Système de caméras intelligentes pour l'étude en temps-réel de personnes en mouvement

Andres Burbano

NNT : 2018SACLS139

Système de caméras intelligentes pour l'étude en temps-réel de personnes en mouvement

Thèse de doctorat de l'Université Paris-Saclay
Préparée à Paris-Sud

École doctorale n°580 : Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : robotique

Thèse présentée et soutenue à Orsay, le 6 juin, par

Andres BURBANO

Composition du Jury :

Saïda BOUAKAZ Professeur, Université Claude Bernard Lyon 1 – LIRIS	Président
Kuros MADANI Professeur, Université Paris-Est Créteil – LISSI	Rapporteur
Julien DUBOIS Maître de conférences HDR, Université de Bourgogne – Le2i	Rapporteur
Samia BOUCHAFA-BRUNEAU Professeur, Université d'Evry-Val-d'Essonne – IBISC	Examinateur
Miguel Octavio ARIAS ESTRADA Maître de conférences, INAOE	Examinateur
Samir BOUAZIZ Professeur, Université Paris-Sud – SATIE	Directeur de thèse
Marius VASILIU Maître de conférences, Université Paris-Sud – SATIE	co-encadrant
Thomas NENNER Docteur, Directeur de Shopline Electronic	Invité

Thèse de doctorat

Remerciements

Ce travail de thèse s'achève enfin, et avec du recul, je prends conscience que beaucoup de monde m'a soutenu et apporté son aide.

Je tiens à remercier les rapporteurs, Messieurs Kuroch Madani, Professeur à l'Université Paris Est Créteil et Julien Dubois, Maître de conférences HDR à l'Université de Bourgogne, pour avoir accepté d'évaluer mon travail de thèse, avec leurs expertises scientifiques.

J'exprime également tous mes remerciements à Mesdames Saïda Bouakaz, Professeure à l'Université Claude Bernard Lyon 1 et à Samia Bouchafa-Bruneau, Professeure à l'Université d'Evry-Val d'Essonne, qui me font l'honneur d'être dans mon jury de thèse pour examiner le travail accompli.

Je remercie Monsieur Miguel Octavio Arias Estrada, Investigador Titular à l'Instituto Nacional de Astrofísica, Óptica y Electrónica, pour avoir accepté de faire partie du jury de thèse.

Je tiens à remercier Messieurs Samir Bouaziz, directeur de thèse et Marius Vasiliu, co-encadrant, pour leur patience et leur suivi scientifique tout au long de cette thèse, qui m'a formé pour mon avenir.

Je pense à mes collègues du SATIE (secrétariat, doctorants et chercheurs) qui ont apporté aide et soutien au moment où j'en avais besoin. Des remerciements en particulier à Madame Sylvie Le-Hégarat et Messieurs Sergio A. Rodriguez, Emanuel Aldea, Roger Reynaud, et la liste est longue. Ils m'ont toujours été de précieux conseils.

Je tiens à saluer particulièrement mes compagnons de bureau Marie Lachaize, Salim Zahir et Romain Saussard, maintenant docteurs du laboratoire SATIE, dont j'ai apprécié les qualités humaines et leur bienveillance à mon égard.

Je remercie Monsieur Gilles Duc, directeur de l'école doctorale pour sa compréhension et sa disponibilité.

Mes plus sincères remerciements vont à Madame Irene Nenner pour sa générosité, son soutien et son assistance inestimable tout au long de la rédaction de mes travaux, en particulier le maniement des mots en langue française.

Bien sûr, mes remerciements vont aussi à toute l'équipe de l'entreprise Shopline Electronic et plus particulièrement, son directeur Monsieur Thomas Nenner, pour toute la confiance qu'il m'a accordée tout au long de cette thèse. Son soutien et sa gentillesse ont été indéfectibles.

Je termine mes remerciements par l'équipe rapprochée familiale et amis, en particulier : Clara Sánchez pour son aide sur les graphiques et les images, ainsi Laura Guzmán pour ses explications mathématiques.

Merci à Amy Wright, Claudia Santana et Gabriel Ramirez pour leurs relectures des articles en anglais et à Hugo Kassimatis, Juan Manuel Hernández et Fredy Martínez pour le français.

Je pense également à toutes les personnes avec lesquelles j'ai eu l'opportunité de collaborer au cours de ces dernières années et qui m'ont tant apporté.

Je tiens à remercier ma famille, ma mère et mon père, qui m'ont toujours soutenu dans tous les domaines ainsi que ma belle-famille pour leurs encouragements constants.

Enfin et surtout, je conclue cette page de remerciement par un grand merci à Lou Courjan-Kompf qui a m'a autant supporté que soutenu, lorsque j'en avais le plus besoin.

—Dedicada a Pedro López, Pedro Carrión y a todos los que aportaron a este proyecto de vida—

Résumé

Dans ce travail de thèse, nous avons exploré des voies technologiques et scientifiques afin d'obtenir des informations fiables et viables (industriellement) pour étudier le comportement de personnes en mouvement dans les grands espaces. La solution proposée se compose d'un réseau de caméras intelligentes en position zénithale avec la caractéristique que sa puissance de calcul se trouve dans des nœuds distribués. De plus, notre système est facile à déployer, a configurer et peut être industrialisé à très faible coût.

Notre travail est divisé en 4 parties principales. Les travaux ont débuté par l'évaluation de l'influence de la position de la caméra sur l'observation de la scène, en mettant en évidence l'importance de son placement en position zénithale pour réduire les occultations et diminuer les variations d'échelle. Ensuite, on a caractérisé les performances des caméras 3D disponibles sur le marché par des méthodes adaptées pour mésuser la stabilité temporelle des cartes de profondeur acquises, la précision de la distance mesurée, la résolution de la profondeur et la fiabilité de la détection des personnes de chaque capteur. Comme résultat, nous avons opté pour le capteur optique actif ASUS Xtion Pro, qui constitue un bon compromis entre les différentes caractéristiques évaluées.

En second lieu, nous avons conçu et réalisé une caméra intelligente autonome, capable d'extraire des propriétés spatio-temporelles et physiques des personnes en produisant des données riches permettant l'identification et le suivi de plusieurs cibles en temps-réel. L'autonomie de cette caméra est assurée par l'intégration des chaînes de traitements (hors ligne et en ligne) et une conception hardware adaptée à une architecture nœuds distribués. Le traitement hors ligne a permis de reconstruire l'arrière-plan, pour permettre la séparation des personnes du fond de la scène et le filtrage des cibles non désirées (enfants ou caddies). Le traitement en ligne mis en place assure la séparation entre personnes (segmentation à deux niveaux), puis la similitude des gens et la déformation du modèle (vecteur de caractéristiques humaines). Enfin, nous avons évalué nos algorithmes de détection et suivi et mesuré ses performances. Notre solution garantit l'exécution des traitements en temps-réel (20 fps), suffisamment réactive pour détecter des déplacements rapides avec une précision jusqu'à 99 %.

En troisième lieu, nous avons créé un réseau de caméras pour étudier le comportement de personnes en mouvement sur de grands espaces, en assurant une collecte massive de données provenant de plusieurs sources. Ce réseau a une architecture distribuée et est composé de plusieurs nœuds intelligents qui apportent de la puissance de calcul et qui étendent la région globale d'observation. Nous avons utilisé un système de calibration extrinsèque pour créer un repère unique global et un système d'étiquetage centralisé pour gérer les personnes qui transitent d'une caméra à une autre, permettant l'extension à de grands espaces.

En quatrième lieu, nous avons étudié le comportement des personnes en mouvement en utilisant des trajectoires récupérées par notre système et d'autres capables de suivre des personnes dans des grands espaces. Nous avons proposé des méthodes pour la détection de personnes en temps-réel dans les zones d'intérêt, la génération des cartes d'occupation, d'entrées et de sorties en fonction de l'utilisation de l'espace. Pour l'analyse de trajectoires, nous avons segmenté de flux comportementaux et représenté dynamiquement les trajectoires.

Nous avons également mis en évidence les verrous scientifiques à lever et confronté notre solution à la réalité du terrain, en termes de faisabilité, de coût de conception, de complexité d'utilisation et de maintenabilité. Nous sommes arrivés à une solution viable techniquement et économiquement, avec une simplicité de mise en œuvre sur le terrain.

Abstract

In this thesis, we explored technological and scientific pathways in order to obtain reliable and viable information (industrially) to study the behavior of people in motion in wide-open spaces. The proposed solution consists of a network of smart cameras in overhead position with the characteristic of a computing power that lies in the distributed nodes. In addition, our system is easy to deploy, to configure and can be industrialized at a very low cost.

Our work is divided into four main parts. First, our work began by assessing the influence of the position of the camera on the observation of the scene, highlighting the importance of its placement in overhead position to reduce the occlusion and decrease the scale variations. Then, the performance of the 3D cameras available on the market was characterized by methods adapted to measure the temporal stability of the acquired depth maps, the accuracy of the measured distance, the depth resolution and the people detection reliability from each sensor. As a result, we opted for the active optical sensor ASUS Xtion Pro, which represents an appropriate compromise between the different evaluated characteristics.

Secondly, we have designed and built an autonomous smart camera, capable of extracting people's spatiotemporal and physical properties by producing rich data, enabling the identification and tracking of several targets in real time. The autonomy of this camera is ensured by the integration of processing chains (offline and online), and a hardware design adapted to a distributed nodes architecture. The Offline processing allowed to rebuild the background, conduct people separation from the background and to filter unwanted targets (children or caddies). The online processing ensures separation between people (two-level segmentation), then the similarity of people and the deformation of the model (human feature descriptor). Finally, we assessed our detection and tracking algorithms performances. Our solution ensures a performance with a throughput of up to 20 frames per second, sufficiently responsive to detect fast movements with a precision of up to 99 %. Thirdly, we have created a network of cameras to study the behavior of people in motion on large spaces, ensuring a massive collection of data from several sources. This network has a distributed architecture and is composed of several smart nodes that bring computational power and extend the overall region of observation. We used an extrinsic calibration system to create a single global coordinate system and a labeling centralized system to manage people transiting from one camera to another, allowing the extension to large spaces.

Fourth, we studied people in motion behavior using trajectories recovered by our and other systems, capable of tracking people in large spaces. We have proposed methods for people detection in real-time in zones of interest, as well as occupancy, points of entry and exit maps generation considering people's use of space. For trajectory analysis, we used behavioral flow segmentation and trajectories dynamic representation.

We have also highlighted the scientific challenges to overcome, and have confronted our solution to the reality on the field, in terms of feasibility, cost of design, usability, and maintainability. We arrived at a technically and economically viable solution that is simple to install.

Table des matières

Liste des figures

Liste des tableaux

Chapitre 1

Introduction

Sommaire

Le développement du secteur de la vente dans les magasins s'est traduit, depuis plusieurs décennies, par le besoin d'améliorer les performances commerciales. L'une des informations pertinentes est le taux de fréquentation des magasins. Pour cela, des systèmes de comptage des clients entrants et sortants ont vu le jour dans les espaces commerciaux. Cette information (fréquentation dans la journée) a permis de fournir aux managers commerciaux de précieuses données pour optimiser les espaces et la visibilité des produits. Ceci permet par exemple de quantifier la valeur marchande des emplacements commerciaux. Enfin, ces informations, une fois comparées aux ventes effectives, permettent de mesurer les performances des points de vente, de mesurer l'efficacité des vendeurs et d'optimiser l'organisation de leurs journées de promotion. Les systèmes de comptage ont apporté une solution certes facile à déployer, mais entachée d'erreurs de mesures. Ces erreurs peuvent être liées à la configuration du lieu, aux conditions de mesure ou bien à l'incapacité du système à discerner des situations (deux personnes qui franchissent le détecteur en même temps, par exemple).

Compte tenu de l'évolution des tendances de consommation dans la société, ce simple mode de comptage s'avère insuffisant. Il devient essentiel d'améliorer la précision de ce comptage indépendamment de la configuration du lieu et de l'environnement (ensoleillement ou autre source de lumière, hauteur sous plafond, etc), de faire ces mesures en temps réel, de savoir distinguer les adultes (les acheteurs potentiels) des enfants et, dans le futur, d'identifier les individus afin de pouvoir mesurer le temps de séjour dans l'espace commercial[1] et leurs manières de se déplacer (d'un présentoir à un autre), toujours dans le but de convertir ces visites en ventes.

Le marché du comptage évolue donc vers la nécessité de mesurer le comportement des personnes dans leur environnement avec une plus grande précision. Ces marchés sont aujourd'hui globaux, c'est-à-dire qu'ils se trouvent aussi bien dans les pays développés que dans les pays émergents, ce qui implique de maîtriser les coûts des systèmes et de leur déploiement. Ce défi constitue un verrou technologique et scientifique qui a intéressé de nombreux chercheurs et industriels pour imaginer des systèmes de comptages performants et précis tout en restant dans une réalité industrielle, c'est-à-dire avec un coût maîtrisé et une simplicité de mise en œuvre sur les lieux de comptage. Il est en effet difficile d'imaginer des systèmes et des solutions sans prendre en compte cette réalité industrielle forte.

Aujourd'hui, les besoins marketing évoluent rapidement et réclament des données plus riches pour prédire le comportement des consommateurs. Il y a un véritable manque de solutions pour obtenir des informations fiables sur l'intention du client, son regard, le temps passé face à un produit et l'utilisation de l'espace dans les magasins et face aux vitrines. La nécessité d'une mesure précise et généralisée de ces indicateurs permettrait donc l'évaluation de l'efficacité d'une campagne de publicité et l'amélioration du service à la clientèle dans les centres commerciaux, les entrepôts et les magasins. Enfin, l'objectif est de maximiser l'attention des clients sur les produits choisis et d'évaluer la fidélité de la clientèle. Le rôle de la détection humaine est très important mais doit s'intégrer à d'autres données relevant du domaine commercial, comme les achats des clients et les données relatives au programme de fidélité mis en place par le magasin.

Au cours des deux dernières décennies, de nombreux chercheurs du monde académique en lien étroit avec l'industrie ont étudié la façon de « suivre » les personnes dans leur environnement naturel avec une variété de capteurs. Nous notons alors l'émergence des premiers travaux de suivi des personnes [RB94, Seg96] utilisant des caméras analogiques couleur pour suivre les gens en temps-réel, ainsi que la création de groupes de recherche ou d'instituts dédiés à ce domaine. Le groupe *Human Sensing Lab* [web16b] de l'université de Carnegie Mellon (USA) a développé des outils informatiques pour modéliser et comprendre le comportement humain à partir de données sensorielles comme la vidéo, la capture de mouvement et l'audio. De la même façon, le groupe Institut international de recherche sur les télécommunications avancées (***A**dvanced **T**elecommunications **R**esearch Institute International* : ATR) [web16a] au Japon a développé un environnement de suivi des personnes dans le centre commercial "ATC". Ces groupes de recherche et ces études ont proposé des solutions automatiques pour suivre et détecter les personnes, permettant une collecte massive de données provenant de plusieurs sources et les convertissant en données exploitables.

[1] Le terme anglais utilisé dans le marketing est « store dwell time ».

Depuis une dizaine d'années, le développement d'une nouvelle génération de capteurs dits à 3 dimensions ou caméras de profondeur ainsi que les réseaux de ces caméras a été marqué par trois percées technologiques importantes. En premier lieu, l'explosion de l'utilisation (et la prise de conscience de l'importance [HS06]) des systèmes embarqués pour la conception des caméras intelligentes entre 2006-2008. Dans le domaine des « caméras intelligentes distribuées » [BV14], on passe de l'utilisation de caméras classiques comme sources d'images et séquences vidéos à l'utilisation d'un système plus complexe composé d'un capteur, d'un processeur embarqué et d'une interface de communication [BDM+, RWS+08]. Ces composants permettent à une caméra intelligente de fournir plus d'informations qu'une simple image, telles que des événements, par exemple la présence d'une personne dans l'image ou le sens de son passage. Une caractéristique très importante d'une caméra intelligente est sa capacité de reprogrammation pour s'adapter à l'application visée et celle de partager les événements et l'information extraite sur un réseau de communication [RW08]. En deuxième lieu, l'apparition de la technologie de PrimeSense 2010 (comme la Kinect et la Xtion, capteurs 3D qui utilisent de la lumière structurée pour construire une carte de profondeur). Dans les années 2014-2015, ces deux éléments ont permis la maturation du concept de réseaux des caméras intelligentes pour le suivi de personnes. La troisième percée technologique consiste en la démocratisation des systèmes embarqués avec l'unité centrale Raspberry PI (et homologues) dès 2013.

Parmi les travaux récents, celui de Teixeira et al [TDS10], montre la définition d'un ensemble de propriétés observables de personnes pour étudier leur comportement en mouvement. C'est sur ce point que le présent travail de recherche s'est focalisé. La figure 1.1 montre les trois groupes de propriétés qui peuvent être extraites, à savoir les observables spatio-temporelles, les observables du comportement et les observables physiques. Ces observables sont hiérarchiquement liées en interne, ce qui signifie qu'une propriété donnée implique la connaissance de la propriété antérieure. Par exemple, avant de pouvoir suivre une personne, nous avons besoin de connaître sa localisation initiale ainsi que les positions suivantes lors de ses déplacements.

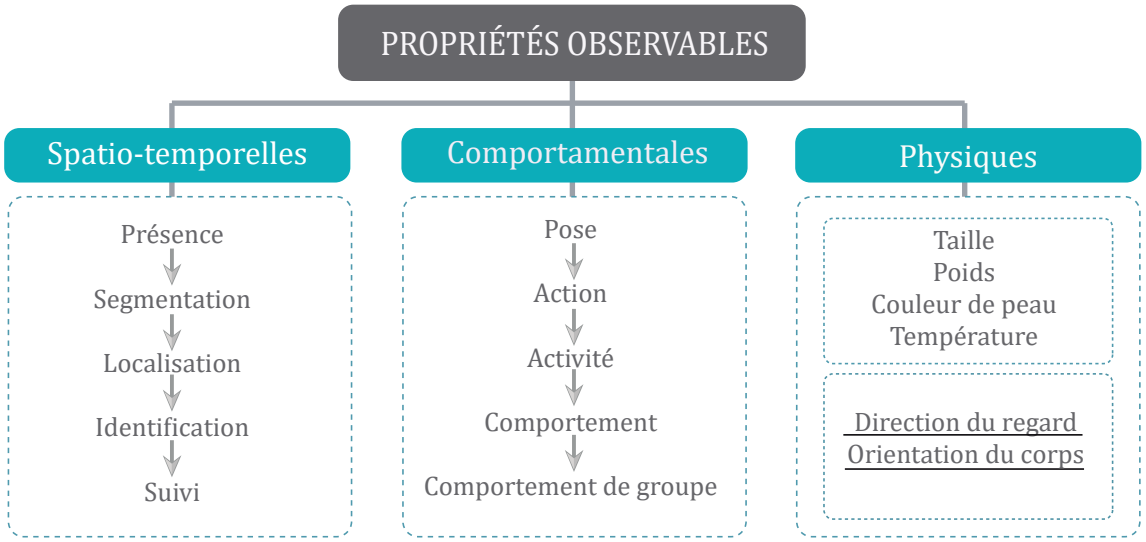


FIGURE 1.1 – Différentes catégories de propriétés observables de personnes en mouvement.

Dans le même temps, il existe des relations entre les groupes de propriétés. Par exemple, afin d'identifier une personne, nous avons besoin de propriétés physiques pour déterminer qui est cette personne. Nous allons définir et développer les relations entre ces groupes de propriétés tout au long de ce mémoire de thèse.

Pour ce qui est des *propriétés spatio-temporelles*, Teixeira précise que le suivi des personnes signifie la capacité de détecter la *présence* d'une personne, de compter les personnes présentes (*segmentation*), de les *localiser*, d'extraire la signature (*identification* dans le système) de chaque personne et de les suivre. Ceci se traduit par des questions concrètes :

- Y a-t-il une personne dans un lieu donné (présence/absence) ?
- Combien de personnes s'y trouvent, à un instant donné ?
- Où se trouve chaque personne (localisation par rapport à la scène ou par rapport aux autres) ?
- Qui est chaque personne observée (classe, catégorie, identité) ?
- Que fait chaque personne ? Quelle est la trajectoire de la personne ?

Nous pouvons considérer les *propriétés comportementales* comme une chaîne de connaissances où la propriété précédente aide à définir la suivante, comme une construction sémantique avec une extension temporelle. Par exemple, une personne qui est assise (pose), soulevant sa fourchette (action) lors d'un déjeuner (activité). Cette personne a pour habitude de déjeuner chaque jour entre 12h et 13h (comportement), avec ses collègues (comportement de groupe).

Les *propriétés physiques* sont les caractéristiques des personnes (traits) que nous pouvons observer et qui nous permettent d'identifier et de différencier les gens entre eux. Ces propriétés peuvent être statiques, dynamiques ou externes. Par exemple, une propriété physique statique est la taille de la personne, une propriété dynamique est la direction du regard et une propriété externe est le signal d'identification par radiofréquence (***R**adio **F**requency **ID**entification* : RFID) émis par un éventuel dispositif porté par cette personne.

Afin d'observer ces propriétés, nous avons un ensemble de difficultés à surmonter et de principes sur lesquels s'appuyer pour concevoir un système évolutif et omniprésent. On note notamment des travaux similaires au nôtre comme celui de [SBR14] qui analyse les déplacements humains, celui de [BKIM13] concernant le suivi des personnes et celui de [Che14] pour ce qui est de la reconnaissance de l'activité humaine. Ces travaux ont pour but de développer des applications portant sur l'étude du comportement humain.

Sur le plan scientifique, de nouveaux sujets ont émergé et sont actuellement étudiés par la communauté [DBC14, web16e], en réponse au manque de solutions pour évaluer l'impact d'une campagne de publicité ou pour améliorer l'organisation de la distribution des rayons dans les magasins. Ce sujet fait l'objet de réflexions et de recherches pour mesurer l'attention des clients sur les produits et leurs caractéristiques physiques. Par conséquent, les recherches actuelles tentent d'utiliser des techniques de vision numérique pour caractériser l'orientation du regard des visiteurs, leur temps de séjour et leur utilisation de l'espace. Ceci doit se traduire par des cartes dites "d'attention maximale" (en anglais « heat maps ») montrant les zones qui retiennent le plus longtemps l'attention des visiteurs dans un espace commercial.

1.1 Autres applications de la détection des personnes

Le fait de détecter des personnes, de les compter, de les localiser, de les identifier représente un objectif qui a largement dépassé les besoins du secteur du commerce. L'arrivée de capteurs 3D pour des marchés grand public (jeux, téléphonie mobile, drones, domotique, surveillance, médecine, etc.) bouleverse le paysage et affecte le marché du comptage des personnes. En effet, du fait que ces nouveaux usages touchent le grand public, le marché des caméras de profondeur par stéréovision (active ou passive) explose et il s'en suit l'arrivée de nouveaux acteurs sur le marché et une concurrence commerciale sur les prix. Si les objectifs des usages dans les jeux et dans le comptage sont différents, les techniques d'analyse du comportement des personnes sont voisines. Il en résulte une synergie importante des technologies de ces caméras avec le domaine du commerce de détail.

Les drones s'installent de plus en plus dans la vie quotidienne car les usages se multiplient, couvrant le domaine professionnel (prises d'images dans des endroits inaccessibles), les loisirs ou la livraison et la logistique à faible coût. On retrouve ces drones dans d'autres domaines

d'application, comme l'agriculture pour inspecter les parcelles et maîtriser les besoins en eau et le traitement des sols. Enfin, ces systèmes couvrent les domaines du bâtiment, les travaux publics, le ferroviaire, ainsi que la sécurité, y compris les applications militaires. Ces drones utilisent tous des systèmes de vision pour de multiples usages avec des exigences de plus en plus élevées en termes de performance.

En matière de médecine [web17e], la recherche s'est intéressée à l'assistance à la chirurgie [RBP15], à la prévention des chutes [RAR+11] et à la rééducation des patients [LCS+11], domaines pour lesquels l'utilisation de caméras est davantage nécessaire.

L'usage des caméras trouve des applications dans différents domaines où l'information vidéo va permettre d'extraire une information concernant la présence des humains, leurs intentions, leurs comportements etc. C'est ainsi que les bâtiments intelligents et les villes intelligentes [Kit14, NP11] utilisent des caméras dans plusieurs domaines d'application : surveillance des biens et des personnes, prévention des risques, fluidification de la circulation, etc.

Enfin, le développement de la voiture autonome est dépendant de systèmes de vision stéréoscopique, à l'intérieur comme à l'extérieur. Ceux-ci doivent être capables de détecter des obstacles à 360°, à différentes distances et à une cadence élevée afin de garantir la sécurité des passagers et des piétons.

Les jeux vidéo utilisent des technologies à base de caméras pour créer une interaction plus riche entre un joueur et le programme. On s'affranchit alors des manettes et des boutons au niveau des interfaces. Un simple mouvement de la main permet de créer une interaction. Pour cela, les dispositifs de vision stéréoscopique ont été largement utilisés comme la caméra Kinect dans les jeux Microsoft. Cette caméra fabriquée par la société israélienne « PrimeSense » a été exploitée par les éditeurs de comptage commerciaux comme Delopt [web17a] ainsi que par l'entreprise dans laquelle cette thèse est effectuée (Shopline Electronic).

1.2 Contexte industriel de la thèse

Cette thèse a été réalisée dans le cadre d'une collaboration entre l'entreprise Shopline Electronic et le laboratoire SATIE MOSS de l'Université Paris Sud, dans le cadre des **C**onventions **I**ndustrielles de **F**ormation par la **RE**cherche (CIFRE), financées par le gouvernement français. Ce partenariat public-privé permet d'encourager les transferts de technologies de l'innovation dans les **P**etites et les **M**oyennes **E**ntreprises (PME). Par ailleurs, la confrontation à des problématiques sociétales implique une recherche proche des réalités des usages et des usagers. Le monde de la recherche y trouve aussi un intérêt car on aborde ces problèmes en mettant en évidence les verrous à lever, tout en confrontant les solutions à la réalité du terrain, en termes de faisabilité, de coût de conception, de complexité d'utilisation et de maintenance, etc. Ces paramètres constituent autant de contraintes pour arriver à une solution viable techniquement et économiquement, et acceptable dans le cas de l'application de Shopline Electronic, le comptage de personnes.

Shopline Electronic est une PME française qui a comme cœur de métier un panel de solutions de comptage de personnes dans des magasins, des centres commerciaux, des musées, offices de tourisme et autres lieux publics. Son marché, d'abord européen, se développe au delà du cadre continental : Afrique du Nord, Moyen Orient, Chine. Une filiale de Shopline Electronic a par ailleurs été implantée en Inde, pays à forte croissance.

L'évolution du marché est à la fois quantitative et qualitative. Dans les pays développés, le comptage des clients est généralisé et le marché est en demande de produits plus performants et plus robustes dans leur utilisation. Dans les pays émergents, le marché du comptage est naissant avec des exigences difficiles : haute technologie tout en maintenant un prix acceptable.

Les systèmes simples à base de cellules infrarouges latérales sont maintenant considérés comme une offre peu sophistiquée et l'utilisation de caméras de comptage se généralise dans tous les lieux publics.

L'entreprise Shopline Electronic a réagi à la nécessité de proposer des systèmes à base de caméras pour le comptage avec l'importation des caméras stéréoscopiques Brickstream [web16d]. La valeur ajoutée de l'entreprise repose sur l'intégration de ces caméras dans ses logiciels d'analyse statistique. Deux modèles de caméras ont été utilisés : une caméra mono-objectif et une caméra stéréoscopique. La caméra Brickstream 2D a été déployée sur des sites commerciaux mais a révélé de graves défauts sur les sites de centre-ville comme les magasins et les offices de tourisme. En effet, ce capteur qui nécessite une lumière stable est perturbé par le soleil et fournit des résultats de comptage trop erronés, donc inexploitables. Par ailleurs, la caméra stéréoscopique Brickstream est coûteuse, ce qui ne permet pas de dégager des marges commerciales suffisantes. De plus, ces produits sont commercialement fragiles puisque les distributeurs de Shopline Electronic peuvent directement les importer, et donc devenir concurrents de la société.

Une des stratégies aurait été d'engager un projet de recherche et développement (R&D) pour la conception d'un système à partir de composants primaires comme les rétines interfacées avec des processeurs rapides et des architectures système sur une puce (***S**ystem **o**n a **C**hip* : SoC). Cette approche que l'on retrouve dans les industries grand public comme les smartphones ou l'automobile, nécessite des ressources de R&D importantes, très spécialisées dans le design de circuits ASIC, et n'est rentable que dans des projets où la production serait à grand volume. Cela ne correspond pas au segment commercial de Shopline Electronic.

La stratégie adoptée par l'entreprise repose alors sur l'utilisation de composants déjà existants, disponibles dans le commerce (caméra 2D, caméra 3D) et qu'elle va assembler autour d'une unité de calcul grand public et économique. Cette dernière est fortement liée à la complexité des algorithmes à implanter et à la puissance de calcul nécessaire pour garantir les contraintes temporelles imposées par l'application de détection.

En effet, la détection doit être suffisamment rapide pour observer les humains en mouvement. Pour cela, l'algorithme et l'unité de calcul devront être traités simultanément pour garantir l'exécution des traitements avec un débit d'au moins 15 images par secondes.

L'utilisation du capteur 2D est plus simple du point de vue des algorithmes mais nécessite des algorithmes robustes. Le capteur 3D sera choisi parmi les imageurs du commerce. Il est donc nécessaire d'évaluer la qualité de l'image de profondeur et la facilité de mise en œuvre de nos algorithmes sur la base des images fournies par ce capteur. La qualité de l'information finale permettra alors les analyses évoquées précédemment : suivi des personnes, temps de séjour et zones de haute fréquentation.

Le développement de produits propriétaires a donc été engagé par l'entreprise Shopline dès 2013, avec les contraintes inhérentes à tout projet d'ingénierie :

- Obtenir une précision suffisante et une ergonomie acceptable.
- Adopter une stratégie de R&D cohérente avec les méthodes de fabrication et la capacité technique de la société.

1.3 Difficultés à relever pour l'analyse d'images

Il est nécessaire de définir les différents facteurs (caractéristiques et qualificatifs) spécifiques à notre domaine [Che14, Che02, Dal06, DWSP12, Rus09, TDS10, WLY15], basés sur l'extraction des propriétés observables (section 1.1) des usagers en mouvement, à partir d'une information fournie par des imageurs optiques.

Parmi ces problèmes posés, on distingue des facteurs spatiaux comme l'occultation, les variations environnementales, la modélisation de l'arrière-plan, la séparation et la similitude des personnes. Pour ce qui est des facteurs temporels, on distingue la modélisation d'arrière-plan, la similitude des gens, la variation d'échelle des personnes dans la scène, la déformation du modèle du sujet et le mouvement rapide des personnes.

(a) **(b)**

FIGURE 1.2 – Exemple d'une image bruitée acquise dans une séquence vidéo.

La figure 1.2 illustre le bruit dans l'image d'une caméra couleur installée dans la cellule pour enregistrer le comportement humain. Les images a et b sont des images consécutives dans un flux vidéo. Cependant, nous pouvons observer quelques lignes de bruit produites par la chaîne d'acquisition. Ce qui montre qu'en milieu industriel, et du fait de la caméra à bas coût, on ne peut pas toujours garantir une image parfaite.

- **Bruits et autres défauts du capteur**
  Les signaux vidéo issus de chaque capteur sont affectés par des bruits liés à la technologie utilisée 1.2. Certains de ces bruits sont statiques ou connus et peuvent donc être annulés ou filtrés par des algorithmes spécifiques de traitement d'image. Cela a toutefois un impact sur la performance globale de la solution car le processus requiert du temps de calcul et, par conséquent, de la puissance de calcul. Par exemple, les caméras ont des bruits de détection comme la distorsion géométrique ou des problèmes de crènelage (ou effets d'escalier, connus sous le terme d'aliasing). Les caméras basées sur la technologie du temps de vol (***Time of Flight*** : ToF) ou à lumière structurée produisent des bruits supplémentaires induits par d'autres sources infrarouges ou par la lumière du soleil.

- **Occultation**
  L'effet d'occultation a lieu lorsque les sujets d'intérêt (par exemple, une personne, une voiture) sont cachés derrière un autre objet de premier plan, empêchant le capteur d'enregistrer une image complète de la cible. Cet effet impose aux algorithmes de pouvoir reconstruire le sujet d'intérêt le mieux possible à l'aide des informations partielles acquises et des informations acquises précédemment, par exemple des trames précédentes. L'occultation est générée par des objets dynamiques (chevauchement de foule, voitures, etc.) et des éléments statiques qui appartiennent à la scène (meubles, murs, autres éléments du bâtiment, etc.).

- **Variations environnementales**
  Les variations environnementales sont les changements inattendus comme les variations d'éclairage et les ombres. Celles-ci introduisent des erreurs dans l'interprétation de l'information. Par exemple, les ombres constituent l'un des problèmes les plus courants dans le suivi des personnes, altérant les systèmes de vision et fournissant des résultats erronés.

- **Modélisation d'arrière-plan**
  La modélisation d'arrière-plan consiste à différencier le signal qui appartient à l'arrière-plan du reste de l'information intéressante, acquise par le capteur. La mauvaise élimination du signal d'arrière-plan conduit à des informations superflues ou manquantes dans les objets d'intérêt, rendant la tâche d'extraction plus difficile. De plus, le calcul de modélisation du fond est généralement assez coûteux, ce qui diminue les performances globales.

- **Séparation des personnes**
  Après avoir séparé avec succès le fond du reste de l'image, le premier plan résultant peut contenir plusieurs personnes en contact ou très proches l'une de l'autre (la présence d'occultation rendant, dans certains cas, le problème encore plus complexe). La difficulté est alors de déterminer les frontières d'une personne et de les différencier de celles d'une autre personne.

- **Similitude des gens**
  Un autre défi à surmonter est celui de caractériser les paramètres pertinents lorsqu'il y a des personnes similaires et de pouvoir les différencier. De même, en combinant ce problème avec l'occultation et le bruit des capteurs (voir ci-dessus), le système risque de présenter des défauts tels qu'il sera incapable d'identifier ou/et classifier les personnes détectées.

- **Variation d'échelle relative**
  Ce problème survient lorsque le même objet dans différentes images a une taille qui diffère en fonction de sa distance par rapport au capteur. Ce problème augmente la complexité d'identification du même objet dans différentes images lorsque les objets se rapprochent ou s'éloignent de la caméra.

- **Déformation du modèle du sujet**
  La modélisation du corps humain est une tâche complexe en raison des parties du corps non rigides qui peuvent prendre des poses différentes. De plus, le placement de la caméra change l'apparence des parties du corps observées. Par exemple, une caméra placée en vue latérale voit les parties du corps en entier tandis qu'une caméra dans une vue visant le bas ne voit que la tête, les épaules, la poitrine et parfois les membres en fonction de la pose du sujet.

- **Déplacement rapide**

  Ce problème est lié à la vitesse de l'objet cible par rapport au temps d'exposition du capteur. Les déplacements imprévisibles des sujets (spécialement les humains) sur une scène conduisent à des ambiguïtés d'identification. Les algorithmes avancés prédisent la position future de l'objet en fonction de leur trajectoire. Cependant, les facteurs mentionnés (vitesse et taux d'acquisition) peuvent générer de fausses prédictions.

1.4 Objectifs pour la conception de notre système

Un système d'étude du comportement d'une personne en mouvement constitue un objectif qui représente un défi technologique, industriel et pose de ce fait des problématiques au niveau de la consistance des algorithmes qui vont s'exécuter sur des calculateurs taillés et limités en puissance de calcul (pour des raisons industrielles et commerciales). Ce qui suit est une liste non exhaustive des contraintes et caractéristiques exigées pour qu'un système soit viable industriellement et techniquement.

- **Précision**
  Le système doit être en mesure, à partir des données acquises, d'extraire des informations avec la plus grande exactitude afin de refléter la réalité de terrain. L'exactitude de l'information extraite dépend de la manière dont les suivis des personnes sont réalisés. Dans notre travail, la précision est liée principalement à l'emplacement, au comptage et au suivi de personnes.

- **Réactivité**
  Il s'agit de la capacité du système à extraire des informations à partir des images acquises à une fréquence élevée, qui répond aux exigences imposées en temps-réel. Dans un espace d'observation donné, le sujet doit être capturé (acquisition des images) un grand nombre de fois pour déterminer une trajectoire fiable. Il doit également fonctionner de manière homogène, même avec des changements soudains de l'environnement ou du nombre de personnes suivies. De la même façon, le système doit capturer, extraire et fournir des

informations comportementales personnelles en temps-réel au moins à 15 images par seconde (***F**rames **P**er **S**econd* : FPS) afin de suivre les gens à une vitesse d'environ 1,33 m/s, ce qui correspond à la vitesse moyenne d'un piéton de 4,8km/h [BBHK06]. La performance de la réactivité est limitée par le taux d'acquisition des capteurs et le temps de calcul requis par les algorithmes de traitement. Ces derniers doivent prendre en compte les conditions externes changeantes lors des acquisitions.

- **Détection non-intrusive**
  Il s'agit de la capacité du système à ne pas être intrusif sur les personnes détectées. Une détection intrusive exploite les appareils personnels (comme les smartphones) ou oblige les personnes cibles à porter des marqueurs sur les vêtements ou sur la peau. Ces derniers pourront servir à faciliter la détection et la fonction de suivi. Un exemple type est le « Motion capture », procédé où le mouvement d'objets ou de personnes est capté avec précision [Che02, MKS+16]. Il est très fréquent d'utiliser des marqueurs (par exemple réflecteurs détectés avec des caméras infrarouges) pour aider le système à déterminer la position du sujet. Cependant, ces marqueurs limitent parfois les mouvements normaux du sujet. Le caractère non-intrusif est une caractéristique fondamentale pour notre système, compte tenu du volume des mouvements d'humains sur les espaces publics et commerciaux.

- **Limites du domaine de vision**
  La zone de détection est une contrainte des sites d'observation. Elle s'applique à la hauteur sous plafond et à la largeur de détection. En pratique, ces contraintes imposent un choix de modèles du capteur ou obligent à changer de technologie. Par exemple, pour le comptage de personnes, la caméra est installée à une hauteur variante couramment entre 2,5 et 6 mètres. Par conséquent, la portée du système doit être comprise entre 50 cm et 4,50 m (en prenant en compte les personnes d'une hauteur entre 1,20 m et 2 m).

- **Prise en charge de plusieurs cibles**
  Le système doit être en mesure de suivre plusieurs cibles avec précision, quel que soit le nombre de personnes observées. Par conséquent, le système ne doit pas être limité par un nombre maximal de personnes à suivre et bien que la performance du système puisse varier, la fréquence minimale de suivi doit toujours être respectée.

- **Robustesse** Un système doit être très fiable, peu importe les conditions extérieures comme la température, la lumière, les bruits ou les interférences. La contrainte de robustesse est étroitement liée à celle de réactivité et de prise en charge de plusieurs cibles (voir ci-dessus).

- **Suivi des personnes sur de grands espaces**
  Le fait de suivre des personnes dans un grand espace public impose de disposer de plusieurs caméras couvrant chacune une partie de l'espace (avec un champ de vision limité). Le système doit permettre de diviser l'espace en sous-espaces, chacun étant couvert par une caméra. Le système doit être conçu pour générer les informations sur la trajectoire des personnes et leur comportement en minimisant et en distribuant la puissance de calcul.

- **Ubiquité**
  Il s'agit du fait d'être partout pour mesurer le comportement humain. Cette propriété est liée à l'évolutivité et au déploiement du système sous forme de nœuds "sous-système", chacun remplissant une fonction de détection et de suivi dans un espace limité. Tous les nœuds transmettent leurs informations vers un superviseur qui aura en charge de suivre les personnes détectées dans l'espace global surveillé par le système. Celui-ci doit aussi régler le problème des personnes qui changent d'espace de capture mais qui doivent garder le même identifiant comme étant la même personne tant qu'elle se trouve dans l'espace global sous surveillance.

Compte tenu de toutes ces caractéristiques qui sont autant de contraintes technologiques, industrielles et parfois scientifiques, il est important de synthétiser le but de cette thèse en une question clé : *Comment obtenir des informations fiables et exploitables pour compter et étudier le comportement de personnes en mouvement dans les grands espaces par une solution facile à déployer, configurable et à faible coût ?*

Dans le processus de détection des personnes dans un espace public, nous identifions trois principales étapes pour l'analyse et la visualisation de leur comportement en mouvement. Le processus commence par l'acquisition de données. La deuxième étape concerne le suivi des personnes. Enfin, la troisième étape vise l'analyse du comportement des personnes en mouvement, dans un grand espace. La figure 1.3 montre le schéma de ces trois étapes, leur imbrication et les caractéristiques extraites.

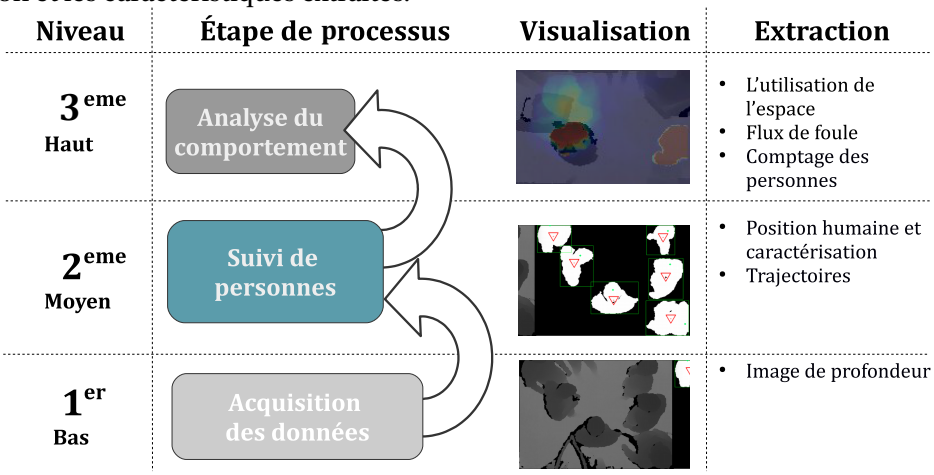


FIGURE 1.3 – Étapes du processus d'obtention du suivi des personnes et de leur comportement en mouvement. A chaque étape, les informations utiles sont extraites de l'analyse des images obtenues.

Les contraintes industrielles (à savoir le coût de fabrication, la facilité de déploiement et de configuration à distance), nous conduisent à utiliser une nouvelle famille de capteurs 3D couplés à une unité de calcul de puissance limitée, dans une configuration distribuée. L'utilisation de capteurs 3D présente des avantages dans le traitement d'images : séparation de l'arrière-plan plus facile et gestion plus aisée des informations géométriques de la scène. L'utilisation des capteurs en position zénithale permet également de réduire l'occultation ainsi que la déformation du sujet observé. Dans le même temps, cela permet une simplification des algorithmes et une réduction du temps de traitement requis pour le suivi des personnes en mouvement. Il a ensuite fallu concevoir une architecture distribuée de nœuds connectés pour pousser les calculs au plus proche des capteurs et transmettre les informations d'un niveau descriptif élevé et avec une taille réduite sur le réseau de communication qui centralise les données vers le superviseur. Ceci a l'avantage de mieux mapper les traitements des données avec la puissance de calcul disponible. C'est sur l'élaboration d'algorithmes plus ou moins complexes que le meilleur compromis de consistance et de facilité d'implantation a été obtenu, tout en gardant la capacité de maintenir le système à distance et à des coûts raisonnables. Tout cela a constitué un défi, pour nous permettre d'étudier le comportement des personnes en mouvement dans les grands espaces publics.

La suite du manuscrit se compose de 2 parties. La première partie, composée des chapitres 2 et 3, étudie la conception d'une caméra intelligente, pour détecter et suivre des personnes, en respectant les objectifs exposés. Dans le chapitre 2 qui concerne l'acquisition des données, nous présentons l'analyse des capteurs disponibles et le choix du positionnement de la caméra. Nous justifions également l'utilisation d'une caméra 3D et notre choix parmi les différents modèles disponibles sur le marché et qui répondent au mieux à nos objectifs. Dans le chapitre 3, nous proposons une méthode pour détecter et suivre des personnes en mouvement ainsi que des solutions pour construire une caméra intelligente. Nous présentons également l'adaptation de nos algorithmes au système que nous avons élaboré, avec la contrainte de consommation d'une puissance de calcul limitée. La deuxième partie, composée des chapitres 4 et 5, se focalise sur l'étude du comportement des personnes dans des grandes espaces. Dans le chapitre 4, nous définissons un système multi-caméras, en décrivant l'architecture du réseau, la calibration et la gestion des personnes entre caméras. Ensuite, nous présentons l'analyse du comportement des personnes en mouvement dans les espaces publics dans le chapitre 5. Dans ce dernier chapitre, nous évaluons l'utilisation de l'espace et les flux de la foule, en analysant l'information extraite et échangée par les algorithmes du suivi des personnes. Cette information est d'un haut niveau descriptif et elle est transmise à travers un protocole léger. Enfin, nous présentons nos conclusions et les perspectives de recherche et d'applications industrielles.

Première partie

Suivi des personnes

Chapitre 2

Acquisition des données

Sommaire

L'étude en temps-réel du comportement des personnes en mouvement dans leur environnement naturel impose de disposer d'un capteur capable de les observer. Dans ce chapitre, nous présentons un aperçu des différents capteurs utilisés pour le suivi des personnes avec une attention particulière sur les capteurs optiques 3D. Nous analyserons ensuite les différentes configurations, notamment les positions et orientations des caméras par rapport aux personnes filmées, dans le but de choisir la meilleure configuration pour la capture et le suivi. Enfin, nous comparerons les caméras 3D du marché pour évaluer la qualité des images 3D acquises dans une configuration vue de dessus, de manière à choisir celles qui seront les plus pertinentes pour le suivi des personnes et l'analyse de leur comportement.

2.1 Familles de capteurs

Afin d'obtenir les propriétés observables de personnes en mouvement, nous avons besoin d'informations provenant de capteurs parmi différentes familles, en termes de technologie et de méthodologie, pour détecter les personnes et leur environnement. Il existe cependant un nombre très important de dispositifs utilisés pour détecter ces propriétés. Par conséquent, nous avons choisi d'analyser seulement les trois familles de capteurs les plus couramment utilisés dans la littérature et liés à notre objectif en vue de concevoir un dispositif industriel. Il s'agit de capteurs optiques, non-optiques et de capteurs « portables » par les personnes filmées, comme illustré sur la figure 2.1. Cette figure donne une liste non exhaustive d'exemples de capteurs de différentes familles basés sur les travaux de la communauté scientifique [TDS10, DMZC12].

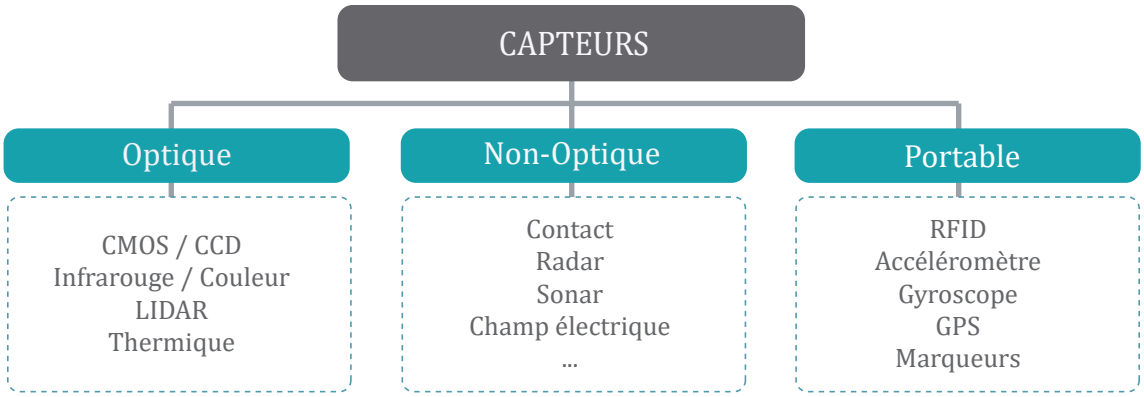


FIGURE 2.1 – Différentes classes de capteurs adaptés au suivi de personnes

Les capteurs optiques s'appuient sur la détection des signaux lumineux de différentes longueurs d'ondes (visibles et infra-rouge), émis par la personne et son environnement. Les capteurs non optiques s'appuient sur la détection d'autres types de signaux (acoustiques, ondes radio, etc.). Les capteurs portables « wearable » imposent d'être portés par la personne à suivre. Par exemple, une *smartwatch* ou un badge d'identité sur le lieu de travail.

Il s'agit maintenant d'évaluer comment les différents types de capteurs permettent d'atteindre les objectifs décrits dans l'introduction : précision, réactivité, détection non intrusive, limites de la portée de vision, prise en charge de plusieurs cibles, robustesse, suivi des personnes dans de grands espaces et ubiquité. Nous présentons les avantages et inconvénients de chacune de ces classes de capteurs pour apprécier leur capacité à obtenir les informations spatio-temporelles sur les personnes comme la présence, le comptage, la localisation, l'identification et leur suivi.

La catégorie des capteurs portables présente un grand inconvénient : la personne qui ne porte pas le dispositif requis ne peut pas être suivie par le système, en violation des principes de non-intrusion et d'ubiquité, vus dans l'introduction. Bien qu'aujourd'hui, les téléphones portables de type « smartphones » équipés de plusieurs capteurs, soient utilisés par une majorité de la population dans les villes, leur couverture (en termes de nombre de personnes qui en ont un) est trop incertaine pour créer un système de détection fiable [Ozc14]. Cependant, on peut noter que tout dispositif portatif pourrait améliorer l'extraction des propriétés spatio-temporelles, en

fusionnant les informations du système avec les informations du capteur portable. Par exemple, la RFID pourrait améliorer l'extraction de l'identification [TDS10], une centrale inertielle (***I**nertial **M**easurement **U**nit* : IMU) ou un ***G**lobal **P**ositioning **S**ystem* (GPS) contribuera de manière significative à la précision de la localisation des personnes. Il y a toutefois plusieurs aléas comme le fait que des personnes peuvent disposer de plusieurs smartphones, la précision ou fiabilité du GPS dans des lieux publics couverts (« indoors »), etc.

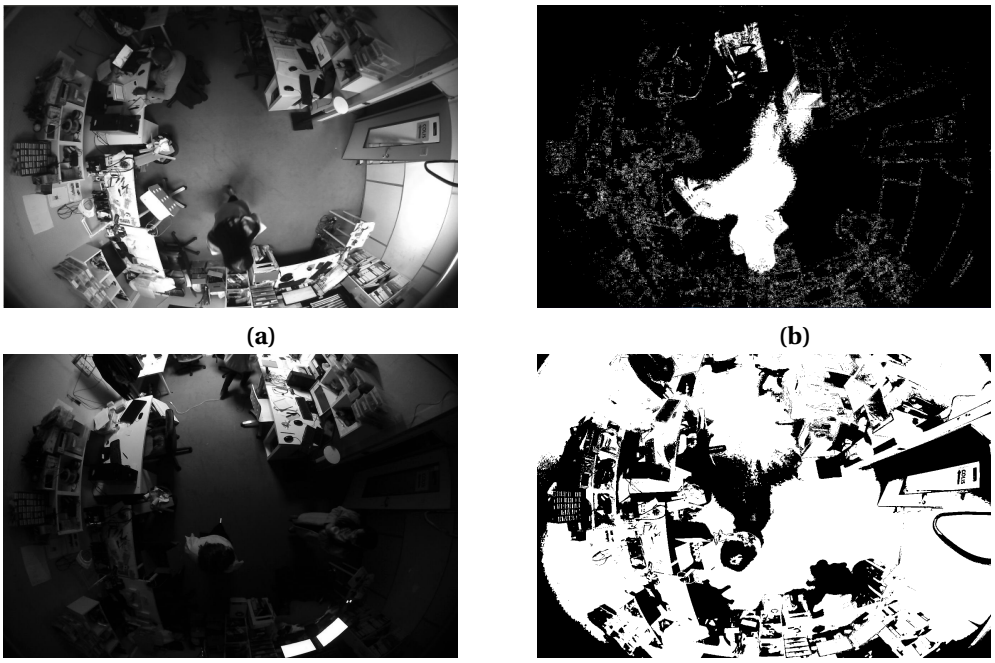
Dans le cas de la famille des capteurs non-optiques, on peut les classifier en deux groupes. Tout d'abord, les capteurs assez coûteux qui utilisent les ondes traversant la matière pour récupérer des informations hautement spécialisées. Par exemple, l'échographie (ondes acoustiques), les radiographies (rayons X) et la résonance magnétique nucléaire (détection de radiofréquences sur un objet soumis à un champ magnétique) qui sont restreints à un usage essentiellement médical et dont l'utilisation est interdite dans les espaces publics. Le deuxième groupe est constitué de capteurs non-optiques comme les radars et les sonars qui sont largement utilisés dans la localisation des objets. Cependant, le manque d'informations physiques sur les personnes ne permet pas de les utiliser pour caractériser les personnes cibles. Parmi les capteurs non-optiques, les capteurs de pression et de vibration ne permettent pas de donner des informations physiques assez fiables et ils sont compliqués à déployer et à entretenir (exemple : tapis matrice de capteur de pression au sol).

La catégorie des capteurs optiques est la plus utilisée pour détecter les personnes. On distingue, dans cette catégorie, les capteurs optiques 2D et 3D.

Les caméras 2D (comme les caméras couleur et infrarouge) présentent des inconvénients : il est difficile d'extraire l'information sur les personnes car il faut réaliser une modélisation précise de l'arrière-plan (les pixels de l'image qui appartient au fond de la scène), en éliminant les réflexions et différents ombrages des objets fixes et mobiles de la scène, ainsi que les réflexions parasites et les fortes variations d'éclairage ambiant. En particulier, la modélisation d'arrière-plan avec une caméra 2D est un processus coûteux en temps de calcul, car cette méthode implique de différencier le premier plan (les pixels qui portent l'information que on souhaite traiter) de l'arrière-plan dans une image, en actualisant constamment un arrière-plan virtuel pour le soustraire de chaque nouvelle image.

Les réflexions et les ombres des personnes conduisent à détecter des faux positifs qui n'existent pas dans le champ de vision de la caméra, comme représenté sur la figure 2.2b. De plus, les fortes variations d'éclairage ambiant, qui proviennent des réflexions de la lumière sur les vitrines, affectent l'arrière-plan global des images. Ces variations d'éclairage peuvent déstabiliser la détection du premier plan, ce qui aveugle le système jusqu'à ce que la variation de lumière soit absorbée par le fond virtuel (Fig. 2.2d). La résolution de ces problèmes implique des algorithmes complexes et un temps de calcul important [TDS10, WLY15] rendant les caméras 2D non viables pour une solution rapide de traitement à faible coût. Afin diminuer ces effets, Shopline a développé un système qui utilise une caméra active disposant de son propre éclairage stable infrarouge, tandis que la lumière extérieure ambiante est éliminée par un filtre optique. Même si ces effets ont diminué substantiellement, la précision du comptage reste encore très faible.

La figure 2.2 présente les problèmes de fausses détections d'ombrages et de changements d'illumination dans la scène comme des mouvements à l'intérieur de l'image. Cette figure est composée de couples d'images où la première image est l'image infrarouge (***Infra**Red* : IR) de la scène et la deuxième image est l'image binaire de détection de mouvement. Une image binaire représente en blanc les pixels où l'on détecte des mouvements et en noir les pixels qui appartiennent au fond de la scène. Dans le premier couple, on observe une personne qui se déplace. Cependant, dans l'image binaire on détecte non seulement le mouvement de la personne, mais aussi deux ombres différentes au sol. On détecte donc de faux mouvements dus aux ombres, en plus des vrais mouvements des personnes. Dans le deuxième couple, on observe une scène sombre qui vient de changer de luminosité mais dans laquelle tous les objets sont restés statiques. Dans l'image binaire, on observe que la majorité de la scène est interprétée comme mouvement.



(a) **(b)** **(c)** **(d)**

FIGURE 2.2 – Détection de mouvement à l'aide d'une caméra infrarouge. Les images a et c sont les images d'entrée fournies par le capteur IR. Les images b et d sont des images binaires où chaque pixel blanc représente le mouvement détecté.

Les caméras 3D représentent une catégorie de caméras permettant d'obtenir des images de profondeur, c'est-à-dire les mesures de distance des surfaces de la scène au capteur. Généralement, les approches d'acquisition de ces capteurs optique 3D sont telles qu'elles ne sont pas perturbées par les ombres des personnes ou les fortes variations d'éclairage. Le processus de modélisation du fond devient alors plus simple, impliquant des algorithmes moins complexes et un temps de calcul moins coûteux, ces conditions étant requises pour notre solution embarquée. Les conditions d'utilisation ne sont pas soumises à une réglementation trop contraignante. Ces technologies ne sont pas coûteuses à mettre en œuvre, à l'exception de la technique ***LA**ser **D**etection **A**nd **R**anging* (LADAR) / ***LI**ght **D**etection **A**nd **R**anging* (LIDAR) assimilée à une solution reposant sur les temps de vol. C'est pour ces raisons que les capteurs actifs de type optique sont souvent utilisés dans de nombreuses applications industrielles comme celles de la société Point Grey [web16d] o Delop [web17a], qui réalise des dispositifs similaires à ceux produits par l'entreprise Shopline Electronic.

Pour conclure, les capteurs les plus adéquats répondant aux exigences de notre cahier des charges (détecter et suivre des personnes dans un espace public) sont les caméras 3D, en raison des avantages qu'elles offrent du point de vue de l'exploitation sur le terrain, ainsi que par la simplicité relative des algorithmes (moins gourmands en calcul) pour détecter et suivre les gens. Nous présentons par conséquent une description plus détaillée des principes et méthodes de l'approche 3D.

2.2 Fondements de l'imagerie 3D

Dans cette section, on présente ce qu'est une image de profondeur, sa représentation en 3D, les deux grands principes d'acquisition des images de profondeur (la triangulation et le temps de vol) ainsi que les différentes méthodes d'acquisitions optiques.

2.2.1 Carte de profondeur

Les capteurs optiques actifs et passifs produisent des données images sous un format de matrices bidimensionnelles appelées «cartes, images ou matrices de profondeur» et qui représentent, plus ou moins fidèlement, pour chaque élément de la scène projetée sur la matrice de capteurs, la distance entre le capteur et le point acquis, comme représenté sur la figure 2.3.

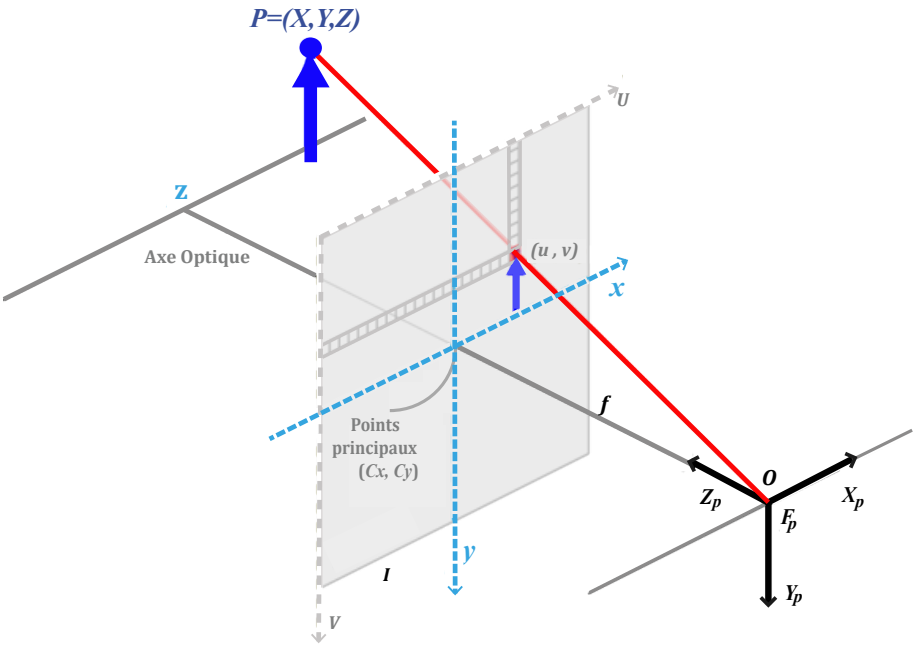
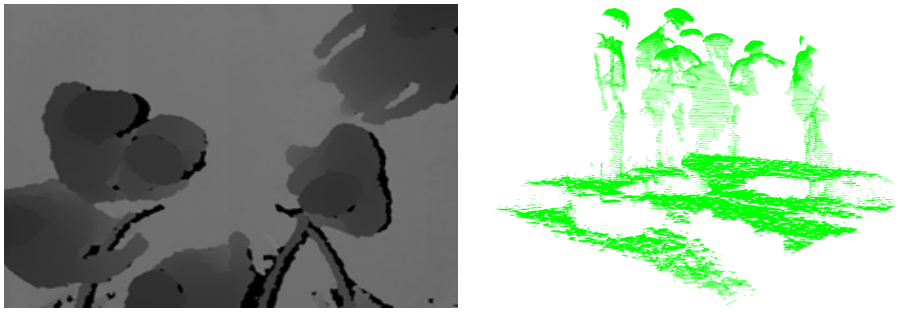


FIGURE 2.3 – Modèle de projection en perspective : un point de scène P projeté dans le pixel du capteur (u, v).

Sur l'illustration de la figure 2.3, la ligne en pointillés bleus représente le système de coordonnées S-2D. Les axes en noir représentent le système de référence 3D.

Ces matrices de dimension N par M pixels, dans lesquelles chaque pixel contient la distance entre le capteur et un point acquis de la scène, portent une information de profondeur. Optionnellement, cette donnée peut être accompagnée d'une information de couleur pour chaque pixel (ce qui correspond à une image 2D classique). L'ensemble de ces deux informations est appelé S-2D. Une carte de profondeur S-2D peut être traduite en une représentation 3D (de type nuage de points) par rétroprojection de perspective, comme le montre la figure 2.4.



(a) **(b)**

FIGURE 2.4 – a) une image de profondeur S-2D et b) sa représentation 3D type nuage de points vue de côté.

Cette projection utilise un modèle classique de caméra pin-hole avec ses paramètres intrinsèques. Soient *(X, Y, Z)* le système de coordonnées de la caméra placée à l'origine O et *(U, V)* le système de coordonnées d'image où U et V sont les axes du plan d'image I et $\mathrm{C} = (\mathrm{C}_x, \mathrm{C}_y)$ est le centre de l'image I (point principal) défini par l'intersection entre I et l'axe Z. Soit f la distance focale définie comme la distance entre O et le plan I. Dans la projection de perspective, chaque rayon de projection passant par l'origine O et reliant un point de l'image (x_p, y_p) avec un point de la scène $(\mathrm{X}_p, \mathrm{Y}_p, \mathrm{Z}_p)$ crée des triangles similaires suivant les axes X et Y du système de coordonnées (Fig. 2.5). Par conséquent, nous pouvons écrire la correspondance des côtés comme :

$$\frac{x_p}{f} = \frac{\mathrm{X}_p}{\mathrm{Z}_p} \quad \text{et} \quad \frac{y_p}{f} = \frac{\mathrm{Y}_p}{\mathrm{Z}_p}; \tag{2.1}$$

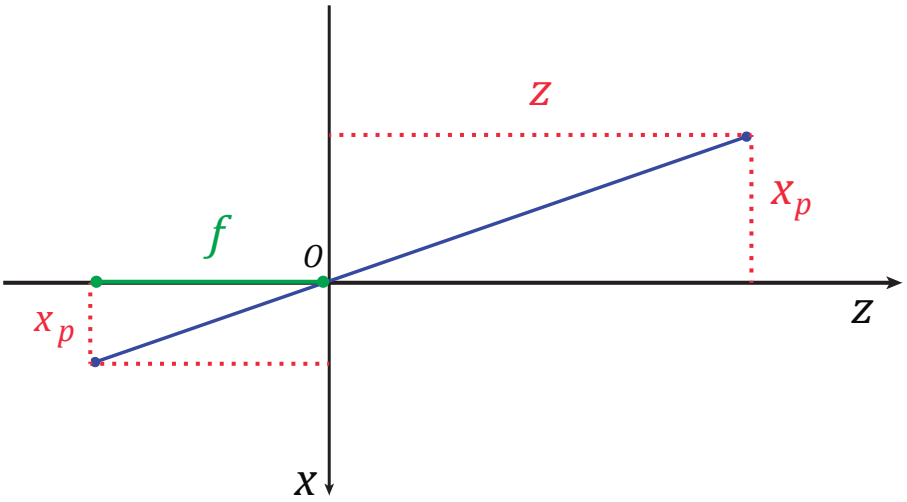


FIGURE 2.5 – Vue latérale du modèle pinhole.

On remarque que dans ces équations x_p, y_p, f_p, X_p, Y_p et Z_p sont exprimés en mètres. En réalité x_p et y_p ne peuvent être, physiquement, que des multiples entiers de la distance inter-pixelique. Il est donc utile de diviser x_p et f d'une part et y_p et f d'autre part par les distances inter-pixels respectivement horizontale et verticale. Après cette division et un décalage pour passer au système de coordonnées *(U,V)* de centre *C*, le pixel occupe une position *(u,v)* exprimée en pixels qui dépend des distances focales f_x et f_y exprimées aussi en pixels, comme dans les équations suivantes [HS97] :

$$u - C_x = X_p \frac{f_x}{Z_p} \qquad \text{et} \qquad v - C_y = Y_p \frac{f_y}{Z_p} \tag{2.2}$$

En raison du facteur de forme pas toujours carré des pixels du capteur, les distances inter-pixels horizontale et verticale ne sont pas toujours égales donc la caméra possède des distances focales f_x et f_y a priori différentes. En conclusion, une caméra pin-hole est caractérisée par les paramètres intrinsèques suivants : f_x, f_y, C_x, C_y. En utilisant les coordonnées homogènes, communément utilisées dans le domaine de la vision numérique [Sze10], les équations 2.2 peuvent s'écrire d'une manière plus compacte sous la forme matricielle suivante :

$$z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.3}$$

Ainsi, la projection de perspective nous aide à comprendre la relation entre une carte de profondeur et un nuage de points. Habituellement, les paramètres intrinsèques de cette projection sont souvent fournis par les bibliothèques logicielles accompagnant la caméra.

2.2.2 Principes d'imagerie 3D

Les principaux concepts de l'acquisition 3D sont le temps de vol ToF et la triangulation. Ces deux principes sont utilisés dans différents types de méthodes d'acquisition. Par exemple, le principe ToF est utilisé dans les radars, les sonars, les ultrasons et autres. Le principe de triangulation est utilisé dans la vision stéréo, la lumière structurée, etc.

Principe du temps de vol

Le principe du temps de vol ToF estime la distance radiale de la scène au capteur (Fig. 2.6). Un signal émis depuis l'émetteur vers la surface de la scène, et réfléchi vers le capteur, parcourt une

distance *2D* à une vitesse *c* qui dépend du type de signal (lumière, son, etc.) et arrive au récepteur après un délai ΔT :

$$D = \frac{c\Delta T}{2} \tag{2.4}$$

Ceci est une explication simple du principe de ToF qui est la base de la technologie des caméras 3D à temps de vol, mais qui peut aussi s'appliquer aux autres dispositifs optiques et non-optiques plus sophistiqués tels que les capteurs ***Laser Measurement System*** (LMS) ou LIDAR, Radars ou sonars.

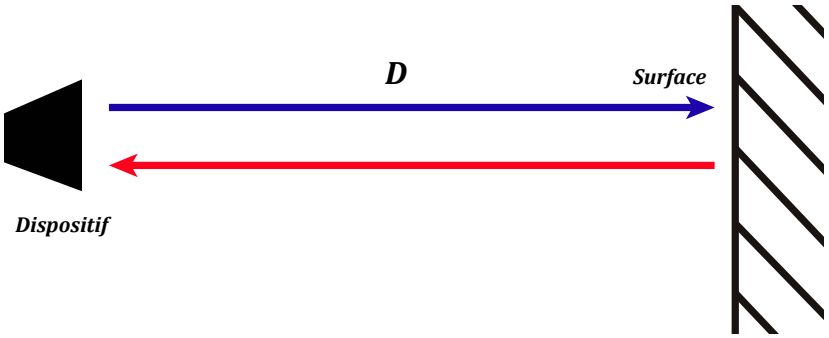


FIGURE 2.6 – Fonctionnement de la camera ToF.

Comme illustré dans la figure 2.6, le trajet du rayon aller et retour est le double de la distance entre la caméra et la surface.

Principe de la triangulation pour la vision 3D

L'approche de vision 3D par triangulation ou stéréovision utilise une paire d'images visualisant la même scène avec différents points de vue afin de récupérer l'information de profondeur, c'est-à-dire la distance entre la caméra et les sujets/objets de la scène. Cette méthode est plus simple à comprendre parce qu'elle utilise le même principe que la vision stéréoscopique de l'homme qui utilise une paire d'yeux séparés par une courte distance interoculaire, un chiasma optique et une projection corticale pour avoir une sensation de profondeur. Cela permet d'identifier les objets proches ou éloignés.

La relation de triangulation dans un système stéréo rectifié (Fig. 2.7) selon [Sze10] est obtenue par :

$$z = f\frac{b}{d} \tag{2.5}$$

où f représente la distance focale des deux caméras, b est la ligne de base, ou distance entre les capteurs, et d est le décalage relatif de la projection du même point dans les deux images (typiquement $U_L - U_R$) également appelée disparité. Ce modèle de triangulation est la base de l'estimation 3D.

La figure 2.7 présente le modèle de la géométrie d'un système stéréo rectifié vu de l'axe de *Y*. P_L et P_R sont les points 2D de la projection du point 3D P dans l'image gauche et droite respectivement, qui ont la même coordonnée verticale (sur l'axe y). Z_L et X_L représentent le système de référence de la caméra de droite et Z_R et X_R représentent celui de la caméra de gauche. Soient Z la distance entre le point et la caméra, f la distance focale, b la ligne de base, U_R et U_L les valeurs des axes x dans les systèmes de référence 2D respectifs. Ce diagramme se base sur l'étude de [ZMM+16].

De plus, cette figure montre que les deux axes optiques des caméras sont parallèles et situés à une distance b (ligne de base). Ce *système stéréo rectifié* est l'élément de base pour la reconstruction 3D qui nous permet de passer d'une carte de disparité à une carte de profondeur.

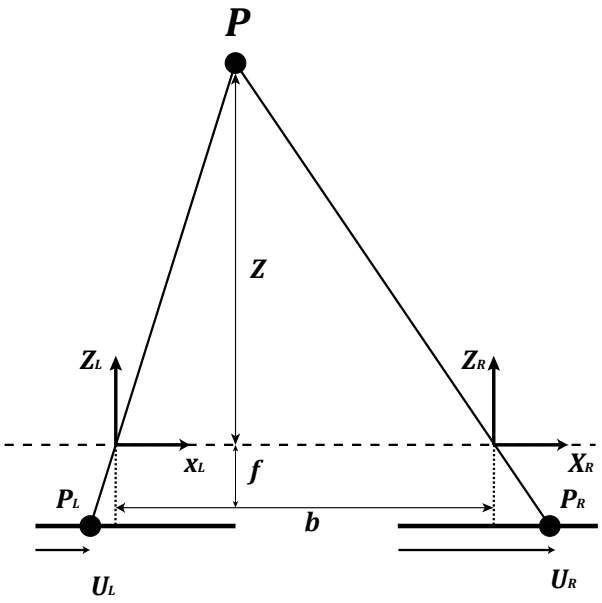


FIGURE 2.7 – Principe de la caméra par triangulation utilisant un système stéréo rectifié

2.3 Méthodes d'acquisition optiques d'imagerie 3D

Selon [DMZC12, TDS10] (Fig. 2.8), on distingue deux méthodes de détection optique, actives et passives :

- Les capteurs passifs utilisent la lumière de l'environnement comme source de lumière pour l'acquisition de l'image.
- Les capteurs actifs génèrent leur propre source de lumière dans les longueurs d'onde visibles et/ou infra-rouge pour produire l'image de la scène 3D.

Nous présentons sur la figure 2.8 un aperçu de la taxonomie des méthodes de détection optique généralement utilisées dans le suivi des personnes.

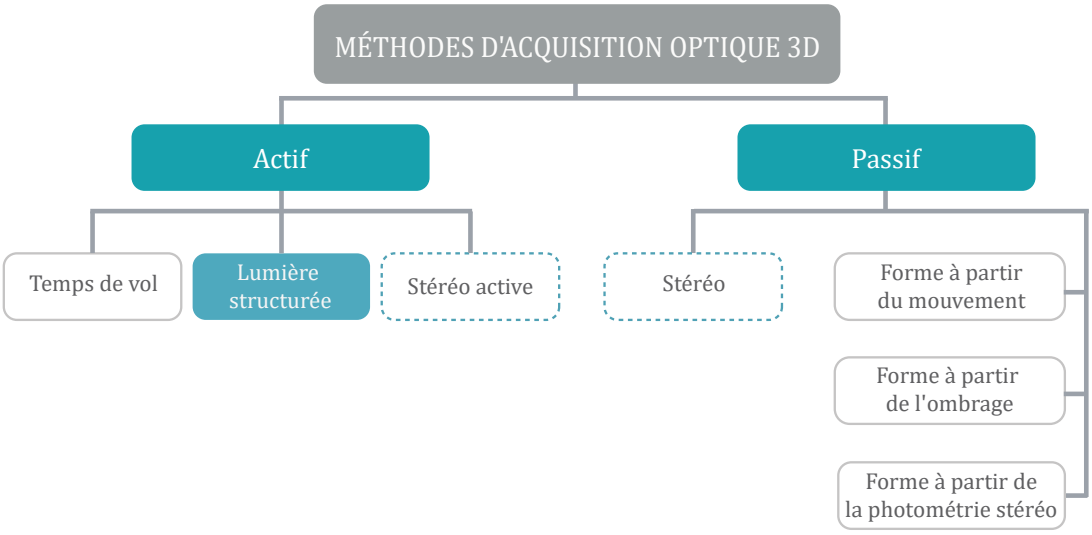


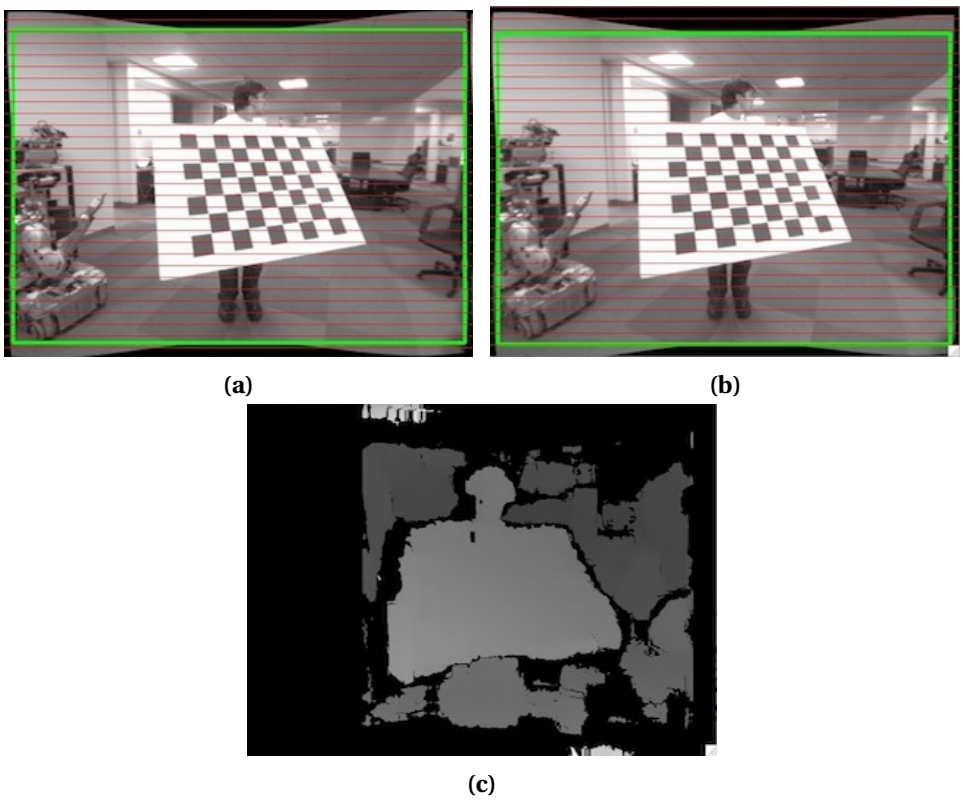
FIGURE 2.8 – Taxonomie des méthodes de détection optique 3D (basées sur [DMZC12]).

2.3.1 Méthodes d'acquisition passive

Reconstruction par stéréovision

On part du même principe que celui de la triangulation. La paire d'images est obtenue par deux capteurs identiques, étalonnées et rectifiées, observant la même scène [DMZC12]. Le processus comprend des modules d'acquisition, d'étalonnage, de rectification, de mise en correspondance stéréoscopique et de post-filtrage [Mat13]. Chacune de ces étapes réduit le nombre de correspondances potentielles en accélérant la correspondance entre pixels tout en augmentant sa fiabilité.

Suite au processus de rectification [LZ99], on obtient une paire d'images représentant la vue stéréoscopique de la scène faite (Fig. 2.9a et 2.9b), virtuellement, par deux caméras de type *pin-hole* avec des paramètres intrinsèques parfaitement identiques, avec des axes optiques X et Y identiques et axes Z parallèles et plongés dans le plan Y = 0. L'origine de la caméra gauche se trouve à $(0,0,0)$ et celle de droite se trouve à $(b,0,0)$ ou b est l'écartement stéréoscopique (la ligne de base des caméras). Dans ces conditions, les contraintes épipolaires sont respectées et les deux projections de n'importe quel point de la scène vue par les deux caméras ont la même coordonnée v et un écartement entre les coordonnées U_L et U_R égal à la disparité stéréoscopique. Il existe de très nombreux algorithmes de mise en correspondance des deux projections [MKS89, KO94, ZK00, BVZ01, TMDSA08] plus ou moins fiables, en sachant que le bruit, les occlusions partielles, les réflexions dans la scène perturbent ce processus. Une autre raison de l'échec de la mise en correspondance est l'absence de texture (zones uniformes) dans la scène : dans ce cas il y a ambiguïté sur la mise en correspondance et les résultats non-fiables sont souvent écartés dans un processus de post-filtrage.



(a) **(b)**

(c)

FIGURE 2.9 – Exemple d'image par stéréo-vision. Extrait de la documentation OpenCV. a) Image gauche, b) image droite et c) image de profondeur résultante.

Dans l'illustration de la figure 2.9, les rectangles verts sont les régions que les deux capteurs voient en commun.

Le résultat final de la mise en correspondance stéréoscopique est une carte de disparités dense mais avec des régions où l'information est absente (Fig. 2.9c). Parfois, cette carte peut être

accompagnée par une carte de fiabilité qui indique donc la fiabilité d'appariement de chaque pixel sur la carte de disparités [MKS89]. En fonction de la finesse de l'algorithme employé, les disparités sont exprimées en multiples de pixels ou de fractions de pixels [KO94, TH86] (exemple Kinect...). En utilisant les paramètres intrinsèques et extrinsèques [HS97] de la paire stéréoscopique rectifiée, on peut facilement traduire une carte de disparité en une carte de profondeur, mais la nature discrète des valeurs de disparité implique une discrétisation des valeurs de profondeur avec des pas proportionnels au carré de la distance appelée résolution de profondeur [CC92, JC94]. La résolution de profondeur stéréo théorique dZc de la caméra à une distance Z est calculée comme [CC92] :

$$dZ_c = \frac{Z^2}{fb} dp_x \tag{2.6}$$

où f est la distance focale, b la ligne de base et dp_x la précision de disparité qui dépend de l'algorithme de correspondance de disparité utilisée par la caméra.

Formes à partir de X

Les méthodologies de type « formes à partir de X » sont basées sur la problématique de récupération de la troisième dimension $Z(x,y)$ à partir d'une ou plusieurs images 2D. Ce sont des séries de méthodologies pour reconstruire la géométrie 3D, en inférant la forme de différents repères comme la vision stéréo (section précédente), le mouvement, l'ombrage, la photométrie stéréo, la texture, le foyer, la silhouette, etc. Les humains utilisent la combinaison à partir de différents indices pour générer une perception finale de la profondeur. Les repères les plus couramment utilisés dans la vision numérique sont la stéréo et le mouvement. Nous présentons en annexe les méthodologies de reconstruction 3D à partir du mouvement (annexe A), l'ombrage (annexe B) et la photométrie stéréo (Annexe C) pour donner une vision générale du domaine. De même, plus d'informations sur les autres techniques sont présentées dans le travail de [Sze10] dans le domaine de la vision par ordinateur.

2.3.2 Méthodes d'acquisition optique active

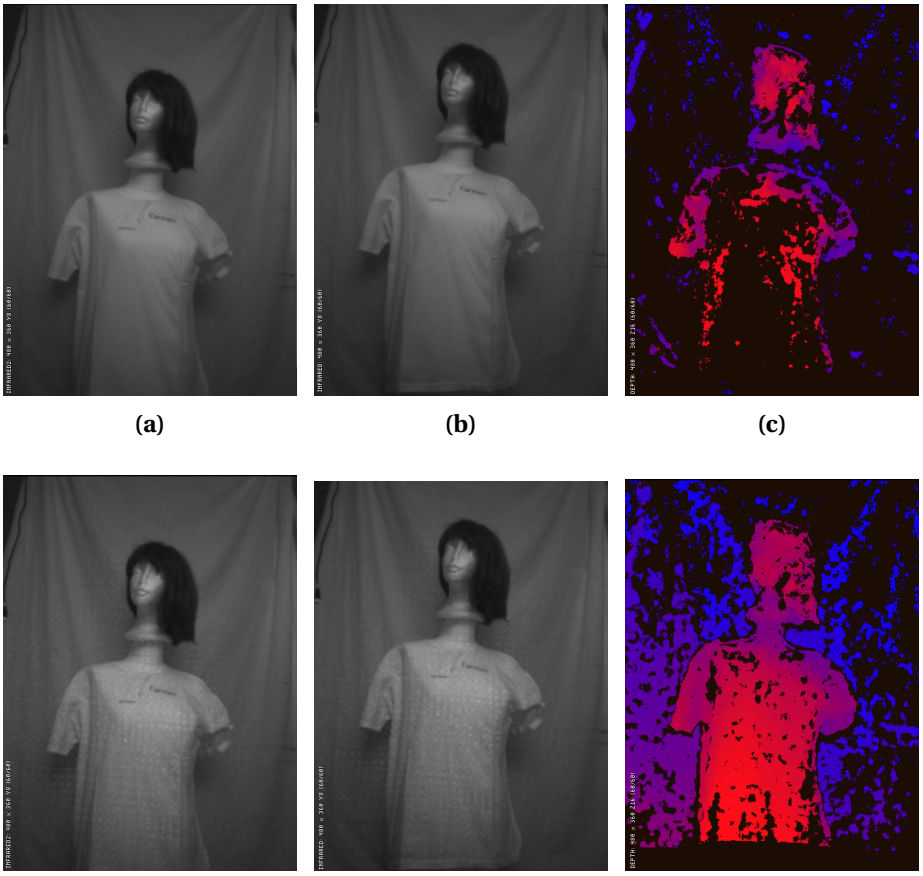
Stéréovision active

Cette méthodologie est basée sur la stéréoscopie mais pour pallier le manque de texture dans certains appareils stéréo, on installe un projecteur qui ajoute de la texture à la scène, améliorant ainsi la correspondance stéréo.

Dans la figure 2.10, on montre un exemple de la différence entre l'approche de la stéréovision active et passive en prenant la même scène. Cette scène est composée d'un mannequin au premier plan et d'un fond blanc. Dans les figures 2.10a et 2.10b, on observe la paire d'images obtenues en utilisant l'approche stéréo classique et l'image résultante de profondeur dans la figure 2.10c. Les figures 2.10d, 2.10e sont obtenues avec une approche stéréo active où l'on peut observer des points projetés dans la chemise du mannequin. De cette paire d'images, on obtient l'image de profondeur (Fig. 2.10f) plus remplie. Dans les images de profondeur, les distances les plus proches sont en rouge alors que les plus éloignées sont en bleu et la couleur noire représente des valeurs de profondeur inconnues.

Dans cette deuxième approche, on utilise une source active de lumière pour projeter de la texture sur la scène (grille de points éclairés qui n'appartiennent pas à la scène originale) dans les régions non texturées comme le fond et la chemise du sujet. Cette technique nous permet de récupérer plus d'informations, même des parties homogènes de l'image originale.

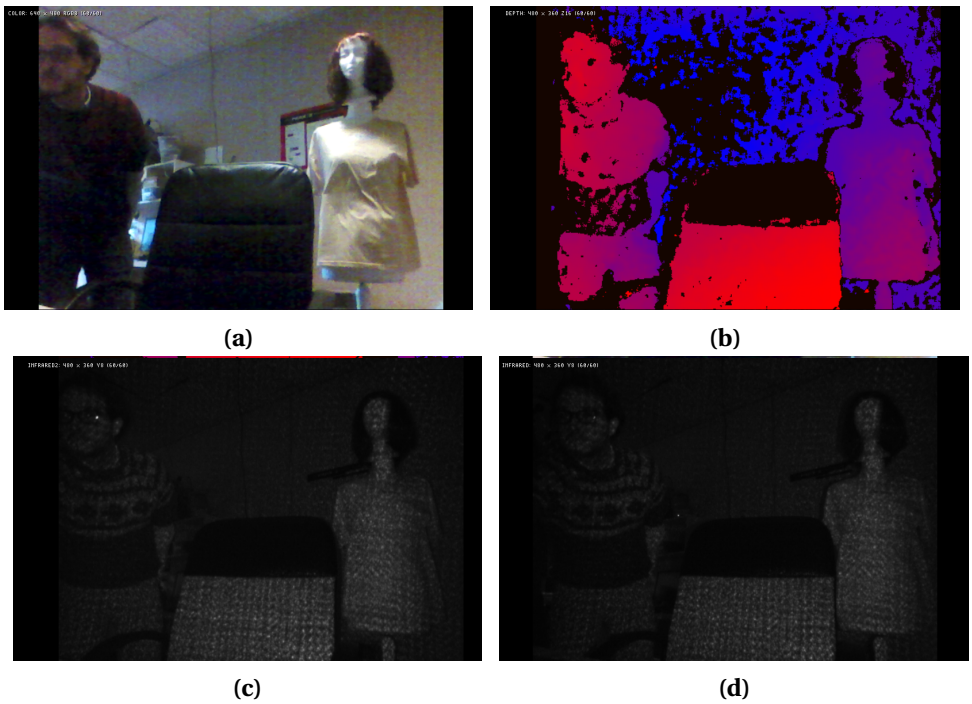
Contrairement à la figure 2.10, la figure 2.11 montre comment certaines régions de scène ne peuvent être récupérées même en utilisant des techniques stéréo actives. Cette image est importante pour souligner les limites de la vision stéréo active, par exemple, on peut voir que la partie supérieure du dossier de la chaise n'est pas texturée, montrant dans ce cas que la technologie de stéréo active ne permet pas récupérer l'information de profondeur par rapport à la technologie de stéréo classique. Une autre exemple est dans les régions les plus éloignées (centre gauche) de l'image, la texture projetée n'a aucun effet sur le résultat du « stéréo-matching ».



(a) **(b)** **(c)**

(d) **(e)** **(f)**

FIGURE 2.10 – Images d'entrée et de sortie des approches stéréo active et passive. La première ligne montre l'approche passive et la deuxième l'approche active. Dans chaque ligne, on observe l'image gauche, droite et de profondeur respectivement.



(a) **(b)**

(c) **(d)**

FIGURE 2.11 – Images obtenues par stéréo active : a) image couleur, b) l'image de profondeur résultante, c) et d), images infra-rouge de la scène.

Principe de la vision 3D par lumière structurée

Basé sur le principe de la triangulation, on doit obtenir deux images pour estimer la carte de profondeur. La première image est obtenue à partir de la virtualisation d'une caméra où l'image acquise est un motif stocké dans la mémoire de la caméra (Fig.2.12a). La deuxième image est créée à partir de la projection de ce motif, en respectant le principe de la projection de perspective inverse, à travers le centre de la projection sur les surfaces de la scène. Les rayons réfléchis dans la scène sont capturés par le capteur IR et l'on obtient la deuxième image (fig 2.12b). Avec ces paires d'images, on peut à nouveau appliquer le modèle de triangulation pour calculer la carte de profondeur (Fig. 2.12c) où b est la distance entre le projecteur et la caméra et f est la distance focale de de celles-ci, créant un système rectifié.

Nous pouvons trouver les premiers travaux de cette méthode [Ber95, BK87, CH85, CHCW97], mais elle a été largement popularisée après la création du Microsoft Kinect v1 en 2010, Caméra 3D basée sur la méthode de la lumière structurée. Cette technologie a été créée par la société israélienne PrimeSense. Elle est utilisée dans plusieurs caméras industrielles telles que la version Kinect, Asus Xtion, Asus Xtion Pro, PrimeSense Carmine et Occipital Structure Sensor.

(a) **(b)** **(c)**

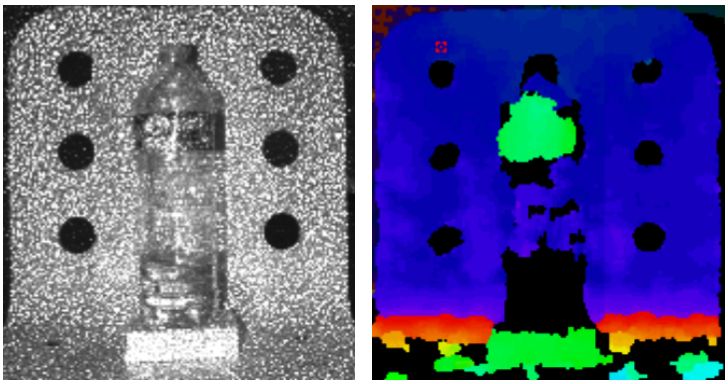
FIGURE 2.12 – Images acquises avec une caméra à lumière structurée (ASUS Xtion Pro). a) Image IR du motif projeté sur une surface plane. b) Image infrarouge d'une scène. c) Profondeur de l'image résultant de la scène b.

La figure 2.12 montre les images acquises par l'Asus XTION Pro. Les deux premières images (Fig. 2.12a et Fig. 2.12b) sont les images IR où l'on peut observer le motif projeté dans une surface plane (Fig. 2.12a) et une scène avec plusieurs objets (Fig. 2.12b). L'image (Fig. 2.12c) est l'image de profondeur résultant de la même scène que (Fig. 2.12b).

(a) **(b)**

FIGURE 2.13 – Exemple du motif projeté dans une surface hautement réfléchissante et un objet transparent. a) Image couleur avec un portefeuille réfléchissant et une bouteille d'eau transparente en plastique.

Dans la figure 2.13, nous observons quelques exemples de surfaces qui présentent des problèmes de réflectance et de réfraction. Nous observons que les surfaces du portefeuille et la bouteille d'eau ne peuvent pas être estimées. La surface du portefeuille reflète très fortement la projection IR de sorte que la caméra n'est pas en mesure d'estimer sa profondeur (problèmes de réflexion). Dans le cas de la bouteille d'eau, le motif passe à travers la bouteille en plastique et à travers l'eau, en changeant sa trajectoire et rendant impossible l'estimation de la profondeur (problèmes de réfraction). Seule la surface de l'étiquette de la marque sur la bouteille permet une estimation de profondeur.



(a) **(b)**

FIGURE 2.14 – Exemple d'une bouteille transparente à moitié remplie a) image infra rouge b) image de profondeur.

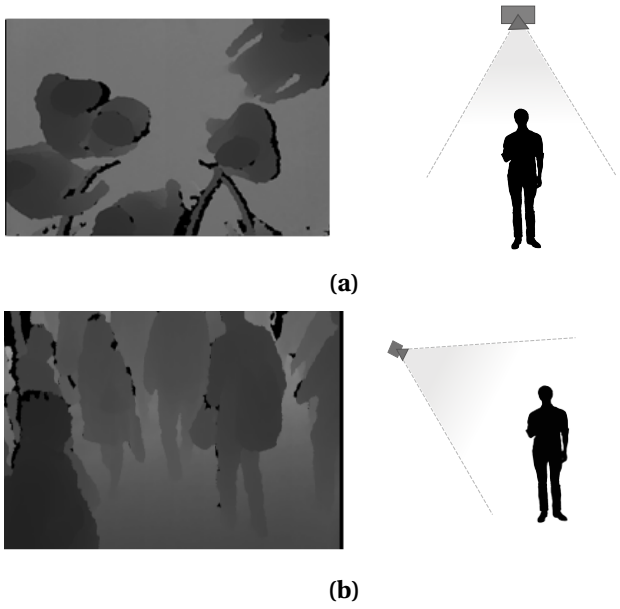
Dans la figure 2.14, nous pouvons observer la réfraction de la projection du motif dans deux parties différentes de la bouteille en plastique. Tout d'abord, dans la partie centrale de la bouteille où la lumière est passée à travers la bouteille, puis la surface du fond et revient vers la caméra IR, nous obtenons une estimation de profondeur de l'objet derrière la bouteille. Deuxièmement, dans la partie inférieure de la bouteille qui est pleine d'eau, le motif projeté souffre d'un phénomène de forte réfraction, ce qui ne permet pas d'estimer la profondeur.

En conclusion, nous avons deux principes fondamentaux et plusieurs méthodes pour obtenir des images 3D. Chaque méthode comporte également plusieurs implémentations technologiques qui seront évaluées à la section 2.4. Cependant, ces constats mettent en évidence un point important qui est l'influence de la position de la caméra dans l'acquisition de données 3D avant de continuer avec une évaluation des capteurs disponibles sur le marché.

2.4 Influence de la position de la caméra dans l'observation de la scène

On souhaite trouver le meilleur placement des caméras par rapport à la fiabilité des informations acquises, la quantité d'informations et la quantité de calculs impliqués, de façon à obtenir une réponse fiable pour la détection et le suivi de personnes. Par ailleurs, on souhaite également respecter des contraintes imposées (dans ce cadre industriel de thèse CIFRE), notamment le fait d'obtenir des résultats en temps-réel et sur des architectures à faibles coûts (et de type systèmes embarqués), et à cela s'ajoute la contrainte de disposer d'une faible puissance.

On veut utiliser les cartes de profondeur pour segmenter, identifier et traquer des personnes que l'on va désigner par "piétons". Le positionnement de la caméra et son angle de vue sont des paramètres importants pour l'analyse et l'identification des piétons dans des images acquises. On considère deux cas de figure typiques : la vue de dessus (ou zénithale) et la vue latérale. Les travaux antérieurs [Che03, GG13, TYOY99, VZS13, Rau13] démontrent l'avantage du positionnement zénithal de la caméra (Fig. 2.15), où les piétons sont observés depuis le plafond. L'avantage essentiel est d'éviter un maximum d'occultations de personnes (partielles ou complètes) lors de situations de foules. Ainsi, on peut améliorer la détection et diminuer la complexité du cas où la personne est dans la scène mais en occulte une autre. Un autre avantage est de disposer d'une vue claire des têtes des personnes indépendamment de leur emplacement dans le champ de vision de la caméra. Toutefois, certains cas d'occultation ne pourront être évités. D'autres études comme [FMX12, KAD+14, NSMH10] préfèrent la vue latérale (figure 2.15) pour recueillir plus d'informations sur les détails du corps de la personne pour la suivre ou pour avoir une interaction avec la personne (type commandes manuelles [CLC+13]). Ainsi, à partir de méthodes complexes, il est possible d'obtenir la trajectoire individuelle de ces personnes, même si elles se touchent ou si elles sont très proches [KAD+14]. Ces travaux semblent suggérer que la position de la caméra dépend de son utilisation. Par conséquent, nous avons examiné le type d'informations acquises dans chaque cas, en ayant à l'esprit que nous avons l'intention de maximiser la zone couverte par la caméra, de minimiser l'occultation entre les piétons et de minimiser la quantité d'information à traiter afin de pouvoir implanter une solution en temps-réel et à faible coût.



(a)

(b)

FIGURE 2.15 – a) Carte de profondeur d'une foule avec une caméra Asus Xtion Pro (lumière structurée) en position de vue de dessus et b) vue latérale extraite de [KAD+14].

Dans la figure 2.15b la premier image de chaque ligne est la carte de profondeur d'une foule produite par une caméra PrimeSense ***R**ed – **G**reen – **B**lue – **D**epth* (RGB-D) (lumière structurée) et la deuxième image montre la position de la caméra par rapport à la personne observée.

2.4.1 Relation de distance entre la caméra et les personnes dans la scène

La position de la caméra a un impact sur le comportement de la variation de la distance de la personne par rapport à la caméra à l'intérieur de son champ de vision. L'analyse de cette variation est importante parce qu'elle influence directement la variation du modèle du sujet (voir 1). De la même manière, ces mesures de profondeur représentent les données pour localiser les personnes par rapport aux repères de la caméra, ce qui permet de déterminer sa position dans le système. Dans le cas où la caméra est placée en position zénithale (Fig 2.16a), la variation des mesures des distances d_n entre la caméra et la personne par rapport à sa position à l'intérieur du champ de vision, est donnée par :

$$\Delta d = \begin{cases} d_r & \alpha_n = 0 \\ d_r \left[\frac{1}{\cos\alpha_n} - 1 \right] & \alpha_n \leq \frac{\varphi}{2} \end{cases} \tag{2.7}$$

où Δd est la variation de la distance mesurée, d_r est la distance de la personne au centre du champ de vue (***F**ield **o**f **v**iew* : FoV) et α est l'angle entre la droite d_r et une droite d_n qui part du centre de la caméra et intercepte la personne à une position variable à l'intérieur du champ de vision de la caméra. La valeur de α varie entre $0 < \alpha \leq \varphi/2$, ou φ est l'angle diagonal (Fig. 2.16) du champ de vision de la caméra. De plus, cette relation est symétrique par rapport au centre optique de la caméra.

Dans le cas de la caméra en position latérale, la distance entre la caméra et la personne varie à l'infini (où la limite de la portée de profondeur du champ de vision de la caméra). Bien qu'il soit possible de calculer la relation entre les distances d_n de la figure 2.16b, cette relation n'est pas forcément bornée et dépend de l'angle d'installation de la caméra, qui est également variable. De plus, sa géométrie dépend d'un modèle externe différent de celui de la caméra. Par exemple, pour extraire la distance d_r, il faut estimer la distance entre le point p^{d_i} et sa projection à un plan parallèle au sol et à la hauteur d'installation de la caméra. A cela s'ajoute la complexité pour l'extraction des propriétés observables des personnes. p^{d_i} est le point d'intersection de la personne et le cercle de rayon d_n. Ce processus ajoute de la complexité aux calculs pour extraire les propriétés géométriques des personnes.

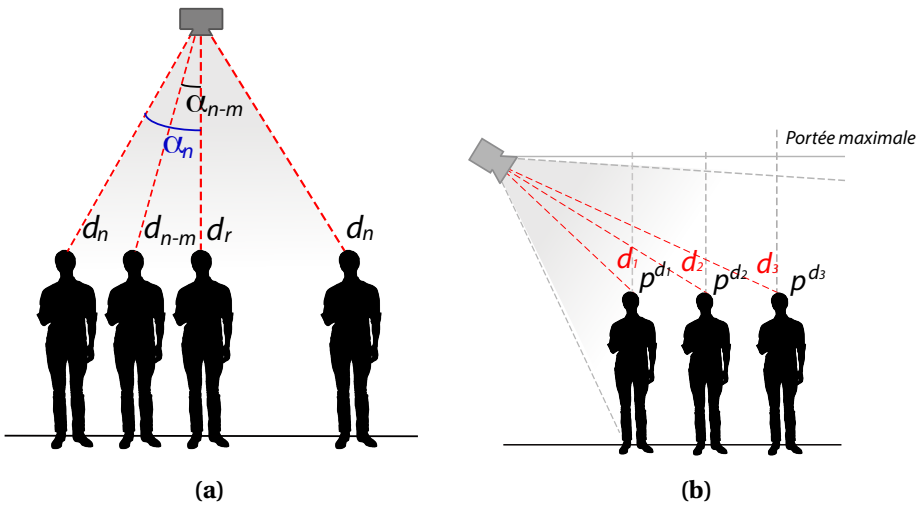


FIGURE 2.16 – Variation de la taille des personnes par rapport à la caméra en position : a) zénithale et b) latérale.

En conclusion, nous pouvons dire que le modèle géométrique de la caméra en position zénithale est plus simple que celui d'une caméra en position latérale. Par conséquent, le processus d'extraction géométrique est simplifié et sera performant. La propriété de la similitude géométrique directe fait que nous avons un champ de vision symétrique qui réduit la complexité de la mise à jour du modèle du sujet et la variation de l'échelle.

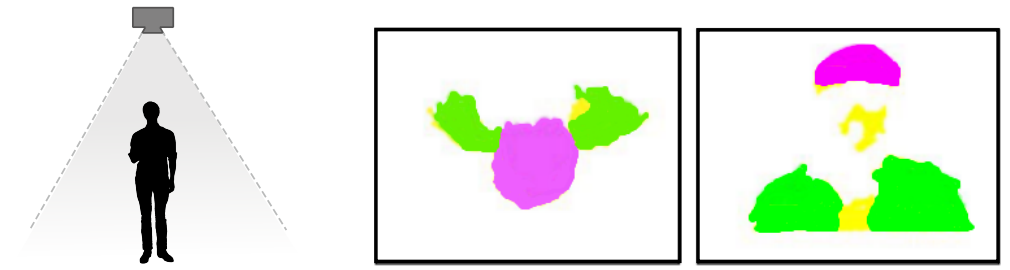
2.4.2 Influence de la position de la caméra dans l'acquisition de données

Nous souhaitons comprendre les propriétés extraites des images acquises par les caméras dans des positions différentes. Ces propriétés doivent être stables, faciles à interpréter et à segmenter. Cela signifie que nous voulons que ces informations soient stables sur la durée et par rapport à la position de la personne dans le champ de vision. On a besoin d'informations appartenant aux parties du corps concernées pour pouvoir caractériser et identifier une personne. Cela signifie aussi de minimiser la complexité à séparer les différentes parties du corps.

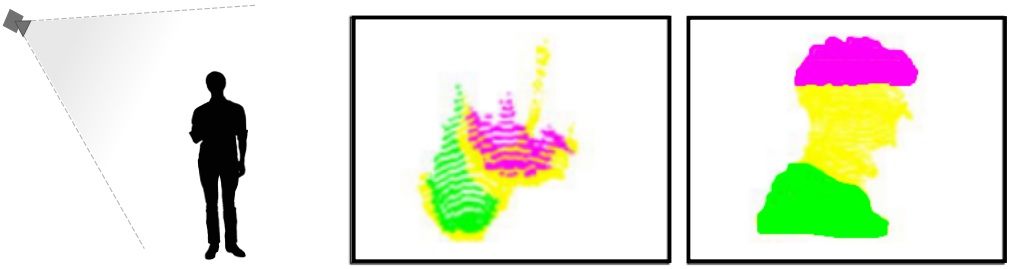
Les travaux sur le comptage des personnes basé sur des capteurs 3D ont montré l'intérêt d'identifier spécifiquement la tête et les épaules pour reconnaître une personne dans une séquence vidéo. Le concept tête-à-épaules (head-to-shoulders) a été développé et utilisé dans plusieurs travaux [BKIM13, KAD+14, Rau13] de suivi de personnes. Ce concept sert à caractériser les personnes dans les images de profondeur. Les résultats des travaux de [BKIM13] sont illustrés dans la figure 2.17. Ils représentent les images obtenues avec une caméra 3D en vue de dessus (Fig. 2.17a) et en vue latérale (Fig. 2.17b). On observe un nuage de points associés à une personne où la tête est représentée en violet, les épaules en vert et le reste du corps en jaune. On peut observer dans les images de la figure 2.17 comment la position de la caméra affecte la distribution de l'information de profondeur sur les images acquises. Pour chaque position de la caméra, l'information est distribuée différemment sur les parties du corps humain (à savoir la tête, le visage, le cou et l'épaule). Dans la vue de dessus, on observe comment la plupart des points sont rassemblés sur la tête et les épaules alors qu'il n'y a que peu de points sur le visage et le cou. Dans la vue latérale, nous disposons de beaucoup d'informations sur le côté latéral de la tête, du visage, du cou et des épaules, bien distribuées le long du corps et bien moins d'informations sur l'intégralité du dessus de la tête et les épaules. On constate que l'on obtiendra toujours une information partielle sur les parties du corps à cause de l' « auto occultation » en vue latérale. Par exemple, on verra soit la face, soit le dos, le côté gauche ou le côté droit.

Notre objectif est d'extraire la tête et les épaules des personnes pour réaliser le suivi. Nous avons donc focalisé notre intérêt sur les images en position zénithale.

Si notre but était de détecter le mouvement des mains pour contrôler des jeux ou détecter les gestes, nous aurions préféré positionner les caméras en position latérale.



(a)



(b)

FIGURE 2.17 – a) Image acquise par une caméra en vue de dessus ; b) Image acquise en vue latérale.

Dans les images 3D de la figure 2.17, la trace de la tête est représentée par la couleur magenta, la trace des épaules est représentée en vert et le reste des parties de corps en jaune. Ces images sont extraites de [BKIM13].

L'avantage principal d'utiliser des images orientées vers le bas est que la majorité des points acquis de l'image appartiennent à la tête et aux épaules. Ceci implique moins de puissance de calcul, donc aussi de temps de calcul, nécessaire pour avoir une segmentation optimale. Cependant, il manque l'information sur les autres parties du corps qui pourraient être utilisées pour identifier la personne.

2.4.3 Influence de la position de la caméra dans les grands espaces publics

Dans le scénario où nous avons besoin de plusieurs caméras pour couvrir un espace public (en raison de la taille de l'espace public ou des caractéristiques de l'environnement comme des obstacles physiques), la position zénithale permet de réduire les zones de chevauchement (Fig. 2.18a), évitant une gestion complexe de l'identification des personnes dans ces régions. À l'inverse, plusieurs caméras placées en position de vue latérale (Fig.2.18b) doivent résoudre les problèmes de *variation d'échelle* et de *déformation du modèle du sujet* pour chaque personne dans les zones de chevauchement.

Dans les images de la figure 2.18, les piétons sont représentés par la couleur verte, les caméras par des carrés gris foncé et leurs champs de vision par des régions en gris clair (translucide), afin de mettre en évidence les régions de chevauchement. Les zones de chevauchement qui apparaissent de plus en plus foncée, selon le nombre de caméras qui surveillent cette zone. De plus, la zone de surveillance commune (**Z**one of **c**ommon **s**urveillance : ZCS) est représentée par des tracés rouges ou bleu foncé. La zone temporairement avaugle (***T**emporarily **b**lind* **Z**one: ZTB) est représentée par des régions noires. Enfin, les pièces apparaissent sous la forme d'un carré noir.

Nous définissons une ZCS comme la zone de chevauchement des champs de vision des caméras. On parlera de chevauchement d'intensités différentes : bleue pour la zone moins intense et rouge pour la zone plus intense. De la même manière, nous définissons une ZTB comme un espace dans lequel aucune des caméras du système n'est capable de surveiller la zone à cause de l'occultation générée par des objets dynamiques qui n'appartiennent pas à la scène. Dans

certains cas, on trouvera probablement des zones aveugles dans des systèmes de surveillance, dues à la distribution des espaces et des obstacles qui appartiennent à l'environnement comme des colonnes ou des murs mais nous ne considérerons pas ces situations dans notre analyse. Si cela était nécessaire, il serait important que ces zones aveugles soient entourées de zones où les cameras couvrent le champ, de sorte qu'il soit impossible de se trouver dans une zone neutre sans passer par une zone contrôlée.

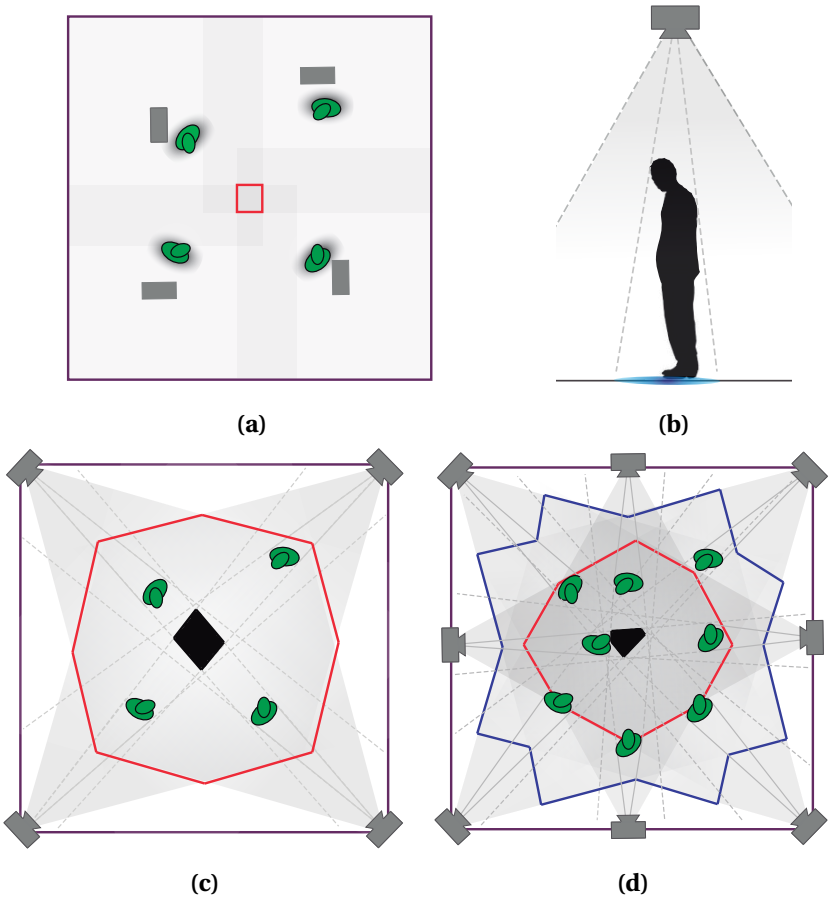


FIGURE 2.18 – Représentation du placement de caméras (rectangles gris foncé), leurs champs de vision (régions gris clair), de personnes (figures vertes), des zones de chevauchement (contournées par des lignes rouges et bleu foncé) et des zones aveugles (régions noires) dans une chambre (carrée externe noire). Les projections des ombres d'occultation sont représentées par des lignes grises entrecoupées après l'interception avec la personne.

Les figures 2.18a, 2.18c et 2.18d sont des vues de dessus du sol (plan de sol) avec la disposition des caméras installées en position zénithale (Fig. 2.18a) et latérale (Fig. 2.18c et 2.18d) pour couvrir la totalité de la pièce. La figure 2.18b est la vue en coupe de représentation de l'occultation générée par une personne en configuration zénithale. Le plan de sol et vue en coupe sont des termes utilisés dans le domaine de l'architecture. Le plan de sol représente la projection orthogonale des objets sur le plan horizontal et la vue en coupe une vue latérale sur le plan vertical[1].

Dans la figure 2.18a, nous observons la couverture de la pièce pour quatre caméras en position zénithale. Nous observons qu'il y existe une petite ZCS haute ou les quatre champs de vision des caméras sont superposés. De plus, les personnes projettent des ombres d'occultation qui ne sont pas visibles dans cette figure, mais qui sont illustrées dans la figure 2.18b. Ces projections sur le sol de taille réduite, sont les ZTB dans le cas de positionnement zénithal des caméras.

La figure 2.18c illustre une pièce carrée où se trouve une caméra dans chaque coin. La ZCS rouge représente une surface assez grande de la pièce. Pourquoi est-ce un problème ? Par exemple, une personne proche de la caméra va cacher une zone importante des donc des personnes

[1] Nous utiliserons ces termes le long de la thèse pour la description des figures.

se trouvant derrière. Le fait d'avoir plus d'images de la même personne à partir de différents points de vue augmente la robustesse de sa détection et de sa localisation, cependant le coût de calcul et celui de l'installation augmente. D'autre part, chaque fois qu'une personne se trouve dans une zone couverte par plus d'une caméra, le système global doit vérifier dans les zones de chevauchement de chaque caméra afin de gérer la localisation globale de la personne.

2.4.4 Analyse de résultats et conclusions

La figure 2.19 présente, sous la forme d'une rose des vents dans les deux cas de position de la caméra (vue de dessus, vue de côté), cinq axes d'estimation de performances. L'information est définie comme la quantité de données collectées du corps humain étudié ; le temps de calcul représente le temps pour segmenter les personnes et les parties du corps humain (propriété inverse de l'information) ; la couverture est définie comme la quantité de couverture de zone par une seule caméra dans sa gamme de vision fiable ; le recouvrement se réfère à la zone de chevauchement des champs de vision couverts par deux caméras voisines. L'occultation se réfère à la zone cachée par une personne surtout dans le cas où celle-ci est proche de la caméra. Dans un scénario parfait (représenté par le périmètre du pentagone dans la figure 2.19), nous aimerions avoir le maximum d'informations du sujet étudié avec un temps de calcul très faible pour segmenter les personnes et le corps humain, une occultation minimale possible entre les personnes et la couverture maximale avec la zone de chevauchement minimum.

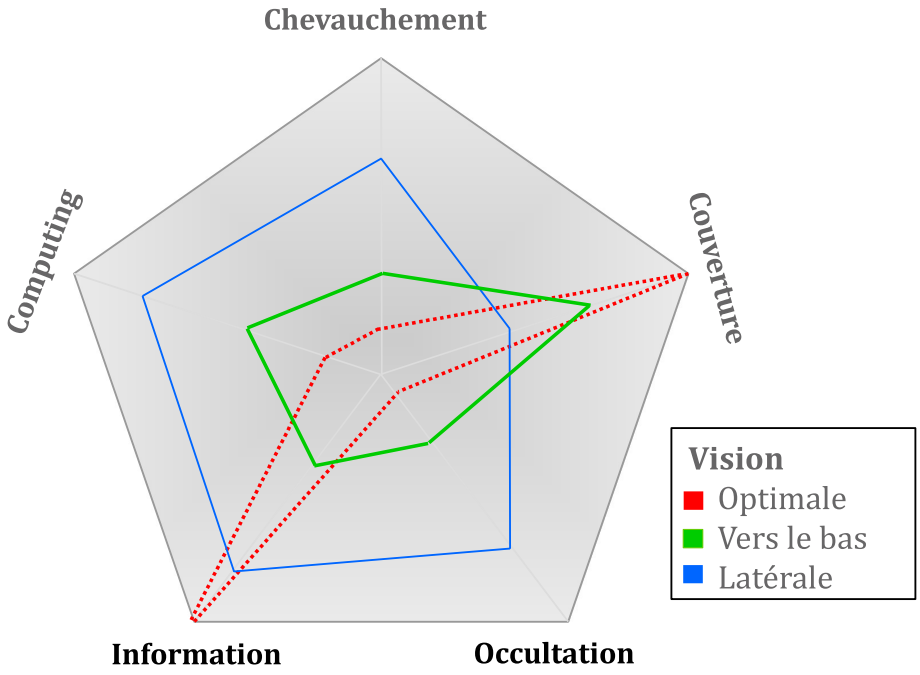


FIGURE 2.19 – Rose des vents de comparaison de la position (vue de dessus et vue de côté) de la caméra avec la performance attendue. La comparaison concerne le temps de calcul, le recouvrement, la couverture, l'occultation et l'information.

Sur l'illustration de la figure 2.19, le bord extérieur représente la performance optimale de la position de la caméra. La ligne rouge représente la performance de la position de la vue de dessus et la ligne verte représente la performance de la position de la vue latérale.

En conclusion, les informations acquises dans chaque position de la caméra ont des propriétés différentes reposant sur la distribution des données le long des parties du corps, l'évaluation de la position de la caméra dépend de l'application visée. Comme nous avons l'intention de maximiser la zone de couverture, de minimiser l'occultation et de rechercher à minimiser les temps de calcul, la position la plus appropriée qui répond à nos besoins est la vue de dessus. Ainsi, nous utilisons moins de périphériques et évitons des calculs complexes pour détecter et séparer des personnes. Ceci confirme la stratégie de la vue de dessus pour les applications de comptage des personnes, généralement installées au plafond.

2.5 Évaluation des caméras 3D du marché

Afin de comprendre les propriétés des images 3D et la différence entre les technologies et les approches, nous avons étudié différents capteurs 3D du marché. Ces capteurs sont basés sur la vision stéréo, la stéréo active, la lumière structurée et la technologie de temps de vol. Ces capteurs sont livrés avec des bibliothèques logicielles pour faciliter l'accès aux données et aux images.

Les critères de sélection retenus pour la comparaison sont les suivants : le système de capteurs doit acquérir des images en temps réel, une solution récente à faible coût et une capacité d'intégration dans une solution embarquée. C'est un critère important dans le contexte de l'entreprise car la caméra doit pouvoir être intégrée dans un environnement de manière à ce qu'elle soit autonome, facile à installer et évolutive.

Nous décrivons, dans la suite du document et pour chaque caméra, le type de signal de la caméra (actif ou passif) [TDS10], l'approche d'acquisition 3D, le type et la quantité de capteurs imageurs, les bibliothèques ou ***S**oftware **D**evelopment **K**it* (SDK), et leur capacité à être intégrés. Ensuite, nous sélectionnons un groupe de capteurs pour étudier en détail les propriétés des informations fournies.

2.5.1 Caméras utilisant stéréo vision

Caméra de l'université de Bologne

La solution proposée par l'université de Bologne est une caméra stéréo passive avec deux capteurs ***R**ed – **G**reen – **B**lue* (RGB) et un ***F**ield **P**rogrammable **G**ate **A**rray* (FPGA) (voir figure qui calcule la carte de disparité). Cette caméra peut être utilisée dans les environnements intérieurs et extérieurs et est déjà intégrée dans un système de comptage de personnes [MM14]. On trouvera toutes les informations nécessaires dans la référence [MP15]. La conception de la caméra permet de modifier la distance de référence et les capteurs d'imagerie entre RGB et niveaux de gris. Il possède une bibliothèque qui offre la fonctionnalité d'acquisition d'image et la configuration des paramètres.

Etron eSP870

Le chipset Etron eSP870 [web17d] est un contrôleur produisant une carte de profondeur, à partir de caméras stéréos passives, et capable de fournir également des images de couleur simultanément [Lu15]. Etron fournit une bibliothèque eSPDI pour Linux, Linux embarqué et un système plus complet pour Windows. Le système d'exploitation Windows fournit une fonctionnalité de post-traitement, effectuée par le processeur hôte. Le modèle évalué est composé de deux capteurs d'image ***C**omplementary **M**etal-**O**xide-**S**emiconductor* (CMOS) HD 720P (1280x720) et du chipset eSP870. Le contrôleur fournit une paire d'images couleur 1080p synchronisées à 60 fps et fournit des images de profondeur VGA jusqu'à 30 fps sur Windows et jusqu'à 15 fps sous Linux. Cependant, au cours de notre test, la caméra n'a pas été en mesure de fournir une paire de 1080p d'images et les images de profondeur simultanément. L'entreprise Etron fabrique actuellement des modèles de caméras avec des lignes de base de 3 cm et de 12 cm, avec une interface de communication USB 3.0 et 2.0.

Parrot S.L.A.M. Dunk

La nouvelle solution de navigation développée par Parrot est connue comme le S.L.A.M. Dunk [web16c]. Le système de capteurs 3D est composé de deux caméras à œil de poisson haute définition. Chaque capteur a une résolution de 1500x1500 pixels. La carte de disparité est calculée avec les pixels centraux des images à une résolution de 640x480 pixels. Cette solution assure la compatibilité avec le ***R**obot **O**perating **S**ystem* (ROS). Il peut être utilisé comme capteur ou système de navigation autonome grâce à IMU, aux capteurs à ultrasons, au processeur Tegra K1 (NVidia) et à un système d'exploitation Ubuntu intégré. En raison de la méthode d'acquisition 3D de la caméra, il convient aux environnements intérieurs et extérieurs. Le dispositif a besoin d'une alimentation externe de 12V.

Sony PS4eye

Le PS4eye [web17h] est une paire synchronisée de capteurs CMOS couleur. Il s'agit d'une solution propriétaire pour la console de jeu PS4 de Sony. Toutefois, un pilote open source a été développé grâce à la rétro-ingénierie et est disponible sur le web. Cette caméra est alimentée par le port AUX (auxiliaire), connectique propriétaire de Sony.

Pour tester la caméra, le câble a dû être modifié pour l'adapter à une interface USB 3.0 standard. Cette caméra ne fournit que des images synchronisées en couleur. Les cartes de disparités doivent être calculées par le processeur hôte.

Stereolabs ZED

Ce capteur de profondeur haute résolution est proposé par Stereolabs. La caméra ZED [web17i] fournit une paire de caméras couleur avec une résolution maximale de 2208x1242 pixels et une vitesse de capture de 15 fps jusqu'à 100 fps avec une résolution VGA. Le SDK ZED traite la carte de disparité sur la machine hôte nécessitant un GPU NVidia récent avec CUDA ***A**pplication **P**rogramming **I**nterface* (API) 6.0. Les autres exigences minimales de la machine PC hôte sont un processeur dual core 2,4 GHz et 4 Go de RAM. La caméra a été conçue principalement pour la navigation autonome et la cartographie.

2.5.2 Caméras utilisant la stéréo active

Intel R200

La caméra Intel R200 [web17f] est une caméra stéréo active / passive à lumière structurée [ASR+15, LBPF11]. Cette caméra est composée de deux capteurs IR et d'un laser infrarouge qui projette la texture, améliorant les mesures de courte distance dans les environnements intérieurs. En outre, il fournit une image 1080p RGB. Cette caméra dispose d'un SDK RealSense complet pour Windows et d'une API (Interface de programmation) multiplate-forme désignée par libRealsense. L'API permet de modifier les propriétés des capteurs et les paramètres de l'algorithme stéréo. Il est possible de configurer les aspects suivants des caméras couleur et infrarouge : luminosité, gain, exposition, fps, équilibre des blancs, auto-exposition et gain automatique. De plus, il est possible de mettre en place une zone d'intérêt dans l'image pour calculer des valeurs automatiques. En outre, le pilote RealSense fournit la compatibilité avec le ROS et fournit également un ensemble complet d'exemples de codes sources. Ce module est alimenté par USB, mais est limité aux câbles de courte distance en raison de la consommation d'énergie. Il est important de noter que les caméras Intel ne fonctionnent qu'avec les processeurs Intel.

2.5.3 Caméras utilisant la lumière structurée

ASUS Xtion Pro

L'Asus Xtion Pro est une caméra active, basée sur la technologie de vision 3D par lumière structurée, décrite dans [LNL+13]. Cette caméra est composée d'un émetteur IR de modèle fixe, d'une caméra infrarouge et d'une caméra RVB auxiliaire. Il peut acquérir des images de profondeur et des images en couleur en même temps. Le SDK fourni est bien connu, OpenNI, qui permet de contrôler aussi les caméras Kinect et PrimeSense Carmine (les trois caméras utilisent le même capteur). Cette caméra ne peut fonctionner que dans des environnements intérieurs. Ce produit est identique aux PrimeSense Carmina et Microsoft Kinect.

Intel SR300

Le produit Intel SR300 est une caméra active basée sur le principe de vison 3D à lumière structurée [BMKB12] qui projette un motif grisé. L'estimation de la disparité se fait entre l'image virtuelle (motif de code gris projeté) et l'image acquise, comme cela a été expliqué plus haut. Le même principe, expliqué, se compose d'une caméra infrarouge, d'un projecteur IR et d'un capteur d'obturateur 1080p RVB auxiliaire. Il utilise le même SDK que le modèle SR300 et ce module est alimenté par USB.

2.5.4 Temps de vol

Microsoft Kinect pour Xbox One

Cette caméra, également connue sous le nom de Kinect 2.0 [web17g], est une caméra active basée sur la technologie de temps de vol [PP15]. Il est composé d'un émetteur infrarouge (IR), d'une caméra infrarouge et d'une caméra couleur auxiliaire (RGB). Il dispose d'un kit de développement logiciel (SDK) 2.0 de Kinect pour Windows qui inclut des exemples et l'accès au code source. En outre, ce SDK prend en charge les langages .NET. Une description détaillée de la caméra Kinect 2 est fournie dans la référence [PP15].

2.6 Comparaison des spécifications techniques des caméras

Dans cette section, nous comparons les caméras sélectionnées en analysant plusieurs paramètres. Le tableau 2.1 présente les systèmes d'exploitation de chaque caméra, le tableau 2.2 présente les langages pris en charge par le kit SDK ou l'API et le tableau 2.3 présente et compare les spécifications générales des caméras.

TABLEAU 2.1 – Système d'exploitation pris en charge

Camera SDK/API	Embedded Linux	Linux	Windows	Mac OS
Xtion Pro	x	x	x	x
Kinect v2	x	x	x	
SR300	x	x	x	x
R200	x	x	x	x
UniBo	x	x	x	
eSP870	x	x	x	x
Parrot	x	x	x	x
PS4eye	x	x		x
ZED	x	x	x	

TABLEAU 2.2 – Langages de programmation compatibles

Caméra SDK/API	C	C++	Java	Other(s)
Xtion Pro	x	x	x	x
Kinect v2		x		.NET
SR300		x*	*	Python
R200		x*	*	Python
UniBo		x		
eSP870	x	x		
Parrot		x		Python
PS4eye		x		Python
ZED		x		

* Nécessite à compilateur C++11/14.

Dans le tableau 2.1, nous pouvons constater que de nos jours presque toutes les caméras sont compatibles avec les systèmes d'exploitation les plus utilisés. Le tableau 2.2 montre pour chaque SDK ou API le support des différents langages de programmation. Il montre assez clairement que le langage de développement préféré en vision numérique est le langage C ++ suivi par le Python.

TABLEAU 2.3 – Spécifications générales de l'appareil

Spécifications	Xtion	Kinect v2	SR300	UniBo	R200	eSP870	Parrot	PS4eye	ZED
Capteurs de profondeur	IR	IR	IR	2 RGB VGA	2 IR	2 RGB	2 RGB	2 RGB	2RGB
Résolution	VGA	512x424	VGA		VGA	HD	1500x1500	WXGA	2208x1242
Obturateur de profondeur	Global	Global	Global	Global	Global	Global	Rolling	Rolling	Rolling
Cámeras auxiliaires	RGB	RGB	RGB	N/A	RGB	N/A	N/A	N/A	N/A
Résolution	VGA	FHD	VGA		FHD				
Obturateur de couler	Global	Global	Rolling	N/A	Rolling	N/A	N/A	N/A	N/A
Mesure de profondeur	Active	Active	Active	Passive	Mix	Passive	Passive	Passive	Passive
Calibration	Prop.	Prop.	Prop.	OpenCV	Prop.	OpenCV	Prop.	Prop.	Prop.
Ligne de base (mm)	74	25	47	var.	70	120	200	85	120
Fov W×H (deg)	58x45	70x60	68x54	var.	59x46	62x38	200x200	85x53	96x54
Interface USB	2	3	3	2/3	3	2/3	3	Aux/3	2/3
Portée à l'intérieur (m)	0.8-4	0.5-4.5	0.20-1.50	1-5	0.7-3.5	1-10	1-5	0.3-inf	1-15
Portée à l'extérieure (m)	N/A	N/A	N/A	1-5	1-10	1-10	1-5	N/A	1-15
Fréquence d'images (fps)	30	30	30	22	30	15	30	7.5-240	15-100

Le tableau 2.3 montre une comparaison globale entre l'ensemble des caméras étudiées, du point de vue de leurs différentes spécifications techniques. Ainsi, le tableau décrit le type de capteurs et le type de bande spectrale (infrarouge ou couleur) utilisée dans les processus d'estimation de l'image 3D, la résolution d'affichage de la carte de profondeur et le mode d'obturation (global ou déroulant). De même, il décrit (le cas échéant) les capteurs auxiliaires exclus du processus d'estimation de la carte de profondeur. On décrit également la ligne de base des caméras et la méthode de calibration de la caméra utilisée (OpenCV ou propriétaire). Deux des caméras comparées utilisent la bibliothèque OpenCV de vision artificielle tandis que les autres capteurs utilisent un algorithme de calibrage propriétaire. De plus, les angles horizontaux (largeur) ainsi que les angles verticaux (hauteur) (Fig. 2.20) du champ de vision de la caméra (degrés unitaires) sont disponibles. Enfin, nous disposons du nombre d'fps, de la portée nominale de profondeur (la distance minimale et maximale où la caméra est capable d'estimer la profondeur de manière fiable) ainsi que de la version de l'interface USB (2/3), spécifiée par les fournisseurs.

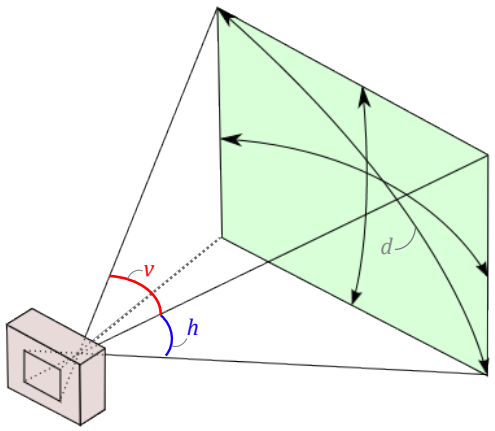


FIGURE 2.20 – Champ de vision défini par les angles de vue (l'un horizontal, l'autre vertical) de la caméra.

Nous avons synthétisé un ensemble d'informations sur le tableau 2.4 (basé sur [NML+13]) en évaluant la pertinence de l'utilisation de chaque caméra 3D dans certains domaines d'applications, en fonction de ces propriétés. Le tableau 2.4 montre l'aptitude de chaque caméra dans les domaines d'utilisation comme :

- Santé : les parties du corps sont détectées dans le but d'assurer la rééducation des patients à mobilité réduite [CCH11, NABF13].
- Contrôle gestuel : les parties du corps remplacent les manettes pour déclencher des actions [RMYZ11].
- Surveillance d'une pièce : on détecte et identifie des personnes à l'intérieur de façon robuste

pour améliorer la prise de décisions du système pour une personne donnée par rapport aux configurations de la pièce [KHM+00].

- Assistance de conduite pour fournir au conducteur les information sur les piétons : les voitures et autres obstacles [GLSU13].
- La robotique : les caméras sont capables de fournir des nuages de points 3D pour les applications de localisation et cartographie simultanées [HKH+12].
- Multimédia : on crée des contenus multimédias 3D en temps-réel [PKT14].
- Reconstruction 3D : on effectue l'enregistrement des grandes scènes et d'objets pour ensuite les reconnaître [Rus09].
- En extérieur : ces applications qui peuvent en principe être utilisées en extérieur : mais souvent l'influence de la lumière du soleil ne permet pas de le faire [MP15, PP15].
- Solutions embarquées : on intègre les caméras dans un système avec des ressources informatiques limitées comme les drones où le rôle de la caméra est d'apporter des informations spatiales en plus des informations visuelles [MM14]

TABLEAU 2.4 – Pertinence d'utilisation des caméras : ++ hautement recommandé, + recommandé, 0 faisable, − non recommandé, −− fortement déconseillé

Applications	Caméras								
	Xtion	Kinect v2	SR300	UniBo	R200	eSP870	Parrot	PS4eye	ZED
Caméras intelligentes	+	+	0	+	-	+	−	−	−
Santé	+	+	+	+	+	+	0	−	+
Contrôle gestuel	+	++	+	-	+	0	−	++	0
Surveillance de pièce	+	+	0	0	0	+	−	−	+
Assistance de conduite	-	0	0	+	0	0	++	-	++
Robotique	0	+	+	+	+	+	++	−	++
Multimédia	+	+	+	+	+	0	0	+	+
Reconstruction 3D	-	++	0	+	0	0	+	-	++
A l'extérieur	−	0	−	+	0	+	++	0	+
Solutions embarquées	+	-	0	+	+	+	++	−	−

2.6.1 Pré-sélection des caméras

Dans l'objectif de construire un système à faible coût, évolutif et facile à installer, nous présentons une analyse des capacités des périphériques à s'intégrer dans notre système souhaité. Cette analyse des caméras nous amène à un autre point de vue de l'évaluation de la performance des caméras décrites dans la section suivante.

En premier lieu, nous écartons de notre évaluation les caméras pour lesquelles l'estimation de la disparité est effectuée par un CPU hôte (hors caméra), nécessitant des capacités de traitement importantes. Ceci augmenterait le coût global du système souhaité. Par conséquent, les caméras ZED et PS4eye sont exclues de notre panel de choix, car elles ne fournissent qu'une paire d'images synchronisées (sans carte de profondeur).

En second lieu, nous écartons de notre évaluation les caméras qui ont des problèmes d'intégration matérielle dans un nouveau système embarqué. La caméra PS4eye dispose d'une interface AUX propriétaire, qu'on doit modifier pour obtenir une interface USB3. De plus, on ne dispose pas de pilotes d'interfaces officiels. Dans le cas des caméras Kinect 2, l'inconvénient principal est l'alimentation propriétaire qui est incompatible avec une intégration dans un système embarqué, car cela augmente le volume de l'ensemble.

(a) ASUS Xtion Pro

(b) Intel R200

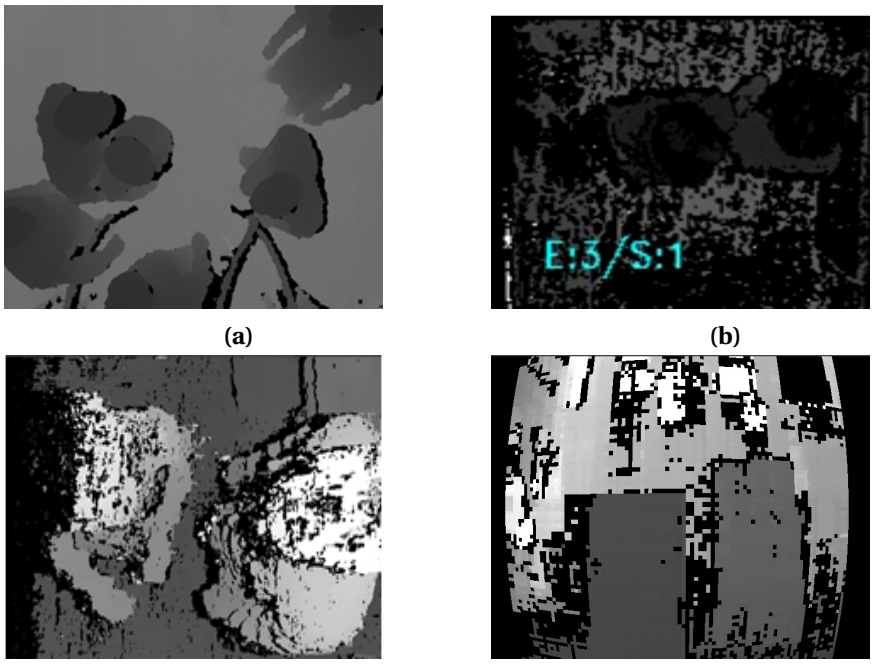
(c) Etron stereo

(d) Parrot S.L.A.M. Dunk

FIGURE 2.21 – Caméras sélectionnées pour la caractérisation des performances.

En troisième lieu, la portée nominale de profondeur doit être suffisamment grande pour observer des personnes en espaces publics, depuis des plafonds ou des sites élevés pour être installée en position zénithale (section 2.3). La caméra SR300 a une portée nominale de profondeur très limitée (jusqu'à 1,5 m), donc nous considérons que ce n'est pas possible de l'utiliser dans notre système.

Enfin, la maturité du produit jusqu'à l'industrialisation de la caméra est un critère important pour tout projet d'intégration industrielle. La caméra de l'Université de Bologne n'est pas encore dans une étape d'industrialisation, donc nous étions obligés de l'éliminer comme option pour une caractérisation plus poussée. Finalement, nous avons sélectionné les caméras ASUS Xtion PRO, Intel R200, Etron eSP870 et Parrot S.L.A.M. Dunk (Fig. 2.21) pour l'étape suivante d'évaluation des cartes de profondeur fournies (Fig. 2.22).



(a)

(b)

(c)

(d)

FIGURE 2.22 – Cartes de profondeur produites par les caméras a) ASUS Xtion PRO, b) Intel R200, c)Etron eSP870 et d) Parrot S.L.A.M. Dunk.

2.6.2 Caractérisation des performances des caméras sélectionnées

Dans cette section, nous décrivons un ensemble de procédures pour la caractérisation des caméras de profondeur. Nous avons évalué les caméras sélectionnées (L'ASUS Xtion PRO, l'Intel R200, l'Etron eSP870 et le Parrot S.L.A.M. Dunk) en utilisant trois expériences, dont deux ont été proposées par [PP15]. La troisième, spécifique, est destinée à l'usage de caméras 3D pour le comptage des personnes. Nous avons commencé par analyser la stabilité temporelle du signal de profondeur pour ensuite étudier la précision des mesures de profondeur de la caméra par rapport aux valeurs réelles. Enfin, nous avons évalué la fiabilité de la détection des personnes en estimant la quantité de pixels correspondant à chaque personne en fonction de la position dans le champ de vision d'une caméra orientée vers le bas, position la plus favorable pour notre objectif principal (voir 2.3).

La résolution des images de profondeur testées était de 640x480 pixels et la vitesse d'acquisition utilisée était de 30 fps (à l'exception de la caméra Etron qui plafonne à 15 fps quand elle est connectée à un hôte opérant sous Linux). Normalement, les caméras ont un réglage automatique pour le gain et le temps d'exposition et disposent donc d'une boucle de réglage-réaction. Ces paramètres se stabilisent dans une scène statique. Les séquences d'images utilisées pour les essais ont été acquises après la stabilisation de la boucle de réglage-réaction.

Le système hôte utilisé pour nos expériences est équipé d'un processeur Intel Core i7 4712MQ à 2,3 GHz, de 16 Go de DDR3-RAM de mémoire, d'un disque dur de 250 Go et d'un système d'exploitation Linux Ubuntu 14.04. Les expérimentations ont été réalisées dans un environnement intérieur, dans un espace fermé. Nous avons supposé comme dans [PP15] qu'aucun changement environnemental majeur ne se produirait lors de l'acquisition des données de profondeur (pendant une durée typique de mesure de moins d'une minute).

Stabilité temporelle des cartes de profondeur

Nous définissons la stabilité temporelle des cartes de profondeur comme une propriété d'une caméra 3D pour obtenir les mêmes mesures (ou avec très peu de variation) d'une scène statique (sans déplacer la caméra ou les objets dans son champ de vision). En d'autres termes, nous voulons évaluer la variation des mesures données par la caméra, lorsque ni la caméra ni les objets de la scène ne se sont déplacés. Cette propriété est temporelle, car nous faisons l'acquisition de plusieurs images à différents instants pour comparer les mesures.

Nous devons savoir qu'un capteur est constitué d'une matrice de pixels capables de mesurer la quantité de lumière incidente (nombre de photons) pendant une période de temps (temps d'intégration) et de la convertir en une tension qui est convertie en valeur numérique. Le processus de comptage des photons a une distribution de poison introduisant un bruit Poison dans les images acquises [LNL+13]. Par conséquent, une caméra fixe regardant une scène statique produit un signal bruité entre des trames consécutives, qui est naturellement transmis à l'image en profondeur [PP15].

Par ailleurs, les *pixels volants* sont un autre facteur d'instabilité de l'image de profondeur, présents à cause de l'inhomogénéité de la scène entre le fond et les objets [LNL+13]. L'effet de *pixels volants* se produit à la limite entre deux objets voisins dans les axes x et y, mais à une distance différente dans l'axe z, dans le système de coordonnées de la caméra. Par conséquent, la mesure de profondeur résultante sera un mixage de la lumière réfléchie des deux objets.

L'objectif de cette expérience est d'estimer la variation dans le temps des mesures dans les cartes de profondeur, c'est-à-dire la stabilité temporelle du signal de la carte de profondeur en conditions constantes et vérifier si les images de profondeur ainsi obtenues sont utiles pour caractériser les personnes.

Procédure

Dans la première procédure, nous avons recréé une scène composée de deux surfaces plates parallèles au plan de la caméra (évitant au maximum l'apparition de *pixels volants*), et occupant

la quasi-totalité du FoV des caméras ETRON, ASUS et INTEL. Dans le cas de la caméra Parrot, il n'est pas possible de couvrir l'intégralité du FoV, en raison du grand champ de vision des optiques d'œil de poisson. De plus, nous avons recouvert les surfaces non réfléchissantes avec un matériau réfléchissant approprié pour pouvoir évaluer l'algorithme d'estimation de la profondeur et isoler des problèmes de réfléchissement en ce qui concerne les caméras actives.

Nous avons mesuré l'erreur quadratique moyenne (***R**oot **M**ean **S**quare **E**rror* : RMSE) du signal de profondeur à travers le temps en utilisant T trames consécutives D^t, pour $t = \{1,\ldots,\mathrm{T}$. Pour chaque pixel p appartenant à l'ensemble R de pixels de profondeur valides, soit $\mathrm{D}^t(p)$ sa valeur de profondeur dans l'image t et soit $\bar{\mathrm{D}}(p)$ sa moyenne temporelle à travers la séquence entière du pixel p. On obtient alors la formule suivante :

$$\mathrm{RMSE}(p) = \sqrt{\frac{1}{\mathrm{T}}\sum_{t=1}^{\mathrm{T}}(\mathrm{D}^t(p) - \bar{\mathrm{D}}(p))^2} \tag{2.8}$$

Afin de disposer d'un indicateur de comportement global de la caméra, nous avons défini la stabilité globale du signal (***G**lobal **S**tability **S**ignal* : GSS) comme suit.

$$\mathrm{GSS} = \frac{1}{|\mathrm{R}|}\sum_{p\in\mathrm{R}}\mathrm{RMSE}(p) \tag{2.9}$$

où $|\mathrm{R}|$ est le nombre cardinal de l'ensemble R. Ces indicateurs globaux (présentés dans le tableau 2.4) ne caractérisent que le bruit temporel du signal de profondeur et non sa précision.

Dispositif expérimental

Nous avons mis en place et utilisé un dispositif expérimental de mesure présenté dans la (Fig. 2.23). Toutes les caméras ont été placées à la même position et avec la même orientation vers la scène, où les surfaces observées sont parallèles au plan de la caméra. Deux matériaux de texture différente ont été utilisés : un contreplaqué nommé « tripli » comportant une texture forte et un contreplaqué lisse avec une texture fine. Cette scène était composée de deux surfaces carrées (d'environ 1 m 2) parallèles au plan de l'image de la caméra (à 2 m et 4 m) et d'un arrière-plan dépourvu d'objets pour éviter l'introduction de bruit dans les mesures.

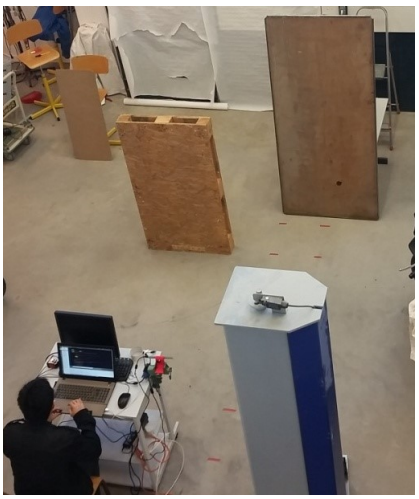


FIGURE 2.23 – Dispositif expérimental pour évaluer les quatre caméras.

Résultats

Nous avons d'abord évalué pour chacune des quatre caméras les valeurs de GSS, définissant la stabilité globale du signal (tableau 2.5) et la valeur du signal de profondeur RMSE (Fig. 2.24). Une valeur GSS faible signifie que la technologie et l'algorithme de mesure des caméras fournissent les mêmes résultats pendant le temps où la caméra regarde la même scène statique. Au contraire,

des valeurs plus élevées montrent que la caméra a des difficultés à obtenir les mêmes mesures reproductibles à partir de la même scène statique. Nous recherchons une caméra avec un faible GSS, ce qui nous permet d'obtenir une détection stable d'une personne pendant un temps donné dans un environnement statique. Simultanément, une faible valeur de GSS facilite le suivi des personnes dans des conditions difficiles, comme la *déformation du modèle du sujet*, la *variation d'échelle*, la *séparation des personnes*, la *modélisation de l'arrière-plan*. Dans le cas de la modélisation d'arrière-plan, les pixels de mesure instables pourraient créer des erreurs pour distinguer le premier plan du fond. Certains algorithmes classiques de modélisation d'arrière-plan utilisent un seuil de variation des valeurs (illumination, profondeur, etc.) des pixels pour décider s'il y a une variation de fond. Par conséquent, un seuil très élevé dû à une variation élevée du signal de caméra, a un impact direct sur l'identification des pixels de premier plan.

En ce qui concerne le reste des difficultés citées, en particulier dans le modèle du sujet, un signal instable (en profondeur et dans le temps) pourrait conduire à une mauvaise identification des personnes. Premièrement, pour la même raison que dans la modélisation d'arrière-plan, il n'y aura pas la même quantité de pixels pour représenter une personne. Deuxièmement, les contraintes du modèle pour caractériser et identifier les personnes sont directement liées à l'indicateur de GSS. Par conséquent, quand la valeur GSS est élevée (une basse stabilité temporelle), les contraintes pour ré-identifier une personne doivent être relâchées. Par exemple, deux images de la même personne, obtenues à des instants différents peuvent être identifiées comme des personnes différentes à cause de l'instabilité du signal et des contraintes très strictes au moment de la comparaison.

TABLEAU 2.5 – Résultats globaux pour les séquences de profondeur

Pixels	Valide %	Valeurs extrêmes %	Invalide %	GSS(mm)
Xtion Pro	98,3	0,5	1,2	0,77
R200	66,6	13,2	20,1	14,53
Etron	67,4	17,3	15,3	6,19
Parrot	90,9	5,7	3,4	20,64

Dans le tableau 2.5, nous observons les résultats globaux pour les séquences d'enregistrement de scènes de profondeur. On observe que la caméra Xtion a pu récupérer les 98 % de la scène, suivie par la caméra Parrot.

Le tableau 2.5 illustre le pourcentage de pixels valides, aberrants et invalides acquis par chaque caméra de la scène fixe. Les valeurs aberrantes sont les pixels avec un RMSE > 40 mm, une valeur de seuil où nous considérons une mesure comme instable. Les pixels invalides sont les pixels sans information de profondeur en raison d'un décalage ou d'une occultation. Enfin, le pixel valide est celui qui a des valeurs sous le seuil sélectionné et qui porte des informations de profondeur. D'autre part, la figure 2.24 présente le RMSE de 1800 trames de profondeurs consécutives et son histogramme pour chaque caméra évaluée. Un RMSE faible indique une carte de profondeur plus stable.

La caméra Parrot a la valeur GSS la plus élevée (tableau 2.5), ce qui indique que son signal est moins fiable pour la caractérisation humaine et le suivi. Cependant, le champ de vision du capteur à œil de poisson capte une grande partie de la scène. Un large angle génèrera des pixels invalides dans les régions latérales de l'image et les zones plus éloignées, ce qui conduit à des valeurs élevées de GSS. C'est pour cette raison que les régions latérales de l'image n'ont aucune signification (Fig. 2.24c) et produisent de nombreux pixels invalides. De ce fait, les pixels non valides ne sont pas pris en compte pour l'indicateur GSS. En revanche, le fait de disposer d'un champ de vision FoV plus grand permet de capturer des régions plus éloignées de la caméra, en augmentant la valeur de GSS.

La carte de disparité qui en résulte est composée de carreaux de 10×10 pixels, avec une profondeur constante dans chaque carreau, donnant une stabilité qui compense les erreurs introduites par le champ de vision plus large. Les objets plus proches ont un comportement stable, alors que le reste de l'image présente des problèmes de déformation et d'instabilité. En raison de

l'importante déformation radiale sur les optiques de l'œil de poisson, plusieurs pixels du bord de l'image ne portent aucune information (pixels noir sur la figure 2.24c).

La caméra Intel R200 présente le pourcentage le plus élevé de pixels invalides et la deuxième valeur de GSS la plus élevée (tableau 2.5). Elle occupe aussi le deuxième rang pour le champ de vision le plus large. Cette caméra R200 (voir figure 2.24b) présente des mesures instables même dans les surfaces planes des objets les plus rapprochés. De plus, la faible stabilité de la mesure de profondeur risque d'impacter directement la performance de modélisation du sujet (personne mobile à détecter et à suivre). Ainsi, pour être en mesure de ré-identifier une personne, nous devons comparer leurs propriétés. De cette manière, si les mesures d'une personne obtenues par la caméra changent constamment, leur modèle devrait également évoluer. Cependant, si nous élargissons la gamme des limites d'acceptation pour distinguer les personnes, cela impactera la robustesse du système d'identification des personnes. L'instabilité de la R200 dépend de la nature du processus d'estimation de la carte de profondeur, qui dépend de la texture projetée et de la configuration des paramètres de l'algorithme. Des mesures plus rapprochées sont plus stables, tandis que pour d'autres objets plus éloignés, ces mesures deviennent plus bruitées, jusqu'au point d'obtenir des pixels non valides représentés en noir (aucune disparité calculée).

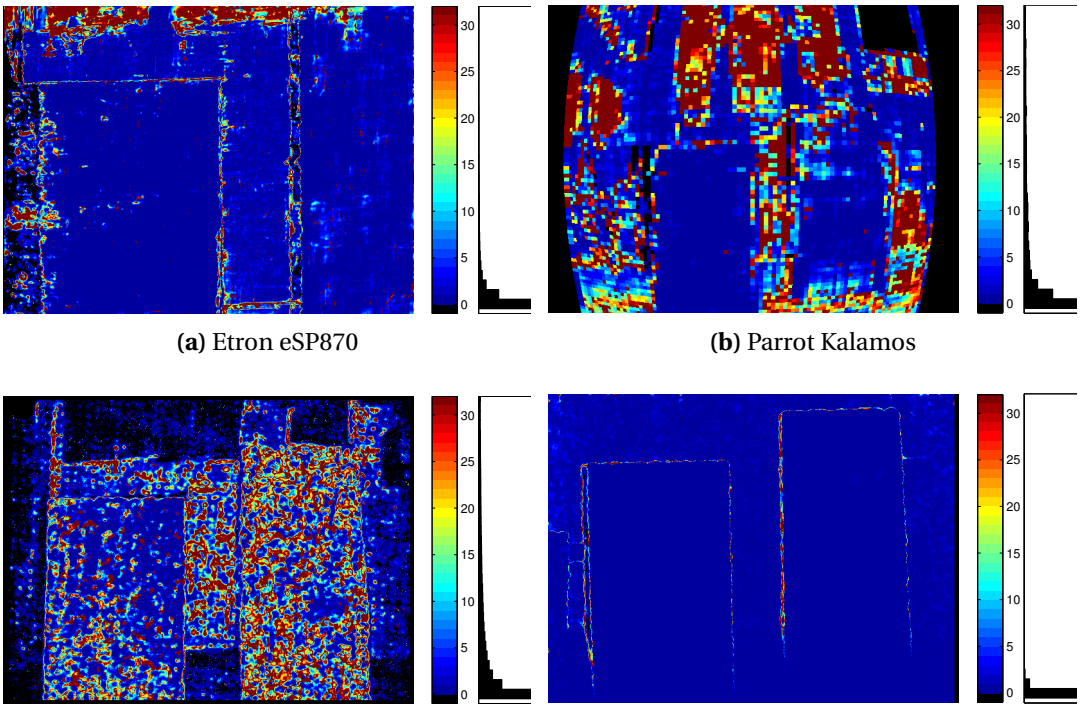
La caméra Etron a la plus petite valeur du champ de vision FoV et une valeur assez faible de GSS (tableau 2.5). Les mesures de profondeur des surfaces planes (présentées dans la figure 2.24d) sont plus stables que celles des caméras Parrot et R200. Nous avons trouvé les principales erreurs de mesure sur les contours et aux distances les plus éloignées (partie supérieure de l'image). D'autre part, de nombreux pixels n'ont pas de données de profondeur sur une bande verticale étroite sur le côté droit de la carte de profondeur, en raison du calcul de disparité.

La caméra Xtion présente une valeur très faible de GSS (tableau 2.5) et un champ de vision d'ouverture moyenne. Les images présentées dans la figure 2.24a montrent que les variations de mesure se trouvent dans la zone de contour des objets, et que les images des surfaces planes sont significativement stables. Globalement, on observe que la majorité des pixels de l'image contiennent des informations de profondeur valides.

Si l'on essaie de tirer une conclusion extrêmement rapide sur notre expérience, on peut affirmer que l'instabilité de mesure majeure se trouve à la frontière des objets en raison de l'occultation et des pixels "volants". Les caméras stéréo ont moins d'informations en raison de la nature de la technologie. De plus, les mesures de la caméra Etron ont été jugées plus stables que leurs homologues (caméras stéréo) en dépit des valeurs de champ de vision plus petites. La caméra R200 a un champ de vision plus grand que celui de la caméra Etron, pourtant la stabilité du signal de la R200 diminue à mesure que la distance à la surface s'éloigne de la caméra. D'autre part, le champ de vision plus large de la caméra Parrot permet de couvrir de plus grandes zones, mais la carte de profondeur sous-échantillonnée grossièrement empêche une détection d'humains de manière détaillée.

Enfin, ce test montre que la caméra Xtion Pro produit les meilleurs résultats. Elle couvre presque entièrement la scène observée (en proportion de pixels valides), et elle fournit des images plus stables que les autres caméras. Cependant, la stabilité des mesures ne garantit pas leur exactitude, par conséquent, nous avons jugé nécessaire d'évaluer la précision et la résolution de la profondeur.

Dans l'illustration de la figure 2.24, la scène est composée de deux surfaces rectangulaires plates à 2 et 4 mètres de la caméra. Chaque image RMSE utilise la même carte de couleur dans la plage 0 ... 30 mm. Les pixels aberrants (avec une erreur de profondeur supérieure à 30 mm) sont en rouge foncé et les pixels non valides (sans données de profondeur du capteur) sont en noir.



(a) Etron eSP870 **(b)** Parrot Kalamos

(c) Intel R200 **(d)** Xtion Pro

FIGURE 2.24 – RMSE et histogrammes associés de 1800 trames de profondeur acquises par différentes caméras regardant la même scène statique.

2.6.3 Précision de la distance mesurée et résolution de la profondeur

La précision de la profondeur des systèmes basés sur la vision stéréo (suivant le principe de la triangulation présenté dans la section 2.2.2) est affectée par différentes sources d'erreurs, y compris l'étalonnage de la caméra ou l'échec de la mise en correspondance. Par conséquent, pour une meilleure caractérisation de la caméra [KNO11], il est nécessaire d'évaluer la précision d'une caméra de profondeur de deux façons : a) la résolution en profondeur et b) la précision absolue des mesures de profondeur. Comme nous l'avons montré dans la section de reconstruction par stéréovision, on observe que la résolution de profondeur stéréo (éq. 2.6) augmente quadratiquement par rapport à la distance à la caméra [KAW08]. Par conséquent, l'erreur entre la profondeur mesurée et la profondeur réelle augmente avec l'éloignement de la caméra. Nous avons donc effectué cette expérience pour évaluer la précision de la distance mesurée, la résolution de la profondeur ainsi que la portée nominale de la caméra.

Procédure

Dans une deuxième expérience, nous avons placé l'axe optique du capteur 3D de manière orthogonale à un mur de référence, et nous avons fait varier la distance entre le capteur et le mur. Nous avons ainsi procédé à plusieurs mesures entre 80 cm et 520 cm du mur, en variant la distance par un intervalle de 40 cm. Pour chaque point de mesurage **s**, nous avons acquis T cartes de profondeur d'une surface plane afin d'éviter *pixels volants*. Dans le même temps, nous nous sommes assurés : que les pixels mesurés étaient à la même distance du capteur, et que les plans de la caméra et du mur étaient parallèles. Pour chaque point de mesurage, nous estimons l'erreur de précision E_s des données de profondeur pour une région spécifique R^c comme étant la différence entre la profondeur moyenne, dans le temps, de cette région D_s et la distance mesurée manuellement d_s :

$$\mathrm{E}_s = \left[\frac{1}{|\mathrm{R}^c|} \sum_{p \in \mathrm{R}^c} \bar{\mathrm{D}}_s(p) \right] - d_s \tag{2.10}$$

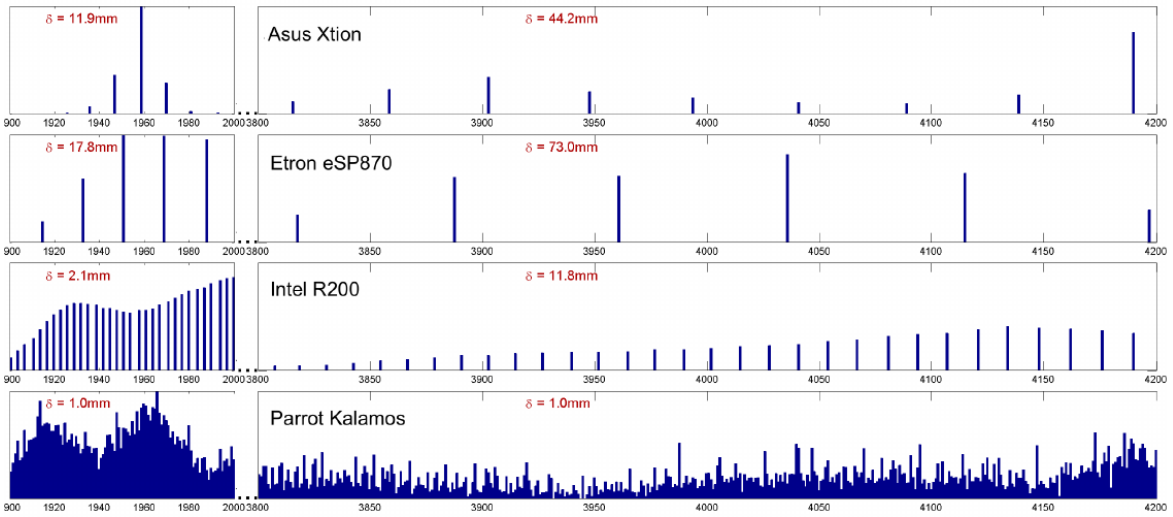


FIGURE 2.25 – Histogramme des données de profondeur pour plusieurs caméras.

Résultats des tests

Nous présentons sur la figure 2.25 les résultats pour les quatre caméras considérées. Nous avons fait un zoom sur un histogramme de la mesure acquise pour chaque caméra dans l'intervalle de 1900-2000 mm et de 3800-4200 mm. On peut observer comment la résolution de profondeur augmente avec de grandes distances. Ce comportement est visible dans presque toutes les caméras, à l'exception de la caméra Parrot. Nous attribuons ce comportement à l'algorithme de profondeur pour créer des pavés, il donne comme résultat une valeur moyenne pour tous les pixels du pavé. Cela signifie que même si les valeurs possibles à obtenir à certaines distances ne sont pas possibles en utilisant le principe de triangulation (Eq. 2.10), le processus de moyenner les valeurs de pixels du pave produit des valeurs *mesurées* entre les possibles valeurs *théoriques* de la résolution de profondeur. En général, la résolution de profondeur de caméra évaluée répond aux exigences de caractérisation des personnes. Même dans le pire des cas, l'erreur de mesure possible liée à la résolution de profondeur est de 36,5 mm (la distance moyenne entre les étapes) pour la caméra Etron. Cependant, nous devons évaluer la précision des mesures.

Dans la figure 2.25, les fenêtres de zoom de 1900-2000 mm et de 3800-4200 mm montrent la précision locale de la profondeur pour chaque technologie de caméra. L'axe vertical utilise une échelle normalisée non typée La figure 2.26, montre que la caméra Parrot a la pire performance

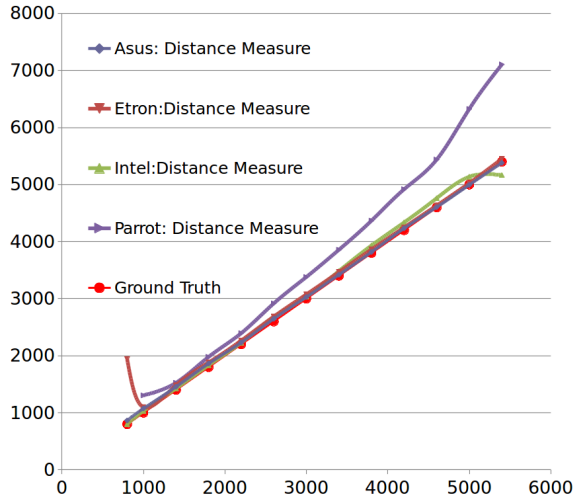


FIGURE 2.26 – Précision sur la distance mesurée E_s par les caméras en fonction la distance réelle.

pour mesurer les distances après 3 m par comparaison avec les trois autres caméras, qui présentent une mesure correcte de la distance d_s à l'intérieur de leur portée nominale (entre 1 m et 5 m).

2.6.4 Fiabilité de la détection des personnes

L'objectif principal de cette étape d'évaluation est de sélectionner la caméra la plus fiable pour caractériser et suivre les gens. Par conséquent, nous avons développé un test permettant de mesurer la fiabilité de détection humaine (***H**uman **D**etection **R**eliability* : HDR) pour les quatre caméras.

Procédure

La valeur de HDR est mesurée par la quantité de pixels de profondeur acquis et associés à une personne dans le champ de vision, par rapport à la quantité réelle de pixels que la personne occupe dans l'image (en utilisant des caméras RGB ou auxiliaires du capteur 3D). Du fait des avantages décrits dans la section 2.4, nous avons choisi de mettre la caméra en position zénithale dans ce test. Nous avons filtré les pixels sous le seuil de 120 cm du sol pour détecter la tête et les épaules de la personne. L'influence de perspective est mesurée en analysant la taille de la région segmentée pour une personne à différentes positions dans le champ de vision et en regardant différentes directions. Dans cette procédure, nous avons divisé le champ de vision en trois régions égales R_n horizontalement, comme illustré dans la figure 2.27a. Nous avons sélectionné quatre directions différentes qui, par symétrie, permettent de couvrir toutes les directions possibles qu'une personne peut prendre dans le champ de vision d'une caméra en position zénithale. Les directions ϕ_m, $m = 1,\ldots,4$ sont illustrées dans la figure 2.27b.

Chaque échantillon S_m^n consiste à placer une personne dans chaque région R_n en regardant chaque direction ϕ_m. A chaque position, on obtient une image de couleur et de profondeur synchronisée. Nous avons veillé à ce que toutes les images S_m^n contiennent la personne et l'arrière-plan. Nous avons utilisé une couleur facile à identifier pour l'arrière-plan afin de faciliter la segmentation de la personne dans l'image couleur. Ensuite, nous avons compté les pixels résiduels pour obtenir la quantité de pixels attendues $P_x(S_m^n)$. Nous avons alors répété le processus de segmentation de la personne et les pixels la représentant dans l'image de profondeur afin d'obtenir la personne acquise en pixels $P_a(S_m^n)$. Enfin, nous avons comparé la quantité de pixels comptés dans chaque image, calculant le taux de pixels P_a/P_x. De même, nous comparons la quantité de pixels obtenus à partir d'une même personne dans des positions différentes, mais en regardant dans la même direction.

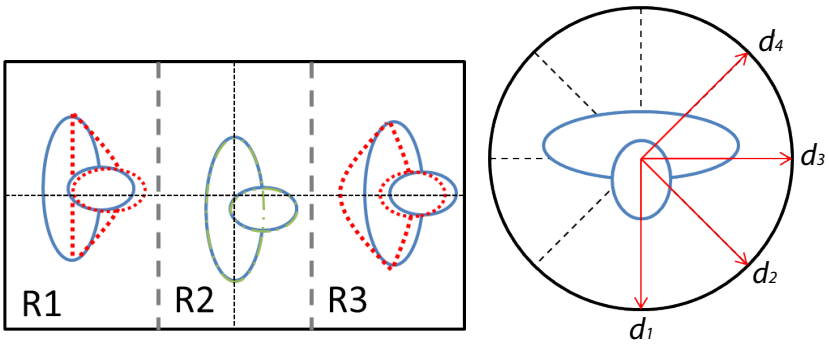
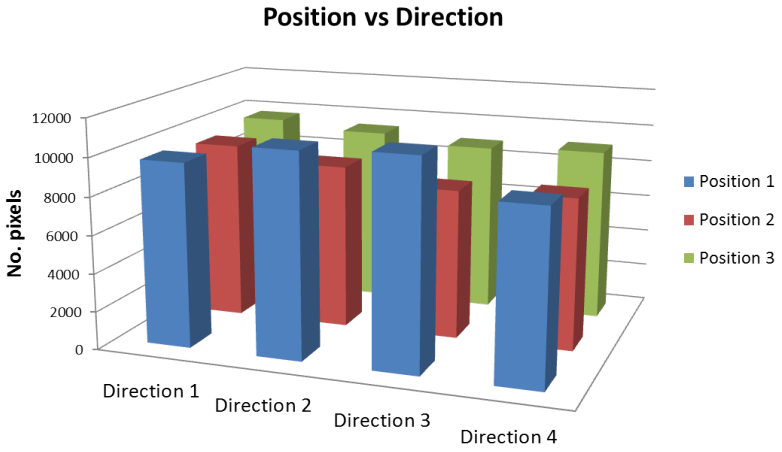


FIGURE 2.27 – Régions et directions utilisées dans les tests.

Dans la figure 2.27, on observe dans la première image la partition approximative du champ de vision d'une caméra 3D tournée vers le bas dans trois régions R_n. Chaque région a un comportement différent par rapport au mécanisme d'auto-occultation. Les régions en pointillés rouges ont une exposition privilégiée à la caméra. Dans la deuxième image, on observe les directions selon quatre directions avec différents angles ϕ_m prises par la cible humaine pendant le test de détection.

TABLEAU 2.6 – Rapport de détection humaine entre régions, par rapport à la région 2

	Région 1	Région 3
Etron	1,02	1,18
Parrot	1,34	1,92
Intel	1,39	1,33
Asus	1,14	1,23



(a)

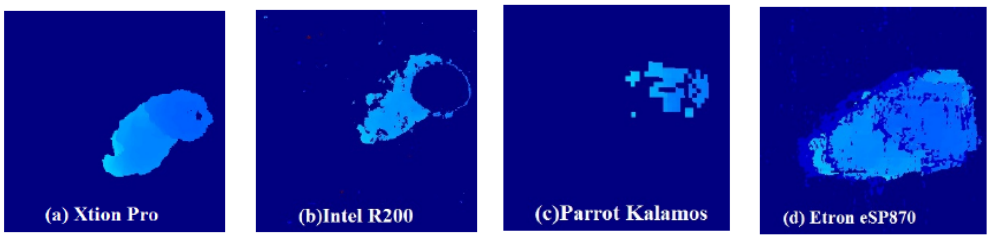
FIGURE 2.28 – Diagramme à barres de pixels acquis d'une personne observée dans les différentes combinaisons de directions et de positions évaluées.

Résultats des tests

La comparaison prévue entre les pixels couleur et les pixels de profondeur n'a pas été possible parce que la segmentation couleur des personnes, même avec un fond uniforme, présentait beaucoup de bruit en raison des ombres projetées par le soleil et d'autres sources de lumière. Une autre option pour effectuer cette évaluation pourrait être un étiquetage manuel des pixels. Pourtant, nous n'avons pas eu besoin de corriger la segmentation puisque le résultat des images filtrées (Fig. 2.28) montre une efficacité suffisante pour détecter la forme des personnes.

Nous présentons dans la figure 2.28 l'évaluation de l'influence de la position et de la direction par rapport à la quantité de pixels associés à une personne à l'intérieur du champ de vision de la caméra Xtion. Chaque colonne représente la quantité de pixels segmentés qui sont associés à une personne regardant dans la direction D_n et située dans la région R_n. Dans ces résultats, nous avons observé systématiquement l'auto-occultation générée par les parties du corps proches de la caméra (comme la tête ou les épaules) qui masquent le reste du corps (comme la poitrine ou les jambes) spécialement dans la région R_2 (région centrale illustrée dans la figure 2.27a) dans toutes les directions (Fig.2.27b). Au contraire, dans les régions éloignées du centre de la caméra (R_1 et R_2), on observe un petit effet de positionnement latéral introduisant des pixels qui n'appartiennent ni à la tête, ni aux épaules, et qui vont introduire des différences dans la caractérisation de la même personne. Ces pixels peuvent être filtrés pour obtenir des résultats plus précis. Cette différence est présentée dans le tableau 2.6 qui est composé par le ratio entre les pixels obtenus de la personne observée dans la région centrale R_2 et les pixels obtenus dans les régions latérales, R_1 et R_3 de la figure 2.27.

Dans la figure 2.28, on présente un diagramme à barres ou chaque barre représente le nombre moyen de pixels acquis dans une position et une zone fixe. Cette représentation nous sert à visualiser facilement l'impact qu'il existe entre les pixels acquis, la position et la direction du regard d'une personne. Cette figure nous sert également à visualiser la différence possible entre les modèles du sujet de la même personne dans les différentes positions et régions à l'intérieur du champ de vision de la caméra.



(a) **(b)** **(c)** **(d)**

FIGURE 2.29 – Différentes images de 300x300 pixels de profondeur de la même personne. Les pixels non valides (non-donnés) ont une couleur bleu foncé.

La figure 2.29 représente des images en profondeur d'une même personne dans la même position et la même direction pour les quatre caméras. Nous remarquons que la caméra Intel R200 ne détecte pas bien le haut des têtes et que la caméra Parrot montre des images de profondeur de faible résolution (en raison des pavés de 10×10 pixels) qui ne permettent pas de détecter des personnes. Les caméras Etron et Intel R200 sont capables de fournir plus de pixels par personne ; cependant, les surfaces acquises ne sont pas contiguës, en raison de l'absence de texture dans des régions comme les cheveux ou les épaules. Enfin, la caméra Asus Xtion Pro montre que les images affichent les parties du corps humain de manière contiguë et complète.

Ces résultats sont les plus importants pour l'objectif principal de cette thèse, à savoir la capacité de détecter et suivre des personnes. Cette expérience évalue la caméra qui correspond le mieux à nos besoins pour détecter une personne en vision de dessus. A partir de ces résultats, nous déduisons que les meilleures caméras sont l'Asus Xtion Pro et l'Etron eSP870, qui présentent la meilleure fiabilité pour être un point d'entrée dans le processus de détection d'une personne et extraire les propriétés observables (Fig. 1.1).

2.7 Sélection finale de la caméra

Nous avons présenté un ensemble de propriétés caractérisant les caméras RGB-D. Chaque propriété a été évaluée avec un ensemble de procédures qui peuvent être répétées et exécutées avec n'importe quel nouveau capteur de profondeur. Nous avons comparé et évalué un ensemble de caméras récemment mises sur le marché et présenté les résultats de notre analyse. Dans les différents tests, la caméra **Xtion Pro** obtient les meilleurs résultats avec un signal de profondeur plus stable pour l'acquisition de la carte de profondeur de la scène et une meilleure précision sur la distance mesurée. Sa portée nominale de profondeur est entre 80 cm et 5 m, une plage correcte dans notre contexte industriel pour le comptage de personnes où les plafonds se situent entre 2.5 m est 5 m selon l'expérience des installations de l'entreprise Shopline. Par conséquent, nous avons sélectionné cette caméra pour la suite de nos travaux de caractérisation des personnes et de leur suivi.

Un autre facteur important dans le choix du capteur était la popularité croissante de la caméra Kinect RGB-D de Microsoft (même technologie que l'Asus Xtion PRO). Des travaux basés sur cette caméra ont été publiés en 2010 et elle a été utilisée en recherche en vision par ordinateur [BMNK13, MR13], d'abord en raison de son faible prix,mais aussi pour ses images de profondeur précises et efficaces. De même, des travaux similaires ont été publiés [BKIM13] à l'aide du capteur Kinect. Ces travaux ont influencé nos orientations pour la sélection de notre capteur.

2.8 Conclusions

Pour résoudre notre problématique de détection et suivi des personnes dans les espaces publics, nous avons évalué la pertinence des données fournies par trois familles différentes de capteurs (portables, non-optiques et optiques), appropriés pour notre recherche. Dans ces grandes familles, nous avons mis en évidence le fait que les *capteurs portables* présentent le désavantage d'imposer à la personne suivie de les porter en permanence. Cela présente l'inconvénient d'impacter le comportement et surtout d'être en violation des principes de *non-intrusivité* et *d'ubiquité*.

Les *capteurs non-optiques* présentent, eux, comme désavantage leur coût élevé et leur application fortement spécialisée (par exemple la résonance magnétique nucléaire). Certains capteurs apportent des informations insuffisantes pour caractériser les humains (par exemple le tapis matrice de capteur de pression au sol), et ne permettent pas *de différencier les gens* ni de les *modéliser*.

Enfin, dans la famille des capteurs *optiques*, on trouve les capteurs 2D et 3D. Ce type de capteurs est non-intrusif et donne des informations riches et bien adaptées pour caractériser des personnes présentes dans une scène. Cependant, les capteurs 2D présentent des problèmes de fausses détections des objets, liés aux changements radicaux de l'illumination et à la complexité de la modalisation de l'arrière-plan. Par contre, les capteurs 3D fournissent des images de profondeur et ne sont pas perturbés par les ombres des personnes, s'adaptant aux fortes variations d'éclairage. De plus, la modélisation d'arrière-plan devient une tâche moins complexe, grâce au filtrage par la distance au capteur. Nous avons défini ce qu'est une image de profondeur. Celle-ci représente la distance des objets à la caméra. Nous avons alors décrit les différentes approches (passives et actives) pour obtenir des images 3D exploitant les méthodes par lumière structurée, stéréo vision et stéréo active. Nous avons ensuite présenté l'influence de la position de la caméra (IPC) par rapport à la scène à observer : dans la relation de la distance entre la caméra et les personnes, dans l'acquisition des données, et dans les grands espaces publics. La pertinence d'utiliser la caméra en position zénithale plutôt qu'en position latérale est mise en évidence par les arguments suivants :

- Par rapport à l'IPC dans la relation de la distance entre les caméras et les personnes, nous pouvons dire que le modèle géométrique de la caméra en position zénithale est plus simple que celui d'une caméra en position latérale. Par conséquent, le processus d'extraction géométrique est simplifié. La propriété de la similitude géométrique directe fait que nous avons un champ de vision symétrique qui réduit la complexité de la mise à jour du *modèle du sujet* et la *variation de l'échelle*.

- Par rapport à l'IPC dans l'acquisition de données, l'utilisation des images orientées vers le bas présente un avantage car la majorité des points acquis de l'image appartiennent à la tête et aux épaules. Ceci implique moins de puissance de calcul, donc moins de temps de calcul pour avoir une segmentation optimale des parties du corps. Cependant, il manque l'information sur les autres parties du corps qui pourraient être utilisées pour identifier la personne. Par contre, l'utilisation des images en vue latérale implique la création d'algorithmes plus complexes et robustes pour filtrer les pixels associés à d'autres parties du corps que la tête et les épaules, et avoir une segmentation fiable.

- Par rapport à l'IPC, dans le cas où l'utilisation de plusieurs caméras soit nécessaire pour observer une scène, le positionnement de la vue latérale crée de grandes zones de chevauchement, ce qui signifie que plusieurs caméras vont regarder de manière redondante la même partie de la scène. Dans ces zones, la gestion de mise à jour du modèle des personnes et le transfert des informations sur les personnes détectées seront plus complexes en raison des différents points de vue (par exemple vues avant, arrière et latérales de la même personne dans les différentes caméras). En même temps, cette position génère plus d'occultations fortes et de zones aveugles dans des scénarios avec une ou plusieurs caméras. En conclusion, les grandes zones de chevauchement ou d'occultation créent des ambiguïtés dans la localisation et l'identification des personnes.

Le point pertinent de nos expérimentations est que la position zénithale réduit l'occultation entre les personnes, diminue le chevauchement de FoV des caméras, facilite la séparation et la caractérisation des personnes et diminue les calculs.

Nous avons réalisé la comparaison des différentes caméras 3D du marché (section 2.5), utilisant différentes technologies. Puis, nous avons sélectionné les caméras disponibles sur le marché selon les critères suivants : le système de capteurs doit acquérir des images en temps-réel, une solution récente à faible coût et une capacité d'intégration dans une solution embarquée (taille et traitement dans la caméra) de manière à ce qu'il soit autonome, facile à installer et évolutif.

Sur la base de cette expertise, nous avons dressé un ensemble de caractéristiques montrant la pertinence de l'utilisation de chaque caméra 3D dans certains domaines d'application, en fonction de ses propriétés.

Dans le groupe de caméras retenu (L'ASUS Xtion PRO, l'Intel R200, l'Etron eSP870 et le Parrot S.L.A.M. Dunk), nous avons caractérisé chacune d'elles en utilisant trois expériences, « la stabilité temporelle du signal de profondeur », « Précision de la distance mesurée et résolution de la profondeur », et une expérience spécifique à notre domaine d'usage de caméras 3D pour le comptage des personnes « Fiabilité de la détection des personnes ».

Les résultats de ces expériences nous ont permis de sélectionner la caméra la plus appropriée pour détecter et suivre des personnes en mouvement dans des grands espaces publics. L'utilisation de la caméra ASUS comme dispositif d'acquisition nous permet de répondre aux difficultés exposées dans l'introduction telles que les *variations de l'environnement* et la détection *non-intrusive*.

Chapitre 3

Suivi des personnes en mouvement

Sommaire

Dans ce chapitre, nous allons présenter la démarche adoptée pour obtenir une solution embarquée fiable au problème du suivi de personnes en mouvement. Pour cela, nous utiliserons un capteur 3D placé en position zénithale. Cette démarche est basée sur deux aspects importants : les algorithmes de traitement et l'architecture de calcul du système embarqué. Ces algorithmes ainsi que l'architecture de calcul répondent à nos besoins (évoqués dans la section 1.3). Nous respectons également les contraintes industrielles imposées dans le cadre de cette thèse CIFRE, en particulier la construction d'une solution à faible coût. Pour suivre les personnes dans une scène, nous devons extraire les propriétés observables (Fig.1.3) de ces personnes, vues dans l'introduction, en se focalisant sur les propriétés spatio-temporelles. Il s'agit de réduire la complexité de ces algorithmes tout en optimisant l'architecture de calcul afin de respecter les contraintes de ressources limitées de notre système. Cet équilibre nous permet d'atteindre la plus haute précision de détection des personnes tout en minimisant la complexité. Cela va constituer le point central de ce chapitre.

Nous passerons ainsi en revue l'état de l'art du domaine du suivi de personnes dans une scène et le concept (de plus en plus répandu) de caméra intelligente. Nous décrirons notre méthodologie et nos algorithmes proposés pour le suivi des personnes sur la base de notre architecture de la caméra intelligente « *smart camera* » développée dans le cadre de cette thèse CIFRE. Enfin, nous aborderons l'évaluation des résultats du suivi et de la caractérisation des personnes dans une scène.

3.1 État de l'art

Au cours des deux dernières décennies, un groupe important de chercheurs académiques et industriels ont étudié le suivi des personnes dans leur environnement naturel, en utilisant une variété de capteurs. Plusieurs domaines de recherche ont abordé ce problème, employant différentes méthodes et algorithmes en fonction des domaines applicatifs visés. Nous pouvons par exemple citer la vision par ordinateur [HSXS13], la robotique [HKH+12], l'informatique omniprésente [TJS10], l'interface homme-ordinateur et, plus récemment, le réseau de capteurs [ELYR14] et la cinétique humaine [SBR14].

Dans cette section, nous présentons les différents travaux associés à la détection de piétons, le suivi des personnes et les *smart cameras*, qui représentent les points-clés de base de notre approche. Cela nous permettra d'avoir une vue générale des différentes approches et d'évaluer les avantages et inconvénients liés à leurs utilisations dans le cadre de notre problématique.

3.1.1 Détection des piétons

L'application de la détection de piétons, fait l'objet d'une littérature abondante notamment en ce qui concerne les familles d'algorithmes, les jeux de données et les benchmarks (tests de performance). L'objectif est de détecter un piéton (ou d'autres objets, mobiles ou statiques) qui apparaît dans le champ de vision d'une caméra mobile (ou statique). En général, le processus consiste à extraire des vecteurs de caractéristiques « ***features*** » (ou vecteurs de descripteurs visuels de dimension élevée [Dal06]) des "objets/piétons" sur une « ***fenêtre glissante de détection*** » et à utiliser une technique de classification pour décider s'ils appartiennent à un piéton ou à un autre objet.

Dans le but d'aborder la détection à différentes échelles, on utilise une *analyse multi-résolution*. Cette analyse est effectuée par une *fenêtre glissante de détection*, qui est une région de taille constante, en se déplaçant à travers la *pyramide de l'image* (Fig. 3.2). Cette pyramide est construite à partir de la réduction de l'image à un facteur constant λ, en créant des niveaux l_i. Dans cette analyse, les opérateurs de chaque approche sont appliqués sur la fenêtre glissante de détection aux différents niveaux. Par exemple, l'algorithme original [VJS03] utilise un facteur de 0,8 de la résolution de l'image pour la construction de pyramides jusqu'à ce que la taille de l'image devienne plus petite que 20×15 pixels (taille des images sélectionnées arbitrairement par l'auteur).

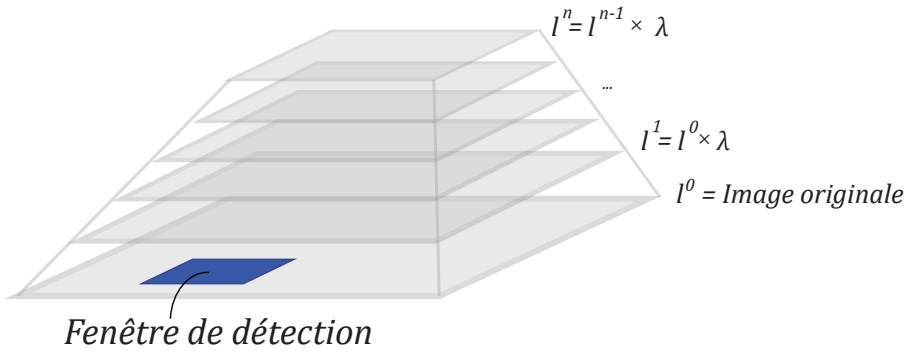


FIGURE 3.1 – Pyramide de l'image. Chaque niveau de l'image I est construit par un facteur de réduction λ.

Ensuite, les détections résultant de cette classification sont ensuite localisées via des boîtes englobantes (***B**ounding **B**oxes* : BBs) [BOHS14] qui représentent des rectangles autour des objets détectés. Ainsi, une boîte englobante définit les extrêmes horizontaux et verticaux de l'objet entouré, comme le montre la figure 3.2. En même temps, les BBs sont utilisées pour annoter des vidéos afin de fournir la localisation réelle, « vérité-terrain », d'un objet. Ceci se fera soit d'une manière manuelle soit d'une manière semi-automatique 3.2 sur les images. Les annotations sont utilisées pour faire une analyse comparative des résultats fournis par l'algorithme de positionnement avec la vérité-terrain.

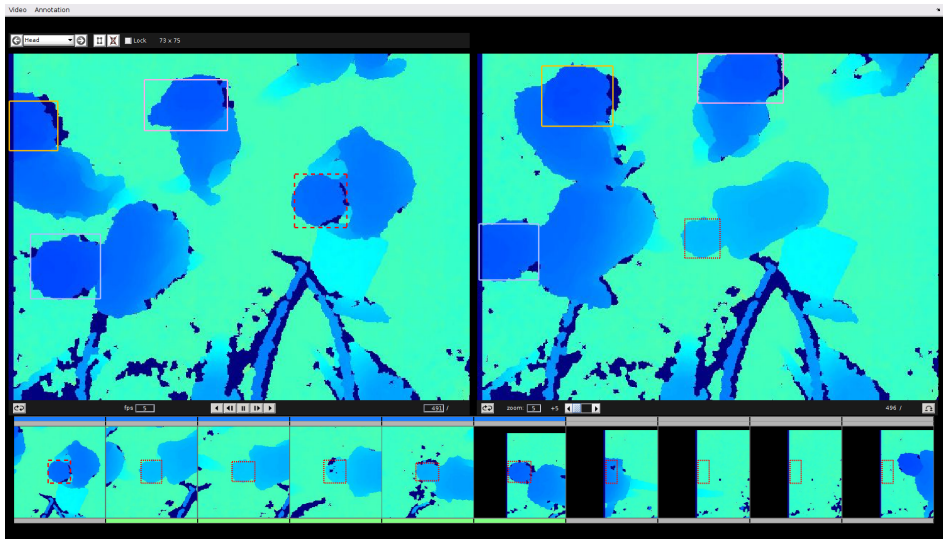


FIGURE 3.2 – Outil de Caltech pour l'annotation de vidéos en Matlab. A gauche l'image initiale, à droite l'image finale et en bas les images intermédiaires avec l'interpolation de la position de l'étiquette.

Des groupes de recherche du domaine de la détection de piétons ont créé plusieurs **jeux de données** pour tester toute méthode proposée et évaluer sa performance. Un jeu de données est composé par un ensemble d'éléments (images, vidéos, coordonnées ou autres) et des annotations pour chacun. Les *jeux de données* les plus couramment utilisés pour la détection des piétons sont l'INRIA [DT05], l'ETH [ELSVG08], le TUD-Bruxelles [WWS09], le Daimler[EG09], le Daimler Stéreo [KLG09], le CALTECH [DWSP09] et le KITTI [GLSU13]. Cependant, dans la majeure partie de la littérature, on teste la performance des algorithmes sur les jeux de données INRIA, CALTECH et KITTI. Les jeux de données CALTECH et KITTI fournissent une plus grande quantité de données ou des séquences vidéo plus longues. Tous ces *jeux de données*, à l'exception de l'INRIA, sont obtenus à partir de vidéos. Il est donc possible d'utiliser des propriétés spatio-temporelles (section 1.2) pour la détection des piétons. De plus, seuls les jeux de données Daimler Stereo, ETH et KITTI disposent des informations stéréo.

Concernant le contenu des jeux de données, ceux d'INRIA [DT05] contiennent 1805 images de personnes statiques, habituellement debout, dans plusieurs orientations avec une grande variété

des fonds d'image. Ces images sont de taille 64 × 128 pixels et ne font pas partie d'une séquence temporelle (vidéo par exemple). Le jeu de données « Caltech Pedestrian Dataset » [DWSP12] contient, quant à lui, une vidéo d'une résolution de 640 × 480 pixels à 30Hz, avec 10 heures d'enregistrement à partir d'un véhicule en mouvement dans une zone urbaine, accompagnée de trois types d'annotations (piétons, groupe de piétons, et objets non-définis). Enfin, les jeux de données KITTI [GLSU13] offrent 6 heures de vidéo, produite par différents types de capteurs (caméras stéréo en couleur ou en échelle de gris, GPS / IMU et LIDAR). Ces jeux de données fournissent aussi des scénarios de tournage assez diversifiés (route, ville, résidentiel et campus). De plus, à cause des différents types de capteurs et de ses dimensions, les auteurs ont été obligés de remplacer les BB par des « trackles ». Les trackles sont des *boîtes englobantes* 3D qui contiennent des coordonnées (pour lesquelles le repère est le LIDAR), le type d'étiquette (voiture, fourgon, camion, piéton, personne (assis), cycliste, tram et divers fait référence aux objets comme des caravanes, segways et autres), la taille 3D (hauteur, largeur, longueur) et autres. Chaque *jeu de données* propose un kit de développement et d'évaluation pour comparer les résultats des algorithmes de détection des personnes avec des images annotées.

Par ailleurs, on trouve une grande richesse de méthodes dans les travaux sur la détection de piétons, surtout focalisés sur les applications automobiles. Par exemple, l'utilisation des vecteurs de caractéristiques pour le suivi de piétons a été introduite par [VJS03] en utilisant des filtres *d'apparence* sur chaque image et des filtres de *mouvement* sur une paire d'images consécutives, qui nous permettent de trouver la position de la fenêtre de détection (une région de taille fixe qui appartient à l'image) d'une manière plus précise. Finalement, on réalise une classification en cascade (c'est-à-dire que les classificateurs utilisées sont disposés en série, du plus simple au plus complexe, dans un souci d'efficacité) de cette fenêtre, en décidant si l'on trouve ou non une personne à l'intérieur de la fenêtre.

Dans [DT05], on utilise les histogramme de gradients orientés (***H**istogram of **O**riented **G**radients* : HOG) pour extraire des vecteurs de caractéristiques par blocs sur la pyramide de l'image. Ensuite, on utilise une support vector machine (SVM) linéaire comme système de décision pour déterminer s'il y a des personnes sur l'image (HOG + SVM). Dalal a nommé « *recherche 3D* » la recherche sur différents niveaux de la pyramide, où chaque position 3D est exprimée par 3 coordonnées qui correspondent aux axes X et Y de l'image et au niveau l de la pyramide (Fig 3.1). Finalement, on agglomère les détections (scores positifs donnés par la classification de la SVM) à différents niveaux de la pyramide de l'image en utilisant une « *recherche 3D* » pour fournir une détection localisée dans l'image et pour éviter des détections multiples de la même personne.

La SVM utilisée par Dalal a été entraînée, à partir d'une base de données d'apprentissage, une première fois avec ces images positives (qui contiennent une personne) et négatives (sans aucune personne à l'intérieur de l'image), puis a repassé les images négatives complètes en passant sur toutes les fenêtres de détection possibles. Dalal utilise toutes les fenêtres de détection dans lesquelles la classification est incorrecte pour recycler le classificateur encore une fois et ainsi améliorer les résultats. Par conséquent, la classification est entièrement dépendante des données [Dal06].

Le travail modèle de pièces déformables (***D**eformable **P**art **M**odel* : DPM) de [FMR08], est fondé sur la détection de personnes à partir de HOG. L'amélioration principale de cette approche a été l'introduction d'un modèle de pièces déformables mobiles (déformations non rigides) du corps humain (membres et tête) avec une structuration des parties du corps. On utilise un filtre initial appelé **filtre racine** (comme dans Dalal HOG) pour identifier les personnes mais aussi des filtres partiels à plus haute résolution pour chacune des différentes parties du corps et avec une *matrice de coût de décalage* associée à chaque membre du corps, nommée modèle de déformation. Ce modèle (racine et pièces) aide à traiter la variabilité intra-classe (des personnes dans différentes positions ou différents types d'automobiles : voiture, van, sportif), la variation des points de vue et de l'éclairage. Ce modèle sera utilisé dans l'apprentissage d'une SVM latente. Une SVM latente est une machine de classification particulière pour laquelle il n'y a pas de relation directe (explicite) entre les données d'entrée et les résultats de classification à la sortie. Pour exprimer cette relation plus complexe, la machine a besoin de variables internes (non-explicites) qui peuvent définir d'une manière satisfaisante la relation (complexe) entre l'entrée (une image) et la sortie (les

détections de la personne et les parties du corps).

Benenson [BOHS14] a analysé les travaux des dix dernières années (2004-2014) sur la détection de piétons et introduit une courte présentation chronologique des principaux paradigmes de détection des piétons comme les trois derniers cités (VJ, HOG et DPM). De plus, Benenson surligne qu'actuellement, la détection de piétons se réalise par les réseaux de neurones convolutionnels [OW13, SKCL13] et les arbres de décision renforcés arbres de décision renforcés (***D**ecision **F**orest* : DF) [BOHS14, DWSP09]. Ces deux derniers paradigmes sont très coûteux en termes de temps de calcul et nous éloignent de nos objectifs industriels (calculateur bas coût).

De manière générale, les processus de détection des personnes permettent de décider si une personne se trouve dans une partie de l'image. Les algorithmes utilisés se déclinent en deux étapes :

- L'étape d'extraction des vecteurs de descripteurs visuels pour localiser une région qui contient éventuellement une personne.
- L'étape de classification de ces régions qui contiennent ou non une personne.

Conclusions

A partir des travaux présentés, nous remarquons l'utilisation d'annotations pour évaluer la précision des algorithmes, l'extraction des signatures (*features*) pour caractériser les personnes et une structuration du corps humain.

Cependant, le principal inconvénient de ces approches de détection des piétons est qu'elles sont dépendantes de la base de données d'apprentissage utilisée. Dans notre contexte d'application, on doit compter les personnes dans des magasins. D'un côté, les différentes marques commerciales essaient de transmettre une *expérience client unique* dans chaque magasin. Elles mettent tout en œuvre pour se démarquer des concurrents. En conséquence, chaque magasin est complètement différent des autres, en termes de couleurs, de textures de meubles, d'agencement des espaces, etc. D'un autre côté, les approches présentées dans cette section dépendent de la diversité ainsi que de la quantité de données récoltées au moment de l'apprentissage pour détecter correctement les personnes.

Dès lors, cette dépendance est contradictoire avec l'objectif du magasin de se différencier le plus possible des autres. Par conséquence, l'utilisation de ces méthodes est difficile dans notre domaine. Enfin, ces approches sont très coûteuses en termes de calcul, allant également à l'encontre de nos objectifs industriels. Nous devons donc rechercher d'autres approches dans la littérature, plus appropriées à notre scénario d'application.

3.1.2 Suivi des personnes en mouvement

Dans le comptage des personnes, il existe plusieurs travaux associés aux caméras fixes, pour des foules dans les lieux publics et pour le comptage des personnes. L'objet principal est de détecter et de suivre les personnes en mouvement dans le champ de vision d'une caméra fixe qui observe une zone d'intérêt. On a regroupé les différentes contributions et approches pour le suivi et comptage des personnes sur deux points de discussion fortement reliés entre eux :

- La **conception de l'architecture physique du système** : le type de capteur pour l'acquisition des images, position de la caméra (traitée dans le chapitre précédent) et l'approche de la configuration matérielle.
- La **conception de l'architecture d'extraction des propriétés observables** : la chaîne de traitement classique de l'extraction, les différentes configurations de cette chaîne de traitement et les multiples approches pour l'extraction de chacune des propriétés.

Conception de l'architecture physique du système

La conception de l'architecture physique du système est basée sur le type de capteur, la position de la caméra et la configuration matérielle. Dans le chapitre 2, nous avons déjà exposé les avantages d'avoir une caméra 3D en position zénithale pour l'acquisition des images. Par ailleurs, les configurations du matériel ont des approches opposées : l'une fondée sur une *restriction des ressources*, l'autre sur des *systèmes de haute performance*. L'approche par restriction de ressources est une solution impliquant du matériel (habituellement les systèmes embarqués) à une faible puissance de calcul et une quantité de mémoire ***R**andom **A**ccess **M**emory* (RAM) limitée : cela revient à une solution à faible coût et à faible consommation d'énergie (une exigence de plus en plus importante dans la conception de solutions modernes à basse consommation électrique). D'autre part, un scénario basé sur la performance est une solution impliquant des ressources (puissance de calcul, mémoire RAM, interfaces d'I/O) qui ne limitent ni les algorithmes, ni les approches utilisées pour résoudre le problème posé.

Certains travaux [BKMQB16, TS08, YVC10] sont basés sur des solutions avec des contraintes de ressources. Ces travaux prennent en compte ces limitations et élaborent des algorithmes avec une faible localisation et caractérisation des personnes dans le champ de vision, en utilisant de multiples caméras 2D. Ces représentations (faibles) sont également utilisées pour créer des messages légers [TS08, YVC10] qui permettent d'établir des protocoles de communication rapides pour partager des informations avec ses voisines [BKMQB16]. La cohérence des résultats est basée sur la performance globale du système [BKMQB16, TS08].

Au cours des dernières années, plusieurs approches utilisant des capteurs de profondeur similaires à la Microsoft Kinect V1, ont été proposées [FML11, KAD+14, LLCC13, LZL+15, QLYWj10, Rau13, VZS13, YMKC08]. Ces derniers travaux de suivi des personnes, sont basés sur les approches axées sur la performance, en utilisant de puissantes machines de traitement. Ces travaux ont une précision spatiale (localisation) plus grande et une caractérisation plus complexe qui permet d'avoir de meilleurs résultats de comptage. De manière générale, ces algorithmes sont basés sur l'extraction des signatures de la tête et des épaules [KAV12] pour décrire une personne . En cas d'ambiguïté, ils utilisent des algorithmes de suivi reliés à l'historique des trajectoires [KAD+14].

S'agissant de la construction du matériel, on trouve des travaux académiques fondés sur les systèmes avec *restriction des ressources*. Ces travaux sont spécialisés sur des capteurs optiques associés à une architecture embarquée capable de traiter des images, de communiquer avec des dispositifs externes et de produire des informations sémantiques de haut niveau à la place de simples images, nommées *systèmes de caméras intelligentes (Smart camera systems)* [FBBS06, MM14, MB99, RRRC14, BES06]. Les principales difficultés pour la conception d'une telle caméra sont : l'analyse des images, la protection de la vie privée [RRRC14], l'utilisation de différents types de capteurs dans la même architecture (capteurs hétérogènes : [DSE+13]), le management et traitement des données de complexité sémantique de haut niveau [RRRC14, RW14, RJQ07].

Les systèmes de caméras intelligentes ont évolué d'une caméra isolée à des systèmes distribués de caméras intelligentes avec des caractéristiques collaboratives. Rinner [RWS+08] propose une étude sur la taxonomie (Fig. 3.3) des systèmes de caméras intelligentes, la classification de l'architecture du système et les difficultés au moment de construire ce type de solutions. Dans cette section, on analyse la taxonomie ainsi que les difficultés. Une discussion sur les architectures de systèmes de caméras intelligentes aura lieu dans le chapitre suivant.

La **taxonomie d'un système de caméras intelligentes** est composée des capacités de la plateforme, le degré de calcul distribué et l'autonomie du système (Fig. 3.3). Les *capacités de la plateforme* sont données par : le type de capteur optique (section 2.1) ; la capacité de calcul (embarqué) pour effectuer des tâches ; les moyens de communication pour transmettre des informations obtenues ; et les exigences d'alimentation électrique de la caméra intelligente.

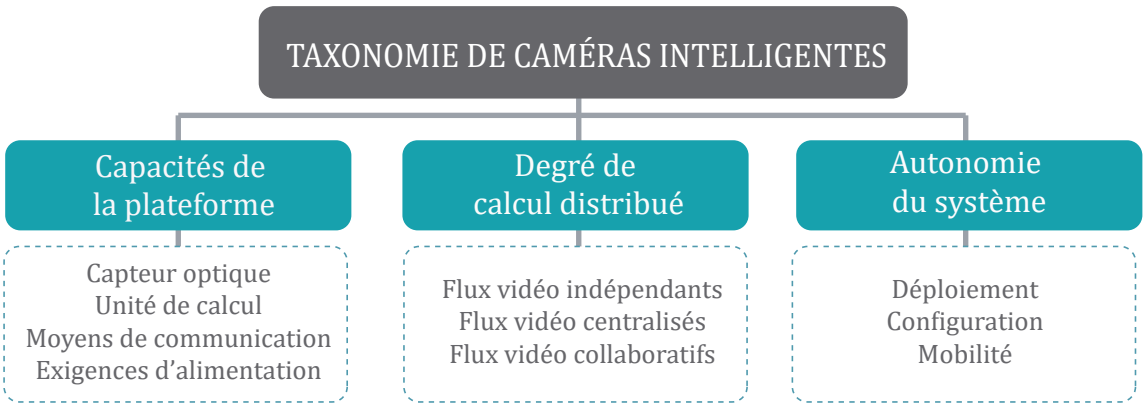


FIGURE 3.3 – Schéma de la taxonomie des caméras intelligentes tiré des travaux de [RWS+08]).

Le *degré de distribution du calcul* est classé comme un traitement de flux de vidéo de manière : i) individuelle où l'information générée par la caméra n'est pas reliée à des entités externes ; ii) centralisée où le traitement de l'information générée par les caméras est fait dans un serveur central ; ou iii) collaborative où le traitement de données est fait localement et l'information est générée en collaboration avec les autres caméras et de maniéré distribuée.

Finalement, l'*autonomie du système* qui fait référence aux moyens disponibles : du déploiement au moment de l'installation de la caméra ; de la configuration où l'on ajuste les paramètres de la caméra en fonction de l'environnement ; et de la mobilité de la caméra qui dépend de son usage, par exemple, les caméras de surveillance sont mobiles sur leurs axes. Ces dernières caractéristiques d'autonomie d'une caméra intelligente définies par [RWS+08] sont très similaires aux contraintes industrielles pour notre sujet de thèse.

Les ***capacités de la plateforme*** sont un point central de notre travail en commençant par le *type de capteur*, dont on a déjà parlé dans la section 2.1. De plus, la *capacité de calcul* est définie par l'unité de calcul à usage général telle que : l'unité centrale de traitement (***C**entral **P**rocessing **U**nit* (CPU)), les processeurs de traitement du signal (***D**igital **S**ignal **P**rocesssors* (DSP)), des réseaux logiques programmables (FPGA), des solutions (SoC) où de nouvelles architectures contenant des combinaisons de ces types d'unités de calculs [RWS+08]. La *capacité de communication* dépend du type de technologie utilisé dans le dispositif, par exemple : Wi-Fi, Ethernet, ZigBee, Bluetooth ou la plus récente LoRa, une technologie de télécommunication sans fil à longue portée dans le cadre de l'internet des objets (***I**nternet **o**f **T**hings* (IoT)). Finalement, la *capacité d'autonomie* d'un dispositif est donnée pour la source d'alimentation électrique. Actuellement, on a des dispositifs connectés aux réseaux électriques publics ou bien des dispositifs autonomes qui ont leur propre source électrique comme des batteries jetables ou rechargeables et d'autres sources complémentaires comme l'énergie solaire.

Ces capacités déterminent la performance globale du système. Par exemple, le taux de FPS du capteur détermine le taux de FPS maximal dont peut disposer le système en termes d'acquisition d'images. De la même façon, la capacité de calcul peut limiter le taux de fps des images traitées en fonction des tâches à effectuer. Ce taux devient alors le taux de fps final de la caméra intelligente pour aboutir à une fonctionnalité donnée (comme le comptage de personnes). Un autre exemple est l'autonomie électrique du dispositif alimenté par des batteries qui dépend, en partie, du type de technologie de communication utilisée. Une classification de plusieurs travaux publiés sur les systèmes de caméras intelligentes se trouve en annexe , celle-ci étant basée sur l'évaluation faite par [RWS+08]. Cette annexe présente également une description de la taxonomie de chaque système de caméras intelligentes.

Les difficultés de construction d'une caméra intelligente sont les suivantes : conception du matériel informatique, conception du logiciel du système, sécurité et protection de la vie privée ; adaptabilité et autonomie ; et partage de l'information et coopération entre les caméras (on l'appellera collaboration).

- La conception du *matériel informatique* fait référence à la sélection et à la construction des systèmes électroniques qui contiennent ou intègrent tous les éléments décrits par la taxonomie de la caméra intelligente. Sa construction représente la difficulté de mise en place des éléments électroniques. Pour cette difficulté, nous devons tenir compte des contraintes industrielles économiques (limitation des coûts).
- La conception du *logiciel du système* est composée de la gestion de ressources (qui repose souvent sur le système d'exploitation), du traitement d'images pour l'extraction des informations sur les images acquises, de la gestion de la communication et de la configuration de l'application.
- La *sécurité et la protection de la vie privée* fait référence à la façon dont on doit assurer l'inviolabilité (face aux différentes attaques potentielles) des données acquises par la caméra et/ou traitées et mémorisées par le système.
- L'*adaptation et l'autonomie* d'une caméra intelligente fait référence à la gestion de la configuration et la maintenance. Ce système doit avoir une opération et adaptation autonome par rapport à l'environnent où il est installé. De plus, on cherche à minimiser les opérations de maintenance sur la durée.
- La *collaboration* entre les caméras intelligentes (ou autres systèmes) est importante en raison de ces ressources limitées, spécialement pour les systèmes envisagés sur les grands espaces.

On observe que les *difficultés de construction* d'une caméra intelligente sont fortement liées à la gestion des *capacités de la plateforme*.

Conception de l'architecture d'extraction des propriétés observables

L'architecture conceptuelle de l'extraction des propriétés observables pour suivre les gens est basée, de manière générale, sur l'extraction des propriétés observables (Fig. 1.1) des personnes, en se focalisant sur les propriétés spatio-temporelles. Nous allons appeler le processus (et toutes ses étapes) d'extraction de ces propriétés comme la chaîne de traitement pour le suivi des personnes (Fig. 3.4). L'entrée de la chaîne de traitement consiste en une séquence de vidéo et la sortie en une liste des personnes associées à leurs trajectoires. C'est à l'intérieur de cette chaîne de traitement que nous devons trouver comment résoudre les difficultés (du suivi de personnes) d'une manière la plus précise possible et en même temps réduire au maximum la complexité des algorithmes pour respecter les contraintes de ressources limitées de notre système.

FIGURE 3.4 – Chaîne de traitement proposée par [TDS10] pour le suivi de personnes basé sur l'extraction des propriétés spatio-temporelles.

Nous considérons l'extraction des propriétés spatio-temporelles comme la base de l'architecture algorithmique du processus de suivi des personnes. Ces propriétés sont traduites en étapes plus fines. La chaîne de traitement (voir figure Fig. 3.4) commence par la présence, puis la segmentation, la localisation, l'identification dans le système (local ou global) et se termine par le suivi.

On définit le bloc de ***présence*** comme le sous-processus qui permet de détecter la présence d'une ou plusieurs personnes dans le champ de vision de la caméra. On relie normalement ce bloc à la détection du premier plan ou des pixels « utiles » pour le reste de la chaîne de traitement. Le bloc de ***segmentation*** est responsable du groupement des pixels utiles par région (blob). Une région est un groupe de pixels qui partage des caractéristiques spatiales (proximité) et/ou d'intensité similaires. Selon les cas, on peut grouper des régions non connectées parce que les

pixels appartiennent à la même personne ou alors on divise une même région parce qui contient plusieurs personnes. Le bloc de ***localisation*** estime la position des personnes, trouvées dans l'étape précédente, par rapport à la caméra ou à un système global de coordonnées en 3D ou 2D. Le bloc d'***identification*** permet d'abord de donner une étiquette à la personne la première fois qu'elle est détectée et ensuite de la ré-identifier pendant le temps où la personne se trouve dans le champ de vision de la caméra. Dans certains cas, le système peut permettre d'identifier la personne d'une manière unique grâce à des systèmes externes [TJS10], comme des badges ou des téléphones portables. Enfin, le bloc de ***suivi*** utilise toute l'information extraite, permettant de reconstruire la trajectoire de la personne dans le laps de temps où celle-ci est dans le champ de vision de la caméra.

Pour extraire ces propriétés, les blocs sont parfois associés à des noms différents dans la littérature, en effectuant l'extraction d'une ou plusieurs propriétés, en changeant leur ordre par rapport à la chaîne de traitement proposée. Dans les applications de comptage de personnes, nous ajoutons un bloc supplémentaire à la fin de la chaîne de traitement (Fig. 3.5) qui contient l'application finale (dans notre cas, la logique de comptage) [KCCK02, KAD+14, RB94, TS08, Che03]. Par exemple, dans [RB94] on divise le processus de comptage de personnes en 3 phases. Dans la première (phase d'alerte), il réalise la *détection* et la *localisation*. Dans la seconde (phase de suivi), il réalise l'identification de l'objet et le suivi. Enfin, dans la troisième (phase d'interprétation) il réalise la *séparation* et l'*application*. Dans ce cas, l'application est le comptage du nombre de personnes traversant une ligne virtuelle dans la zone surveillée.

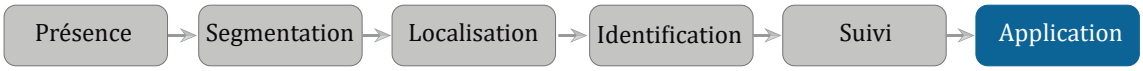


FIGURE 3.5 – Chaîne de traitement classique d'un processus de suivi de personnes avec le rajout d'un module additionnel « Application ».

Dans la figure 3.5 on ajoute un module additionnel « Application » à la fin de la chaîne qui représente une étape de traitement des informations sortantes spécifique à une application commerciale donnée (par exemple, cela pourrait être le comptage de personnes dans les magasins pour établir des statistiques de fréquentation).

Les approches utilisées dans le comptage des personnes peuvent être principalement regroupées selon [TDS10] en modélisation d'arrière-plan [ATJAK12], segmentation d'objets [KAV12] et filtrage par motif [LZL+15, Rau13, YMKC08]. Nous pouvons trouver plusieurs travaux qui utilisent une de ces approches ou une combinaison de celles-ci. La modélisation d'arrière-plan, normalement utilisée dans les solutions avec des caméras RVB, implique des algorithmes complexes et un temps de calcul lourd [CCWC12]. La plupart de ces algorithmes visent les changements de luminosité de l'image et évitent l'absorption des personnes immobiles dans l'arrière-plan. D'autre part, la segmentation est préférée dans les approches qui utilisent des capteurs 3D car cela simplifie l'extraction du premier plan et évite la modélisation d'arrière-plan [GG13, RG12, KAD+14, VZS13]. Dans les travaux de [QLYWj10], un seuil de hauteur permet une segmentation rapide, mais cette méthode ne prend pas en compte les irrégularités du scénario. En revanche, Rauter et al. [Rau13] utilisent des descripteurs de caractéristiques pour éviter la modélisation d'arrière-plan. Cela nécessite cependant un paramétrage manuel de la hauteur des caméras. Finalement, le filtrage par motif, qui utilise des caméras 3D et 2D, est d'une grande complexité car les résultats du filtrage sont réévalués [LZL+15, YMKC08]. On trouve plusieurs descriptions de ces travaux dans les sections suivantes.

Conclusions

Dans cette section, nous avons décrit le concept de caméra intelligente et ses composants, par une chaîne de traitement classique du processus de suivi des personnes pour l'extraction des propriétés spatio-temporelles ainsi que l'importance de définir une conception du matériel informatique et des logiciels. En examinant la littérature abondante sur le suivi de personnes, nous avons présenté les approches principales pour l'extraction des propriétés de chaque bloc de la chaîne des suivis et son application. Dans tous ces travaux, on remarque l'utilisation

systématique de caméras en position zénithale, l'utilisation de caméras de profondeur et de systèmes avec des ressources limitées pour différentes approches. Cependant il n'existe pas, à notre connaissance, d'utilisation de tous ces éléments qui, réunis, permettent la construction d'une caméra intelligente de faible coût.

3.2 Système proposé

Nous présentons dans cette partie l'architecture conceptuelle de l'extraction des propriétés observables et la conception de l'architecture physique du système pour suivre des personnes dans les espaces publics. Nous décrivons les algorithmes développés pour répondre aux problèmes du suivi des personnes et la conception d'un système prototype d'une caméra intelligente industrialisable.

On commencera par définir notre approche basée sur la chaîne de traitement du cadre général (Fig.3.4) en utilisant une approche multi chaîne de traitement pour le suivi des personnes en mouvement. Ensuite, on propose une solution avec des ressources limitées en calcul et en s'assurant de disposer d'un système de caméras intelligentes. Celui-ci nous permet de fournir une solution de faible coût et extensible à de grandes surfaces. Enfin, on présentera l'évaluation des résultats de l'implantation réalisée.

3.2.1 Conception de l'architecture d'extraction des propriétés observables

Comme prémisses de notre approche, on utilise une caméra de profondeur en position zénithale. Dans ces conditions, on obtient un flux de vidéo dans lequel une ou plusieurs personnes peuvent entrer par l'un des bords du champ de vision de la caméra dans n'importe quelle direction. Notre objectif final est d'obtenir les trajectoires des personnes à l'intérieur du FoV en extrayant leurs caractéristiques cinématiques avec une résolution la plus élevée possible. Ces trajectoires et caractérisations nous permettent d'étudier leur comportement et l'utilisation de l'espace.

Dans la première étape du cadre général du suivi des personnes, on commence normalement par la détection de la présence de personnes dans le champ de vision de la caméra. Comme on l'a vu dans la section 3.1.1, il y a des travaux qui utilisent une pyramide d'échelle et une fenêtre glissante pour détecter la présence d'une personne [BOHS14, FGMR10, VJ04, Dal06]. D'autre part, les approches de modélisation d'arrière-plan se servent des changements d'illumination [ATJAK12] ou de mouvements des objets dans la scène [SS14] pour détecter les personnes. Cependant, dans les travaux reliés à l'utilisation d'une caméra 3D, on a identifié certaines étapes spécifiques liées à la compréhension de l'environnement où la caméra est installée [BSA06, GG13, KAD+14, LZL+15]. Compte tenu de ces travaux, la compréhension de l'environnement répond à deux questions :

- Quelle est la position de la caméra par rapport à la scène ?
- Quelles sont les caractéristiques de la scène ?

Cette compréhension de l'environnement est un point clé dans notre processus de conception de la chaîne de traitement et permet également d'extraire des informations pertinentes pour améliorer les résultats [GG13] et diminuer la complexité des étapes suivantes de calcul. Il est évident qu'une telle opération ne peut pas s'effectuer pour chaque nouvelle image car cela serait trop complexe en matière de temps de calcul et incompatible avec la contrainte de ressources limitées imposée à notre système. Dans cette approche, il est nécessaire d'estimer *la position de la caméra* par rapport à son environnement et de représenter l'image capturée par la caméra par un modèle statique de la scène dénommé *modelé de l'arrière-plan.*

Pour nos applications, la caméra est fixée au plafond et immobile par rapport au sol. On peut alors supposer que le modèle de la scène ne change pas pendant les périodes d'analyse, donc

ce modèle peut être extrait (une seule fois) pendant l'étape d'initialisation avant de commencer l'opération de suivi des personnes, sans affecter la qualité des résultats de détection et de suivi.

Ainsi, on diminue la consommation de ressources de calcul dans le processus de suivi de personnes. En conséquence, on introduit le concept « en ligne » qui fait référence à une chaîne de traitement qui s'exécute en temps-réel et le concept « hors ligne » qui fait référence à la chaîne de traitement qui s'exécute une seule fois au moment de l'installation de la caméra. Ainsi, nous proposons une approche double chaîne de traitement où la chaîne de traitement *hors ligne* est en charge de la compréhension de l'environnement de la caméra et la chaîne de traitement *en ligne* est basée sur le cadre général de suivi des personnes décrit dans la section précédente et illustré ci-dessous (Fig. 3.6).

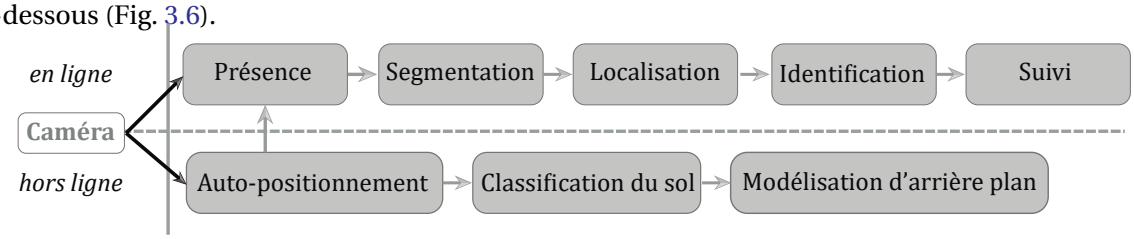
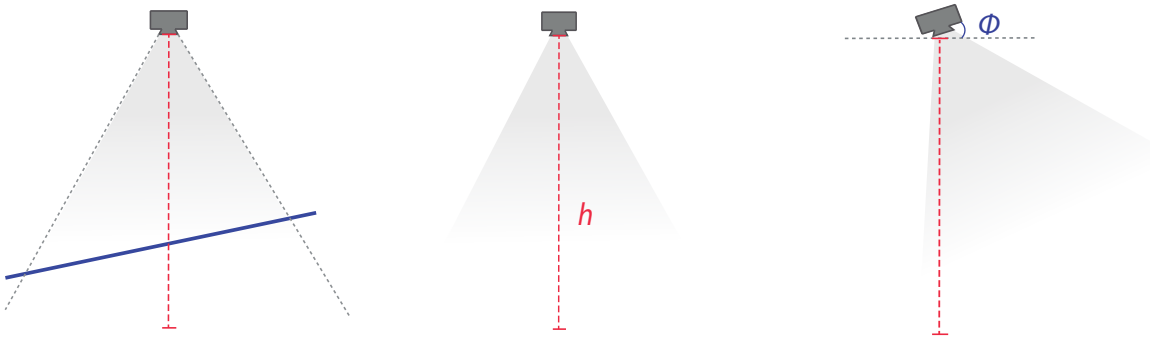


FIGURE 3.6 – Approche double chaîne de traitement (*en ligne* et *hors ligne*) sur la caméra

Chaîne de traitement hors ligne

Cette chaîne de traitement *hors ligne* réalise l'extraction de paramètres statiques de l'environnement où est placée la caméra. Cette chaîne de traitement est composée des blocs d'auto-positionnement, d'estimation des surfaces des objets hors-sol et de la modélisation d'arrière-plan. À la différence de la chaîne de traitement *en ligne*, celle-ci est faite sans la contrainte du *temps-réel* mais en respectant un temps raisonnable pour un opérateur (temps d'installation d'un système). De plus, on peut améliorer notre approche si l'on effectue la chaîne de traitement *hors ligne* sur les moments inoccupés du système, par exemple lorsque le magasin est fermé. Il convient aussi de noter que cette chaîne de traitement contribue à la propriété *d'adaptation et d'autonomie* des caméras intelligentes.



(a) Équation du plan du sol **(b)** Hauteur d'installation **(c)** Orientation de la caméra

FIGURE 3.7 – Paramètres obtenus dans la calibration extrinsèque de la caméra.

Auto-positionnement de la caméra

Dans ce bloc, on estime automatiquement le positionnement de la caméra dans son environnement. Pour bien comprendre ce dernier, il est nécessaire de déterminer où est le sol. L'estimation du sol est un paramètre extrêmement important pour le reste de la chaîne de traitement. Par exemple, Vera et al. calcule le plan du sol pour exprimer la hauteur des objets par rapport à leur distance au sol. On appelle ce processus d'estimation la calibration extrinsèque de la caméra par rapport à son environnement. Cette calibration fournit donc une estimation robuste du plan du sol (Fig. 3.7a) qui permet d'obtenir l'équation du plan du sol, la hauteur d'installation (Fig. 3.7b) et l'orientation de la caméra (Fig. 3.7c).

Nous supposons que le sol de la scène est uniforme et plat, il est donc possible de le modéliser avec un plan [VZS13]. On suppose également que le plan du sol est la région la plus grande de l'image (ceci est vrai uniquement dans le cas où la caméra est en position zénithale) qui nous permet aussi d'éliminer d'autres régions plates (comme les tables ou les murs). Voici l'algorithme que nous proposons :

D'abord, on transforme chaque pixel de profondeur de l'image d'entrée I(x_p, y_p) en un point 3D *pi=(X_i, Y_i, Z_i)* (section 2.2.1). L'ensemble de ces points forme un nuage des points. Ensuite, on génère un histogramme de la profondeur de la scène (Fig. 3.9a et Fig. 3.9c) pour identifier l'intervalle B_{max} où la densité est la plus élevée, car le sol occupe normalement la majorité de l'image. Ceci nous permet de filtrer les points du sol dont la profondeur se situe dans une plage [B_{max} - d_0 , B_{max} + d_0]. Cette plage prend en compte les erreurs introduites par l'acquisition des images de profondeur (déjà présentées dans le chapitre précédent). On peut observer les phénomènes de rapport signal / bruit dans la déformation aux bords des images acquises des figures 3.9c et 3.9d qui font varier d_0. La valeur de d_0 dépend de la hauteur d'installation et du rapport signal / bruit du signal de profondeur, typiquement entre 10 et 15 centimètres, selon nos évaluations en différents scenarios où la hauteur d'installation de la caméra a varié entre 2,50 et 4 mètres (Fig. 3.9c et 3.9d).

Nous extrayons ensuite l'équation du plan du sol en minimisant le carré de la distance entre ce nuage de points filtrés et le plan recherché (figure 3.8). On note qu'à la différence de [VZS13], nous déterminons l'équation du sol dans le but de générer un modèle d'arrière-plan et non pour faire pivoter le nuage de points acquis (en évitant ainsi une étape supplémentaire à exécuter dans notre chaîne de traitement à chaque image).

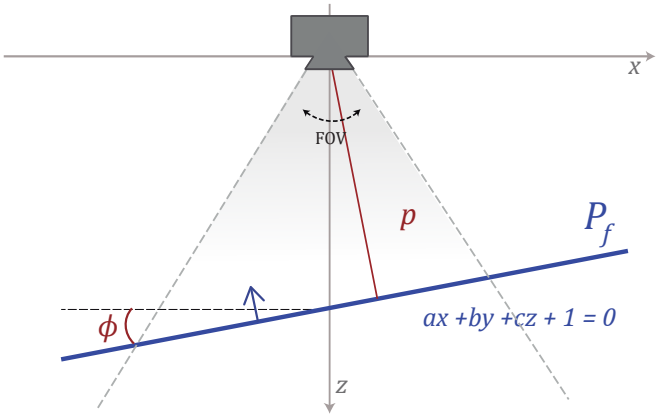


FIGURE 3.8 – Plan du sol à l'aide d'un système de coordonnées dont l'origine est la caméra (vue latérale à partir de l'axe *y*).

Soit $\Psi = \{p_i | \mathrm{B_{max}} - d_0 \leq z_i \leq \mathrm{B_{max}} + d_0\}$ l'ensemble sélectionné de N points dans l'image de profondeur supposée appartenir au plan du sol P_f défini par l'équation $ax + by + cz + 1 = 0$. Si on définit $d = \sqrt{a^2 + \mathrm{b}^2 + \mathrm{c}^2}$ alors la somme des distances carrées à ce plan est :

$$\mathrm{S}(\Psi) = \frac{1}{d^2} \sum_{i=1}^{\mathrm{N}} (ax_i + by_i + cz_i + 1)^2 \tag{3.1}$$

où la meilleure estimation des paramètres du sol est trouvée par :

$$\underset{a,b,c}{argmin} \quad \mathrm{S}(\Psi) = \underset{a,b,c}{argmin} \quad \frac{1}{d^2} \sum_{i=1}^{n} (ax_i + by_i + cz_i + 1)^2 \tag{3.2}$$

Si les coefficients a et b sont trop grands, le plan du sol n'est pas quasi horizontal et nous devons répéter la procédure d'estimation avec un nouvel histogramme de profondeur par rapport à la normale du dernier plan estimé. Enfin, on obtient la distance estimée au sol p (hauteur d'installation de la caméra Fig. 3.7) et l'angle Φ entre l'axe X de la caméra et le plan du sol.

Pour finir, on classifie les pixels de la scène pour déterminer quels sont ceux appartenant au sol sous la forme d'un nuage des points. D'ailleurs, on définit la région des points du sol R_F comme un ensemble connecté de pixels de profondeur plus proche que d_0 du plan du sol P_f. En d'autres termes, nous regroupons tous les points qui sont à une distance maximale de d_0 du plan estimé.

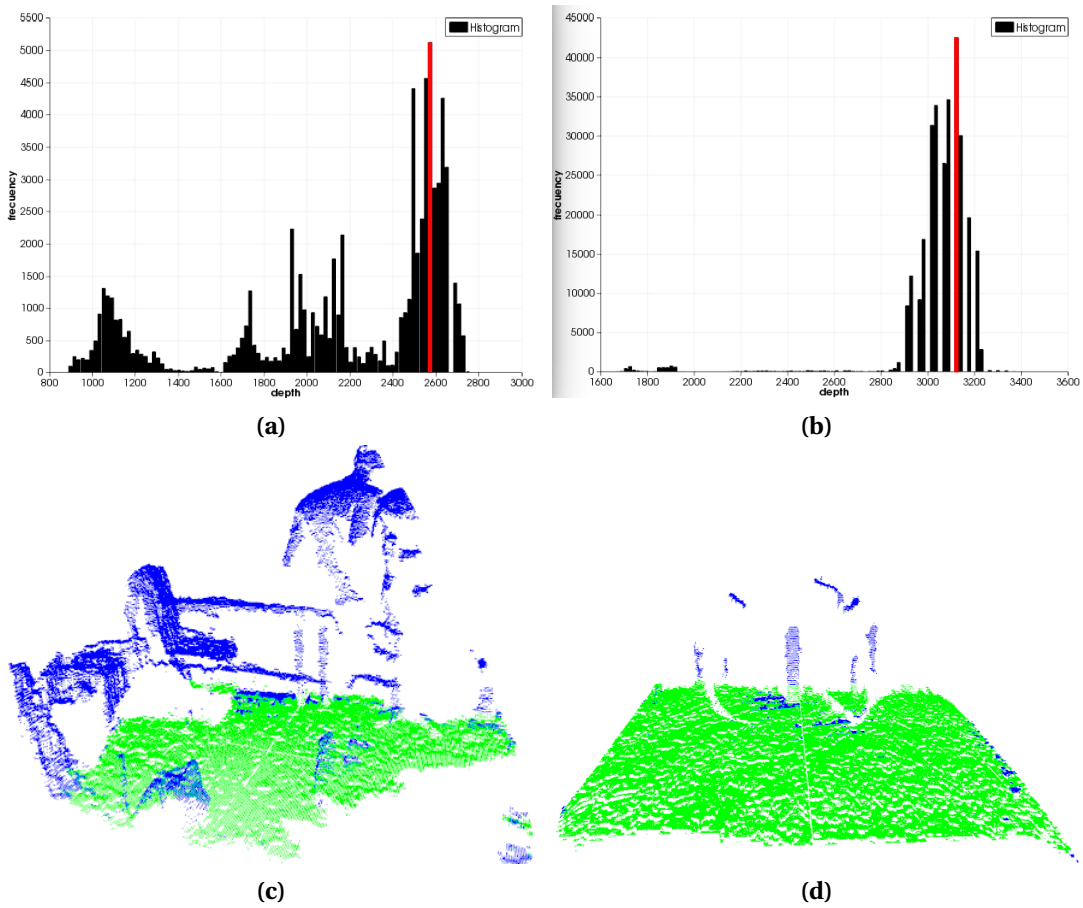


FIGURE 3.9 – Classification du sol en utilisant l'histogramme, dans deux cas opposés : une scène (a) avec fond simple, scène (b) avec un arrière-plan complexe et des objets dynamiques

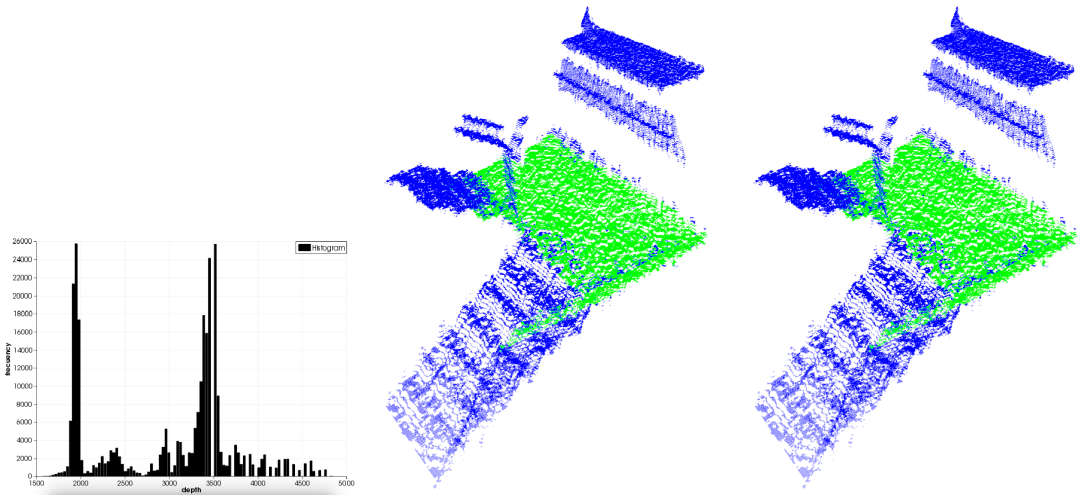
La figure 3.9 montre la classification du sol en utilisant l'histogramme associé, dans deux cas opposés : une scène (3.9c) avec un fond simple, sans la présence d'objets dynamiques et une autre scène (3.9d) avec un arrière-plan complexe et des objets dynamiques. Un fond simple peut être un sol sans la présence d'objets dynamiques et avec très peu d'objets. Un fond complexe contient de nombreux objets, comme des tables, des chaises, des murs, des portes, des escaliers ou des rampes, des objets autres que le sol.

Estimation des surfaces des objets hors-sol

Dans ce bloc, on interprète les pixels restants de la scène. Ces pixels de profondeur sont classés dans une ou plusieurs régions connectées R_i, autour de la région du sol et dans les bords de d'image. En outre, nous supposons que la plupart des objets hors-sol peuvent être modélisés approximativement en tant que surfaces planes. Pour chaque région R_i, nous estimons les paramètres (a_i, b_i, c_i) du plan local $\mathrm{P}r^i$.

$$\mathrm{P}r^i := \{a^i x + b^i x + c^i z + 1 = 0\} \tag{3.3}$$

Si la normale de R_i est quasi-parallèle au plan du sol, cette région est identifiée comme un mur ou une porte. Pour le reste des régions significatives, nous calculons l'angle relatif ϕ_i au plan du sol. Si cet angle est plus grand que l'angle maximal de construction ω [Gib13], nous les classons comme des meubles, sinon il devient une région inclinée R_i où l'on peut faire du suivi de personnes. Les régions quasi-parallèles au plan du sol peuvent être classées comme différents étages à l'intérieur de la scène (Fig. 3.10). Finalement, tous les pixels non classés qui ne forment pas une région connectée ou qui sont des pixels invalides (sans information de profondeur), sont marqués comme indéterminés pour les traiter dans le bloc suivant.



(a) **(b)** **(c)**

FIGURE 3.10 – Scénario de comptage dans le cas d'une situation avec deux étages a) Histogramme avec les multiples pics de densité à la distance de chaque étage. b) nuage de points de l'image et identification du sol (en vert). c) nuage de points de l'image et identification du sol du deuxième étage (en vert).

Modélisation de l'arrière-plan

Dans ce dernier bloc de la chaîne de traitement *hors ligne*, on estime le modèle de l'arrière-plan. Ce modèle est une représentation de la scène statique observée qui permet de séparer, dans la chaîne de traitement *en ligne,* les pixels utiles par rapport à l'ensemble des pixels acquis. L'ensemble de ces pixels utiles va constituer ce que l'on appelle le premier plan.

Les travaux qui se servent d'un capteur 3D ont tendance à utiliser un algorithme non-itératif [QLYWj10, GG13] pour estimer le modèle de l'arrière-plan et pour filtrer rapidement les pixels qui appartiennent aux personnes suivies. Par exemple, [GG13] crée un modèle basé sur les valeurs moyennes du même pixel sur plusieurs trames au moment du déploiement en absence des personnes dans la scène. Dans le travail de [QLYWj10], on utilise une distance t_d (Fig. Fig. 3.11) qui sert à filtrer rapidement tous les pixels éloignés de la caméra pour se focaliser sur les pixels proches. De plus, dans les supermarchés ou les centres commerciaux, il y a un grand nombre d'objets avec lesquels les gens peuvent interagir, tels que des chariots, et ceci est une difficulté au moment de détecter le premier plan. Ces objets indésirables doivent être éliminés pour le comptage en utilisant la modélisation d'arrière-plan. Dans le cas de Shopline, on doit éviter de compter les enfants parce qu'ils ne sont pas des acheteurs potentiels et ne sont pas pertinents pour les études commerciales et marketing des clients potentiels.

Par conséquent, nous avons décidé de nous inspirer de ces derniers travaux cités pour générer une nouvelle approche adaptée à nos besoins. D'un côté, les données intrinsèques de la caméra sont dans un espace 2D (l'image de profondeur). De l'autre, les régions identifiées dans la section précédente sont exprimées dans l'espace 3D (équations de plusieurs plans). Une approche classique devrait enchaîner plusieurs transformations (projection et rétroprojection) à chaque acquisition : ainsi, on transforme tous les pixels de l'image d'entrée en points 3D, puis on teste si ces points dépassent les planes 3D Pr^i, pour finalement les projeter à nouveau dans l'espace 2D. Chaque transformation implique des coûts de calcul assez importants et le travail en 3 dimensions

augmente encore davantage cette charge. On retrouve donc la difficulté d'harmoniser les données dans un seul espace. C'est pour cette raison que l'on propose, comme modèle de l'arrière-plan, un filtre 2D de la taille de l'image d'entrée, nommé *Bg*, pour représenter la scène. Pour estimer ce filtre, on propose une méthode pragmatique qui utilise seulement le système de coordonnées 2D intrinsèque à la caméra pour séparer les pixels du premier plan de l'arrière-plan. Chaque pixel du filtre $\mathrm{B}_g(x,y)$ représente en réalité le seuil de la profondeur à laquelle chaque pixel d'entrée est séparé entre le premier plan et l'arrière-plan. Ce filtre est dérivé d'une mosaïque composée des plusieurs facettes planes dans l'espace 3D, définies de la façon suivante.

Soit L l'ensemble des régions qui appartiennent à la scène mais ne font pas partie de l'ensemble du sol. Soit $\mathrm{L}_t \subseteq \mathrm{L}$ l'ensemble des régions traçables. Soit R_i une région traçable si $\mathrm{R}_i \in \mathrm{L}_t$, R_i est connecté avec R_f et $|\phi_i| \leq \omega$ (sections précédentes). La surface de filtrage est une mosaïque composée de plans de filtrage local P_t^i pour chaque région de traçage R_i. Les plans locaux sont obtenus par une *procédure d'élévation* du plan estimé Pi de la région locale par un seuil de hauteur t_d.

$$\mathrm{P}_t^i := \{a^i x + b^i y + c^i (z + t_d) + 1 = 0\} := \{a_t^i x + b_t^i y + c_t^i z + 1 = 0\} \tag{3.4}$$

Ensuite, le seuil de chaque pixel du filtre est composé par :

$$\mathrm{B}g(x,y) = \begin{cases} \dfrac{1}{a_{t\mathrm{I}r(x,y)}\left(\frac{x-cc_x}{f_x}\right) + b_{t\mathrm{I}r(x,y)}\left(\frac{y-cc_y}{f_y}\right) + c_{t\mathrm{I}r(x,y)}} & \mathrm{G}(x,y) > 0 \quad \wedge \quad \mathrm{I}r(x,y) \neq 0 \\ d - t_d & \text{autrement} \end{cases} \tag{3.5}$$

où $\mathrm{I}_r(x,y)$ est une fonction d'étiquetage qui renvoie pour chaque pixel de profondeur $\mathrm{G}(x,y)$ l'indice i de $\mathrm{R}_i \in \mathrm{L}_t$ afin d'obtenir les valeurs d'estimation planes (a_t^i, b_t^i, c_t^i) pour calculer le $\mathrm{B}_g(x,y)$ ou 0 sinon. Tous les pixels marqués comme indéterminés dans la section précédente rentrent dans le deuxième cas de *Bg* puisqu'ils ne sont pas une étiquette, $\mathrm{I}_r(x,y) = 0$. Comme l'estimation des triplets (a_t^i, b_t^i, c_t^i) a été faite dans le système de coordonnées 3D (section précédente), il faut faire la rétroprojection de ces planes au système de coordonnées 2D de l'image.

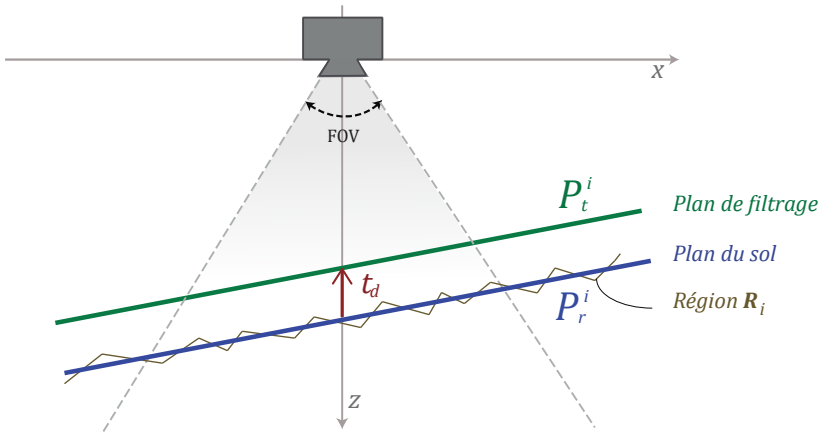
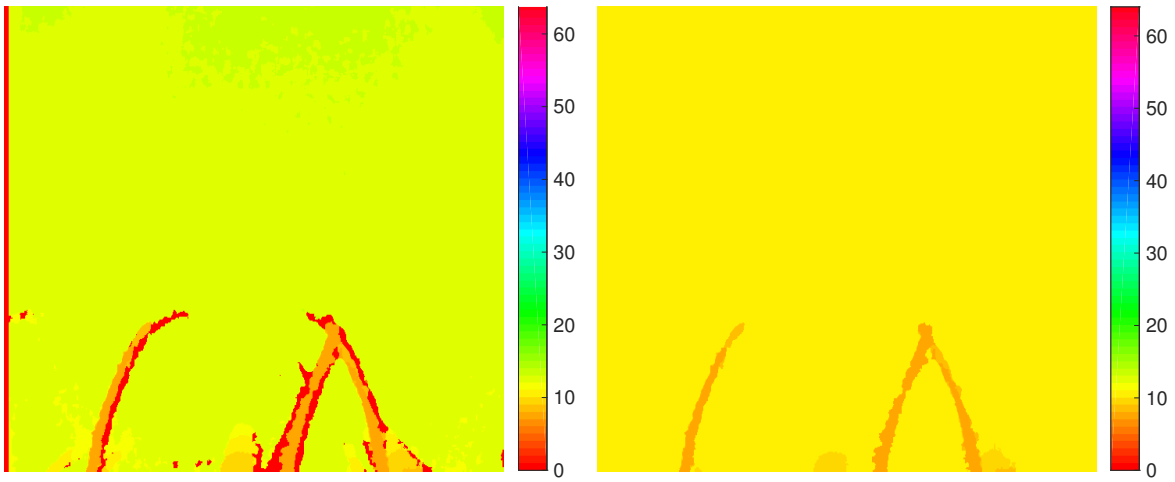


FIGURE 3.11 – Diagrammes de modèle d'arrière-plan et filtrage par hauteur (vue latérale à partir de l'axe y).

La Fig. 3.11 représente une vue projetée sur le plan XZ de cette procédure d'élévation du plan demandée par la construction du modèle d'arrière-plan final où la ligne verte représente P_t^i et la ligne bleue représente P_r^i.

On peut observer la différence entre un modèle classique de l'arrière-plan (Fig. 3.12a) et notre modèle *Bg* (Fig. 3.12b). Les deux images présentent une différence importante : dans la première image, on observe des irrégularités dans la partie supérieure de l'image qui montrent que le sol n'est pas parfaitement horizontal. Dans la deuxième image, on trouve une surface de couleur homogène qui est due à la procédure d'élévation. D'autre part, les pixels indéterminés sont représentés en rouge dans l'image gauche, et ils disparaissent dans l'image de droite en

prenant la valeur de t_d. Par contre, on garde les régions dont la hauteur est supérieure ou égal à t_d. Ce processus induit une érosion d'une partie des objets dont la surface va diminuer (visible dans la partie inférieure de l'image de gauche).



(a) **(b)**

FIGURE 3.12 – (a) Représentation de l'arrière-plan obtenu par moyennage sur une séquence de 15 images et (b) le modèle de l'arrière-plan *Bg*.

Dans la figure 3.12, la première image représente la moyenne de l'arrière-plan sans personnes où les pixels indéterminés sont en couleur rouge. La deuxième image représente le modèle de l'arrière-plan *Bg*.

Exemple d'approche pour la modélisation de l'arrière-plan

Dans le contexte industriel de Shopline, on a besoin de compter des personnes dans les magasins. Pour cette raison, on a installé des caméras intelligentes qui utilisent notre approche pour obtenir des données dans un contexte réel. En analysant ces données, nous avons essayé d'optimiser le processus de filtrage pour deux catégories d'objets en particulier : les objets non désirés et les enfants (Fig. 3.13). Le paramètre *d'élévation* du plan t_d joue un rôle essentiel dans ce processus et nous avons pu établir les meilleures valeurs par rapport aux faux positifs.

Dans le cas des objets non désirés, nous avons identifié des caddies, des poussettes et des animaux de compagnie. Dans ces expériences, on a modifié successivement t_d, trouvant que la valeur permettant de filtrer au mieux ces objets est de 1,10 mètre. Pour le cas du comptage d'adultes, on a installé une caméra intelligente dans une laiterie qui montre le processus de traite aux visiteurs. Le service des ventes avait besoin de compter les possibles acheteurs (qu'ils définissaient comme « adultes ») parmi les visiteurs. A cause de la grande dispersion de tailles des enfants et des adultes, trouver une valeur qui permet de filtrer tous les enfants est assez compliqué. Cependant, par des expériences empiriques, nous avons trouvé une valeur pour le paramètre t_d qui se rapproche le plus du nombre de personnes comptées par rapport au nombre de ventes réalisées pendant l'expérience. Et ceci malgré les éventuelles erreurs de comptage dues aux enfants non filtrés et aux adultes filtrés. La valeur approuvée par le service des ventes, que Shopline recommande actuellement à ses clients pour filtrer les enfantes, est de 1,30 mètre.

La figure 3.13 décrit l'utilisation du paramètre *d'élévation* du plan t_d (déterminé précédemment) dans le processus de comptage d'adultes en éliminant les enfants et les autres objets non désirés. On propose donc deux choix : si l'utilisateur final souhaite filtrer les objets non désirés, il doit utiliser une valeur t_d de 1,10 m ou la valeur de 1,30 m dans le cas des enfants.

En conclusion, cette étape qui modélise l'arrière-plan permet de déterminer un filtre fiable qui sépare les pixels d'arrière-plan des pixels d'avant-plan. Ces derniers seront utilisés pour effectuer l'étape de détection des personnes.

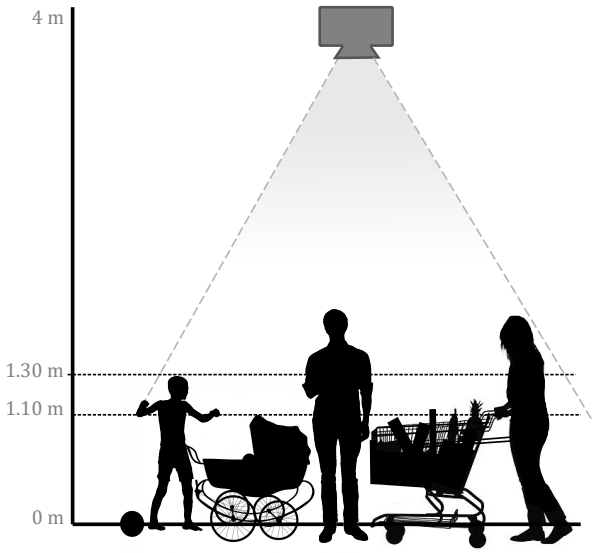


FIGURE 3.13 – Diagramme de modèle d'arrière-plan et filtrage par hauteur à différentes hauteurs de filtrage pour t_d.

Chaîne de traitement *en ligne*

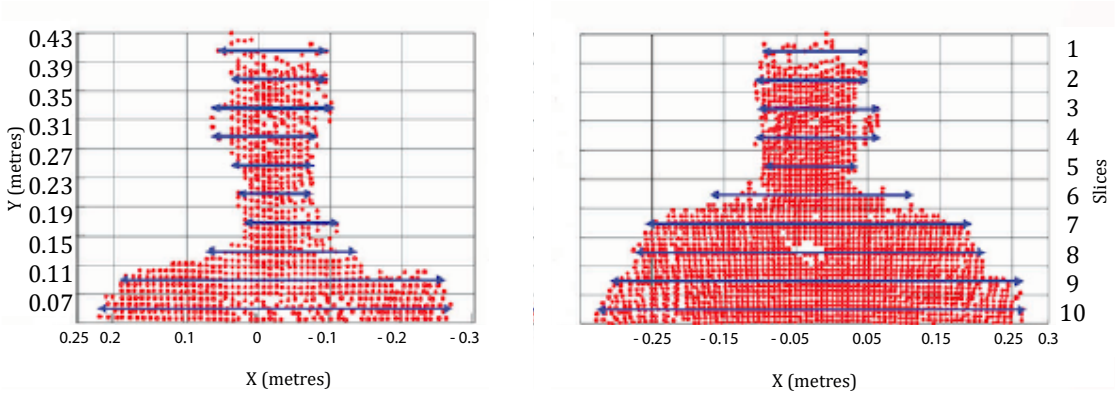
La chaîne de traitement *en ligne* pour suivre des personnes en mouvement est basée sur l'approche décrite dans la section 3.1.2 (Fig. 3.4). Dans cette section, on décrit d'abord la littérature associée au suivi de personnes avec caméras 3D. Ensuite, on propose une procédure de modification de la chaîne de traitement classique pour mieux répondre à nos besoins spécifiques. Dans tous les cas, cette chaîne de traitement devra s'exécuter en temps-réel.

Dans le travail de [VZS13], les auteurs utilisent une caméra 3D zénithale, restreignant leur utilisation à seulement 2 sens contraires de mouvement de personnes (des entrées et des sorties). Pour ce faire, ces auteurs calculent le plan du sol en estimant l'inclinaison de la caméra. A chaque acquisition, ils appliquent une rotation du nuage des points jusqu'à ce que le plan du sol devienne parallèle au plan de l'image de la caméra. De plus, ils expriment la hauteur des objets par rapport à la distance au sol virtuel. Cette démarche classique alterne entre les deux systèmes de coordonnées, contrairement à notre méthode optimisée présentée dans la section « modélisation de l'arrière-plan ». Ensuite, ils utilisent une variante de HOG [DT05] pour détecter des personnes, une méthode lourde pour notre système avec des restrictions de ressources. Finalement, ils réalisent la classification du sens de la trajectoire d'une personne détectée, en utilisant une méthode de SVM.

Dans le travail de [SA11], les auteurs utilisent une caméra en position latérale et une variante de HOG nommée combo-HOD implantée sur un GPU. À la différence de [VZS13], ils utilisent les données de l'image couleur et de profondeur pour localiser les personnes, en augmentant le nombre de détections correctes par rapport à leur ancienne méthode basée sur HOG [STS08]. Ainsi, Spinello et al. ont montré que l'utilisation d'information de profondeur virtuelle réduit largement l'espace de recherche sur l'image, spécialement dans l'approche de recherche pyramidale.

Dans le travail de [KAD+14], les auteurs utilisent une caméra en position latérale. D'abord, les pixels d'intérêt sont filtrés grâce à la soustraction du fond calculé *hors ligne*. Ensuite, les régions résultantes sont évaluées pour extraire la signature appelée «*head-to-shoulders* » (de la tête aux épaules) [KAV12]. Celle-ci détermine un nombre de rayons horizontaux qui traversent la région verticalement depuis les points situés plus haut (cime de la tête), jusqu'à une valeur H=0.4 m (empiriquement déterminée) en évaluant les distances horizontales de la région (Fig. 3.14).

Ensuite, chaque région est classée avec la méthode SVM pour passer à l'étape suivante de la chaîne de traitement. Dans cette méthode, l'espace de recherche des personnes dans l'image est réduit en utilisant la soustraction du fond et le filtrage par la distance maximale. En conséquence, on n'a pas besoin de faire de recherche multi-résolution.



(a) **(b)**

FIGURE 3.14 – Illustration de deux nuages de points de personnes différentes avec 10 rayons horizontaux.

Dans le travail [GG13], les auteurs affirment qu'une acquisition de la scène en *position zénithale* est efficace pour le comptage des personnes : en effet leur détection est relativement facile en cherchant les minimales locales dans une image à trois dimensions. Ce minimum correspond aux têtes des personnes, qui sont les plus proches de la caméra. Cependant, l'utilisation des algorithmes comme « mean shift » [FH75] ou « *ligne de partage des eaux* » [BM92] pour trouver les minima locaux est très lourde pour un calcul rapide sur systèmes embarqués.

En conclusion, on trouve dans la littérature des méthodes différentes pour obtenir les pixels du premier plan dans les images de profondeur (des variantes de HOG et des minima locaux). De plus, on trouve différentes méthodes pour caractériser les personnes (le « head-to-shoulders » et le descripteur de HOG). Cependant, ces algorithmes sont très lourds pour s'exécuter sur des systèmes embarqués en temps-réel.

Suite à l'évaluation de ces travaux, nous proposons une chaîne de traitement de détection des personnes divisée en 4 blocs qui s'exécutent pour chaque nouvelle image de profondeur et que nous décrivons dans les sections suivantes :

- le premier bloc effectue une détection rapide de la présence de personnes ;
- le deuxième bloc effectue une segmentation et une localisation ;
- le troisième bloc procède à l'identification et le suivi de personnes ;
- et finalement, le quatrième bloc est spécifique à chaque application envisagée (il peut par exemple assurer le comptage des personnes).

Détection de la présence de personnes

Le bloc de présence est la première étape du traitement en ligne classique (Fig. 3.4). Son rôle est de répondre à la première question posée dans l'introduction « Y a-t-il une personne dans un lieu donné ? ». Il assure la détection de la présence de personnes dans le champ de vision de la caméra. Ce bloc reçoit les images acquises par la caméra 3D et applique le filtrage *Bg* pour obtenir le premier plan.

Grâce à la construction du *Bg*, on filtre tous les pixels qui sont plus éloignés que le plan virtuel estimé rapidement en utilisant l'équation 3.5 :

$$\mathrm{F}i(x,y) = \begin{cases} \mathrm{G}(x,y) & \mathrm{G(x,y)} \geq \mathrm{Bg(x,y)} \\ 0 & \text{autrement} \end{cases}$$

On obtient une image qui représente les pixels du premier plan comme le montre la figure suivante :

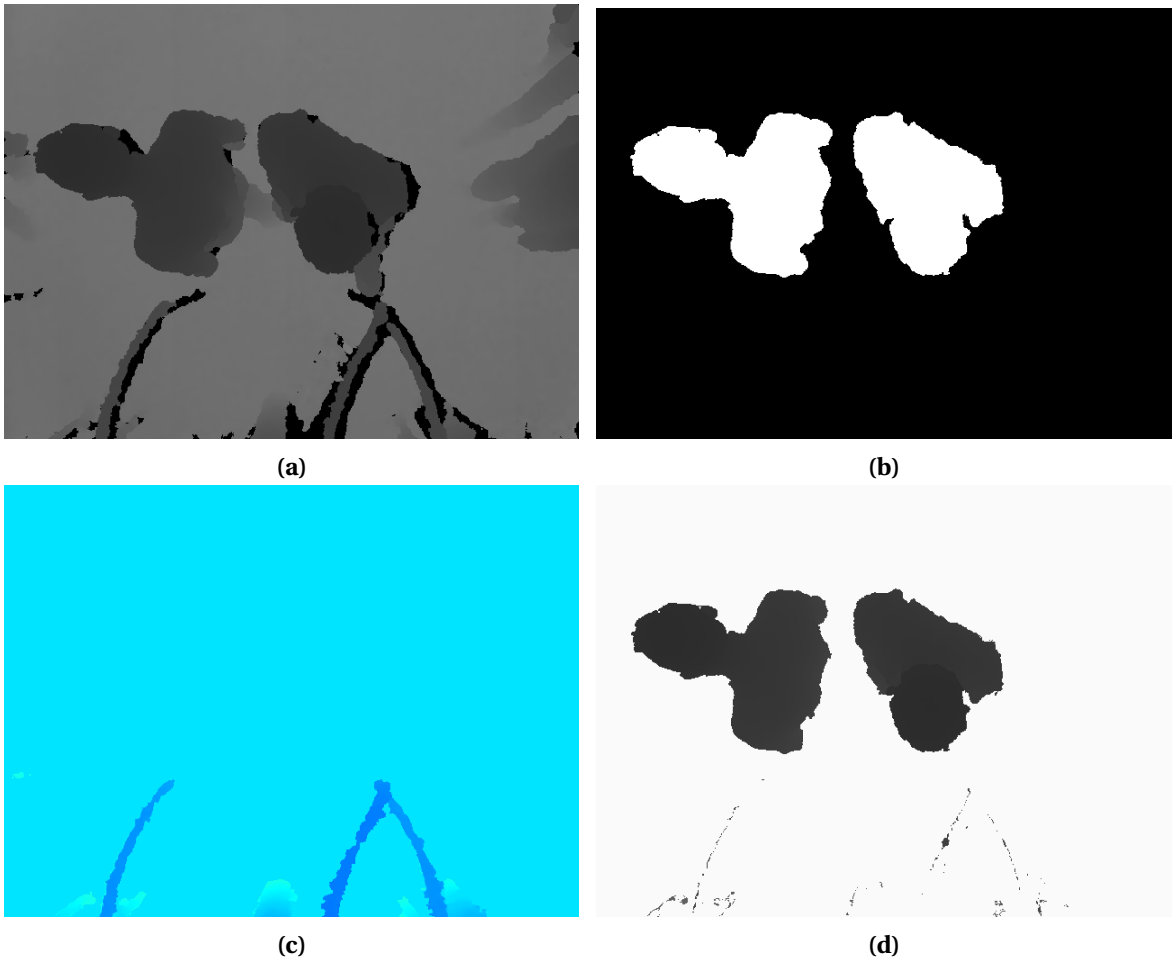


FIGURE 3.15 – a) Image de profondeur d'entrée *G*, b) masque binaire des pixels qui appartient au premier plan, c) fond *Bg* obtenu à partir de la scène et d) image filtrée *Fi* de l'image d'entrée (a).

Segmentation et localisation de personnes

La méthode que nous proposons réalise la segmentation et la localisation des éventuelles personnes dans le même bloc de la chaîne de traitement (Fig. 3.16). A la différence d'autres méthodes, on obtient les pixels d'intérêt rapidement et facilement grâce à l'utilisation du capteur 3D et au positionnement de la caméra. Dans ce cas, il n'est pas nécessaire de faire une recherche multi-résolution. En conséquence, on obtient une seule détection par personne (à la différence d'une recherche pyramidale), donc nous n'avons pas besoin de regrouper des détections multiples, comme dans les méthodes [FGMR10, DT05], pour obtenir une localisation précise. Ce bloc de segmentation / localisation de la chaîne de traitement est composé de plusieurs sous-blocs : un pour la segmentation légère, un autre pour la segmentation par graphes de niveaux et un dernier pour l'extraction du vecteur de caractéristiques de la forme humaine (***H**uman **F**eature **D**escriptor* : HFD) que nous allons décrire ci-dessous.

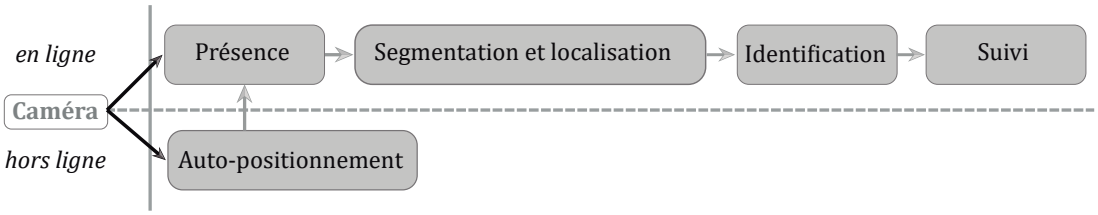


FIGURE 3.16 – Nouvelle configuration de la chaîne de traitement où les blocs de segmentation et de localisation sont fusionnés. De plus, les blocs de la chaîne de traitement hors ligne sont fusionnés pour simplifier la figure.

Segmentation légère

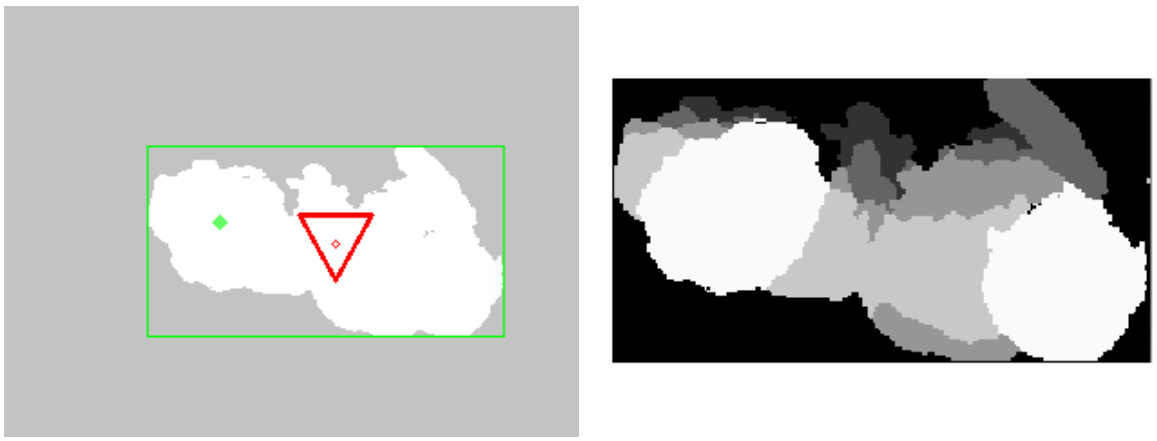
Après avoir trouvé les pixels d'intérêt (dans l'image binaire Fig. 3.15b), l'algorithme continue d'étiqueter les pixels connectés en utilisant la méthode proposée dans [DSB99], pour obtenir des régions d'intérêt Θ_n initiales suite à une première segmentation dite légère. On le désigne par segmentation légère car elle basée sur un seuillage bi-niveau ce qui la rendre très rapide mais aussi assez imprécise : il y a une incertitude sur l'existence d'une ou plusieurs personnes dans la même région. De plus, notre méthode proposée par [ROB+10] sauvegarde pour chaque région Θ_n des informations importantes comme le barycentre, le nombre de pixels, la hauteur maximale (MaxVal) et minimale (MinVal), la hauteur moyenne et la boite de délimitation (BB) initiale dans ce que l'on appelle une table auxiliaire de composants. L'extraction de toutes ces caractéristiques nécessite un seul passage sur l'image, ce qui permet d'économiser le temps de calcul à la différence des autres algorithmes de segmentation comme par exemple « *la ligne de partage des eaux* ». Ensuite, on analyse les régions Θ_n résultantes. D'abord, toutes celles dont sa surface $\#\Theta_n < {a_0}^{min}$, sont immédiatement éliminées où $\#\Theta_n$ est la surface de la région Θ_n définie comme le cardinal des pixels qui la compose et ${a_0}^{min}$ est le seuil inférieur de la surface pour lequel on peut prendre en compte une région Θ_n. D'un autre côté, toutes les régions où de surface supérieure $\#\Theta_n > {a_0}^{max}$ sont renvoyées à la segmentation par graphe de niveaux, où ${a_0}^{max}$ est un seuil supérieur à définir suite aux expériences. Et finalement, toutes les autres régions entre $a_0^{min} < \#\Theta_n < a_0^{max}$, sont renvoyées au sous-processus d'extraction de HFD. Les valeurs vraisemblables de ${a_0}^{\min}$ et ${a_0}^{\max}$ ont été évaluées au sein de l'entreprise Shopline par rapport à la taille moyenne typique d'une région représentant une personne selon la hauteur d'installation. Nous présentons plus de détails sur l'évaluation de l'influence de ces paramètres dans notre section de résultats.

Segmentation par graphe de niveaux

La segmentation légère fournit comme résultat une liste des régions étiquetées Θ de taille N. Dans ce sous-bloc, on analyse chaque Θ_n en détail dans les cas de présence de plusieurs personnes. Ensuite, on extrait une structure de graphe en préparant l'information pour le sous-bloc suivant où l'on réalise la création du descripteur de la forme humaine pour la caractérisation des personnes.

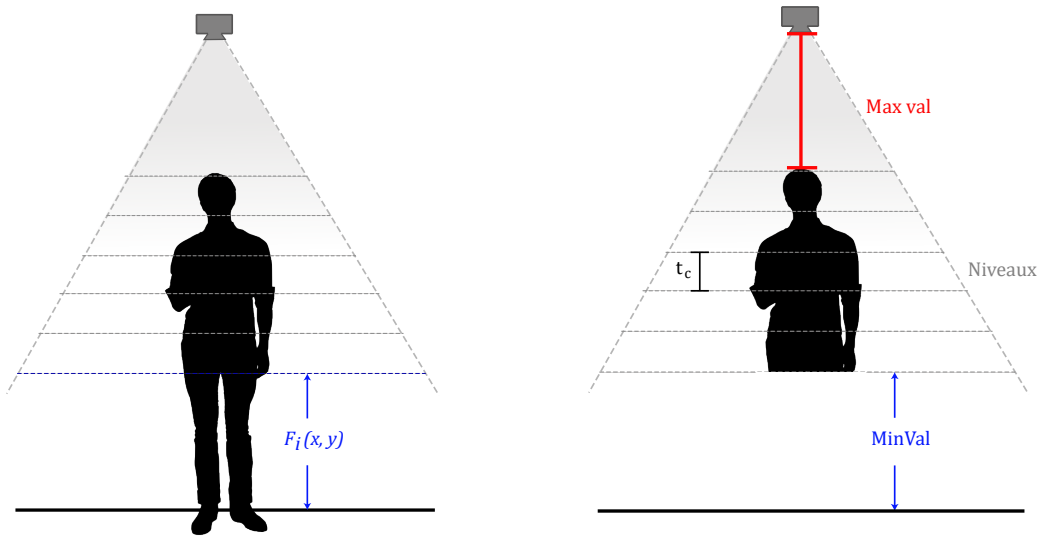
Le processus de segmentation par graphes de niveaux se décompose en deux étapes. La première est appelée *segmentation par niveaux :* chaque région est divisée en tranches horizontales avec une épaisseur constante t_c (Fig. 3.18) (de manière similaire à [KAV12] mais dans une position zénithale), regroupant tous les pixels à l'intérieur de la tranche horizontale en ordre descendant par rapport à sa profondeur. Par exemple, dans la figure 3.18, les pixels qui appartiennent à la première tranche (pixels de la tête) sont regroupés dans le premier niveau, les pixels qui sont à l'intérieur de la deuxième tranche (du cou et des épaules) appartiennent au deuxième niveau et ainsi de suite. Cette segmentation commence à la valeur maximale de profondeur *MaxVal* de la région Θ, jusqu'à la valeur de profondeur minimale *MinVal* (Fig. 3.18) déterminée par la valeur du seuil de filtrage $Fi\ (x,\ y)$. En conséquence, les différents pixels sont regroupés en niveaux (tranches), ce qui nous permet de différencier la tête des épaules. Maintenant que les pixels sont segmentés par hauteur, la deuxième étape utilise la même méthode d'étiquetage que [DSB99] pour obtenir la table d'étiquetage de sous-régions. La figure 3.17 présente un exemple d'image à l'entrée du sous-bloc de segmentation (Fig.3.17a) et le résultat de

la segmentation par graphes de niveaux (Fig. 3.17b). On remarque que cette procédure s'effectue seulement dans la région détectée et non sur toute l'image. Sur l'illustration de la figure 3.17, les niveaux commencent à *MaxVal*, représentée par le point rouge à l'intérieur du triangle dans l'image 3.17a, et se terminent par une valeur minimum *MinVal*.



(a) **(b)**

FIGURE 3.17 – Cas de deux personnes en contact où $\#\Theta_n > 1,5 * a_0$ a) Image de profondeur filtrée et étiquetée. b) Sous-régions segmentées par niveaux.



(a) **(b)**

FIGURE 3.18 – Représentation des niveaux virtuels par les tranches horizontales d'épaisseur t_c utilisées dans la construction du graphe.

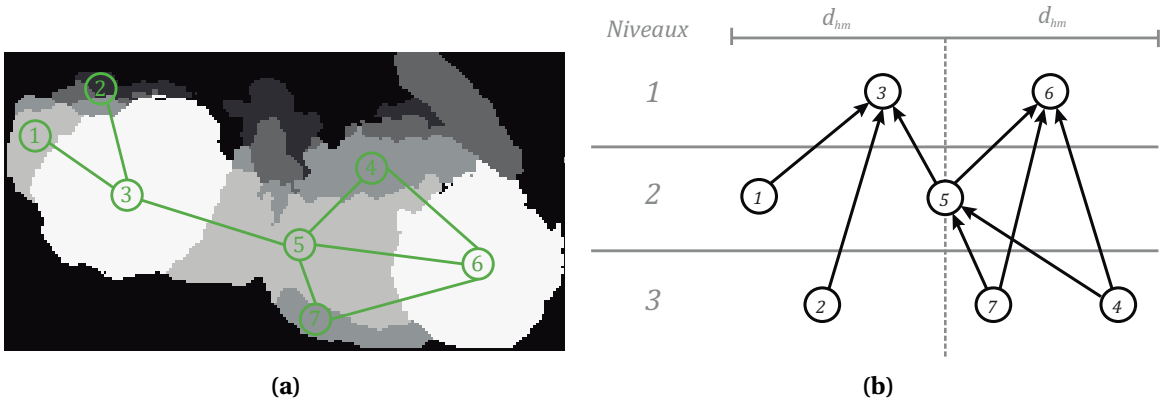
Sur la figure 3.18, la distance *MaxVal* est représentée par le segment rouge et la valeur *MinVal* est représentée pour le segment bleu.

Création de la structure du graphe

La structure utilisée pour caractériser une région détectée est un graphe orienté appelé Graphe de Recherche (en anglais Search Graph : SG) où chaque nœud représente une sous-région et chaque arc représente une connexion topologique entre deux sous-régions voisines, orientée de la sous-région la plus basse vers la plus haute (en profondeur) et qui possède un poids égal à la différence des niveaux des deux sous-régions (voir Fig. 3.19). L'algorithme réutilise la table générée

pendant l'étiquetage des sous-régions pour déterminer les arcs du graphe et leur orientation. Pour chaque pixel de la région, on évalue la relation de voisinage en 4-connectivité topologique avec les autres pixels. Si l'un des pixels voisins n'appartient pas à la même sous-région, nous créons (si nécessaire) un nouvel arc entre les nœuds dans le graphe. Ensuite, on évalue la différence de hauteur entre les deux sous-régions concernées et on assigne cette différence comme poids de l'arc. Par exemple, si un nœud A est connecté à un nœud B et que le poids de l'arc est 2, le nœud A est à 2 niveaux plus haut que le nœud B. Indépendamment de la construction du graphe, on identifie chaque pixel situé à la frontière d'une sous-région comme un pixel de contour de celle-ci (cette information sera utilisée dans l'étape suivante).

L'avantage de représenter une cible humaine en tant que graphe est qu'elle nous permet d'utiliser une représentation efficace des données et l'utilisation des méthodes de recherche rapide sur les graphes. Pour rendre encore plus efficace notre méthode, on ne prend en compte que les trois premiers niveaux de hiérarchie (du haut vers le bas) pour construire les graphes de segmentation. Le *SG* est représenté en mémoire par une matrice antisymétrique de taille égale au nombre de sous-régions segmentées à l'intérieur de la région Θ. Chaque case accueille le poids de l'arc. Une case nulle indique qu'il n'y a pas de connexion entre les nœuds en question. Pour trouver les têtes, nous recherchons les nœuds avec seulement des valeurs de connexion positives (nœuds racines) avec une surface minimale H_s^{min} (par exemple, les nœuds 3 et 6 de la figure 3.19b). Les nœuds racines avec une surface inférieure à H_s^{min} sont éliminés de notre hiérarchie. En peut envisager de ne pas considérer comme nœuds racines ceux avec une surface supérieure à H_s^{max} car ces nœuds correspondent parfois à des sacs à dos et on ne souhaite pas les compter comme une deuxième tête. Ces seuils sont exprimés en nombre de pixels. Le tableau 3.1 montre les valeurs de la matrice de connexion résultante du *SG* (Fig. 3.19b).



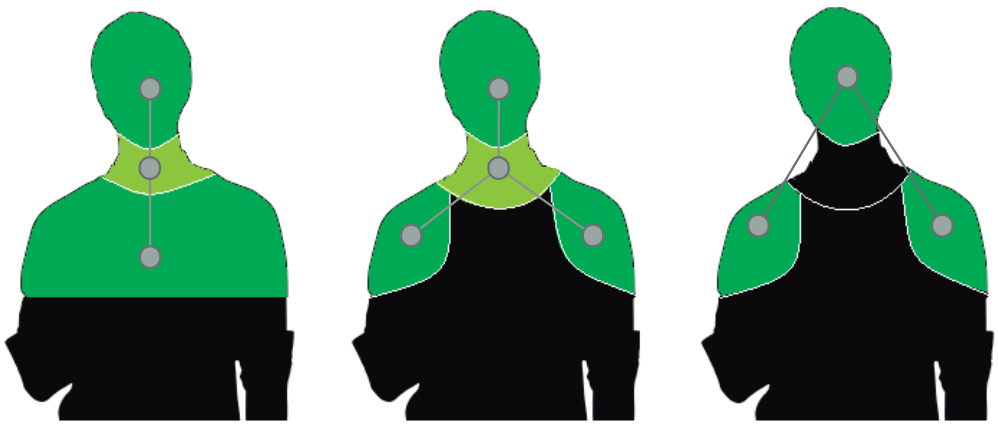
(a) **(b)**

FIGURE 3.19 – Présentation des niveaux virtuels créés pour générer les régions pour construire la structure de graphe. a) Graphe extrait à partir de la segmentation par niveaux. b) Représentation verticale du graphe.

TABLEAU 3.1 – Représentation matricielle du graphe extrait

Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	0	0	-1	0	0	0	0
2	0	0	-2	0	0	0	0
3	***1***	***2***	***0***	***0***	***1***	***0***	***0***
4	0	0	0	0	-1	-2	0
5	0	0	-1	1	0	-1	1
6	***0***	***0***	***0***	***2***	***1***	***0***	***2***
7	0	0	0	0	-1	-2	0

Dans le tableau 3.1, les lignes 3 et 6 représentent les connexions des nœuds racine extraits de la figure 3.19, où toutes ses valeurs sont positives. Par contre, les nœuds enfants ont au moins une valeur négative. Cette structure permet d'identifier rapidement les différentes parties du corps à l'intérieur de l'image pour finalement extraire les caractéristiques de la forme humaine.



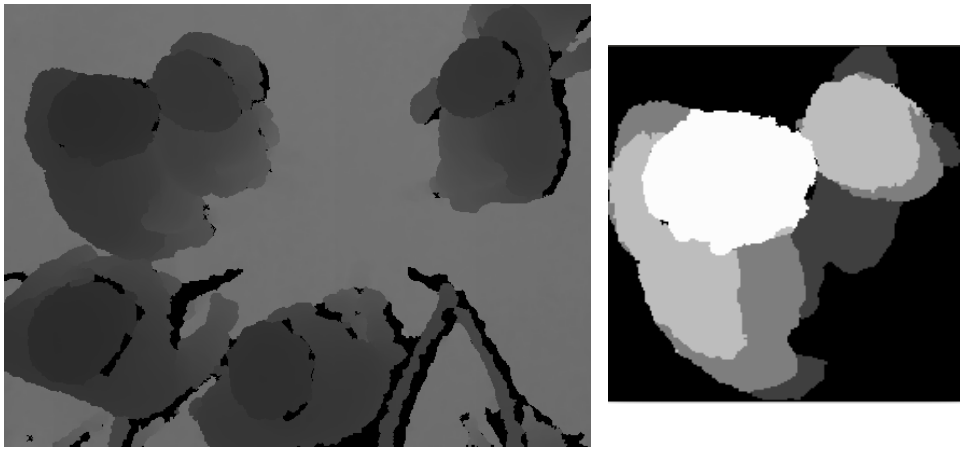
(a) **(b)** **(c)**

FIGURE 3.20 – Graphes représentant les trois versions possibles de segmentation d'une personne.

A partir d'une région Θ, on peut extraire différents types de graphe, comme cela est illustré dans la figure 3.20. Chaque type de graphe est relié à une position et à une orientation dans le champ de vision de la caméra (section 2.4.2).

Il y a des cas où il n'est pas possible des séparer les têtes de plusieurs personnes à cause de leurs postures particulières (par exemple leur façon de marcher, leur position par rapport aux autres ou leur position trop penchée) : dans ce cas l'algorithme détecte moins de nœuds racines que de personnes en réalité. Nous pouvons quand même détecter cette situation en divisant la surface totale de la région par la surface moyenne attendue pour une personne. Dans ce cas, nous mémorisons au niveau de la région la différence entre le nombre des personnes potentiellement détectables et réellement détectées. Cette différence sera appelée ambiguïté de détection.

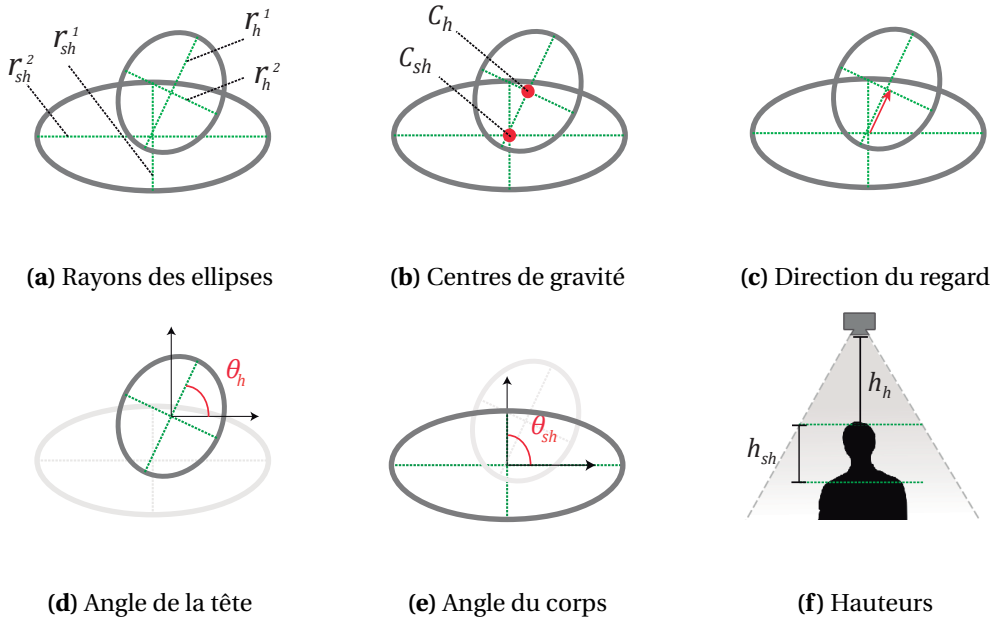
Par exemple dans la figure 3.21, on observe une région suffisamment grande pour supposer qu'il y a deux personnes. Comme on peut le voir, il y a bien deux personnes, cependant la tête de la première (celle de droite dans l'image 3.21b) est à la même hauteur que les épaules de la deuxième. Comme l'algorithme ne connaît pas la localisation ou la quantité des personnes dans la région évaluée, la posture des personnes rend très difficile la détection des deux têtes. Ainsi, cette région aura une ambiguïté de détection de 1 (2-1, cette-à-dire, une tête trouvée sur 2 possibles).



(a) **(b)**

FIGURE 3.21 – Exemple de génération d'occultations d'une tête.

Dans la figure 3.21, on observe un cas avec beaucoup des conditions particulières comme la différence de taille entre les deux personnes, la posture de la personne la plus grande (à gauche de la figure 3.21b) et lorsque les personnes sont très proches. On peut observer que la hauteur des épaules de la personne moins haute tombe dans le $4^{\text{ème}}$ niveau et que sa tête qui est située dans le $2^{\text{ème}}$ niveau (et partiellement dans le $3^{\text{ème}}$) peut facilement être interprété comme l'épaule de la personne la plus haute.



(a) Rayons des ellipses **(b)** Centres de gravité **(c)** Direction du regard

(d) Angle de la tête **(e)** Angle du corps **(f)** Hauteurs

FIGURE 3.22 – Propriétés du descripteur des caractéristiques humaines.

Le descripteur des caractéristiques de la forme humaine

On utilise les descripteurs de la forme humaine (HFD) Ω de manière à caractériser les personnes détectées et suivies par nos algorithmes. Cette caractérisation sert à identifier les personnes de manière unique à travers l'intégralité de la chaîne de traitement du système (qui peut être distribuée) et à diminuer l'ambiguïté dans le processus de suivi. Ce vecteur de caractéristiques est composé de propriétés spatio-temporelles. D'une part, on décrit sa morphologie à partir des mesures du corps extraites dans chaque trame. D'autre part, on extrait des propriétés temporelles comme la vitesse et la direction de la personne à partir de deux trames consécutives. L'ensemble de ces caractéristiques définit un *modèle de sujet* et son actualisation présente des difficultés dues aux possibles déformations du corps humain, aux différents points de vue de la caméra, à la variation d'échelle, au mouvement rapide du sujet, et à la similitude des gens [WLY15].

Dans les travaux précédents [GG13, KAV12], on trouve que la tête et les épaules sont fortement utilisés pour caractériser des personnes dans des systèmes basées sur des caméras de profondeur. De la même manière que [KAV12], nous nous sommes basés sur l'étude anthropométrique de 9000 soldats américains [GWT+89] pour faire la preuve d'un concept empirique, à savoir que les mesures de la tête et des épaules sont suffisamment différentes selon les sujets pour caractériser ces personnes. On se focalise donc sur la séparation entre les pixels appartenant à la tête et les autres pixels appartenant aux épaules, à la poitrine et au dos. Le positionnement zénithal de la caméra facilite cette tâche de segmentation parce que la tête et les épaules génèrent des occultations sur les autres parties du corps [BKIM13].

Dans le vecteur de caractéristiques de la forme humaine, on mémorise la tête et le corps de chaque personne par une représentation approximative : des ellipses de largeurs différentes. Donc, les vecteurs des caractéristiques Ω sont composés des deux rayons de deux ellipses, de leurs angles d'orientation et de leurs centres de gravité, des hauteurs de la tête et des épaules de la personne, de la différence des hauteurs entre les épaules et la tête, du vecteur de direction formé à partir de la position du centre de gravité de la tête à la position du centre de gravité du corps (en supposant qu'il représente la direction du regard, voir Fig. 3.29b), et du *vecteur de mouvement* entre deux position successives (Fig. 3.29).

Pour estimer les caractéristiques de la personne, l'algorithme se focalise sur la recherche des ellipses à partir des contours de la tête et du corps extraites dans l'étape précédente. Dû à nos contraintes de ressources, on diminue le nombre de pixels qui composent le contour en utilisant

la méthode d'échantillonnage [TC89]. Pour estimer les ellipses, on utilise la méthode de [FPF99] qui est robuste en utilisant un faible nombre de points. On obtient comme résultat les rayons de la tête $r_h{}^1$ et $r_h{}^2$, du corps $r_{sh}{}^1$ et $r_{sh}{}^2$ (Fig. 3.22a), le centre de gravité de la tête c_h, et celui du corps c_{sh} (Fig. 3.22b), et l'angle d'orientation de chaque ellipse Θ_h et $\Theta_s h$ par rapport à l'axe horizontal (Fig. 3.22d et 3.22e). Ces valeurs sont exprimées respectivement en pixels et dégrées. Les distances exprimées en pixels subissent les effets de la variation d'échelle dus à la hauteur d'installation de la caméra. On préfère donc les transformer en distances réelles exprimées en millimètres. Ensuite, on calcule le vecteur de la direction du regard à partir des positions des centres de gravité (Fig. 3.22c), puis on évalue finalement la distance h_h entre la tête et la caméra, et la hauteur relative h_{sh} de la tête par rapport aux épaules (Fig. 3.22f).

A chaque nouvelle image, on obtient une liste de détections et leurs HFDs associés. Donc à chaque image, les modèles du sujet doivent être mis à jour. Une des difficultés de cette mise à jour est de la protéger contre tout changement brusque de ses caractéristiques extraites. Ces changements brusques peuvent provenir des occultations, des effets de bords (de l'image), de la vue partielle de la personne ou d'autres phénomènes. Pour éviter ces changements brusques, la valeur que l'on met à jour est égale à la nouvelle valeur multipliée par δ à laquelle on ajoute la valeur ancienne multipliée par $(1-\delta)$. De cette manière, on lisse les variations des caractéristiques pendant d'éventuels « erreurs » de détection, en évitant ainsi la perte brutale de la cible.

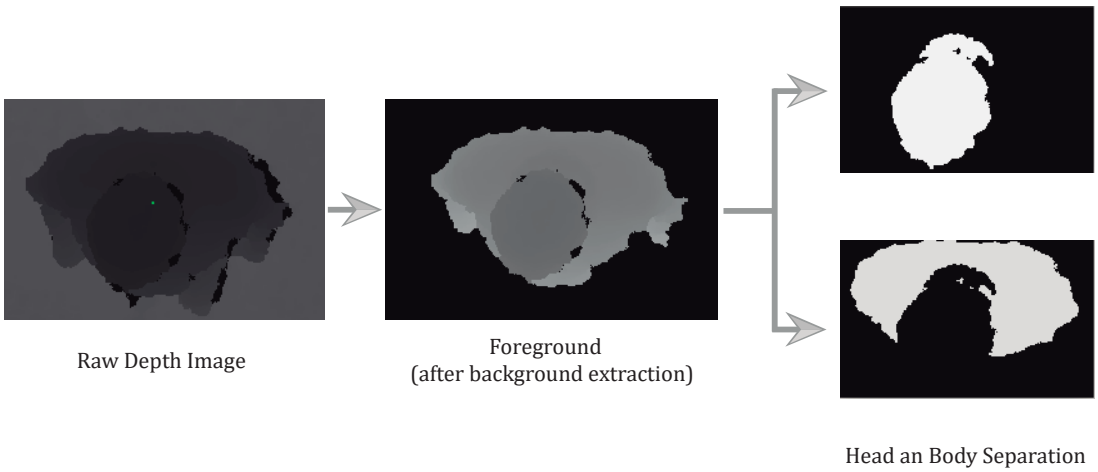


FIGURE 3.23 – Processus d'exaction de HFD.

La figure 3.23 montre le processus d'extraction du vecteur des caractéristiques humaines individuelles Ω à partir du graphe ou de régions simples Θ_n.

Identification et suivi de personnes

Ce bloc utilise les propriétés spatiales extraites Ω (HFD) pour identifier et suivre une personne tout le long d'une séquence vidéo. À la différence de la chaîne de traitement classique, on met ensemble les blocs d'identification et de suivi, comme le montre la figure 3.24. Ce processus n'est pas linéaire et pose des difficultés à cause de plusieurs facteurs perturbateurs :

1. L'occultation entre personnes et l'occultation de personnes par l'infrastructure (même si pour une caméra en position zénithale, elle est minimale).
2. Les changements fréquents et abrupts de direction de déplacement des personnes.

Le principe du suivi consiste à réduire l'espace de recherche pour associer une personne dans l'image actuelle avec les trajectoires extraites des images précédentes. On distingue trois

méthodes différentes pour relier les personnes sans utiliser les autres propriétés spatiales. Dans les travaux de [Rau13], on utilise la distance minimale entre les positions des personnes détectées et les trajectoires possibles (sans prédiction). Dans les travaux de [BSA06], on utilise un filtre de Kalman discret pour prédire les positions possibles (normalement limitées à des mouvements linéaires) des personnes dans l'image actuelle par rapport aux trajectoires. Finalement, [KAD+14] utilise la méthode du filtre à particules pour diminuer l'espace de recherche qui est plus adapté aux mouvements erratiques (deuxième difficulté). De plus, sa méthode de suivi utilise un système de fusion et de séparation des trajectoires au cas où les personnes soient très proches les unes des autres. Les désavantages de cette dernière méthode sont sa nature itérative et sa complexité de calcul liée aux nombre de particules.

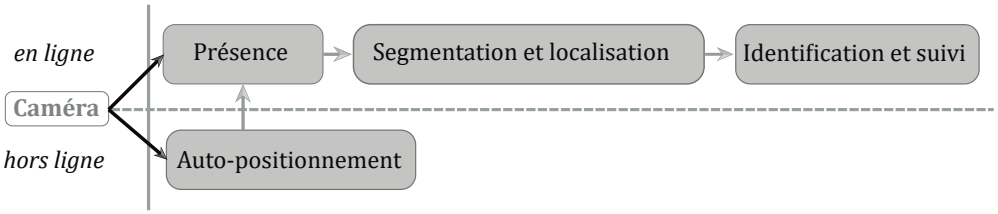


FIGURE 3.24 – Nouvelle configuration de la chaîne de traitement où le bloc d'identification et de suivi sont fusionnés.

Nous proposons un algorithme basé sur deux opérations essentielles : la *correspondance* et la *liaison*. D'abord, on trouve la *correspondance* entre les détections de personnes dans l'image actuelle et les trajectoires des personnes détectées dans la succession des dernières images reçues. Ensuite, on *relie* les détections aux trajectoires correspondantes ou on génère de nouvelles trajectoires, si besoin. Ceci nous permet de mettre à jour les trajectoires précédentes, d'en créer d'autres ou d'arrêter leur suivi, selon le cas. Dans le cas où l'on trouve une *correspondance*, la *liaison* entre une détection et une trajectoire est faite à travers la mise à jour du descripteur HFD, associé à la trajectoire avec les valeurs du descripteur de la détection.

D'abord, on définit les *a priori* suivants pendant le processus de suivi :

- Il n'y pas de décalage d'offset systématique entre la position détectée et la position réelle, même si elle peut être affectée par un bruit de moyenne nulle.
- On définit une région de recherche avec une distance maximale d_{mc} (à partir de la position de la dernière détection) qu'une cible peut parcourir dans l'espace de l'image. On estime la valeur de d_{mc}, en prenant en compte la hauteur d'installation, la vitesse d'acquisition des images et la vitesse moyenne des personnes en marchant qui est de 1.33 m/s [BBHK06].
- La caméra est en position zénithale (le plan focal quasi parallèle au sol), donc :
  1. On peut affirmer que pour une personne donnée, il y a un rapport quasiment fixe entre son mouvement horizontal exprimé en pixels et celui exprimé en mètres.
  2. La distance d'une personne à la caméra est quasi constante, quelle que soit sa position dans l'image (section 2.3.1), en évitant des problèmes d'échelle. On fait l'hypothèse que cette distance ne varie pas plus de 10 % (calculée à partir de l'angle entre l'axe Z de la caméra et la position de la personne).

Ensuite, on définit le processus de mise en correspondance comme une minimisation de la distance entre la position des cibles suivies et les nouvelles détections dans un voisinage limité par d_{mc}. Dans les conditions décrites, ce critère est suffisant pour avoir une correspondance précise entre les détections actuelles et les trajectoires acquises.

Plus formellement, la trajectoire d'une personne est composée de l'historique des positions détectées dans des images consécutives et de son descripteur HFD. Nous pouvons formaliser ceci de la manière suivante :

- Soit P^j l'historique des positions de détections d'une personne j dans des images successives, défini comme l'ensemble de points de détection $P^j := \{\rho_1^j, \ldots, \rho_N^j\}$ ordonnés chronologiquement, où ρ_i^j représente la i^{ème} position (x_i^j, y_i^j) de la personne j suivie.

- Soit Ω^j le vecteur de caractéristiques (descripteur) HFD de taille K, qui représente une personne *j* détectée :

  – Alors on note Ω_k^j une propriété k parmi les K propriétés possibles de Ω^j. En particulier Ω_{pos}^j est la dernière position détectée de la personne *j* en question.

- Soit τ^j la trajectoire de la personne j composée par le couple (Ω^j, P^j) et $T := \{\tau^j\}$ l'ensemble de ces trajectoires détectées depuis le début de l'application de l'algorithme. On remarque que pour un couple correctement associé (Ω^j, P^j), ${\Omega^j}_{pos}$ est égal au dernier point ρ_N^j.

- Soit $T_a \subset T$ l'ensemble des trajectoires dont on dispose à l'instant actuel (donc actualisées à la dernière image acquise par la caméra).

- Soit ω^j le descripteur HFD associé à la détection d'une personne *j* dans l'image actuelle et $D := \{\omega^j\}$ l'ensemble des détections de personnes dans l'image actuelle.

Notre algorithme est sensé, pour chaque nouvelle image, trouver la meilleure correspondance entre chaque détection ω^i et l'ensemble de trajectoires actuelles T_a. Cette tâche revient à un problème d'**identification** d'une détection ω^i avec correspondance possible Ω^j présente dans l'une de trajectoires de T_a. Dans notre cas concret, on cherchera la meilleure correspondance entre le descripteur ω^i et les descripteurs Ω^j qui sont à une distance euclidienne $d^e(\omega_{pos}^i, \Omega_{pos}^j)$ inférieure à la distance maximale d_{mc}.

- Soit $T_p := \{\tau^j \in T : d^e(\omega_{pos}^i, \Omega_{pos}^j) < d_{mc}\}$ et $card(T^p) = J' <= J$.

- Soit d^e la fonction qui calcul la distance euclidienne entre deux positions des HFDs différentes.

Après avoir identifié les trajectoires possibles, on trouve la meilleure correspondance. Pour ce faire, on redéfinit le coefficient de similarité introduit en [VSC+08] comme la distance totale d^{HFD} entre deux HFDs, donné par la somme des distances au carré d^{sq} entre toutes les propriétés de deux descripteurs :

$$d^{\mathrm{HFD}}(\omega, \Omega) = \sum_{k=1}^{K} d^{sq}(\omega_k, \Omega_k) \tag{3.6}$$

L'utilisation de la distance au carré d^{sq} donne plus de poids à la différence entre propriétés des descripteurs et, en même temps, évite des opérations mathématiques complexes dans les systèmes embarqués comme la racine carrée. Finalement, on trouve la meilleure correspondance Ω_c^i entre une détection actuelle Ω^i et un des descripteurs Ω^j de T_p, en minimisant le coefficient de similarité :

$$argmin \sum_{j=1}^{J'} d^{\mathrm{HFD}}(\omega^i, \Omega^j) \tag{3.7}$$

On définit la ***liaison*** comme l'association de ω_i à la trajectoire avec la meilleure correspondance $\tau_c^i(\Omega_c^i, P_c^i)$ et la mise à jour du modèle du sujet (c'est-à-dire le descripteur). Chaque fois qu'une personne est reliée à une trajectoire, le HFD est mis à jour avec les informations de la détection actuelle. Cette mise à jour du modèle sert à améliorer la relation spatio-temporelle entre les objets suivis à travers la séquence d'images. Cette mise à jour correspond à l'actualisation des valeurs des K propriétés de ω^i sur Ω_c^i et l'addition de la nouvelle position détectée ω_{pos}^i.

- Nous définissons une fonction d'actualisation qui calcule un nouveau descripteur Ω^{i+1} à partir de deux HFDs, Ω_c^i et Ω^i et un vecteur δ (de même taille que les descripteurs) composé de facteurs d'actualisation avec des valeurs entre $0,1 < \delta_k < 0,5$ pour éviter de forts changements pendant l'actualisation. On détermine ${\Omega_k}^{i+1} = (1\text{-}\delta_k)\ \Omega_{ck}^i + \delta_k \omega_k^i$.

- Ensuite, on ajoute le point ω_{pos}^i à l'ensemble P_c^i. Cependant, dans le cas où il n'y a pas de trajectoire possible (c'est-à-dire card(T_p) =0), on crée une nouvelle trajectoire τ_c^n (d'indice n) où le descripteur de la trajectoire Ω_c^n est égal au descripteur de la détection ω^i non associé et $P_c^n = \{\omega_{pos}^i\}$.

Comptage multi-modal des personnes

Ce bloc représente la logique d'application pour une comptage multi-modal des personnes multi-modal (bloc d'application dans notre chaîne de traitement Fig. 3.25). Le caractère multi-modal signifie différentes logiques de comptage comme le passage de ligne virtuelle ou l'entrée et sortie de zones virtuelles. Cette logique est imposée par la logique du métier. Comme nous l'avons vu dans l'introduction, les applications de comptage des personnes dans les centres commerciaux font partie des objectifs de l'entreprise Shopline. Nous avons développé trois logiques différentes de comptage qui répondent à ces besoins de compter la quantité de personnes qui entrent et sortent d'un magasin. Cette information est sauvegardée par tranches horaires pour donner aux utilisateurs de notre système une information chronologique fiable sur le comportement des personnes suivies.

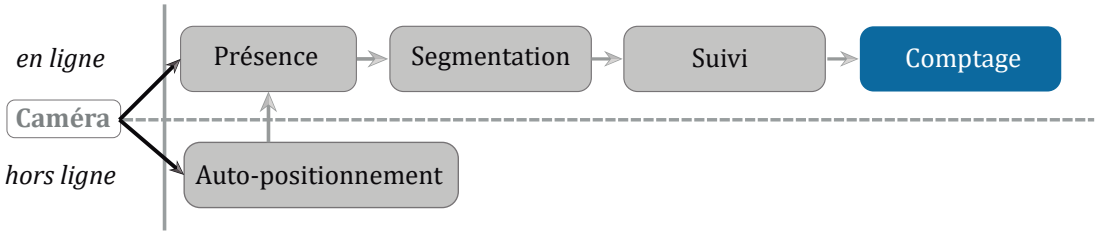


FIGURE 3.25 – Nouvelle configuration de la chaîne de traitement où le bloc de comptage (application) est ajouté.

D'abord, on décrit les paramètres généraux utilisés dans les différents types de comptage. Chaque type de comptage est composé d'une région d'intérêt (***Region of Interest*** : ROI) et d'un groupe de lignes de comptage comportementales L2C. Ainsi, une ROI délimite la région où la caméra est sensée suivre les personnes. Toute partie du champ de vision en dehors du ROI n'est pas prise en compte par la chaîne de traitement du comptage. La forme d'un ROI doit être composée d'au moins trois points exprimés dans le système de coordonnées (u,v) de l'image. De même, le groupe des L2C démarque où et comment les personnes doivent être comptées, accompagnées d'un conditionnement de comportement humain défini pour chaque type. Dans toutes ces méthodes, l'analyse de la trajectoire d'une personne est effectuée jusqu'à ce que la personne sorte du ROI, où l'on détermine alors si elle est entrée ou sortie du magasin.

Dans les travaux [Rau13, RB94, VZS13, TYOY99], on trouve différentes logiques de comptage pour des cas spécifiques, par exemple dans [VZS13], on compte dans deux directions au sein d'une zone séparée physiquement par des barrières. Il n'est donc pas possible pour une caméra visant la zone de trouver des personnes marchant dans une direction différente. On peut conclure qu'il n'y a pas de méthode générale ni d'ordre spécifique pour compter des personnes puisque le bloc d'application (dans notre cas le comptage de personnes) dépend complètement de la logique du métier envisagé pour résoudre une problématique particulière de ce métier et non pas les difficultés du suivi des personnes dans des espaces publics.

Comptage par ligne

La logique de comptage de cette méthode est composée d'un ROI et de lignes avec une direction de passage perpendiculaire (Fig. 3.6). Dans cette méthode, on compte les personnes qui traversent la ligne dans la direction considérée. On distingue deux types de lignes, l'une pour l'entrée et l'autre pour la sortie : on peut avoir plusieurs lignes du même type pour limiter des régions de comptage plus complexes, par exemple dans le cas où on a plusieurs entrées et une seule sortie (Fig. 3.26).

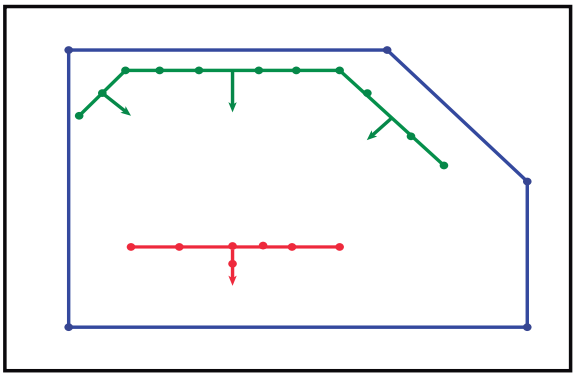
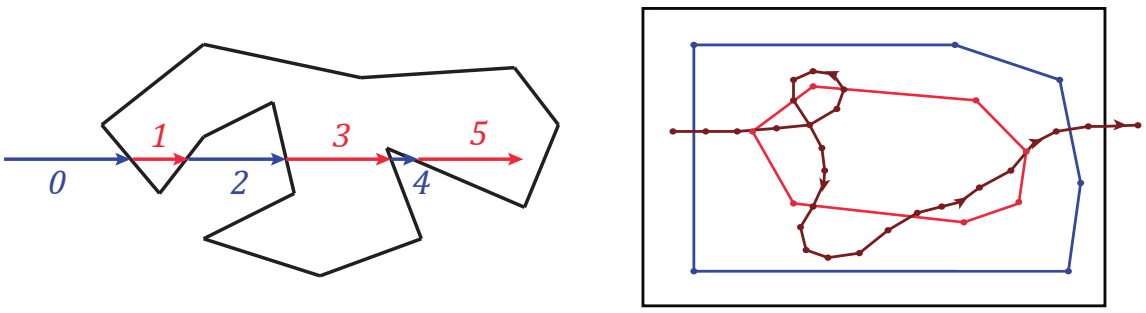


FIGURE 3.26 – Diagramme de la méthode de comptage par ligne. La zone avec un trait bleu représente le ROI, les lignes vertes les différents entrées, les flèches vertes la direction de passage en entrée, la ligne rouge de sortie et la flèche rouge la direction en sortie.

Comptage sur une région

On peut imaginer cette méthode comme la définition d'une deuxième région à l'intérieur du ROI et l'analyse d'une trajectoire au moment où la personne sort du ROI. Dans cette méthode, on compte une seule entrée et sortie par personne, même si la personne est sortie et entrée plusieurs fois dans la région. L'intérêt de cette condition de comptage permet aussi d'éviter les faux positifs. A la différence de la méthode précédente, on a intérêt à compter toutes les trajectoires dans toutes les directions(même si cela limite le comptage), en s'assurant que la même personne ne soit comptée qu'une fois. Il faut noter que l'identification se fait à l'intérieur de notre région, par conséquent une fois que la personne est sortie du ROI, la prochaine entrée de cette personne sera identifiée comme une personne différente.



(a) **(b)**

FIGURE 3.27 – Diagramme du comptage par région.

Comptage par zone

Dans cette méthode, on divise la ROI en trois zones différentes. La personne doit entrer dans une des zones et sortir par la zone opposée. Elle peut se balader à l'intérieur du FOV en traversant les zones, mais l'analyse de sa trajectoire est effectuée jusqu'à ce que la personne sorte de la ROI. Cette condition au moment du comptage nous permet d'éliminer les faux positifs. Le désavantage de cette méthode est que le comptage de personnes est fait seulement dans deux directions. Normalement, ce type de comptage est utilisé au niveau des portes du magasin, où l'on s'intéresse au comptage dans les directions d'entrée et de sortie par la porte. Les autres trajectoires qui traversent le FoV dans une direction différente sont ignorées.

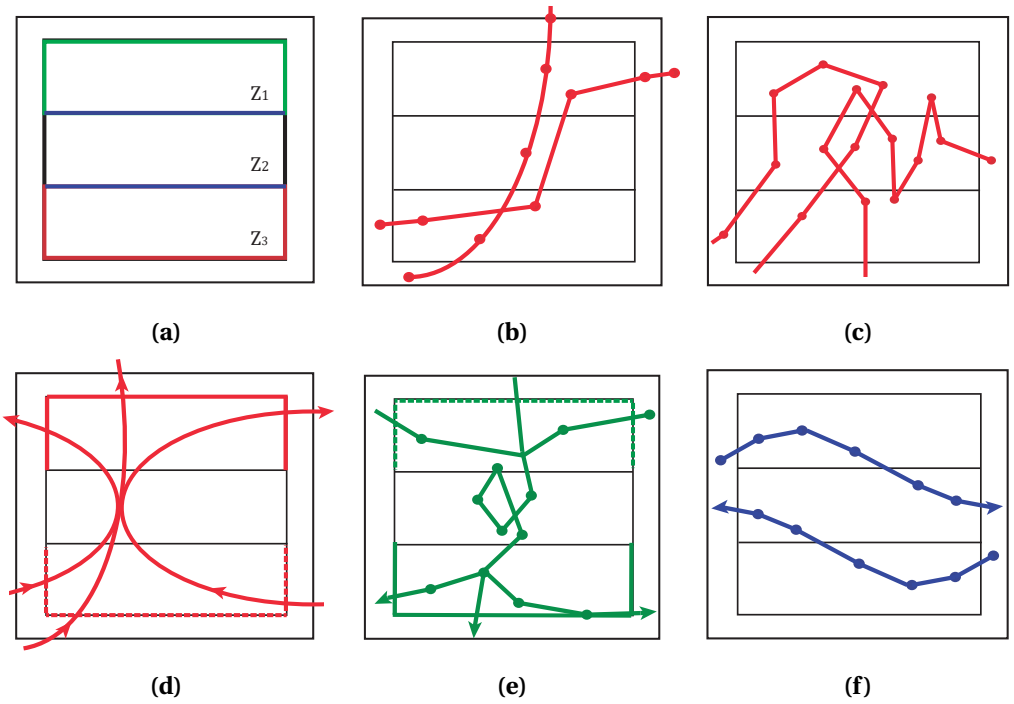


FIGURE 3.28 – Diagramme de comptage par zones intelligents. a) trajectoires valides, b) trajectoire invalides. c) En vert les lignes des entrées et en rouge les lignes des sorties.

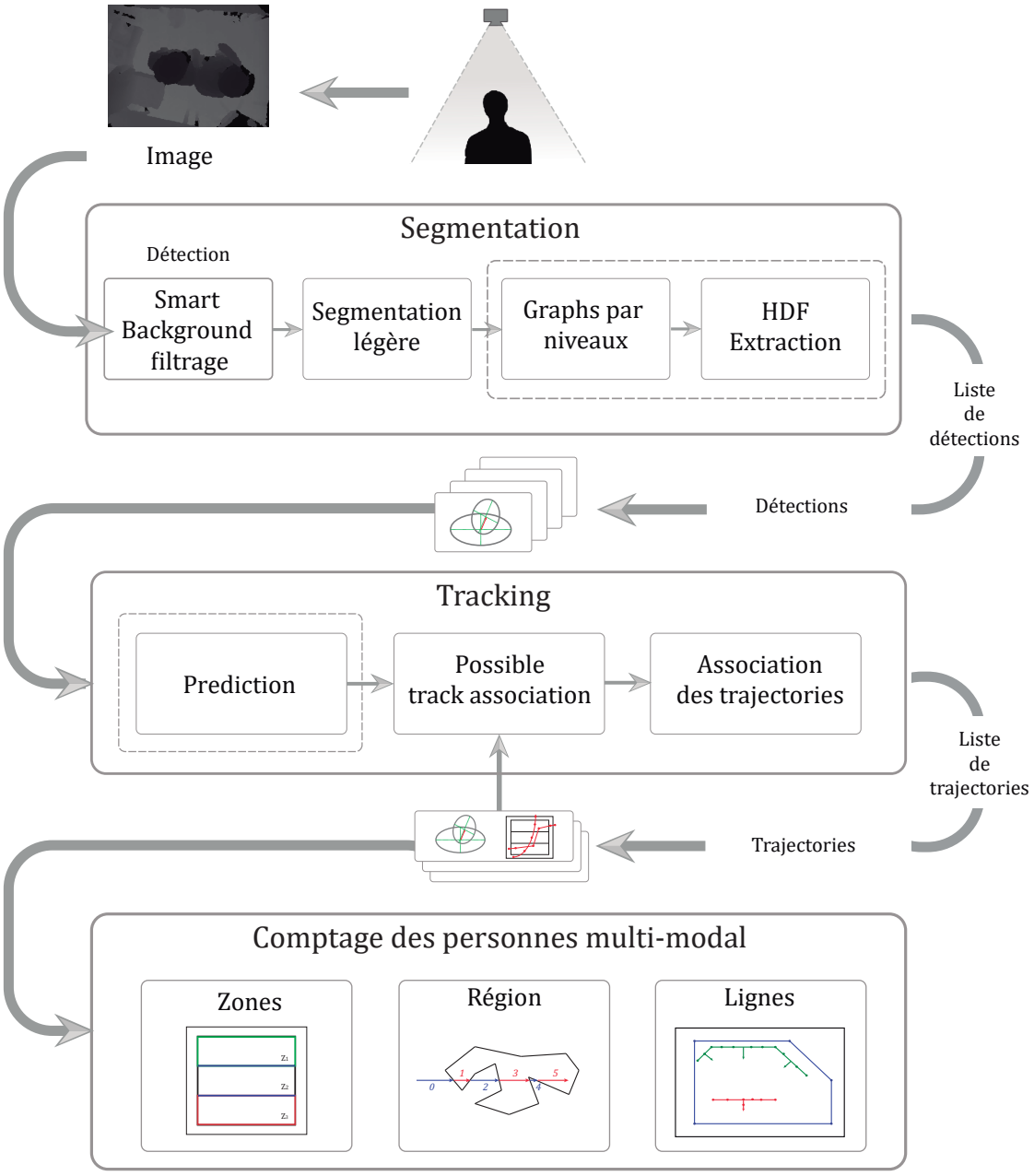


FIGURE 3.29 – Modèle final de la chaîne de traitement du système.

Dans la figure 3.29, on illustre la chaîne de traitement finale de suivi de personnes à l'intérieur de notre caméra intelligente proposé.

3.2.2 Conception de l'architecture physique du système

Nous avons décidé, au sein de Shopline, de développer notre propre solution de caméra intelligente de sorte de disposer de la flexibilité suffisante pour l'adapter à nos besoins et atteindre un prix de production compétitif par rapport à la concurrence. La philosophie de design, en termes de hardware et de software, est basée sur la construction de briques fonctionnelles interchangeables. On a alors conçu trois types de caméras intelligentes différents qui utilisent une caméra 2D, un système stéréoscopique de lumière structurée et un système stéréoscopique passif. Le système 2D a été développé pour deux raisons. D'abord Shopline a besoin d'avoir des systèmes de faible coût pour accéder aux marchés des pays avec une économie plus restreinte (voir introduction). Deuxièmement, c'est une voie pour commencer à comprendre les difficultés du suivi des personnes. Ensuite, on a conçu la solution stéréoscopique basée sur la

lumière structurée, qui utilise la caméra choisie dans le chapitre 2, et finalement, le système stéréoscopique passif qui est encore en étude et prototypage. Dans cette section, on se focalise sur la description matérielle et logicielle qui représente la base de nos caméras intelligentes. A l'exception du capteur, les descriptions qui vont suivre s'appliquent à tous les types de cameras.

Définition du matériel de caméra intelligente 3D

Dans cette section, on présente les caractéristiques essentielles de notre système : les capacités de la plateforme, son degré de distribution de calcul et l'autonomie du système. De plus, on décrit notre vision des briques hardwares et ses capacités d'interchangeabilité.

Capacités de la plateforme

Comme nous l'avons décrit dans la section précédente, les capacités d'une caméra intelligente sont données par le capteur, l'unité de calcul, les moyens de communication et l'alimentation électrique. Dans notre plateforme, on identifie le capteur comme la première brique HW. La deuxième brique est consacrée au traitement des données apportées par la brique précédente. Finalement, les moyens de communication et d'alimentation électrique sont intégrés dans une troisième brique HW.

- Caméra de profondeur : Comme résultat de l'étude d'acquisition de données, nous utilisons l'Asus Xtion Pro. Cette caméra peut capter jusqu'à 30 images par seconde en fournissant (à travers un lien USB) des images en profondeur à des résolutions VGA (640x480) ou inférieures. Chaque pixel de 12 bits représente la disparité stéréo locale et peut être traduit par la distance entre les objets acquis et la caméra. Lorsqu'il n'y a aucune information de détection due à des problèmes d'occlusion ou d'interférence, le pixel de profondeur est égal à 0. Les angles de FoV typiques sont 58 ° horizontalement et 45 ° verticalement.

- Ensemble du matériel de calcul : On a opté pour une carte embarquée avec tous les composants nécessaires à un système capable de traiter des données. On a testé notre solution algorithmique avec différentes cartes embarquées (L'Intel UpBoard, La ASUS tinker board et la Raspberry Pi 3). Ces cartes sont suffisamment puissantes mais seule la Raspberry-Pi 3 a un prix très concurrentiel. C'est celle que nous avons choisie. La Raspberry-Pi 3 est doté d'un processeur quadricœur Broadcom (BCM2837) à 1.2 GHz, d'un GPU double-cœur VideoCore IV, d'un SDRAM de 1 Go, d'une carte SD de classe 10, d'un port réseau Ethernet et d'un système d'exploitation Linux Rasbian.

- Module de communication et alimentation électrique : On a développé une carte propriétaire auxiliaire qui est principalement composée d'une connexion ***P**ower **O**ver **E**thernet* (POE) à 48V, un régulateur pour transformer les 48 V à 5V 1A et d'un circuit intégré ***R**eal-**T**ime **C**lock and **C**alendar* (RTCC). Cette carte nous permet, en même temps, de transporter les données et l'énergie dans le même câble physique, en facilitant l'installation et l'évolutivité de notre solution. De plus, la carte assemblée est conçue pour être aussi bon marché que possible et pour faciliter la fourniture de puissance.

Distribution du calcul

La distribution du calcul de notre système à l'intérieur de la caméra est divisée en : le calcul de la profondeur et le suivi des personnes. L'estimation de la profondeur est un processus lourd, dans notre système la caméra est en charge des estimations et de la livraison à la chaîne de traitement des images acquises. Ensuite, la chaîne de traitement transforme l'image de profondeur d'une résolution de 640 x 480 pixels (1228800 octets) dans une liste des personnes détectées représentées par un HFD d'une taille de 104 octets et de 12 octets pour chaque position (détection sur une image) dans le système.

L'adaptation et l'autonomie d'installation

Au moment de l'installation de la caméra, on a comme moyen disponible de déploiement une interface web qui permet de configurer et d'ajuster les paramètres de la caméra. De plus, le bloc d'auto- positionnement permet au système de s'ajuster en dépendant de l'environnement où la caméra est installée. Notre solution n'inclue pas la mobilité de la caméra car elle n'est pas nécessaire pour le comptage des personnes, sans parler des coûts supplémentaires induits par un système mécanique et logiciel complexe.

Composants logiciels

Dans cette section, on décrit la conception logicielle de notre système de manière à répondre aux difficultés de suivi des personnes et, en même temps, de répondre aux besoins industriels comme la facilité d'utilisation et de configuration ainsi que le déploiement.

Nous avons défini une architecture orientée composée de briques logicielles interchangeables selon les besoins. Sur ce point précis, nous devons répondre à d'autres défis industriels importants qui s'ajoutent à ceux décrits dans l'introduction. Il s'agit de la réutilisation de code source, la réduction du temps d'accès au marché dans le développement de nouveaux produits, l'interopérabilité entre architectures et composants, et la cohérence logiciel/interface pour satisfaire les utilisateurs. Ces exigences doivent être satisfaites en assurant la performance et la précision de nos algorithmes.

Notre système est composé de trois grandes couches logicielles (Fig. 3.30). La première couche gère une série de pilotes ou « middleware » qui nous permet d'acquérir les images de la caméra d'une manière transparente avant d'attaquer la chaîne de traitement (Figure 3.29). Nos efforts à ce niveau sont de fournir une image standard (3D et/ 2D selon le cas). Par exemple, on utilise différent types de caméras 2D qui utilisent différents pilotes. Ensuite, la seconde couche de suivi des personnes met en œuvre des algorithmes décrits dans la section 3.2.1 (cas des caméras 3D). De plus, dans cette couche, on transmet les résultats de comptage à la couche suivante. Finalement, la troisième couche est une couche web qui gère la configuration et le déploiement de la caméra d'une manière la plus aisée possible. Cette couche web nous permet de communiquer les résultats de comptage avec le système logiciel de Shopline et de gérer la GPIO pour permettre d'interagir avec les utilisateurs finaux (par exemple, on peut choisir d'allumer des LEDs de couleurs différentes dans le cas où une personne entre ou sort). De plus, on gère le temps RTCC à travers des serveurs spécialisés en utilisant le protocole protocole d'heure réseau (***N**etwork **T**ime **P**rotocol* : NTP), pour synchroniser les messages avec les différentes entités du système (serveur et autres caméras, sujet davantage développé plus loin dans le chapitre suivant).

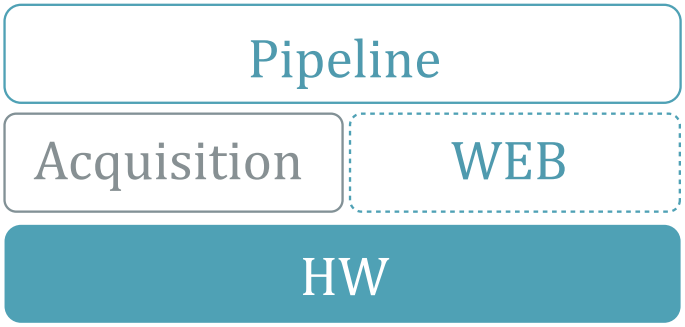


FIGURE 3.30 – Diagramme des couches logicielles de notre solution pour caméras intelligentes.

Cette architecture est valide pour différents types des caméras 2D (couleur, IR, échelle de gris) et 3D (Stéréo-vision, lumière structurée), en réécrivant certaines briques sans revoir le logiciel dans son ensemble. Le gros avantage est que le suivi, le comptage et la couche de configuration est réutilisable qu'elle que soit la technologie de la caméra (2D ou 3D), même si la méthode d'acquisition et de traitement des images est différente entre les deux. De plus, l'évolution des différentes méthodes et mises en œuvre pour la segmentation ou le suivi sont interchangeables au travers de paramètres de configuration.

Mise en œuvre

Il y a deux aspects importants de cette mise œuvre à mettre en évidence. D'abord, la création d'un modèle orienté objets OO capable de donner une flexibilité à la gestion des différents composants de notre système embarqué. Deuxièmement, l'utilisation de technologies orientées pour l'optimisation d'utilisation des ressources limitées.

Notre système est développé à partir de technologies multiplateforme comme la *programmation* orientée objets OO en utilisant c++, la *gestion de ressources* du système d'exploitation à travers QT, OpenCV et OpenMP, ainsi que la création d'une interface HMI web et des services web en utilisant l'API RESTful en Python.

Techniques d'optimisation du code pour des systèmes embarqués

La recherche de l'optimisation pour le développement dans des systèmes embarqués est basée sur trois axes : la performance, la mémoire et l'énergie. On aborde dans cette section les différents moyens d'optimisation et de parallélisation utilisés dans la mise en œuvre des algorithmes de cette thèse. De plus, on exploite les différentes méthodes et outils au moment de la compilation, de l'utilisation de la mémoire et de la parallélisation.

Compilation du logiciel

Le processus de compilation traduit le langage C++ vers un langage assembleur lié à des bibliothèques externes utilisées dans les programmes. Ce processus est automatique, dépend de la plateforme (architecture du processeur) et génère un fichier exécutable. Il optimise le code basé sur la performance et sa taille de l'exécutable final, en utilisant les règles définies dans le compilateur. Normalement, le compilateur génère un module à partir de chaque fichier C, ensuite il optimise le code à l'intérieur de chaque module et finalement relie tous les modules générés et externes pour créer l'exécutable. Cependant, dans certains cas, on peut changer ces règles pour optimiser notre code à travers le mode de compilation, des directives *pragma* ou des annotations.

De plus, les instructions du logiciel sont divisées en pages dont la taille dépend de l'architecture du système, et qui sont chargées à l'intérieur de la mémoire RAM et de la mémoire cachée du processeur selon les besoins au moment de l'exécution. Si on imagine un logiciel où les instructions sont complètement séquentielles, le chargement de ces pages dans la mémoire est également séquentiel. Les instructions conditionnelles (IF, ELSE) sont représentées par des sauts sur les pages du logiciel. De plus, les instructions en boucle (FOR, WHILE) sont représentées par des retours au début de la boucle. On peut imaginer le cas où le nombre des instructions d'une boucle est égal à la taille des instructions par page. L'exécution de cette boucle peut éviter le chargement des nouvelles pages dans la mémoire du processeur pendant l'exécution des cycles de la boucle. Comme le chargement d'une page dans la mémoire prend du temps, notre objectif est de minimiser le nombre de chargements et ainsi augmenter la performance. Les chargements excessifs des pages sont appelés « overhead ». Ce procédé de chargement des instructions dans le processeur est en fait plus complexe mais il sert tout de même à exposer l'intérêt de l'utilisation des différents modes de compilation, d'annotations « *inline* » et des directives « *pragma* » décrites ci-après.

En premier lieu, on compile une série de fichiers C++ qui appartient à notre logiciel pour générer un exécutable binaire. Dans la compilation traditionnelle, on génère un module pour chaque dossier C++. Ensuite, pour chaque module le compilateur optimise le code pour finalement relier tous les modules dans un seul exécutable. Cependant, Il existe six niveaux de compilation : O0) le premier niveau est une compilation non optimisée où le code est traduit simplement en assembleur ; O1) le deuxième niveau réalise une optimisation de haut niveau indépendante du processeur cible. Celle-ci permet de réduire la taille et le temps d'exécution de l'application ; O2) le troisième niveau réalise les mêmes optimisations que O1 en ajoutant celles reliées au processeur cible ; O3) le quatrième niveau reproduit les mêmes optimisations que O2 et en plus, il définit des registres globaux en permettant l'accès le plus rapide possible aux objets les plus utilisés ou en activant des annotations *inline ;* Os) le cinquième niveau est une optimisation

reliée à la taille de l'exécutable résultant. Finalement, Og) le sixième niveau est une compilation optimisée pour faire du débogage.

En second lieu, on peut utiliser les annotations *inline*. Normalement, quand on fait appel à une fonction, le processeur doit, en sautant de page en page, trouver cette fonction, l'exécuter et finalement retourner au point d'exécution avant l'appel de la fonction. L'annotation *inline* permet de copier les instructions qui composent la fonction dans les parties du code où la fonction est appelée. Cette copie des instructions *inline* nous permet de diminuer les pas supplémentaires pour les appels de fonction. En conséquence, on gagne en performance, mais on obtient un exécutable d'une taille plus grande. Par exemple, l'encapsulation est un concept très important et utile dans la programmation OO. Cela nous permet d'extraire les comportements et les attributions de variables (getter et setters) qui appartiennent à une classe dans des fonctions pour sécuriser le comportement du code (cohérence) et éviter des intrusions indésirables (sécurité). Cependant, l'attribution d'une valeur à une variable en utilisant une fonction, dans le code compilé, conduit à des pas (sauts) supplémentaires au détriment de la performance du logiciel (overhead). L'annotation *inline* de ces fonctions contribue à faire gagner de la performance.

En troisième lieu, on utilise des directives *pragma* pour donner des informations additionnelles au compilateur, ce qui n'est pas possible dans le langage c++. Parmi les *pragmas* possibles, on se sert de ces directives : i) on utilise « *dependency* » pour définir la bibliothèque à relier, laquelle dépend de l'architecture, et ii) on utilise le dépliement « *unroll* » de certaines parties du code pour accélérer la performance (copier les instructions d'une boucle pour éviter de faire des comparaisons de fin de boucle). Cependant, l'utilisation de ces instructions incrémente la taille du binaire résultant, ce qui revêt une importance cruciale pour limiter les ressources. Or si on applique la directive « unroll » à l'analyse de tous les pixels d'une image, la taille du binaire devient énorme. Si on restreint l'utilisation de la directive à l'analyse des 15 derniers points de la trajectoire pour déterminer le comportement des entrées et sorties, on améliore la performance sans alourdir la taille du binaire.

L'allocation statique et l'accès séquentiel de la mémoire

L'allocation de mémoire est une opération essentielle pour n'importe quel processus informatique car cette ressource sera utilisée pendant toute sa durée de vie. On a deux types d'allocation de mémoire : l'allocation statique dont la mémoire allouée a une taille fixe qui ne change pas pendant la vie du processus et l'allocation dynamique qui est variable dans le temps et qui concerne des zones de mémoire qui peuvent changer de taille. Pour ce dernier cas, si les demandes sont mal maitrisées ou mal gérées, on risque de ne pas avoir assez de mémoire libre disponible, ralentissant ainsi la performance de l'application parce que le système doit utiliser d'autres ressources (comme le fichier de swap) pour trouver la place nécessaire. Par ailleurs, l'allocation de mémoire consomme en général du temps de calcul. Pour cette raison, on utilise l'allocation de mémoire statique en définissant les images comme vecteurs de taille fixe, au début de l'application, pour chacune des étapes de traitement comme l'extraction des pixels d'intérêt, l'étiquetage, la segmentation par niveau, etc. Il en résulte deux avantages. Le premier permet de gagner de la performance en accédant d'une manière séquentielle au bloc de mémoire qui représente une image (on peut imaginer qu'on lit les pixels de l'image ligne par ligne horizontalement). Le second permet d'éviter la fragmentation de la mémoire. Toutes les deux réduisent le temps d'accès à la mémoire et les sauts supplémentaires « overhead ». De cette façon, le coût en temps de calcul de l'allocation de mémoire est pris au moment du démarrage. Par ailleurs, la création et la destruction d'objets a besoin d'allocation de mémoire dynamique et consomme du temps de calcul, ce qui est à éviter le plus possible.

L'alignement des données

L'alignement des données concerne la manière avec laquelle la mémoire est organisée et se lit. Ceci a un poids important dans la performance de notre logiciel. Chaque type de donnée a un alignement associé qui permet au processeur d'aller chercher les données d'une manière efficace dans la mémoire et dépend de l'architecture du processeur. Par exemple, dans une architecture de 32-bit, un pointeur a une taille de 8 octets tandis que dans une architecture de 64-bit, il a

une taille de 16 octets. De plus, en C++ une structure est le rassemblement de plusieurs éléments alignés à la valeur (d'alignement) plus représentative entre eux. Cette valeur a une taille exprimée en puissance de 2 et connue comme « *mot* ». Les adresses de chaque élément sont assignées en multiples de la taille de « mot ». Donc, les compilateurs organisent les structures des données dans la mémoire RAM par rapport à son alignement dans le même ordre qu'ils sont déclarés dans le code source. Par conséquent, quand il y a des données d'une taille d'alignement plus petite que la taille de « mot », le compilateur essaie de conserver l'alignement en ajoutant de la mémoire additionnelle à la fin de l'élément et celle-ci n'est pas utilisée (en pressentant des problèmes d'utilisation de mémoire). Cette action du compilateur est connue sous le nom de « Padding ». De plus, l'effet d'ajouter de la mémoire non utilisée à la fin de la structure quand la taille de l'ensemble de la structure n'est pas alignée, est connu sous le nom de « Tail Padding ». Afin de minimiser ces problématiques, on doit déclarer ces structures, en organisant les données du plus grand alignement au plus petit. De cette manière, les espaces en mémoire non utilisée se réduisent qu'à « Tail Padding » en évitant un gaspillage de mémoire RAM.

De plus, il y a des directives pour forcer le compilateur à grouper les données sans ajouter de mémoire. Ce groupement implique cependant des opérations supplémentaires au moment de la lecture des données en dégradant la performance.

Parallélisation

Finalement, on se sert de la parallélisation à deux niveaux. Dans le premier niveau, on crée des *threads* de longue durée pour l'acquisition des images, leur traitement et la communication entre notre système et l'extérieur. Ces *threads* travaillent de manière continue et sont capables d'appeler des t*hreads* de deuxième niveau. Dans un deuxième niveau, on trouve des tâches de courte durée comme l'extraction des graphes, des HFDs pour chaque personne détectée ou l'évaluation des trajectoires pour déterminer si une personne doit être comptée dans la brique d'application. Ce deuxième type de *thread* est aussi utilisé au travers des bibliothèques OpenCV et OpenMP de manière automatique en les compilant ensemble dans l'architecture désirée.

Bonnes pratiques

Dans cette section, on décrit les bonnes pratiques de développement pour les systèmes embarqués qui ont émergé pendant l'élaboration de notre solution, dans le but d'éviter au maximum la saturation des canaux de communication pendant la lecture des données ou l'exécution inutile d'instructions au niveau CPU. Ces bonnes pratiques se déclinent sur plusieurs niveaux.

Au niveau des fonctions :

- Avoir moins de 4 paramètres dans les fonctions C++ (dépend de taille du registre de CPU).
- Passer de paramètres reliés dans une structure e.g. « Point3D » à la place de x, y et z.
- Utilisation massive de pointers pour éviter les copies inutiles de données.
- Éviter un nombre variable de paramètres dans les fonctions (e.g. printf).
- Annoter les fonctions (e.g. inline, pure, etc.) selon le type de compilation.

Au niveau des variables :

- Réaliser une allocation statique de la mémoire « malloc » et pas dynamique.
- Minimiser la création et la destruction d'objets car l'utilisation de mémoire dynamique coûte cher.

- Annoter les variables dont l'accès est unique avec *restrict* pour optimiser le code (e.g. les fonctions setter).

Au niveau des opérations arithmétiques :

- Éviter les divisions en privilégiant les multiplications.
- Éviter des opérations entre des variables de point floutant et évaluer le type de variable requis (e.g. short, int, etc) pour éviter de surdimensionner le système.

Au niveau de la compilation :

- Toujours compiler en mode « release » pour sécurité, performance et taille d'exécutable.

3.3 Résultats

Cette section évalue les résultats de différentes mises en œuvre du système décrit le long de ce chapitre. On évalue d'abord la performance de l'algorithme de détection (et segmentation), ensuite la performance de l'algorithme de suivi et finalement la capacité de calcul de nos mises en œuvre (mono et multi-thread) dans différents systèmes embarqués. Pour chaque expérience, on décrit la méthodologie d'évaluation ainsi que les paramètres à évaluer. Ces expériences nous permettent d'évaluer si notre solution répond aux besoins industriels et nous permettent de surmonter les difficultés décrites dans l'introduction (section 1.3).

3.3.1 Évaluation de la détection

Dans cette section, on décrit la méthodologie d'évaluation, la méthodologie et la politique d'annotation et les paramètres à évaluer dans l'algorithme de détection de personnes en utilisant une caméra 3D en position zénithale.

Méthode d'évaluation

On utilise la méthode pour évaluer la détection de personnes, proposée par [DWSP09] dite l'évaluation « par image ». Cette méthode d'évaluation consiste à créer des annotations (vérité-terrain) de la position des personnes sur toutes les images de la vidéo à analyser et les comparer avec les détections fournies par l'algorithme de détection. Comme on l'a décrit au début de ce chapitre, les annotations sont représentées par BBs. Ainsi, une détection $\mathrm{BB_{dt}}$ fournie par l'algorithme est mise en correspondance avec une annotation (vérité-terrain) $\mathrm{BB_{gt}}$ si la région de chevauchement s_0 atteint au moins 50 % de la surface de $\mathrm{BB_{gt}}$:

$$s_0 = \frac{surface(\mathrm{BB}_{dt} \cap \mathrm{BB}_{gt})}{surface(\mathrm{BB}_{dt} \cup \mathrm{BB}_{gt})} > 0.5 \tag{3.8}$$

En utilisant l'équation 3.8, on peut déterminer si une détection de notre algorithme est un vrai positif (VP), ou un faux positif (FP). Dans le cas où il n'y a pas de détection associée à une annotation, alors c'est un faux négatif (FN). A partir de cette évaluation des détections, on utilise trois critères pour mesurer la performance de notre algorithme : la précision, le rappel (Recall en anglais) et le ratio de faux négatifs (Miss rate en anglais).

Le *rappel* (R) est défini comme le nombre de détections correctement trouvées (VP) au regard du nombre total de détections annotées (VP+FN). Ce critère évalue la capacité de l'algorithme à détecter les personnes dans la scène. Ainsi, une valeur plus élevée dénote un algorithme capable de trouver plus correctement les personnes.

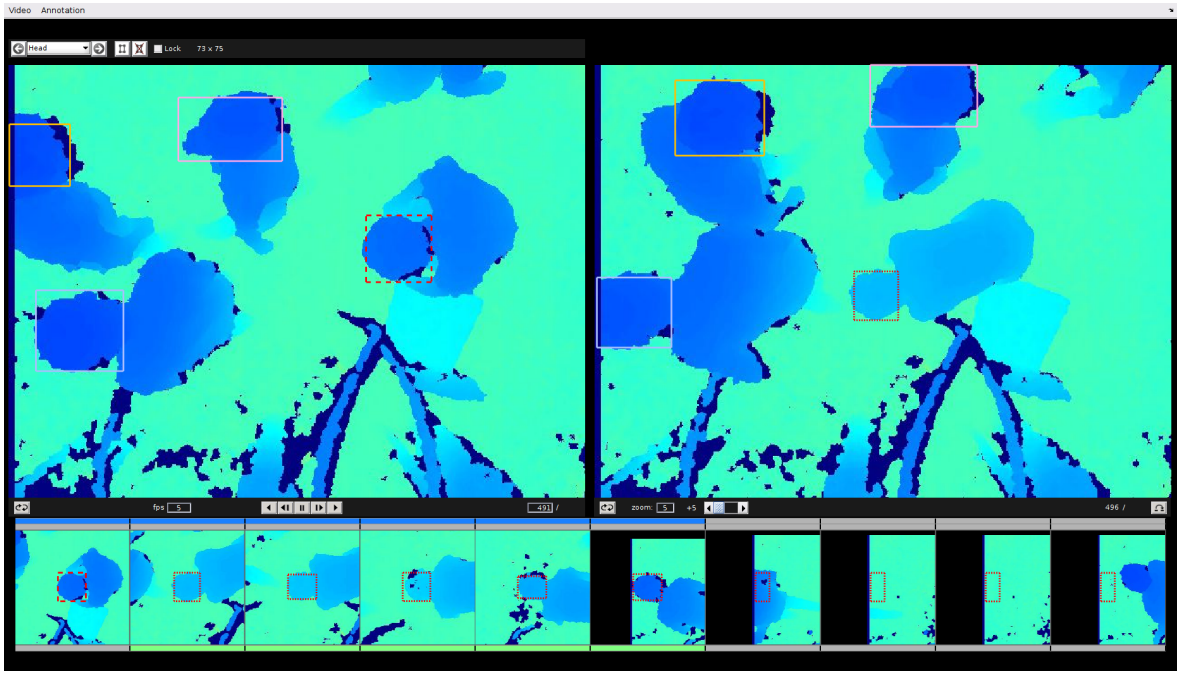


FIGURE 3.31 – Outil d'annotation de vidéos en Matlab. A gauche l'image initiale, à droite l'image finale et en bas les images intermédiaires avec l'interpolation de la position de l'étiquette.

La *précision* (P) est définie comme le nombre de détections correctement trouvées (VP) par rapport au nombre de détections de l'algorithme (VP+FP). Ce critère évalue la pertinence des détections trouvées, indépendamment de la quantité de personnes correctement trouvées. Une valeur plus élevée de ce critère dénote un algorithme où chaque détection est plus pertinente.

Finalement, le ratio de faux négatifs (MR) est défini comme le rapport entre les fausses détections et les détections correctement trouvées. On cherche à minimiser ce critère, ce qui signifie que l'on diminue le taux de fausses détections par rapport aux détections correctement trouvées.

Plus formellement les critères sont définis comme :

$$\mathrm{R} = \frac{\mathrm{VP}}{\mathrm{VP}+\mathrm{FN}}, \mathrm{P} = \frac{\mathrm{VP}}{\mathrm{VP}+\mathrm{FP}}, \mathrm{MR} = \frac{\mathrm{FN}}{\mathrm{VP}}. \tag{3.9}$$

Méthode d'annotation

Pour générer les annotations, nous avons utilisé la toolbox de Caltech en Matlab (Fig. 3.31) présentée par [DWSP09]. Cet outil permet d'annoter la vérité-terrain de manière semi-automatique, en sélectionnant le point initial de la BB dans une image de la vidéo et la position finale de la même BB quelques images après, interpolant les positions intermédiaires automatiquement. Si nécessaire, on peut régler les positions intermédiaires générées.

On exporte les vidéos de profondeur sous un format « avi », en utilisant une échelle de couleur pour faciliter la visibilité des têtes des personnes et la différenciation du sol. Ensuite, on transforme la vidéo en format « seq » qui est le format utilisé par l'outil de Caltech.

Notre politique d'annotation impose qu'une annotation soit valide si la tête de la personne annotée est visible au moins à 40 % dans l'image. Si la personne annotée ne respecte pas la condition précédente, elle est considérée comme étant en dehors du champ de vision, clôturant ainsi son annotation. Si la même personne rentre dans la scène, on utilise une nouvelle étiquette.

Paramètres à évaluer

Notre algorithme a sept paramètres qui permettent une certaine liberté de configuration selon le cas d'utilisation et l'environnement dans lequel la caméra est installée. Ces paramètres sont normalement liés à un bloc de la chaîne de traitement et aussi entre eux.

Dans la « modélisation d'arrière-plan », on utilise un premier paramètre t_d qui est la distance utilisée pour segmenter les pixels du premier plan. En plus, on rajoute un seuil de filtrage additionnel pour les pixels proches de la caméra que l'on appellera ${t_d}^{min}$. Ceci nous permettra d'éliminer les fausses mesures très proches de la caméra dues par exemple au phénomène de réflexion de certaines surfaces. Pour plus de clarté, notre paramètre original t_d sera renommé ${t_d}^{max}$. Ces deux paramètres sont exprimés en millimètres.

Dans la « segmentation légère », on utilise les paramètres a_0^{min} et a_0^{max} pour déterminer si l'on prend en compte la région à évaluer ($\#\Theta_n < a_0^{min}$) ou si l'on doit chercher plusieurs personnes dans cette région ($\#\Theta_n > a_0^{max}$). Ces deux paramètres sont exprimés en nombre de pixels.

Dans la « Segmentation par graphe de niveaux », on utilise le paramètre t_c pour définir l'épaisseur des tranches horizontales dans la construction du graphe. De plus, pour trouver les têtes, nous recherchons les nœuds racines avec une surface minimale ${H_s}^{min}$ pour discriminer les parties du corps comme les mains et les bras situées au-dessus de la tête et une surface maximale ${H_s}^{max}$ pour éliminer des objets comme les sacs à dos qui pourront être pris comme une deuxième tête. Le paramètre t_c est exprimé en millimètres et les autres en nombre de pixels.

En somme, on a : ${t_d}^{min}$ et ${t_d}^{max}$ pour détecter les pixels du premier plan, ${a_0}^{\min}$ et ${a_0}^{\max}$ pour décider si l'on fait une deuxième segmentation, t_c pour couper en niveaux notre région, et finalement ${H_s}^{min}$ et ${H_s}^{max}$ pour discriminer les nœuds racines dans la création du graphe.

Jeu de données

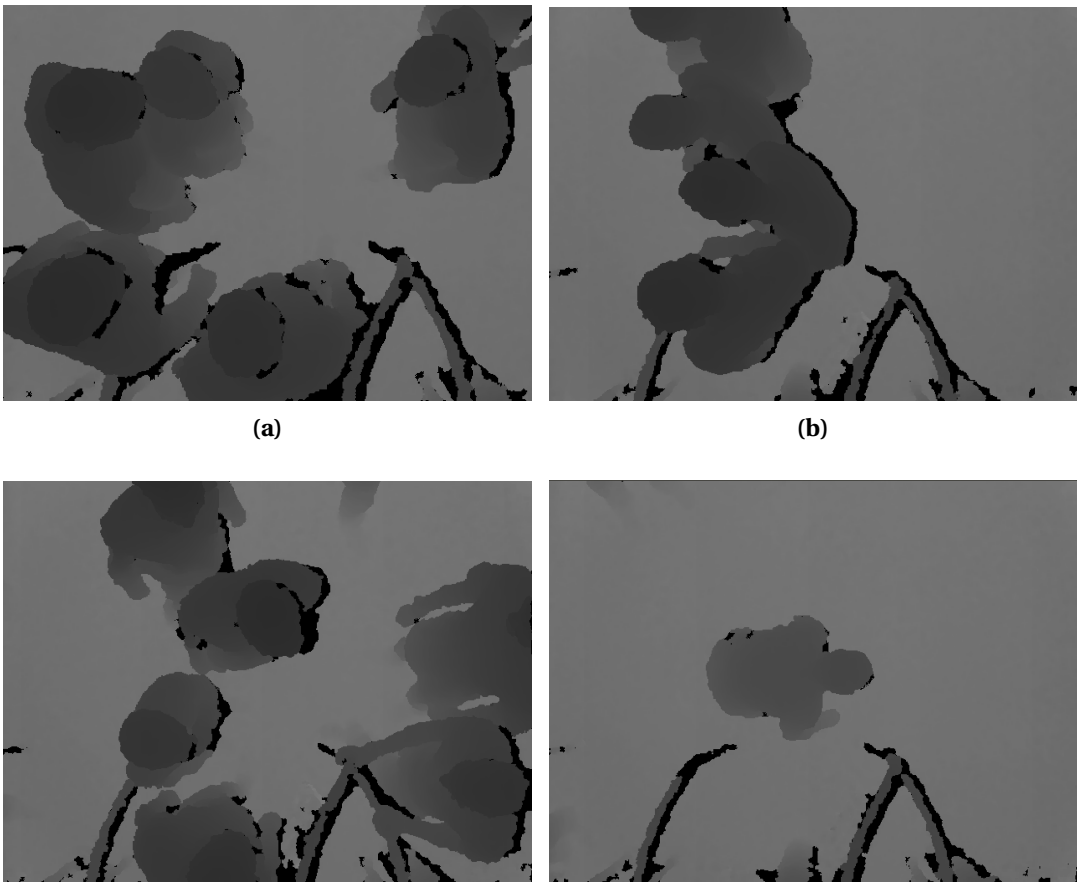
On a généré un jeu de données à partir des images fournies par une caméra de profondeur en position zénithale. Notre jeu de données se compose des annotations sur 7000 images d'une scène occupée par plusieurs personnes (jusqu'à 6) en mouvement à différentes vitesses (marchant et courant) observées par des caméras montées en position zénithale à différentes hauteurs d'installation. Ce jeu de données comprend aussi des cas difficiles où le gens se déplacent tout en étant très proches entre aux, ce qui rend difficile la segmentation. Il s'agit de quatre scenarii en particulier (voir Fig. 3.32) :

- a) Une scène où jusqu'à 4 personnes marchent en même temps, en se touchant latéralement.
- b) Une autre scène où l'on assiste à des occultations entre personnes, du fait de la différence de hauteur entre les personnes et de leur position en marchant.
- c) Une autre scène où 6 personnes sont très proches les unes des autres, en formant des groupes de 2 à 4 personnes dans une seule région du premier plan.
- d) Une autre avec des personnes marchant en pliant les jambes.

Résultats de l'algorithme de détection

On présente pour chaque paramètre l'évaluation des critères de précision et de rappel. L'évaluation du ratio de faux négatifs est présentée seulement si le comportement de la réponse de l'algorithme au critère est monotone. Par extension, on va désigner par monotone tout paramètre qui provoque un tel comportement.

Ensuite, on fait le choix de fixer une valeur optimale pour tous les paramètres non-monotones et de laisser à l'utilisateur le contrôle des autres paramètres monotones, par exemple la hauteur d'installation, le filtrage de caddies, etc.



(a) **(b)**

(c) **(d)**

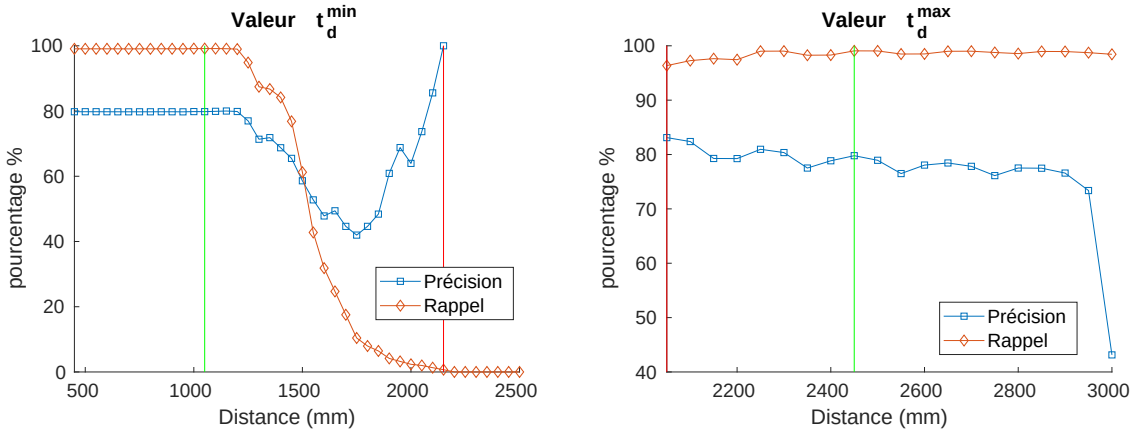
FIGURE 3.32 – Exemples d'images extraites de quatre scenarii (de a à d).

Les figures présentées ci-dessous nous montrent en bleu le pourcentage de la précision, en orange le pourcentage du rappel, en rouge une ligne horizontale qui représente la valeur où la précision est la plus élevée et en vert une autre ligne horizontale qui montre la valeur de rappel la plus élevée. L'axe X représente la valeur du paramètre évalué exprimée en millimètres ou pixels, selon le paramètre. Finalement, l'axe Y représente, en pourcentage, la valeur résultante des critères évalués.

Sur chaque paramètre, on cherche le meilleur équilibre entre précision et rappel. Dans le comptage des personnes, détecter toutes les personnes est très important parce que ceci impacte le salaire des vendeurs. Dans notre cas d'utilisation, on va favoriser le rappel sur la précision, d'abord parce que la majorité de FNs sont liés à des effets de bord et en deuxième lieu parce que les FNs de courte durée n'affectent pas le comptage, ils ne génèrent pas de trajectoire valide, juste de petites trajectoires de quelques points. De plus, on observe deux types de comportement des résultats de l'évaluation des paramètres. Le premier type de comportement fourni un seul point maximal dans la courbe du rappel (Fig. 3.35) et le deuxième fourni une plage de valeurs du paramètre à évaluer dans laquelle on obtient la valeur maximale du rappel (Fig. 3.34). Ainsi, on dénote la meilleure valeur d'un paramètre comme : la valeur du paramètre pour laquelle le rappel est maximal dans le premier cas et la valeur du paramètre pour laquelle la précision est maximale à l'intérieur de la plage du rappel dans le deuxième cas.

Voici les résultats de l'évaluation des paramètres : d'abord, on évalue les paramètres qui définissent la plage de profondeur $[{t_d}^{min}, {t_d}^{max}]$ où la caméra sélectionne les pixels qui appartiennent au premier plan. On évalue ${t_d}^{min}$ entre les valeurs 400 et 2500 mm et ${t_d}^{max}$ entre 2000 et 3000 mm. Dans l'évaluation de ${t_d}^{min}$, on obtient le meilleur rappel en fixant le paramètre entre 400 et 1100 mm (Fig. 3.33a). Cependant, la meilleure précision correspond à une valeur très basse de rappel. Ce paramètre ne présente pas un comportement monotone, donc on sélectionne une valeur à l'intérieur de la plage maximale de rappel qui doit convenir à la majorité de cas réels. Par exemple, en prenant en compte 2,10 mètres comme la hauteur maximale de personnes et 3

mètres comme la hauteur d'installation de la caméra, on doit fixer $t_d{}^{min}$ à 800 mm. Finalement, pour le paramètre $t_d{}^{max}$, on choisit la valeur de rappel maximale étant donné que : on ne cherche pas à filtrer d'enfants, on cherche à compter les personnes qui marchent en pliant les jambes et que le paramètre n'a pas un comportement monotone.



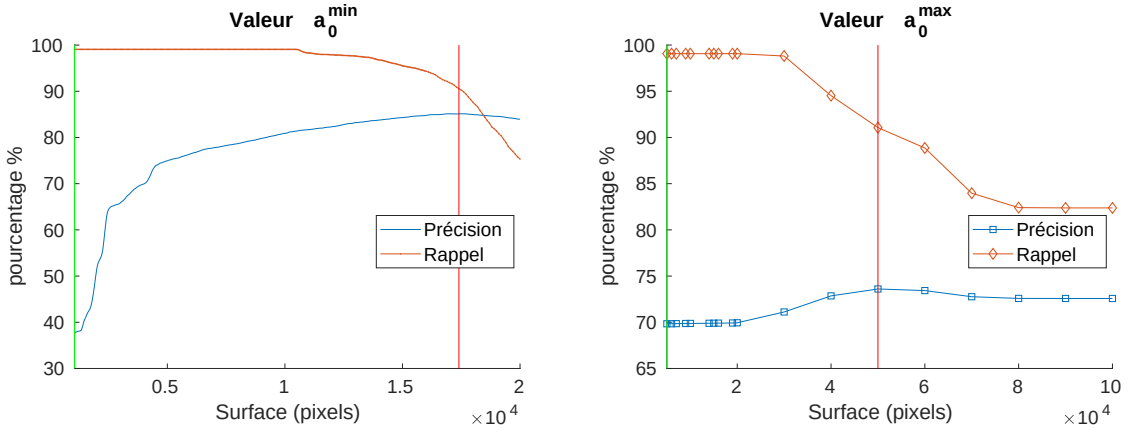
(a) **(b)**

FIGURE 3.33 – Représentation de résultats de précision et rappel en fonction des paramètres $t_d{}^{min}$ et $t_d{}^{max}$.

Dans la figure 3.34, on évalue l'influence de la taille de la surface des régions dans la segmentation légère $[a_0{}^{\min}, a_0{}^{\max}]$. On évalue la taille minimale de la surface d'une région $a_0{}^{min}$ entre les valeurs de 1000 et 20000 pixels et la taille maximale de la surface d'une région $a_0{}^{max}$ entre 4000 et 100000 pixels.

En évaluant $a_0{}^{\min}$, on trouve que ce paramètre a une forte influence dans les FN parce qu'il est capable de filtrer rapidement les petites régions qui ne contiennent pas de têtes. Il a moins d'influence dans les VP lorsque que sa valeur est inférieure à 10000 pixels, qui représente la taille de la surface la plus petite du jeu de données. Dans le cas où sa valeur est supérieure à 10000, il devient contre-productif parce qu'il évite la détection des personnes dans la scène. Ce résultat confirme notre première approche qui consiste à fixer la limite inférieure pour prendre en compte une région à 50 % de la taille moyenne.

On observe la faible influence de $a_0{}^{\max}$ par rapport aux grandes valeurs qu'il prend. Il commence à diminuer le rappel au moment où sa valeur a la surface moyenne occupée par deux personnes (en évitant le deuxième niveau de segmentation). Par contre, on estime que ce paramètre a plus d'influence sur la vitesse de traitement parce qu'il évite de faire une recherche d'une deuxième personne dans de petites régions.



(a) **(b)**

FIGURE 3.34 – Représentation de résultats de précision et rappel en fonction des paramètres a_0^{min} et a_0^{max}.

Dans la figure 3.35, on évalue t_c entre les valeurs de 150 et 500 millimètres, une valeur plus élevée reviendrait à ne pas réaliser de deuxième segmentation. On trouve que ce paramètre est le plus sensible dans notre algorithme puisqu'il coupe horizontalement les régions pour chercher les têtes. Quand cette coupe tombe correctement dans la plage de valeurs (là où se trouve la hauteur de têtes observées), cela permet d'avoir un rappel maximal. Au contraire, il induit de faux négatifs. Le meilleur résultat (où il a le rappel maximal et le pic de précision) est une coupe à 200 millimètres.

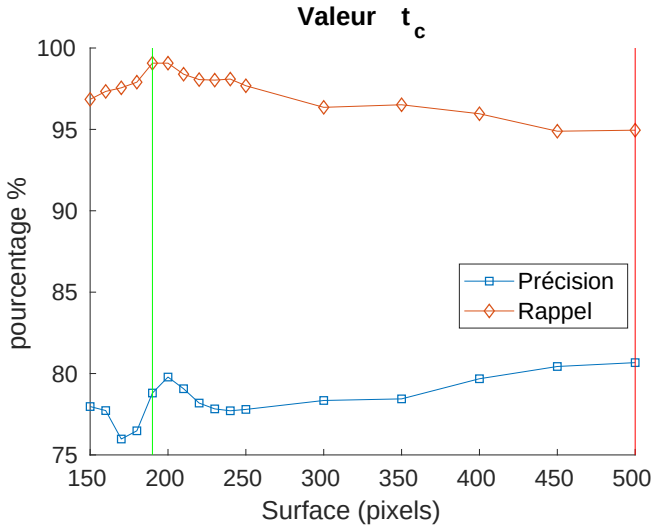


FIGURE 3.35 – Représentation de résultats de précision et rappel en fonction des paramètres t_c.

Dans la figure 3.36, on évalue l'influence de la taille de la surface de la tête dans la segmentation par graphe de niveaux $[H_s^{\min}, H_s^{\max}]$. On évalue la taille minimale de la surface d'une tête entre les valeurs de 1000 et 12000 pixels et la taille maximale de la surface d'une région tête entre 2000 et 14000 pixels. La meilleure valeur pour $H_s^{\min}$ est 1600 pixels, valeur que l'on peut interpréter comme la surface de la tête la plus petite pour des personnes qui marchent d'une manière très rapprochée. La meilleure valeur pour $H_s^{\max}$ est de 11000 pixels, valeur de la tête la plus grande du jeu de données de test. On trouve que l'influence de ces paramètres est minimale parce que $H_s^{\min}$ évite légèrement de détecter les faux positifs et dans le cas de $H_s^{\max}$ pour toutes les valeurs plus grandes que la valeur choisie, la précision ou le rappel ne s'améliorent pas.

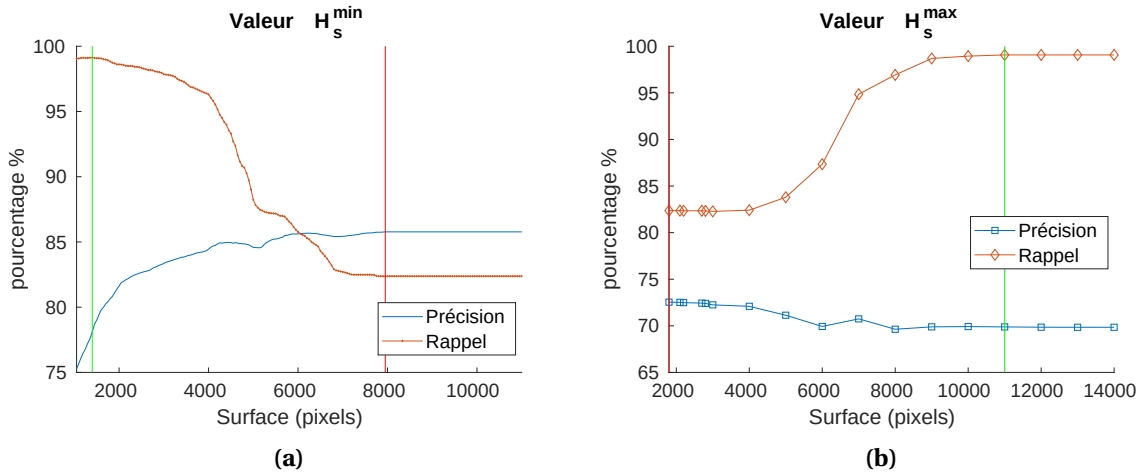


FIGURE 3.36 – Représentation de résultats de précision et rappel en fonction des paramètres $H_s^{\min}$ et $H_s^{\max}$.

Analyse des résultats

Suite à l'évaluation des critères définis, on a caractérisé les types de ***faux négatifs*** de la détection de l'algorithme. De manière générale, nous avons trouvé des erreurs de segmentation à cause des occultations et de la coupe précoce de la tête.

- L'*occultation* se produit dans les cas décrits dans le scénario b (Fig. 3.32a).

- La *coupe précoce* de la tête se produit quand la hauteur de la tête d'une deuxième personne est très proche de la valeur de tc, la surface résultant de cette tête dans le niveau le plus haut est trop petite et l'algorithme ignore cette détection.

De notre point de vue, il est difficile de surmonter ces problèmes, qui ne sont par ailleurs pas très fréquents. En analysant chaque cas, les occultations entre les personnes sont normalement temporelles, c'est-à-dire qu'on peut chercher à les résoudre dans un algorithme de suivi plus sophistiqué qui garde en mémoire les régions marquées comme ambiguës pour générer des trajectoires indépendantes à la fin du phénomène.

Dans le cas de la *coupe précoce*, la hauteur et la taille de la tête des personnes varient énormément, par ailleurs il n'existe pas de valeur fixe qui serve dans tous les cas. De plus, le nombre d'erreurs à cause de cet effet est non-significatif. On pourrait explorer la dynamique de ce paramètre pour trouver des astuces permettant de déterminer la valeur correcte selon le cas, en évitant des solutions itératives, si on trouve des cas d'usage où ce problème a un effet plus significatif sur la détection.

En s'intéressant aux ***faux positifs***, on a trouvé que la majorité est causée par des effets du bord où notre algorithme détecte une personne, celle-ci n'étant toutefois pas annotée dans la vérité-terrain puisque la tête n'est pas suffisamment visible (plus de 40 %).

En relation à d'autres types de faux positifs, on a l'avantage d'utiliser des caméras de profondeur disposant d'une bien meilleure fiabilité de détection des pixels du premier plan. Le défi reste de bien caractériser et classifier ces pixels pour trouver les personnes et discriminer des objets qui apparaissent dans la scène.

3.3.2 Évaluation de l'algorithme de suivi de personnes en mouvement

Dans cette section, on évalue la ré-identification sur la séquence vidéo sur le jeu de données annoté. Nous utilisons les valeurs des paramètres de la section précédente qui ont obtenu les meilleurs résultats de détection. On mesure la performance de suivi en utilisant le taux de ré-identification de personnes le long de la séquence vidéo[1].

Méthode d'évaluation

Pour déterminer le *taux de ré-identification de personnes,* on définit une ré-identification comme la liaison de la détection d'une personne entre deux images consécutives qui est un vrai positif. Chaque fois que l'algorithme rate une ré-identification, c'est un faux positif.

Le taux de ré-identification de personnes est équivalent mathématiquement au ratio de faux négatifs (MR) et il est défini comme le rapport entre les fausses détections et les détections correctement trouvées :

$$\mathrm{MR} = \frac{\mathrm{FN}}{\mathrm{VP}} \tag{3.10}$$

On utilise la même méthode de la section précédente sur le même jeu de données, en évaluant le paramètre d_{mc}. Ce paramètre nous impose la distance maximale de recherche de correspondance dans l'étape de suivi.

Résultats de l'algorithme de ré-identification

On présente le ratio de faux négatifs comme le critère de performance que l'on utilise pour évaluer notre algorithme de suivi.

[1] Ces résultats sont évalues dans un PC sans prendre en compte la vitesse (FPS) ni les caractéristiques de la machine pour nous concentrer sur les mesures décrites.

On évalue l'influence de la distance maximale d'association des détections d_{mc}. On évalue la distance maximale entre les valeurs de 50 et 200 pixels. On trouve que le meilleur résultat est d_{mc}=110 avec MR égal à 0,0235, c'est-à-dire une réussite de suivi de 97,65 % pour le jeu de données de test. Ces résultats sont représentés dans la figure 3.37.

La figure 3.37 nous montre en bleu le ratio de faux négatifs et en rouge une ligne horizontale qui représente la valeur où ce ratio est le plus bas (valeur optimale). L'axe X représente la valeur du paramètre évalué exprimée en millimètres et l'axe Y représente la valeur résultant du ratio.

On observe que ce paramètre introduit beaucoup d'erreurs avec de petites valeurs et qu'il restreint l'espace de déplacement des personnes à une distance très proche de sa position précédente. Par contre, on observe la faible influence de d_{mc} à partir de 110 pixels en raison de l'absence de correspondance entre trajectoires et détections.

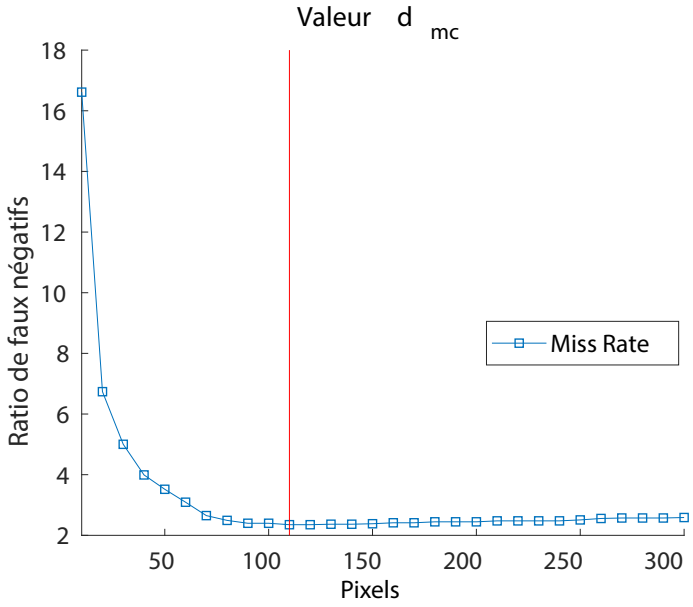


FIGURE 3.37 – Représentation de résultats de précision et rappel en fonction du paramètre d_{mc}.

Analyse des résultats

On peut conclure que ces résultats sont suffisants pour les besoins industriels pour le comptage de personnes en magasin, en permettant d'évaluer la performance en vitesse des algorithmes sur différents systèmes embarqués.

3.3.3 Évaluation des performances

Dans cette section, on mesure la performance du système en termes d'images traitées par seconde, c'est-à-dire la vitesse de notre système pour traiter des images 3D en détectant et en suivant des personnes en mouvement. Nous mesurons la performance en FPS livrés en temps-réel.

Méthode d'évaluation

Nous avons testé notre système logiciel sur quatre architectures embarquées différentes :

- L'architecture RASPI 3 (RPI3), basée sur le SoC BCM2837 et composée de :
  - Quatre processeurs ARM Cortex A53 disposant d'une architecture 64-bits cadencée à une fréquence de 1,2 GHz.
  - Une mémoire RAM 1GB LPDDR2.

  - Une consommation qui varie entre 2W et 3,7W.

- L'architecture Up-Board (UPB), basée sur le SoC de l'Intel Atom™ x5-Z8350 et composée de :

  - Deux processeurs Armont double cœur disposant d'une architecture 64-bits cadencée à une fréquence comprise entre 1,44 et 1,92 GHz.
  - Une mémoire RAM de 2G DDR3L-1600.
  - Une consommation de 2W.

- L'architecture ASUS Tinker-board (TKER), basée sur le Soc Rockchip RK3288, composée de :

  - Quatre processeurs ARM Cortex-A17 avec 4 cœurs disposant d'une architecture 32-bits pouvant aller jusqu'à 1,8 GHz.
  - Une mémoire RAM de 2Go LPDDR3 double canal.
  - Une consommation qui varie entre 2W et 5W.

- L'architecture Odroid-XU4 (ODR), à base du processeur Samsung Exynos5422, composée de :

  - Un processeur ARM Cortex™-A15 avec 4 cœurs disposant d'une architecture 32-bits, cadencée à une fréquence de 2 GHz.
  - Un processeur ARM Cortex™-A7 avec 4 cœurs disposant d'une architecture 32-bits, cadencée à une fréquence de 1,4 GHz.
  - Une mémoire RAM 2 LPDDR3/DDR3 à 933MHz.

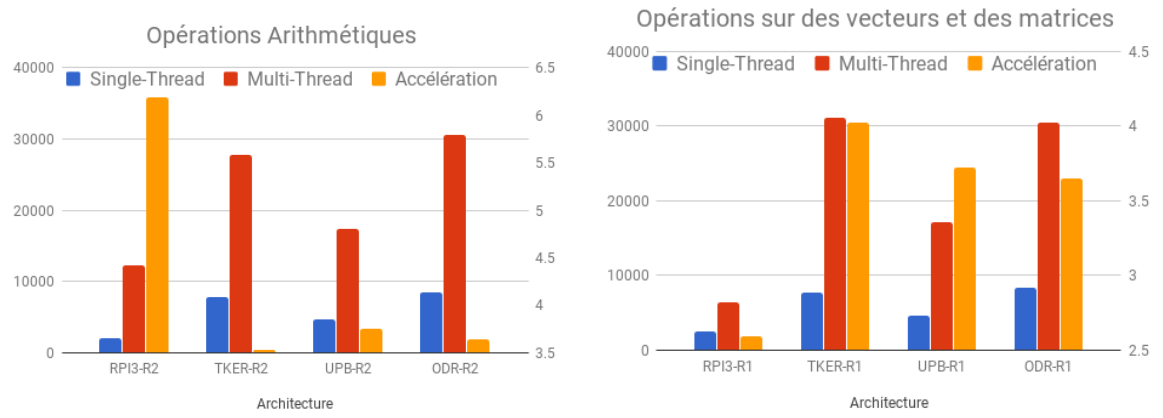
Nous avons utilisé l'outil de benchmark CoreMark développé par EEMB [web17c] pour évaluer la puissance des architectures décrites plus haut et nous permettre ainsi de comparer les performances.

Résultats d'évaluation des performances

On évalue la performance de différents systèmes embarqués en mesurant le score donné par le logiciel de benchmark CoreMark. Ce score reflète le nombre d'itérations exécutées par seconde. On peut augmenter le nombre de *threads* parallèles pour exécuter le même nombre d'itérations et avoir ainsi une première tendance du potentiel d'accélération possible en utilisant le parallélisme. Selon la description fournie par EEMB, le benchmark exécute deux routines. La première exécute des opérations arithmétiques avec différents types de données et la deuxième réalise des opérations sur des vecteurs et des matrices. Les scores de l'évaluation pour chacune des architectures sont présentés dans la figure 3.38, où la troisième colonne représente le potentiel d'accélération parallèle pour l'architecture en question.

TABLEAU 3.2 – Scores et accélérations évalués avec CoreMark

Architecture	Programme	Multi-thread	Accélération
RPI3-R1	2 443,91	6 336,93	2,59
RPI3-R2	1 979,98	12 242,15	6,18
TKER-R1	7 726,93	31 125,98	4,03
TKER-R2	7 856,69	27 739,25	3,53
UPB-R1	4 615,74	17 190,99	3,72
UPB-R2	4 622,03	17 329,52	3,75
ODR-R1	8 357,88	30 522,70	3,65
ODR-R2	8 396,83	30 574,03	3,64



(a) **(b)**

FIGURE 3.38 – Comparaison des itérations par seconde dans les différentes architectures. a) Résultat des opérations arithmétiques. b) Résultat des opérations sur des vecteurs et des matrices. L'accélération est représentée en pourcentage.

Le meilleur potentiel d'accélération intrinsèque sera sur la première architecture (RPI3), cependant le nombre d'itérations est plus important sur ODR. L'outil de benchmark ne permet actuellement pas de gérer des processeurs différents sur la même architecture, donc les résultats obtenus pour l'architecture ODR (qui possède deux processeurs d'architecture différente) ne sont pas forcément pertinents.

Si nous prenons en compte les contraintes d'industrialisation (coût, approvisionnement, etc.) combinée avec les performances intrinsèques, c'est l'architecture TKER qui ressort comme étant la plus pertinente pour notre système.

Notre solution logicielle développée et optimisée en C++ a été compilée pour les différentes plateformes et exécutée en traitant la séquence décrite précédemment (section 3.1.2). Les résultats sont présentés dans le tableau 3.3.

TABLEAU 3.3 – Performance de la solution de comptage sur les différentes architectures embarquées

Architecture	RPI3	TKER	UPB	ODR
FPS	15	18	20	20

En conclusion, la parallélisation et les bonnes pratiques dans le développement des applications en systèmes embarqués nous permettent de respecter les contraintes de temps pour effectuer du comptage de personnes en utilisant un capteur 3D en position zénithale en temps réel. Même si toutes les architectures ont dépassé le seuil FPS minimum de performance pour fonctionner en temps-réel, on a choisi l'architecture TKER de ASUS car elle permet également de répondre à nos contraintes industrielles, notamment de prix.

3.4 Conclusions

Nous avons présenté une solution originale pour concevoir une caméra intelligente autonome en proposant une analyse spécifique des contraintes matérielles et logicielles. De plus, nous avons décrit notre chaîne de traitement de suivi de personnes en réduisant le processus à 3 blocs accompagnés d'un bloc préliminaire qui permet à la caméra de se paramétrer par rapport à son environnement. Ensuite, nous avons présenté une analyse de performances de notre système logiciel sous différentes architectures embarquées. Nous avons également évalué les résultats de l'algorithme de suivi des personnes, appliqué à notre jeu de données des images de profondeur d'une caméra en position zénithale.

Nous avons surmonté les difficultés de conception d'un système pour le suivi de personnes en mouvement de la manière suivante :

- Bruit du détecteur : Comme on a vu dans le chapitre précédent, la camera ASUS a un rapport signal / bruit le plus faible par rapport aux autres capteurs. Cependant, on doit gérer le bruit produit par la caméra dans la carte de profondeur. On diminue la possible influence du bruit dans notre résultat en considérant les régions par groupes de pixels (segmentation par niveaux), à l'intérieur de plages de distance. On évite, en même temps, de faire des calculs lourds comme le moyennage ou le remplissage de régions.

- Occultation : De la même manière, la position zénithale de la caméra réduit énormément les occultations entre personnes. Cependant, il nous reste à résoudre quelques cas où il y a des occultations entre personnes. Nous les gérons de deux façons différentes. D'abord, tous les cas où une partie du corps d'une personne (sauf la tête) est occultée, la segmentation par niveaux est capable d'identifier la personne en question. Ensuite, si la tête n'est pas visible, mais la personne a été détectée dans des images précédentes, nous sommes capables de prédire sa position éventuelle et attribuer sa correspondance à des régions ambiguës positionnées dans la zone de prédiction (suivi).

- Variations environnementales : le capteur 3D traite en grande partie cette difficulté. Cependant, nous avons ajouté un filtrage des hauteurs adapté à l'environnement d'installation pour éliminer tous les objets dynamiques dans la scène qui peuvent influencer négativement le comptage de personnes.

- Modélisation de l'arrière-plan : nous avons créé un module d'auto-positionnement de la caméra qui nous permet de modéliser l'arrière-plan une seule fois et d'obtenir rapidement les pixels d'intérêt pour suivre des personnes en mouvement.

- Séparation des personnes : nous avons conçu un bloc dédié à la segmentation des personnes en cas d'occultation ou de proximité. De plus, dans les cas extrêmement difficiles rencontrés au stade de notre module de segmentation, nous avons introduit une méthode pour lever l'ambiguïté de ces séparations non résolues.

- Similitude des gens : nous avons proposé une nouvelle méthode pour caractériser les personnes, les HFDs (Human Features Descriptors), qui nous permettent d'identifier les personnes à l'intérieur du système. Ces HFDs combinées avec le positionnement de la caméra évitent la confusion au moment d'identifier les personnes dans les cas que nous avons évalués.

- Variation d'échelle : la variation d'échelle est résolue avec l'utilisation de mesures dans l'espace 3D, d'une part avec les données de profondeur fournies par le capteur 3D, d'autre part, avec les caractéristiques mesurées dans les HFDs faites dans l'espace 3D.

- Déformation du modèle du sujet : grâce à la position zénithale, les déformations possibles de notre modèle sont dues en grande partie à la position de la tête par rapport au dos et (dans certain cas) la position des bras. Cette difficulté est traitée dans la construction du graphe par niveaux en nous permettant d'éliminer les parties différentes à la tête et les épaules. Ensuite, la déformation entre ces parties est prise en compte dans la construction du HFD. De plus, la mise à jour du HFD est faite de manière à ce que les changements brusques dans le modèle du sujet soient amortis par un lissage dynamique.

- Suivi en déplacement rapide : les résultats montrent que notre solution est capable de fonctionner à une vitesse de plus des 15 fps en assurant la performance requise pour le suivi de personnes dans les espaces publics.

Les objectifs de conception du système :

- Performance : nous avons développé des algorithmes légers et optimisés pour les architectures embarquées et dans des conditions de basse consommation électrique (moins de 20W). Nous avons utilisé des technologies de parallélisation pour optimiser l'utilisation de ressources limitées sur différents systèmes embarqués. Nous avons développé également un protocole de communication en envoyant des messages légers et de haut niveau

sémantique (HFD) pour réduire la consommation de la bande passante du réseau (HFD à la place d'images) et en même temps en diminuant l'utilisation du processeur à cette fin. Finalement, on a décrit la distribution du calcul des données à travers les différents niveaux (capteur, caméra intelligente et serveur).

- Configuration et déploiement facile : nous avons développé une couche de configuration et d'installation permettant aux utilisateurs de pouvoir changer facilement les paramètres de la caméra. De plus, le module d'auto-positionnement réduit les tâches des configurations du système. L'utilisation de la technologie POE pour la transmission de données et alimentation électrique ont contribué à la conception d'un système facilement évolutif.
- Faible coût : on a évalué différentes composantes pour choisir finalement celles qui ont la consommation électrique la plus basse et le plus faible coût sous les contraintes matérielles et logicielles imposées par le cahier des charges, tout en gardant nos exigences de performance.

Les difficultés rencontrées pour construire une caméra intelligente :

- La conception *matérielle informatique :* nous avons proposé une solution de caméra intelligente en intégrant un capteur 3D, une carte Raspberry PI, un module de communication et une alimentation POE. Comme résultat, nous avons construit un système de suivi de personnes à faible coût.
- La conception du *logiciel du système :* nous avons développé une nouvelle solution de suivi de personnes grâce à des technologies multiplateforme capables de fonctionner dans divers systèmes d'exploitation. Nous avons conçu un système de briques interchangeables pour traiter différents types d'images et échanger différents algorithmes à l'intérieur de la même solution avec une faible complexité. En parallèle, nous avons développé un système de communication basé sur l'échange des messages légers sur un réseau Ethernet et Wi-Fi. Finalement, nous avons développé un module de configuration Web facile à utiliser pour les utilisateurs finaux, qui permet de configurer la caméra et la logique du métier.
- La *sécurité et la protection de la vie privée :* le développement du système a pris en compte les contraintes à respecter pour assurer la sécurité des données et la protection de la vie privée. Ainsi, les données de profondeur que notre système utilise font appel aux formes géométriques de la scène mais elles ne contiennent pas de détails physiques comme la couleur de peau ou les traits du visage, qui pourraient permettre d'identifier les personnes suivies. En plus, les données recueillies par notre caméra intelligente ont un tel niveau sémantique que leur vol éventuel ne permet pas de réaliser une intrusion dans la vie privée des personnes en question.
- L'*adaptation et l'autonomie :* le bloc d'auto-positionnement et le module web nous permettent d'assurer l'adaptation et l'autonomie souhaitées pour notre système. De plus, l'interface web nous permet de réaliser les opérations de maintenance à distance.

On a mappé le traitement sur la puissance de calcul disponible en associant différentes étapes de la chaîne de suivi des personnes. Celle-ci nous permet la mise en place d'algorithmes simples et efficaces, d'extraire en un seul passage sur l'image plusieurs informations en économisant du temps de calcul et les autres ressources limitées de notre plateforme. De même, on a conditionné les comportements des algorithmes en cas d'ambiguïté dans le suivi, en évitant de faire des calculs supplémentaires. Finalement, le bloc qui assure le comptage des personnes offre aux utilisateurs trois choix possibles de configuration selon le besoin. En résumé, ce processus nous permet d'extraire d'une séquence des images de profondeur, un historique des positions et le vecteur de caractéristiques des personnes qui ont traversé la scène.

Les difficultés qui nous restent à surmonter sont la *collaboration* entre les caméras intelligentes autonomes et le suivi de personnes dans de grands espaces publics, sujet qui sera traité dans les chapitres suivants.

Deuxième partie

Etude comportamentale sur de grands espaces

Chapitre 4

Étude des personnes en mouvement dans des grands espaces

Sommaire

La notion de *grand* espace fait référence à une zone de grande surface avec différents types d'infrastructures comme les parkings publics ou les centres commerciaux. Un grand espace peut être composé de plusieurs zones ou pièces (Fig. 4.1a). Pour cette raison, le suivi des personnes en mouvement dans des ***grands espaces*** nécessite l'élargissement du champ de vision de notre système. Ainsi, la configuration d'un grand espace nous pose deux problèmes :

- L'***extension d'une zone*** : parce que le champ de vision de la caméra ne couvre pas entièrement l'espace désiré d'évaluation, en raison de sa grande taille (Fig. 4.1c). D'une part, on rappelle que nous avons choisi la position zénithale de la caméra, au détriment de son champ de vision, pour bénéficier d'une meilleure qualité d'information. De ce fait, les capteurs de vision 3D ont deux types de limitations : la portée en profondeur de la caméra (qui limite la distance entre la caméra et les personnes observées) et les angles d'ouverture du champ de vision de la caméra (normalement décrits en degrés horizontaux, verticaux ou diagonaux).
- L'***infrastructure d'une zone*** : parce que la couverture de la caméra est limitée par les éléments statiques qui appartiennent à la scène observée (Fig. 4.1d). Dans ce cas, la portée et le FOV de la caméra peuvent être suffisants, même si les infrastructures qui composent l'espace comme les murs ou les portes limitent le champ de vision de la caméra.

En raison de ces limitations, nous avons été contraints de passer à une approche multi-caméra, avec un nombre de caméras suffisant pour couvrir de grands champs de vision et éviter des obstructions (Fig. 4.1b). Nous obtenons alors une couverture plus large mais cela implique de surpasser des **verrous technologiques** comme le positionnement automatique des caméras dans un repère global, l'identification unique d'une personne qui se déplace dans cet espace multi-caméra, la gestion des personnes à l'intérieur des régions des champs de vision qui se chevauchent et la gestion de trajectoires globales [BKIM13, FBBS06, TJS10].

Nous avons identifié quatre *tâches principales de traitement* de données pour le suivi de personnes dans des grands espaces en utilisant des capteurs 3D : l'acquisition des données (carte de profondeur), le suivi de personnes (chaîne de traitement), la couche d'application (bloc d'application) et l'analyse des données globales (traitée dans le chapitre suivant). La difficulté d'utiliser une approche multi-caméra nécessitera de lever les verrous technologiques décrits et en même temps d'effectuer ces tâches principales sur les données issues de toutes les caméras de l'espace à observer.

Dans ce chapitre, on présente les choix technologiques et algorithmiques pour d'une part surmonter les verrous technologiques et industriels dans la conception d'un réseau de caméras, et d'autre part, pour augmenter la capacité de notre système à suivre des personnes dans de grands espaces.

Dans la figure 4.1, la première image (4.1a) représente un grand espace qui est composé de plusieurs pièces dont une de grande surface. La deuxième image (4.1b) illustre l'espace couvert par plusieurs caméras. Dans cette figure, les points jaunes représentent la position des caméras et les rectangles bleu clair les champs de vision des caméras. On observe les régions en bleu plus foncé représentant les régions de chevauchement entre les champs de vision des caméras. Dans l'image (4.1c), le même espace est couvert partiellement par une seule caméra. La dernière image (4.1d) présente l'analyse de la couverture d'une caméra sur la pièce la plus grande du bâtiment. La région rouge représente l'espace non visible à cause du mur de la pièce. La région verte représente une surface de la pièce non couverte par la caméra.

FIGURE 4.1 – Images d'un bâtiment en considérant les différentes problématiques d'un grand espace.

4.1 État de l'art

On trouve dans la littérature deux approches pour la création de réseaux de caméras intelligentes : l'approche centralisée et l'approche distribuée [RWS+08, Wol14]. On identifie trois axes principaux liés à ces approches qui sont : l'architecture du réseau de caméras intelligentes, la distribution du calcul dans le réseau et les niveaux de description de l'information. Ces axes sont interdépendants et coexistent dans les différents éléments du système.

Dans l'axe de l'*architecture du réseau de caméras intelligentes*, on trouve les éléments qui composent le système comme les entités du système (capteurs, systèmes embarqués ou serveurs), la technologie de communication (Ethernet, Wifi, Bluetooth, etc.) et l'infrastructure électrique (type d'alimentation, comme par exemple autonome ou POE : Power Over Ethernet). La technologie de communication et l'infrastructure électrique ont déjà été définies dans le chapitre 3. On se focalisera sur la définition des entités composant le système, avec les capacités de communication et de calcul associées. Pour chaque entité, on définit ses fonctions et l'interaction avec les autres entités.

Dans l'axe de la *distribution de calcul dans le réseau*, on trouve où (dans quelle entité) et comment sont traitées les tâches principales de suivi des personnes.

L'axe des niveaux de description est dépendant des autres axes (réseau et distribution de calcul). Celui-ci rend compte de la complexité du type des données (par exemple d'images ou de vecteurs de caractéristiques) qui se traduit par un ou plusieurs niveaux de description. Par exemple, le niveau de description le plus bas fait référence à des données brutes comme une image couleur ou une carte de profondeur. Un niveau de description plus élevé fait référence par exemple aux descripteurs de personnes HFD introduits dans la section 3.2.1.

La section suivante décrit, pour chaque approche, les caractéristiques des différents axes mis en évidence, en évaluant ses capacités d'évolutivité, la facilité d'installation, l'adaptation et l'autonomie du système.

4.1.1 Architecture du réseau centralisé

Dans cette première approche, on utilise une architecture reposant sur un réseau centralisé de caméras. Elle est composée d'une entité centralisée (désignée par l'hôte ou le serveur) dotée d'une grande puissance de calcul et d'un réseau de capteurs 2D [BE06, FBLF08] et 3D [BKIM13, GBMH15, SBR14, SBR12]. D'un côté, le capteur a pour fonction de capturer des images (acquisition) de la scène pour les envoyer à l'entité centrale. De l'autre, l'entité centrale a pour fonction d'exécuter toutes les *tâches principales de traitement* sur les images qu'elle reçoit de toutes les caméras du réseau. Dans l'alternative de capteurs 3D, ces derniers assurent le calcul des images de profondeur.

Les architectures centralisées échangent des données brutes car les capteurs fournissent des images à l'entité centrale. Celle-ci est en charge du reste du traitement, du stockage pour l'analyse et la visualisation, en évitant la retransmission des données. Cependant, les analyses des données globales génèrent une information de haut niveau descriptif représentant le comportement des personnes détectées et suivies (le travail réalisé par Zanlungo et al., en 2017 en est un exemple [ZYB+17]).

4.1.2 Architecture du réseau distribué

L'architecture du réseau distribuée est un réseau d'entités connectées avec des capacités de traitement propres. Dans la littérature, on distingue deux types d'entités : le serveur et les nœuds [KCCVZ15, KHM+00, VSC+08, VKA15]. A la différence de l'entité centrale de l'architecture précédente, le serveur exécute moins des *tâches principales de traitement*. Son rôle est de centraliser l'information et de prendre les décisions de l'application finale et, dans certains cas, de faire des analyses sur les données globales. Dans cette architecture, un nœud est une entité associée à une caméra qui a pour fonction de détecter les personnes et d'informer de leurs positions (et autres caractéristiques observées et transmises au serveur).

On différencie dans certains cas un nœud d'une caméra intelligente parce que le nœud n'assume pas toute la chaîne de traitement (présenté dans le chapitre 3) pour le suivi de personnes. Par exemple, dans [KHM+00] les nœuds font tous les blocs de la chaîne de traitement sauf le bloc du suivi. Ces auteurs ont construit un système pour identifier les personnes à l'intérieur d'une salle multimédia et pour connaître leurs modes d'utilisation préférés de cette salle. Le serveur est identifié par les auteurs comme le « traqueur ». Il a pour rôle de suivre les personnes dans un repère global du système, de contrôler la salle multimédia, et, si besoin, il initie une demande d'informations supplémentaires auprès des nœuds en cas d'ambiguïté pour prendre des décisions. Dans le travail de Virgona et al. en 2015, les nœuds sont des caméras intelligentes qui suivent les personnes localement et les décisions du système sont toujours prises par une entité centrale.

Dans une architecture distribuée, on trouve différentes répartitions des traitements et des décisions sur les éléments du système. Le calcul de l'image 3D est réalisé par les capteurs de profondeur, la détection des personnes dans le nœud et le suivi, et la visualisation dans le serveur [FBBS06, KHM+00]. On trouve des travaux où les nœuds sont responsables de la majorité des traitements et des décisions mais en utilisant des caméras monoscopiques [DSE+13, TS08].

Dans les architectures distribuées, plusieurs types de données sont échangés à différents niveaux de description. Par exemple dans [KHM+00], on trouve le premier échange de données de la même manière que dans l'architecture de réseau centralisée. Ensuite, le nœud communique les positions locales et la forme des personnes détectées au serveur. Ce dernier effectue le suivi et la prise de décisions de l'application. Dans [FBBS06], le deuxième niveau regroupe l'échange des positions globales et la texture des personnes suivies utilisées par le serveur pour la couche applicative.

4.1.3 Remarques

L'inconvénient du premier type d'architecture est son manque évolutivité, pour ajouter des caméras supplémentaires par exemple. Pour surmonter cette difficulté, deux solutions sont possibles : on surdimensionne, au départ, la capacité de calcul de la machine centrale pour permettre un nombre fixe de caméras supplémentaires ou bien on augmente cette capacité de calcul au moment où l'on a besoin d'ajouter les caméras. Dans les deux cas, on engage des dépenses supplémentaires, entravant la facilité d'installation, l'adaptation, l'autonomie du système et sans évoquer le goulot d'étranglement que va constituer le réseau de communication dont la bande passante n'est pas infinie.

L'inconvénient du deuxième type d'architecture concerne la responsabilité de la prise de décisions concentrée sur une seule machine. Velipasalar et al. soulignent dans leurs travaux [VSC+08] qu'il y a le risque lié au fait que la totalité du système soit fortement liée au bon fonctionnement d'un seul serveur central. Donc, on doit chercher à diminuer ces responsabilités.

Plus concrètement, on a identifié la difficulté d'utiliser une approche multi-caméra pour le suivi de personnes dans des grands espaces. Pour ce faire, les traitements principaux à réaliser sont les suivants :

- l'acquisition des données,
- le suivi de personnes,
- la couche d'application,
- et l'analyse des données globales.

Et cela implique de surmonter des verrous technologiques liés à la gestion de données globales :

- le positionnement des caméras dans un repère global,
- la gestion et l'identification de personnes suivies par le système dans le temps et dans l'espace.
- la gestion de trajectoires globales.

Par conséquent, nous devons définir une architecture capable de dépasser ces difficultés et dont la description contient :

- les entités et ses fonctionnalités,
- leurs responsabilités sur les tâches principales,
- et les niveaux de description des données échangées.

A notre connaissance, il n'y a pas de travaux utilisant des caméras intelligentes 3D connectées en réseau, dans lesquels le nœud réalise le suivi, la prise de décisions de l'application et une partie des analyses globales pendant que le serveur reçoit seulement des informations pour la visualisation ou l'analyse globale de données, solution que l'on a en partie déjà proposée et que l'on élargit dans ce chapitre et le chapitre suivant.

4.2 Architecture du réseau de caméras intelligentes

Dans cette section, nous présentons notre solution en décrivant les détails de l'architecture du réseau de caméras intelligentes, de la distribution de calcul et des niveaux de description du système. Nous présentons également la gestion de données globales en explicitant les méthodes de positionnement des caméras intelligentes dans un repère global et du suivi des personnes dans une espace multi-caméra. Ce suivi multi-caméra nous permettra de gérer et d'identifier des personnes suivies localement dans le repère global.

4.2.1 Architecture du réseau de caméras intelligentes distribué

Nous proposons une architecture de réseau distribué composée de nœuds interconnectés et d'un serveur. Chaque nœud est composé d'une caméra intelligente autonome (décrite dans la chapitre 3), capable d'effectuer localement l'acquisition, le suivi de personnes, le bloc d'application et une partie de l'analyse des données globales. Le serveur a en charge de gérer les informations globales, le stockage de données et l'accès de l'utilisateur final aux résultats d'analyse.

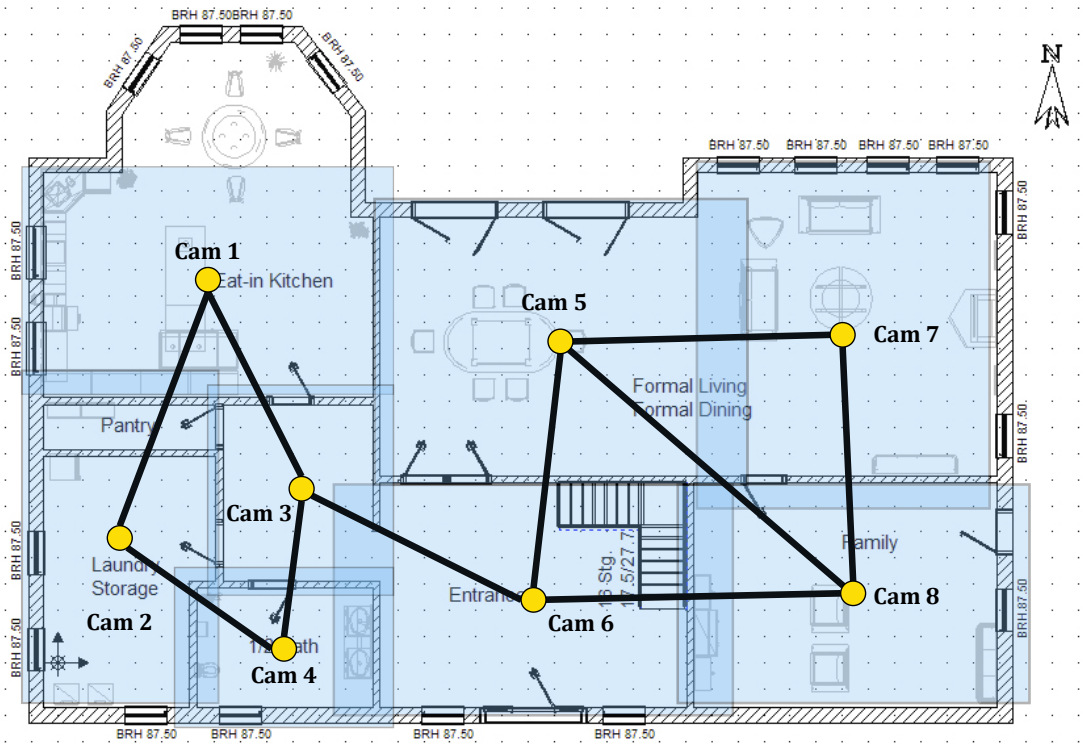


FIGURE 4.2 – Exemple d'un grand espace couvert par 8 caméras.

Dans la figure 4.2, chaque rectangle bleu représente le champ de vision d'une caméra indiqué par un cercle jaune, les lignes noires correspondant, quant à elles, la connexion de couverture. Globalement, on installe une caméra par pièce, sauf si cette dernière est trop grande pour la portée de la caméra.

La différence entre notre architecture distribuée et les solutions existantes est que nous plaçons au niveau des caméras intelligentes la majorité des tâches principales de traitement, les rendant ainsi autonomes. Cependant, pour pouvoir gérer correctement les données globales, nous ajoutons un calculateur considéré comme serveur à couplage faible. Ce dernier sert à centraliser et assurer certaines tâches qui doivent se faire de manière unique, comme l'étiquetage des personnes détectées et suivies sur l'ensemble du système, pour éviter les ambiguïtés et donc des doublons. De plus, le *serveur faible* sert au stockage et à la visualisation des informations globales.

Ce type d'architecture distribuée, dans laquelle les nœuds portent l'intelligence, permet une expansion quasi-illimitée avec l'avantage que chaque nouveau nœud du réseau apporte sa propre puissance de calcul. De plus, si le serveur tombe en panne, cette fonctionnalité de gestion des données globales peut être prise en charge par un autre nœud inoccupé (par exemple dans une zone de faible circulation de piétons). Cette approche nous semble la plus pertinente à mettre en œuvre dans notre système du fait que si l'un des nœuds ou le serveur s'arrêtent de fonctionner, la globalité du système reste fonctionnelle, dans un mode dégradé.

Pour créer un système composé de plusieurs entités, on doit fournir pour chaque nœud des capacités à se positionner dans son environnement et s'identifier par rapport aux autres nœuds. Une fois la caméra positionnée dans son environnement, elle doit être capable d'identifier de manière unique les personnes suivies. Pour ce faire, on définit un repère global nommé ***S**ystème de **C**oordonnées **G**lobales* (SCG) et un ***S**ystème d'**I**dentification **G**lobale* (SIG) des personnes.

Le système de coordonnées globales permet de créer un seul repère spatial pour toutes les caméras appartenant au même système. Le processus de création du SCG commence par l'installation d'une caméra et son positionnement dans l'environnement. Ce premier calcul de coordonnées de la première caméra devient la base de notre repère global. Ensuite, chaque nouvelle caméra ajoutée au système doit se calibrer de manière extrinsèque, c'est-à-dire, estimer sa position par rapport à cette première caméra. Les détails du processus de calibration multi-caméra sont expliqués dans le chapitre 4.3.1. Le résultat de ce processus est nommé *localisation globale*.

Enfin, le système d'identification globale utilise une identification unique attribuée à chaque personne suivie dans le système nommée *étiquette globale unique* EGU. Ce système est responsable de la génération et de la gestion des EGUs. Chaque nouvelle personne qui entre dans le champ de vision de notre système pour la première fois sera identifiée par un nœud, qui lui attribuera une étiquette locale pour gérer le suivi localement et demandera une nouvelle étiquette unique au SIG. Ensuite, quand une personne passe d'une zone à une autre, les nœuds échangent les EGUs. De cette manière, l'EGU permet au système d'unifier les trajectoires locales d'une personne qui traverse plusieurs zones surveillées le long de l'espace global pour générer une seule trajectoire. Ce processus est détaillé dans la section 4.3.2.

Application au comptage des personnes

Dans notre étude de comptage des personnes, on définit deux entités : l'agent de comptage et le superviseur.

L'*agent de comptage* est responsable du suivi, de l'identification et du comptage des personnes localement. De plus, l'agent doit convertir les positions locales des personnes au SCG. Par rapport au serveur, il doit envoyer les positions globales, le comptage et demander de nouvelles EGU pour chaque nouvelle personne dans le système. Il gère également l'échange d'EGU de chaque personne suivie avec ses nœuds voisins immédiats.

D'autre part, le *superviseur* est responsable du SIG, du traitement des informations globales, du rassemblement de trajectoires, du comptage et de la génération des analyses de comportement. Cette fonctionnalité peut être effectuée par un serveur local ou un service Cloud.

4.2.2 Niveaux de description

Dans cette section, on analysera les données produites par notre système pour proposer une hiérarchie de l'information extraite en différents niveaux d'abstraction, selon sa complexité et sa fonction, nommés niveaux de description.

Dans notre système, nous identifions quatre niveaux de description (voir Fig. 4.3). Cela commence par un grand volume brut de données, pour se terminer par des informations d'un niveau d'abstraction élevé et de taille de données réduite. Dans notre système, chaque niveau utilise l'information du niveau précédent pour extraire des données plus riches et plus compactes,

en créant de nouvelles informations prêtes à être exploitées par le niveau de description d'information supérieur. Le premier niveau est composé des données brutes acquises par la caméra, c'est-à-dire les images de profondeur. Ensuite, le deuxième niveau est composé des HFDs produits par notre *caméra intelligente* de manière locale et sa fonction est de décrire les personnes et leurs mouvements localement. Le troisième niveau est composé par l'identifiant unique EGU, les positions locales transformées au SGC et le comptage (ou données de la couche d'application). Cette transformation de coordonnées et la gestion de l'EGU par personne sont réalisées par la caméra intelligente pour situer l'information extraite dans un contexte global, en assurant la création des trajectoires complètes des personnes et le comptage total. Le dernier niveau est constitué par l'information des trajectoires récoltées (décrites dans le chapitre suivant). Nous distinguons les trois derniers niveaux de description du premier parce qu'ils fournissent des informations abstraites en lien avec la notion de personne, leur trajectoire et leur comportement.

Nous définissons l'information descriptive de chaque niveau comme une information significative, autosuffisante et concise. Il est important de fournir des informations plus riches et plus faciles à interpréter par les étapes suivantes du processus de suivi multi-caméra. L'autosuffisance indique que l'information contient toutes les données requises au moment du traitement, évitant de faire des appels aux étapes précédentes de traitement. Par information concise, nous nous référons non seulement à des informations claires et complètes, mais aussi plus compactes que les prédécesseurs en termes de taille de données. Les structures de données sont plus élaborées, par exemple une personne est représentée par deux ellipses au lieu d'une région segmentée composée de plusieurs pixels.

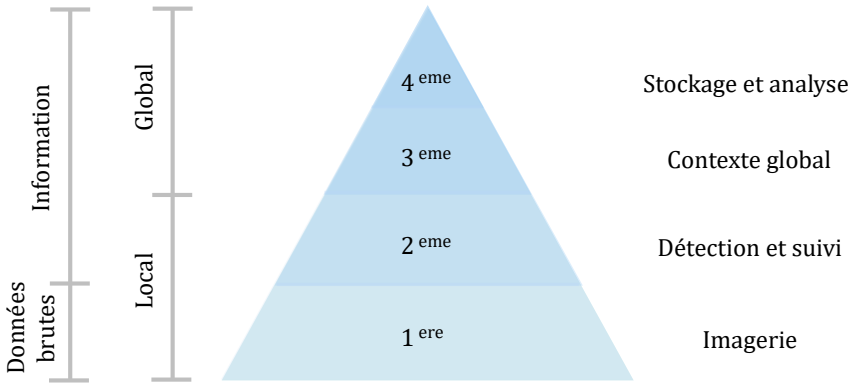


FIGURE 4.3 – Niveaux de description d'information

Par exemple, si nous souhaitons obtenir la localisation des personnes dans une image de profondeur, nous pourrions avoir comme sortie une nouvelle image avec des boîtes englobantes autour de chaque personne. Cependant, cette sortie ne répond pas à nos exigences : à la place, nous fournissons un vecteur de HFDs pour représenter les personnes à l'intérieur de la scène. C'est significatif parce que nous savons combien de personnes sont dans la scène par la taille du vecteur et chaque HFD a un point qui représente la position d'une personne dans la caméra. Cette information est autosuffisante car on n'a pas besoin de l'image originale pour produire de nouvelles informations, telles que la distance entre les personnes. Enfin, l'information est concise car nous passons d'une masse de pixels représentant une personne vers un HFD, représentant des coordonnées de cette personne dans l'image.

4.3 Gestion de données globales

Nous présentons notre approche de conception d'un système multi-caméra pour le suivi des personnes, en respectant les principes définis dans l'introduction de ce manuscrit de thèse, à savoir l'évolutivité, la facilité d'installation, l'adaptation et l'autonomie. Nous apportons des explications concernant le processus de positionnement des caméras dans un repère global en utilisant le SCG. D'autre part, nous expliquerons la gestion du suivi de personnes entre les caméras en utilisant le SIG.

4.3.1 Calibration de plusieurs caméras

La méthode pour estimer le positionnement des caméras dans un repère global est désignée par la calibration extrinsèque des caméras. Dans cette section, nous présentons la méthode de calibration de plusieurs caméras de profondeur en position zénithale. Cette calibration nous sert à créer un repère unique et global pour les positions des personnes qui seront capturées par les différentes caméras intelligentes de notre système.

Pour cela, on établit la relation spatiale entre le système de coordonnées globales SCG et le système de coordonnées de chaque caméra. Cette relation est normalement exprimée comme une roto-translation. Comme les caméras sont fixes, l'estimation de cette roto-translation est faite une seul fois à l'initialisation.

Une roto-translation est composée par un vecteur de translation $\mathbf{t}$ et une matrice de rotation $\mathbf{R}$ de taille 3×3. Un point X_i est transformé vers un point X'_i dans un deuxième système de coordonnées par l'équation [FP02, Sze10] :

$$\mathrm{X}'_i = \mathbf{R}\mathrm{X}_i + \mathbf{t} \tag{4.1}$$

Pour estimer la roto-translation d'une caméra par rapport à l'autre, on doit acquérir un nombre important de points communs entre les 2 caméras (donc entre les 2 systèmes de coordonnées) puis résoudre le système d'équations linéaires surdimensionné impliqué par 4.1. Cela se traduit par la minimisation de l'erreur quadratique E de rétroprojection des points en question :

$$\mathrm{E} = \sum_{i=1}^{\mathrm{M}} |\mathrm{X}'_i - \mathbf{R}\mathrm{X}_i + \mathbf{t}|^2 \tag{4.2}$$

La difficulté de cette calibration est de trouver les points en commun entre les deux caméras. Normalement, dans les méthodes de reconstruction 3D, on extrait les points d'intérêt [Rus09] de la scène pour estimer la roto-translation. Les points d'intérêt sont des points caractéristiques de la scène. Dans notre cas spécifique (de caméras de profondeur installées au plafond), la difficulté de la calibration multi-caméra augmente d'une part par la distance entre la caméra et les objets de la scène [SBR12] et d'autre part, par les caractéristiques des surfaces observées [BKIM13]. D'abord, la majorité des objets dans la scène sont normalement éloignés de la caméra et la précision de notre capteur diminue au-delà de 4 mètres, c'est-à-dire qu'il y a une forte chance que les mesures de ces objets soient erronées, rendant l'estimation fausse (voir chap. 2). En deuxième lieu, les surfaces des objets sont quasi parallèles ou perpendiculaires au plan de la caméra (comme les murs, sol et mobilier), elles possèdent des normales homogènes (avec la même direction), rendant la caractérisation difficile et réduisant la possibilité de trouver des points d'intérêt. Pour ces raisons, les méthodes classiques de calibration ne fonctionnent pas dans ce cas.

Dans la littérature, on trouve principalement deux méthodes pour calibrer les caméras 3D en position zénithale. La première méthode [SBR14, SBR12] génère les correspondances à partir de la position du centre de gravité des objets de calibration placés dans la région de chevauchement entre les champs de vision des caméras voisines. L'objet de calibration utilisé est un cercle en bois posé sur une base mobile. Les positions des objets de calibration sont segmentées manuellement sur les images d'entrée pour créer les correspondances et l'utilisation ultérieure dans la minimisation de l'erreur de la distance entre les correspondances (Eq. 5.3).

La deuxième méthode [BKIM13, GBMH15, ZYB+17] utilise les détections automatiques d'une seule personne à l'intérieur des champs de vision des différentes caméras du système pour générer les correspondances. Au moment de la calibration, le système est mis en mode calibration et une seule personne entre dans les champs de vision du système. Le système commence à détecter la personne en extrayant le sommet de la tête pour enregistrer les correspondances entre les caméras concernées. Lorsque la personne a traversé les champs de visions des caméras, leur système utilise les positions du centre de gravité de la personne extrait dans chaque caméra, pour déterminer la roto-translation en utilisant l'équation (Eq. 4.2). L'avantage de cette deuxième méthode par rapport à la première est l'extraction automatique de correspondances. Le désavantage est que ces correspondances sont extraites en mouvement et sur un objet non symétrique, ces deux caractéristiques introduisant des bruits d'estimation. Le bruit introduit par le mouvement de la personne est dû au fait que les caméras ne sont pas synchronisées, c'est-à-dire qu'elles peuvent acquérir des images à des instants différents. Les images obtenues sont toutefois traitées comme si elles avaient été prises simultanément (synchronisées). La forme de la tête est elle-même déformée par la perspective différente des 2 caméras. De plus, cette méthode devient plus complexe à utiliser dans un scénario de taille plus importante, comme par exemple dans le travail [GBMH15] où les auteurs utilisent une calibration manuelle des caméras (pour trouver les valeurs de **R** et **t)**, en déplaçant manuellement le nuage de points qui représente la scène partielle (obtenue par chaque caméra) d'un centre commercial.

Comme la calibration des caméras est un processus essentiel pour le fonctionnement du système, nous proposons un équilibre entre une approche manuelle mais précise, et une approche automatisée mais bruitée. Nous proposons l'extraction automatique des positions d'un objet de calibration pour calculer la roto-translation entre deux caméras. Notre objet de calibration est composé d'une pièce en bois circulaire et d'un trépied mobile en direction des axes X et Y et réglable dans l'axe Z entre 1,4 et 1,8 mètres. D'abord, on pose dans la zone commune des caméras à calibrer. Ensuite, on déplace l'objet sur au moins 4 positions différents, pour estimer la position de la deuxième caméra par rapport à la première. Plus on prend de points, plus précise sera la calibration. L'utilisation d'un objet de calibration assure une estimation plus précise du fait qu'on élimine le bruit introduit par le mouvement et les points acquis sont à l'intérieur de la portée en profondeur de la caméra.

4.3.2 Suivi de personnes par plusieurs caméras

Disposer d'un système de plusieurs caméras signifie avoir plusieurs trajectoires générées par la même personne. Le défi est de produire une seule trajectoire. Ce sujet n'est pas nouveau et a déjà été abordé par d'autres auteurs.

Dans le travail de Seer et al [SBR12] les auteurs utilisent un algorithme de minimisation de la correspondance des trajectoires. Ils utilisent une méthode post-traitement faite hors-ligne à la fin de l'acquisition des images. D'abord, les trajectoires sont projetées vers le même système de coordonnées. Ensuite, ils utilisent l'algorithme hongrois [Mun57] à l'intérieur d'un processus itératif pour associer le point (spatio-temporel) final d'une trajectoire au point initial d'une autre. Dans chaque itération, on extrait les trajectoires associées, en respectant un seuil, du groupe de trajectoires à traiter. De plus, le seuil est augmenté pour passer à l'itération suivante. Après le processus itératif, les trajectoires associées sont sous-échantillonnées pour avoir une trajectoire lissée. Malgré les bons résultats de cette méthode hors ligne et itérative, on trouve qu'elle ne s'adapte pas à notre contrainte d'évolutivité, aux ressources de calculs limitées et aux besoins d'autonomie du système. Donc, on a besoin de trouver un processus capable de respecter ces contraintes.

Dans cette thèse, nous avons fait le choix de traiter le problème du suivi des personnes dans un système multi-caméras, d'une part en temps-réel (*agent de comptage)* et d'autre part avec une approche de gestion des étiquettes centralisées (*superviseur*). Pour ce faire, on traite d'abord localement le suivi de la manière décrite dans le chapitre 3. On obtient une liste de HFD avec ses positions dans le repère local de la caméra. Ensuite, en utilisant la calibration décrite dans la section précédente, on transforme les positions locales en positions globales dans un repère global commun. Le calcul de la position globale est fait localement, mais l'étiquetage est géré par

le système d'identification globale, par le superviseur. Quand une personne entre pour première fois dans le champ de vision d'une caméra, cette caméra est *en charge du suivi* de la personne et demande une étiquette au superviseur. En conséquence, les coordonnées de cette personne correspondent à l'information générée par la caméra *en charge* alors que l'étiquette est fournie par le superviseur.

Dès qu'une personne s'approche du bord du champ de vision de la caméra, on évalue s'il existe une caméra voisine qui surveille cette frontière. Dans ce cas, on informe les caméras voisines qui ont une frontière en commun de la même manière que celle de [DSE+13]. Ensuite, on envoie le HFD de la personne dont les coordonnées sont ramenées au repère global. Une fois que la caméra voisine reçoit le HFD, on minimise la distance d^{HFD} (section 3.2.1) entre le HFD reçu et les détections actuelles de la caméra voisine en calculant la meilleure correspondance. Quand la personne arrive à la limite de l'image de la caméra en charge, qui se trouve dans le FoV de l'autre caméra, on transfère la gestion de la personne à la deuxième caméra. On appelle ce processus de transfert de gestion des personnes « handover » [FBBS06].

De plus, la caméra en charge envoie à chaque détection la position globale de la personne traquée. De cette manière, on trouve dans le serveur la trajectoire complète des personnes, en évitant les processus hors-ligne et en produisant une trajectoire globale en temps-réel.

4.4 Conclusions

Pour assurer une évolutivité de notre solution, nous avons utilisé un réseau distribué et extensible, capable de réaliser le suivi des personnes dans des grands espaces. Pour satisfaire les objectifs définis et les contraintes industrielles, nous avons exploité notre développement d'une caméra intelligente autonome, pour construire un réseau distribué de caméras intelligentes, en répartissant l'observation d'un grand espace sur différents nœuds (caméras intelligentes) de manière collaborative.

Dans l'architecture présente, la caméra intelligente a la charge de l'acquisition des données, du suivi de personnes, du partage des informations et de la couche applicative (fonction comptage). L'entité identifiée comme serveur traite et analyse les données globales d'un haut niveau de description produites par les caméras.

Pour lever les verrous technologiques, nous avons utilisé la calibration extrinsèque entre caméras pour construire un système de coordonnées globales (SCG) utilisé comme repère global pour le suivi de personnes et la création des trajectoires globales dans le temps. De plus, nous avons mis en place un système d'identification globale SIG qui permet d'affecter un identifiant unique à chaque personne suivie dans l'espace couvert par toutes les caméras du système.

Nous avons défini quatre niveaux de description d'information qui commencent par des données brutes acquises par les capteurs, qui subissent des transformations d'un niveau à l'autre pour produire des informations plus compactes et plus riches, c'est-à-dire en contenant plus d'informations abstraites liées aux personnes suivies. Le dernier niveau de description est traité dans le chapitre suivant et concerne le comportement des personnes suivies par le réseau des caméras que nous avons mis en œuvre.

Dans ce chapitre, nous avons contribué : à l'évolutivité, à la facilité d'installation, à l'adaptation et à l'autonomie de notre système, par rapport aux objectifs proposés dans l'introduction.

Chapitre 5

Analyse de comportement des personnes en mouvement

Sommaire

Les méthodes d'analyse comportementale des personnes en mouvement sont fondées sur l'utilisation de systèmes de suivi des personnes (cf. chapitre précédent), dans le contexte académique et industriel. Dans ce chapitre, nous aborderons ces méthodes puis nous présenterons nos méthodes basées principalement sur l'évaluation de l'utilisation de l'espace et l'analyse des trajectoires des personnes en mouvements. Notre objectif est de développer de nouvelles applications industrielles. Nous présentons les résultats issus de l'application de méthodes originales proposées par d'autres auteurs et appliquées sur nos jeux de données ainsi que des jeux de données externes de suivi des personnes.

5.1 Contexte de l'analyse comportementale

Cette section permet de donner une vision générale des sciences du comportement, de présenter les travaux qui émergent de l'utilisation de caméras de profondeur et des besoins industriels attendus par les objectifs de ce chapitre.

5.1.1 Sciences comportementales

Selon [Bov64],on peut définir le comportement comme l'ensemble des ***mouvements observables*** exécutés par un individu, ses interactions avec son environnement et ses congénères. De plus, on divise cet ensemble en groupes de mouvements par rapport à la *fonction* qu'ils accomplissent. Par exemple, « on parlera ainsi, de comportement alimentaire lorsqu'on voudra parler de l'ensemble des mouvements impliqués dans la nutrition » qu'ils accomplissent. Par exemple, « on parlera ainsi de comportement alimentaire lorsqu'on voudra parler de l'ensemble des mouvements impliqués dans la nutrition » [Bov64]. On peut caractériser le mouvement observé du comportement à partir de *trois propriétés* : le relationnel, la dynamique et la haute dimensionnalité [GMPK+14]. Le comportement est d'ordre relationnel entre un individu et « son environnement et les autres individus ». Le comportement est dynamique s'il évolue au fil du temps. Ces périodes de temps pendant lesquelles on étudie le comportement sont dénommées *séries chronologiques*. Enfin, le comportement revêt une dimensionnalité élevée par le nombre de variables que l'on peut utiliser pour le décrire et qui traduit sa complexité.

L'étude actuelle du comportement humain *moderne* s'appuie sur la systématisation des données pour l'extraction de ces propriétés. Gomez-Marin souligne les *trois axes principaux* à partir desquels ces données sont conceptualisées : le nombre de contraintes, le niveau de description et la dimensionnalité des données.

- Dans le premier axe, les *contraintes imposées* (*CIs*) dans l'expérience d'observation limitent l'étendue du comportement de l'individu. La plage de validité de cet axe varie d'un scénario très *contrôlé* (fortement restreint) à un scénario *non contrôlé* (sans restrictions, c'est-à-dire dans un environnement naturel).
- Dans le deuxième axe, le *niveau de description* (*ND*) fait référence à la nature des données et à leur pertinence par rapport à l'expérience évaluée. Celui-ci dépend du niveau d'abstraction des observations. A un niveau d'abstraction faible, on trouve habituellement une grosse quantité de données pour décrire un comportement. Plus on remonte dans le niveau d'abstraction, plus la quantité de données nécessaires va diminuer.
- Enfin, le troisième axe concerne la *dimensionnalité des données* (*DD*). Ceci signifie la quantité d'informations et le détail d'une observation. Cet axe est relié à la propriété de dimensionnalité élevée en ajoutant la précision et le volume de données pouvant être gérées par les technologies actuelles.

L'étude du comportement entre dans le cadre des sciences comportementales. Selon Gomez-Marin, il existe deux approches fortement différenciées : psychologique et éthologique. La première approche utilise principalement des environnements *contrôlés* pour leurs études. Elle se focalise sur les comportements appris pour analyser la relation entre un stimulus, les actions et leurs conséquences. De la même façon, l'approche psychologique s'intéresse aux

principes généraux de l'apprentissage et de la motivation des individus [GMPK+14]. Cette dernière s'intéresse à l'analyse causale de ***mouvements observables et objectivables*** pour des individus dans leur environnement naturel (*non contrôlés*) [Amy06]. Du point de vue de Bovet [Bov64], l'approche *psychologique* est basée sur une étude subjective du comportement d'individus sans prendre en compte les aspects biologiques ni éléments de leur environnement. Une variante de cette approche est appelée le « *behaviorisme* », qui aborde l'étude du comportement d'une manière objective. Cependant, cette approche est focalisée sur l'étude des situations strictement contrôlées et l'adaptation du comportement (stimulus-réponse) des individus à ces situations (un *scénario restreint* si l'on se réfère au *premier axe cité ci-dessus*). Contrairement à la psychologie, l'éthologie (étant une branche de la biologie), est basée sur des observations du comportement en utilisant une méthode scientifique et en répondant aux trois questions principales de la biologie (causalité, survie et évolution) [Tin63] dans des situations *non contrôlées* (un *scénario non restreint* évoqué dans le premier axe), par opposition au « behaviorisme ». L'approche de notre travail sera plutôt inspirée de l'éthologie, considérant notre intérêt pour l'étude des personnes en mouvement dans des espaces publics d'une manière non-intrusive.

Pour approfondir, l'éthologie décrit le comportement d'un groupe d'individus à travers d'éthogrammes. Un éthogramme est un inventaire de mouvements communs à plusieurs individus. La caractérisation précise des tous les éléments d'un mouvement corporel se heurte aux difficultés suivantes : la vitesse des mouvements, la quantité des éléments du corps ou des objets portés par l'individu qui bougent simultanément, les conditions de l'environnement, l'influence de l'observateur sur le comportement de l'individu et la durée de l'observation [Bov64]. Il est très intéressant de noter que dès 1964, Bovet a remarqué l'importance de l'utilisation des technologies telles que les caméras de cinéma ou les enregistreurs de sons, comme outils auxiliaires pour diminuer la complexité et résoudre les difficultés d'interprétation. C'est ici que notre solution peut apporter des éléments pratiques et pertinents pour réaliser une étude du comportement humain, parce que nous sommes capables de détecter des personnes, leur mouvement et de relever les caractéristiques intrinsèques des personnes d'une manière automatique et systématique. Notre capacité à traiter un grand volume de données, nous permet de déterminer le mouvement commun d'un ensemble de personnes dans de grands espaces.

Dans le domaine des sciences comportementales, une des idées nouvelles proposées dans [GMPK+14] vise à concevoir une unification des approches de l'éthologie et de la psychologie. Celle-ci est possible principalement pour deux raisons. D'abord, l'émergence de nouvelles technologies pour la perception humaine a permis d'acquérir un grand nombre d'attributs objectivables (avec une description plus riche pour décrire les mouvements d'un individu. En second lieu, la capacité de sauvegarder et de traiter une grande quantité de données (big data) permet d'effectuer de nouvelles études sur les mêmes données d'une expérience avec différents types d'approches pour expliquer le comportement, en respectant les paradigmes de chacun. Nous ajoutons à cette dernière la possibilité d'évolutivité de ces technologies par rapport à l'augmentation de l'espace et le nombre de personnes observées, en induisant un plus grand volume de données acquises simultanément.

De plus, Gomez-Marin et al. mettent en évidence la complexité d'établir une description complète du comportement. Ce dernier doit être considéré dans un environnement et décrit sémantiquement. C'est-à-dire qu'on doit bien définir les attendus (lieux et objectifs) de nos expériences.

5.1.2 État de l'art

Cet état de l'art permet de faire le lien entre le contexte des sciences du comportement et les travaux associés au suivi des personnes et leurs intentions par l'utilisation de systèmes optiques 3D. Nous relevons les travaux qui ont créé un système complet, de la description et la mise en œuvre de l'approche technologique, jusqu'à l'analyse comportementale. Les travaux de la littérature concernée peuvent être classés en distinguant les domaines de la reconnaissance des actions et celui de la modélisation du comportement microscopique du piéton [BK15]. De plus, on trouve différents travaux, isolés, qui s'intéressent aux applications spécifiques comme le comportement d'achat sur les étagères de supermarché ou le comportement des piétons dans les transports publics.

Dans les travaux pour la reconnaissance des actions, on trouve [ASS16, DWB+15]. Les auteurs ont développé des approches que nous décrivons comme éthologiques, parce qu'ils essaient de reconnaître des mouvements humains (la forme et la dynamique) pour identifier l'action de la personne observée dans une série chronologique. Ces travaux utilisent l'évolution de l'adaptation d'un squelette (formé par des articulations virtuelles du corps humain) sur les personnes détectées, pour trouver une similarité avec les actions préalablement connues par le système. Ces systèmes utilisent les traces des trajectoires de différentes articulations pour reconnaître l'action exécutée par la personne observée. Cependant, ces travaux utilisent une approche où la caméra, en position latérale, se focalise sur les actions des personnes mais pas sur leur relation avec l'environnement. Ces travaux n'abordent pas leur problème du comportement dans de grands espaces parce ce qu'ils ne disposent pas de multi-caméras.

Dans le même domaine de reconnaissance des actions, [MA13, MA16] étudient le comportement d'achat des personnes dans un supermarché. Ce travail se focalise sur la relation entre les clients et les produits dans les rayons du supermarché pour comprendre leur décision d'achat. Au contraire de travaux de reconnaissance des actions décrites précédemment, les auteurs utilisent une caméra en position zénithale pour détecter les personnes et suivre le mouvement des bras en temps-réel d'une manière non intrusive. Dans cette approche, les méthodes de traitement des images sont robustes mais utilisent beaucoup de ressources de calcul pour détecter les personnes, estimer leur pose et suivre les différentes articulations des bras. Cependant, ces auteurs, même s'ils visent l'objectif de reconnaissance d'une action d'achat (attraper un objet sur une étagère), ne proposent pas de moyens pour analyser le comportement ni de donner les pistes pour prouver un acte d'achat.

Selon [SBR14, BK15] la modélisation du comportement microscopique du piéton s'intéresse au comportement individuel de chaque piéton. Par exemple, celui de piétons qui se déplacent vers un but tout en s'évitant les uns des autres, ou celui de formation des groupes. Cette modélisation s'intéresse également à la simulation et prédiction des mouvements de plusieurs piétons dans la foule. Ces modèles servent à la simulation et la prédiction du mouvement de plusieurs piétons à l'intérieur d'un espace donné [SBR12]. De plus, ces travaux sont fondés sur le modèle de force sociale [HM95], en créant des variantes qui décrivent différentes forces dynamiques pour obtenir un meilleur modèle de prédiction de mouvement. Par exemple, ces variantes du modèle de force sociale sont : la répulsion circulaire [HM95], la répulsion elliptique [HJ09] et la combinaison de forces de ralentissement et d'évasion [RMSB11]. De plus, les applications des modèles de comportement microscopique de piétons se focalisent sur l'évaluation de l'accessibilité [HD04], la sécurité des espaces pendant les situations d'urgence [SKK+11] et la détection automatique de comportement inhabituel [PMS04].

Parmi les travaux de la modélisation du comportement microscopique du piéton, on trouve à Seer et al [SBR12, SBR14] qui étudient la cinétique du mouvement humain dans le couloir d'une université pour construire un modèle de prédiction des mouvements des personnes. Cette étude se focalise sur la relation du mouvement d'un individu par rapport à la foule. Pour ce faire, les auteurs détectent des personnes dans un système composé de plusieurs caméras, générant des trajectoires dans un système commun de coordonnées. Ensuite, ils utilisent ces trajectoires, dans un processus de post-traitement, pour calibrer les paramètres du modèle de force sociale de manière à reproduire les mouvements des individus dans deux situations simples : un couloir permettant un transit libre et le même couloir présentant un obstacle. Ce modèle vise à simuler le comportement des individus déambulant dans ces deux situations. La détermination des trajectoires par leur méthode de suivi permet de sélectionner un groupe des trajectoires dites « de calibration » et un groupe de trajectoires dites « de validation » pour évaluer la précision du modèle de prédiction de mouvement. Ces travaux servent donc à améliorer les modèles de foules et à comprendre leur comportement pour le design des espaces dédiés à accueillir des foules, comme par exemple des stades.

Les travaux du laboratoire ATC [web16a]concernent le comportement des personnes à l'intérieur d'un centre commercial. Ces travaux se basent sur la même approche technologique pour suivre des personnes en utilisant un réseau de capteurs 3D en position zénithale [BKIM13, GBMH15]. Cependant, on trouve deux voies de recherche : la première utilise un modèle de force

sociale pour prédire et simuler le comportement de personnes [ZBK14]. La deuxième s'intéresse à l'utilisation de l'espace et à la segmentation de trajectoires [BK15]. Dans les travaux de [ZBK14], les auteurs modélisent le comportement microscopique de la foule. A la différence du modèle de force sociale [HM95], ce modèle prend en compte l'interaction sociale des individus en remplaçant l'individu par des groupes de 3 personnes et en appliquant à ce groupe les forces décrites plus haut. Les auteurs proposent d'analyser la formation de ces groupes selon la densité de la foule. Ils proposent deux types de formation : l'une « côte à côte » à basse densité et une formation « fermée » en V pour des densités plus élevées. De plus, pour valider leur modèle, les auteurs ont dû utiliser deux « codeurs » (les personnes qui introduisent la vérité-terrain manuellement en regardant la séquence vidéo) pour étiqueter la relation entre personnes, en ajoutant de la complexité à l'utilisation de cette approche.

S'agissant de la deuxième ligne de recherche, c'est-à-dire l'utilisation de l'espace et la segmentation de trajectoires, le travail de [BK15] est focalisé sur la collection de données (images et trajectoires) sur une longue durée dans le même centre commercial que [ZBK14]. Le travail de [BK15] est le seul qui utilise des capteurs 3D pour analyser le comportement macroscopique (à l'échelle de plusieurs individus) de la foule. Pour ce faire, les auteurs présentent, sur une année, différentes analyses statistiques sur plusieurs séries chronologiques dans les conditions suivantes : le centre commercial a une surface de 900 m_2, l'enregistrement des données est réalisé un jour particulier en semaine (mercredi) et un jour de week-end (dimanche). Les auteurs proposent donc une analyse de trajectoires, de l'utilisation de l'espace et des changements des propriétés de la foule (comme la densité et la vitesse des personnes) selon les heures de la journée et en fonction des différentes périodes de l'année.

Dans les travaux de Kirchner et al. [KCCVZ15] et Virgona et al. [VKA15], les auteurs montrent d'une part l'utilisation des technologies de perception humaine pour étudier l'influence d'une signalisation (stimuli) sur le comportement humain dans le quotidien et d'autre part pour améliorer le confort des passagers dans les transports en Australie. Leur système est capable d'identifier les zones bondées sur le quai d'un métro et d'activer une signalisation intelligente à distance pour proposer, en temps-réel, aux piétons un chemin moins congestionné. Ce travail est un bon exemple d'une nouvelle technologie (de perception et de signalisation intelligente), qui rapproche l'éthologie et la psychologie. D'un côté, les notions de l'éthologie pour étudier l'individu dans son environnement naturel et le principe de non-intrusion par la méthode d'observation (perception) sont respectées. D'un autre côté, il est possible d'évaluer l'aspect psychologique de la réponse des personnes après un stimulus (signalisation intelligente) ou encore l'impact de l'amélioration du confort des passagers et celle du service de transport. Cependant, ce travail explore une problématique très spécifique au transport qui ne s'adapte pas à nos besoins. De plus, celle-ci sert à observer des zones déjà identifiées comme pertinentes et n'évalue pas le reste de l'espace pour avoir une compréhension globale de cet espace utilisé par les personnes.

5.1.3 Contexte industriel

Nous rappelons que cette thèse s'effectue dans un contexte de thèse CIFRE en lien avec la réponse à un problème posé par un industriel. Cela nous a incités à évaluer les besoins du marché en matière de services et d'innovation dans les systèmes de comptage, ce que l'on appelle l'axe de compétitivité industrielle. L'entreprise Shopline Elecronic propose trois types de services : les services d'analyses de comptage au niveau d'un point de vente (boutique), au niveau d'une chaîne de boutiques (plusieurs points de vente avec les mêmes produits) et dans des centres commerciaux. En termes de points de vente, l'entreprise Shopline offre : l'analyse du flux de visiteurs et leur répartition sur la journée (Fig. 5.1a) ou la semaine (Fig. 5.1b). En particulier l'entreprise peut fournir par exemple, aux heures de pointes, le taux de transformation de ventes *tx* (ratio ventes/visiteurs) par jour (Fig. 5.1a) et par rapport au mois précédent ou le même mois de l'année précédente ainsi que l'influence de la météo sur le nombre de visiteurs (Fig. 5.1c). En termes de chaîne de boutiques, l'entreprise Shopline peut fournir des informations sur la performance des points de ventes (Fig. 5.1d), des comparaisons de fréquentation entre différents points de vente suivant le moment de l'année. Dans les centres commerciaux, l'entreprise Shopline peut identifier les points de passages les plus fréquentés. Ces services permettent aux

clients de prendre une décision sur l'optimisation des forces de ventes, des forces de sécurité et les actions marketing. Sur ce dernier point, ces services permettent par exemple, d'évaluer l'augmentation des flux de visiteurs à la suite d'une campagne publicitaire, ou de mesurer les effets de changement de la décoration ou de l'atmosphère sur le flux d'entrée.

Cependant, ces types de services sont communément proposés par la concurrence. Pour cette raison, nous avons cherché de nouvelles informations pertinentes à explorer, comme l'utilisation de l'espace. Cette vision nous demande d'utiliser des méthodes d'analyse comportementale des personnes dans l'espace observé. Ces méthodes ont pour objectif de faire évoluer les systèmes de comptage à une analyse des trajectoires des visiteurs et leur utilisation de l'espace.

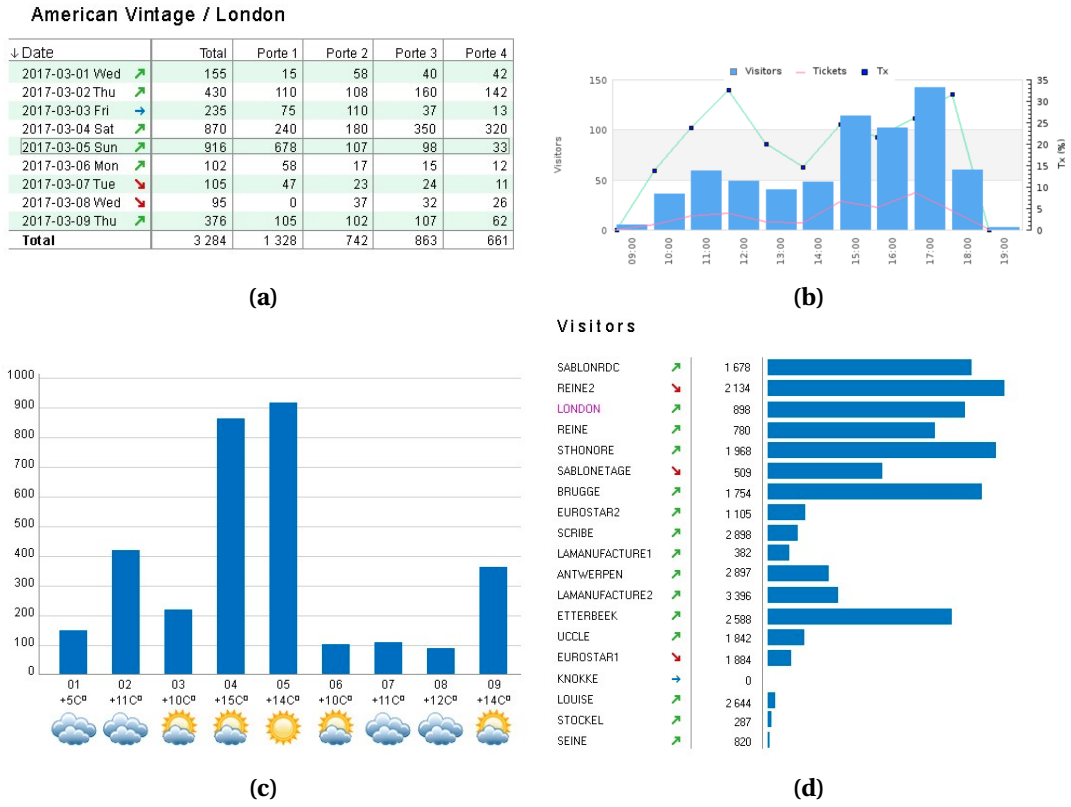


FIGURE 5.1 – Exemple de rapports générés par l'entreprise Shopline. a) Comparatif de fréquentation journalière dans une boutique. b) Comparatif du nombre de visiteurs horaires et *tx* c) Comparatif de visiteurs par rapport à la météo. d) Comparatif du nombre de visiteurs sur un jour par boutiques.

5.1.4 Évaluation des besoins pour l'analyse comportementale

L'étude des sciences comportementales montre l'intérêt de l'éthologie comme approche intéressante pour notre sujet de recherche. De plus, les travaux de la littérature concernant le comportement humain montrent que :

- La reconnaissance des actions implique qu'on a besoin d'une grande quantité d'informations sur le corps humain pour être capable d'identifier des actions dans une série chronologique. L'évolution de ces méthodes sur de grands espaces n'est pas encore démontrée et on considère que le détail requis (par exemple le suivi des articulations du corps) ne peut pas être obtenu facilement puisqu'il y a trop d'informations, issues de plusieurs caméras, à consolider.
- Les modèles de prédiction de mouvement nous permettent de simuler le comportement d'un ou plusieurs individus dans une foule mais ne donnent pas d'information sur l'environnement.
- L'analyse de la foule permet de provoquer un changement de comportement dans un système de transport mais ne fournit pas une analyse de comportement.

- Les travaux les plus avancés sur le comportement de personnes en mouvement dans un espace commercial montrent que l'analyse des trajectoires et l'utilisation de l'espace sont possibles pour approfondir l'étude du comportement humain.

A partir de ce constat, nous nous sommes fixés les objectifs et les principes suivants :

- Objectifs des méthodes :
  - Compréhension de la relation entre les personnes et l'espace observé.
  - Identification des habitudes de déplacement en analysant les trajectoires.
- Principes de conception des méthodes :
  - Facilité d'utilisation et de mise en œuvre en utilisant les données de suivi.
  - Évolution vers de grands espaces.
  - Distribution de calcul dans le temps et vers le réseau des nœuds.
  - Faisabilité pour fonctionner sur différents types de caméras (2D et 3D), justifié par le fait que l'entreprise Shopline utilise les deux types de capteurs pour les solutions de comptage.

5.2 Méthodes d'analyse comportementale

Nous décrivons maintenant les méthodes que nous avons développées et utilisées pour l'analyse comportementale avec les objectifs décrits dans la section précédente. Pour comprendre la relation spatio-temporelle entre les personnes et l'espace observé, nous avons développé des méthodes d'analyse complémentaires pour étudier différentes modalités de l'interaction individus-espace-déplacements. Pour ce faire, nous distinguons deux catégories d'analyses :

Utilisation de l'espace : composée par les méthodes de détection de personnes en temps-réel dans les zones d'intérêt, la génération de la carte d'occupation et la détection des points d'entrée et de sortie.

Analyse des trajectoires : composée par la segmentation d'un flux comportemental, la validation et la représentation des trajectoires.

Un autre défi est de trouver la meilleure représentation des informations obtenues pour synthétiser de manière claire les résultats et optimiser le transfert d'informations vers d'autres acteurs [FKSS13, LHW07] Toutes les méthodes évoquées dans cette section ont comme entrées les trajectoires extraites par un système de suivi de personnes. A partir des celui-ci, on doit extraire des informations et les présenter aux utilisateurs. Pour montrer des données à traiter, on présente dans la figure 5.2, 74 trajectoires obtenues à l'entrée d'une boutique. La figure 5.2 illustre une manière de représenter les différentes trajectoires effectuées par les individus dans la scène, en mettant en évidence leurs points de début et de fin de cette trajectoire.

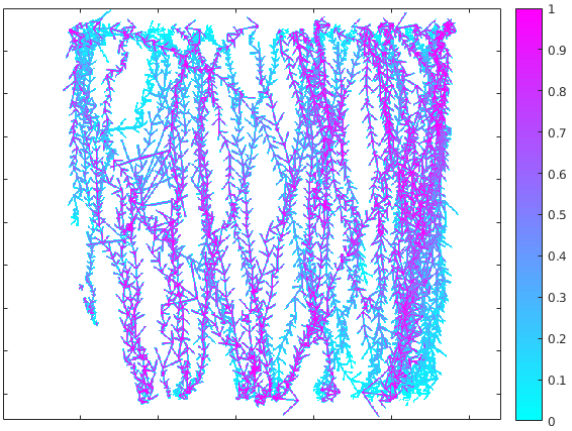


FIGURE 5.2 – Information d'entrée pour la création de la carte d'occupation. Trajectoires reconstituées à partir des données discrètes des trajectoires $\tau^j \in T$ (entrée en cyan et sortie en magenta).

5.2.1 Analyse de l'utilisation de l'espace

Dans cette section, nous nous sommes intéressés à identifier les zones les plus visitées [BK15] ou les points d'intérêt points d'intérêt (***Points of Interest*** : POIs)des scènes surveillées. Ceci nous permet, en fonction des conditions spécifiques d'une zone, de lancer des alarmes à la destination des utilisateurs du système. L'intérêt d'identifier les POIs est de comprendre ce qui attire les personnes et de mesurer cette « attraction ». De plus, en utilisant de nouvelles méthodes du Cloud pour reconnaître des objets dans les images [web17b], on pourra comprendre le contexte de la scène. Par exemple, dans le rayon légumes d'un supermarché, les POIs sont placés sur les tomates et les avocats, par contre il y a une très faible fréquentation sur les oignons.

Détection de personnes en temps-réel dans les zones d'intérêt

Pour le marché du comptage de personnes, nous avons besoin de savoir en temps-réel combien de personnes se trouvent dans un endroit particulier comme par exemple une file d'attente dans un supermarché ou à la caisse d'un magasin, de manière à améliorer le service et diminuer le temps total d'attente des clients, augmentant ainsi la qualité du service. Dans ces cas, d'autres informations peuvent être utiles, comme la durée de présence, le temps moyen d'attente des clients et le nombre de personnes dans la file d'attente. Dans cette section, nous nous focalisons sur l'analyse de l'utilisation de l'espace en évaluant un *espace spécifique* de notre champ de vision désigné par *zone*. Il faut noter que l'activité d'une zone dépend des activités des personnes à l'intérieur de cette zone. C'est par une analyse spécifique que nous pourrons créer des alertes en temps-réel, et déclencher une action. Par exemple, si le temps d'attente de clients dépasse une certaine limite, l'action sera d'appeler une caissière pour ouvrir une nouvelle caisse.

L'analyse par zone d'intérêt que nous proposons permet d'évaluer à l'intérieur de la zone : le nombre de personnes qui la traverse, le temps passé par chacune et le temps moyen de présence de ces personnes. Pour ce faire, on définit des *zones* z^i de forme rectangulaire à l'intérieur du champ de vision de la caméra. Ainsi on détermine, à chaque nouvelle acquisition d'image, si chaque trajectoire d'une personne τ^j est à l'intérieur d'une zone. Au moment où un point d'une trajectoire τ^j rentre dans une zone, on crée une nouvelle *sous-trajectoire d'attente* γ^j en associant la zone et le point. On désigne ce point comme un point *d'attente d'une personne* p_i^j. Pour chaque nouveau point d'une trajectoire (identifié dans une nouvelle image acquise), on évalue également si celui-ci est à l'intérieur de la zone pour l'ajouter à la sous-trajectoire. Au moment où la trajectoire sort de la zone, on détermine la fin de la sous-trajectoire. De cette manière, une trajectoire peut avoir de multiples sous-trajectoires d'attente associées à la même ou aux différentes zones.

Nous allons présenter l'algorithme de « détection de personnes en temps-réel dans les zones d'intérêt » en utilisant comme moment de référence la dernière image (N) acquise par la caméra.

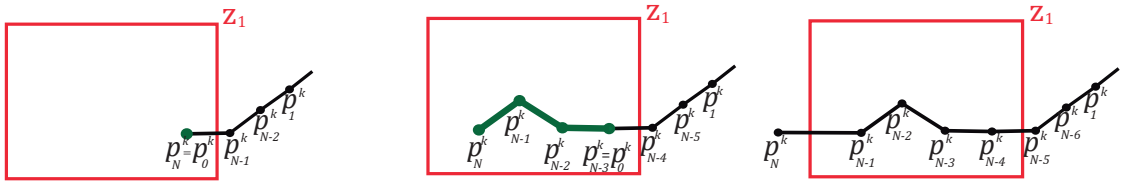
Ceci n'est pas une contrainte car n'importe quelle image du flux vidéo peut être considérée à un moment donné comme la dernière.

L'ensemble des trajectoires détectées pour l'ensemble des M personnes suivies au moment N est $\mathrm{T} := \{\tau^k | k = 1, ..., \mathrm{M}\}$. La trajectoire τ^k de la personne k est composée par le couple $(\Omega^k_{\mathrm{N}}, \mathrm{P}^k)$ où :

- Ω^k_{N} est le vecteur de caractéristiques HFD associé à la personne k détectée au moment N.
- P^k est l'historique des positions de détection de la personne k (en cours de suivi) dans des images précédentes, successives, défini comme un ensemble de points de détection $\mathrm{P}^k := \{p^k_1, \ldots, p^k_{\mathrm{N}}\}$ ordonnés chronologiquement. La $i^{ème}$ position de la personne k suivie est notée $p^k_i = \{x^k_i, y^k_i, t^k_i\}$ où x^k_i et y^k_i sont les coordonnées exprimées en pixels, à l'intérieur du champ de vision D_t (projection bidimensionnelle) d'une caméra et t^k_i est le temps d'acquisition du triplet.

L'utilisateur peut définir plusieurs zones rectangulaires z^j à l'intérieur du champ de vision D_t définies par leurs extrémités exprimées en pixels $[x^j_{\mathrm{L}}, x^j_{\mathrm{R}}] \times [y^j_{\mathrm{T}}, y^j_{\mathrm{B}}]$. Dans ces conditions, si la dernière détection d'une trajectoire τ^k appartient à la zone en question : $(x^k_{\mathrm{N}}, y^k_{\mathrm{N}}) \in z^j$ alors on peut définir une sous-trajectoire d'attente τ^k_j comme étant un sous-ensemble de la trajectoire initiale composée par : $\tau^k_j = (\Omega^k_{\mathrm{N}}, \mathrm{P}^k_j)$ où $\mathrm{P}^k_j = \{p^k_i | p^k_i \in \mathrm{P}^k \wedge (x^k_i, y^k_i) \in z^j\}$.

Dans le processus en temps-réel, la présence d'une personne est comptée pour la zone z^j s'il y a une sous-trajectoire d'attente associée à cette zone. La figure 5.3 illustre trois moments-clés de la traversée d'une zone par une personne suivie : l'entrée dans la zone (la sous-trajectoire n'a qu'un premier point noté p^k_0 Fig. 5.3a), juste avant de sortir de la zone (la sous-trajectoire a un maximum de points Fig. 5.3b) et après la sortie de la zone (la sous-trajectoire définie par cette zone n'existe plus Fig.5.3c). Ce processus se généralise à l'ensemble de personnes suivies et à l'ensemble des zones de surveillance définies par l'utilisateur.



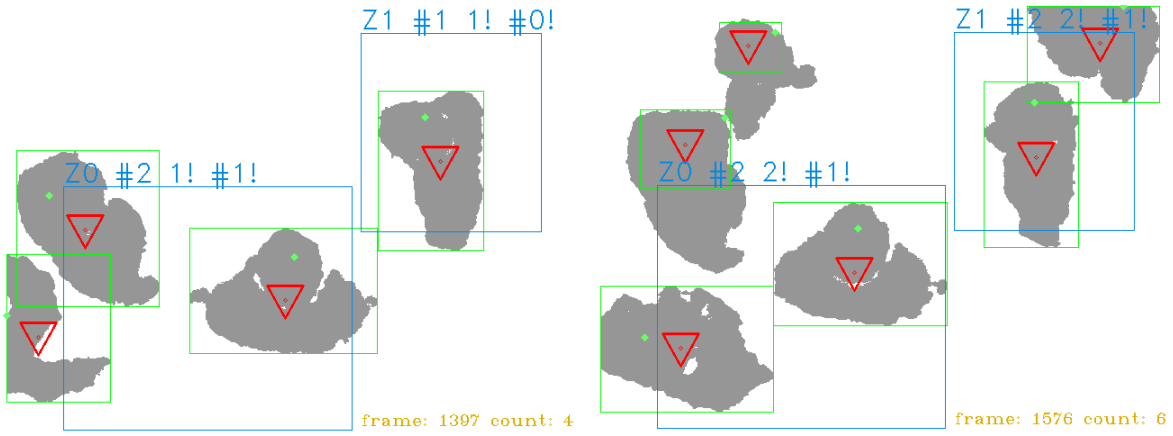
(a) Entrée dans la zone **(b)** Avant de sortir de la zone **(c)** Sortie de la zone

FIGURE 5.3 – Processus de transformation d'une trajectoire en une trajectoire d'attente.

Le *temps total* Δt^k_j d'une sous-trajectoire est défini comme la différence entre les temps du dernier et du premier point de la sous-trajectoire : $\Delta t^k_j = t^k_{\mathrm{N}} - t^k_0$ où t^k_{N} est le temps de la dernière position d'attente détectée au moment N et t^k_0 est le temps de la première position d'attente détectée à l'intérieur de la zone z^j.

Le *nombre de personnes* NP^j à l'intérieur d'une zone z^j est tout simplement le nombre total des sous-trajectoires associées couramment à cette zone alors que le *temps total d'attente* pour cette même zone z^j est $\Delta t_j = \sum_k \Delta t^k_j$

Ensuite, on utilise ces propriétés pour créer des alarmes en temps-réel, en définissant des seuils sur chacune. Dès qu'un des seuils est dépassé, le système envoie automatiquement un message d'alarme. Nous pouvons aussi établir une relation logique entre les zones, par l'évaluation de leurs propriétés. Une alarme peut être déclenchée quand les propriétés de deux zones dépassent certains seuils prévus, par exemple : $\Delta t_1 > 1$ minute et $\mathrm{NP}^2 \geq 3$ personnes (voir Fig.5.4).



(a) **(b)**

FIGURE 5.4 – Visualisation de la détection des personnes en temps-réel dans 2 zones à l'intérieur de l'image. Les zones sont représentées par des rectangles bleus.

Dans la figure 5.4, chaque rectangle porte l'information du nombre n de personnes à l'intérieur (marqué sur la figure comme $\#n$), le nombre de personnes m qui ont dépassé la limite de temps d'attente (marqué sur la figure comme $m!$) et une alerte (marquée sur la figure comme $\#k!$) quand le nombre de personnes n à l'intérieur de la zone est supérieur au nombre maximal configuré pour la zone évaluée. La différence entre les sous-figures (5.4a) et (5.4b) est le nombre d'alertes de temps de séjour ($m!$) qui sont augmentées du fait que ces personnes sont restées plus de 10 secondes dans la zone en évaluation de cet exemple. De plus, on observe les alertes de dépassement ($\#k!$) du nombre de personnes permis dans les zones, dans cet exemple où la valeur est de 1.

Cette analyse par zone d'intérêt nous permet d'évaluer les zones que l'utilisateur considère comme importantes et dépend de sa compréhension de l'environnement évalué. Mais comment peut-on vérifier si nous sommes en train d'évaluer l'espace pertinent ? Comment choisir une telle zone ? Cette analyse nous permet d'avoir une évaluation en temps-réel de la zone, toutefois on peut aller au-delà en repérant les points d'intérêt comme ceux qui correspondent au temps maximal de séjour des personnes ou ceux qui se réfèrent au flux maximum de ces personnes à cause de la configuration de l'espace évalué.

Génération de la carte d'occupation

Pour identifier les zones les plus visitées et les points d'intérêt POIs à l'intérieur de la scène, on crée une carte qui représente le champ de vision de la caméra, en utilisant deux approches. La première approche, la « *carte de chaleur* » (voir annexe D)est une approche simple fondée sur la détection de régions qui appartiennent au premier plan et l'accumulation du nombre de fois qu'une région détectée traverse un pixel de l'image. La deuxième approche, la « *carte d'occupation* » est une approche plus robuste basée sur les trajectoires reconstruites à partir des données discrètes (les coordonnées des centres de gravité) des personnes détectées. Nous avons appliqué cette méthode sur des données issues d'une caméra 2D présenté dans l'annexe E.

La deuxième approche a comme entrée les trajectoires $\tau^j(P^j) \in T$ générées par le bloc de suivi de la chaîne de traitement en ligne (section 3.2.1). Les trajectoires sont représentées par l'historique de positions de détections P^j d'une manière discrète. Pour cette raison, on doit estimer les points manquants qui appartiennent aux segments entre les positions de τ^j. Ceci nous permet de reconstruire les trajectoires quelle que soit la fréquence d'acquisition ou celle des personnes observées. Cependant, cette méthode impose une perte de détail (au niveau des pixels) de la forme de la personne détectée puisque la personne est réduite à son centre de gravité (un seul pixel), contrairement à la méthode de la *carte de chaleur*. En conséquence, les trajectoires ont très peu de chances de se recouper avec d'autres (comme le montre la figure 5.2) pour identifier des zones très utilisées ou des points d'intérêt.

En conséquence, on a besoin de réduire l'espace de notre problème. Dans le travail de [BK15] en raison de la taille de l'espace observé, les analyses spatio-temporelles ont été faites en divisant

l'espace en une grille de 50×50 centimètres. De la même manière, on définit la carte d'occupation O_c comme une matrice qui représente l'espace dans la scène à taille réduite qui accumule les passages des trajectoires. Soit $\lambda(\lambda_x, \lambda_y)$ le facteur de réduction anisotropique d'échelle suivant les axes X et Y de l'image d'entrée originale I de résolution $I_x \times I_y$ pixels de sorte que la taille moyenne d'une personne soit représentée par un pixel de la carte O_c. Ainsi, la taille résultant de O_c est $\lambda_x I_x \times \lambda_y I_y$. De plus, λ_x et λ_y doivent être diviseurs respectivement de I_x et I_y, pour éviter des problèmes d'arrondi.

Par ailleurs, les valeurs de λ dépendent de la hauteur d'installation de la caméra. Un des avantages de cette approche est la réduction de la taille des données à traiter, la taille en pixels de $|O_c| < |I|$. De plus, la carte O_c est plus petite que la taille de $|I|$.

Pour calculer O_c, on transforme toutes les trajectoires T dans l'espace réduit par λ. On incrémente ensuite la valeur O_c dans chaque point de chacune des trajectoires transformées. De plus, on estime tous les points des segments de chaque trajectoire en utilisant le traçage de ligne dans une image pixelique. Puis, on incrémente la valeur de O_c dans les points estimés. Finalement, les valeurs de O_c sont exprimées par rapport au temps d'observation. On formalise l'estimation de O_c de la manière suivante :

Pour calculer O_c à partir des trajectoires $\tau^j \in T$ nous devons :

1) transformer $\tau^j(P^j)$ en $\tau_o^j(P_o^j)$ où $\tau_o^j(P_o^j)$ est la représentation d'une trajectoire $\tau^j(P^j)$ dans l'espace D_c.

   - Chaque point $p^i \in P^j$ est transformé en $p_o^i \in P_o^j$, en utilisant la fonction $\theta(p^j(x^j, y^j), \lambda) = p_o^j(\lambda_x x^j, \lambda_y y^j)$.

2) Incrémenter la valeur $O_c(p_o^j)$, $\forall p_o^j \in P_o$.

3) Calculer tous les points manquants $P^n \in P_u$ entre deux points de détection consécutifs $(p_o^i, p_o^{i+1}) \in P_o^j$ de la trajectoire τ_o^j.

   - Pour compléter τ_o entre deux points de détection consécutifs $(p_o^j, p_o^{j+1}) \in P_o$, on doit trouver $N = max(|p_x^{j+1} - p_x^j - 1|, |p_y j + 1 - p_y^j - 1|)$ points manquants p^n en utilisant l'équation vectorielle de la ligne tel que $p^n = [p^j + (p^{n+1} - p^n) \cdot n/(N+1)]$ où $n = 1, \ldots, N$ et $[x]$ représente la partie entière de x.
   - Incrémenter la valeur $O_c(p^n)$, $\forall p^n \in P_u \subseteq D_c$.
   - Une fois que toutes les trajectoires sont évaluées, on divise chaque cellule de O_c par le temps total d'observation en obtenant les pourcentages d'occupation de l'espace de la scène.

Le résultat de ce procédé est montré dans la figure 5.5. Dans la figure 5.5a, on a choisi de rajouter une $3^{ème}$ dimension qui illustre l'occupation d'un espace pendant le temps d'évaluation. Les endroits avec la fréquentation la plus élevée représentent les espaces où les personnes sont restées le plus de temps. Par analogie, les endroits les plus occupés sur la figure 5.5b sont de couleur rouge, qui représente la valeur la plus élevée sur l'échelle de coloris, à droite de l'image. Puisque les endroits les plus occupés sont, dans certains cas, vers le coin inférieur de l'image et peuvent ainsi occulter le reste de l'image, on préfère la représentation 2D de la carte d'occupation, qui peut être interprétée facilement avec l'échelle, une fois que le concept d'occupation est compris.

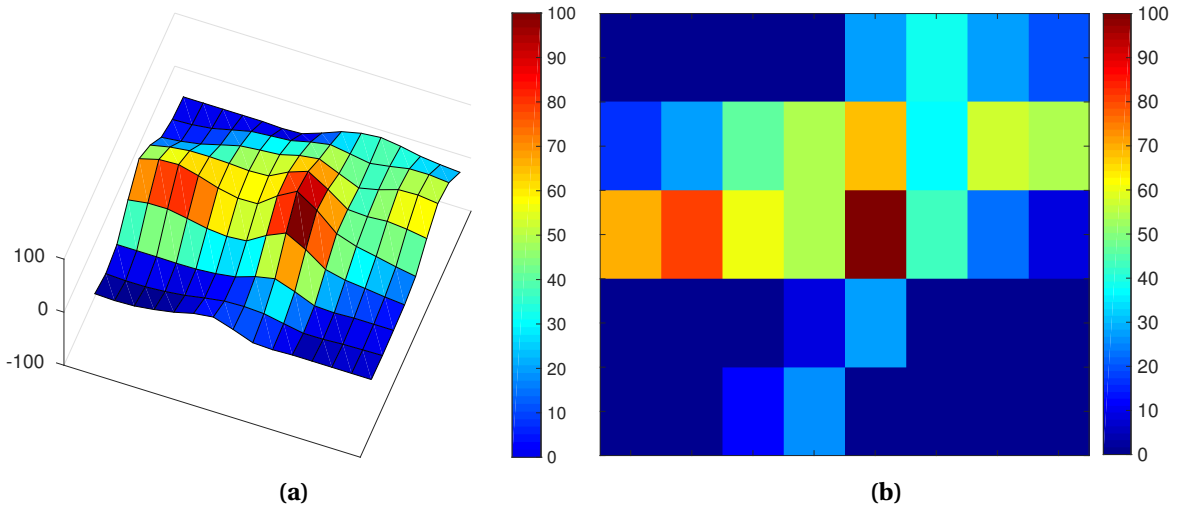


FIGURE 5.5 – Représentation de la carte d'occupation. a) Représentation 3D de la carte d'occupation. b) Représentation 2D de la carte d'occupation.

De plus, on ajoute un filtre de saturation pour éviter des zones avec peu de trajectoires de personnes et un filtre de saturation pour éviter des anomalies de haute fréquentation dans un seul endroit (Fig. 5.5) ou les zones bruitées dues aux fausses détections (surtout dans les systèmes utilisant des caméras 2D). Pour ce faire, on définit deux seuils de filtrage exprimés en pourcentage de temps d'occupation O_{min} et O_{max} avec lesquelles O_c va être filtrée, obtenant O_f :

$$O_f(x,y) = \begin{cases} 0 & O_m(x,y) < O_{min} \\ O_{max} & O_m(x,y) > O_{max} \\ O_c(x,y) & \text{autrement} \end{cases} \tag{5.1}$$

Dans la figure 5.6, on présente des exemples du filtrage de O_c, en utilisant l'équation 5.1, sur un ensemble de 1090 trajectoires à l'intérieur d'un centre commerciale [BK15]. Cette figure est composée par quatre couples d'images. Chaque couple représente une carte d'occupation filtrée O_f en 3D et 2D avec différentes valeurs de seuillages. Le premier couple d'images représente une carte d'occupation sans filtrage (O_{min} = 0% et O_{max} = 100%). Dû à la grande concentration des trajectoires dans certains points (les pics rouges), le reste de la carte ne reflète pas l'occupation de la scène dans la série chronologique. Dans le deuxième couple, on observe la même source des données mais avec un filtrage entre 0% et 50% du temps d'occupation. Le troisième couple représente la carte d'occupation filtrée entre 0% et 10%. Ce dernier filtrage est configuré automatiquement entre la valeur minimale et la moyenne des valeurs de la carte en question. Dans l'exemple de la figure 5.6, on commence à observer la dynamique d'occupation qui traverse l'image horizontalement (voir parties 5.6e et 5.6f) mais sans différencier la dynamique de ce traçage. Dans le dernier couple, le même ensemble de trajectoires filtrées entre les seuils de 0 % et 5 % nous permet d'observer plus clairement l'utilisation de l'espace et les différentes dynamiques, comme une chaîne de montagnes, sans perdre les points de fréquentation maximale trouvés dans le premier couple.

Une fois qu'on a obtenu notre carte d'occupation filtrée, on est capable d'identifier visuellement les points les plus fréquents et les points d'intérêt POIpls dans l'image. Ces POIss correspondent aux maximas locaux de O_c. Pour déterminer les points maximaux de manière automatique, nous proposons d'utiliser l'opération morphologique de dilatation $Dil(O_c)$ avec un élément structurant rectangulaire de taille 3×3 pixels sur la carte d'occupation. Nous soustrayons ensuite O_c tel que $Diff(O_c)$=Dil(O_c)-O_c et obtenons finalement, on obtient les maxima locaux $M_l(O_c)$ sur les pixels où la différence $Diff(O_c)$ est égale à 0 et la valeur initiale de O_c est différente de 0.

$$M_l(x,y) = \begin{cases} 1 & Diff(x,y) = 0 \wedge |Diff(x,y) - Oc(x,y)| > 0 \\ 0 & \text{autrement} \end{cases} \tag{5.2}$$

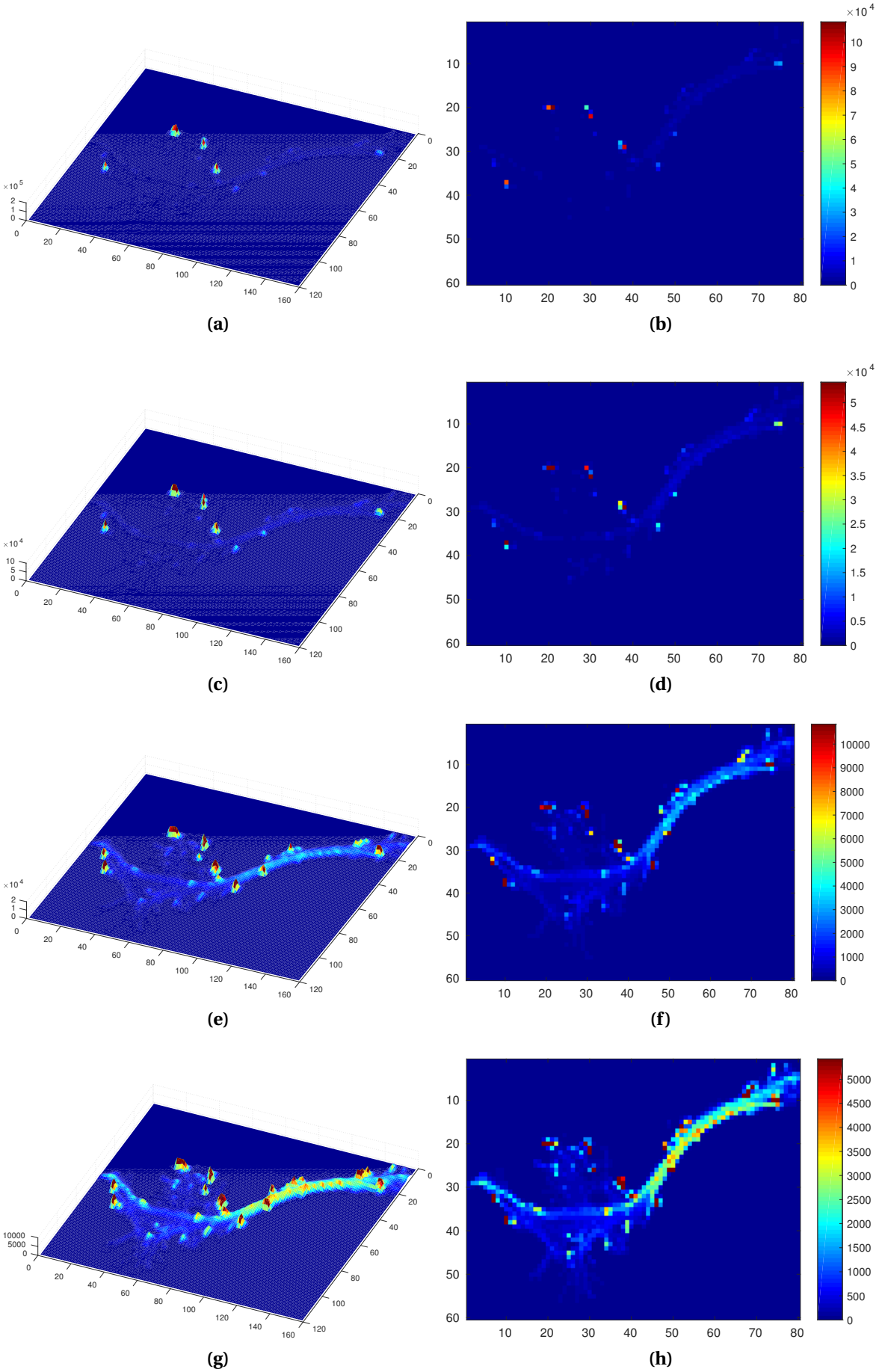
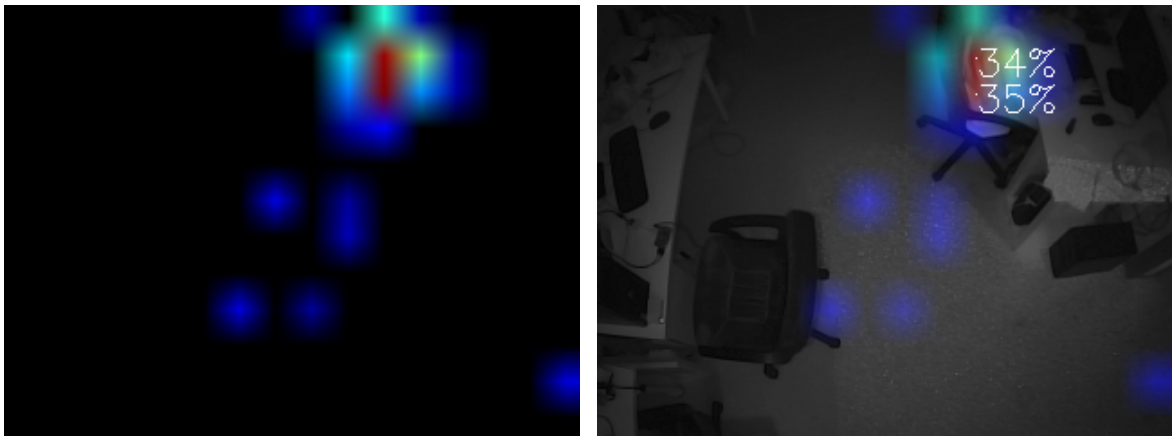


FIGURE 5.6 – Représentation du filtrage de la carte d'occupation. Les images a, c, e et g sont une représentation 3D de la carte d'occupation. Les images b, d, f et h sont la représentation 2D de la carte d'occupation.

On peut définir les maximas locaux comme $p_m^j \in \mathrm{P}_m \subseteq \mathrm{D}_c$, $\mathrm{P}_m := \{x, y \in \mathbb{N} : 1 < x < \lambda_x \mathrm{I}_x \bigwedge 1 < y < \lambda_y \mathrm{I}_y | \mathrm{M}_l(x, y) = 1\}$. Cependant, pour donner du sens à cette valeur, il faut lui donner un repère de temps. On calcule donc la durée totale de $t(\mathrm{T})$ et on normalise la valeur de $\mathrm{O}_c(p_m^j)/t(\mathrm{T})$, en obtenant le pourcentage de temps qu'un point p_m (maximum local) dans O_c a été occupé. On peut filtrer aussi les points P_m en éliminant les points en dessous d'un seuil exprimé en pourcentage de temps d'occupation minimal Op_{min} par rapport au temps maximal d'occupation t_{max}. La valeur t_{max} correspond à la valeur maximale entre tous les points P_m.

On ajoute un deuxième type de filtrage de saturation que l'on peut utiliser pour obtenir des points d'intérêt secondaires qui ne sont pas pris en compte dans une première étape, pour faire une analyse plus détaillée de l'utilisation de l'espace. Par exemple, il s'agit de déterminer les zones secondaires d'attention des personnes pour trouver des points optimaux pour distribuer la charge des points principaux à ces endroits. Ces POIs secondaires ne sont pas détectés initialement comme points importants, du fait que la valeur des points maximas est très élevée en comparaison avec la valeur de ces points. Pour ce faire, on définit un seuil de filtrage exprimé en pourcentage de temps d'occupation Op_{max}.

Par exemple, dans l'image 5.7, on observe dans a) des points secondaires proches du point le plus intense (rouge). Ces points ne sont pas pris en compte dans les P_m mais mériteraient d'être analysés. Dans b), on superpose l'image d'entrée, O_c et P_m, pour fournir l'information pertinente (lieu et temps d'occupation) pour l'utilisateur.



(a) **(b)**

FIGURE 5.7 – a) Carte occupation O_c d'exemple. b) Visualisation complète de la O_c, les maximas locaux et l'image d'entrée en échelle de gris. Ces images sont prises en utilisant une caméra intelligente monoscopique qui travaille en faible résolution.

Cette méthode nous permet d'obtenir les points d'intérêt automatiquement à l'intérieur de la scène pour mieux comprendre l'utilisation de l'espace. Ces points représentent les lieux les plus fréquentés par les personnes observées. Ces POIs nous donnent des clés pour identifier l'endroit où l'on doit placer les zones d'intérêt pour mieux exploiter cette fonctionnalité (section précédente). Cependant, on pourra améliorer la compréhension de la scène si l'on peut identifier les objets qui sont dans ces points d'intérêt. Comment pouvons-nous exploiter plus efficacement ces points d'intérêt ? On peut faire des annotations sur les images par rapport à des objets qui sont dans la scène. Pourtant, les produits dans les magasins changent continuellement. La solution pour trouver un moyen de reconnaître les objets de la scène sans impacter la performance de notre système est donc d'utiliser des services de classification des objets à distance. Ainsi, on peut utiliser une méthode externe e.g. [web17b], pour écarter cette difficulté du reste de l'analyse.

Détection des points d'entrée et sortie

En utilisant la même méthode pour représenter la scène qui a permis de générer la carte d'occupation, on crée deux cartes différentes : la carte d'entrées et la carte de sorties. Dans la première carte, on place seulement les premiers points des trajectoires. Dans la deuxième, on place seulement les derniers points des trajectoires. De cette manière, on trouve facilement les points d'entrées et sorties dans l'espace observé.

De la même manière que la carte d'occupation, on définit les cartes d'entrées et de sorties O_e et respectivement O_s comme des matrices de la même taille que la carte d'occupation, qui accumulent les débuts et les fins de toutes les trajectoires détectées. Pour illustrer ce procédé, on utilise un ensemble de 5 trajectoires (Fig.5.8a) et on calcule la carte d'entrées O_e (Fig. 5.8b et 5.8c) et la carte de sorties O_s (Fig. 5.8d et Fig. 5.8e).

Dans la figure 5.8, on observe dans la première image 5 trajectoires qui traversent la scène où le début de la trajectoire est en bleu et la fin en violet. La deuxième et troisième image sont une représentation 3D et 2D, respectivement, de la carte d'entrées de l'ensemble des trajectoires à évaluer. Dans la deuxième image, on utilise des barres 3D où chaque barre représente la surface d'une partie de la scène observée et sa hauteur représente le pourcentage de trajectoires qui ont commencé dans cet espace. La troisième image est une vision plane de la représentation en barres 3D. Chaque espace avec une valeur est dénommé « *point d'accès* ». La quatrième et cinquième représentent la carte de sorties de l'ensemble évalué. Chaque représentation de l'espace a comme valeur le pourcentage de trajectoires qui finissent à l'intérieur de cet espace et détermine les points de sorties, de la même manière que les images précédentes. Dans cet exemple simple, on peut intuitivement déterminer les directions des trajectoires illustrées dans l'image 5.8a qui partent du haut à droite (Fig. 5.8c) et se terminent en bas à gauche (Fig. 5.8e) en regardant l'image 5.8a et les cartes d'entrées et sorties.

5.2.2 Analyse des trajectoires

Dans cette section, on présente les méthodes pour analyser les trajectoires à travers la segmentation du flux comportemental, la validation à distance et la visualisation dynamique des trajectoires.

Segmentation d'un flux comportemental

Une des principales tâches de l'étude du comportement humain est de trouver les mouvements communs des individus. Pour cette raison, la segmentation de flux comportemental est une approche très importante qui cherche à grouper (segmentation) les différentes trajectoires des personnes qui ont un comportement similaire (flux). La difficulté qu'impose cette tâche est de trouver le nombre de groupes [BK15] et de trouver les propriétés pertinentes des trajectoires [Fer15], parmi les dimensions (propriété de haute dimensionnalité), pour les grouper dans une *série chronologique*. Dans cette section, nous proposons une méthode de segmentation de flux comportemental et nous décrivons les variables (dimensions) qui ont le plus d'influence sur cette segmentation.

Le processus de segmentation de flux comportemental des personnes en mouvement dans une zone d'observation que nous proposons est basé sur le travail de groupement de trajectoires de Ferreira et al. [FKSS13, Fer15]. Ces auteurs proposent une méthode qui utilise des champs vectoriels pour caractériser un groupe de trajectoires, caractérisées dans le temps et l'espace [FKSS13]. Dans cette section, nous nous référerons à cet algorithme en tant que ***Vector Field k-means*** (VFKM). Par ailleurs, Ferreira, ajoute l'utilisation d'attributs de la trajectoire comme paramètres de groupement [Fer15]. Dans cette section, nous allons nous référer à ce deuxième algorithme en tant que ***Attributes Field k-means*** (AFKM). Le but de ces méthodes est d'assigner les trajectoires à un nombre K de clusters. Dans notre contexte, chaque cluster présente un flux comportemental (c'est-à-dire, un groupe de trajectoires similaires). Cette méthode est composée de quatre étapes importantes : le pavage de trajectoires, l'initialisation de clusters, l'estimation du champ vectoriel et l'assignation des trajectoires au champ vectoriel. Les détails concernant cette partie d'algorithmes VFKM et AFKM sont donnés dans l'annexe F.

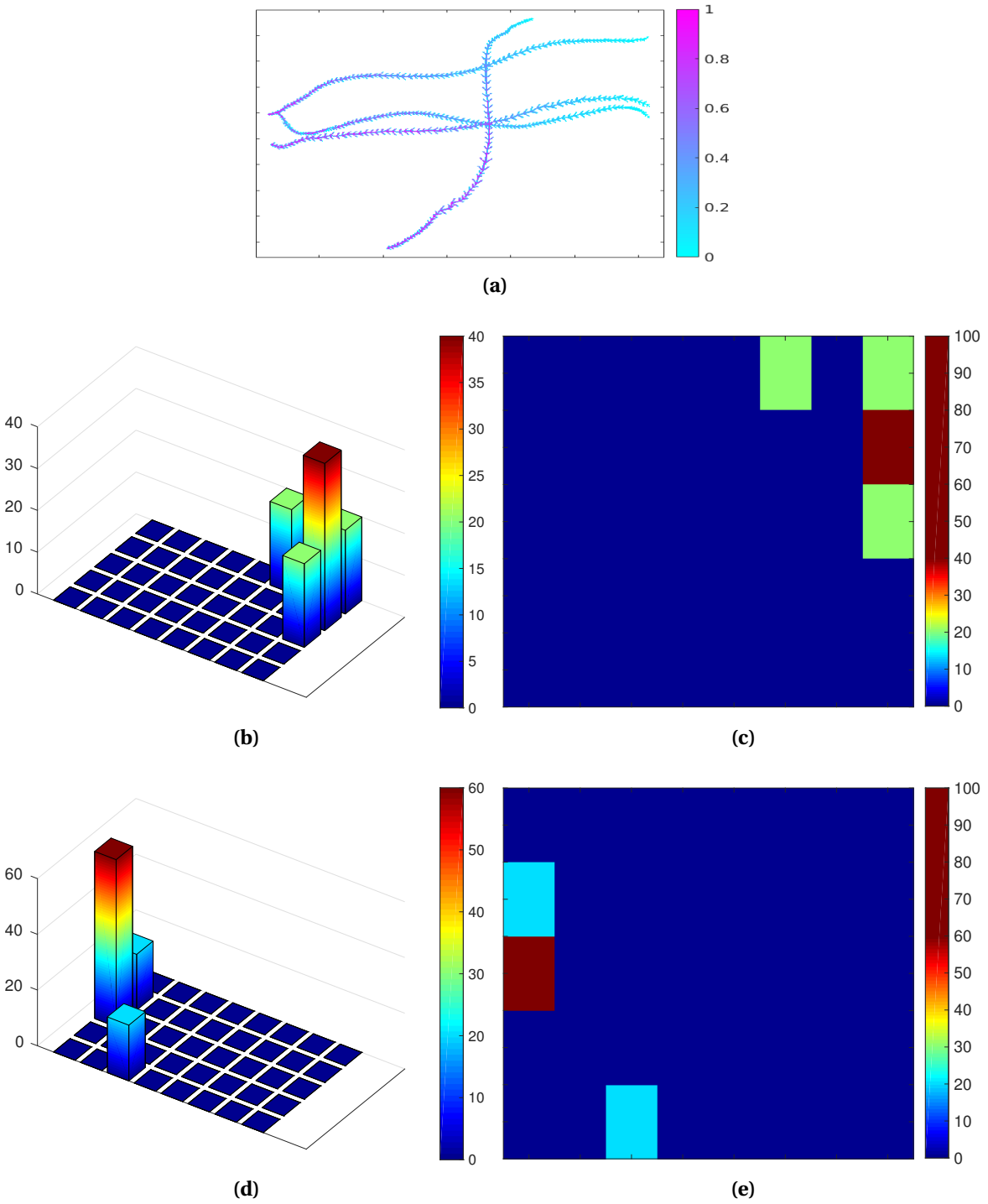


FIGURE 5.8 – Représentation des points d'entrée et de sorties de la scène observée à partir de 5 trajectoires. a) Représentation de 5 trajectoires. b) Représentation 3D des points d'entrée. c) Représentation 2D des points d'entrée. d) Représentation 3D des points de sortie. e) Représentation 2D des points de sortie.

Concernant ces algorithmes, ils sont basés sur l'algorithme K-means, donc ils dépendent fortement de l'initialisation d'assignation des trajectoires et de la quantité des clusters [FKSS13]. De la même manière, déterminer le nombre correct de clusters dépend de l'utilisation. Par exemple, dans une caméra située face à la porte d'une boutique, une segmentation en deux clusters sert à différencier les entrées des sorties. Cependant, toutes les autres trajectoires (e.g. une personne qui passe à côté de la porte à l'intérieur de la boutique sans sortir) sont aussi ajoutées à ces deux groupes. Un exemple d'utilisation de la segmentation de flux et du choix de la valeur de K clusters sera donné dans la section « résultats » de ce chapitre.

L'avantage de ces algorithmes est de représenter les centres des clusters comme un champ vectoriel et son utilisation pour évaluer la similarité entre les trajectoires en éliminant fortement le calcul de la métrique des trajectoires et de son barycentre [Fer15]. De plus, les champs vectoriels résultants représentent le mouvement global de chaque cluster, le flux comportemental.

La mise en œuvre de ces algorithmes est itérative. A chaque itération, on calcule les nouveaux champs vectoriels, puis on assigne les trajectoires. On itère jusqu'à la convergence de l'algorithme, c'est-à-dire quand le nombre de trajectoires qui ont changé d'assignation de cluster est égal à 0.

Une des contraintes de l'utilisation de ces algorithmes dans notre système est que l'on a besoin de multiples trajectoires pour trouver un flux comportemental pertinent, en conséquence il n'est pas possible d'estimer la segmentation en temps-réel. Pour dépasser cette limitation, nous avons pris le problème de deux façons. D'abord, on améliore la performance en parallélisant l'algorithme pour l'exécuter à la fin de la journée quand les boutiques sont fermées. Ensuite, une fois que nous avons des flux définis (de la veille), on calcule simplement, pour chaque nouvelle trajectoire en temps-réel, l'assignation du cluster (équation F.2). Cette dernière astuce nous permet de réduire les itérations et d'arriver à la convergence de la minimisation en améliorant l'initialisation de trajectoires, de manière à se rapprocher de la solution.

Une autre application explorée par cette méthode est la distinction des comportements spécifiques que l'on cherche à éviter pour ensuite le transférer au module statistique de l'entreprise Shopline (voir section 5.1.3). Par exemple, dans le cas des boutiques, on cherche à éviter des personnes qui s'approchent de la porte d'un magasin, hésitent à entrer et finalement s'éloignent. Une telle attitude peut affecter l'affluence à la porte du magasin : ces personnes peuvent en décourager d'autres [Ban65] pour y rentrer et ainsi créer une congestion dans les points d'accès au magasin. Le fait d'identifier ces trajectoires nous permettra d'extraire plus d'informations statistiques déjà gérées par l'entreprise Shopline (voir section 5.1.3) telles que sa périodicité, le jour de la semaine, la relation avec la météo, etc.

Dans l'image 5.9, on observe la segmentation d'un ensemble des trajectoires dans 4 directions. Les paires d'images (5.9a ,5.9e) (5.9b, 5.9f), (5.9c,5.9g) et (5.9d, 5.9h) correspondent à un sous-ensemble de trajectoires et à son flux comportemental. Chaque flux est représenté par un champ vectoriel avec un motif indiquant une direction globale sur la scène. De la même manière, on observe que le sous-ensemble de trajectoires suit la même direction. Les trajectoires sont représentées par une séquence de flèches où les débuts de la trajectoire sont en bleu foncé et finissent par la couleur rouge. De plus, une flèche plus longue représente un mouvement plus rapide. Le premier couple représente les trajectoires de droite à gauche, le deuxième de haut en bas, le troisième de gauche à droite et le dernier de bas en haut. De plus, on peut observer comment les champs vectoriels représentent le mouvement global des trajectoires du sous-ensemble.

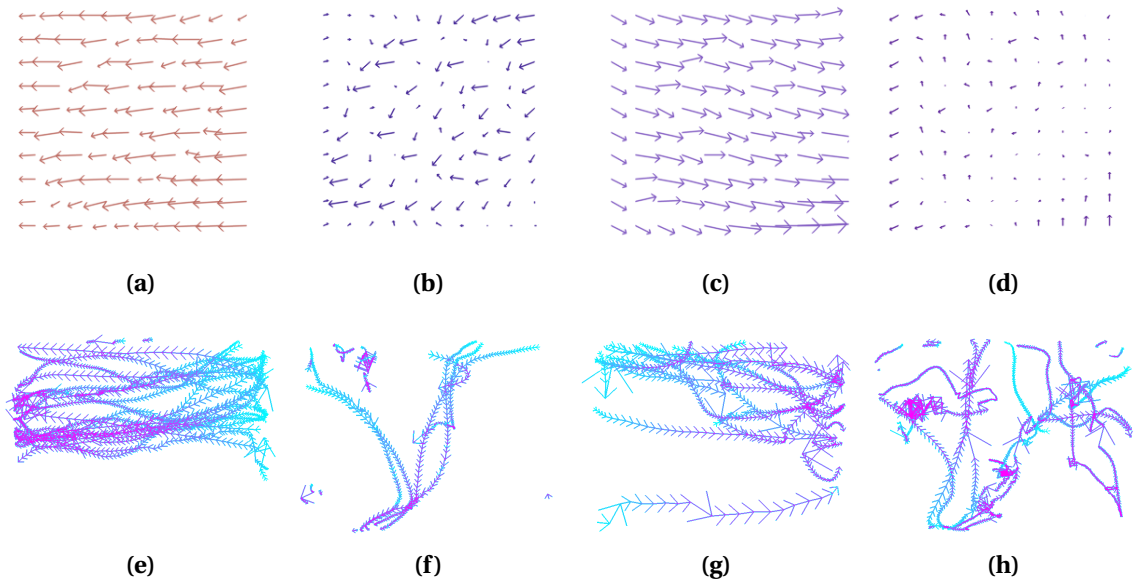


FIGURE 5.9 – Exemple des segmentations de flux comportemental en 4 clusters. La première ligne est composée par la représentation des champs vectoriels de chaque cluster. La deuxième est composée de la représentation des trajectoires qui appartiennent à chaque cluster.

La validation et la représentation dynamique des trajectoires

La représentation dynamique des données acquises est un outil important de validation et d'évaluation du bon fonctionnement du système. On décrit deux méthodes de représentation dynamique des données de différente nature : les images et les historiques des positions des personnes. Ces méthodes servent à valider (à distance) dans les cas de contestation des utilisateurs sur le comptage. Elles servent aussi à comprendre l'environnement où la caméra est installée et de vérifier que celle-ci se trouve dans des conditions correctes (installée en position zénithale et sans grands obstacles dans le champ de vision) pour son fonctionnement. Finalement, elles servent à avoir un outil de visualisation des endroits observés sous forme de vidéo ou de représentation virtuelle (voir figure 5.10). Une des applications de ces méthodes dans le contexte industriel est la possibilité de valider l'installation et la maintenance à distance.

Dans la première méthode, on sauvegarde les images acquises sur de courtes périodes pour les enregistrer sur un serveur via FTP. Pour ce faire, on a étudié les différents types de compression des données auxquelles on a eu accès. La table 5.1 montre les différents codecs et leur taux de compression sur deux vidéos de test.

TABLEAU 5.1 – Taux de compression vidéo 1

Codec	MegaBytes	Compression rate
FLV1	3,34	11,51
H263	3,38	11,66
motion-jpeg	3,86	13,30
MPEG-1	3,53	12,18
MPEG-4.2	3,17	10,92
MPEG-4.3	3,24	11,19
MPEG-4	3,18	10,97
Video 1	28,98	100

On sélectionne le codec H263 pour ses bons résultats en comparaison avec les autres, mais aussi parce qu'il a l'accès à la librairie open source « avcodec ». Cependant, la difficulté dans l'utilisation de ces outils est le détriment de la qualité au moment de la compression des images pour les envoyer sur le réseau. Cette dégradation de la qualité des images peut apporter des erreurs de comptage dans certaines conditions de lumière difficiles à reproduire.

TABLEAU 5.2 – Taux de compression vidéo 2

Codec	MegaBytes	Compression rate
FLV1	119,32	10,77
H263	120,43	10,87
motion-jpeg	119,81	10,82
MPEG-1	112,63	10,17
MPEG-4.2	109,00	9,84
MPEG-4.3	103,20	9,32
MPEG-4	109,49	9,88
Video 2	1107,82	100

Dans la deuxième méthode, on envoie les coordonnées des personnes détectées au serveur de rassemblement pour les enregistrer en nous permettant de reproduire virtuellement les passages des personnes par le champ de vision de la caméra. La figure 5.10 est un exemple d'une séquence de vidéo pour représenter virtuellement les trajectoires de personnes.

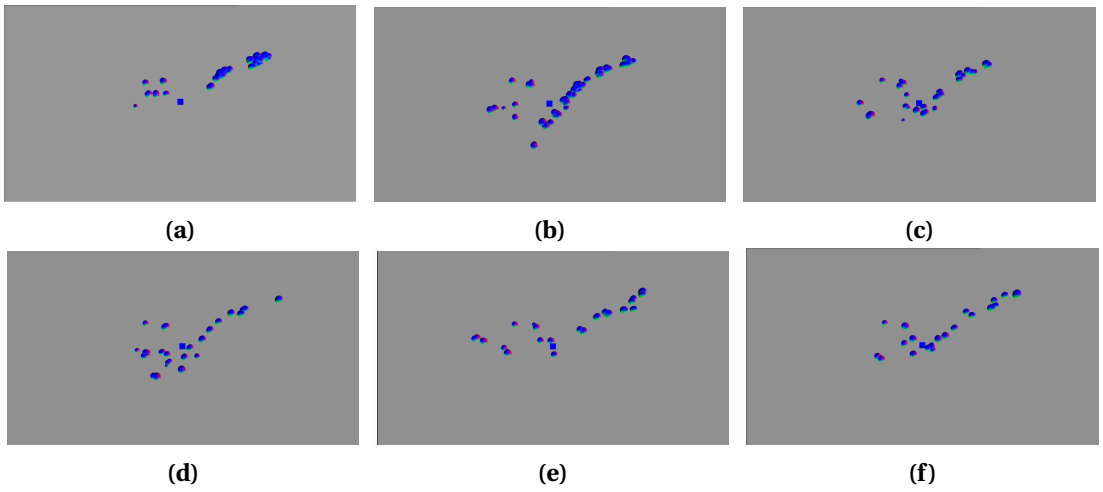


FIGURE 5.10 – Séquence d'images consécutives virtuelles en 3D des coordonnées de personnes.

Cette deuxième méthode permet d'avoir une visualisation dynamique des données. De plus, le volume de données à sauvegarder diminue énormément, c'est-à-dire que l'on passe de plusieurs giga-octets de la séquence d'images à quelques méga-octets des coordonnées des personnes détectées. La visualisation d'un groupe de lignes statiques proposée dans la méthode de segmentation du flux manque de dynamisme réel du comportement des personnes apporté par cette fonctionnalité. En utilisant cette visualisation active et la segmentation du flux comportemental, on permet à l'utilisateur d'obtenir une meilleure compréhension de la dynamique de la scène.

5.3 Résultats

Dans cette section, on présente les résultats en utilisant les méthodes proposées dans la section précédente. On évalue ces méthodes sur les données produites par notre système et celles produites par d'autres systèmes, dans le cadre d'autres travaux.

5.3.1 Méthode d'évaluation

A partir des jeux de données présentés dans la section suivante, on évalue les méthodes d'analyses comportementales des personnes en mouvement proposées dans ce chapitre. Dans nos expériences, on suppose que les systèmes de suivi des personnes qui fournissent les trajectoires sont déjà validés et que les trajectoires fournies représentent, de manière assez précise, les mouvements des personnes observées.

De plus, la méthode de détection de personnes en temps-réel dans les zones d'intérêt et la méthode de la validation à distance sont exécutées sur les caméras intelligentes directement. Le reste des traitements sont exécutées de manière centralisée pour leur évaluation. Ces dernières consultent les données stockées de trajectoires pour générer les analyses. Malheureusement, comme chaque jeu de données vient de sources différentes, on a d'abord homogénéisé les formats des trajectoires pour ensuite faire les analyses. Finalement, les résultats des analyses sont enregistrés pour la visualisation en dernière étape.

5.3.2 Jeu de données

Notre jeu de données est composé de *quatre séquences chronologiques* de *trois sources* différentes de génération de trajectoires. Chaque source est composée d'un ensemble de trajectoires que l'on utilise pour évaluer nos méthodes. On utilise plusieurs sources de données pour assurer une diversité et donc l'adaptabilité de nos méthodes à différents types de capteurs (3D active et passive) et multi-caméra. La première source est une caméra stéréoscopique (3D passive) où les personnes traversent la scène majoritairement dans deux sens, entrée et sortie. La deuxième source provient d'une caméra Asus Xtion-Pro (3D active) à l'entrée d'un laboratoire. La troisième est une des séries chronologiques prises par [BK15] avec une approche multi-caméra dans un centre commercial.

Nous décrivons ci-dessous nos jeux de données, en indiquant le nombre de trajectoires, le système avec lequel les trajectoires ont été prises et la résolution de la vidéo. On commence par l'ensemble des tests qui servent à valider le fonctionnement de chaque méthode d'une manière simple.

JDT

Cette série chronologique composée de 5 trajectoires (Fig. 5.11a) est utilisée pour valider facilement le fonctionnement de nos méthodes. Ces trajectoires ont été prises avec notre caméra intelligente et sont les premières 5 trajectoires du JD02.

JD01

Notre premier jeu de données est identifié comme JD01 (Fig. 5.2). Il est composé de 76 trajectoires sur une séquence vidéo de $320 \times 240 pixels$ de résolution prises par une caméra intelligente stéréoscopique prototype développée par Shopline avec une précision de comptage à 80 %. Ce jeu de données contient les trajectoires de personnes de différents sexes et de différentes tailles, qui traversent la scène majoritairement dans le sens vertical (vers le bas et vers le haut). Ce jeu de données représente le comportement typique d'une entrée d'un magasin, ce qui est le cas habituel traité par l'entreprise Shopline.

JD02

Notre deuxième jeu de données est identifié comme JD02. Il est composé de 87 trajectoires sur une séquence vidéo de 640×480 pixels de résolution prises par la caméra intelligente décrite au chapitre 3. La majorité des mouvements dans ces jeux des données se font dans les 2 sens verticaux et le 2 sens horizontaux (vers la droite et vers la gauche). Ce jeu de données a été créé pour évaluer la segmentation et le suivi des personnes. Il contient les différents cas difficiles pour la segmentation décris dans le chapitre 3, en apportant des éléments intéressants à analyser dans les résultats d'évaluation des méthodes.

JD03

Notre troisième jeu de données est identifié comme JD03. Il est composé de 1090 trajectoires prises dans un centre commercial d'une surface de 900 m_2. Les trajectoires ont été extraites par le système de Brščić and Kanda [BK15], travail avec lequel on a comparé certaines de nos méthodes. Ce système utilise différent types de capteurs, pourtant on ne parle pas de résolution, mais de la zone que l'ensemble des trajectoires occupe dans leur système de coordonnées globales (-40863 <x<47973, -27243<y<24237). On utilise ce jeu de données pour démontrer la faisabilité d'utilisation de nos méthodes dans de grands espaces publics. Les expériences faites avec notre système dans le chapitre précédent n'ont pas la même quantité de données que le JD03.

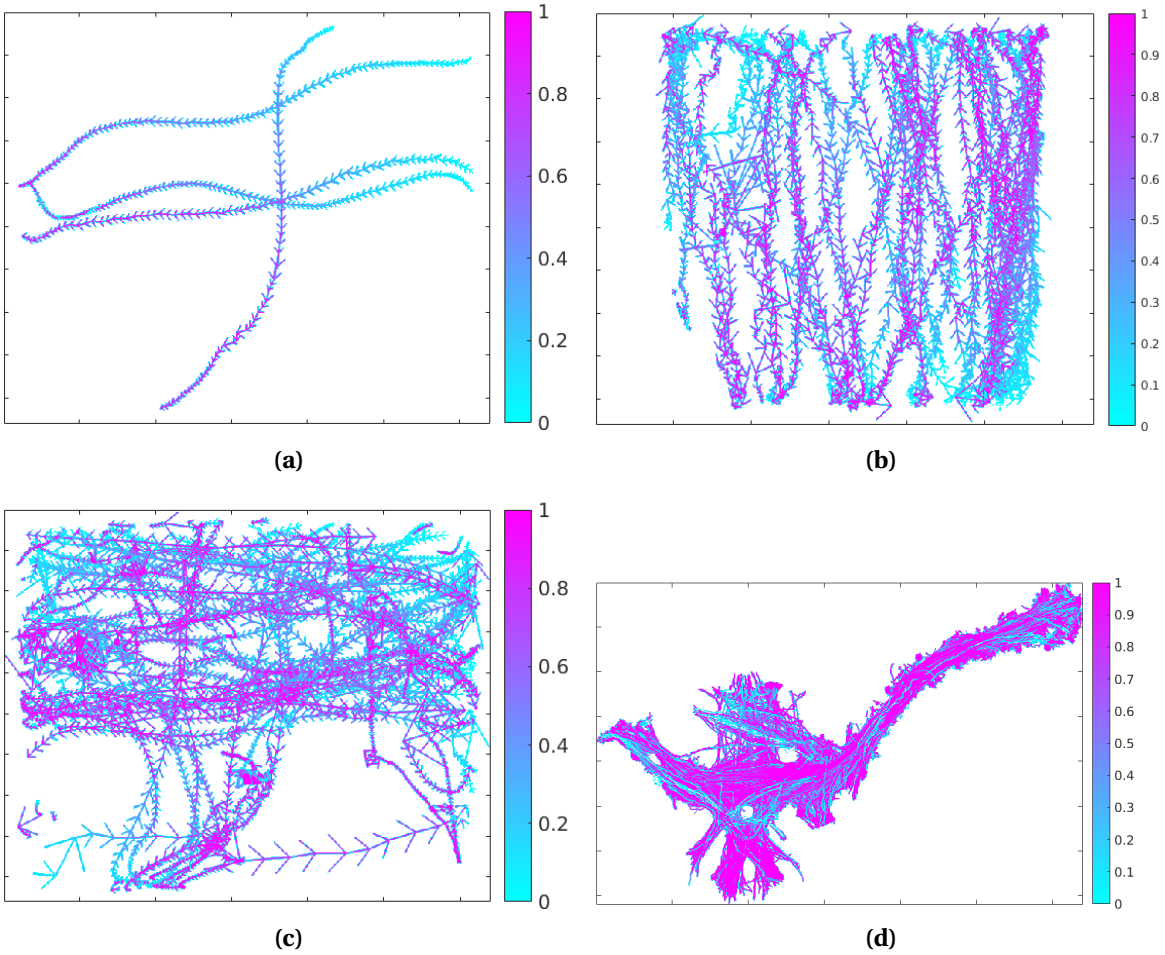


FIGURE 5.11 – Représentation de jeux de données. Le début de trajectoire est en bleu cyan et la fin est en magenta.

5.3.3 Analyse de l'utilisation de l'espace

On présente dans cette section les résultats, l'analyse et l'application des méthodes pour extraire les cartes d'occupation, d'entrées et de sorties. L'objectif principal de cette étude est de trouver des représentations pratiques et significatives pour évaluer l'utilisation de la scène. Dans cette section, on affiche d'abord la précision de nos résultats, mais en même temps, on évalue la pertinence et la signification des représentations utilisées.

La détection de personnes en temps-réel dans les zones d'intérêt

On met en œuvre cette méthode sur notre caméra intelligente autonome. Chaque caméra de notre système est capable d'évaluer plusieurs zones dans l'image en temps-réel. La pertinence des alarmes produites par cette méthode dépend de la précision de l'algorithme de suivi et de l'évaluation de conditions comme la quantité de personnes à l'intérieur de la zone ou le temps de séjour d'une personne. Bien que n'ayant pas développé de protocole de test, nous avons réalisé une vérification visuelle sur le JD02 avec une configuration de deux zones paramétrées de la manière suivante : le temps maximal d'attente de 10 secondes et un nombre maximal de personnes par zone, égal à 1. Nous avons vérifié visuellement que le fonctionnement global des alarmes et celui des compteurs sont corrects. Finalement, seules les erreurs liées à la perte de l'étiquette de suivi subsistent (dans les cas où la segmentation de personnes a échoué), mais aucune erreur n'est liée à notre méthode.

Cette méthode est censée fonctionner en temps-réel, c'est la raison pour laquelle nous ne l'avons pas testée sur les autres jeux de données, puisqu'il ne nous est pas possible de l'ajouter sur les autres caméras développées en dehors de ce travail.

Cette méthode est censée être utilisée sur les files d'attente des magasins, pour estimer les temps d'attente des clients au moment de payer, automatiser l'appel à des caissiers et évaluer globalement la performance du personnel dans le service aux points de paiement. Bien que l'algorithme de comptage soit déjà élaboré, sa mise en œuvre dans le système n'est pas encore faite, il est encore au stade de prototypage.

Carte d'occupation

On a évalué cette méthode dans tous nos jeux de données. Les paramètres de cette méthode sont le ***facteur de réduction*** anisotropique d'échelle $\lambda(\lambda_x, \lambda_y)$ pour les axes X et Y, les ***seuils de temps d'utilisation*** minimum et maximum. Pour chaque jeu de données, on décrit les paramètres utilisés, avant de présenter les résultats et leur analyse.

Facteur de réduction

Comme on l'a expliqué dans la section précédente, le *facteur de réduction* facilite l'analyse de grands espaces en créant des sous-zones de la taille d'une personne. Ce facteur sert également à faire des analyses plus ou moins détaillées de l'espace, selon les besoins de l'utilisateur. Par exemple, si l'on analyse l'entrée d'un magasin (JD01) la partition de l'espace par rapport à la taille d'une personne est pertinente. Cependant, si on fait l'analyse d'un centre commercial (JD03), on est davantage intéressé par un espace plus important (par exemple, la taille de la vitrine d'un magasin). Ainsi, la réduction à différentes échelles permet de s'adapter à la variété des besoins d'analyse de l'espace. De plus, cette démarche s'applique aussi aux cartes d'entrées et de sorties.

La figure 5.12 présente la carte d'occupation en 3D avec un facteur d'occupation de $16 \times 12 pixels$ sur le jeu de données JD01. Les espaces non utilisés sont représentés par une surface de couleur bleue foncée de hauteur 0, plus facilement identifiable au bord de l'image. Dans cette figure, on observe la formation des vallées de hauteur 0 sur des espaces où les personnes ont traversé la scène. Cet effet est dû à l'utilisation d'une valeur du facteur de réduction plus petite que la taille moyenne d'une personne et parce que les trajectoires des personnes sont formées par les centres des gravité et non pas par toutes leurs formes.

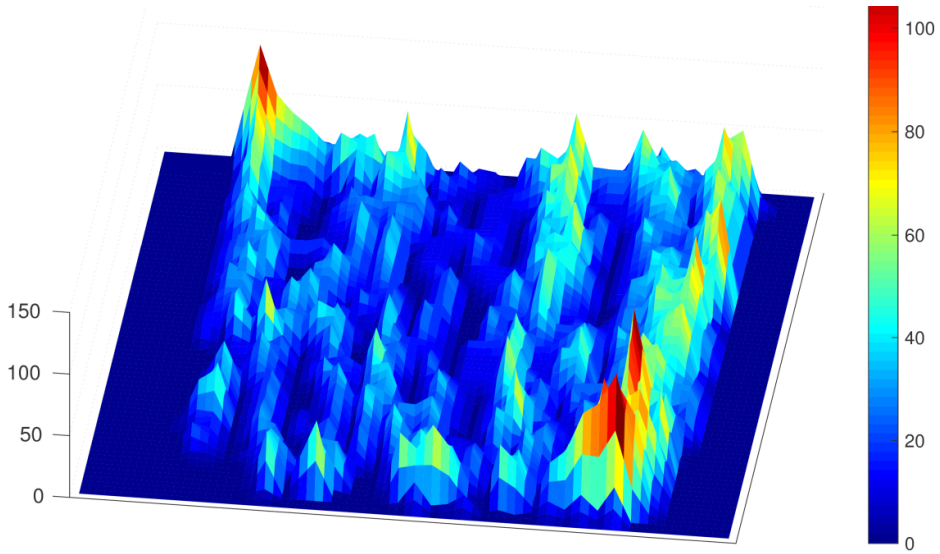


FIGURE 5.12 – Représentation 3D de la carte d'occupation de JD01 avec facteur de 16 × 12 pixels

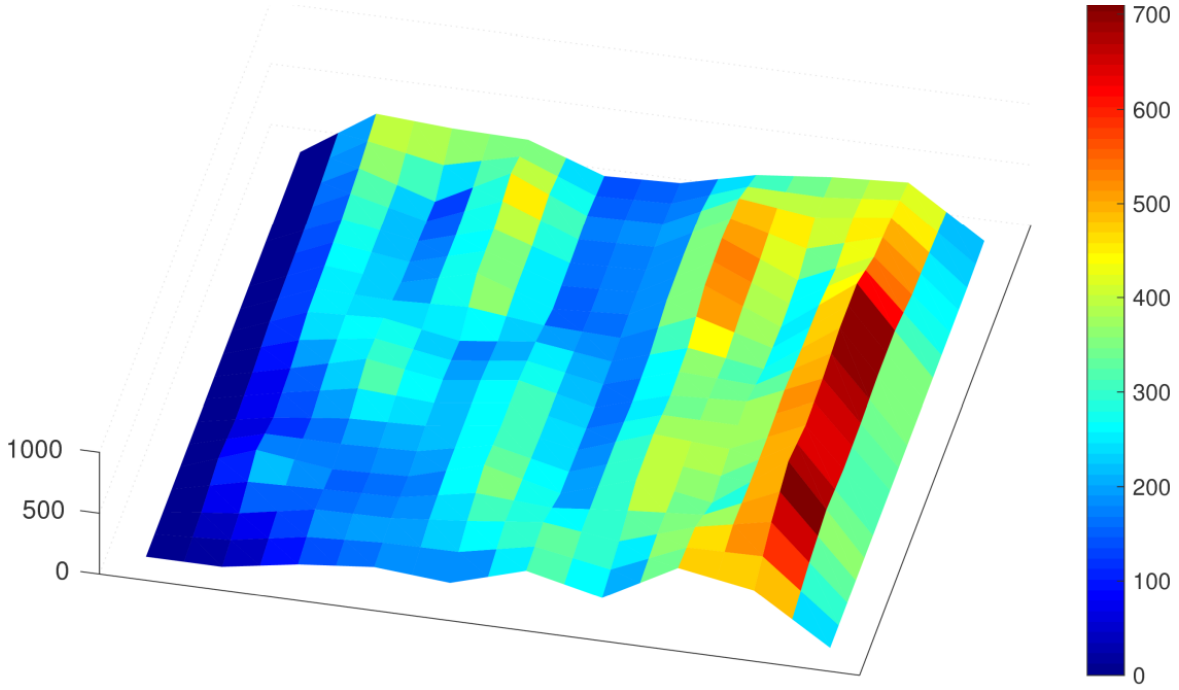


FIGURE 5.13 – Représentation 3D de la carte d'occupation de JD01 avec facteur de 64 × 48 pixels.

Dans la figure 5.13, qui résulte de l'augmentation du facteur de réduction par rapport à la figure 5.12, on observe que les vallées de hauteur 0 à l'intérieur de la surface ont disparu. A gauche de l'image, on retrouve encore une petite région de hauteur 0. Ceci provient du fait que la caméra qui permet d'extraire ces trajectoires est stéréoscopique et n'estime pas de valeurs de profondeur dans cette région (voir chapitre 2.3.2).

Seuils de temps d'utilisation

Les *seuils de temps d'utilisation* servent à filtrer le bruit et à améliorer l'analyse. On propose pour chaque jeu de données une analyse initiale de la carte non filtrée et de la carte filtrée par la moyenne de ses valeurs (Fig. 5.14).

Dans la figure 5.14, on présente des exemples du filtrage de O_c. Sur le JD03, chaque image représente une carte d'occupation filtrée O_f, en 2D de la même manière que la figure 5.6, avec différents valeurs de seuillage. La première image représente une carte d'occupation sans filtrage où O_{min} = 0% et O_{max} = 100%. Dans la deuxième image, on observe la carte d'occupation filtrée O_f entre 0% et 50% du temps d'occupation. La troisième image représente le filtrage par défaut (entre la valeur minimale et la moyenne). Dans ce cas, O_f est filtrée entre 0% et 10%. La dernière image représente O_f filtrée entre 0% et 5%.

Comme le montrent ces différentes images, les seuils de temps d'utilisation nous permettent d'observer plus clairement l'utilisation de l'espace et les différentes dynamiques d'utilisation, et en même temps, d'observer les points de fréquentation maximale. Plus de détails sur l'utilisation et le fonctionnement des seuils de temps d'utilisation sont donnés dans la section 5.1.2.

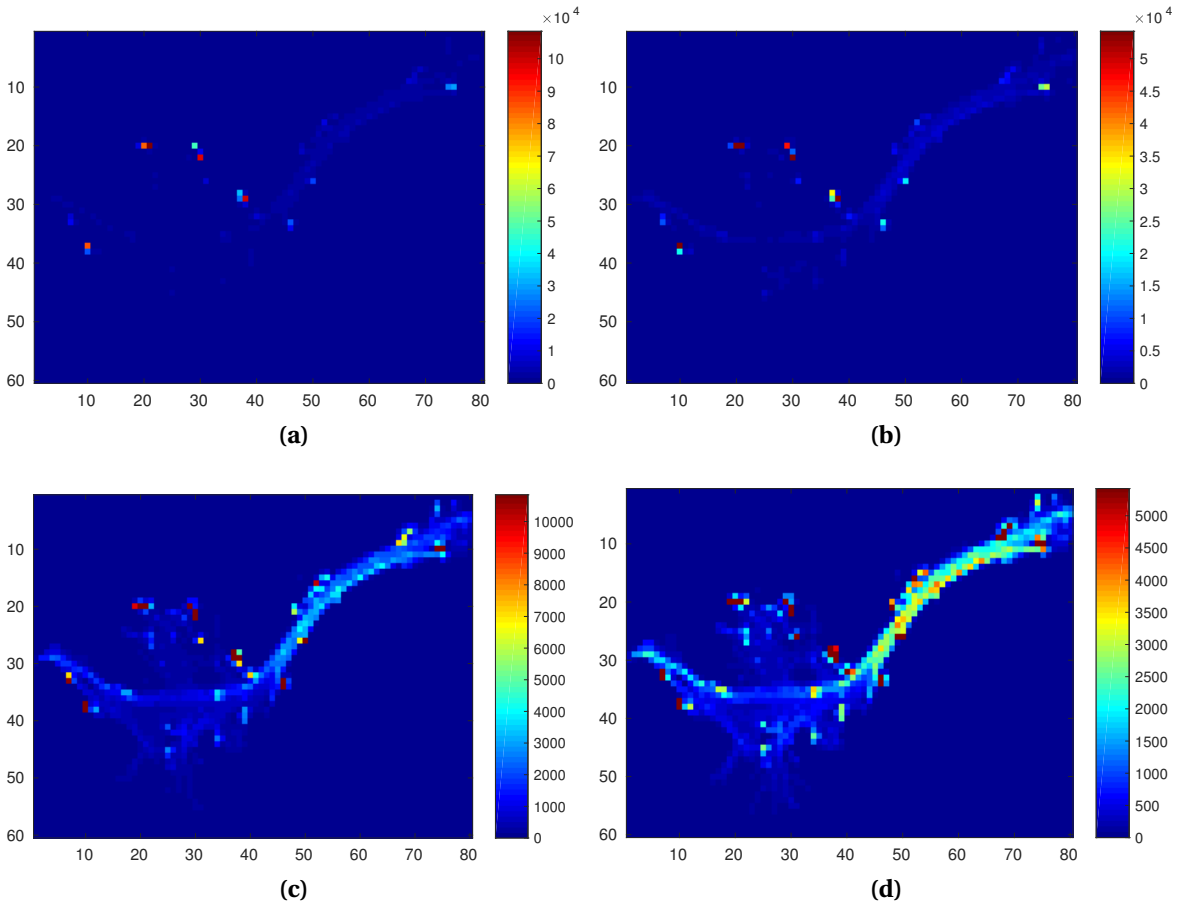


FIGURE 5.14 – Représentation du filtrage de la carte d'occupation.

Résultats de la génération de la carte d'occupation

En estimant la carte d'occupation, on cherche à comprendre la dynamique de la scène. Cette dynamique est décrite par le pourcentage d'utilisation de chaque région et par la similitude de temps d'utilisation des régions. En utilisant les seuils de filtrage, on cherche à trouver une similarité dans l'utilisation de l'espace dans les différentes régions, représentée par une homogénéité de couleurs. Cette similarité est importante parce qu'elle nous permet de comprendre l'utilisation globale et le comportement des personnes.

D'abord, on présente l'estimation de la carte d'occupation de JD01. Dans la figure 5.15, on observe les cartes d'occupation de JD01 avec des seuils de filtrage de 0 % à 100 % et 0 % à 63 %. Comme ce jeu de données est petit et que la majorité des trajectoires sont dans un sens vertical, on peut faire une appréciation de l'utilisation de l'espace global, en regardant les trajectoires placées sur la figure 5.16. On observe une utilisation de l'espace chargée à l'intérieur du rectangle rouge sur le coté droit. Ce comportement commence à être visible dans la première carte non filtrée (Fig. 5.15a) et il est plus clairement observable dans la deuxième carte filtrée comme la grande région rouge homogène (Fig. 5.15b). Cependant, il est difficile de faire une appréciation similaire pour des jeux de données plus complexes en nombre ou en sens des trajectoires.

On présente maintenant l'estimation de la carte d'occupation de JD02. Dans la figure 5.17, on observe les cartes d'occupation de JD02 avec des seuils de filtrage de 0 % à 100 % et 0 % à 58 %. On constate que la partie supérieure de la scène est la plus utilisée. En filtrant la carte par la valeur moyenne, une zone très homogène est révélée (Fig. 5.17b). De plus, cette carte nous permet d'identifier le schéma le plus utilisé. Si l'on se situe sur le rectangle rouge de coordonnées (5,2), on peut se déplacer vers le rectangle voisin avec la valeur la plus élevée sur le voisinage jusqu'à arriver au bord gauche de l'image. Ce chemin est le plus utilisé et il est facilement identifiable.

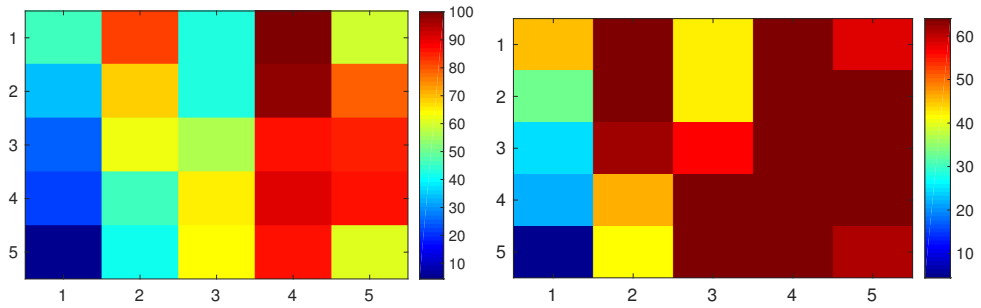


FIGURE 5.15 – Représentation 2D des cartes d'occupation de JD01 avec un facteur 128 × 96. a) Carte non filtrée. b) Carte filtrée à 63 % (valeur moyenne).

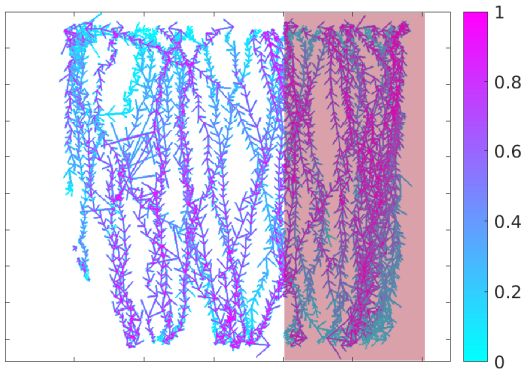


FIGURE 5.16 – Représentation de trajectoires du JD01. Le début d'une trajectoire est de couleur bleu cyan et la fin de couleur magenta. Le pic d'utilisation de l'espace est à l'intérieur du rectangle rouge.

Dans le cas du JD02, à la différence de la carte de chaleur (Fig. D.2), la carte d'occupation nous montre l'utilisation globale de l'espace sans créer d'artefacts sur les régions où des personnes sont restées statiques sur de longues périodes de temps.

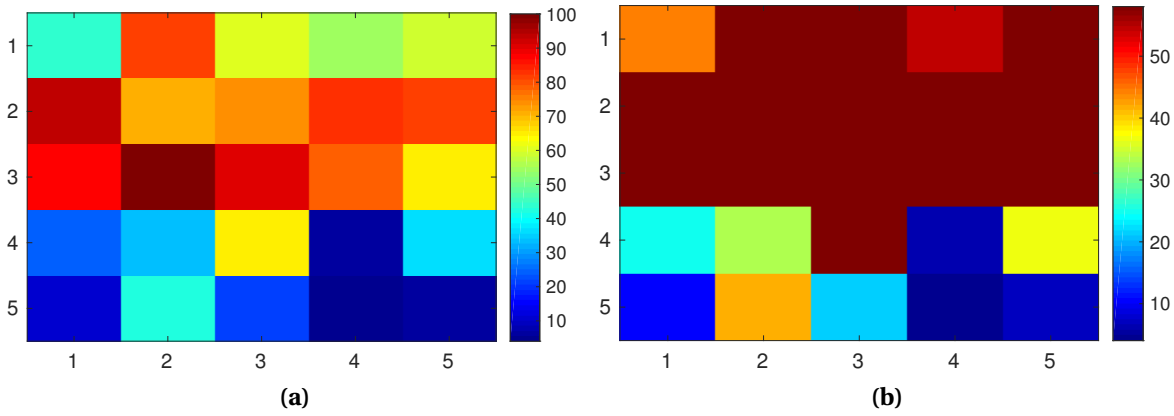


FIGURE 5.17 – Représentation 2D des cartes d'occupation de JD02 avec un facteur 128 × 96. a) Carte non filtrée. b) Carte filtrée à 63 % (valeur moyenne).

On présente l'estimation de la carte d'occupation de JD03. Dans la figure 5.18, on observe les cartes d'occupation de JD03 avec des seuils de filtrage déjà décrits. De plus, dans la figure 5.18, on augmente le seuil de filtrage pour extraire une région rouge qui traverse le centre commercial de manière horizontale. On peut observer également d'autres régions homogènes formées au centre en couleur jaune et bleu cyan. Ces régions apparaissent comme des ramifications du schéma principal ou des régions de dispersion de points maximas situés au centre nord de l'image. La carte d'occupation n'est pas suffisante pour comprendre toute la dynamique de la scène. Pour cette raison, on doit continuer par l'exploration des données avec les autres cartes d'entrées et de sorties. Ces cartes pourraient nous donner la signification des points maximas isolés, par exemple les points rouges repérés par les coordonnées (20, 20) dans la figure 5.18.

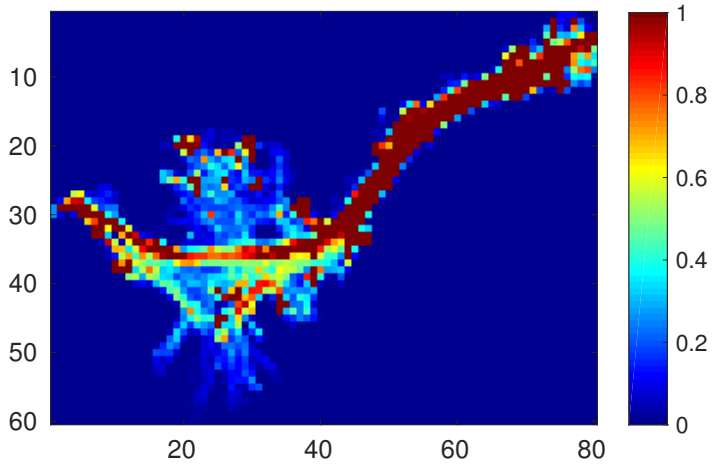


FIGURE 5.18 – Représentation 2D de la carte d'occupation de JD03 réduite par un facteur 16 × 12 et filtrée entre 0% et 1%.

De manière générale, en utilisant le filtrage du temps d'utilisation, on peut homogénéiser les cartes pour commencer à avoir une piste et déduire des comportements de la scène. Par exemple, il est intéressant d'observer que l'on trouve des régions (plus ou moins) homogènes verticalement dans la figure 5.15 et horizontalement dans les figures 5.21 et 5.18. Ces cartes d'occupation nous donnent donc des pistes sur le flux comportemental de la scène. En effet, la prévisualisation des vidéos à partir desquelles les trajectoires ont été extraites confirme la pertinence de ces analyses.

En conclusion, cette méthode apporte une visualisation plus significative de l'utilisation de l'espace observé. Il nous permet de visualiser les endroits plus fréquentés et les schémas les plus utilisés, d'éviter la création d'artefacts dans des régions où les personnes sont restées sur de longues périodes de temps, et de trouver des régions qui présentent une utilisation de l'espace similaire. De plus, on considère que cette représentation de l'utilisation de l'espace est plus pertinente que la carte de chaleur et qu'il est possible d'utiliser différents types de caméras (3D active et passive, et 2D) sur de courtes et longues périodes de temps.

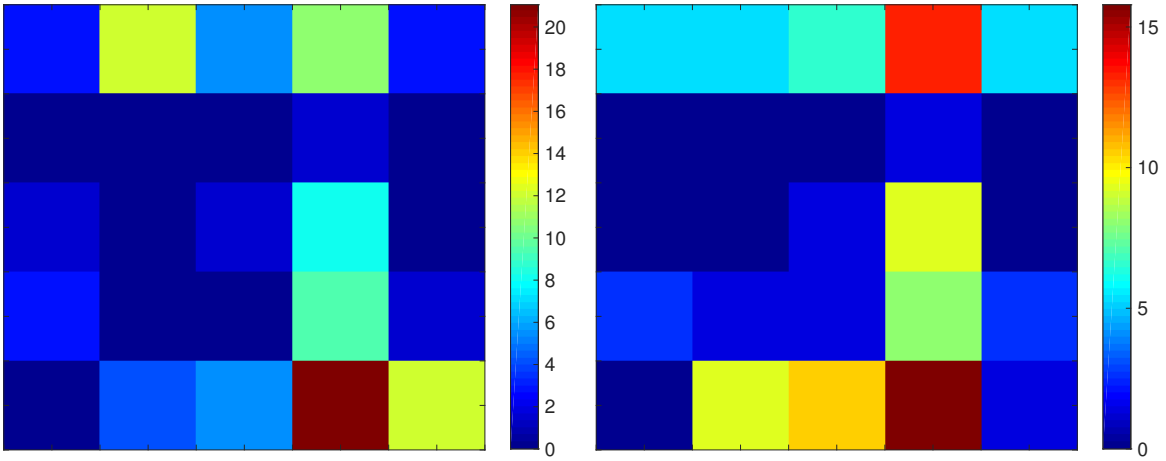
Détection des points d'entrées et de sorties

On a estimé les cartes d'entrées et de sorties simultanément, du fait qu'elles sont complémentaires sur tous les jeux de données. On fait référence aux points de début et de fin des trajectoires comme « points d'accès » à la scène. On a exploré plusieurs valeurs du facteur de réduction λ pour mieux comprendre la dynamique des points d'accès.

D'abord, en explorant les résultats du JD01 avec le même facteur utilisé dans la carte d'occupation (Fig. 5.19). Les gros pavés résultant du groupement des points d'accès reflètent correctement la réalité de la scène où les trajectoires commencent et finissent dans les bords supérieurs et inférieurs, du fait que la caméra est située à l'entrée d'un magasin (voir description). A la différence de la carte d'occupation, ces cartes montrent le pourcentage de trajectoires qui commencent ou finissent dans chaque pavé. On relie cette valeur à une échelle de couleur pour faciliter la compréhension des cartes.

Dans la carte de la figure 5.19, on observe que les bords supérieurs ont des valeurs similaires pour chaque pavé, et qu'il existe une région de concentration dans le bord inférieur (pavé rouge) au même endroit dans les deux cartes d'entrée et de sortie. On note aussi une charge des entrées et des sorties du côté droit, ce qui est cohérent avec la carte d'occupation de la section précédente.

En utilisant les résultats des trois cartes, on peut conclure que dans la scène observée, au niveau du bord inférieur, il existe une configuration spatiale (arrangements des objets dans l'espace) qui facilite les entrées et sorties, comme le montre la région rouge, avec une plus grande fréquentation. Cette configuration influence les comportements des personnes au voisinage de ce point, en modifiant le côté droit de la carte d'occupation. De plus, on observe que les entrées sont plus fréquentes à droite de la région rouge et les sorties à sa gauche, expliquant aussi une dynamique d'accès à la scène.

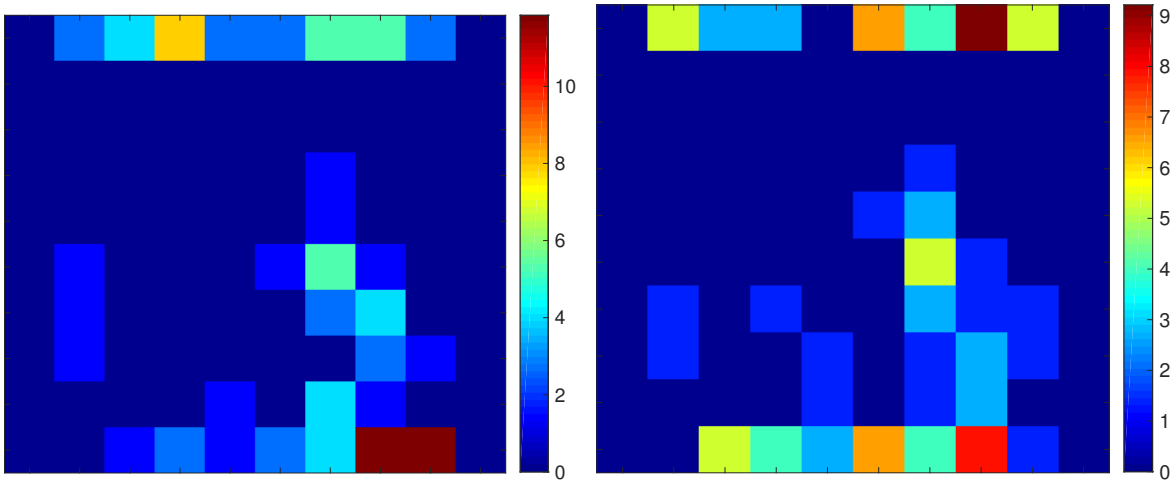


(a) Carte d'entrées **(b)** Carte de sorties

FIGURE 5.19 – Représentation 2D des cartes d'entrées et de sorties de JD01 réduite par un facteur 128 × 96.

Par ailleurs, on a trouvé des points d'accès au milieu de la scène qui ne peuvent pas correspondre à des trajectoires valides. En prenant en compte la faible précision du système utilisé, ce phénomène peut être expliqué par l'introduction des erreurs au niveau du système de suivi. Pour mieux comprendre la nature de ces anomalies, on a diminué le facteur de réduction à 64 × 48 (Fig. 5.20).

En diminuant ce facteur, on désagrège les données pour identifier plus correctement des inconsistances du suivi au milieu de la scène. On peut se permettre de ne pas compter les 3 dernières lignes des cartes d'entrées et de sorties - voir le bas de la figure 5.20 - qui représentent approximativement l'épaisseur d'une personne et de faire la somme des pourcentages des rectangles qui restent au milieu de la scène. En comparant les résultats des figures 5.19 et 5.20, les pourcentages d'anomalies du milieu de la scène sont respectivement de 22 % et 15 % dans les entrées. Pour les sorties, ces valeurs sont respectivement 27 % et 14 %. Ces résultats sont plus cohérents avec les premiers résultats du prototype en cours de test au sein de Shopline.



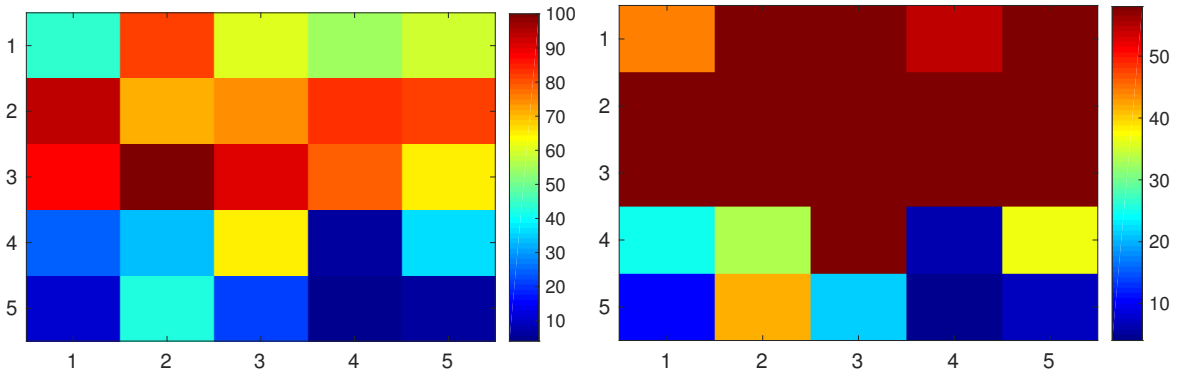
(a) Carte d'entrées **(b)** Carte de sorties

FIGURE 5.20 – Représentation 2D des cartes d'entrées et de sorties de JD01 réduite par un facteur 64 × 48.

Nous pouvons constater que la méthode d'estimation des cartes d'entrées et sorties nous fournit un autre moyen pour évaluer la performance des algorithmes et pourrait nous aider à détecter de fausses trajectoires qui affectent le comptage.

En explorant les résultats du JD02 réduite par un facteur de réduction de 128 × 96 (Fig. 5.21), on identifie correctement les points d'accès de la scène, et on observe particulièrement le seul point d'entrée possible visible dans le bord inférieur de la scène. Dans ce scénario, les personnes sont bloquées par une barrière qui fait de ce tourniquet le seul point d'accès. Dans les autres

endroits (sur le haut et les côtés) de la scène, on trouve effectivement des personnes qui rentrent. Cependant, on trouve encore une fois des indications de points d'accès à l'intérieur de la scène. Ceux-ci proviennent de problèmes de mauvaise détection. A la différence du jeu de données précédent, ces erreurs ne dépassent pas 4 %. Exceptionnellement, ces erreurs (le pire cas est de 4 %) sont provoquées par le scenario des occultations de têtes présenté dans le chapitre 3.



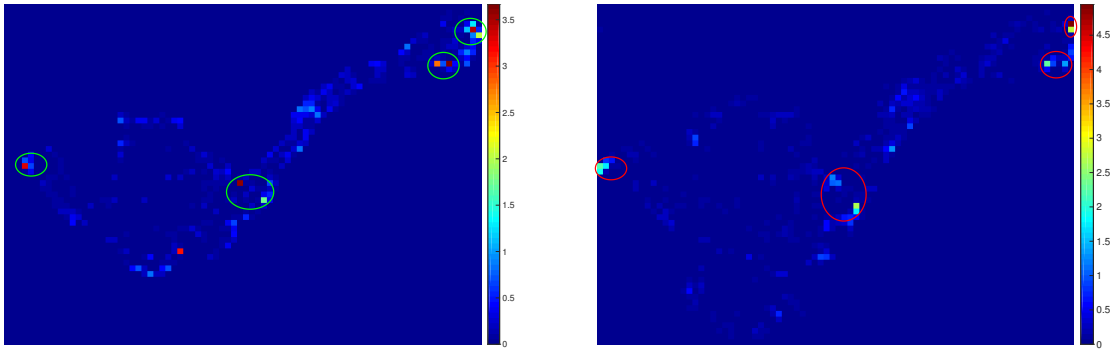
(a) **(b)**

FIGURE 5.21 – Représentation 2D des cartes d'entrées et de sorties de JD02 réduite par un facteur 128 × 96.

On peut conclure que les résultats du JD03 sont très intéressants parce qu'on peut extraire des informations sur la structure du centre commercial à travers cette méthode (voir Fig. 5.22). D'abord, on extrait, à partir des points maxima des deux cartes, les informations suivantes :

- Les points d'accès les plus fréquentés avec un pourcentage entre 2.5 % et 3.5 %.
- Les points d'accès sont spécialement élevés dans certaines régions des cartes, par exemple aux extrémités horizontales de l'espace observé. Ils sont signalés par des cercles verts pour les entrées et rouges pour les sorties (Fig. 5.22).
- Les points d'accès sont voisins et complémentaires topologiquement, c'est-à-dire que les points d'entrées maxima sont voisins topologiques des points de sorties maxima dans la majorité des cas et ils n'occupent pas le même espace.

A partir de ces observations, on peut déduire quels sont les points d'accès au centre commercial. Le reste des points dans ces cartes représentent les entrées et sorties des magasins de ce centre. Dans ce cas, cette méthode nous permet de déduire la structure du centre commercial à partir des données et, en même temps, d'identifier le poids de chaque point d'accès sur la totalité des trajectoires.



(a) **(b)**

FIGURE 5.22 – Représentation 2D des cartes d'entrées et de sorties de JD03 réduite par un facteur 16 × 12.

En conclusion, cette méthode nous permet à la fois de déterminer les points d'accès d'une manière fiable et nous offre un autre moyen pour discriminer les erreurs de suivi et ainsi améliorer le comptage. Elle nous a également permis de mieux comprendre un environnement inconnu à la fois dans sa dynamique et sa distribution spatiale, ce qui fait de cette méthode un outil puissant pour l'étude du comportement.

5.3.4 Analyse des trajectoires

Dans cette section, on présente les résultats de la segmentation d'un flux comportemental, une validation d'implantation de celle-ci et la visualisation dynamique des trajectoires.

Segmentation d'un flux comportemental

On présente dans cette section les résultats, l'analyse et l'application de la méthode pour segmenter le flux du comportement des personnes. L'objectif principal de cette étude est de trouver le moyen pour grouper les trajectoires et trouver des similitudes de comportements entre différentes personnes. On utilise comme paramètres, dans cette section, le nombre de clusters K, la résolution du champ vectoriel, le lissage et les attributs qui nous permettront d'améliorer le groupement. On attend comme résultat un flux comportemental représenté par le champ vectoriel de chaque cluster et les groupes de trajectoires similaires entre elles. Pour ce faire, on teste la méthode VFKM et sa variante améliorée AFKM.

On cherche à trouver le nombre correct de clusters qui caractérise chaque jeu de données. Ce nombre K dépend fortement de l'usage et de la signification de la segmentation, plutôt que d'une condition mathématique. En effet, notre objectif pour le JD01 est de séparer les entrées des sorties, en obtenant principalement deux clusters. Le premier doit contenir les trajectoires qui partent de la partie haute de l'image et finissent vers le bas. Le champ vectoriel espéré pour ce cluster devra montrer tous les vecteurs (ou la majorité) en direction du bas de l'image. Le deuxième cluster doit contenir les trajectoires qui partent de la partie basse de l'image et finissent en haut. Le champ vectoriel espéré pour ce cluster doit montrer tous les vecteurs (ou la majorité) en direction du haut de l'image.

De la même manière, notre objectif pour le JD02 est de trouver 4 clusters en représentant les principales directions souhaitées et les champs vectoriels respectifs vers ces directions.

Enfin, pour le JD03, nous présentons les résultats avec notre méthode sans connaître a priori sa dynamique.

VFKM

On teste d'abord cette méthode pour vérifier si cette approche est fiable pour segmenter les trajectoires des personnes et trouver le flux comportemental attendu.

On a commencé par le premier jeu des données avec la valeur de K égal à 2. On observe dans la figure 5.23, les deux flux (descendant et montant constituant l'objectif) malgré les artefacts sur le coin supérieur gauche de l'image (Fig. 5.23b), où il y a des flèches dans le sens contraire du reste. Cependant, on obtient une segmentation des trajectoires très faible (Fig. 5.23c et 5.23d) parce qu'il y a un grand nombre de trajectoires dans des sens différents.

On évalue différentes combinaisons des nombres de clusters, résolutions et lissage sans trouver de meilleurs résultats. On présente dans la figure 5.24 le résultat de la segmentation en 4 clusters, une résolution de 10x10 et un lissage de 0,5. On observe, comme dans la segmentation en 2 clusters, une bonne répartition des flux homogènes -(a) et (b)- mais les trajectoires groupées sont dans des sens différents et ne représentent pas le comportement désigné par les flux.

On trouve que cette méthode groupe les trajectoires par leur forme, sans prendre en compte leur direction, ignorant ainsi un paramètre important pour segmenter les flux comportementaux (voir Fig.5.23). Les premiers résultats de cette approche nous ont donné une piste pour faire de la segmentation mais on devait restreindre celle-ci en utilisant des attributs de trajectoires.

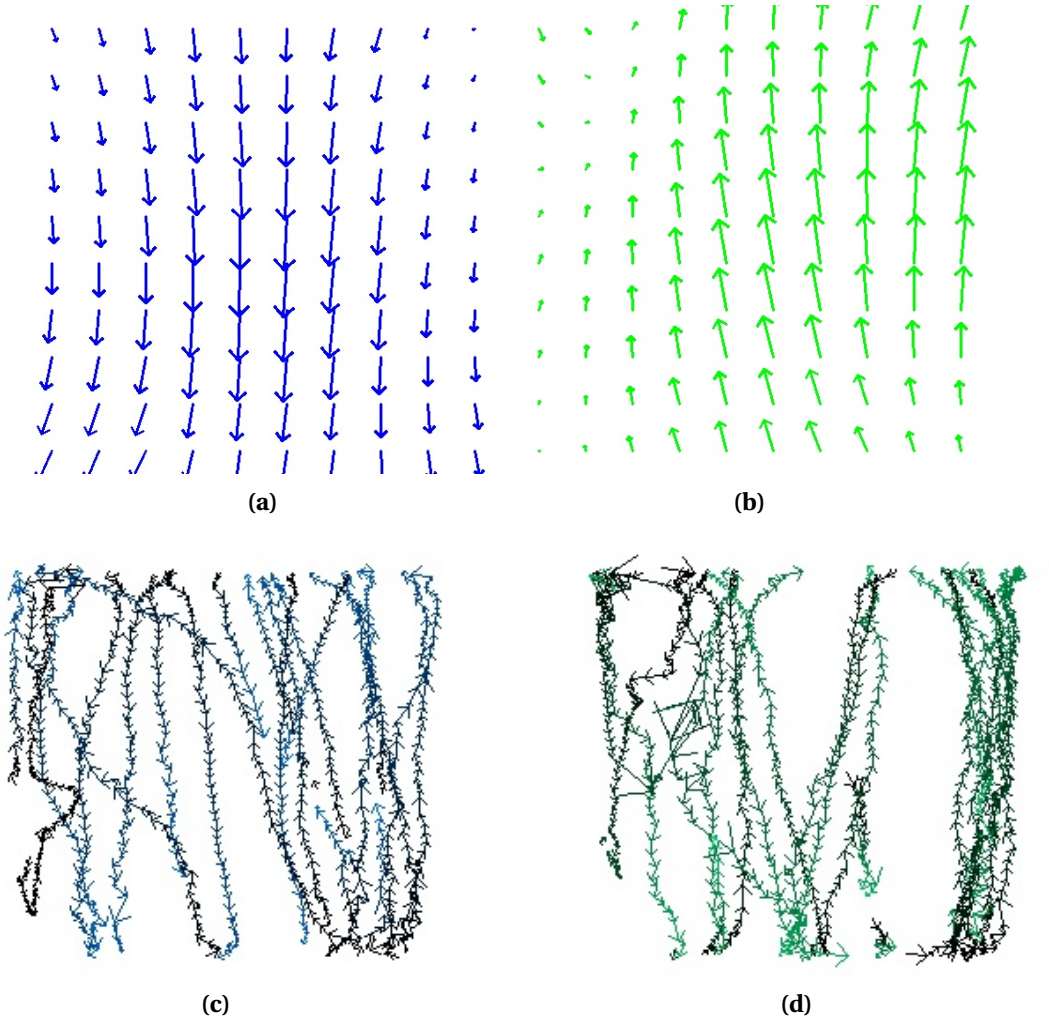


FIGURE 5.23 – Segmentation du flux comportemental à 2 clusters (VFKM). Les images (a) et (b) sont la représentation des champs vectoriels de chaque cluster. Les images (c) et (d) sont des trajectoires appartenant au même cluster.

En conclusion, en utilisant VFKM comme méthode pour faire de la segmentation du flux comportemental de JD01, on observe : des similarités dans les trajectoires groupées en termes de forme, mais pas réellement un comportement similaire, par exemple des personnes qui rentrent par rapport aux personnes qui sortent du magasin. Ce phénomène est cohérent avec le fait que la méthode cherche les lignes de courant « streamlines » qui suivent la forme des trajectoires et pas leur direction. Cette méthode ne permet pas de grouper les trajectoires par comportements similaires. Pour répondre à cette difficulté, nous avons utilisé la méthode AFKM.

AFKM

A la différence de la méthode précédente, avec AFKM, nous sommes capables de prendre en compte les attributs des trajectoires comme la direction (entre autres) pour segmenter les flux de chaque jeu de donnés.

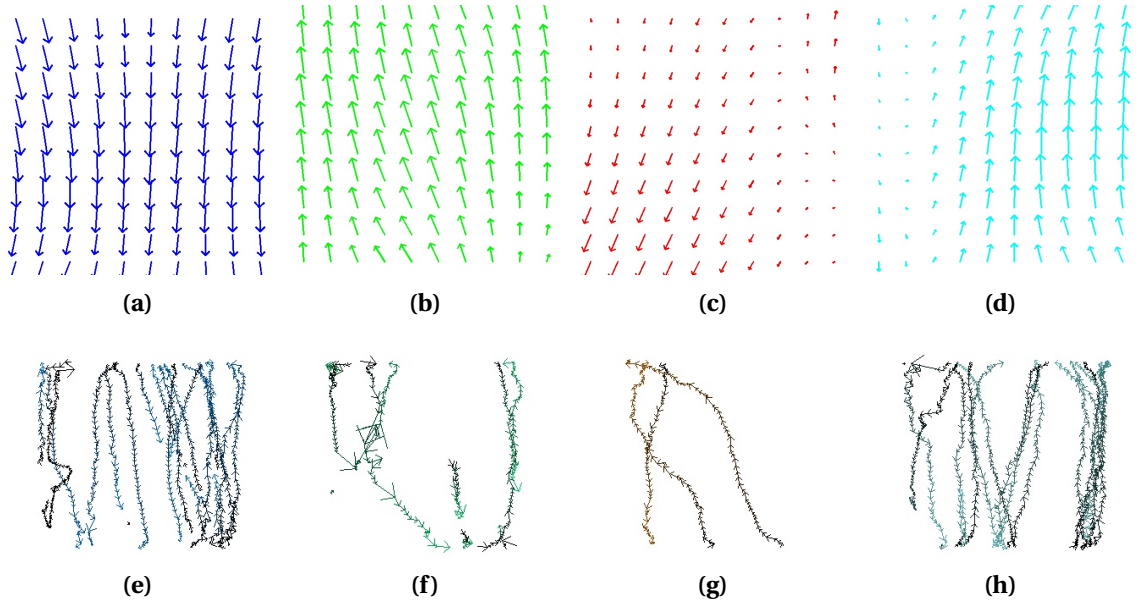


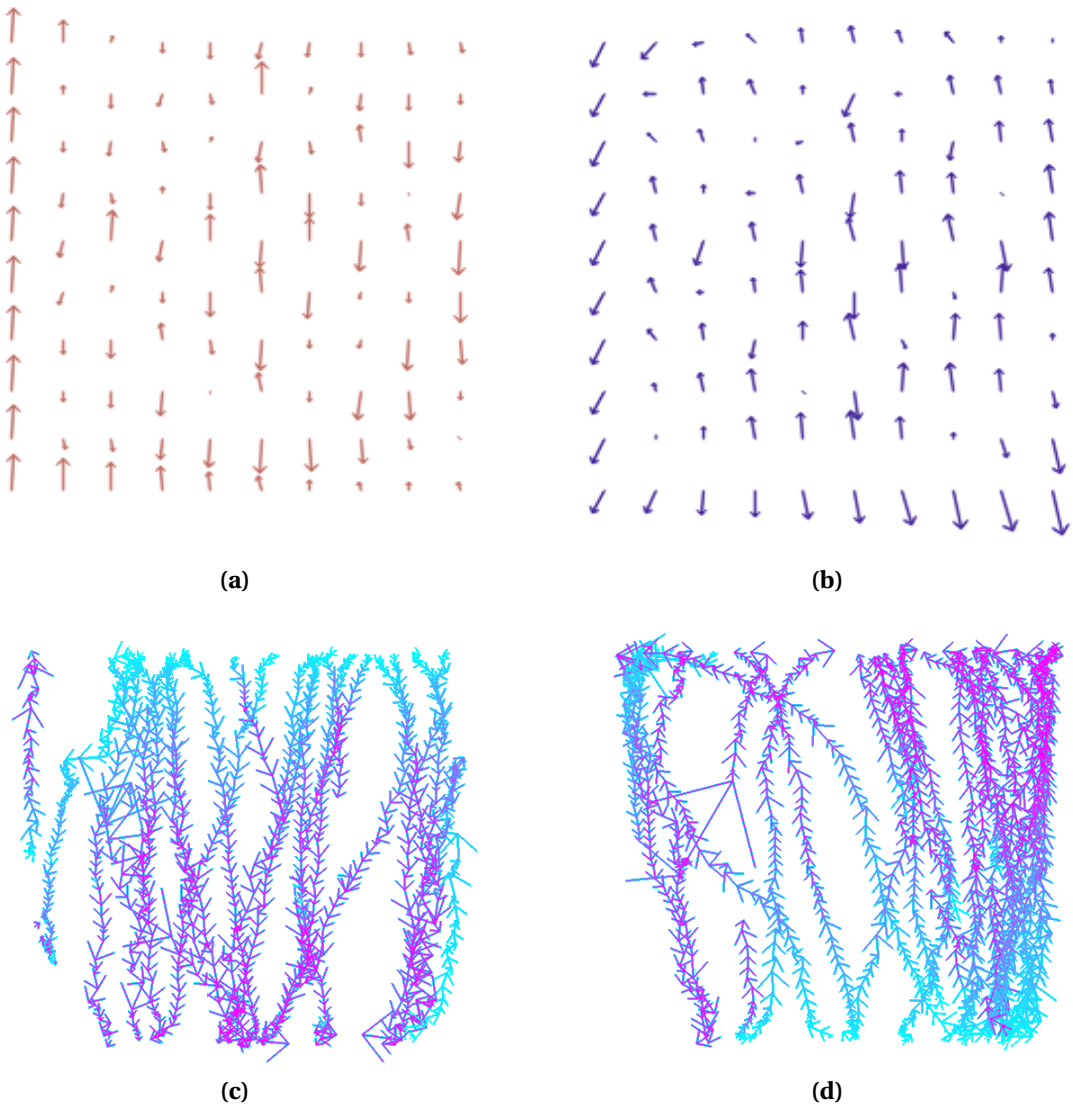
FIGURE 5.24 – Segmentation du flux comportemental à 4 clusters (VFKM). Les images (a, b, c) et (d) sont la représentation des champs vectoriels de chaque cluster. Les images (e, f, g) et (h) sont des trajectoires qui appartiennent à chaque clusters.

On a commencé par utiliser deux attributs : les composantes x et y du vecteur de vitesse. Dans la figure 5.24, on observe les résultats de la segmentation du JD01 en 2 clusters, une résolution de 10×10 et un lissage de 0,5 (de la même manière que l'expérience précédente). La première ligne d'images contient les flux comportementaux représentés par des champs vectoriels et la deuxième ligne contient la représentation des trajectoires associées au même cluster. Dans chacune de ces dernières images, les trajectoires sont représentées par une séquence de vecteurs consécutifs où le début est de couleur cyan et la fin est de couleur magenta.

En analysant ces images, les vecteurs de flux sont plus erratiques que dans la méthode VFKM. La majorité des flèches partent dans le même sens mais quelques vecteurs partent dans le sens contraire avec une grande magnitude. D'autre part, le groupement des trajectoires est plus cohérent (Fig. 5.25c et 5.25d). Cette cohérence se manifeste, dans le cas du JD01, par le fait que toutes les trajectoires groupées partent du même endroit et dans le même sens. Dans la figure 5.25c et 5.25d, on observe effectivement que la majorité des trajectoires partent dans le même sens, sauf quelques-unes. Comme les champs vectoriels dépendent des trajectoires qui lui appartiennent, on observe que les trajectoires mal groupées ou les trajectoires particulières introduisent des artefacts sur les flux de comportement (Fig. 5.25a et Fig. 5.25b).

Dans la figure 5.25, les images (a) et (b) sont la représentation des champs vectoriels de chaque cluster. Les images (c) et (d) sont la représentation des trajectoires qui appartiennent respectivement à chaque cluster.

On continue à explorer le groupement en augmentant le nombre K de clusters et en conservant les autres valeurs (résolution et lissage). La valeur de K est choisie entre 3 et 7. Dans la figure 5.26, on trouve une amélioration des flux résultant par rapport à l'objectif et de plus, on obtient la séparation des trajectoires particulières – voir images (f) et (h) de la figure 5.26. Pour K=3, on obtient des flux similaires aux résultats K=2 de la figure 5.25, mais elle est moins bruitée avec une segmentation cohérente. Le troisième cluster additionnel groupe des trajectoires qui ne traversent pas toute la scène ou qui ont des mouvements particuliers (Fig. 5.27a). Pour K=4, on obtient des flux de comportement corrects avec une segmentation cohérente (Fig. 5.26). Les deux clusters additionnels (Fig. 5.26b et 5.26d) servent à grouper des trajectoires courtes, erratiques et particulières (Fig. 5.26f et 5.26h). A partir de K=5 à 7 voir figure 5.28, on commence à voir une détérioration sur le flux, c'est-à-dire des répétitions anormales des flux de comportement avec des petites variations (Fig. 5.28).



(a) **(b)** **(c)** **(d)**

FIGURE 5.25 – Segmentation du flux comportemental à 2 clusters en utilisant la méthode AFKM.

Dans la figure 5.26, les images (5.26a, 5.26b, 5.26c) et (5.26d) sont la représentation des champs vectoriels de chaque cluster. Les images (5.26e, 5.26f, 5.26g) et (5.26h) sont la représentation des trajectoires appartenant à chaque cluster. On observe que les deux sens cherchés (5.26a) et (5.26c) sont bien définis et les trajectoires qui lui appartiennent sont correctement groupées. Dans l'image (5.26c), on observe une seule trajectoire qui semble être mal classifiée (voir figure 5.26g), une flèche magenta dans le coin inférieur droit. Au contraire, cette trajectoire part du bas vers le haut, mais sa direction change et finit sur le bord inférieur de l'image (Fig. 5.27b).

Dans la figure 5.27, on présente dans la première image le troisième cluster pour K = 3. On observe qu'il est composé de trajectoires qui ne traversent pas toute la scène ou des trajectoires particulières. Dans la deuxième image, on observe la trajectoire particulière qui a les deux comportements et peut se retrouver dans n'importe quel cluster. Ces types de trajectoires (5.27a) et (5.27b) expliquent aussi les artefacts formés dans les champs vectoriels.

Dans la figure 5.28, on observe la dégradation des flux et de groupement. Par exemple, on trouve 3 groupes dans lesquels leurs trajectoires partent du haut vers le bas (5.28e, 5.28f et 5.28k). En réalité, ces 3 groupes peuvent être représentés par un seul flux, comme dans la segmentation entre 2 et 4 clusters. De la même manière, les flux (5.28c) et (5.28m) représentent un groupe des trajectoires similaires.

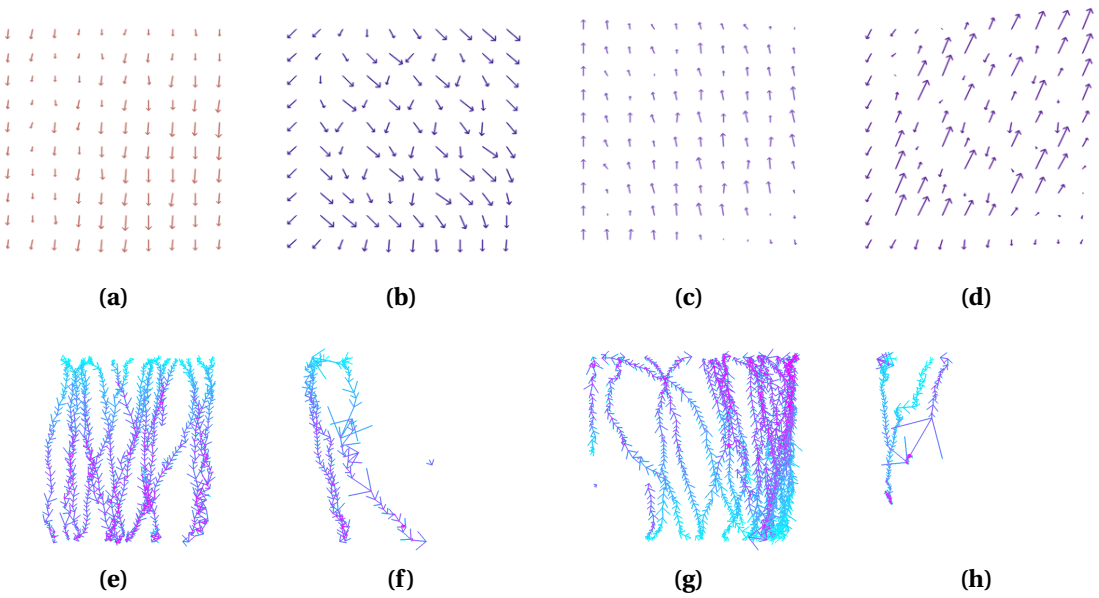


FIGURE 5.26 – Représentation d'une segmentation du flux comportemental à 4 clusters de JD01 en utilisant la méthode AFKM.

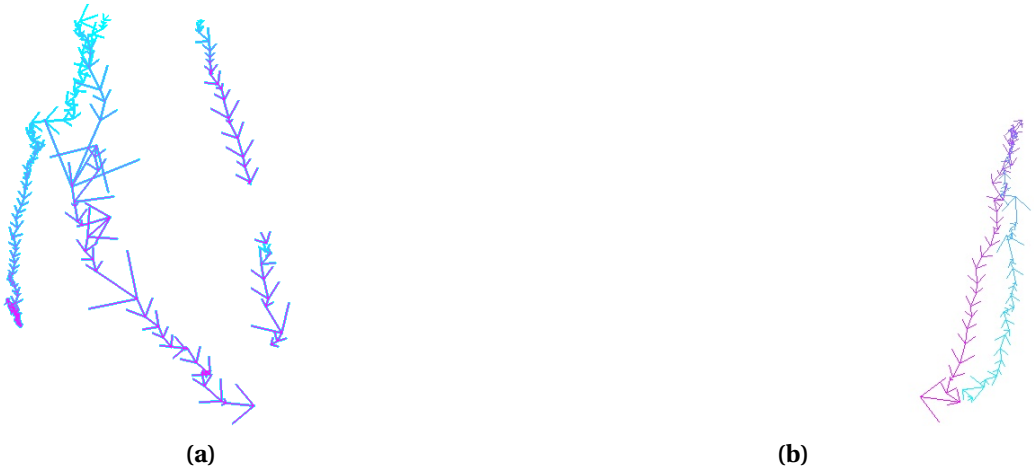


FIGURE 5.27 – Représentation des trajectoires particulières de JD01. a) Troisième groupe des trajectoires pour K=3. b) Trajectoires particulier qui a les deux sens.

Dans la figure 5.28, les images (5.28a,5.28b , 5.28c, 5.28d, 5.28i, 5.28j) et (5.28k) sont la représentation des champs vectoriels de chaque cluster. Les images (5.28e, 5.28f, 5.28g, 5.28h, 5.28k, 5.28l) et (5.28m) sont la représentation des trajectoires qui appartiennent à chaque cluster.

Du fait des bons résultats de segmentation en 4 clusters, on considère que les deux attributs (la composante de directions des axes X et Y) sont suffisants pour obtenir une segmentation correcte des trajectoires.

Pour analyser le JD02, on utilise les mêmes attributs des composantes x et y du vecteur de vitesse. On teste la segmentation du JD02 entre 2 et 7 clusters avec une résolution de 10×10 et un lissage de 0.5. Les images qui représentent les flux comportementaux et les trajectoires segmentées se trouvent en annexe H sauf pour K=4.

Pour K=2, on obtient un flux de comportement de droite à gauche et un autre dans le sens inverse. Effectivement, la majorité des trajectoires présentent les deux sens. Les trajectoires que l'on cherche à segmenter dans les sens verticaux sont réparties dans les deux groupes existants, dépendant du mouvement global de la trajectoire, vers la droite ou la gauche.

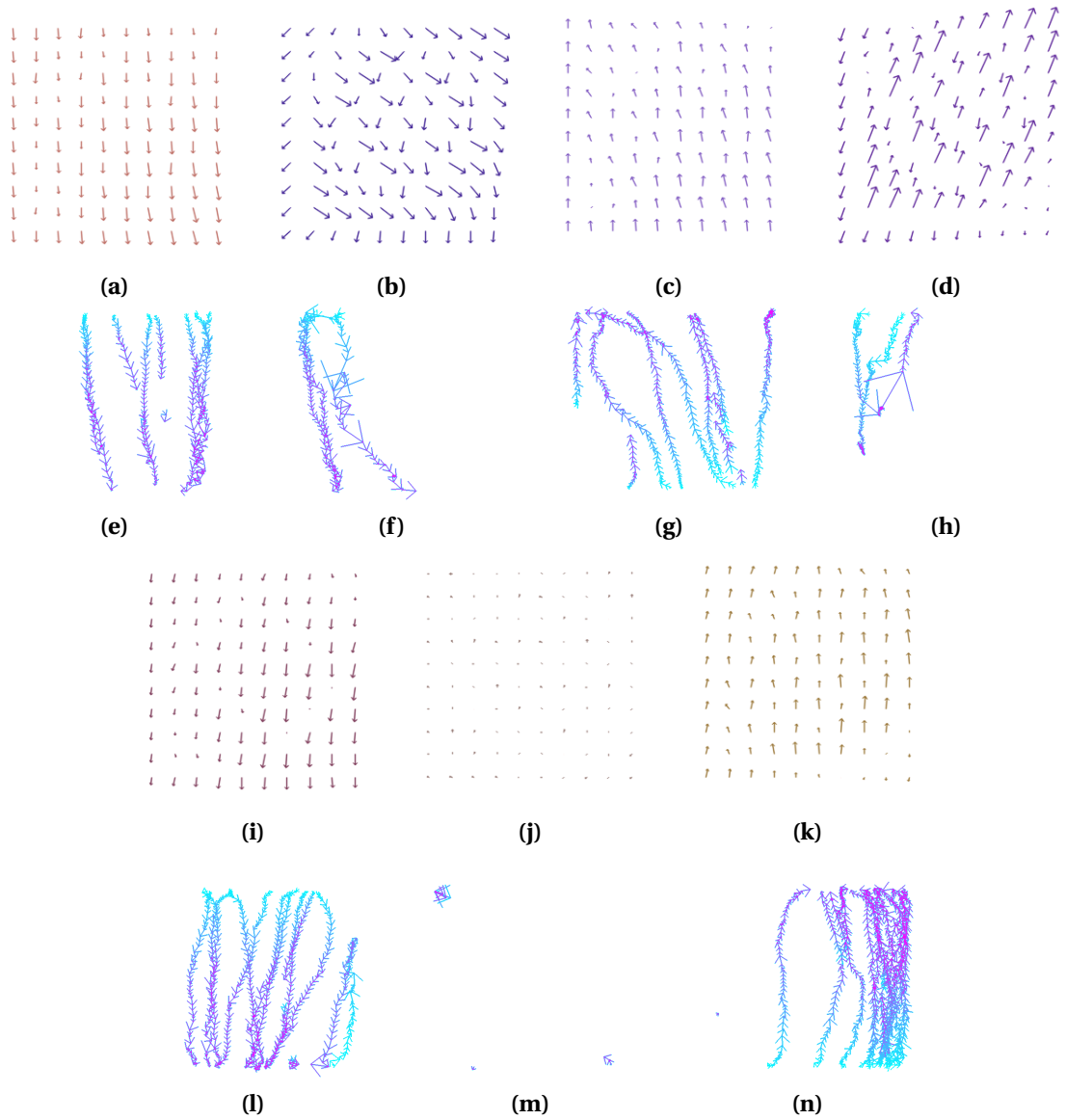


FIGURE 5.28 – Représentation d'une segmentation du flux comportemental à 7 clusters du JD01 en utilisant la méthode AFKM.

Pour K=3, en obtient un comportement de plus vers le bas de l'image avec une segmentation cohérente. Les trajectoires manquantes (vers le haut) sont groupées comme dans le cas précédent.

Pour K=4, on obtient une segmentation correcte et cohérente des trajectoires, cependant les flux de comportement vers le haut et vers le bas ne sont pas bien définis. On attribue cette déficience au faible nombre de trajectoires dans ces directions et aux trajectoires particulières de ce jeu de données qui ajoutent des artefacts aux champs vectoriels auxquelles elles appartiennent.

En analysant la figure 5.29, les vecteurs de flux verticaux (5.29b) et (5.29d) sont plus erratiques que les flux horizontaux (5.29a) et (5.29c), mais la segmentation des trajectoires est correcte par rapport à notre objectif. Dans le flux *sud,* on observe le comportement des trajectoires qui descendent en diagonale de deux coins supérieurs de l'image (voir image 5.29f) réfléchis dans la direction des vecteurs de la représentation (5.29b). Dans le cas du dernier flux (5.29d), on observe un comportement très erratique dans plusieurs sens. Encore une fois, on note la dépendance des champs vectoriels sur les trajectoires qu'ils regroupent comme dans ce cas, les trajectoires qui montent et les trajectoires particulières, en introduisant des artefacts.

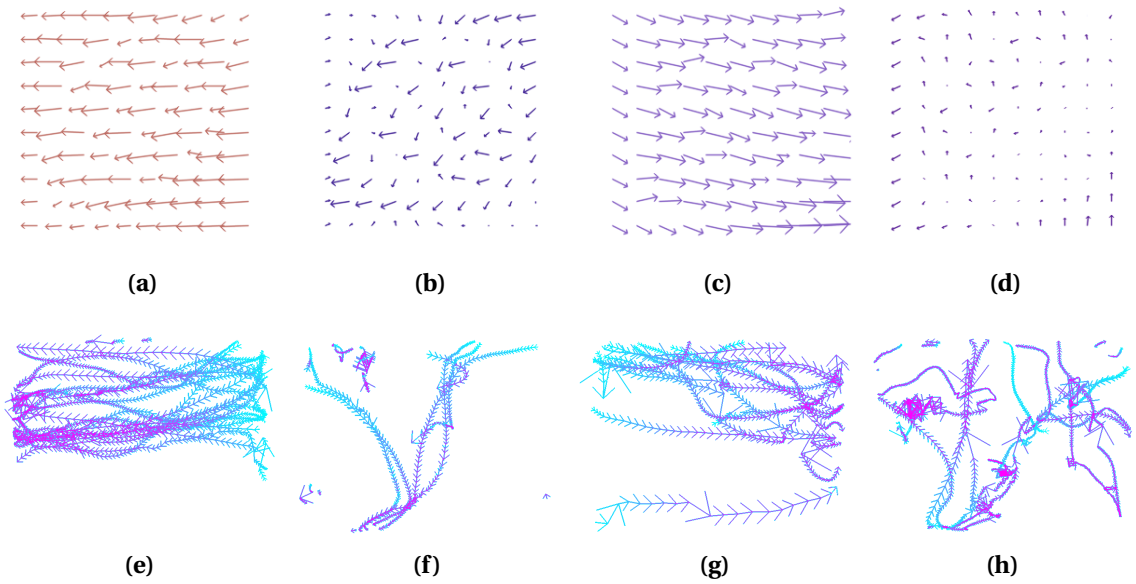


FIGURE 5.29 – Représentation d'une segmentation du flux comportemental à 4 clusters du JD02 en utilisant la méthode AFKM.

Dans l'illustration de la figure 5.29, les images (5.29a, 5.29b, 5.29c) et (5.29d) sont la représentation des champs vectoriels de chaque cluster. Les images (5.29e, 5.29e, 5.29f) et (5.29g) sont la représentation des trajectoires appartenant à chaque cluster.

Pour K entre 5 et 7, on observe le même comportement que dans le jeu de données précédent où l'un de flux est dupliqué en présentant de légères modifications du champ vectoriel relié aux trajectoires qu'il regroupe.

Le tableau 5.3 décrit les résultats obtenus pour les algorithmes de segmentation du flux comportemental. Le tableau est composé par le type d'algorithme, le jeu de données segmenté (JD), le nombre de clusters utilisé (NC), le nombre de clusters obtenu par comportement attendu et enfin le nombre de clusters en représentant un comportement différent. Le champ de comportement est composé de 5 éléments dont 4 sont les comportements attendus expliqués dans la description du jeu de données. Le dernier élément est un cluster qui ne représente aucun comportement spécifique (comportament inattendu CI) mais qui est identifié dans les trajectoires extraites.

En conclusion, les meilleurs résultats sont obtenus à partir de la segmentation AFKM. On trouve les flux de comportement attendus et un groupement de trajectoires cohérent avec ce flux. Cela nous permet d'utiliser cette méthode comme un outil fiable pour segmenter les trajectoires et extraire les informations sur le comportement global. Cependant, la configuration des paramètres est fortement liée aux des objectifs de l'utilisateur.

S'agissant des paramètres de lissage et de résolution, ceux-ci ont un rapport 1 : 1, c'est-à-dire que l'on peut prendre la résolution comme seul paramètre pour les utilisateurs, comme l'a déjà mentionné Ferreira [FKSS13]. Le nombre de clusters est un paramètre que l'on introduit manuellement mais qui requiert l'expertise de l'utilisateur en fonction de ses besoins.

TABLEAU 5.3 – Résultats de la segmentation du flux comportemental

Algorithme	JD	NC	Comportements				
			Attendus				Autres
			↑	↓	←	→	CI
VFKM	1	2	0	0	-	-	2
AFKM	1	2	1	1	-	-	0
AFKM	1	3	1	1	-	-	1
AFKM	1	4	2	1	-	-	1
AFKM	1	5	3	1	-	-	1
AFKM	1	6	3	1	-	-	2
AFKM	1	7	3	2	-	-	2
AFKM	2	2	0	0	1	1	0
AFKM	2	3	0	1	1	1	0
AFKM	2	4	0	1	1	1	1
AFKM	2	5	1	1	1	1	1
AFKM	2	6	1	1	2	1	1
AFKM	2	7	1	1	2	2	1

La validation et la représentation dynamique des trajectoires

La validation à distance est introduite sur les systèmes en production dans l'entreprise Shopline. Dans le processus normal d'installation, un électricien doit assurer la fourniture d'énergie et la communication au point d'installation, un technicien doit installer et paramétrer la caméra et, dans certains cas difficiles, un ingénieur doit re-paramétrer la caméra. Comme le coût du temps d'un ingénieur est plus élevé, l'entreprise Shopline cherche à réduire ce temps et donc son coût. Pour ce faire, la fonctionnalité de la validation a été mise en œuvre dans les caméras. Actuellement, il sert à valider l'installation correcte du système et permet d'améliorer les paramètres de configuration pour la détection de personnes en prenant en compte la spécificité des magasins.

La fonctionnalité de la reproduction de trajectoires a servi à débugger les méthodes décrites ici mais elle n'a pas encore été mise en œuvre en temps-réel (H24 et 7/7) dans le système des caméras intelligentes. Ceci s'explique par le fait que cela nécessite une forte consommation en communication et en stockage.

5.4 Conclusions

Dans ce chapitre, on définit le comportement comme un groupe de mouvements observables associés à des individus avec la vision des sciences comportementales, en soulignant ses propriétés relationnelles, dynamiques et de haute dimensionnalité pour décrire ce comportement dans une série chronologique. Ces propriétés nourrissent l'information collectée en fonction : des contraintes imposées par les expériences, du niveau de description et de la quantité d'informations collectées (axes comportementaux). De plus, on présente les différentes approches pour l'étude du comportement en identifiant l'approche éthologique comme la plus pertinente pour notre recherche.

Ensuite, on a présenté l'état de l'art parmi des travaux reliés à l'utilisation des systèmes optiques 3D, notamment les domaines de la reconnaissance des actions et celui de la modélisation du comportement microscopique du piéton. Parmi ceux-ci, dans le groupe de recherche du laboratoire ATC, on identifie un travail focalisé sur la collection des trajectoires dans un centre commercial pendant une année pour faire une analyse du comportement macroscopique. Ce travail est particulièrement intéressant pour nous, tout d'abord parce que l'étude statique réalisée est très similaire aux analyses faites au sein de l'entreprise Shopline et d'autre part, parce qu'il propose une nouvelle vision pour étudier les trajectoires des personnes en analysant l'utilisation de l'espace.

Cette analyse de l'état de l'art associée aux besoins industriels, l'étude de la relation entre les personnes et l'espace observé ainsi que l'analyse des trajectoires, sont très pertinentes pour identifier le déplacement d'un groupe de personnes. De plus, les méthodes proposées respectent les principes d'évolutivité, d'exécution en temps-réel et d'adaptation multi-capteurs dans la conception de méthodes d'analyses comportementales.

Nous avons concentré notre travail sur la création de méthodes et outils pour analyser le comportement humain à partir des données obtenues de la solution proposée dans le chapitre 3. On considère cette solution comme une approche de type éthologique du fait que l'on fait une observation objective des mouvements des personnes dans leur environnement naturel d'une manière non-intrusive. Nous nous sommes focalisés plus particulièrement sur la relation entre les personnes et l'espace observé, c'est-à-dire l'utilisation de l'espace et l'analyse de trajectoires. Ceci a été réalisé en respectant les principes d'évolutivité, d'exécution en temps-réel et d'adaptation multi-capteurs dans la conception de méthodes d'analyse comportementale.

Pour ce faire, on propose pour l'analyse de l'utilisation de l'espace : la détection de personnes en temps-réel dans les zones d'intérêt, l'estimation des cartes de chaleur, d'occupation, d'entrées et de sorties. Pour l'analyse de trajectoires on utilise les méthodes VFKM et AFKM en proposant les attributs requis pour segmenter les trajectoires des personnes. Nous avons testé nos méthodes en utilisant 3 jeux de données obtenus par des caméras stéréo passives et actives et dans des approches de caméra intelligentes autonomes et de multi-caméras.

A partir des méthodes proposées, on a écarté l'utilisation de la carte de chaleur et la segmentation par VFKM en raison de mauvais résultats lors de notre évaluation. Le reste des méthodes a obtenu des résultats satisfaisants.

De manière générale, ces méthodes nous permettent de :

- Générer des alarmes en temps-réel par rapport au nombre de personnes et à la durée de leur séjour à l'intérieur des zones d'intérêt de la scène.
- Identifier les régions d'utilisation et leur fréquentation en nous permettant à la fois d'estimer la pertinence des zones d'intérêt évaluées par la méthode de « détection de personnes en temps-réel dans les zones d'intérêt » et à la fois de comprendre la dynamique de la scène pour la prise de décisions sur la configuration ou l'utilisation de l'espace.
- Explorer l'utilisation de l'espace à différents niveaux en utilisant les seuils de temps d'utilisation et le facteur de réduction.
- Identifier les régions les plus fréquentées et la route la plus utilisée dans la scène.
- Identifier les points d'accès les plus fréquentés de la scène et dans certains cas de l'infrastructure où le système a été mis en place.
- Estimer le mouvement global de la scène et différencier les flux comportementaux selon les besoins de l'utilisateur à partir du nombre de clusters.
- Grouper les trajectoires avec des trajets similaires qui appartiennent au même flux.
- Visualiser les trajectoires et les séquences de vidéos à distance pour améliorer la configuration des paramétrages des caméras.

En somme, notre solution est capable de faire un premier niveau d'étude du comportement avec les méthodes proposées, en se focalisant sur l'utilisation de l'espace et sur les flux comportementaux. De plus, notre approche peut servir à un autre type d'expériences de comportement comme le « conditionnement », pour évaluer la motivation ou de type psychologique ou « behavioriste » grâce à sa capacité à extraire des informations riches en caractéristiques et en grand volume, qui finalement permettent l'unification des approches des sciences du comportement.

Chapitre 6

Conclusions et perspectives

Sommaire

6.1 Conclusions

Dans ce travail de thèse, nous avons exploré des voies technologiques et scientifiques afin d'obtenir des informations fiables et viables (industriellement) pour étudier le comportement de personnes en mouvement dans les grands espaces. La solution devait être facile à déployer, configurable et à faible coût, dans le but d'étudier l'utilisation d'un espace commercial par rapport aux trajectoires des clients potentiels avec une application visant le comptage de ces derniers.

Au fil de nos travaux, nous avons eu à surmonter plusieurs verrous technologiques et scientifiques, pour accomplir les objectifs du cahier des charges explicités dans l'introduction de ce manuscrit. Nous avons atteint les résultats suivants :

- Après avoir évalué les capteurs disponibles sur le marché par une méthode adaptée à notre domaine, nous avons opté pour le capteur ASUS Xtion Pro, qui est relativement précis et constitue donc un bon compromis entre les différentes caractéristiques. Ce capteur fournit localement les cartes de profondeur les moins bruitées et n'est pas perturbé par les variations de la lumière et des ombres en intérieur. Même s'il est perturbé par la lumière solaire directe, il a une portée acceptable en profondeur et en termes de champ de vision, en comparaison avec les autres capteurs. C'est le capteur optique actif le plus viable du point de vue industriel.
- En second lieu, nous avons conçu et réalisé une caméra intelligente autonome. Celle-ci est capable d'extraire des propriétés spatio-temporelles et physiques des personnes en produisant des données riches qui nous permettent d'identifier et de suivre plusieurs cibles en temps réel. Grâce à son placement, après étude, en position zénithale, nous avons réduit les occultations et diminué la variation d'échelle. Nous avons traité les aspects logiciels, notamment la complexité des algorithmes à implanter et leurs instanciations sur une architecture dont la puissance de calcul permet de garantir les contraintes temporelles particulièrement sur la partie applicative de détection. L'autonomie de cette caméra est assurée par l'intégration des chaînes de traitements (hors ligne et en ligne), et une conception hardware adaptée à une architecture en nœuds distribués. Le traitement hors ligne a permis de reconstruire l'arrière-plan 3D pour permettre la séparation des personnes du fond de la scène et le filtrage des cibles non désirées (enfants, animaux de compagnie ou caddies). Dans la chaîne de traitement en ligne, nous avons traité la séparation entre personnes avec la segmentation à deux niveaux, puis la similitude des gens et la déformation du modèle avec l'utilisation d'un vecteur de caractéristiques humaines (HFD).
- En troisième lieu, nous avons créé un réseau de caméras intelligentes pour étudier le comportement de personnes en mouvement sur de grands espaces. Nous avons conçu ce réseau en assurant une collecte massive de données provenant de plusieurs sources pour en faire des données exploitables. Ce réseau composé de nœuds intelligents ajoute de la robustesse et une impression d'ubiquité, puisque chaque nœud apporte de la puissance de calcul et un champ d'observation pour répondre à un besoin global, permettant d'évaluer de grands espaces.

Notre solution est bien sûr perfectible, car les choix technologiques que nous avons eu à faire étaient très souvent guidés par des contraintes industrielles, de coût, de simplicité de mise en œuvre et enfin de temps de conception et de mise sur le marché.

Si nous devions revoir certains choix, nous pensons que la méthodologie utilisée pour évaluer la pertinence d'une caméra pour la détection des personnes est limitée. Il est difficile de connaître exactement la vérité-terrain qui entre dans le calcul du pourcentage des pixels correctement détectés par les caméras comme appartenant aux personnes observées dans les différentes positions à l'intérieur de la scène. Il est donc nécessaire de pouvoir ajouter d'autres indicateurs à explorer pour les inclure dans le processus d'évaluation et de sélection des capteurs 3D.

Dans la chaîne de traitement, la partie qui introduit le plus d'erreurs est la segmentation des personnes très proches entre elles, dans certains cas spécifiques (quatre scenarii illustrés

dans la fig. 3.32, par exemple l'occlusion de la tête par une personne plus grande). Un moyen pour résoudre ce problème serait la fusion et la séparation des trajectoires comme dans le travail [KAD+14] : lors de la perte de la détection d'une personne avec détection d'une tache irrégulière, on peut marquer cette trajectoire comme appartenant à deux cibles. Il serait intéressant d'explorer d'autres méthodes de suivi comme le filtrage de Kalman ou le filtrage à particules en évaluant leur embarquabilité et leur impact en termes de mémoire et de temps de calcul, tout cela pour respecter les contraintes industrielles.

Notre constat est qu'il manque une méthode quantitative pour évaluer la délégation du suivi des personnes entre les caméras. Même si la génération des trajectoires est réalisée, cette délégation n'est pas évaluée de manière quantitative.

Par ailleurs, une étude précise des limites de la « scalabilité » de notre système en termes de bande passante du réseau et du nombre maximal de nœuds voisins pour une caméra sans affecter sa performance. Pour cela, il faudrait mettre en place un ensemble d'outils de mesure de performance sur le réseau. Une solution plus abordable pourrait être la construction d'un simulateur de réseau de caméras permettant de déployer virtuellement un grand nombre de nœuds (caméras) avec différents scénarios.

En ce qui concerne les analyses de l'utilisation de l'espace et des trajectoires des personnes suivies, on peut étendre ce travail pour fiabiliser les résultats obtenus par notre système. Les erreurs de suivi sont actuellement traduites comme des fausses entrées/sorties dans des espaces inattendus. Nous proposons l'identification de ces erreurs dans les zones de chevauchement entre les caméras comme un indicateur d'efficacité de la délégation du suivi entre les caméras. D'autre part, la connaissance de l'espace global observé par les caméras et les points d'entrées et de sorties des personnes permet de détecter et d'éliminer des trajectoires dont le commencement n'est pas une entrée réelle dans cet espace.

Sur le plan industriel, nous avons réussi le transfert de connaissances à la société Shopline Electronic en augmentant ses capacités techniques pour pouvoir concevoir un système de suivi de personnes en mouvement. Nous avons également mis en évidence les verrous scientifiques à lever et confronté notre solution à la réalité du terrain en termes de faisabilité, de coût de conception, de complexité d'utilisation et de maintenabilité. Nous sommes arrivés à une solution viable techniquement et économiquement, avec une simplicité de mise en œuvre sur le terrain. Notre solution garantit l'exécution des traitements avec un débit jusqu'à 20 images par seconde, suffisamment réactif pour détecter des déplacements rapides avec une précision jusqu'à 99 %.

Nous avons testé nos algorithmes avec différents types de capteurs. Nous avons détecté des problèmes dans la segmentation à deux niveaux et dans l'extraction de HFDs, en raison de la qualité de l'image de profondeur fournie avec beaucoup de pixels invalides et des régions bruitées, rendant difficile la détection de têtes et d'épaules. Le suivi et le comptage de personnes à partir d'une segmentation simple ont été évalués, validés et intégrés au système en utilisant une caméra stéréoscopique passive à bas coût, la caméra d'ETRON et la caméra R200 d'INTEL.

Nos contributions académiques les plus significatives sont :

- La proposition de méthodes étendues pour l'évaluation de performance des capteurs 3D pour la détection des personnes.
- Le suivi de personnes en utilisant la segmentation à deux niveaux et un vecteur de caractéristiques humaines (Human features descriptor : HFD) pour décrire les propriétés pertinentes et observables des personnes.
- La création d'un réseau de caméras intelligentes dotées de puissances de calcul sur les nœuds distribués, ce qui fait de ce dispositif une solution facilement extensible et évolutive.
- La proposition de méthodes pertinentes pour l'étude de l'utilisation de l'espace et l'analyse des trajectoires dans de grandes surfaces.

Nos contributions industrielles ont permis d'apporter les points suivants :

- L'industrialisation d'une caméra IR pour le comptage de personnes.
- L'industrialisation d'une caméra 3D pour le comptage de personnes (basée sur la technologie PrimeSense).
- L'industrialisation d'une caméra stéréoscopique pour le comptage des personnes.
- La conception d'un environnement R&D pour l'évaluation et la mise en œuvre de nouveaux systèmes de comptage de personnes.
- L'aperture de créer un nouveau type de caméra spécialisé sur l'évaluation de l'espace à l'intérieur des espaces commerciaux.

6.2 Perspectives

La richesse de notre sujet d'étude nous permet d'envisager des travaux de réflexion et d'amélioration technologiques et scientifiques sur quatre axes principaux : le capteur, la conception de la caméra intelligente, le réseau de caméras et les outils d'exploration de grosses bases de données pour l'étude du comportement des personnes en mouvement.

Au niveau du capteur, l'arrêt de commercialisation de la caméra ASUS Xtion PRO nous oblige à remplacer le capteur avec des caractéristiques similaires, comme la caméra R200 d'INTEL ou une solution stéréoscopique propriétaire qui nous permet d'agrandir le champ de vision et donc meilleure que la solution Intel. Compte tenu des résultats obtenus avec d'autres capteurs, nous devons adapter nos algorithmes à des images de profondeur moins précises pour profiter de la totalité des méthodes proposées dans le présent travail. On peut envisager un lissage sur les zones remplies et une érosion de la valeur de profondeur sur les pixels invalides en utilisant l'image d'entrée couleur. Dans le cas où l'on décide d'utiliser la caméra R200, on doit réévaluer le bloc d'auto-positionnement puisque cette caméra n'est pas capable d'estimer la profondeur du sol qui est habituellement en dehors de sa portée en profondeur. On pourra, par exemple, utiliser les positions d'une personne dans différentes parties de la scène pour estimer le plan du sol. Ces améliorations peuvent nous permettre d'élargir le cadre de fonctionnement vers un système hybride multi-capteur et multi-caméra.

Au niveau de la caméra intelligente autonome et de sa performance, on peut aller plus loin en utilisant des traitements massivement parallèles dans les unités GPU (qui sont déjà inclus dans notre système embarqué mais non exploités actuellement) sans modifier l'architecture matérielle, avec des prévisions d'augmentation de la performance. En conséquence, on augmente la fréquence de traitement des images et le nombre de points pour décrire les trajectoires, en nous permettant une étude plus fine du comportement humain.

En ce qui concerne les grands espaces, même si notre solution permet d'installer facilement les dispositifs, on pourrait envisager moins de caméras sans perte de fiabilité du système, dans certaines conditions d'organisation des espaces à surveiller. Par exemple, dans les grands couloirs sans issue entre les deux extrémités, une caméra à l'entrée et à la sortie de ce couloir suffirait, puisque les personnes qui y entrent ne peuvent que sortir par les deux extrémités surveillées. Donc, un réseau de caméras éloignées et sans chevauchement entre leurs champs de vision impose d'augmenter fortement la fiabilité et la robustesse des algorithmes d'extraction des caractéristiques observables, qui dépendront de la nouvelle technologie choisie pour obtenir des cartes précises.

En ce qui concerne l'étude du comportement d'une manière générale, on pourrait pousser le développement vers des méthodes d'analyses comportementales plus sophistiquées, utilisant la segmentation d'un flux comportemental unitaire (qui identifie un type de personne en particulier), la recherche de séquences régulières ou la construction d'éthogrammes.

En ce qui concerne l'étude sur l'utilisation de l'espace, l'ajout de services de reconnaissance d'objets externes constituerait une amélioration pour mieux comprendre le contexte de la scène (comme les produits sur les étagères). L'intérêt d'externaliser cette fonctionnalité est de pas alourdir les algorithmes de notre caméra et de profiter des architectures de type Cloud pour faire ces tâches qui ne sont pas très fréquentes et qui requièrent une haute puissance de calcul hors réseau des nœuds de caméras intelligentes.

Il serait intéressant de construire un outil pour la comparaison automatique des points d'intérêt attendus versus les points trouvés, et de les analyser dans une dimension temporelle. L'analyse de cette évolution permettrait une exploitation optimale des zones d'intérêt vis-à-vis des objectifs de l'exploitant (de ces données), par exemple la promotion de nouveaux produits.

Par ailleurs, une autre direction de travail concerne l'exploration de données type « Big data » et l'identification de personnels ou de produits dans les magasins en profitant du fait que l'entreprise Shopline Electronic dispose de milliers de dispositifs de comptage installés dans

différents environnements et conditions. On pourrait envisager l'utilisation de la grande quantité de données de manière massive pour générer des informations d'un haut niveau de généralisation et ainsi mieux comprendre la dynamique entre les visiteurs, les ventes et le contexte du magasin. Cette compréhension peut améliorer les résultats économiques et assurer la réussite des enseignes commerciales, clientes de Shopline Electronic, et espérer positionner l'entreprise comme leader du comptage.

Une des problématiques actuelles de l'entreprise Shopline Electronic est la reconnaissance du personnel des surfaces commerciales qui ne doit pas être considéré comme des visiteurs ou des clients. Il faudrait éviter de les compter pour améliorer davantage la précision du système. Il serait aussi important de localiser des produits spécifiques pour évaluer le comportement des personnes vis-à-vis de ces produits. Pour résoudre ces deux problématiques, nous pourrions utiliser des solutions basées sur des balises Bluetooth pour le suivi du personnel et le positionnement des produits spécifiques. En effet, certaines solutions actuelles, basées sur des modules Bluetooth, permettent de déclencher des systèmes publicitaires (vidéos ou promotions sur les écrans des magasins) quand les clients interagissent avec les produits. Ces solutions peuvent être intégrées à notre système et servir de base pour étendre les fonctionnalités et capacités d'analyse disponibles. On améliore ainsi les applications pour l'étude du comportement des personnes en mouvement dans de grands espaces.

6.3 Publications

6.3.1 Conférences internationales avec comité de lecture

- Andres Burbano, Marius Vasiliu, Samir Bouaziz. "3D Cameras Benchmark for Human Tracking in Hybrid Distributed Smart Camera Networks". 10th international conference on distributed smart cameras. Paris, France. September 12th-15th, 2016
- *Andres Burbano*. "3D-Sensing Distributed Embedded System for the Study of Human Kinetic Behavior: PhD Forum". 10th international conference on distributed smart cameras. Paris, France. September 12th-15th, 2016.
- Andres Burbano, Samir Bouaziz, Marius Vasiliu. "3D-Sensing Distributed Embedded System for People Tracking and Counting", International Conference on Computational Science and Computational Intelligence, December 2015

6.3.2 Atelier et autres

- Andres Burbano, Samir Bouaziz, Marius Vasiliu. "3D-Sensing Distributed Embedded System for People Tracking and Counting: Fast Cameras extrinsic Calibration and Fast Depth Cameras Integration framework", 5th Workshop on the Architecture of Smart Cameras. Dijon, France. July 4th-5th, 2016.
- Poster : *Andres Burbano*, Samir Bouaziz, Marius Vasiliu. "SHUKB: Study of Human Kinetic Behavior" 10th International computer vision summer school. Catania, Italy. July 18th, 2016.
- Poster : Andres Burbano, "Public People Counting System Based on 3D vision", La journée des doctorants du laboratoire des Systèmes et applications des Technologies de l'Information et de l'Énergie UMR CNRS, Juin 2014

Bibliographie

[ABL] Clemens Arth, Horst Bischof, and Christian Leistner. Tricam-an embedded platform for remote traffic surveillance. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 125–125. IEEE. URL : http://ieeexplore.ieee.org/abstract/document/1640569/. XXIX

[ACC+] L. Albani, Pietro Chiesa, Daniele Covi, G. Pedegani, A. Sartori, and M. Vatteroni. VIsoc : a smart camera SoC. In *Solid-State Circuits Conference, 2002. ESSCIRC 2002. Proceedings of the 28th European*, pages 367–370. IEEE. URL : http://ieeexplore.ieee.org/abstract/document/1471541/. XXIX

[AEL+18] Mohamed Abouzahir, Abdelhafid Elouardi, Rachid Latif, Samir Bouaziz, and Abdelouahed Tajer. Embedding SLAM algorithms : Has it come of age ? *Robotics and Autonomous Systems*, 100 :14–26, 2018. I

[Amy06] Mathieu Amy. Les quatre questions de Tinbergen. *Linx. Revue des linguistes de l'université Paris X Nanterre*, (54) :27–33, 2006. URL : http://linx.revues.org/499, doi:10.4000/linx.499. 113

[ASR+15] Ziv Aviv, David Stanhill, Dror REIF, Roi ZISS, Jeffrey Danowitz, and Ehud Pertzov. Structured stereo, October 2015. International Classification H04N13/04, H04N13/02, H04N13/00 ; Cooperative Classification G06T2207/10048, G06T7/0057, G06T2207/10012, G06T7/0075. URL : http://www.google.fr/patents/WO2015163995A1. 32

[ASS16] Boulbaba Ben Amor, Jingyong Su, and Anuj Srivastava. Action recognition using rate-invariant analysis of skeletal shape trajectories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(1) :1–13, 2016. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7115162. 114

[ATJAK12] R. Amali Therese Jenifa, C. Akila, and V. Kavitha. Rapid background subtraction from video sequences. In *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*, pages 1077–1086. IEEE, 2012. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6203780. 57, 58

[Ban65] Albert Bandura. Influence of models' reinforcement contingencies on the acquisition of imitative responses. *Journal of personality and social psychology*, 1(6) :589, 1965. URL : http://psycnet.apa.org/journals/psp/1/6/589/. 127

[BBD+] Daniel Bauer, Ahmed Nabil Belbachir, Nikolaus Donath, Gerhard Gritsch, Bernhard Kohn, Martin Litzenberger, Christoph Posch, Peter Schön, and Stephan Schraml. Embedded vehicle speed estimation system using an asynchronous temporal contrast vision sensor. 2007(1) :34–34. URL : http://dl.acm.org/citation.cfm?id=1287539. XXIX

[BBHK06] Raymond C. Browning, Emily A. Baker, Jessica A. Herron, and Rodger Kram. Effects of obesity and sex on the energetic cost and preferred speed of walking. *Journal of Applied Physiology*, 100(2) :390–398, February 2006. URL : http://jap.physiology.org/content/100/2/390, doi:10.1152/japplphysiol.00767.2005. 9, 74

[BBRS] Michael Bramberger, Josef Brunner, Bernhard Rinner, and Helmut Schwabach. Real-time video analysis on an embedded smart camera for traffic surveillance. In *Real-Time and Embedded Technology and Applications Symposium, 2004. Proceedings. RTAS 2004. 10th IEEE*, pages 174–181. IEEE. URL : http://ieeexplore.ieee.org/abstract/document/1317262/. XXIX

[BDM+] Michael Bramberger, Andreas Doblander, Arnold Maier, Bernhard Rinner, and Helmut Schwabach. Distributed embedded smart cameras for surveillance applications. 39(2) :68–75. URL : http://ieeexplore.ieee.org/abstract/document/1597091/. 3, XXIX

[BE06] James Black and Tim Ellis. Multi camera image tracking. *Image and Vision Computing*, 24(11) :1256–1267, November 2006. URL : http://www.sciencedirect.com/science/article/pii/S0262885605000806, doi:10.1016/j.imavis.2005.06.002. 102

[Ber95] Dirk Bergmann. New approach for automatic surface reconstruction with coded light. In *SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation*, pages 2–9. International Society for Optics and Photonics, 1995. URL : http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1007846. 24

[BES06] Keni Bernardin, Alexander Elbs, and Rainer Stiefelhagen. Multiple object tracking performance metrics and evaluation in a smart room environment. In *Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV*, volume 90, page 91. Citeseer, 2006. URL : http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.7070&rep=rep1&type=pdf. 54

[BK87] Kim L. Boyer and Avinash C. Kak. Color-encoded structured light for rapid active ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1) :14–28, 1987. URL : http://ieeexplore.ieee.org/abstract/document/4767869/. 24

[BK15] Dražen Brščić and Takayuki Kanda. Changes in usage of an indoor public space : Analysis of one year of person tracking. *IEEE Transactions on Human-Machine Systems*, 45(2) :228–237, 2015. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6981971. 113, 114, 115, 118, 120, 122, 125, 130

[BKIM13] D. Brščić, T. Kanda, T. Ikeda, and T. Miyashita. Person Tracking in Large Public Spaces Using 3-D Range Sensors. *IEEE Transactions on Human-Machine Systems*, 43(6) :522–534, November 2013. doi:10.1109/THMS.2013.2283945. 4, 27, 28, 45, 72, 100, 102, 107, 108, 114

[BKMQB16] Lobna Ben Khelifa, Luca Maggiani, Jean-Charles Quinton, and François Berry. Ant-Cams Network : A Cooperative Network Model for Silly Cameras. In *Proceedings of the 10th International Conference on Distributed Smart Camera*, ICDSC '16, pages 104–109, New York, NY, USA, 2016. ACM. URL : http://doi.acm.org/10.1145/2967413.2967437, doi:10.1145/2967413.2967437. 54

[BLM+] J. Boice, X. Lu, C. Margi, G. Stanek, G. Zhang, R. Manduchi, and K. Obraczka. Meerkats : A power-aware, self-managing wireless camera network for wide area monitoring. In *Proc. Workshop on Distributed Smart Cameras*, pages 393–422. URL : https://pdfs.semanticscholar.org/f429/3c399e46cdfadda822adbcb28e1f5f9ebd3a.pdf. XXIX

[BM92] Serge Beucher and Fernand Meyer. The morphological approach to segmentation : the watershed transformation. *Optical Engineering-New York-Marcel Dekker Incorporated-*, 34 :433–433, 1992. URL : https://www.researchgate.net/profile/Serge_Beucher/publication/233950923_Segmentation_The_Watershed_Transformation_Mathematical_Morphology_in_Image_Processing/links/55f7c6ce08aeba1d9efe4072/Segmentation-The-Watershed-Transformation-Mathematical-Morphology-in-Image-Proces pdf. 66

[BMKB12] Sagi Ben Moshe, Ron Kimmel, and Michael Bronstein. Method and system for structured light 3d camera, August 2012. U.S. Classification 348/369, 359/225.1, 348/E05.04 ; International Classification G02B26/08, H04N5/238 ; Cooperative Classification G01S17/48, G02B26/0833 ; European Classification G02B26/08M4. URL : http://www.google.fr/patents/US20120218464. 32

[BMNK13] Kai Berger, Stephan Meister, Rahul Nair, and Daniel Kondermann. A state of the art report on kinect sensor setups in computer vision. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 257–272. Springer, 2013. URL : http://link.springer.com/chapter/10.1007/978-3-642-44964-2_12. 45

[BOHS14] Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. Ten years of pedestrian detection, what have we learned ? In *European Conference on Computer Vision*, pages 613–627. Springer, 2014. URL : http://link.springer.com/chapter/10.1007/978-3-319-16181-5_47. 51, 53, 58

[Bov64] Jacques Lucien Bovet. Position de l'éthologie au sein des sciences du comportement. *Bulletin de la Société vaudoise des sciences naturelles*, 68 :474–482, 1964. 112, 113

[BSA06] Alessandro Bevilacqua, Luigi Di Stefano, and Pietro Azzari. People tracking using a time-of-flight depth sensor. In *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on*, pages 89–89. IEEE, 2006. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4020748. 58, 74

[BV14] Christophe Bobda and Senem Velipasalar, editors. *Distributed Embedded Smart Cameras : Architectures, Design and Applications*. Springer-Verlag, New York, 2014. URL : //www.springer.com/gp/book/9781461477044. 3

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11) :1222–1239, November 2001. URL : http://dx.doi.org/10.1109/34.969114, doi:10.1109/34.969114. 21

[CC92] C. Chang and S. Chatterjee. Quantization error analysis in stereo vision. In *[1992] Conference Record of the Twenty-Sixth Asilomar Conference on Signals, Systems Computers*, pages 1037–1041 vol.2, October 1992. doi:10.1109/ACSSC.1992.269140. 22

[CCH11] Yao-Jen Chang, Shu-Fang Chen, and Jun-Da Huang. A Kinect-based system for physical rehabilitation : A pilot study for young adults with motor disabilities. *Research in developmental disabilities*, 32(6) :2566–2570, 2011. URL : http://www.sciencedirect.com/science/article/pii/S0891422211002587. 34

[CCWC12] Chao-Ho Chen, Tsong-Yi Chen, Da-Jinn Wang, and Tsang-Jie Chen. A cost-effective people-counter for a crowd of moving people based on two-stage segmentation. *Journal of Information Hiding and Multimedia Signal Processing*, 3(1) :12–25, 2012. URL : http://www.jihmsp.org/~jihmsp/2012/vol3/JIH-MSP-2012-01-002.pdf. 57

[CH85] Brian Carrihill and Robert Hummel. Experiments with the intensity ratio depth sensor. *Computer vision, graphics, and image processing*, 32(3) :337–358, 1985. URL : http://www.sciencedirect.com/science/article/pii/0734189X85900568. 24

[CHCW97] Chu-Song Chen, Yi-Ping Hung, Chiann-Chu Chiang, and Ja-Ling Wu. Range data acquisition using color structured lighting and stereo vision. *Image and Vision Computing*, 15(6) :445–456, 1997. URL : http://www.sciencedirect.com/science/article/pii/S0262885696011481. 24

[Che02] Xing Chen. *Design of Many-Camera Tracking Systems for Scalability and Efficient Resource Allocation*. PhD thesis, Stanford University, 2002. URL : http://graphics.stanford.edu/papers/xcchen_thesis/. 6, 9

[Che03] Thou-Ho Chen. An automatic bi-directional passing-people counting method based on color image processing. In *Procs. 37th Inter Carnahan Conf. on Security Technology*, pages 200–207. IEEE, 2003. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1297560. 25, 57

[Che14] Muhammad Shahzad Cheema. *Efficient Human Activity Recognition in Large Image and Video Databases*. PhD thesis, Universitäts-und Landesbibliothek Bonn, 2014. URL : https://www.researchgate.net/profile/Muhammad_Shahzad_Cheema/publication/271273405_Efficient_Human_Activity_Recognition_in_Large_Image_and_Video_Databases/links/54c4711e0cf256ed5a949ef6.pdf. 4, 6

[CLC+13] Xiujuan Chai, Guang Li, Xilin Chen, Ming Zhou, Guobin Wu, and Hanjing Li. Visualcomm A tool to support communication between deaf and hearing persons with the kinect. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, page 76. ACM, 2013. URL : http://dl.acm.org/citation.cfm?id=2513398. 25

[Dal06] Navneet Dalal. *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2006. URL : https://tel.archives-ouvertes.fr/tel-00390303/. 6, 50, 52, 58, XXVII, XXVIII

[DBC14] Cosimo Distante, Sebastiano Battiato, and Andrea Cavallaro, editors. *Video Analytics for Audience Measurement*, volume 8811 of *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 2014. URL : http://link.springer.com/10.1007/978-3-319-12811-5. 4

[DBSM] Fabio Dias, François Berry, Jocelyn Sérot, and François Marmoiton. Hardware, design and implementation issues on a FPGA-based smart camera. In *Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on*, pages 20–26. IEEE. URL : http://ieeexplore.ieee.org/abstract/document/4357501/. XXIX

[DMZC12] Carlo Dal Mutto, Pietro Zanuttigh, and Guido M. Cortelazzo. *Time-of-flight cameras and microsoft Kinect™*. Springer Science & Business Media, 2012. URL : https://books.google.fr/books?hl=en&lr=&id=0ga3V4JBYmIC&oi=fnd&pg=PR3&dq=carlo+dal+mutto&ots=G51lZ0nV1s&sig=cwp5RRFPbZ87Rsr9Friqsafx5kM. 14, 20, 21

[DSB99] Luigi Di Stefano and Andrea Bulgarelli. A simple and efficient connected components labeling algorithm. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 322–327. IEEE, 1999. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=797615. 68

[DSE+13] B. Dieber, J. Simonjan, L. Esterle, B. Rinner, G. Nebehay, R. Pflugfelder, and G.J. Fernandez. Ella Middleware for multi-camera surveillance in heterogeneous visual sensor networks. In *2013 Seventh International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, October 2013. doi:10.1109/ICDSC.2013.6778223. 54, 102, 109

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467360. 51, 52, 65, 67

[DWB+15] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. 3-D human action recognition by shape analysis of motion trajectories on Riemannian manifold. *Cybernetics, IEEE Transactions on*, 45(7) :1340–1352, 2015. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6894548. 114

[DWSP09] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection : A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5206631. 51, 53, 85, 86

[DWSP12] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection : An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, 34(4) :743–761, 2012. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5975165. 6, 52

[EG09] Markus Enzweiler and Dariu M. Gavrila. Monocular pedestrian detection : Survey and experiments. *IEEE transactions on pattern analysis and machine intelligence*, 31(12) :2179–2195, 2009. URL : http://ieeexplore.ieee.org/abstract/document/4657363/. 51

[ELSVG08] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2008. URL : http://ieeexplore.ieee.org/abstract/document/4587581/. 51

[ELYR14] Lukas Esterle, Peter R. Lewis, Xin Yao, and Bernhard Rinner. Socio-economic vision graph generation and handover in distributed smart camera networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(2) :20, 2014. URL : http://dl.acm.org/citation.cfm?id=2530001. 50

[FBBS06] S. Fleck, F. Busch, P. Biber, and W. Straber. 3d Surveillance A Distributed Network of Smart Cameras for Real-Time Tracking and its Visualization in 3d. In *Conference on Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06*, pages 118–118, June 2006. doi:10.1109/CVPRW.2006.6. 54, 100, 102, 109

[FBLF08] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2) :267–282, February 2008. URL : http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4359319, doi:10.1109/TPAMI.2007.1174. 102

[FBSs] Sven Fleck, Florian Busch, and Wolfgang Stra{\textbackslash}s ser. Adaptive probabilistic tracking embedded in smart cameras for distributed surveillance in a 3d model. 2007(1) :24–24. URL : http://dl.acm.org/citation.cfm?id=1287529. XXIX

[Fer15] Nivan Ferreira. *Visual analytics techniques for exploration of spatiotemporal data*. PhD thesis, Polytechnic Institute of New York University, 2015. URL : http://search.proquest.com/openview/61bdc7ad67a4f9c92b71e6aa5635e7fa/1?pq-origsite=gscholar&cbl=18750&diss=y. 125, 127, XVI, XVII

[FGMR10] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9) :1627–1645, 2010. URL : http://ieeexplore.ieee.org/abstract/document/5255236/. 58, 67

[FH75] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1) :32–40, January 1975. doi:10.1109/TIT.1975.1055330. 66

[FKSS13] Nivan Ferreira, James T. Klosowski, Carlos E. Scheidegger, and Cláudio T. Silva. Vector Field k-Means : Clustering Trajectories by Fitting Multiple Vector Fields. In *Computer Graphics Forum*, volume 32, pages 201–210. Wiley Online Library, 2013. URL : http://onlinelibrary.wiley.com/doi/10.1111/cgf.12107/full. 117, 125, 127, 145, XV, XVI, XIX

[FML11] Huiyuan Fu, Huadong Ma, and Liang Liu. Robust Human Detection with Low Energy Consumption in Visual Sensor Network. In *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pages 91–97, December 2011. doi:10.1109/MSN.2011.84. 54

[FMR08] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4587597. 52

[FMX12] Huiyuan Fu, Huadong Ma, and Hongtian Xiao. Real-time accurate crowd counting based on RGB-D information. In *2012 19th IEEE International Conference on Image Processing (ICIP)*, pages 2685–2688, September 2012. doi:10.1109/ICIP.2012.6467452. 25

[FP02] David A. Forsyth and Jean Ponce. *Computer vision : a modern approach*. Prentice Hall Professional Technical Reference, 2002. URL : http://dl.acm.org/citation.cfm?id=580035. 107

[FPF99] Andrew Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5) :476–480, 1999. URL : http://ieeexplore.ieee.org/abstract/document/765658/. 73

[GBMH15] D. F. Glas, D. Brscic, T. Miyashita, and N. Hagita. SNAPCAT 3d : Calibrating networks of 3d range sensors for pedestrian tracking. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 712–719, May 2015. doi:10.1109/ICRA.2015.7139257. 102, 108, 114

[GG13] František Galčík and Radoslav Gargalík. Real-Time Depth Map Based People Counting. In *Advanced Concepts for Intelligent Vision Systems*, volume 8192 of *ACIVS 2013*, pages 330–341, New York, NY, USA, 2013. Springer International Publishing. URL : http://dx.doi.org/10.1007/978-3-319-02895-8_30, doi:10.1007/978-3-319-02895-8_30. 25, 57, 58, 62, 66, 72

[Gib13] Michael P. Gibbens. *Caldag 2013 : An Interpretive Manual and Checklist*. International Code Council, September 2013. 62

[GLSU13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets Robotics : The KITTI Dataset. *The International Journal of Robotics Research*, page 0278364913491297, August 2013. URL : http://ijr.sagepub.com/content/early/2013/08/22/0278364913491297, doi:10.1177/0278364913491297. 35, 51, 52

[GMPK+14] Alex Gomez-Marin, Joseph J. Paton, Adam R. Kampff, Rui M. Costa, and Zachary F. Mainen. Big behavioral data : psychology, ethology and the foundations of neuroscience. *Nature neuroscience*, 17(11) :1455–1462, 2014. URL : https://www.nature.com/neuro/journal/v17/n11/abs/nn.3812.html. 112, 113

[GWT+89] Claire C. Gordon, R. A. Walker, I. Tebbetts, J. T. McConville, B. Bradtmiller, C. E. Clauser, and T. Churchill. *1988 Anthropometric Survey of US Army Personnel-Methods and Summary Statistics. Final Report*. 1989. 72

[HD04] S. P. Hoogendoorn and Winnie Daamen. Design assessment of Lisbon transfer stations using microscopic pedestrian simulation. *WIT Transactions on The Built Environment*, 74, 2004. 114

[HJ09] Dirk Helbing and Anders Johansson. Pedestrian, Crowd and Evacuation Dynamics. In Robert A. Meyers Ph. D, editor, *Encyclopedia of Complexity and Systems Science*, pages 6476–6495. Springer New York, 2009. DOI : 10.1007/978-0-387-30440-3_382. URL : http://link.springer.com/referenceworkentry/10.1007/978-0-387-30440-3_382. 114

[HKH+12] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping : Using Kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5) :647–663, 2012. URL : http://journals.sagepub.com/doi/abs/10.1177/0278364911434148. 35, 50

[HM95] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5) :4282–4286, May 1995. URL : https://link.aps.org/doi/10.1103/PhysRevE.51.4282, doi:10.1103/PhysRevE.51.4282. 114, 115

[HPFA] Stephan Hengstler, Daniel Prashanth, Sufen Fong, and Hamid Aghajan. MeshEye : a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 360–369. ACM. URL : http://dl.acm.org/citation.cfm?id=1236406. XXIX

[HS97] Janne Heikkila and Olli Silvén. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=609468. 18, 22

[HS06] Thomas A. Henzinger and Joseph Sifakis. The Embedded Systems Design Challenge. In Jayadev Misra, Tobias Nipkow, and Emil Sekerinski, editors, *FM 2006 : Formal Methods*, number 4085 in Lecture Notes in Computer Science, pages 1–15. Springer Berlin Heidelberg, August 2006. URL : http://link.springer.com/chapter/10.1007/11813040_1. 3

[HSXS13] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor : A review. *Cybernetics, IEEE Transactions on*, 43(5) :1318–1334, 2013. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6547194. 50

[JC94] J. a. M. Jennings and W. N. Charman. Depth resolution in stereoscopic systems. *Appl. Opt., AO*, 33(22) :5192–5196, August 1994. URL : http://www.osapublishing.org/abstract.cfm?uri=ao-33-22-5192, doi:10.1364/AO.33.005192. 22

[KAD+14] Nathan Kirchner, Alen Alempijevic, X. Dai, P. Plöger, and R. K. Venkat. A robust people detection, tracking, and counting system. In *International Conference on Robotics and Automation*. IEEE, 2014. URL : http://www.araa.asn.au/acra/acra2014/papers/pap122.pdf. 25, 26, 27, 54, 57, 58, 65, 74, 151

[KASD] Richard Kleihorst, Anteneh Abbo, Ben Schueler, and Alexander Danilin. Camera mote with a high-performance parallel processor for real-time frame-based video processing. In *Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on*, pages 109–116. IEEE. URL : http://ieeexplore.ieee.org/abstract/document/4357513/. XXIX

[KAV12] N. Kirchner, A. Alempijevic, and A. Virgona. Head-to-shoulder signature for person recognition. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1226–1231, May 2012. doi:10.1109/ICRA.2012.6224901. 54, 57, 65, 68, 72

[KAW08] Bahador Khaleghi, Siddhant Ahuja, and QM Jonathan Wu. An improved real-time miniaturized embedded stereo vision system (MESVS-II). In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4563144. 41

[KCCK02] Jae-Won Kim, Kang-Sun Choi, Byeong-Doo Choi, and Sung-Jea Ko. Real-time vision-based people counting system for the security door. In *International Technical Conference on Circuits/Systems Computers and Communications*, pages 1416–1419, 2002. 57

[KCCVZ15] Nathan Kirchner, Sonja Caraian, Peter Colborne-Veel, and Michelle Zeibots. Influencing Passenger Egress to Reduce Congestion at Rail Stations. 2015. URL : https://pdfs.semanticscholar.org/538b/2dbb3b29e8492e9bab0c48a196c2e070a82d.pdf. 102, 115

[KHM+00] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-camera multi-person tracking for easyliving. In *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, pages 3–10. IEEE, 2000. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=856852. 35, 102

[Kit14] Rob Kitchin. The real-time city? Big data and smart urbanism. *GeoJournal*, 79(1) :1–14, 2014. URL : http://link.springer.com/article/10.1007/S10708-013-9516-8. 5

[KLG09] Christoph Gustav Keller, David Fernández Llorca, and Dariu M. Gavrila. Dense stereo-based ROI generation for pedestrian detection. In *Joint Pattern Recognition Symposium*, pages 81–90. Springer, 2009. URL : http://link.springer.com/chapter/10.1007/978-3-642-03798-6_9. 51

[KNO11] Mikko Kytö, Mikko Nuutinen, and Pirkko Oittinen. Method for measuring stereo camera depth accuracy based on stereoscopic vision. page 78640I, January 2011. URL : http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.872015, doi:10.1117/12.872015. 41

[KO94] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window : Theory and experiment. *IEEE transactions on pattern analysis and machine intelligence*, 16(9) :920–932, 1994. URL : http://ieeexplore.ieee.org/abstract/document/310690/. 21, 22

[LBPF11] Qiang Li, Moyuresh Biswas, Mark R. Pickering, and Michael R. Frater. Accurate depth estimation using structured light and passive stereo disparity estimation. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 969–972. IEEE, 2011. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6116723. 32

[LCS+11] Belinda Lange, Chien-Yen Chang, Evan Suma, Bradley Newman, Albert Skip Rizzo, and Mark Bolas. Development and evaluation of low cost game-based balance rehabilitation tool using the Microsoft Kinect sensor. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 1831–1834. IEEE, 2011. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6090521. 5

[LHW07] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory Clustering : A Partition-and-group Framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pages 593–604, New York, NY, USA, 2007. ACM. URL : http://doi.acm.org/10.1145/1247480.1247546, doi:10.1145/1247480.1247546. 117, XIX

[LLCC13] Jun Liu, Ye Liu, Ying Cui, and Yan Qiu Chen. Real-time human detection and tracking in complex environments using single RGBD camera. In *2013 20th IEEE International Conference on Image Processing (ICIP)*, pages 3088–3092, September 2013. doi:10.1109/ICIP.2013.6738636. 54

[LNL+13] Damien Lefloch, Rahul Nair, Frank Lenzen, Henrik Schäfer, Lee Streeter, Michael J. Cree, Reinhard Koch, and Andreas Kolb. Technical foundation and calibration methods for time-of-flight cameras. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 3–24. Springer, 2013. URL : http://link.springer.com/chapter/10.1007/978-3-642-44964-2_1. 32, 37

[Lu15] Chao-Chun Lu. Image process apparatus, November 2015. International Classification H04N13/02, H04N13/00 ; Cooperative Classification H04N13/0271, H04N13/0296, H04N13/0292, H04N13/0203, H04N13/004, H04N13/0239, H04N2013/0092, H04N13/0022. URL : http://www.google.fr/patents/US20150319425. 31

[LZ99] Charles Loop and Zhengyou Zhang. Computing rectifying homographies for stereo vision. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 125–131. IEEE, 1999. URL : http://ieeexplore.ieee.org/abstract/document/786928/. 21

[LZL+15] Jun Liu, Guyue Zhang, Ye Liu, Luchao Tian, and Yan Qiu Chen. An ultra-fast human detection method for color-depth camera. *Journal of Visual Communication and Image Representation*, 31 :177–185, 2015. URL : http://www.sciencedirect.com/science/article/pii/S1047320315001133. 54, 57, 58

[MA13] Cyrille Migniot and Fakhreddine Ababsa. 3d human tracking from depth cue in a buying behavior analysis context. In *Computer Analysis of Images and Patterns*, pages 482–489. Springer, 2013. URL : http://link.springer.com/chapter/10.1007/978-3-642-40261-6_58. 114

[MA16] Cyrille Migniot and Fakhreddine Ababsa. Hybrid 3d–2d human tracking in a top view. *J Real-Time Image Proc*, 11(4) :769–784, April 2016. URL : https://link.springer.com/article/10.1007/s11554-014-0429-7, doi:10.1007/s11554-014-0429-7. 114

[Mat13] Stefano Mattoccia. Stereo vision algorithms for fpgas. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 636–641, 2013. URL : http://www.cv-foundation.org/openaccess/content_cvpr_workshops_2013/W10/html/Mattoccia_Stereo_Vision_Algorithms_2013_CVPR_paper.html. 21

[MB99] Moorhead and Binnie. Smart CMOS camera for machine vision applications. In *Image Processing and Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465)*, volume 2, pages 865–869 vol.2, 1999. doi:10.1049/cp:19990448. 54, XXIX

[MKS89] Larry Matthies, Takeo Kanade, and Richard Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3) :209–238, 1989. URL : http://link.springer.com/article/10.1007/BF00133032. 21, 22

[MKS+16] Marion Morel, Richard Kulpa, Anthony Sorel, Catherine Achard, and Séverine Dubuisson. Automatic and Generic Evaluation of Spatial and Temporal Errors in Sport Motions. In *11th International Conference on Computer Vision Theory and Applications (VISAPP 2016)*, pages 542–551, Rome, Italy, February 2016. URL : https://hal.archives-ouvertes.fr/hal-01373822, doi:10.5220/0005778505420551. 9

[MM14] Stefano Mattoccia and Paolo Macri. A Real Time 3d Sensor for Smart Cameras. In *Proceedings of the International Conference on Distributed Smart Cameras*, page 29. ACM, 2014. URL : http://dl.acm.org/citation.cfm?id=2659058. 31, 35, 54

[MP15] Stefano Mattoccia and Matteo Poggi. A passive RGBD sensor for accurate and real-time depth sensing self-contained into an FPGA. In *Proceedings of the 9th International Conference on Distributed Smart Camera*, pages 146–151. ACM, 2015. URL : http://dl.acm.org/citation.cfm?id=2789148. 31, 35

[MPK] Henry Medeiros, Johnny Park, and Avinash Kak. A light-weight event-driven protocol for sensor clustering in wireless camera networks. In *Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on*, pages 203–210. IEEE. URL : http://ieeexplore.ieee.org/abstract/document/4357525/. XXIX

[MPOM] Cintia B. Margi, Vladislav Petkov, Katia Obraczka, and Roberto Manduchi. Characterizing energy consumption in a visual sensor network testbed. In *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on*, pages 8–pp. IEEE. URL : http://ieeexplore.ieee.org/abstract/document/1649166/. XXIX

[MR13] Kenneth David Mankoff and Tess Alethea Russo. The Kinect : a low-cost, high-resolution, short-range 3d camera. *Earth Surface Processes and Landforms*, 38(9) :926–936, 2013. URL : http://onlinelibrary.wiley.com/doi/10.1002/esp.3332/full. 45

[MSD17] Robert D. MacDougall, Benoit Scherrer, and Steven Don. Development of a tool to aid the radiologic technologist using augmented reality and computer vision. *Pediatr Radiol*, pages 1–5, September 2017. URL : https://link.springer.com/article/10.1007/s00247-017-3968-9, doi:10.1007/s00247-017-3968-9. XI

[Mun57] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1) :32–38, 1957. 108

[NABF13] Luca Neri, Giulia Adorante, Gianni Brighetti, and Elena Franciosi. Postural rehabilitation through Kinect-based biofeedback. In *Virtual Rehabilitation (ICVR), 2013 International Conference on*, pages 218–219. IEEE, 2013. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6662110. 34

[NML+13] Rahul Nair, Stephan Meister, Martin Lambers, Michael Balda, Hannes Hofmann, Andreas Kolb, Daniel Kondermann, and Bernd Jähne. Ground truth for evaluating time of flight imaging. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 52–74. Springer, 2013. URL : http://link.springer.com/chapter/10.1007/978-3-642-44964-2_4. 34

[NP11] Taewoo Nam and Theresa A. Pardo. Conceptualizing Smart City with Dimensions of Technology, People, and Institutions. In *Proceedings of the 12th Annual International Digital Government Research Conference : Digital Government Innovation in Challenging Times*, dg.o '11, pages 282–291, New York, NY, USA, 2011. ACM. URL : http://doi.acm.org/10.1145/2037556.2037602, doi:10.1145/2037556.2037602. 5

[NSMH10] Luis E. Navarro-Serment, Christoph Mertz, and Martial Hebert. Pedestrian Detection and Tracking Using Three-dimensional LADAR Data. *The International Journal of Robotics Research*, May 2010. URL : http://ijr.sagepub.com/content/early/2010/05/18/0278364910370216, doi:10.1177/0278364910370216. 25

[OW13] Wanli Ouyang and Xiaogang Wang. Joint deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2056–2063, 2013. URL : http://www.cv-foundation.org/openaccess/content_iccv_2013/html/Ouyang_Joint_Deep_Learning_2013_ICCV_paper.html. 53

[Ozc14] Aydogan Ozcan. Mobile phones democratize and cultivate next-generation imaging, diagnostics and measurement tools. *Lab Chip*, 14(17) :3187–3194, September 2014. doi:10.1039/c4lc00010b. 14

[PKT14] Emmanouil Potetsianakis, Emmanouil Ksylakis, and Georgios Triantafyllidis. A kinect based framework for better user experience in real-time audiovisual content manipulation. In *Telecommunications and Multimedia (TEMU), 2014 International Conference on*, pages 238–242. IEEE, 2014. URL : http://ieeexplore.ieee.org/abstract/document/6917767/. 35

[PMS04] Anand Panangadan, Maja Mataric, and Gaurav Sukhatme. Detecting anomalous human interactions using laser range-finders. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2136–2141. IEEE, 2004. 114

[PP15] Diana Pagliari and Livio Pinto. Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors. *Sensors*, 15(11) :27569–27589, 2015. URL : http://www.mdpi.com/1424-8220/15/11/27569/htm. 33, 35, 37

[QLYWj10] Zhu Qiuyu, Tang Li, Jiang Yiping, and Deng Wei-jun. A novel approach of counting people based on stereovision and DSP. In *The 2nd Inter. Conf. on Computer and Automation Engineering (ICCAE)*, volume 1, pages 81–84, February 2010. doi:10.1109/ICCAE.2010.5451996. 54, 57, 62

[RAR+11] Caroline Rougier, Edouard Auvinet, Jacqueline Rousseau, Max Mignotte, and Jean Meunier. Fall detection from depth map video sequences. In *Toward Useful Services for Elderly and People with Disabilities*, pages 121–128. Springer, 2011. URL : http://link.springer.com/chapter/10.1007/978-3-642-21535-3_16. 5

[Rau13] Mattias Rauter. Reliable human detection and tracking in top-view depth images. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 529–534. IEEE, 2013. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6595924. 25, 27, 54, 57, 74, 76

[RB94] M. Rossi and A. Bozzoli. Tracking and counting moving people. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 3, pages 212–216. IEEE, 1994. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=413857. 2, 57, 76

[RBI+] Mohammad Rahimi, Rick Baer, Obimdinachi I. Iroezi, Juan C. Garcia, Jay Warrior, Deborah Estrin, and Mani Srivastava. Cyclops : in situ image sensing and interpretation in wireless sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 192–204. ACM. URL : http://dl.acm.org/citation.cfm?id=1098939. XXIX

[RBP15] Nicolas Loy Rodas, Fernando Barrera, and Nicolas Padoy. Marker-Less AR in the Hybrid Room Using Equipment Detection for Camera Relocalization. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pages 463–470. Springer, 2015. URL : http://link.springer.com/chapter/10.1007/978-3-319-24553-9_57. 5

[RG12] Zoltan Tomori Radoslav Gargalik. Object tracking in 3d using depth map. pages 28–31, 2012. 57

[RGGN07] Anthony Rowe, Adam G. Goode, Dhiraj Goel, and Illah Nourbakhsh. Cmucam3 : An open programmable embedded vision sensor. Technical Report CMU-RI-TR-07-13, Carnegie Mellon University, Pittsburgh, PA, May 2007. XXIX

[RGR] Anthony Rowe, Dhiraj Goel, and Raj Rajkumar. Firefly mosaic : A vision-enabled wireless sensor networking system. In *Real-time systems symposium, 2007. RTSS 2007. 28th IEEE international*, pages 459–468. IEEE. URL : http://ieeexplore.ieee.org/abstract/document/4408328/. XXIX

[RJQ07] B. Rinner, M. Jovanovic, and M. Quaritsch. Embedded Middleware on Distributed Smart Cameras. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 4, pages IV–1381–IV–1384, April 2007. doi:10.1109/ICASSP.2007.367336. 54

[RMSB11] Christian Rudloff, Thomas Matyus, Stefan Seer, and Dietmar Bauer. Can walking behavior be predicted? Analysis of calibration and fit of pedestrian models. *Transportation Research Record : Journal of the Transportation Research Board*, (2264) :101–109, 2011. 114

[RMYZ11] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition with kinect sensor. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 759–760. ACM, 2011. URL : http://dl.acm.org/citation.cfm?id=2072443. 34

[ROB+10] Akmal Rakhmadi, Nur ZS Othman, Abdullah Bade, Mohd SM Rahim, and Ismail M. Amin. Connected component labeling using components neighbors-scan labeling approach. *Journal of Computer Science*, 6(10) :1099, 2010. URL : http://thescipub.com/abstract/10.3844/jcssp.2010.1099.1107. 68

[RRRC14] Martin Reisslein, Bernhard Rinner, and Amit Roy-Chowdhury. Smart Camera Networks. *Computer*, 47(5) :23–25, May 2014. URL : http://dx.doi.org/10.1109/MC.2014.134, doi:10.1109/MC.2014.134. 54

[Rus09] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, TECHNISCHE UNIVERSITÄT MÜNCHEN, Munich, September 2009. 6, 35, 107

[RW08] B. Rinner and W. Wolf. An Introduction to Distributed Smart Cameras. *Proceedings of the IEEE*, 96(10) :1565–1575, October 2008. doi:10.1109/JPROC.2008.928742. 3

[RW14] Bernhard Rinner and Thomas Winkler. Privacy-protecting Smart Cameras. In *Proceedings of the International Conference on Distributed Smart Cameras*, page 40. ACM, 2014. URL : http://dl.acm.org/citation.cfm?id=2659044. 54

[RWS+08] Bernhard Rinner, Thomas Winkler, Wolfgang Schriebl, Markus Quaritsch, and Wayne Wolf. The evolution from single to pervasive smart cameras. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–10. IEEE, 2008. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4635674. 3, 54, 55, 101, XXIX

[SA11] Luciano Spinello and Kai O. Arras. People detection in RGB-D data. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3838–3843. IEEE, 2011. URL : http://ieeexplore.ieee.org/abstract/document/6095074/. 65

[SBR12] Stefan Seer, Norbert Brändle, and Carlo Ratti. Kinects and human kinetics : a new approach for studying crowd behavior. *arXiv preprint arXiv :1210.2838*, 2012. URL : http://arxiv.org/abs/1210.2838. 102, 107, 108, 114

[SBR14] Stefan Seer, Norbert Brändle, and Carlo Ratti. Kinects and human kinetics : A new approach for studying pedestrian behavior. *Transportation research part C : emerging technologies*, 48 :212–228, 2014. URL : http://www.sciencedirect.com/science/article/pii/S0968090X14002289. 4, 50, 102, 107, 114

[Seg96] J. Segen. A camera-based system for tracking people in real time. In , *Proceedings of the 13th International Conference on Pattern Recognition, 1996*, volume 3, pages 63–67 vol.3. IEEE, August 1996. doi:10.1109/ICPR.1996.546795. 2

[SKCL13] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3626–3633, 2013. URL : http://www.cv-foundation.org/openaccess/content_cvpr_2013/html/Sermanet_Pedestrian_Detection_with_2013_CVPR_paper.html. 53

[SKK+11] Andreas Schadschneider, Wolfram Klingsch, Hubert Klüpfel, Tobias Kretz, Christian Rogsch, and Armin Seyfried. Evacuation dynamics : Empirical results, modeling and applications. In *Extreme Environmental Events*, pages 517–550. Springer, 2011. 114

[SS14] Reza Sabzevari and Davide Scaramuzza. Monocular simultaneous multi-body motion segmentation and reconstruction from perspective views. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 23–30. IEEE, 2014. URL : http://ieeexplore.ieee.org/abstract/document/6906585/. 58

[ST] Yu Shi and Timothy Tsui. An FPGA-based smart camera for gesture recognition in HCI applications. In *Asian conference on Computer vision*, pages 718–727. Springer. URL : http://link.springer.com/chapter/10.1007/978-3-540-76386-4_68. XXIX

[STS08] Luciano Spinello, Rudolph Triebel, and Roland Siegwart. Multimodal detection and tracking of pedestrians in urban environments with explicit ground plane extraction. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1823–1829. IEEE, 2008. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4651109. 65

[Sze10] Richard Szeliski. *Computer Vision - Algorithms and Applications*. Springer, New York, 2010. URL : http://www.springer.com/us/book/9781848829343. 18, 19, 22, 107, V, VI

[TC89] C.-H. Teh and Roland T. Chin. On the detection of dominant points on digital curves. *IEEE Transactions on pattern analysis and machine intelligence*, 11(8) :859–872, 1989. 73

[TDS10] Thiago Teixeira, Gershon Dublon, and Andreas Savvides. A survey of human-sensing : Methods for detecting presence, count, location, track, and identity. *ACM Computing Surveys*, 5 :427–450, 2010. URL : http://thiagot.com/papers/teixeira_techrep10_survey_of_human_sensing.pdf. 3, 6, 14, 15, 20, 31, 56, 57

[TH86] Qi Tian and Michael N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 35(2) :220–233, 1986. URL : http://www.sciencedirect.com/science/article/pii/0734189X86900289. 22

[Tin63] Niko Tinbergen. On aims and methods of ethology. *Ethology*, 20(4) :410–433, 1963. URL : http://onlinelibrary.wiley.com/doi/10.1111/j.1439-0310.1963.tb01161.x/full. 113

[TJS10] Thiago Teixeira, Deokwoo Jung, and Andreas Savvides. Tasking Networked CCTV Cameras and Mobile Phones to Identify and Localize Multiple People. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, pages 213–222, New York, NY, USA, 2010. ACM. URL : http://doi.acm.org/10.1145/1864349.1864367, doi:10.1145/1864349.1864367. 50, 57, 100

[TK92] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography : a factorization method. *International Journal of Computer Vision*, 9(2) :137–154, 1992. URL : http://link.springer.com/article/10.1007/BF00129684. II

[TK93] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams : a factorization method. *Proceedings of the National Academy of Sciences*, 90(21) :9795–9802, 1993. URL : http://www.pnas.org/content/90/21/9795.short. I, II, III

[TMDSA08] Federico Tombari, Stefano Mattoccia, Luigi Di Stefano, and Elisa Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. URL : http://ieeexplore.ieee.org/abstract/document/4587677/. 21

[TS94] Ping-Sing Tsai and Mubarak Shah. Shape from shading using linear approximation. *Image and Vision computing*, 12(8) :487–498, 1994. URL : http://www.sciencedirect.com/science/article/pii/0262885694900027. VI

[TS08] T. Teixeira and A. Savvides. Lightweight People Counting and Localizing for Easily Deployable Indoors WSNs. *IEEE Journal of Selected Topics in Signal Processing*, 2(4) :493–502, August 2008. URL : http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4629882, doi:10.1109/JSTSP.2008.2001426. 54, 57, 102

[TYOY99] K. Terada, D. Yoshida, Shunichiro Oe, and J. Yamaguchi. A method of counting the passing people by using the stereo images. In *Image Processing, Procs. Int.l Conf. on*, volume 2, pages 338–342. IEEE, 1999. URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=822913. 25, 76

[Ull79] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London B : Biological Sciences*, 203(1153) :405–426, 1979. URL : http://rspb.royalsocietypublishing.org/content/203/1153/405.short. I

[VJ04] Paul Viola and Michael J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2) :137–154, 2004. URL : http://link.springer.com/article/10.1023/B:VISI.0000013087.49260.fb. 58

[VJS03] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 734–741 vol.2, October 2003. doi:10.1109/ICCV.2003.1238422. 50, 52

[VKA15] A. Virgona, N. Kirchner, and A. Alempijevic. Sensing and perception technology to enable real time monitoring of passenger movement behaviours through congested rail stations. In *Australasian Transport Research Forum (ATRF), 37th, 2015, Sydney, New South Wales, Australia*, 2015. URL : https://trid.trb.org/view.aspx?id=1395117. 102, 115

[VSC+08] Senem Velipasalar, Jason Schlessman, Cheng-Yao Chen, Wayne H. Wolf, and Jaswinder P. Singh. A scalable clustered camera system for multiple object tracking. *EURASIP Journal on Image and Video Processing*, 2008 :22, 2008. URL : http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.385.115&rep=rep1&type=pdf. 75, 102, 103

[VZS13] Pablo Vera, Daniel Zenteno, and Joaquín Salas. Counting Pedestrians in Bidirectional Scenarios Using Zenithal Depth Images. In *Pattern Recognition*, pages 84–93. Springer, 2013. URL : http://link.springer.com/chapter/10.1007/978-3-642-38989-4_9. 25, 54, 57, 60, 65, 76

[WAK] Chen Wu, Hamid Aghajan, and Richard Kleihorst. Mapping vision algorithms on simd architecture smart cameras. In *Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on*, pages 27–34. IEEE. URL : http://ieeexplore.ieee.org/abstract/document/4357502/. XXIX

[War15] Jamie Ward. *The student's guide to cognitive neuroscience*. Psychology Press, 2015. XI

[web16a] web. ATR | Advanced Telecommunications Research Institute International, 2016. URL : http://www.atr.jp/index_e.html. 2, 114

[web16b] web. Machines that understand our behavior, 2016. URL : http://humansensing.cs.cmu.edu/home. 2

[web16c] web. Parrot S.L.A.M.dunk, December 2016. URL : https://www.parrot.com/us/business-solutions/parrot-slamdunk. 31

[web16d] web. Point Grey - Brickstream People Counting and Tracking Sensors, 2016. URL : http://www.brickstream.com/. 6, 16

[web16e] web. VAAM, 2016. URL : http://vaam.isasi.cnr.it/index.html. 4

[web17a] web. Bidirectional Visitor Counter | 3d People Counting | Queue Management System, 2017. URL : http://www.delopt.co.in/robovision-3d-people-counter.html. 5, 16

[web17b] web. Computer vision – Image processing and analytics | Microsoft Azure, 2017. URL : https://azure.microsoft.com/en-gb/services/cognitive-services/computer-vision/. 118, 124

[web17c] web. EEMBC - CoreMark - Processor Benchmark, 2017. URL : http://www.eembc.org/coremark/index.php. 93

[web17d] web. Etron Technology, Inc., 2017. URL : http://www.etron.com/en/products/depthmap_detial.php?Product_ID=21. 31

[web17e] web. The Future of Health Care : deep data, smart sensors, virtual patients and the Internet-of-Humans - FUTURIUM - European Commission, 2017. URL : https://ec.europa.eu/futurium/en/content/future-health-care-deep-data-smart-sensors-virtual-patients-and-internet-human 5

[web17f] web. Intel® RealSense™ Developer Kit for Tablets R200, 2017. URL : http://click.intel.com/intel-realsense-developer-kit-r200.html. 32

[web17g] web. Kinect pour Xbox One, 2017. URL : http://www.xbox.com/fr-FR/xbox-one/kinect/kinect-for-xbox-one. 33

[web17h] web. PS4™ Technical Specifications - PlayStation®4 System, 2017. URL : http://us.playstation.com/ps4/features/techspecs/index.htm. 32

[web17i] web. ZED - Depth Sensing and Camera Tracking, 2017. URL : https://www.stereolabs.com/zed/specs/. 32

[WLY15] Y. Wu, J. Lim, and M. Yang. Object Tracking Benchmark. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(9) :1834–1848, September 2015. doi:10.1109/TPAMI.2014.2388226. 6, 15, 72

[WOL] W. Wolf, B. Ozer, and T. Lv. Smart cameras as embedded systems. 35(9) :48–53. doi:10.1109/MC.2002.1033027. XXIX

[Wol14] Marilyn Wolf. Platforms and architectures for distributed smart cameras. In *Distributed Embedded Smart Cameras*, pages 3–23. Springer, 2014. 101

[Woo94] Robert J. Woodham. Gradient and curvature from the photometric-stereo method, including local confidence estimation. *JOSA A*, 11(11) :3050–3068, 1994. URL : https://www.osapublishing.org/abstract.cfm?uri=josaa-11-11-3050. V, VII, VIII

[WWS09] Christian Wojek, Stefan Walk, and Bernt Schiele. Multi-cue onboard pedestrian detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 794–801. IEEE, 2009. URL : http://ieeexplore.ieee.org/abstract/document/5206638/. 51

[YMKC08] Tarek Yahiaoui, Cyril Meurie, Louahdi Khoudour, and François Cabestaing. A People Counting System Based on Dense and Close Stereovision. In *Image and Signal Processing*, pages 59–66. Springer, Berlin, Heidelberg, July 2008. URL : https://link.springer.com/chapter/10.1007/978-3-540-69905-7_7, doi:10.1007/978-3-540-69905-7_7. 54, 57

[YVC10] Youlu Wang, Senem Velipasalar, and Mauricio Casares. Cooperative Object Tracking and Composite Event Detection With Wireless Embedded Smart Cameras. *IEEE Transactions on Image Processing*, 19(10) :2614–2633, October 2010. URL : http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5484500, doi:10.1109/TIP.2010.2052278. 54

[ZBK14] Francesco Zanlungo, Dražen Brščić, and Takayuki Kanda. Pedestrian group behaviour analysis under different density conditions. *Transportation Research Procedia*, 2 :149–158, 2014. 115

[ZC91] Qinfen Zheng and Rama Chellappa. Estimation of illuminant direction, albedo, and shape from shading. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 540–545. IEEE, 1991. URL : http://ieeexplore.ieee.org/abstract/document/139750/. V

[ZK00] C. Lawrence Zitnick and Takeo Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on pattern analysis and machine intelligence*, 22(7) :675–684, 2000. URL : http://ieeexplore.ieee.org/abstract/document/865184/. 21

[ZMM+16] Pietro Zanuttigh, Giulio Marin, Carlo Dal Mutto, Fabio Dominio, Ludovico Minto, and Guido Maria Cortelazzo. Operating Principles of Time-of-Flight Depth Cameras. In *Time-of-Flight and Structured Light Depth Cameras*, pages 81–113. Springer International Publishing, 2016. DOI : 10.1007/978-3-319-30973-6_3. URL : http://link.springer.com/chapter/10.1007/978-3-319-30973-6_3. 19

[ZTCS99] Ruo Zhang, Ping-Sing Tsai, J. E. Cryer, and M. Shah. Shape from shading : a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8) :690–706, August 1999. doi:10.1109/34.784284. V, VI

[ZYB+17] Francesco Zanlungo, Zeynep Yucel, Drazen Brscic, Takayuki Kanda, and Norihiro Hagita. Intrinsic group behaviour : dependence of pedestrian dyad dynamics on principal social and personal features. *arXiv preprint arXiv :1703.02672*, 2017. URL : https://arxiv.org/abs/1703.02672. 102, 108

Troisième partie

Annexes

Annexe A

Reconstruction 3D à partir du mouvement

La méthode de la « reconstruction 3D à partir du mouvement » utilise une séquence d'images 2D d'une seule caméra pour reconstruire le mouvement dans l'espace (odométrie visuelle) et la forme tridimensionnelle (reconstruction 3D). C'est l'un des problèmes les plus difficiles à traiter dans la vision par ordinateur tel que présenté dans [Ull79] et qui sont encore explorés scientifiquement [AEL+18]. Afin d'expliquer cette méthodologie, nous nous basons sur [TK93] en utilisant leur méthode de factorisation. Cette méthode repose sur la mise en correspondance des projections d'une ou plusieurs points caractéristiques dans les images successives. Cela suppose que les projections soient orthographiques et que la correspondance à partir du même point caractéristique dans les différentes projections (images 2D) soit connue pour chaque point dans toutes les images. Par conséquent, nous suivons les points caractéristiques P (non coplanaires) de coordonnées (u, v) pixeliques dans les images F, ($F \geqslant 3$) successives du flux vidéo, résultant d'une séquence des coordonnées dans les images :

$$\{(u_{fp}, v_{fp}) | f = 1, ..., \mathrm{F}; p = 1, ...\mathrm{P}\} \tag{A.1}$$

Ensuite, nous écrivons les coordonnées horizontales u_{fp} dans une matrice U de taille F lignes et P colonnes (F × P) et les coordonnées verticales v_{fp} dans une matrice V de taille F × P. Chaque ligne représente la même image et chaque colonne correspond aux coordonnées du même point dans les différentes images F. En utilisant la matrice U et V, nous construisons la matrice de mesures W de taille 2F × P, en plaçant les coordonnées X dans le couple supérieur de la matrice et les valeurs Y dans la partie inférieure.

$$\mathrm{W} = \begin{bmatrix} u_{11} ... u_{1p} \\ \vdots \\ u_{\mathrm{F}1} ... u_{\mathrm{FP}} \\ v_{11} ... u_{1\mathrm{P}} \\ \vdots \\ v_{\mathrm{F}1} ... u_{\mathrm{FP}} \end{bmatrix} \qquad \mathrm{W} = \begin{bmatrix} \mathrm{U} \\ \mathrm{V} \end{bmatrix} \tag{A.2}$$

Ensuite nous normalisons les valeurs en soustrayant le barycentre des points de l'image pour chaque élément :

$$\begin{aligned} \tilde{u}_{fp} &= u_{fp} - a_{fp} \\ \tilde{v}_{fp} &= v_{fp} - b_{fp} \end{aligned} \tag{A.3}$$

oú :

$$a_f = \frac{1}{\mathrm{P}} \sum_{p=1}^{\mathrm{P}} u_p \qquad b_f = \frac{1}{\mathrm{P}} \sum_{p=1}^{\mathrm{P}} v_p \tag{A.4}$$

Puis nous obtenons la matrice des mesures centrées au référentiel orthogonal $\tilde{\mathrm{W}}$

$$\tilde{\mathrm{W}} = \begin{bmatrix} \tilde{\mathrm{U}} \\ \tilde{\mathrm{V}} \end{bmatrix} \tag{A.5}$$

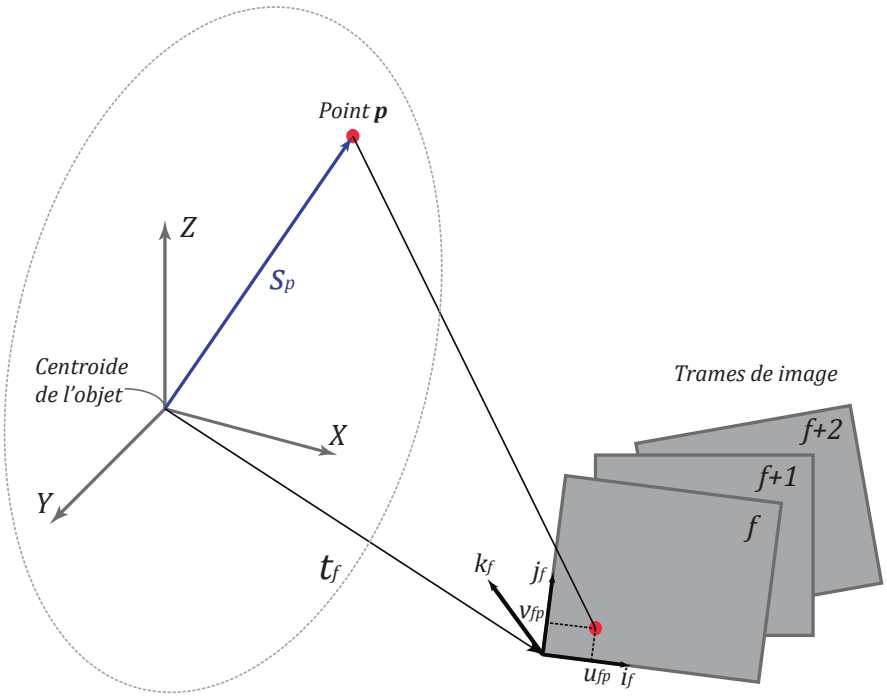


FIGURE A.1 – Diagramme de la relation entre le référentiel orthonormal du monde réel et le référentiel de la caméra pour chaque image [TK93]

L'orientation du référentiel de la caméra pour chaque image f est déterminée par les vecteurs unitaires i_f et j_f (voir A.1) et définie par rapport au référentiel avec coordonnées x, y et z au barycentre des P points. Ensuite, sous la projection orthogonale, on obtient le vecteur unitaire k_f par la propriété :

$$k_f = i_f \times j_f \tag{A.6}$$

Soit $S_p = (X_p, Y_p, Z_p)$ les coordonnées du point réelles dans l'espace qui correspondent aux points caractéristiques des objets suivis $S_p = (X_p, Y_p, Z_p)^T, p = 1, \ldots, P$. La projection orthogonale (u_{fp}, v_{fp}) de S sur l'image f est donnée par les équations [TK92] :

$$\begin{aligned} u_{fp} &= i_f^T (S_P - t_f) \\ v_{fp} &= j_f^T (S_P - t_f) \end{aligned} \tag{A.7}$$

Où t_f est le vecteur qui part de l'origine du référentiel à l'origine d'image (i_f, j_f) de la trame f (Fig. A.1). En utilisant l'équation A.4 (points moyens normalisés) dans la relation entre le référentiel et les images (Eq. A.7), nous obtenons les p points en termes de référentiel.

$$\tilde{u}_{fp} = i_f^T \begin{bmatrix} S_p & -\frac{1}{P} \sum_{q=1}^{P} S_q \end{bmatrix} \tag{A.8}$$

Si nous supposons que l'origine du référentiel est dans le barycentre des points 3D, nous pouvons éliminer le terme du barycentre et nous obtenons :

$$\begin{aligned} \tilde{u}_{fp} &= i_f^T - s_p \\ \tilde{v}_{fp} &= j_f^T - s_p \end{aligned} \tag{A.9}$$

alors, on réécrit la matrice moyenne normalisée $\tilde{\mathrm{W}}$ sous forme de matrice :

$$\tilde{\mathrm{W}} = \begin{bmatrix} i_1^{\mathrm{T}} \\ \vdots \\ i_{\mathrm{F}}^{\mathrm{T}} \\ j_1^{\mathrm{T}} \\ \vdots \\ j_{\mathrm{F}}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} s_1 \dots s_p \end{bmatrix} = \mathrm{RS} \tag{A.10}$$

où R est la matrice de rotation (mouvement) formée par le vecteur unitaire i_f et j_f pour chaque trame par rapport au référentiel, et S est la matrice des positions de points S_p (forme) dans le système référentiel. Comme R est 2F × 3 et S est 3 × P, l'équation A.10 implique que, sans bruit, la matrice de mesures enregistrées est au plus de rank 3 (Théorème de « Rank »). Ayant ceci à l'esprit, exprimons l'équation A.3 dans les termes de U_{fp}, puis en remplaçant $\tilde{u}$ par la partie droite de l'équation A.9, on compare le résultat avec l'équation A.7 et on obtient :

$$a_f = -t_f i_f^{\mathrm{T}} \tag{A.11}$$

nous pouvons écrire la matrice de mesures originales W comme :

$$\mathrm{W} = \mathrm{RS} + t e_p^{\mathrm{T}} \tag{A.12}$$

où t et e_p^t sont :

$$\begin{aligned} t &= (a_1, \dots, a_f, b_1, \dots, d_f)^{\mathrm{T}} \\ e_p^{\mathrm{T}} &= (1, \dots 1) \end{aligned} \tag{A.13}$$

t recueille les projections de la translation de la caméra le long du plan de l'image, et e est un vecteur de taille P, afin de donner cohérence à l'équation, sous forme scalaire

$$u_{fp} = i_f^{\mathrm{T}} s_p + a_f \qquad v_{fp} = j_f^{\mathrm{T}} s_p + b_f \tag{A.14}$$

Cependant, lors de mesures avec du bruit, le rank de $\tilde{\mathrm{W}}$ ne sera pas exactement 3. Par conséquent, on met en application le théorème de l'approximation du *Rank* qu'utilise le concept de décomposition de valeur singulière ***S**ingular **v**alue **d**ecomposition* (SVD) pour extraire des matrices du rank 3 de la manière suivante :

On suppose donc 2F ⩾ P et décomposons la matrice $\tilde{\mathrm{W}}$ en une matrice O_1 de taille 2F × P, une matrice diagonale Σ de taille P × P et une matrice O_2 de taille P × P.

$$\tilde{\mathrm{W}} = \mathrm{O}_1 \Sigma \mathrm{O}_2 \tag{A.15}$$

Ensuite, en imposant les matrices résultantes pour être de rank 3, nous réécrivons la décomposition comme :

$$\mathrm{O}_1 = [\underbrace{\mathrm{O}_1^{'}}_{3} \underbrace{\mathrm{O}_1^{''}}_{\mathrm{P}-3}] \; \}2\mathrm{F} \qquad \Sigma = \underbrace{\begin{bmatrix} \Sigma^{'} \\ \mathrm{O} \end{bmatrix}}_{3} \underbrace{\begin{bmatrix} \mathrm{O} \\ \Sigma^{''} \end{bmatrix}}_{\mathrm{P}-3} \begin{matrix} \} 3 \\ \} \mathrm{P}\text{-}3 \end{matrix} \qquad \mathrm{O}_2 = \underbrace{\begin{bmatrix} \mathrm{O}_2^{'} \\ \mathrm{O}_2^{''} \end{bmatrix}}_{\mathrm{P}} \begin{matrix} \} 3 \\ \} \mathrm{P}\text{-}3 \end{matrix}$$
$$\tilde{\mathrm{W}} = \mathrm{O}_1 \Sigma \mathrm{O}_2 = \mathrm{O}_1^{'} \Sigma^{'} \mathrm{O}_2^{'} + \mathrm{O}_1^{''} \Sigma^{''} \mathrm{O}_2^{''}$$
$$\hat{\mathrm{W}} = \mathrm{O}_1^{'} \Sigma^{'} \mathrm{O}_2^{''} \tag{A.16}$$

où les informations utiles sont dans la première partie de l'équation et les mesures de bruit sont rejetées dans la seconde partie (s'il n'y avait pas de bruit dans les mesures, les valeurs de $\Sigma^{''}$ doivent être 0) (mesures de bruit de théorème) [TK93]. Par conséquent, nous pouvons obtenir :

$$\begin{aligned} \hat{\mathrm{R}} &= \mathrm{O}_1^{'} [\Sigma^{'}]^{1/2} \\ \hat{\mathrm{S}} &= [\Sigma^{'}]^{1/2} \mathrm{O}_2^{'} \\ \hat{\mathrm{W}} &= \hat{\mathrm{R}} \hat{\mathrm{S}} \end{aligned} \tag{A.17}$$

Les $\hat{R}$ et $\hat{S}$ ont la même taille que les matrices R et S, mais la décomposition Eq.A.16 n'est pas unique. Pour obtenir une décomposition unique, nous devons calculer une matrice inversible Q de taille 3×3. Comme nous savons que la matrice de rotation est orthogonale, on peut ajouter la contrainte :

$$\hat{i}_f^T Q Q^T \hat{i}_f = 1 \qquad \hat{j}_f^T Q Q^T \hat{j}_f = 1 \qquad \hat{i}_f^T Q Q^T \hat{j}_f = 0 \tag{A.18}$$

Finalement, nous obtenons les matrices R et S comme suit :

$$\begin{aligned} R &= \hat{R} Q \\ S &= Q^{-1} \hat{S} \end{aligned} \tag{A.19}$$

En résumé, la méthode de la forme à partir du mouvement utilise une séquence d'images de la même caméra qui se déplace. Dans chaque image, nous estimons ensuite le déplacement de la caméra (mouvement) puis les informations de profondeur (forme). Cette méthode est souvent utilisée dans la localisation simultanée et la cartographie ***S**imultaneous **L**ocalization **A**nd **M**apping* (SLAM) pour les applications de véhicules autonomes comme l'automobile et les drones.

Annexe B

Forme à partir de l'ombrage

Dans la méthodologie de formes à partir de l'ombrage, nous utilisons les changements d'éclairage et d'ombrage pour récupérer les informations 3D de la surface. On définit explicitement la surface comme $z = f(x, y)$ dans un système de coordonnées cartésiennes de main gauche, où l'observateur regarde à partir de la direction Z positive, la projection d'image est orthogonale et les axes XY de l'image coïncident avec les axes XY de l'objet [Woo94]. Nous définissons le gradient de surface (p, q) comme le taux de variation de la profondeur dans les valeurs x et y dans l'image par :

$$(p, q) = \left(\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y}\right) \tag{B.1}$$

Par conséquent, on peut définir le vecteur normal de la surface comme $n = [p, q, -1]$. Dans le modèle de Lambert, où la direction de la source de lumière est connue (ou qu'elle peut être calibrée avec un objet réfléchissant), la surface observée a un albédo uniforme (facteur de réflectance de surface) A et une réflectance diffuse (réfléchit la lumière dans toutes les directions) [Sze10]. Nous pouvons modéliser l'équation d'irradiance d'une surface lambertienne [ZTCS99, ZC91] comme suit :

$$\mathrm{E}(x, y) = \mathrm{R}(p, q) = \mathrm{A}\cos\Phi_i \tag{B.2}$$

Où E est l'irradiance d'image, R est la carte de réflectance et dépend de l'angle Φ_i entre la direction du rayon émis d'une seule source de lumière S et de surface N normale comme le montre la figure B.1 [ZTCS99].

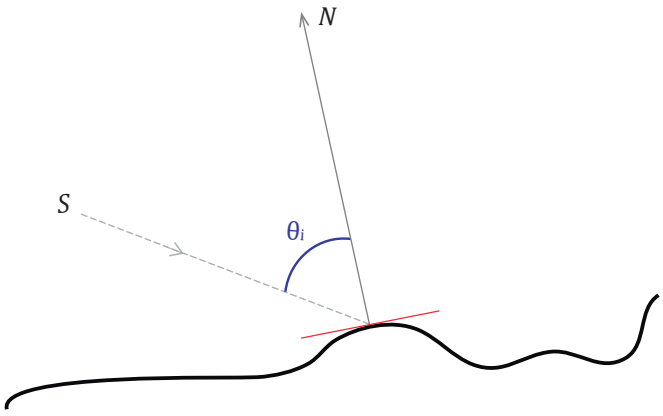


FIGURE B.1 – Géométrie Lambertienne de réflexion

On peut récupérer la forme de la surface exprimée en termes de gradient de surface (p, q)

[Sze10], en utilisant l'équation B.2 comme :

$$E(x,y) = A\overrightarrow{n} \cdot \overrightarrow{s} = \left(\frac{1}{\sqrt{p^2+q^2+1}} (-p,-q,1) \right) \cdot (s_x, s_y, s_z); \tag{B.3}$$

Une autre façon peut être d'exprimer la forme par la normale à la surface. Nous définissons la normale à la surface comme le vecteur perpendiculaire au plan tangent de la surface de l'objet. On considère la surface normale et la source lumineuse comme des vecteurs unitaires. Ainsi, la forme de surface à partir de la variation d'ombrage [ZTCS99], en utilisant l'équation B.2 peut être écrite comme suit :

$$E(x,y) = A\overrightarrow{n} \cdot \overrightarrow{s} = (n_x, n_y, n_z) \cdot (s_x, s_y, s_z) \tag{B.4}$$

Où, E est égal au produit scalaire de la normale de la surface $\vec{n}$ et de la source lumineuse $\vec{s}$.

Dans les figures B.2, les images (B.2a - B.2d) sont les images synthétiques générées avec deux sources de lumière différentes. Les images (B.2e-B.2h) représentent la profondeur résultant de chaque image synthétique en utilisant la forme à partir de l'ombrage de [TS94].

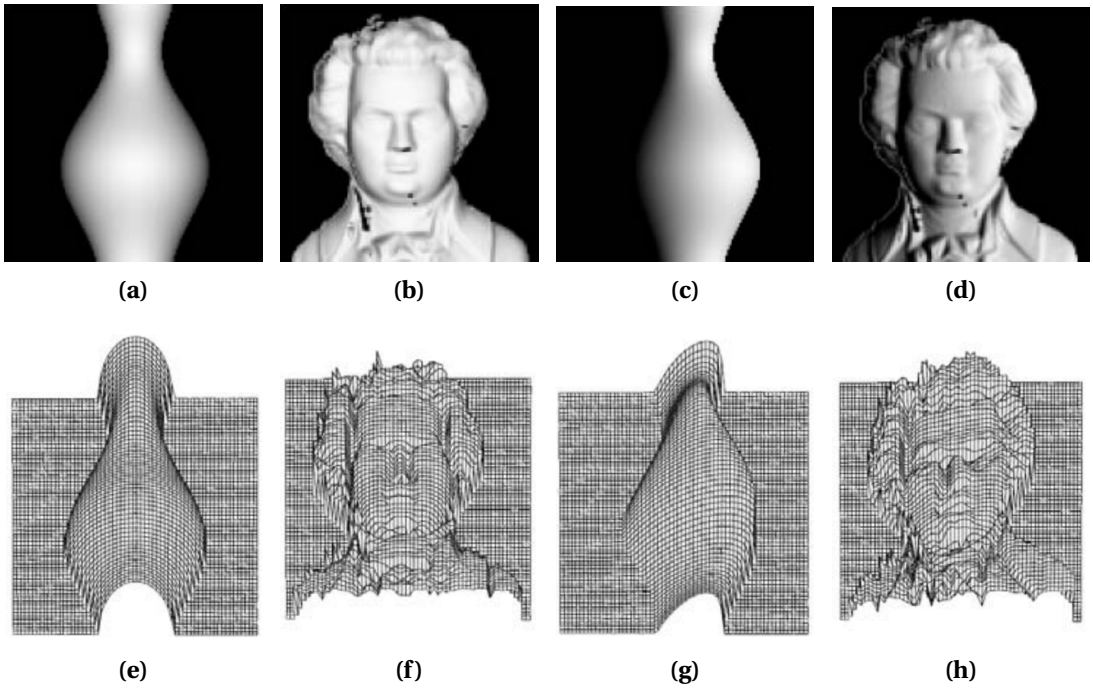


FIGURE B.2 – Formes synthétiques à partir de l'ombrage extrait de [ZTCS99] ; images ombragées, $(a-b)$ avec lumière de face $(0,0,1)$ et $(c-d)$ avec lumière venant de face et la droite $(1,0,1)$; $(E-f)$ forme correspondante des reconstructions à partir d'ombrages en utilisant la technique de [TS94].

L'utilisation de cette méthode dans la vie réelle est devenue difficile car les surfaces n'ont pas de taux de réflectance uniformes (valeurs d'albédo) [Sze10] ou bien nous avons plusieurs sources de lumière qui se déplacent constamment. Par conséquent, on doit la combiner avec une autre technique comme la correspondance stéréo.

Annexe C

Forme à partir de photométrie stéréo

Cette méthode est basée sur le même modèle Lambertien (Fig. A.1) que la forme à partir de l'ombrage mais nous prenons plusieurs images avec différentes sources de lumière pour récupérer la forme 3D. De plus, la valeur de l'albédo est inconnue. Cependant, nous connaissons la direction des sources de lumière et celle-ci est la même pour l'ensemble de l'image ; en plus, les positions des sources de lumière ne sont pas coplanaires. Par conséquent, nous prenons trois images différentes en utilisant l'équation A.8, on obtient :

$$\begin{aligned} \mathrm{I}_1(x,y) &= \mathrm{A}\overrightarrow{\mathrm{S}}^1 \cdot \vec{n}; \\ \mathrm{I}_2(x,y) &= \mathrm{A}\overrightarrow{\mathrm{S}}^2 \cdot \vec{n}; \\ \mathrm{I}_3(x,y) &= \mathrm{A}\overrightarrow{\mathrm{S}}^3 \cdot \vec{n}; \end{aligned} \tag{C.1}$$

On réécris alors l'équation sous la forme matricielle :

$$\mathrm{A}\begin{bmatrix} \mathrm{S}_x^1 & \mathrm{S}_y^1 & \mathrm{S}_z^1 \\ \mathrm{S}_x^2 & \mathrm{S}_y^2 & \mathrm{S}_z^2 \\ \mathrm{S}_x^3 & \mathrm{S}_y^3 & \mathrm{S}_z^3 \end{bmatrix}\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \mathrm{I}_1 \\ \mathrm{I}_2 \\ \mathrm{I}_3 \end{bmatrix} \tag{C.2}$$

Pour récupérer la valeur A, nous supposons que la normale est un vecteur unitaire $||n|| = 1$, alors nous réécrivons la matrice des sources lumineuses S et l'inversons (nous supposons que les sources lumineuses ne sont pas coplanaires), on obtient :

$$\mathrm{A} = ||\mathrm{S}^{-1}\mathrm{I}|| \tag{C.3}$$

Enfin avec la valeur A, on peut récupérer la normale de la surface par :

$$\overrightarrow{\mathrm{N}} = \frac{\mathrm{S}^{-1}\mathrm{I}}{\mathrm{S}} \tag{C.4}$$

On peut observer dans la figure C.1 les trois images d'entrée et la reconstruction 3D obtenue présentée dans [Woo94].

Dans la figure C.1, l'encart (en bas à droite) montre la rosette de couleur utilisée pour coder le dégradé.

(a) **(b)** **(c)** **(d)**

FIGURE C.1 – Exemple de gradient encodé en couleur, $(p;q)$, tel qu'il est produit dans la mise en œuvre quasi temps réel (15Hz) à partir de photométrie stéréo - Images prises de [Woo94].

Annexe D

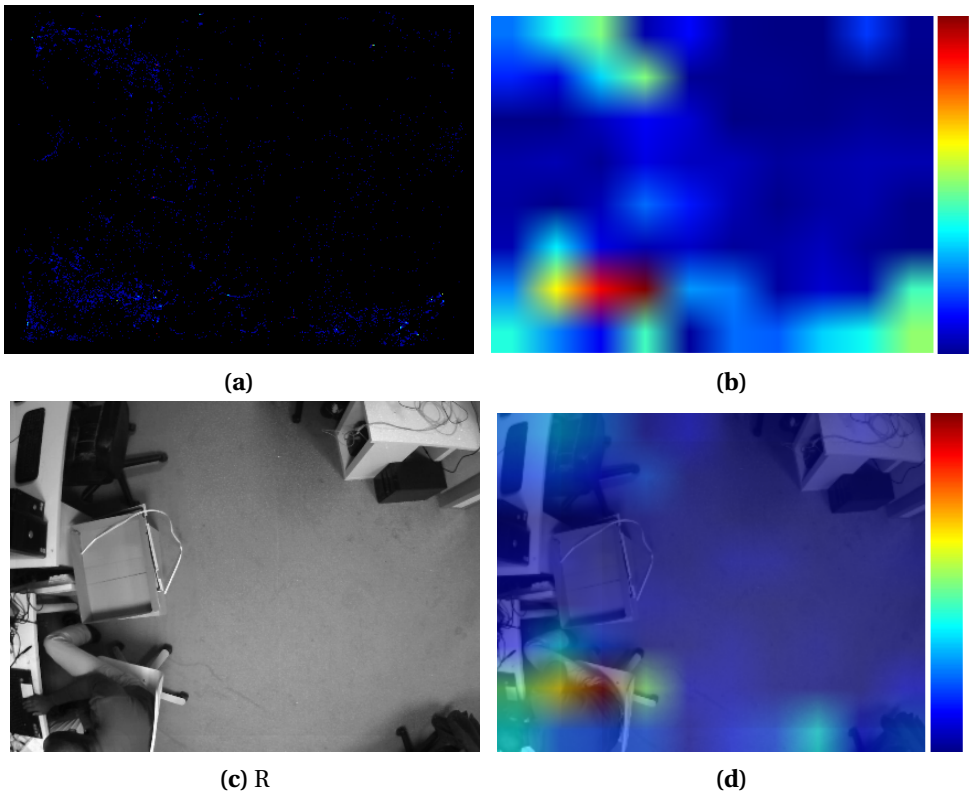
Génération de la carte de chaleur

La première approche utilise comme données d'entrée les régions d'intérêt Θ_n résultant de la «segmentation légère» de la chaîne de traitement en ligne (voir section 3.2.1). Ensuite, dans la carte de chaleur, qui a la même taille que l'image acquise, on ajoute chaque Θ_n sans faire aucun autre traitement. Cette méthode permet d'avoir un haut niveau (par pixel) de détail par rapport à la forme réelle de la personne et les espaces précis traversés par la personne (Fig. D.1a). Par contre, cette approche présente des inconvénients : trajectoires incomplètes et génération de bruit par objets dynamiques dans la scène. Le cas de trajectoires incomplètes (Fig. D.1b) se présente quand la vitesse de traitement des images est très lente ou quand les personnes traversent très rapidement le champ de vision de la caméra. Le cas du bruit par des objets dynamiques se présente quand il y a des objets qui n'appartiennent pas initialement à l'arrière plan et qui sont introduits pendant le temps du suivi. Ce dernier cas se présente du fait que la «segmentation légère» produit des informations brutes. Pour remédier à ce problème en partant de la forme de la personne, on peut remplir les parties de la trajectoire manquante en dupliquant la forme de Θ_n (tous ses pixels) le long du segment qui connecte le dernier point connu au point que l'on vient de détecter. Ce calcul a des chances d'être très lourd à cause du nombre de pixels qui composent la tâche. En effet, pour résoudre cette problématique on a besoin d'utiliser des informations plus riches dans la chaîne de traitement pour reconstruire les parties incomplètes des trajectoires et filtrer des objets inattendus dans la scène. Ceci nous amène à la deuxième approche.

Dans la Figure D.1,on observe des cartes de chaleur fusionnées avec l'image couleur d'une caméra située à l'entrée d'un bureau ayant une résolution de 160 × 120 pixels. Dans les images, le rouge représente les endroits les plus utilisés et bleu foncé les moins utilisés. De plus, les parties de l'image sans la couche de chaleur représentent les endroits où les personnes sont passées. Dans la première image D.1a, on observe l'utilisation de l'espace où l'entrée (coin en bas à droite) est une zone rouge, jusqu'à la bifurcation des trajectoires (les autres régions en jaune et bleu clair). Dans la deuxième image D.1b, on observe différentes couleurs segmentées en petites régions irrégulières à la différence de la première image. Cet effet est produit par un taux bas d'échantillonnage par rapport à la vitesse de la personne observée. Dans la troisième image D.1c, on observe que la porte a généré un bruit intense (points rouges autour de la porte) qui est typique des lieux très fréquentés, en perdant les informations sur les autres trajectoires. Cette perte est dû au fait que le bruit augmente tellement le comptage de passages dans ces lieux que les trajectoires réelles ne sont plus identifiables sur l'échelle de coloris.

Résultats de génération la charte de chaleur

On a utilisé cette méthode pour l'appliquer sur des caméras intelligentes 2D et 3D. En plus des problèmes décrits précédemment, on ajoute les problèmes de détection de mouvements, introduits par les changements de lumière et des ombres dans les caméras 2D, ce qui génère des fausses cartes de chaleur. D'un côté, les ombres génèrent des «faux» mouvements de personnes dans des endroits inaccessibles. D'un autre côté, les forts changements de lumière ajoutent du mouvement dans toute la scène. Ces effets combinés donnent des cartes qui ne sont pas



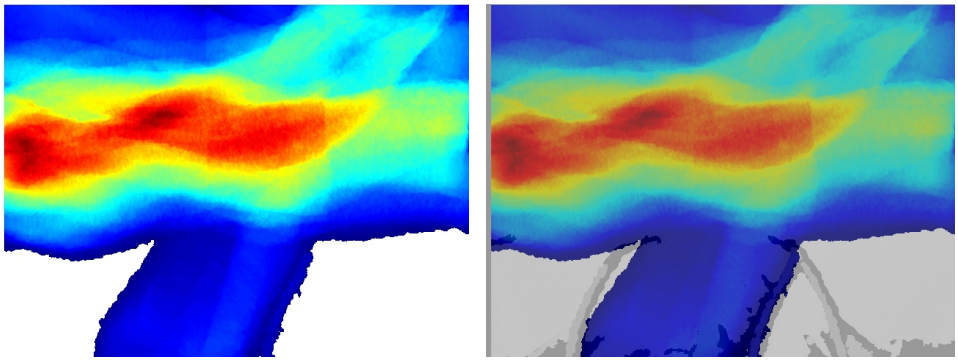
(a) **(b)**

(c) R **(d)**

FIGURE D.1 – Images de la carte de chaleur d'un lieu de passage par une porte située en bas à droite dans différentes conditions. a) Cas d'un flux de personnes marchant à vitesse moyenne. b) Trace d'une personne marchant rapidement. c) Perte d'information des trajectoires du fait que la porte (en bas à droite) soit est un objet dynamique non identifié de la chaîne de traitement hors ligne qui génère un bruit (rouge), en annulant la visualisation du reste des trajectoires. Ces images sont acquises par une caméra intelligente qui travaille à faible résolution.

représentatives de l'utilisation de l'espace ou qui sont bruitées. Par conséquence, cette approche d'estimation de la carte de chaleur ne fonctionne qu'avec une caméra 3D qui fournit la précision de détection de mouvement requis sans interférence avec des perturbations extérieures.

On a testé cette méthode sur les jeux de données décrits sur dans la section 5.3.2. En appliquant cette méthode dans **JDT**, on obtient la carte présentée dans la figure D.2b. De plus, dans la figure D.2b on présente l'image de profondeur d'entrée superposée avec la carte de chaleur. On observe dans le centre de l'image où les trajectoires sont les plus proches et se chevauchent, la formation d'une région rouge. Grâce à l'échelle de coloris on est capable de déterminer rapidement où sont les endroits les plus utilisés. Aussi, on est capable de distinguer certains cas particuliers comme le mouvement d'une personne, limité par l'entrée d'un tourniquet (voir la partie basse de l'image).

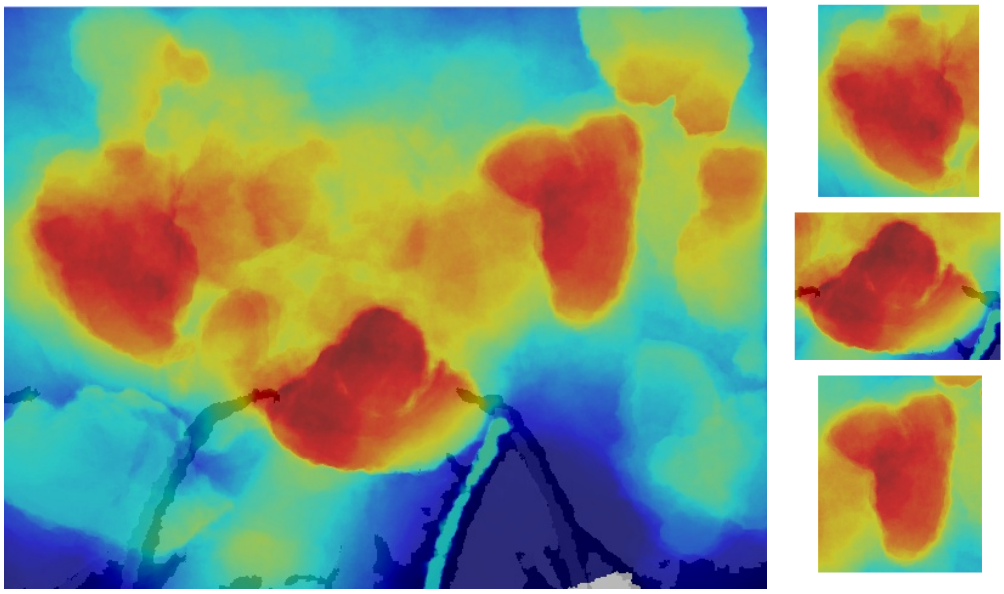


(a) **(b)**

FIGURE D.2 – Carte de chaleur des sous-ensembles des trajectoires **JDT**. a) Carte de Chaleur résultante. b) Chevauchement du fond avec la carte de chaleur.

Le sous-ensemble suivant évalué est JD01. On observe que l'accumulation des mouvements sans traitement génère des artefacts dans des endroits où les personnes sont restées statiques sur une période significative de temps. On peut voir aussi (voir figure D.3b)la direction de leurs regards. Cette accumulation des mouvements introduit des artefacts de forme humaine qui perturbent la compréhension de la scène, en générant une perte de la compréhension de l'utilisation de l'espace.

Si on observe à première vue la figure D.3, on n'atteint pas l'objectif de refléter l'utilisation de l'espace d'une manière assez simple et représentative. En effet, l'image résultante attire l'attention sur les 3 artefacts avec des formes de personnes plutôt qu'un mouvement global de la scène. En fait l'explication est donnée par le modèle de reconnaissance des objets des neurosciences cognitives proposé par [War15]. En regardant une image, d'abord on identifie des composantes basiques mémorisées (comme les artefacts en forme de personnes Fig. D.3b), et ensuite, on passe à la compréhension du contexte pour trouver la signification de l'image.



(a) **(b)**

FIGURE D.3 – Carte de chaleur du jeu de données JD01 mélangé avec le fond d'une entrée avec un tourniquet.

En comparant les deux cartes de chaleur des jeux de données JDT (Fig. D.2) et JD01 (Fig. D.3), on observe que les résultats obtenus dans une période de temps courte sont plus satisfaisants par rapport à ceux obtenus sur des durées longues. En effet, ceci évite de produire des artefacts issus de séjours statiques des personnes dans la scène (Fig. D.1).

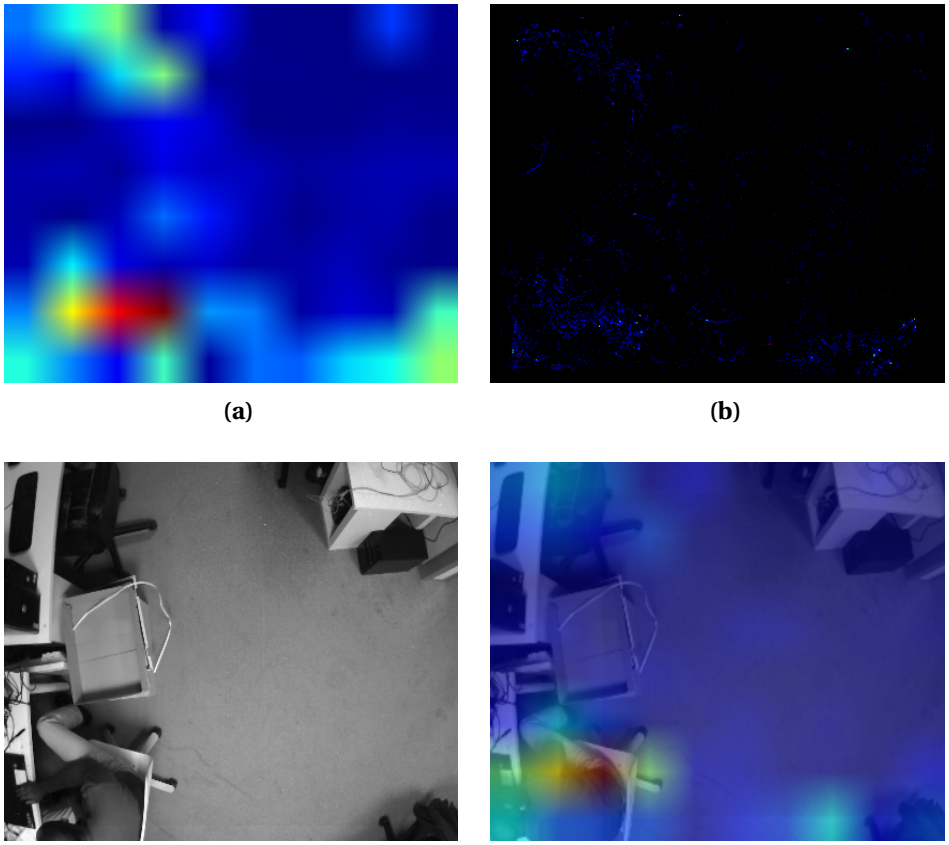
En conclusion, cette approche répond aux besoins de capturer le mouvement de manière précise sur de courtes périodes de temps [MSD17]. Dans ce travail les auteurs utilisent une caméra de profondeur pour calculer la position correcte de la personne au moment de prendre une radiographie. De plus, ils utilisent la détection de mouvement des parties du corps dans la scène pour anticiper la qualité de l'image, notamment en appréciant les flous résultant du mouvement. Cependant, dans le cadre du comptage des personnes les accumulations des positions statiques posent des problèmes pour avoir une vision globale du comportement (voir Fig. D.2) et reconnaître la dynamique de la scène.

Annexe E

Application de la génération de la carte d'occupation - Caméra 2D

Un de nos objectifs industriels est d'appliquer les méthodes d'étude du comportement sur différent types de caméras intelligentes développées par l'entreprise Shopline. On a introduit la méthode pour estimer la carte d'occupation sur une caméra monoscopique infrarouge. On a placé cette caméra à l'intérieur d'un bureau (Fig. E.1d).

Dans la figure E.1, dans la première image, on place les points des trajectoires T sur l'espace *I*. La deuxième représente le résultat de l'estimation de la carte d'occupation O_c à partir de T. De plus, O_c est transformé à l'échelle de I. La troisième est l'image de la scène observée en échelle de gris. La dernière image est la combinaison de la deuxième et troisième image.



(a) **(b)**

(c) **(d)**

FIGURE E.1

Dans la figure E.1, la visualisation des étapes pour créer la carte d'occupation à partir des trajectoires *T*. a) Image brute des points d'entrée des trajectoires T. b) Carte d'occupation O_c agrandie à l'échelle de l'image d'entrée. c) Image d'entrée en échelle de gris de la région observée. d) Mélange de l'image d'entrée avec la carte d'occupation. Ces images sont extraites directement d'une caméra intelligente monoscopique qui travaille à faible résolution.

Annexe F

Formalisation de la notation du VFKM et AFKM

Dans cette annexe contient la formalisation des algorithmes VFKM et AFKM. Ces méthodes sont une reconstruction des trajectoires discrètes à des trajectoires continues sur l'espace 2D. La première méthode utilise des trajectoires T composées de points spatio-temporels, qui sont composés par la position $\rho(x,y)$ sur l'axe X et Y plus le horodatage t. La deuxième méthode utilise des trajectoires ou chaque point est composé par les mêmes valeurs et un nombre variable des attributs auxiliaires associés à chaque point de la trajectoire. Ces attributs ajoutent de la complexité à l'assignation mais aussi de la pertinence. On formalise la notation du VFKM comme :

- Une trajectoire α_i est définie comme une fonction $\alpha_i : \left[t_i^1, t_i^{\mathrm{pf}}\right] \rightarrow \mathbb{R}^2$ où $t_i{}^1$ est le temps initial et $t_i{}^{pf}$ est le temps final de la trajectoire i. C'est-à-dire, $\alpha_i(t)$ représente le point de la trajectoire point α_i au moment t qui appartient à l'intervalle $[t_i{}^1, t_i{}^{pf}]$.
  - Soit I_i la durée d'une trajectoire dans l'intervalle $[t_i{}^1, t_i{}^{pf}]$ où $|I_i|=t_i{}^{pf}-t_i{}^1$.
- Soit $\mathrm{T}=\{\alpha_1, \alpha_2, ..., \alpha_n\}$ l'ensemble de trajectoires à segmenter.
- Soit $t_{\mathrm{T}} = \sum_{\alpha_i \in \mathrm{T}} \left(\mathrm{I}^j\right)$ la durée de temps total de l'ensemble de trajectoires.
- Soit $\alpha'_i(t)$ le vecteur de vitesse associés au point $\alpha_i(t)$.
- Soit X un champ vectoriel considéré comme une fonction sur le domaine $\mathrm{D}_v \subseteq \mathbb{R}^2$ avec de valeurs $\mathbb{R}^2$. On représente D_v comme une grille régulière de triangles [FKSS13] de résolution R (R^2 sommets). C'est-à-dire,$\mathrm{X}(\alpha_i(t))$ représente un vecteur de direction du champ vectoriel X dans le point $\alpha_i(t)$.
- Une trajectoire α_i est une *ligne de courant* de X si $X(\alpha_i\ (t))=\alpha'_j(t)$ au moment t. C'est-à-dire, que la trajectoire α_i suit de manière identique au champ vectoriel (voir Fig. E.1 de cette annexe).

De cette manière on a T trajectoires sur une grille R^2 que l'on a besoin de segmenter sur K clusters représenté par un champ vectoriel $\mathrm{X_j}$. Cet algorithme consiste à capturer les patrons de mouvement en définissant un champ vectoriel où les trajectoires, qui lui appartiennent, sont quasi *lignes de courant*. C'est-à-dire, on regroupe les clusters des trajectoires selon le meilleur champ vectoriel qui les rapproche. On assume que pour un ensemble de trajectoires T, il existe un *champ vectoriel lisse* (qui sa courbure n'est pas trop élevée) $\mathrm{X}_j \in \mathbf{F}$ où $\boldsymbol{F}$ est l'ensemble des champs vectoriels lisses qui représentent le mieux T et $|\boldsymbol{F}|$=k. Le champ vectoriel lisse X_j représente la majorité des mouvements des trajectoires qui lui appartiennent. La définition de ce problème est donnée par la minimisation de l'énergie de la formule suivante :

$$E(X_1,...,X_k,\Phi) = \sum_{j=1}^{k} \lambda_L ||\Delta X_j||^2 + \sum_{\alpha_i \in \Phi^{-1}(j)} \frac{(1-\lambda_L)}{t_T} \int_{t_i^1}^{t_i^{p_f}} ||X_j(\alpha_i(t)) - \alpha_i'(t)||^2 dt \quad (F.1)$$

où Δ est Laplacien du champ vectoriel (courbure), λ_L est un facteur de pesage et Φ :T→ {1,...,k} est la fonction d'assignation de trajectoires pour chaque cluster j=1,...,k (e.g.$\Phi_j(\alpha_i)$=k). La première partie de l'équation vise à lisser les champs vectoriels et la deuxième partie vise à ce que le champ vectoriel représente toutes les trajectoires. Donc, l'algorithme de VFKM est défini comme :

Algorithm 1 Esquise de l'algorithme Vector Field K-Means

Require: k : # de clusters, $T = \{\alpha_1,...,\alpha_n\}$: ensemble de courbes
Ensure: $v := \{X_1,...,X_k\}, \Phi : T \rightarrow \{1,...,k\}$
  $\Phi \leftarrow Initialize(T,k)$
  repeat
    for i=1 to K **do**
      $X_i \leftarrow fitVectorField(\Phi^{-1}(i))$
    end for
    for i=1 to n **do**
      $j_0 \leftarrow \underset{j\in\{1,2,...,k\}}{\text{argmin}}\ E'(X_j,\alpha_i)$
      $\Phi(\alpha_i) \leftarrow j_0$
    end for
  until converge

On observe que les parties plus importantes de cet algorithme sont l'ajustement du champ vectoriel et l'assignation de trajectoires. A partir de l'équation 1, on définit l'ajustement d'un champ vectoriel comme suit :

$$E'(X,T') = \lambda_L ||\Delta X||^2 + \sum_{\alpha_i \in T'} \frac{(1-\lambda_L)}{t_T} \int_{t_i^1}^{t_i^{p_f}} ||X(\alpha_i(t)) - \alpha_i'(t)||^2 dt$$

où *T'* est un sous-ensemble de trajectoires qui partagent des comportements similaires et sont la base pour estimer le champ vectoriel X. Ensuite, l'assignation de trajectoires est donnée par la minimisation de *E"* sur tous les clusters K où :

$$E''(X_j,\alpha_i) = \int_{t_i^1}^{t_i^{p_f}} ||X_j(\alpha_i(t)) - \alpha_i'(t)||^2 dt$$

Pour avoir plus de détails sur la méthode VFKM consulter [FKSS13].

Dans le cas de la méthode AFKM [Fer15], on définit une trajectoire avec M-attributs, à M-trajectoire, comme une fonction de la forme $\overline{\alpha} : [t^1, t^{pf}] \rightarrow \mathbb{R}^2 X \mathbb{R}^M$ où t^1 est le temps initial et t^{pf} est le temps final de la M-trajectoire, tel que $\overline{\alpha} = (\alpha_S(t), \alpha_A(t))$ où $\alpha_S(t) \in \mathbb{R}^2$ est la composante spatiale et $\alpha_A(t) \in \mathbb{R}^M$ sont les attributs de la trajectoire. Ensuite, on définit un champ de M-attributs, à M-champ, comme une fonction $f : \mathbb{R}^2 \rightarrow \mathbb{R}^M$. Soit $T = \{\overline{\alpha_1}, \overline{\alpha_2}, ..., \overline{\alpha_n}\}$ un ensemble de M-trajectoires de taille n. Étant donné un vecteur $x = (x_1, x_2, ..., x_M)$, on dénote le i^{eme} élément comme $x^{(i)}$. Ainsi, l'i^{eme} attribut de $\overline{\alpha}$ est dénoté comme $\overline{\alpha}^{(i)}$ et l'i^{eme} attribut d'un M-champs f est dénoté comme $f^{(i)}$.

De plus, on définit la *concordance* de $\overline{\alpha}$ avec un M-champ f de manière similaire à la notion de *lignes de courant* de la méthode VFKM. Ainsi, une M-trajectoire$\overline{\alpha}$est *concordante* avec un M-champs *f*, si $f(\alpha_S(t)) = \alpha_A(t)$ par tous les temps $\alpha_A(t) \in \mathbb{R}^M$. Dû à la complexité de cette relation, on définit la concordance d'une trajectoire $T = \{\overline{\alpha_1}, \overline{\alpha_2}, ..., \overline{\alpha_n}\}$ avec un champ d'attributs f comme :

$$Cons_w(f,\mathrm{T}) = \sum_{i=1}^{n}\sum_{d=1}^{\mathrm{M}} w_d \int_{t_i^1}^{t_i^{p_f}} ||f^{(d)}(\alpha_{\mathrm{S}_i}(t)) - \alpha_{\mathrm{A}_i}^{(d)}(t)||^2 dt$$

où W=(w_1, w_2 ,..., w_M) est le vecteur de poids pour chaque attribut pour donner plus d'importance aux attributs désirés. Cette notion est représentée dans la figure F.1 dans le cas où M = 1.

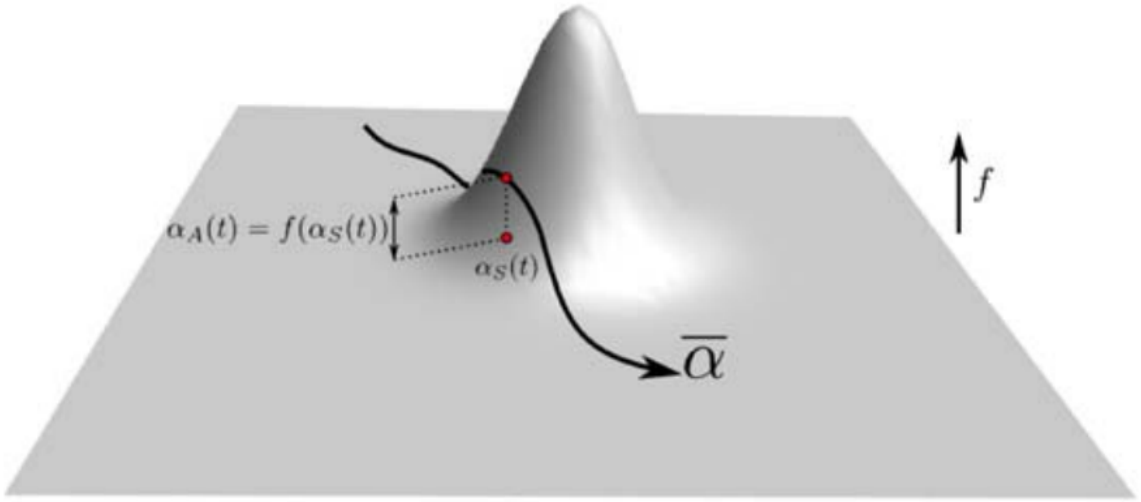


FIGURE F.1 – Représentation des lignes de courant pour un champ de 1-attribut.

La figure F.1 illustre la représentation du concept de ligne de courant pour un champ de 1-attribut avec la fonction f représentée comme un champ de hauteur. Soit$\overline{\alpha}$ une trajectoire concordante avec f. Image prise de [Fer15].

En utilisant la notion de concordance (Eq. F on généralise l'équation F.1 avec M-attributs de la manière suivante :

$$\mathrm{E}(f_1,...,f_k,\Phi) = \sum_{j=1}^{k} \lambda_\mathrm{L}||\Delta f_j||^2 + \frac{(1-\lambda_\mathrm{L})}{t_\mathrm{T}} \sum_{\bar{\alpha_i}\in\Phi^{-1}(j)} Cons_w(f_j,\bar{\alpha_i})$$

Cette équation nous permet de dériver les champs d'attributs f_j et grouper les trajectoires en dépendant de sa concordance avec chaque champ. En utilisant l'équation F, on définit l'ajustement du champ vectoriel comme la minimisation de :

$$\mathrm{E}'(f,\mathrm{T}') = \lambda_\mathrm{L}||\Delta f_j||^2 + \frac{(1-\lambda_\mathrm{L})}{t_\mathrm{T}} \sum_{\bar{\alpha_i}\in\mathrm{T}'} Cons_w(f,\bar{\alpha_i})$$

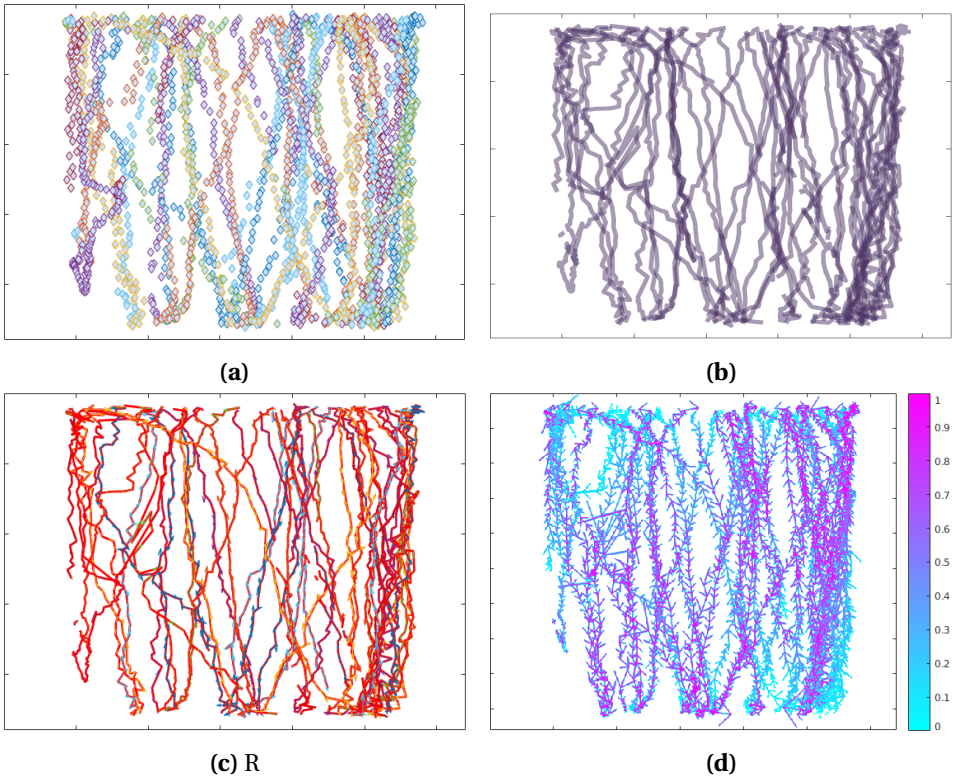
où *T'* est un sous-ensemble de T donné pour chaque cluster K. On résout cette équation avec la méthode des moindres carrés sur chaque attribut $f^{(d)}$ pour estimer le champ de M-attributs *f*. Ensuite, l'assignation de trajectoires est donnée par la minimisation :

$$\underset{j=1,...,k}{argmin} \quad Cons_w(f_j,\bar{\alpha_i}) \tag{F.2}$$

Annexe G

Représentation du jeu de données

Un autre défi est de trouver la meilleure représentation des informations obtenues pour synthétiser de manière claire les résultats et optimiser le transfert d'informations vers d'autres acteurs [FKSS13][LHW07]. Toutes les méthodes évoquées dans cette section ont comme entrées les trajectoires extraites par un système de suivi de personnes. A partir des celui-ci, on doit extraire des informations et les présenter aux utilisateurs. Pour montrer des données à traiter, on présente dans la figure G.1, 74 trajectoires obtenues à l'entrée d'une boutique. La figure G.1 illustre différents manières de représenter les différentes trajectoires effectuées par les individus dans la scène, en mettant en évidence leurs points de début et de fin de cette trajectoire.



(a) **(b)**

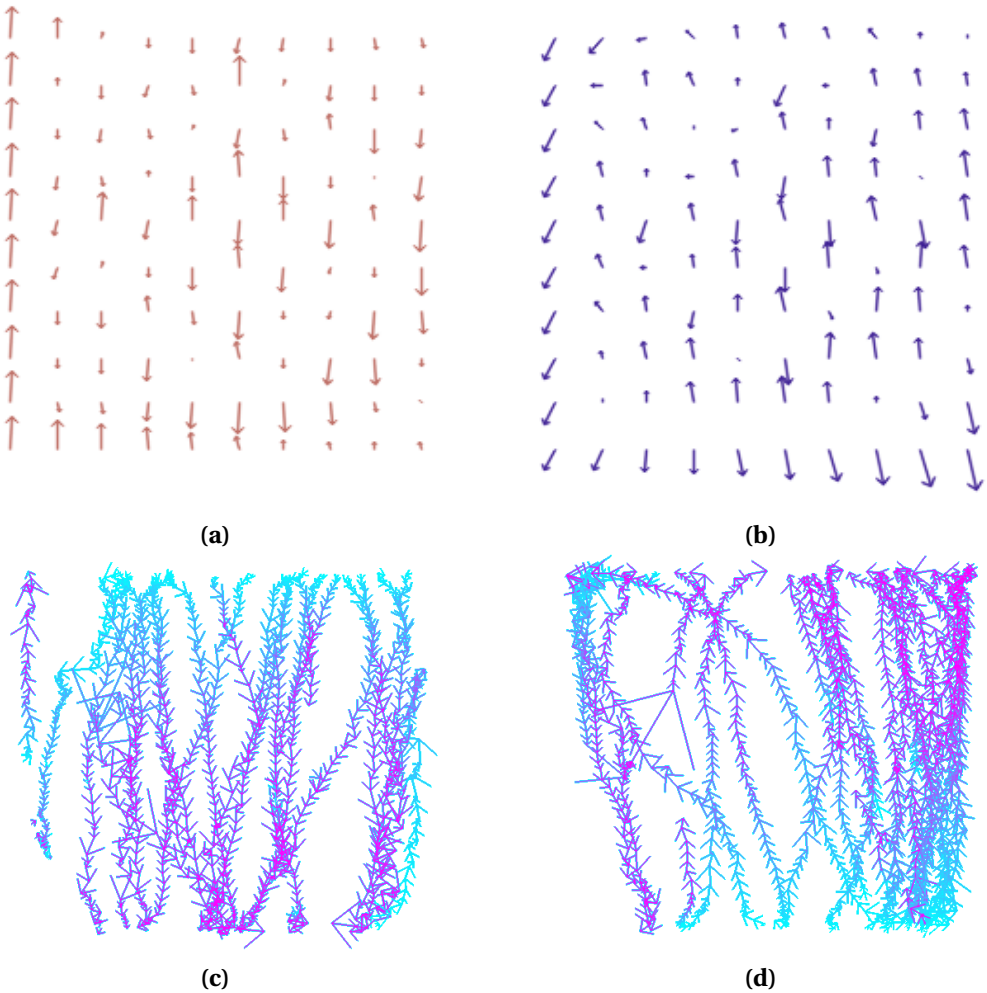
(c) R **(d)**

FIGURE G.1 – Trajectoires reconstituées à partir des données discrètes des trajectoires.

Annexe H

Résultats AFKM sur JD02

Pour analyser le JD02 avec la méthode AFKM, on utilise les attributs de la composante x et y du vecteur de vitesse. Les figures représentent dans cette annexe sont les flux comportementaux et les trajectoires segmentées entre 2 et 7 clusters sauf pour K = 4. On teste la segmentation du JD02 avec une résolution de 10 × 10 et un lissage de 0.5.



(a) **(b)** **(c)** **(d)**

FIGURE H.1 – Représentation d'une segmentation du flux comportemental à R10C2 clusters du JD02 en utilisant la méthode AFKM.

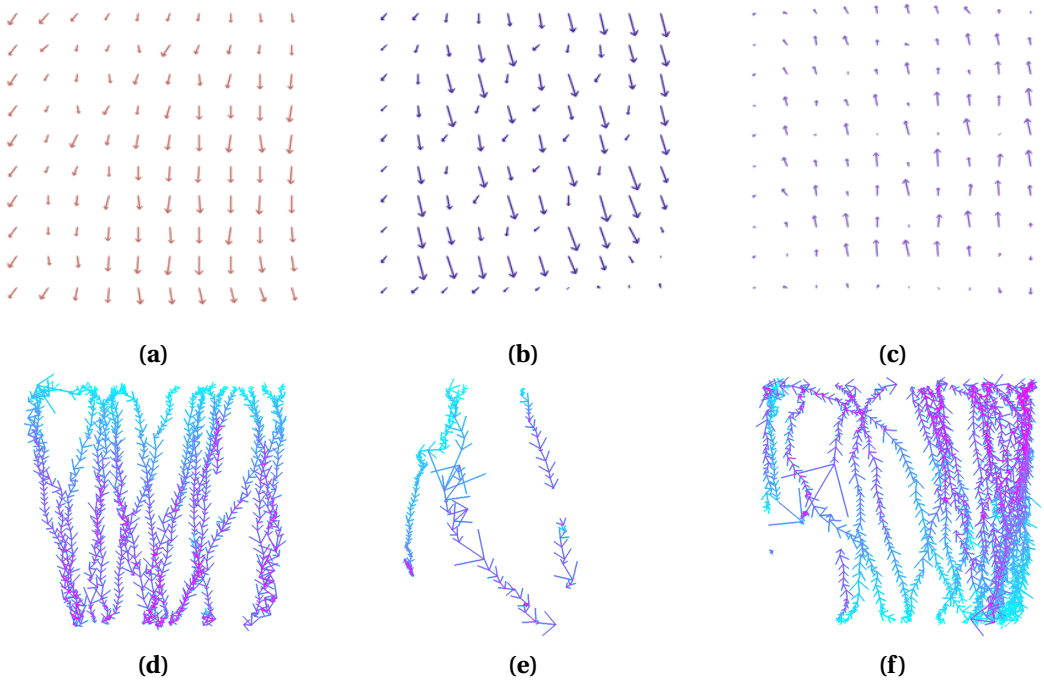


FIGURE H.2 – Représentation d'une segmentation du flux comportemental à R10C3 clusters du JD02 en utilisant la méthode AFKM.

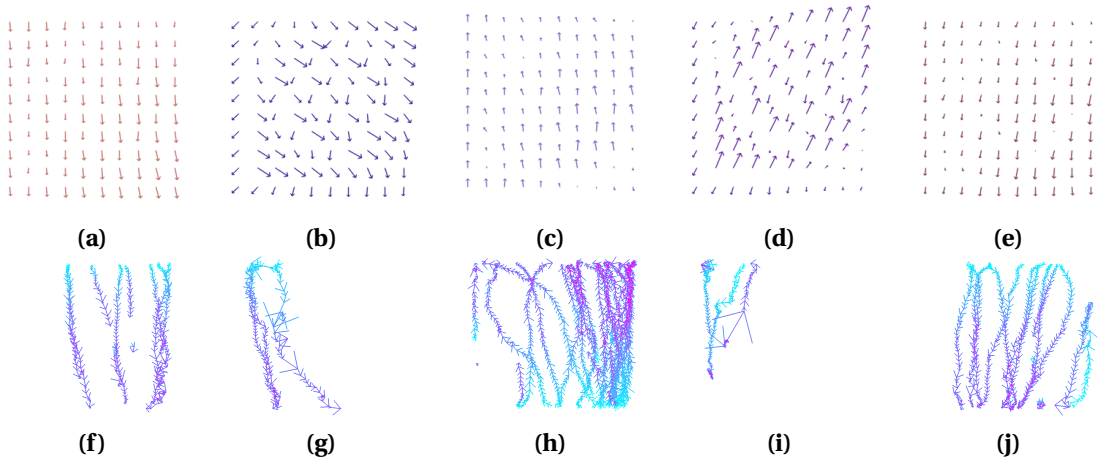


FIGURE H.3 – Représentation d'une segmentation du flux comportemental à R10C5 clusters du JD02 en utilisant la méthode AFKM.

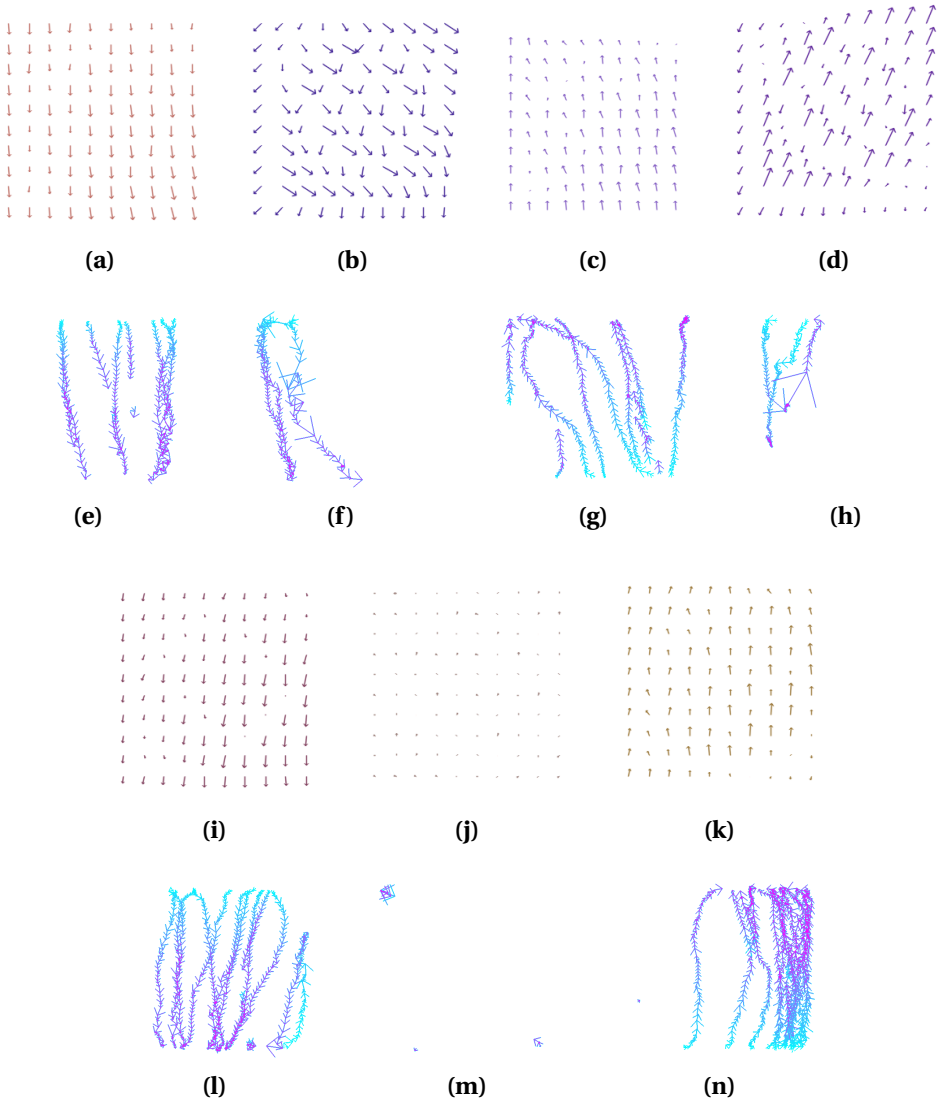


FIGURE H.4 – Représentation d'une segmentation du flux comportemental à R10C7 clusters du JD02 en utilisant la méthode AFKM.

Annexe I

Théorème : SVD Décomposition en valeurs singulières

Theorème : Toute valeur m par n de la matrice A, pour lequel $m \leq n$, peut être écrit comme suit :

$$\begin{gathered} \underbrace{A}_{m\times n} = \underbrace{O_1}_{m\times n}\underbrace{\Sigma}_{n\times n}\underbrace{O_2}_{n\times n} \\ O_1^T O_1 = O_2^T O_2 = I \end{gathered} \tag{I.1}$$

Σ est diagonale et O_1 , O_2 sont orthogonaux

Annexe J

HOG - Complete detection algorithm

Note : Extrait de la thèse de Dalal [Dal06].

The complete detection algorithm is presented in figure. The stride is usually taken to be equal to the cell size in R-HOG and to the diameter of the centre-cell in C-HOG. A scale stride S_r of 1.05 works well in practice. We use σ_s = log(1.3) and [σ_x, σ_y] proportional to the aspect ratio of the detection window times the stride N_s , e.g. for the default person detector [σ_x, σ_y] = [8, 16].

(a) **(b)**

FIGURE J.1 – Classifier response for a dense scan (at every pixel) of the person detector at a scale level.(a) The input image. The highlighted image region corresponds to the detector's normalized image window size. (b) The detector response after a dense scan of the input image at the given scale level. The maximum of the response is indeed centered on the highlighted image region. Note the distribution of the response at the centre of the image region. It is vertically stretched and proportional to the aspect ratio of the detection window.

We stop mean shift iterations when the mean shift vector [Dal06] is below some user supplied threshold. Due to numerical inaccuracies and irregular structure, after convergence, usually several points in the basin of the attraction of a mode cluster around the true location of the mode. We group these mode candidates using a proximity measure. The final location is the mode corresponding to the highest density.

Theoretically, the mean shift vector may converge to a saddle point, but for this application we do not observe this case in practice. In our experiments we find that the points take 4–5 iterations to converge to modes. Thus a naive implementation of mean shift is sufficient. However if the detector is a weak classifier, it may result in lots of detections and can considerably slow down the non-maximum suppression.

TABLEAU J.1 – The complete object detection algorithm

Input : a) Test image b) Trained window classifier with normalised window of width W_n and height H_n c) Scale step size S_r, stride for spatial scan N_s, and sigma values σ_x, σ_y, and σ_s **Output :** Bounding boxes of object detections
Initialise a) Let start scale $S_s = 1$; compute end scale $S_e = min(\frac{W_i}{W_n}, \frac{H_i}{H_n})$, where W_i and H_i are image width and height, respectively b) Compute the number of scale steps S_n to process $S_n = floor(\frac{log(\frac{S_e}{S_s})}{log S_r}) + 1$
For each scale $S_i = [S_s, S_s S_r, ..., Sn]$ a) Rescale the input image using bilinear interpolation b) Extract features and densely scan the scaled image with stride N_s for object/non-object detections c) Push all detections with $t(w_i) > c$ to a list
Non-maximum suppression a) Represent each detection in 3-D position and scale space y_i b) We use scale dependent covariance matrices [Dal06] to compute the uncertainty matrices H_i for each point c) Compute the mean shift vector [Dal06] iteratively for each point in the list until it converges to a mode d) The list of all of the modes gives the final fused detections e) For each mode compute the bounding box from the final centre point and scale

Annexe K

Classification des caméras intelligentes

Cette tableau décribe 15 travaux differents de systèmes de caméras intelligentes. Cette table est extrait de [RWS+08].

TABLEAU K.1 – Classification des caméras intelligentes

	Platform Capabilities				Distributed Processing	Autonomy		
System	**Sensor**	**CPU**	**Comm.**	**Power**		**Deployment**	**Configuration**	**Mobility**
Moorhead and Binnie [MB99]	CMOS	Custom logic for on-chip edge detection	N/A	Mains	Local image analysis; no collaboration	Static	Unknown	Static
VISoc (Albani) [ACC+]	CMOS 320 x 256	32-bit RICS and vision/neural processor	N/A	Battery	Local image analysis; no collaboration	Static	Unknown	Static
Wolf [WOL]	Hi8 Camcorder, NTSC	PC with TriMedia TM-1300 boards	N/A	Mains	Local image analysis; no collaboration	Static	Unknown	Static
Single SmartCam (Bramberger, Rinner) [BBRS]	Color, VGA	DSP	N/A	Mains	Local image analysis; no collaboration	Static	Remote configuration	Static
TRICAM [ABL]	Video in (no sensor)	DSP and FPGA, 128MB RAM	Ethernet	Mains	single node; multiple video inputs	Static	Static configuration	Static
Bauer [BBD+]	Neuromorphic sensor (64×64 pixels)	Blackfin DSP	N/A	Mains	Local image analysis; no collaboration	Static	Unknown	Static
Dias and Berry [DBSM]	2048×2048, gyroscope and accelerometer	Altera Stratix FPGA	Firewire (1394)	Mains	Local image analysis; no collaboration	Static	Active vision	Static
Distributed SmartCam (Bramberger, Quaritsch, Rinner)[BDM+]	VGA	ARM and multiple DSPs	100Mbps Ethernet, GPRS	Mains	Local image analysis; cooperative tracking	Static	Remote and dynamic configuration	Static
BlueLYNX (Fleck)[FBSs]	VGA	PowerPC, 64MB RAM	Fast Ethernet	Mains	Local image preprocessing; central reasoning	Static	Unknown	Static
GestureCam (Shi)[ST]	CMOS, 320x240 (max. 1280x1024)	Xilinx VirtexII FPGA; custom logic plus PowerPC core	Fast Ethernet	Mains	Local image analysis; no collaboration	Static	Unknown	Static
CMUcam 3 (Rowe)[RGGN07]	Color CMOS, 352×288	ARM7 at 60MHz	None onboard (802.15.4 via FireFly mote)	Battery	Local image analysis; inter-node collaboration [RGR]	Static [RGR]	Self configuration with initial learning phase [RGR]	Static
Cyclops (Rahimi)[RBI+]	Color CMOS, 352×288	ATmega128 at 7.3MHz	None onboard (802.15.4 via MicaZ mote)	Battery	Collaborative object tracking[MPK]	Static	Dynamic clustering and cluster head election [MPK]	Static
Meerkats (Margi)[MPOM]	Webcam, 640×480	StrongARM at 400MHz	802.11b	Battery	Local image analysis; collab. object tracking; image transmission to central sink[BLM+]	Static	Static	Static
MeshEye (Hengstler)[HPFA]	2× low resolution sensor, 1× VGA color CMOS sensor	ARM7 at 55MHz	802.15.4	Battery	Unknown	Unknown	Unknown	Unknown
WiCa (Kleihorst)[KASD]	2× color CMOS sensor, 640×480	Xetal 3D (SIMD)	802.15.4	Battery	Local processing; collab. reasoning [WAK]	Static	Static configuration	Static

Liste des acronymes

AFKM *Attributes Field k-means.* XXVII, XXVIII, XXXIII, 125, 139, 140

API *Application Programming Interface.* 32, 33

C-HOG descripteur HOG circulaire. XIII, XIV

CMOS *Complementary Metal-Oxide-Semiconductor*. 31, 32

CPU *Central Processing Unit*. 55

DSP *Digital Signal Processsors* . 55

FPGA *Field Programmable Gate Array*. 31, 55

GPS *Global Positioning System.* 15, 52

IoT *Internet of Things.* 55

LADAR *LAser Detection And Ranging*. 16

LIDAR *LIght Detection And Ranging*. 16, 19, 52

LMS *Laser Measurement System.* 19

PME Petites et les Moyennes Entreprises. 5

POE *Power Over Ethernet*. 80

R-HOG descripteur HOG rectangulaire. XIII, XIV

RAM *Random Access Memory*. 54, 82, 84, 92, 93

RGB *Red – Green – Blue.* 31, 43

RGB-D *Red – Green – Blue – Depth.* 26, 45

ROS *Robot Operating System.* 31, 32

RTCC *Real-Time Clock and Calendar*. 80, 81

SCG *Système de Coordonnées Globales.* 105, 107

SDK *Software Development Kit*. 31–33

SIG *Système d'Identification Globale.* 105

SLAM *Simultaneous Localization And Mapping*. IV

SVD *Singular value decomposition.* III

VFKM *Vector Field k-means.* XXVII, XXVIII, 125, 139–141

Glossaire

ATR Institut international de recherche sur les télécommunications avancées. 2

BB Boîte englobante. 51, 52

CIFRE Le dispositif CIFRE permet aux entreprises de bénéficier d'une aide financière pour recruter de jeunes doctorants dont les projets de recherche, menés en liaison avec un laboratoire extérieur, conduiront à la soutenance d'une thèse.. 5, 25, 50

DF Arbres de décision renforcés. 53

DPM Deformable Part Model en francçais modèle de pièces déformables. XVIII, 52, 53

FoV Champ de vue. 26, 38, 40, 47, 58, 78, 80, 109

fps Unité de mesure correspondant au nombre d'images affichées (traité ou acquis) en une seconde par un caméra. 9, 31, 32, 34, 37, 55

GSS Stabilité globale du signal. 38–40

HDR fiabilité de détection humaine. 43

HFD vecteur de caractéristiques de la forme humaine. 67, 68, 72–76, 101, 106, 108, 109

HOG Histogram of oriented gradients. XIII, XIV, XVII, XVIII, XLIII, 52, 53, 65

IMU Centrale inertielle. 15, 31, 52

IR Le rayonnement infrarouge est un rayonnement électromagnétique d'une longueur d'onde supérieure à celle du spectre visible mais plus courte que celle des micro-ondes. 15, 16, 24, 25, 32, 33

NTP Protocole d'heure réseau. 81

POI . 118, 120, 122, 124

RFID Identification par radiofréquence. 4, 15

RMSE erreur quadratique moyenne. 38–40

RnD Le concept de « recherche et développement » est une traduction littérale de l'anglais *research and development*. 6

ROI Région d'intérêt. 76–78

SoC Système sur une puce. 6, 55

SVM support vector machine. XIV, 52, 65, 66

ToF Le temps de vol est une méthode pour l'acquisition des images de profondeur. 7, 18, 19

ZCS zone de surveillance commune. 28, 29

ZTB zone temporairement avaugle. 28, 29

ÉCOLE DOCTORALE

Sciences et technologies de l'information et de la communication (STIC)

Titre : Système de caméras intelligentes pour l'étude en temps-réel des personnes en mouvement.

Mots clés : caméra intelligente, systèmes embarqués, suivi de personnes, réseau de capteurs.

Résumé : Nous proposons un système de détection et de suivi des personnes en mouvement dans des grands espaces. Notre solution repose sur un réseau de caméras intelligentes pour l'extraction des informations spatio-temporelles des personnes. Les caméras sont composées d'un capteur 3D, d'un système embarqué et de communication. Nous avons montré l'efficacité du placement des capteurs 3D en position zénithale par rapport aux occultations et variations d'échelle. Nous garantissons l'exécution des traitements en temps-réel (~20 fps), permettant de détecter des déplacements rapides avec une précision jusqu'à 99 %, et capable d'un filtrage paramétrique des cibles non désirées comme les enfants ou les caddies.
Nous avons réalisé une étude sur la viabilité technologique des résultats pour de grands espaces, rendant la solution industrialisable.

Title : Smart camera system for kinetic behavior study in real-time.

Keywords : Embedded system, smart cameras, people tracking, sensor networks.

Abstract : We propose a detection and tracking system of people moving in large spaces system. Our solution is based on a network of smart cameras capable of retrieving spatiotemporal information from the observed people. These smart cameras are composed by a 3d sensor, an onboard system and a communication and power supply system. We exposed the efficacy of the overhead position to decreasing the occlusion and the scale's variation. Finally, we carried out a study on the use of space, and a global trajectories analysis of recovered information by our and others systems, able to track people in large and complex spaces.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France