visualization and computer graphics*, 19(4) :671–680, 2013c.

J. Craighead, R. Murphy, J. Burke, and B. Goldiez. A Survey of Commercial Open Source Unmanned Vehicle Simulators. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 852–857, Apr. 2007. doi : 10.1109/ROBOT.2007.363092. ISSN : 1050-4729.

S. Curtis, A. Best, and D. Manocha. Menge : A modular framework for simulating crowd movement. *Collective Dynamics*, 1(A1) :1–40, 2016.

J. Dascal, M. Reid, W. W. Ishak, B. Spiegel, J. Recacho, B. Rosen, and I. Danovitch. Virtual reality and medical inpatients : a systematic review of randomized, controlled trials. *Innovations in clinical neuroscience*, 14(1-2) :14, 2017.

A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi. An atlas of physical human–robot interaction. *Mechanism and Machine Theory*, 43(3) : 253–270, Mar. 2008. ISSN 0094-114X. doi : 10.1016/j.mechmachtheory.2007.03.003. URL http://www.sciencedirect.com/science/article/pii/S0094114X07000547.

L. Devigne, M. Babel, F. Nouviale, V. K. Narayanan, F. Pasteau, and P. Gallien. Design of an immersive simulator for assisted power wheelchair driving. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 995–1000, July 2017. doi : 10.1109/ICORR.2017.8009379. ISSN : 1945-7901.

L. Devigne, M. Aggravi, M. Bivaud, N. Balix, C. S. Teodorescu, T. Carlson, T. Spreters, C. Pacchierotti, and M. Babel. Power wheelchair navigation assistance using wearable vibrotactile haptics. *IEEE transactions on haptics*, 13 (1) :52–58, 2020.

Dongqing Shi, E. G. C. Jr, B. Goldiez, A. Donate, Xiuwen Liu, and D. Dunlap. Human-aware robot motion planning with velocity constraints. In *2008 International Symposium on Collaborative Technologies and Systems*, pages 490–497, May 2008. doi : 10.1109/CTS.2008.4543969.

A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla : An open urban driving simulator. *arXiv preprint arXiv :1711.03938*, 2017.

T. Ducourant, S. Vieilledent, Y. Kerlirzin, and A. Berthoz. Timing and distance characteristics of interpersonal coordination during locomotion. *Neuroscience Letters*, 389(1) :6–11, Nov. 2005. ISSN 0304-3940. doi : 10.1016/j.neulet.2005.06.052. URL http://www.sciencedirect.com/science/article/pii/S0304394005007524.

T. B. Dutra, R. Marques, J. B. Cavalcante-Neto, C. A. Vidal, and J. Pettré. Gradient-based steering for vision-based crowd simulation algorithms. *Computer Graphics Forum*, 36(2) :337–348, 2017. ISSN 1467-8659. doi : https://doi.org/10.1111/cgf.13130. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13130. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13130.

G. Echeverria, N. Lassabe, A. Degroote, and S. Lemaignan. Modular open robots simulation engine : MORSE. In *2011 IEEE International Conference on Robotics and Automation*, pages 46–51, May 2011. doi : 10.1109/ICRA.2011.5980252. ISSN : 1050-4729.

T. Fan, X. Cheng, J. Pan, D. Manocha, and R. Yang. CrowdMove : Autonomous Mapless Navigation in Crowded Scenarios. *arXiv :1807.07870 [cs]*, July 2018. URL http://arxiv.org/abs/1807.07870. arXiv : 1807.07870.

P. W. Fink, P. S. Foo, and W. H. Warren. Obstacle avoidance during walking in real and virtual environments. *ACM Transactions on Applied Perception*, 4 (1) :2–es, Jan. 2007. ISSN 1544-3558. doi : 10.1145/1227134.1227136. URL https://doi.org/10.1145/1227134.1227136.

M. Fiore, A. Clodic, and R. Alami. On Planning and Task Achievement Modalities for Human-Robot Collaboration. In M. A. Hsieh, O. Khatib, and V. Kumar, editors, *Experimental Robotics : The 14th International Symposium on Experimental Robotics*, Springer Tracts in Advanced Robotics, pages 293–306. Springer International Publishing, Cham, 2016. ISBN 978-3-319-23778-7. doi : 10.1007/978-3-319-23778-7_20. URL https://doi.org/10.1007/978-3-319-23778-7_20.

P. Fiorini and Z. Shiller. Motion Planning in Dynamic Environments Using Velocity Obstacles. *The International Journal of Robotics Research*, 17(7) :760–772, July 1998. ISSN 0278-3649. doi : 10.1177/027836499801700706. URL https://doi.org/10.1177/027836499801700706. Publisher : SAGE Publications Ltd STM.

D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1) :23–33, Mar. 1997. ISSN 10709932. doi : 10.1109/100.580977. URL http://ieeexplore.ieee.org/document/580977/.

T. Fraichard and V. Levesy. From Crowd Simulation to Robot Navigation in Crowds. *IEEE Robotics and Automation Letters*, 5(2) :729–735, Apr. 2020. ISSN 2377-3766, 2377-3774. doi : 10.1109/LRA.2020.2965032. URL https://ieeexplore.ieee.org/document/8954815/.

P. Fuchs. *Les interfaces de la réalité virtuelle*. éditeur AJIIMD, 1996.

Y. Gan. *Continuum Mechanics : Progress in Fundamentals and Engineering Applications*. BoD–Books on Demand, 2012.

M. Gérin-Lajoie, C. L. Richards, J. Fung, and B. J. McFadyen. Characteristics of personal space during obstacle circumvention in physical and virtual environments. *Gait & posture*, 27(2) :239–247, 2008.

B. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project : Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th international conference on advanced robotics*, volume 1, pages 317–323. Citeseer, 2003.

D. F. P. Granados, H. Kadone, and K. Suzuki. Unpowered Lower-Body Exoskeleton with Torso Lifting Mechanism for Supporting Sit-to-Stand Transitions. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2755–2761, Madrid, Oct. 2018. IEEE. ISBN 978-1-5386-8094-0. doi : 10.1109/IROS.2018.8594199. URL https://ieeexplore.ieee.org/document/8594199/.

A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN : Socially Acceptable Trajectories With Generative Adversarial Networks. pages 2255–2264, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Gupta_Social_GAN_Socially_CVPR_2018_paper.html.

S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. Lin, and D. Manocha. PLEdestrians : A Least-Effort Approach to Crowd Simulation. *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, page 10 pages, 2010. ISSN 1727-5288. doi : 10.2312/SCA/SCA10/119-128. URL http://diglib.eg.org/handle/10.2312/SCA.SCA10.119-128. Artwork Size : 10 pages ISBN : 9783905674279 Publisher : The Eurographics Association.

Hall. *Hall, The Hidden Dimension*. Garden City, NY : Doubleday, 1966.

R. T. Hays, J. W. Jacobs, C. Prince, and E. Salas. Flight simulator training effectiveness : A meta-analysis. *Military psychology*, 4(2) :63–74, 1992.

M. L. Heilig. Sensorama simulator (1962). *URL : http ://www. freepatentsonline. com/3050870. html*, 1, 1962.

D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5) :4282–4286, May 1995. doi : 10.1103/PhysRevE.51.4282. URL https://link.aps.org/doi/10.1103/PhysRevE.51.4282. Publisher : American Physical Society.

F. Hernoux, E. Nyiri, and O. Gibaru. Virtual reality for improving safety and collaborative control of industrial robots. In *Proceedings of the 2015 Virtual Reality International Conference*, VRIC '15, pages 1–6, New York, NY, USA, Apr. 2015. Association for Computing Machinery. ISBN 978-1-4503-3313-9. doi : 10.1145/2806173.2806197. URL https://doi.org/10.1145/2806173.2806197.

M. C. Howard. A meta-analysis and systematic literature review of virtual reality rehabilitation programs. *Computers in Human Behavior*, 70 :317–327, 2017.

R. L. Hughes. A continuum theory for the flow of pedestrians. *Transportation Research Part B : Methodological*, 36(6) :507–535, July 2002. ISSN 0191-2615. doi : 10.1016/S0191-2615(01)00015-7. URL http://www.sciencedirect.com/science/article/pii/S0191261501000157.

R. L. Hughes. The Flow of Human Crowds. *Annual Review of Fluid Mechanics*, 35(1) :169–182, 2003. doi : 10.1146/annurev.fluid.35.101101.161136. URL https://doi.org/10.1146/annurev.fluid.35.101101.161136. _eprint : https ://doi.org/10.1146/annurev.fluid.35.101101.161136.

D. P. Inman, K. Loge, and J. Leavens. Vr education and rehabilitation. *Communications of the ACM*, 40(8) :53–58, 1997.

S. Ivaldi, V. Padois, and F. Nori. Tools for dynamics simulation of robots : a survey based on user feedback. *arXiv :1402.7050 [cs]*, Feb. 2014. URL http://arxiv.org/abs/1402.7050. arXiv : 1402.7050.

A. Javaheri, N. Moghadamnejad, H. Keshavarz, E. Javaheri, C. Dobbins, E. Momeni, and R. Rawassizadeh. Public vs Media Opinion on Robots. *arXiv :1905.01615 [cs]*, May 2019. URL http://arxiv.org/abs/1905.01615. arXiv : 1905.01615.

X. Jin, J. Xu, C. C. L. Wang, S. Huang, and J. Zhang. Interactive Control of Large-Crowd Navigation in Virtual Environments Using Vector Fields. *IEEE Computer Graphics and Applications*, 28(6) :37–46, Nov. 2008. ISSN 1558-1756. doi : 10.1109/MCG.2008.117. Conference Name : IEEE Computer Graphics and Applications.

M. P. Joosse, R. W. Poppe, M. Lohse, and V. Evers. Cultural differences in how an engagement-seeking robot should approach a group of people. In *Proceedings of the 5th ACM international conference on Collaboration across boundaries : culture, distance & technology*, CABS '14, pages 121–130, Kyoto, Japan, Aug.

2014. Association for Computing Machinery. ISBN 978-1-4503-2557-8. doi : 10.1145/2631488.2631499. URL https://doi.org/10.1145/2631488.2631499.

K. Jordao, J. Pettré, M. Christie, and M.-P. Cani. Crowd sculpting : A space-time sculpting method for populating virtual environments. *Computer Graphics Forum*, 33(2) :351–360, 2014. ISSN 1467-8659. doi : https://doi.org/10.1111/cgf.12316. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12316. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12316.

E. Ju, M. G. Choi, M. Park, J. Lee, K. H. Lee, and S. Takahashi. Morphable crowds. *ACM Transactions on Graphics*, 29(6) :140 :1–140 :10, Dec. 2010. ISSN 0730-0301. doi : 10.1145/1882261.1866162. URL https://doi.org/10.1145/1882261.1866162.

V. K. Narayanan, F. Pasteau, M. Marchal, A. Krupa, and M. Babel. Vision-based adaptive assistance and haptic guidance for safe wheelchair corridor following. *Computer Vision and Image Understanding*, 149 :171–185, Aug. 2016. ISSN 1077-3142. doi : 10.1016/j.cviu.2016.02.008. URL http://www.sciencedirect.com/science/article/pii/S1077314216000539.

I. Karamouzas and M. Overmars. A Velocity-Based Approach for Simulating Human Collision Avoidance. In J. Allbeck, N. Badler, T. Bickmore, C. Pelachaud, and A. Safonova, editors, *Intelligent Virtual Agents*, Lecture Notes in Computer Science, pages 180–186, Berlin, Heidelberg, 2010. Springer. ISBN 978-3-642-15892-6. doi : 10.1007/978-3-642-15892-6_19.

I. Karamouzas, P. Heil, P. van Beek, and M. H. Overmars. A Predictive Collision Avoidance Model for Pedestrian Simulation. In A. Egges, R. Geraerts, and M. Overmars, editors, *Motion in Games*, Lecture Notes in Computer Science, pages 41–52, Berlin, Heidelberg, 2009. Springer. ISBN 978-3-642-10347-6. doi : 10.1007/978-3-642-10347-6_4.

I. Karamouzas, B. Skinner, and S. J. Guy. Universal Power Law Governing Pedestrian Interactions. *Physical Review Letters*, 113(23) :238701, Dec. 2014. doi : 10.1103/PhysRevLett.113.238701. URL https://link.aps.org/doi/10.1103/PhysRevLett.113.238701. Publisher : American Physical Society.

W. Kerr and D. Spears. Robotic simulation of gases for a surveillance task. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2905–2910, Aug. 2005. doi : 10.1109/IROS.2005.1545429. ISSN : 2153-0866.

O. Khatib. Inertial properties in robotic manipulation : An object-level framework. *The international journal of robotics research*, 14(1) :19–36, 1995.

P. M. Kielar, D. H. Biedermann, and A. Borrmann. MomenTUMv2 : A modular, extensible, and generic agent-based pedestrian behavior simulation framework. Technical Report TUM-I1643, Technische Universität München, Institut für Informatik, 2016.

J. Kim, Y. Seol, T. Kwon, and J. Lee. Interactive manipulation of large-scale crowd animation. *ACM Transactions on Graphics*, 33(4) :83 :1–83 :10, July 2014. ISSN 0730-0301. doi : 10.1145/2601097.2601170. URL https://doi.org/10.1145/2601097.2601170.

M. Kim, Y. Hwang, K. Hyun, and J. Lee. Tiling motion patches. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '12, pages 117–126, Goslar, DEU, July 2012. Eurographics Association. ISBN 978-3-905674-37-8.

N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2149–2154. IEEE.

A. Konrad. Simulation of mobile robots with unity and ros : A case-study and a comparison with gazebo, 2019.

L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 1–10, New York, NY, USA, Aug. 2008. Association for Computing Machinery. ISBN 978-1-4503-7845-1. doi : 10.1145/1401132.1401202. URL https://doi.org/10.1145/1401132.1401202.

C. Krogmeier, C. Mousas, and D. Whittinghill. Human–virtual character interaction : Toward understanding the influence of haptic feedback. *Computer Animation and Virtual Worlds*, 30(3-4) :e1883, 2019.

D. M. Krum, S.-H. Kang, and T. Phan. Influences on the elicitation of interpersonal space with virtual humans. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 223–9. IEEE, 2018.

T. Kruse, P. Basili, S. Glasauer, and A. Kirsch. Legible robot navigation in the proximity of moving humans. In *2012 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 83–88, Munich, May 2012. IEEE. ISBN

978-1-4673-0482-5 978-1-4673-0481-8 978-1-4673-0480-1. doi : 10.1109/ARSO.2012.6213404. URL http://ieeexplore.ieee.org/document/6213404/.

T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch. Human-aware robot navigation : A survey. *Robotics and Autonomous Systems*, 61(12) :1726–1743, 2013.

T. Kruse, A. Kirsch, H. Khambhaita, and R. Alami. Evaluating directional cost models in navigation. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, HRI '14, pages 350–357, Bielefeld, Germany, Mar. 2014. Association for Computing Machinery. ISBN 978-1-4503-2658-2. doi : 10.1145/2559636.2559662. URL https://doi.org/10.1145/2559636.2559662.

T. Kwon, K. H. Lee, J. Lee, and S. Takahashi. Group motion editing. *ACM Transactions on Graphics*, 27(3) :1–8, Aug. 2008. ISSN 0730-0301. doi : 10.1145/1360612.1360679. URL https://doi.org/10.1145/1360612.1360679.

Y.-C. Lai, S. Chenney, and S. Fan. Group motion graphs. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '05, pages 281–290, New York, NY, USA, July 2005. Association for Computing Machinery. ISBN 978-1-59593-198-6. doi : 10.1145/1073368.1073409. URL https://doi.org/10.1145/1073368.1073409.

S. M. LaValle and J. J. Kuffner. Randomized Kinodynamic Planning. *The International Journal of Robotics Research*, 20(5) :378–400, May 2001. ISSN 0278-3649. doi : 10.1177/02783640122067453. URL https://doi.org/10.1177/02783640122067453. Publisher : SAGE Publications Ltd STM.

G. Le Lann. *Cyberphysical Constructs and Concepts for Fully Automated Networked Vehicles*. PhD thesis, INRIA Paris-Rocquencourt, 2019.

K. H. Lee, M. G. Choi, Q. Hong, and J. Lee. Group behavior from video : a data-driven approach to crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '07, pages 109–118, Goslar, DEU, Aug. 2007. Eurographics Association. ISBN 978-1-59593-624-0.

A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by Example. *Computer Graphics Forum*, 26(3) :655–664, 2007. ISSN 1467-8659. doi : https://doi.org/10.1111/j.1467-8659.2007.01089.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2007.01089.x.

_eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2007.01089.x.

R. Li, M. v. Almkerk, S. v. Waveren, E. Carter, and I. Leite. Comparing Human-Robot Proxemics Between Virtual Reality and the Real World. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 431–439, Mar. 2019. doi : 10.1109/HRI.2019.8673116. ISSN : 2167-2148.

H. W. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American statistical Association*, 62(318) : 399–402, 1967.

J. Lin, X. Guo, J. Shao, C. Jiang, Y. Zhu, and S.-C. Zhu. A virtual reality platform for dynamic human-scene interaction. In *SIGGRAPH ASIA 2016 Virtual Reality meets Physical Reality : Modelling and Simulating Virtual Humans and Environments*, SA '16, pages 1–4, New York, NY, USA, Nov. 2016. Association for Computing Machinery. ISBN 978-1-4503-4548-4. doi : 10.1145/2992138.2992144. URL https://doi.org/10.1145/2992138.2992144.

P. Long, T. Fanl, X. Liao, W. Liu, H. Zhang, and J. Pan. Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6252–6259, Brisbane, QLD, May 2018. IEEE. ISBN 978-1-5386-3081-5. doi : 10.1109/ICRA.2018.8461113. URL https://ieeexplore.ieee.org/document/8461113/.

C. Louison, F. Ferlay, and D. R. Mestre. Spatialized vibrotactile feedback improves goal-directed movements in cluttered virtual environments. *International Journal of Human–Computer Interaction*, 34(11) :1015–1031, 2018.

S. D. Lynch, R. Kulpa, L. A. Meerhoff, J. Pettre, A. Cretual, and A.-H. Olivier. Collision avoidance behavior between walkers : global and local motion cues. *IEEE transactions on visualization and computer graphics*, 24(7) :2078–2088, 2017.

A. López, F. Chaumette, E. Marchand, and J. Pettré. Attracted by light : vision-based steering virtual characters among dark and light obstacles. In *Motion, Interaction and Games*, MIG '19, pages 1–6, New York, NY, USA, Oct. 2019a. Association for Computing Machinery. ISBN 978-1-4503-6994-7. doi : 10.1145/3359566.3360085. URL https://doi.org/10.1145/3359566.3360085.

A. López, F. Chaumette, E. Marchand, and J. Pettré. Character navigation in dynamic environments based on optical flow. *Computer Graphics Forum*, 38(2) : 181–192, 2019b. ISSN 1467-8659. doi : https://doi.org/10.1111/cgf.13629. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13629. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13629.

S. Massengale, D. Folden, P. McConnell, L. Stratton, and V. Whitehead. Effect of visual perception, visual function, cognition, and personality on power wheelchair use in adults. *Assistive Technology*, 17(2) :108–121, 2005.

C. Mavrogiannis, A. M. Hutchinson, J. Macdonald, P. Alves-Oliveira, and R. A. Knepper. Effects of distinct robot navigation strategies on human behavior in a crowded environment. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 421–430. IEEE, 2019.

D. R. Mestre, C. Louison, and F. Ferlay. The contribution of a virtual self and vibrotactile feedback to walking through virtual apertures. In *International Conference on Human-Computer Interaction*, pages 222–232. Springer, 2016.

O. Michel. Cyberbotics Ltd. Webots^{TM} : Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*, 1(1) :5, Mar. 2004. ISSN 1729-8814. doi : 10.5772/5618. URL https://doi.org/10.5772/5618. Publisher : SAGE Publications.

W. B. Mortenson, L. Demers, M. J. Fuhrer, J. W. Jutai, J. Lenker, and F. DeRuyter. Effects of an assistive technology intervention on older adults with disabilities and their informal caregivers : an exploratory randomized control trial. *American journal of physical medicine & rehabilitation*, 92(4) :297, 2013.

M. Moussaïd, M. Kapadia, T. Thrash, R. W. Sumner, M. Gross, D. Helbing, and C. Hölscher. Crowd behaviour during high-stress evacuations in an immersive virtual environment. *Journal of The Royal Society Interface*, 13(122) :20160414, 2016.

M. Moussaïd, D. Helbing, and G. Theraulaz. How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, 108(17) :6884–6888, Apr. 2011. ISSN 0027-8424, 1091-6490. doi : 10.1073/pnas.1016507108. URL https://www.pnas.org/content/108/17/6884. Publisher : National Academy of Sciences Section : Social Sciences.

M. Moussaïd, M. Kapadia, T. Thrash, R. W. Sumner, M. Gross, D. Helbing, and C. Hölscher. Crowd behaviour during high-stress evacuations in an immersive virtual environment. *Journal of The Royal Society Interface*, 13 (122) :20160414, Sept. 2016. doi : 10.1098/rsif.2016.0414. URL https://royalsocietypublishing.org/doi/10.1098/rsif.2016.0414. Publisher : Royal Society.

R. Narain, A. Golas, S. Curtis, and M. C. Lin. Aggregate dynamics for dense crowd simulation. *ACM Transactions on Graphics*, 28(5) :1–8, Dec. 2009. ISSN 0730-0301, 1557-7368. doi : 10.1145/1618452.1618468. URL https://dl.acm.org/doi/10.1145/1618452.1618468.

V. K. Narayanan, A. Spalanzani, F. Pasteau, and M. Babel. On equitably approaching and joining a group of interacting humans. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4071–4077. IEEE, 2015.

V. K. Narayanan, A. Spalanzani, R. C. Luo, and M. Babel. Analysis of an adaptive strategy for equitably approaching and joining human interactions. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 341–346, Aug. 2016. doi : 10.1109/ROMAN.2016.7745153. ISSN : 1944-9437.

A. H. Olivier, J. Bruneau, R. Kulpa, and J. Pettré. Walking with virtual people : Evaluation of locomotion interfaces in dynamic environments. *IEEE Transactions on Visualization and Computer Graphics*, PP(99) :1–1, 2017a. ISSN 1077-2626. doi : 10.1109/TVCG.2017.2714665.

A.-H. Olivier, J. Bruneau, R. Kulpa, and J. Pettré. Walking with virtual people : Evaluation of locomotion interfaces in dynamic environments. *IEEE transactions on visualization and computer graphics*, 24(7) :2251–2263, 2017b.

J. Ondřej, J. Pettré, A.-H. Olivier, and S. Donikian. A synthetic-vision based steering approach for crowd simulation. *ACM Transactions on Graphics*, 29 (4) :123 :1–123 :9, July 2010. ISSN 0730-0301. doi : 10.1145/1778765.1778860. URL https://doi.org/10.1145/1778765.1778860.

A. K. Pandey and R. Gelin. A mass-produced sociable humanoid robot : Pepper : The first machine of its kind. *IEEE Robotics & Automation Magazine*, PP :1–1, 07 2018a. doi : 10.1109/MRA.2018.2833157.

A. K. Pandey and R. Gelin. A Mass-Produced Sociable Humanoid Robot : Pepper : The First Machine of Its Kind. *IEEE Robotics Automation Magazine*, 25 (3) :40–48, Sept. 2018b. ISSN 1558-223X. doi : 10.1109/MRA.2018.2833157. Conference Name : IEEE Robotics Automation Magazine.

S. Paris, J. Pettré, and S. Donikian. Pedestrian Reactive Navigation for Crowd Simulation : a Predictive Approach. *Computer Graphics Forum*, 26(3) :665–674, 2007. ISSN 1467-8659. doi : https://doi.org/10.1111/j.1467-8659.2007.01090.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2007.01090.x. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2007.01090.x.

S. Patil, J. v. d. Berg, S. Curtis, M. C. Lin, and D. Manocha. Directing Crowd Simulations Using Navigation Fields. *IEEE Transactions on Visualization and Computer Graphics*, 17(2) :244–254, Feb. 2011. ISSN 1941-0506. doi : 10.1109/TVCG.2010.33. Conference Name : IEEE Transactions on Visualization and Computer Graphics.

A. E. Patla. Understanding the roles of vision in the control of human locomotion. *Gait & Posture*, 5(1) :54–69, 1997.

R. Paulin, T. Fraichard, and P. Reignier. Human-Robot Motion : Taking Human Attention into Account. page 1, Oct. 2018. URL https://hal.inria.fr/hal-01856016.

S. Pellegrini, J. Gall, L. Sigal, and L. Van Gool. Destination Flow for Crowd Simulation. In A. Fusiello, V. Murino, and R. Cucchiara, editors, *Computer Vision – ECCV 2012. Workshops and Demonstrations*, Lecture Notes in Computer Science, pages 162–171, Berlin, Heidelberg, 2012. Springer. ISBN 978-3-642-33885-4. doi : 10.1007/978-3-642-33885-4_17.

J. Pettré, J. Ondřej, A.-H. Olivier, A. Cretual, and S. Donikian. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, pages 189–198, New York, NY, USA, Aug. 2009. Association for Computing Machinery. ISBN 978-1-60558-610-6. doi : 10.1145/1599470.1599495. URL https://doi.org/10.1145/1599470.1599495.

L. C. A. Pimenta, N. Michael, R. C. Mesquita, G. A. S. Pereira, and V. Kumar. Control of swarms based on Hydrodynamic models. In *2008 IEEE International*

Conference on Robotics and Automation, pages 1948–1953, May 2008. doi : 10.1109/ROBOT.2008.4543492. ISSN : 1050-4729.

L. Rebenitsch and C. Owen. Review on cybersickness in applications and visual displays. *Virtual Reality*, 20(2) :101–125, June 2016. ISSN 1434-9957. doi : 10.1007/s10055-016-0285-9. URL https://doi.org/10.1007/s10055-016-0285-9.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once : Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi : 10.1109/CVPR.2016.91. URL http://ieeexplore.ieee.org/document/7780460/.

C. W. Reynolds. Flocks, herds and schools : A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.

C. W. Reynolds. Steering behaviors for autonomous characters. In *Game developers conference*, volume 1999, pages 763–782. Citeseer, 1999.

K. Rio and W. H. Warren. The visual coupling between neighbors in real and virtual crowds. *Transportation Research Procedia*, 2 :132–140, 2014.

K. W. Rio, G. C. Dachner, and W. H. Warren. Local interactions underlying collective motion in human crowds. *Proceedings of the Royal Society B : Biological Sciences*, 285(1878) :20180611, 2018.

E. Rohmer, S. P. N. Singh, and M. Freese. V-REP : A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326, Nov. 2013. doi : 10.1109/IROS.2013.6696520. ISSN : 2153-0866.

R. Rossi, M. P. Polverini, A. M. Zanchettin, and P. Rocco. A pre-collision control strategy for human-robot interaction based on dissipated energy in potential inelastic impacts. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 26–31. IEEE, 2015.

D. Roth and M. E. Latoschik. Construction of a validated virtual embodiment questionnaire. *arXiv preprint arXiv :1911.10176*, 2019.

J. P. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal. 3d normal distributions transform occupancy maps : An efficient representation for mapping in dynamic environments. *The International Journal of Robotics Research*, 32 (14) :1627–1644, 2013.

S. Scheggi, M. Aggravi, and D. Prattichizzo. Cooperative navigation for mixed human–robot teams using haptic feedback. *IEEE Transactions on Human-Machine Systems*, 47(4) :462–473, 2016.

M. Schepers, M. Giuberti, and G. Bellusci. Xsens mvn : Consistent tracking of human motion using inertial sensing. 03 2018. doi : 10.13140/RG.2.2.22099.07205.

M. A. Sheik-Nainar and D. B. Kaber. The Utility of a Virtual Reality Locomotion Interface for Studying Gait Behavior :. *Human Factors*, July 2016. doi : 10.1518/001872007X215773. URL https://journals.sagepub.com/doi/10.1518/001872007X215773. Publisher : SAGE PublicationsSage CA : Los Angeles, CA.

M. Shimizu, A. Ishiguro, T. Kawakatsu, Y. Masubuchi, and M. Doi. Coherent swarming from local interaction by exploiting molecular dynamics and stokesian dynamics methods. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 2, pages 1614–1619 vol.2, Oct. 2003. doi : 10.1109/IROS.2003.1248875.

R. Siegwart, K. O. Arras, S. Bouabdallah, D. Burnier, G. Froidevaux, X. Greppin, B. Jensen, A. Lorotte, L. Mayor, M. Meisser, R. Philippsen, R. Piguet, G. Ramel, G. Terrien, and N. Tomatis. Robox at Expo.02 : A large-scale installation of personal robots. *Robotics and Autonomous Systems*, 42(3) :203–222, Mar. 2003. ISSN 0921-8890. doi : 10.1016/S0921-8890(02)00376-7. URL http://www.sciencedirect.com/science/article/pii/S0921889002003767.

S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman. SteerBench : A benchmark suite for evaluating steering behaviors. *Computer Animation and Virtual Worlds*, 20 :533–548, 2009.

M. Slater. A note on presence terminology. *Presence connect*, 3(3) :1–5, 2003.

M. Slater and S. Wilbur. A framework for immersive virtual environments (five) : Speculations on the role of presence in virtual environments. *Presence : Teleoperators & Virtual Environments*, 6(6) :603–616, 1997.

T. J. Sørensen. *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons*, volume 5. Munksgaard Copenhagen, 1948.

A. Staranowicz and G. L. Mariottini. A survey and comparison of commercial and open-source robotic simulator software. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '11, pages 1–8, New York, NY, USA, May 2011. Association for Computing Machinery. ISBN 978-1-4503-0772-7. doi : 10.1145/2141622.2141689. URL https://doi.org/10.1145/2141622.2141689.

M. J. Tarr and W. H. Warren. Virtual reality in behavioral neuroscience and beyond. *Nature Neuroscience*, 5(11) :1089–1092, Nov. 2002. ISSN 1546-1726. doi : 10.1038/nn948. URL https://www.nature.com/articles/nn948. Number : 11 Publisher : Nature Publishing Group.

S. Thrun. Robotic mapping : A survey. Technical report, 2002.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Aug. 2005. ISBN 978-0-262-30380-4. Google-Books-ID : wjM3AgAAQBAJ.

S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. v. Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley : The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9) : 661–692, 2006. ISSN 1556-4967. doi : https://doi.org/10.1002/rob.20147. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20147. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20147.

P. Trautman and A. Krause. Unfreezing the robot : Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 797–803. IEEE, 2010.

P. Trautman, J. Ma, R. M. Murray, and A. Krause. Robot navigation in dense human crowds : the case for cooperation. In *2013 IEEE International Conference on Robotics and Automation*, pages 2153–2160, Karlsruhe, Germany, May 2013. IEEE. ISBN 978-1-4673-5643-5 978-1-4673-5641-1. doi : 10.1109/ICRA.2013.6630866. URL http://ieeexplore.ieee.org/document/6630866/.

P. Trautman, J. Ma, R. M. Murray, and A. Krause. Robot navigation in dense human crowds : Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research*, 34(3) :335–356, 2015.

A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. *ACM Transactions on Graphics*, 25(3) :1160–1168, July 2006. ISSN 0730-0301. doi : 10.1145/1141911.1142008. URL https://doi.org/10.1145/1141911.1142008.

M. Usoh, E. Catena, S. Arman, and M. Slater. Using presence questionnaires in reality. *Presence : Teleoperators & Virtual Environments*, 9(5) :497–503, 2000.

J. van den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-Body Collision Avoidance. In B. Siciliano, O. Khatib, F. Groen, C. Pradalier, R. Siegwart, and G. Hirzinger, editors, *Robotics Research*, volume 70, pages 3–19. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011a. ISBN 978-3-642-19456-6 978-3-642-19457-3. doi : 10.1007/978-3-642-19457-3_1. URL http://link.springer.com/10.1007/978-3-642-19457-3_1. Series Title : Springer Tracts in Advanced Robotics.

J. van den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-Body Collision Avoidance. In C. Pradalier, R. Siegwart, and G. Hirzinger, editors, *Robotics Research*, Springer Tracts in Advanced Robotics, pages 3–19, Berlin, Heidelberg, 2011b. Springer. ISBN 978-3-642-19457-3. doi : 10.1007/978-3-642-19457-3_1.

W. van Toll, N. Jaklin, and R. Geraerts. Towards believable crowds : A generic multi-level framework for agent navigation. In *ASCI.OPEN*, 2015.

M. Vasic and A. Billard. Safety issues in human-robot interactions. In *2013 IEEE International Conference on Robotics and Automation*, pages 197–204, May 2013. doi : 10.1109/ICRA.2013.6630576. ISSN : 1050-4729.

D. Vasquez, B. Okal, and K. O. Arras. Inverse Reinforcement Learning algorithms and features for robot navigation in crowds : An experimental comparison. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1341–1346, Sept. 2014. doi : 10.1109/IROS.2014.6942731. ISSN : 2153-0866.

J. Wainer, D. J. Feil-Seifer, D. A. Shell, and M. J. Mataric. Embodiment and Human-Robot Interaction : A Task-Based Perspective. In *RO-MAN 2007 - The*

16th IEEE International Symposium on Robot and Human Interactive Communication, pages 872–877, Aug. 2007. doi : 10.1109/ROMAN.2007.4415207. ISSN : 1944-9437.

W. G. Walter. An imitation of life. *Scientific american*, 182(5) :42–45, 1950.

W. Warren. Visually controlled locomotion : 40 years later. *Ecological Psychology*, 10(3-4) :177–219, 1998.

W. H. Warren. The dynamics of perception and action. *Psychological Review*, 113(2) :358–389, 2006. ISSN 1939-1471(Electronic),0033-295X(Print). doi : 10.1037/0033-295X.113.2.358. Place : US Publisher : American Psychological Association.

W. H. Warren and B. R. Fajen. Behavioral Dynamics of Visually Guided Locomotion. In A. Fuchs and V. K. Jirsa, editors, *Coordination : Neural, Behavioral and Social Dynamics*, Understanding Complex Systems, pages 45–75. Springer, Berlin, Heidelberg, 2008. ISBN 978-3-540-74479-5. doi : 10.1007/978-3-540-74479-5_3. URL https://doi.org/10.1007/978-3-540-74479-5_3.

W. H. Warren, B. A. Kay, W. D. Zosh, A. P. Duchon, and S. Sahuc. Optic flow is used to control human walking. *Nature Neuroscience*, 4(2) :213–216, Feb. 2001. ISSN 1546-1726. doi : 10.1038/84054. URL https://www.nature.com/articles/nn0201_213. Number : 2 Publisher : Nature Publishing Group.

K. Wilmut and A. L. Barnett. Locomotor adjustments when navigating through apertures. *Human Movement Science*, 29(2) :289–298, 2010.

D. A. Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

B. Yersin, J. Maïm, J. Pettré, and D. Thalmann. Crowd patches : populating large-scale virtual environments for real-time applications. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, I3D '09, pages 207–214, New York, NY, USA, Feb. 2009. Association for Computing Machinery. ISBN 978-1-60558-429-4. doi : 10.1145/1507149.1507184. URL https://doi.org/10.1145/1507149.1507184.

F. Zanlungo, T. Ikeda, and T. Kanda. Social force model with explicit collision prediction. *EPL (Europhysics Letters)*, 93(6) :68005, Mar. 2011. ISSN 0295-5075. doi : 10.1209/0295-5075/93/68005. URL https://doi.org/10.1209%2F0295-5075%2F93%2F68005. Publisher : IOP Publishing.

M. Zhao, S. J. Turner, and W. Cai. A Data-Driven Crowd Simulation Model Based on Clustering and Classification. In *2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*, pages 125–134, Oct. 2013. doi : 10.1109/DS-RT.2013.21. ISSN : 1550-6525.

X. Zhao, L. Xia, J. Zhang, and W. Song. Artificial neural network based modeling on unidirectional and bidirectional pedestrian flow at straight corridors. *Physica A : Statistical Mechanics and its Applications*, 547 :123825, June 2020. ISSN 0378-4371. doi : 10.1016/j.physa.2019.123825. URL http://www.sciencedirect.com/science/article/pii/S0378437119321272.

J. Zhong, W. Cai, L. Luo, and M. Zhao. Learning behavior patterns from video for agent-based crowd modeling and simulation. *Autonomous Agents and Multi-Agent Systems*, 30(5) :990–1019, Sept. 2016. ISSN 1573-7454. doi : 10.1007/s10458-016-9334-8. URL https://doi.org/10.1007/s10458-016-9334-8.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Simulation de foule et expérimentations pour l'évaluation de la navigation de robots dans la foule

Mots clés : Simulation, robotique mobile, foule, Réalité virtuelle, Interaction robot-foule

Résumé : Les robots mobiles investissent de plus en plus la place publique. Cela nécessite des outils de développement et d'évaluation adaptés au problème de la navigation de robots mobiles dans la foule. Notre approche du problème consiste à utiliser la simulation : nous avons créé le premier outil dédié à la robotique simulant une foule pouvant montrer des comportement très variés. Notre méthode de simulation de foule s'appuie sur un principe unificateur permettant d'imiter des algorithmes différents et de créer des algorithmes de simulation de foule hybride. Nous avons ensuite imaginé des métriques pour évaluer la capacité des robots mobiles à naviguer sans risque et de manière efficace dans une foule dense. Le simulateur de foule et les métriques ont été fusionnées avec un simulateur de robot pour créer un banc d'essai créant ainsi un outil innovant à la communauté. La simulation peut cependant souffrir d'un manque de réalisme. En particulier lors d'une interaction proche entre humain et robot, les algorithmes de foule peuvent se comporter de façon aberrante. Nous avons donc imaginé un outil de réalité virtuelle permettant de plonger un humain et un robot dans la simulation, conservant ainsi le réalisme de l'interaction tout en virtualisant les collisions. Nous avons ensuite mené une experience metant en pratique cette idée. Nous avons ensuite cherché à améliorer l'immersion de nos participants dans la simulation en ajoutant des dispositifs de retour haptique, et avons montré l'interêt de tels outils à travers une experience consitant à faire naviguer des participants dans une foule dense simulée. Nous avons également participé au développement d'une plateforme mécanique permettant la simulation d'un fauteuil roulant électrique grâce à une synchronisation des retours visuels, haptiques et vestibulaires. En conclusion, nous avons imaginé et implémenté les outils permettant l'évaluation en simulation de systèmes robotiques naviguant dans la foule et nous avons montré que l'usage de la réalité virtuelle s'avère judicieux pour l'étude d'interactions proches.

Title : Crowd simulation and experiments for the evaluation of robot navigation in a crowd

Keywords: Simulation, mobile robotics, crowd, virtual reality, crowd-robot interaction

Abstract : Mobile robots are increasingly taking over the public square. This requires development and evaluation tools adapted to the problem of navigating mobile robots in a crowd. Our approach to the problem is to use simulation: we have created the first tool dedicated to robotics that simulates a crowd that can show a wide range of behaviors. Our crowd simulation method is based on a unifying principle that allows us to mimic different algorithms and to create hybrid crowd simulation algorithms. We then devised metrics to evaluate the ability of mobile robots to safely and efficiently navigate through a dense crowd. The crowd simulator and metrics were merged with a robot simulator to create a test bed creating an innovative tool for the community. The simulation may however suffer from a lack of realism. In particular, during close human-robot interaction, crowd-sourced algorithms can behave in aberrant ways. We therefore imagined a virtual reality tool that allows to immerse a human and a robot in the simulation, thus preserving the realism of the interaction while virtualizing the collisions. We then conducted an experiment to put this idea into practice. We then sought to improve the immersion of our participants in the simulation by adding haptic feedback devices, and showed the interest of such tools through an experiment consisting in making participants navigate in a dense simulated crowd. We also participated to the development of a mechanical platform allowing the simulation of an electric wheelchair thanks to a synchronization of visual, haptic and vestibular feedbacks. In conclusion, we have designed and implemented tools allowing the evaluation of robotic systems navigating in a crowd and we have shown that the use of virtual reality is judicious for the study of close interactions.