



**HAL**  
open science

# Déploiement d'un système de suivi par vidéo et analyse comportementale : application à la surveillance urbaine

Amaury Auguste

## ► To cite this version:

Amaury Auguste. Déploiement d'un système de suivi par vidéo et analyse comportementale : application à la surveillance urbaine. Traitement du signal et de l'image [eess.SP]. Université de Bretagne occidentale - Brest, 2022. Français. NNT : 2022BRES0066 . tel-03922664

**HAL Id: tel-03922664**

**<https://theses.hal.science/tel-03922664>**

Submitted on 4 Jan 2023

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

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

# THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE BRETAGNE OCCIDENTALE

ÉCOLE DOCTORALE N° 601  
*Mathématiques et Sciences et Technologies  
de l'Information et de la Communication*  
Spécialité : *Signal, Image, Vision*

Par

**Amaury AUGUSTE**

**Déploiement d'un système de suivi par vidéo et analyse comportementale.**

Application à la surveillance urbaine.

Thèse présentée et soutenue à Brest, le 26 Septembre 2022  
Unité de recherche : L@BISEN

## Rapporteurs avant soutenance :

Denis HAMAD    HDR - Professeur à ULCO  
Fadi DORNAIKA    HDR - Professeur à EHU

## Composition du Jury :

Président :	Christian BROSSEAU	Professeur à UBO
Examineurs :	Denis HAMAD	HDR - Professeur à ULCO
	Fadi DORNAIKA	HDR - Professeur à EHU
	Badr-Eddine BENKELFAT	HDR - Professeur à Télécom SudParis
	Marwa EL BOUZ	Enseignante-Chercheuse et responsable équipe LSL - Yncréa Ouest
Directeur de la thèse :	Ayman AL FALOU	Directeur de la recherche à Yncréa Ouest
	Ayman AL FALOU	Directeur de la recherche à Yncréa Ouest

## Invité(s) :

Wissam KADDAH    Enseignant-Chercheur - Yncréa Ouest  
Ghislain OUDINET    Enseignant-Chercheur - Yncréa Méditerranée



# REMERCIEMENT

---

Je remercie Denis HAMAD et Fadi DORNAIKA d'avoir accepté de rapporter ce manuscrit de thèse. Je tiens également à remercier Badr-Eddine BENKELFAT d'avoir accepté de faire partie des membres du jury et Christian BROSSEAU d'avoir accepté la responsabilité de le présider.

Je tiens à remercier Ayman AL FALOU qui a accepté de diriger cette thèse, Marwa EL BOUZ et Ghislain OUDINET qui ont accepté de la co-encadrer et qui ont pu m'aider par leurs soutiens et leurs accompagnements durant ces années. Je remercie aussi l'équipe L@BISEN et plus particulièrement Wissam KADDAH pour l'ensemble de ses conseils et pour sa participation active au suivi de mes recherches.

Je remercie l'ensemble des équipes d'Yncréa Méditerranée de m'avoir apporté leurs soutiens et leurs expertises. Je remercie Stéphane VERA pour sa contribution et son investissement qu'il fut pour m'aider à trouver ce sujet de thèse ou pour me permettre d'en arriver à sa conclusion dans les meilleures conditions.

Je remercie mes amis qui ont vécu cette thèse avec moi et qui ont su me remotiver quand c'était nécessaire.

Enfin, je tiens à remercier mes parents et ma famille pour leurs bienveillances et leurs conseils, mais surtout de m'avoir toujours soutenu, quelles que puissent être mes décisions.



# TABLE DES MATIÈRES

---

<b>1</b>	<b>Contexte et état de l'art</b>	<b>11</b>
1.1	Contexte . . . . .	11
1.1.1	Formulation des problématiques . . . . .	13
1.1.2	La méthodologie de travail employée . . . . .	17
1.2	État de l'art . . . . .	20
1.2.1	L'analyse comportementale . . . . .	21
1.2.2	La détection vidéo . . . . .	22
<b>2</b>	<b>Base de données et Simulateur</b>	<b>29</b>
2.1	Préambule . . . . .	29
2.2	Base de données . . . . .	29
2.2.1	Introduction . . . . .	29
2.2.2	Le besoin . . . . .	30
2.2.3	Alternative aux vidéos de surveillance . . . . .	31
2.2.4	Choix du matériel . . . . .	31
2.2.5	Choix des scénarios . . . . .	32
2.2.6	Contrainte de l'acquisition vidéo . . . . .	33
2.2.7	L'extraction d'informations . . . . .	36
2.2.8	Expérimentation sur l'extraction d'information . . . . .	37
2.2.9	Conclusions des expérimentations . . . . .	39
2.2.10	Estimation de l'environnement . . . . .	40
2.2.11	Expérimentations sur les mires . . . . .	41
2.2.12	L'anonymisation des données . . . . .	45
2.3	Simulateur . . . . .	46
2.3.1	Le besoin . . . . .	46
2.3.2	Le Rejeu . . . . .	50
2.3.3	Le Suivi de chemin . . . . .	51
2.3.4	Les Boids . . . . .	59
2.3.5	L'optimisation du temps de traitement . . . . .	64

2.3.6	Identification des causes du temps de traitement . . . . .	67
2.3.7	L'optimisation de la perspective . . . . .	72
<b>3</b>	<b>Analyse comportementale de foule</b>	<b>77</b>
3.1	Préambule . . . . .	77
3.2	Le regroupement de points . . . . .	80
3.2.1	Introduction . . . . .	80
3.2.2	Classification automatique . . . . .	82
3.2.3	Expérimentation préliminaire . . . . .	85
3.2.4	Amélioration de la méthode . . . . .	86
3.2.5	Conclusion des expérimentations . . . . .	88
3.3	Analyse d'écart . . . . .	89
3.3.1	Introduction . . . . .	89
3.3.2	Cas d'études . . . . .	90
3.3.3	Cas particuliers . . . . .	90
3.3.4	Gestion des cas particuliers . . . . .	92
3.3.5	Résultats et discussions . . . . .	95
3.3.6	Amélioration de l'analyse . . . . .	96
3.3.7	Division en secteur d'intérêt . . . . .	97
3.3.8	Conclusion . . . . .	99
3.4	Les heatmaps . . . . .	100
3.4.1	Introduction . . . . .	100
3.4.2	Principe de fonctionnement . . . . .	102
3.4.3	Expérimentation . . . . .	102
3.4.4	Résultat et discussions . . . . .	104
3.4.5	Conclusion . . . . .	106
<b>4</b>	<b>Suivi individualisé (Analyse individuelle)</b>	<b>107</b>
4.1	Préambule . . . . .	107
4.2	Le suivi . . . . .	112
4.2.1	Introduction . . . . .	112
4.2.2	Le regroupement . . . . .	113
4.2.3	Interprétation des données . . . . .	115
4.2.4	Résultats . . . . .	119
4.3	Prédiction de trajectoire . . . . .	119

4.3.1	Introduction . . . . .	119
4.3.2	Principe de fonctionnement . . . . .	121
4.3.3	Correspondance réalité et prédiction . . . . .	127
4.3.4	Discutions et conclusion . . . . .	129
4.3.5	Conclusion . . . . .	131
<b>5</b>	<b>Perspectives Simulateur et Analyseur et Livrable</b>	<b>133</b>
5.1	Livrable . . . . .	133
5.2	Perspectives simulateur et analyseur . . . . .	135
5.2.1	Simulateur . . . . .	135
5.2.2	Analyseur . . . . .	136
5.2.3	Intelligence Artificielle . . . . .	140
<b>A</b>	<b>Fiche technique AXIS M1123-E Network Camera</b>	<b>145</b>
	<b>Bibliography</b>	<b>147</b>





# INTRODUCTION

---

De nos jours, les villes cherchent à répondre à de nombreuses problématiques afin d'améliorer la qualité de vie de ses citoyens. Cette amélioration se fait par la création de nouveaux établissements ou d'espaces de loisirs, mais aussi au travers de l'amélioration constante du niveau de sécurité. Nous pouvons malheureusement être témoins ou victimes de nombreux événements ayant des conséquences parfois tragiques. C'est donc dans ce contexte que de nombreux projets visant à améliorer la sécurité ont vu le jour.

Pour s'assurer que cette sécurité soit la plus efficace possible, les villes investissent dans des dispositifs innovants. Nous pouvons par exemple voir l'émergence de boutons d'appel d'urgence disposés dans les différents endroits de la ville afin de permettre à chaque citoyen de contacter les secours en cas de besoin. Ces dispositifs viennent en complément des caméras de vidéosurveillance présentes dans les villes et généralement utilisées par un centre de surveillance urbaine qui permettra de faire des demandes d'intervention en cas d'anomalie observée.

L'usage de ces dispositifs permet aux services de sécurité de la ville d'agir rapidement et d'être ainsi le plus efficace possible. Ces dispositifs de plus en plus efficaces restent en demande permanente d'amélioration. C'est dans ce contexte que cette thèse a vu le jour, au travers d'une demande formulée par le projet SafeCity visant à renforcer la sécurité des espaces urbains.

L'objectif de cette thèse est de venir améliorer la détection de comportements anormaux au travers des caméras de la ville déjà en place. Cette amélioration se fera au travers de l'utilisation d'outils numériques permettant aux opérateurs utilisant les caméras d'être notifiés des événements anormaux. La réussite de cet objectif permettra alors à ces opérateurs d'être informés de la caméra sur laquelle il est nécessaire d'avoir une analyse visuelle humaine afin de permettre l'envoi de secours sur place si nécessaire.

Le chapitre 1 de cette thèse présente l'état de l'art ainsi que le contexte global de l'étude au travers des différents échanges qu'il y a eu au sein du projet SafeCity et des différentes décisions qui ont pu influencer les recherches.

Le chapitre 2 met en avant la constitution de la base de données qui a permis la validation des différents outils. Ce chapitre met notamment en avant des outils de simulation innovante permettant de constituer nos propres bases de données.

Le chapitre 3 présente la première méthode que nous avons imaginée afin de répondre aux différents besoins du projet. Ce chapitre aborde principalement une analyse de manière statistique qui se base sur une situation immédiate. Cette situation est directement extraite d'une analyse des caméras de vidéosurveillance.

Le chapitre 4 propose des solutions permettant non seulement d'améliorer la méthode proposée dans le chapitre 3, mais également d'offrir la possibilité d'un suivi individualisé au travers de prédiction. Ce suivi basé sur les filtres de Kalman amène la particularité de pouvoir être fait de manière anonyme.

Enfin, le chapitre 5 met en avant les perspectives envisageables à la suite des différents résultats que nous avons pu obtenir et fait également la synthèse du livrable du projet SafeCity qui a eu lieu au cours de la thèse.

# CONTEXTE ET ÉTAT DE L'ART

## 1.1 Contexte

Le développement des villes ne cesse de s'accélérer avec l'urbanisation galopante à la surface du monde. En plus de la population grandissante, les villes doivent faire face à des menaces de plus en plus importantes, à la fois liées aux risques naturels dont le dérèglement climatique est une composante de plus en plus forte et aux risques d'origine humaine, dont la criminalité et le terrorisme qui se développent dans toutes les régions du monde. En parallèle, le développement des technologies permet aux villes d'intégrer de nouveaux moyens leur permettant d'accéder au rang de villes intelligentes. C'est dans ce contexte que le projet Safecity a vu le jour. Son objectif est d'amener des solutions à ces problématiques en se servant de "l'intelligence" de ses villes. La figure 1.1 présente un peu



FIGURE 1.1 – Coordination et collaboration des différentes entités impliquées dans la sécurité des citoyens.

plus en détail les capacités offertes par la solution. Elle fait apparaître clairement

les systèmes métier en bas (poste de commande mobile, patrouille connectée, sécurité route, sécurité école) et la partie fédératrice du système City Manager. Le projet permet de construire une solution cohérente et ouverte, permettant par la suite, d'adresser des marchés soit de demande globale d'une solution complète, soit seulement certaines parties.

La figure 1.2 présente la plate-forme City Manager et ses capacités de collecte, de traitement et de présentation des données de la ville pour augmenter l'efficacité de la gestion de la sécurité du citoyen : Le consortium est constitué de 15 parte-



FIGURE 1.2 – Les grandes fonctions

naires qui apportent leurs contributions sur certaines composantes de la solution globale. Cela permet à chacun d'accélérer le développement de sa propre solution, tout en visant une cohérence d'ensemble qui permettra de reconduire les accords entre les industriels pour fournir, en particulier sur le marché international, une solution déjà intégrée et efficiente.

Le projet porté par Thalès regroupe 12 autres industriels incluant des ETI/PME (Arclan, BCA, Deveryware, Egidium, Gemalto, Geol Semantics, IGO, Luceor, ONHYS, IDEMIA, SIS, Sysnav) et deux institutions académiques (INRIA et YNCREA).

### 1.1.1 Formulation des problématiques

Arclan, BCA, Yncrea et Gemalto participent, dans le cadre du lot 6, à la construction d'un système de protection des établissements scolaires, pour lesquels nous prévoyons la mise en place d'équipements périmétriques (portail intelligent, système de surveillance vidéo) et internes (boîtier d'alarme, porte intelligente) garantissant au personnel enseignant et aux responsables de la sécurité un comptage précis des personnes présentes dans l'établissement et la détection très rapide des comportements à risques. Pour atteindre les objectifs de ce lot, le groupe a défini

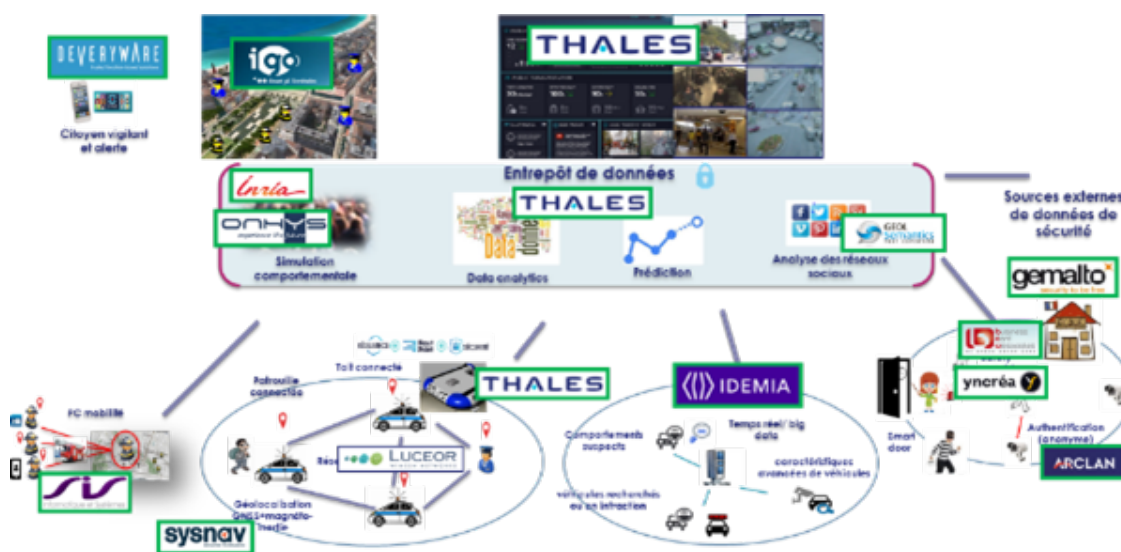


FIGURE 1.3 – Architecture générale, site de Nice

trois phases successives de déploiement, notées v1 à v3, qui viseront à tester un ensemble de simulation :

Phase	Démonstration
v1	<p>Gestion du risque inondation :</p> <ul style="list-style-type: none"> <li>● Boîtiers d'alertes et solution de supervision</li> <li>● Prototype de portail : comptage.</li> </ul>
v2a	<p>Sécurisation d'un établissement : exercice PPMS Alerte Attentat-Intrusion :</p> <ul style="list-style-type: none"> <li>● Boîtiers d'Alerte évacuation ou confinement.</li> <li>● Wearables pour élèves et adultes.</li> <li>● Portail pour la gestion de l'accès.</li> <li>● Serveur école pour le comptage des personnes présentes par catégorie et la gestion des pré-alertes.</li> <li>● Solution de supervision centralisée.</li> <li>● Détection intrusions.</li> </ul>
v2b	<p>Sécurisation d'un établissement : 2a + utilisation des moyens de détection vidéo :</p> <ul style="list-style-type: none"> <li>● Boîtiers d'Alerte évacuation ou confinement.</li> <li>● Wearables pour élèves et adultes.</li> <li>● Portail pour la gestion de l'accès.</li> <li>● Serveur école pour le comptage des personnes présentes par catégorie et la gestion des pré-alertes.</li> <li>● Différenciation adulte enfant, reconnaisse adulte.</li> </ul>

Phase	Démonstration
v3	<p>Gestion de crise :</p> <ul style="list-style-type: none"> <li>• Comptage des personnes présentes par catégorie, alerte confinement et flux vidéo école autorisés</li> <li>• Détection intrusions et comportements anormaux</li> </ul> <p>Arclan :</p> <ul style="list-style-type: none"> <li>• Analyses sonores</li> <li>• Détection comportements anormaux (sonore et visuelle ; ex. détection morphologie)</li> <li>• Alarme silencieuse (ex. salle de sieste)</li> <li>• Auto-confinement (à définir)</li> </ul> <p>BCA :</p> <ul style="list-style-type: none"> <li>• Comptage absolu avec détection de sens</li> <li>• Détection de présence : dans l'établissement / dans une salle</li> <li>• SmartDoor</li> </ul> <p>Thales DIS :</p> <ul style="list-style-type: none"> <li>• Badge d'accès temporaire</li> <li>• Reconnaissance faciale dans l'IR</li> </ul> <p>Yncréa :</p> <ul style="list-style-type: none"> <li>• Analyse comportementale</li> <li>• Wearable : montre fonctionnelle / OS passeport / wake up radio (sortie école, alerte)</li> <li>• Application mobile</li> </ul>

TABLE 1.1: Objectifs fonctionnels des expérimentations

Comme le montre le tableau 1.1, chacun des partenaires aura à proposer des solutions sur des thèmes différents. En ce qui concerne les travaux présentés dans cette thèse, les efforts se concentreront principalement sur le sujet de l'analyse comportementale. Parmi les objectifs à atteindre, nous retrouvons notamment :

- Analyse des mouvements de foule à proximité de l'école.
- Détection de sédentarité individuelle ou en groupe.

Afin d'atteindre ses objectifs, nous avons identifié et proposé trois scénarios autour desquels a lieu l'intégralité des recherches en ce qui concerne l'analyse comportementale :



- La détection de dispersion,
- la détection de rassemblement,
- la détection de maraudage.

Un élément notable de l’architecture de ce projet est que la partie concernant l’analyse comportementale est relativement indépendante du reste du projet. En effet comme le montre la figure 1.4, l’ensemble des solutions proposées par les partenaires doivent pouvoir communiquer entre elles sur un même réseau, ce qui n’est plus le cas pour cette partie. En effet, dans le cas de la partie concernant l’analyse

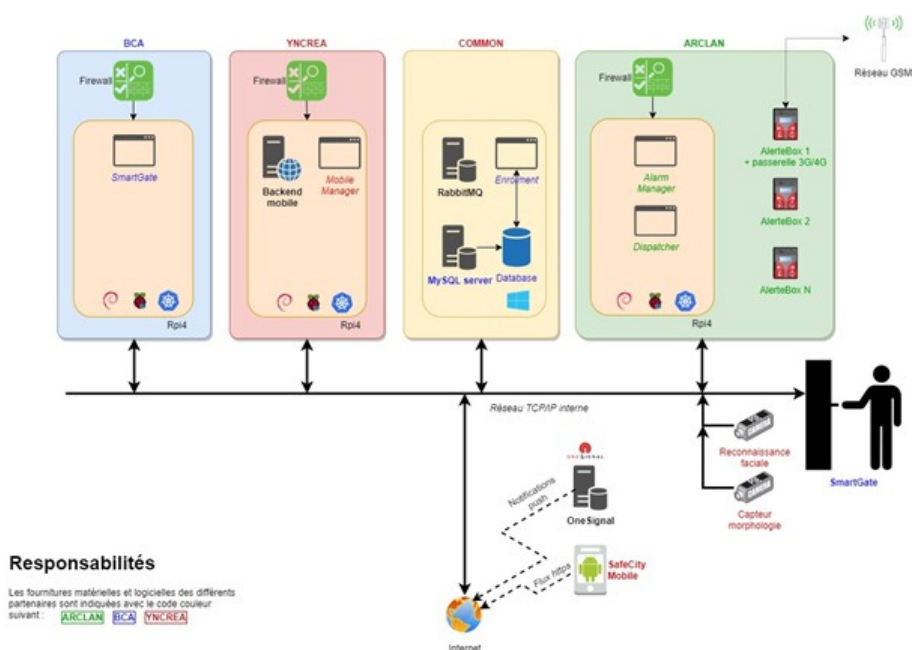


FIGURE 1.4 – Architecture générale, site de Nice

comportementale, il était initialement prévu d’installer des dispositifs d’acquisition vidéo tels que des caméras IP dans l’enceinte de l’école Nikaïa. Cette possibilité ayant été retirée en cours de projet afin de répondre au mieux aux demandes du RGPD (Règlement Général sur la Protection des Données), nous avons dû nous contenter de l’utilisation des caméras déjà mise en place par la ville de Nice dans le cadre de la protection urbaine. De plus, une contrainte liée à l’anonymisation des données nous a été imposée, cette contrainte nous empêche notamment de faire transiter des informations permettant d’identifier une personne sur le réseau. En ajustant le cadre de nos recherches à ces contraintes, nous obtenons l’architec-

ture présentée en figure 1.5. Cette nouvelle contrainte nous permet toutefois de

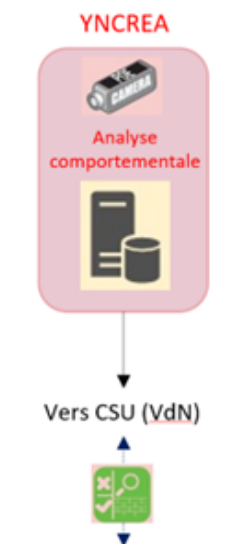


FIGURE 1.5 – Architecture Analyse comportementale V3

répondre au mieux à un autre besoin qui est celui d’alerter le plus rapidement possible les secours. Étant donné que le déploiement se fait sur le réseau de la ville et plus particulièrement sur le réseau de surveillance urbain, la levée d’alerte sera donc plus rapide et moins sensible aux éventuels dysfonctionnements dans la mesure où il n’y a qu’une seule architecture réseau à gérer.

### 1.1.2 La méthodologie de travail employée

Les travaux menés lors de cette thèse ont été pensés de sorte à répondre aux besoins du projet SafeCity. En plus d’apporter un thème aux recherches effectuées, le projet SafeCity est un projet professionnel impliquant de nombreux partenaires. Les travaux ont dû être réalisés de manière itérative afin de répondre aux différents besoins imposés par les jalons du projet. La suite de ce manuscrit présente les travaux réalisés par bloc logique où chacun d’entre eux a été réalisé idéalement sans interruption en respectant les spécifications initiales. Toutefois, il a parfois été nécessaire d’en réaliser certains tout au long du projet en étant soit complété, soit modifié pour s’adapter à de nouvelles contraintes ou imprévu.

Une des contraintes de ce projet réside dans la collaboration de plusieurs partenaires autour d'un même sujet. Dans le cas de notre étude, les éléments à produire ont été réalisés, dès que cela était possible, en restant le plus indépendant de ces partenaires. Toutefois, il a fallu prendre en compte l'ensemble des éléments du projet afin de proposer une solution permettant une intégration dans un système plus complexe.

Comme il a été expliqué en 1, le projet sur lequel repose les recherches de cette thèse se découpe en plusieurs lots. Chaque lot se focalise sur un ensemble de solutions pour un cas d'étude particulier. Le lot qui concerne cette thèse est le lot 6 dont l'objectif est la sécurisation d'une école primaire. Parmi les solutions, certaines d'entre elles permettent de déclencher une alerte afin de prévenir les secours. Toutefois, chacune de ces solutions se trouve sur un réseau restreint, la remontée de ces alertes doit donc se faire par des canaux utilisant par exemple le wifi ou le réseau mobile. Dans la mesure où la solution sur laquelle nous avons travaillé est indépendante de ce réseau, il n'était pas nécessaire d'ajouter des contraintes d'intégration à notre système. Toutefois, étant donné qu'il était question de produire une preuve de concept et non un produit fini les réflexions qui ont permis d'aboutir au système d'analyse comportementale ont été menées en considérant la possibilité de recevoir des alertes externes au système afin d'être complété ultérieurement. Ce point d'entrée aux alertes externe a été pensé de sorte à pouvoir recevoir des messages dans un format standard (JSON) qui pourra être facilement réutilisé en cas de besoin.

Parmi les autres contraintes qu'il a fallu intégrer, il a été nécessaire d'échanger avec la sécurité de la ville afin de mieux comprendre les besoins auxquels cette dernière fait face. Ces échanges ont été d'autant plus nécessaires que l'obtention de données vidéo issues des caméras de la ville s'est révélée être impossible dans la mesure où ces dernières sont à l'usage unique de la sécurité et ne doivent pas être communiqués à l'extérieur afin de respecter le RGPD. Étant donné l'aspect relativement restrictif d'accès aux données de la ville qui nous a été imposé, nous avons fait en sorte de proposer de manière itérative des solutions suffisamment spécifiques pour qu'elle ne laisse pas de doute possible sur leur interprétation. C'est donc à l'issue de ces différents échanges qu'il a été choisi de cibler dans le

cadre du projet un certain nombre d'anomalies urbaines à détecter :

- Détection de dispersion à proximité de l'école,
- Détection de rassemblement à proximité de l'école,
- Détection de maraudage à proximité de l'école.

D'autres scénarios avaient été envisagés notamment sur une étude vidéo à l'intérieur même de l'école. Toutefois, l'ensemble de ces scénarios ont dû être écartés, car certaines décisions qui ont été prises alors même que le projet était déjà en cours de développement nous ont contraints à devoir renoncer à la possibilité d'installer des caméras dans l'enceinte de l'école. Cette décision a donc impacté les possibilités d'analyse qui au départ devait s'orienter sur des analyses plus fines.

Les premières propositions menées autour du projet devaient notamment permettre de détecter la posture d'un individu dans l'enceinte de l'école afin de détecter une attitude inappropriée ou encore détecter un ensemble d'objets dangereux. Ces détections se sont alors révélées impossibles dans la mesure où les seules caméras sur lesquels nous pouvions déployer le système d'analyse étaient celle disposée à l'extérieur de l'école. Ces caméras sont généralement dans des lieux spécifiques sur lesquels il nous est impossible de changer l'orientation et généralement trop hauts pour pouvoir effectuer les analyses voulues.

Le choix des trois scénarios ne s'est toutefois pas fait par cette seule contrainte. Il a été admis, dès lors qu'il était impossible de mettre des caméras dans l'enceinte de l'école, que cette dernière soit un lieu dans lequel les individus s'y trouvant ont forcément dû décliner leur identité. Il est alors apparu évident que pour s'assurer que l'intérieur de la zone soit sécurisé, il fallait mettre en œuvre une solution permettant de s'assurer que cette zone reste saine en analysant ses extérieurs. Les scénarios ont alors été validés par les services de sécurité de la ville qui ont exprimé une problématique qui a été au cœur de nos réflexions, à savoir : comment identifier sur un réseau de caméras, pouvant compter plus d'une centaine d'appareils, la scène qui présente une anomalie nécessitant une intervention rapide ?

De cette problématique à découler plusieurs réflexions qui nous ont demandé de définir dans un premier temps ce qui était considéré comme une anomalie. Il est difficile de lister toutes les anomalies qu'il serait possible de rencontrer dans

une ville. Toutefois, en orientant nos recherches sur l'étude du mouvement, il s'est avéré que nous pouvions mettre en avant certains des cas les plus critiques.

La détection de dispersion permet notamment de détecter un mouvement de foule impactant plusieurs individus. Ce genre de mouvement est généralement anormal dans la mesure où une dispersion peut-être due à un individu ou un événement ayant provoqué une certaine panique auprès des citoyens.

La détection de rassemblement et quant à elle plus modérée, il est normal de pouvoir observer des rassemblements dans un espace urbain. Toutefois, un rassemblement peut être considéré comme anormal s'il a lieu dans des heures ou des lieux inadéquats.

Enfin, le maraudage est le comportement pour lequel le danger est le moins important dans la mesure où ce dernier n'implique pas un danger immédiat. L'objectif de la détection du maraudage est sensiblement le même que celui de la détection de rassemblement, son objectif est d'identifier la présence récurrente d'un individu dans des lieux sensibles avec si besoin la possibilité de définir des plages horaires de surveillance.

Nous avons donc obtenu une validation de la part des services de sécurité de la ville à l'issue de la présentation de ces 3 scénarios qui leur permettrait de couvrir suffisamment de cas anormaux pour permettre aux opérateurs du centre de surveillance urbaine d'être notifiés rapidement d'une potentielle anomalie.

## **1.2 État de l'art**

Le contexte relativement restreint de l'étude permet toutefois d'identifier avec précision l'ensemble des différents axes de recherche qui doivent être abordés. Une étude globale du système que nous souhaitons mettre en place nous a permis de définir que ce dernier se décomposerait en deux grands modules :

- La détection vidéo,
- L'analyse comportementale.

La détection vidéo est dans notre cas la méthode qui sera employée afin d'extraire d'une vidéo des informations de présence d'individus. Cette détection est essentielle afin de pouvoir anonymiser l'ensemble des données qui en seront issues dans le but d'analyser uniquement des données anonymes. L'analyse comportementale repose quant à elle sur l'étude de l'ensemble des comportements que l'on souhaite identifier au travers des scénarios définis en 1.1. Toutefois, afin d'effectuer cette analyse comportementale il a fallu définir ce que représente l'analyse comportementale dans notre cas et quel type d'outils pourrait nous servir afin d'effectuer cette analyse.

### 1.2.1 L'analyse comportementale

L'analyse comportementale est dans la littérature un domaine qui peut prendre plusieurs formes. Une analyse comportementale étant généralement une étude sociologique engageant l'étude de nombreux paramètres physiques ou psychologiques. L'étude de ces paramètres mène alors à des conclusions sur le comportement d'un individu. Cette analyse peut également être sous la forme du suivi direct d'un patient afin d'en déceler, par exemple, un état psychologique préoccupant tel que la possibilité d'un trouble dépressif persistant [Rei+21]. Notre objectif ne nous permet pas de mener une étude sociologique sur les individus, nous devons donc orienter nos recherches sur des méthodes d'analyse comportementale spécifique au domaine du numérique. Il existe par exemple des méthodes permettant de déterminer si le comportement d'un utilisateur sur internet présente dans un contexte particulier un risque nécessitant que l'on s'en préoccupe [RK22].

En orientant nos recherches sur l'analyse comportementale numérique nous nous apercevons qu'il est difficile de faire une corrélation entre un comportement numérique et un comportement observable au travers de caméras. L'analyse comportementale qui nous incombe est dépendante d'un comportement humain d'une part, mais aussi de l'environnement dans lequel le comportement a eu lieu. Nous comprenons alors qu'un comportement pouvant être détecté comme dangereux dans un contexte psychologique ne se retranscrirait pas de la même manière dans un environnement urbain.

Nous pouvons tout à fait imaginer effectuer une analyse comportementale sur une personne dans un environnement médical afin de déceler d'éventuelles pathologies psychologiques lors d'études relativement poussées et longues. Dans notre cas, nous chercherons une méthode afin de détecter un comportement anormal le plus vite possible afin de mettre en sécurité rapidement les personnes se trouvant à proximité de la situation ou de l'individu dangereux.

Il se trouve qu'il est difficile de trouver de la littérature sur les aspects d'analyse de comportement à une échelle aussi globale que la surveillance vidéo urbaine. Il est donc intéressant de disposer dans notre cas de scénarios spécifiques à étudier et de pouvoir mener des réflexions sur la normalité d'une scène.

### 1.2.2 La détection vidéo

Parmi les méthodes les plus populaires d'analyse comportementale nous retrouvons notamment l'analyse de scène globale reposant principalement sur l'étude du mouvement de foule dans des environnements chargés [WM10; RML17]. L'utilisation de ces méthodes permet une étude globale du mouvement dans l'image, mais ne permet pas une identification formelle de chaque individu. La majorité de ces méthodes emploie une soustraction de fond permettant alors de ne se focaliser que sur les éléments mobiles de la vidéo [ZL10].

Les méthodes basées sur de la soustraction de fond permettent de mettre en évidence les contours des objets mobiles, toutefois dans un environnement urbain la méthode devient plus complexe à utiliser. En effet, le fait de faire une soustraction de fond nécessite d'établir une image initiale qui sera considéré comme fond et qui sera donc soustraite aux autres images à analyser. L'objectif de notre étude étant de pouvoir être utilisé dans tout type l'environnement urbain, le fond à utiliser devra être défini manuellement pour chaque caméra. Cette action requiert alors une phase d'initialisation à prendre en compte, mais qui peut poser plusieurs problèmes :

- Le décalage de la caméra,
- le passage d'objets non humains,
- la modification de l'image de fond par l'environnement.

Dans le cas du décalage de la caméra, ce problème peut survenir dans la mesure où les caméras que l'on utilise, bien qu'elle soit ancrée sur un point fixe, peuvent légèrement être déplacées par les éléments extérieurs. Les intempéries telles que le vent, la pluie et la grêle ou encore des animaux tels que des volatiles peuvent effectuer ce genre de décalage léger, mais important dans la mesure où le fond n'est plus le même que celui utilisé au départ.

Un problème similaire est également observable sans que la caméra ne soit décalée, lors de la modification de la luminosité due au simple cycle jour-nuit d'une journée. Il en est de même pour les intempéries qui, même si elles ne décalent pas la caméra, vont occulter le fond initialement défini et empêcher sa bonne soustraction. Il existe toutefois des méthodes des soustractions de fond robuste qui permettent par exemple de régler tout ou partie de cette problématique [HMT15; OCS11].

Le problème persistant toutefois réside dans le passage d'éléments mobiles dans le champ de vision de la caméra où les éléments en question ne sont pas humains. Étant dans un environnement urbain, tout type d'éléments peuvent s'y déplacer, tels que des voitures, des animaux ou des objets de toutes sortes.

La soustraction de fond, bien qu'elle fasse partie des méthodes les plus communes pour ce genre d'analyse vidéo, n'est qu'une des méthodes permettant de nous aider à atteindre nos objectifs. Il existe en effet une multitude de méthodes qui pourrait nous permettre de faire de la détection vidéo [Yao+20]. Il existe notamment des méthodes permettant d'identifier sur une image des éléments spécifiques sans prendre en considération l'environnement.

Pour ce faire, les méthodes les plus appropriées sont celles consistant à traiter la vidéo au travers d'un système entraîné. Un système entraîné est un système qui, à l'aide d'une base de référence, est capable de reconnaître des objets ou des formes. Un des outils les plus reconnus dans le domaine est openCV [GAM15]. Ce programme de vision par ordinateur (Computer Vision) est capable avec peu de puissance de calcul de détecter des formes et des objets, mais également, s'il y est entraîné, des humains. De plus, des bibliothèques plus légères ont pu voir le jour,



permettant ainsi de l'utiliser sur des micro-ordinateurs tels que des Raspberry Pi par exemple [AK21].

Cette notion de légèreté de calcul n'a pas été mise en avant lors des spécifications du projet, toutefois elle a été considérée de sorte à envisager l'éventualité d'un déploiement sur des systèmes peu performants. Il s'avère toutefois qu'une problématique de puissance de calcul peut vite devenir importante lorsqu'il s'agit de traitements vidéo, car ce genre de traitements peuvent parfois être coûteux [Sme+19].

OpenCV étant une librairie de vision par ordinateur Open Source, de nombreuses autres librairies l'utilisant voient régulièrement le jour. Un des outils basés sur OpenCV que nous aborderons dans cette thèse est YOLO (v3) [RF18]. YOLO est un outil capable de détecter dans une image une multitude d'objets [TS18]. Il s'agit d'un réseau entraîné à reconnaître les objets et disposant d'une communauté d'utilisateur grandissante lui permettant aujourd'hui d'être un des systèmes de détection et d'annotation d'images les plus puissants et les plus documentés. L'engouement autour de ce logiciel a également permis à une version "LITE" de voir le jour qui permet également d'obtenir des résultats sur des micro-ordinateurs [SAP20]. Les travaux effectués dans cette thèse se basent sur la version 3 de YOLO qui était la version la plus utilisée au moment du démarrage du projet. Il existe aujourd'hui une version 7 de YOLO qui permettrait d'améliorer la détection [WBL22].

L'utilisation de ces outils met à notre disposition la possibilité de détecter à tout moment d'une vidéo les différents individus qui y sont présents. Toutefois, ces détections se font sur des images dont la résolution est définie en pixels. Il n'y a alors pas de notion de distance, car le rôle de ces outils n'est que de détecter ce que l'on souhaite leur faire détecter, quelle que soit leur distance par rapport à l'objectif.

Ce constat nous mène alors à nous poser la question de correspondance entre pixels et distance réelle. Étant donné que, quel que soit le système de détection utilisée, il utilisera comme données d'entrée des images issues de caméras de vi-

déosurveillance. Cette contrainte implique donc que l'image sera obtenue à partir d'une caméra dont la distance et l'orientation par rapport à la scène peuvent varier. Par conséquent, la taille des personnes tout comme leur position dans l'image sont donc variables à cause d'un effet de perspective.

Afin de prévenir d'éventuels problèmes liés à ce phénomène, il est possible de disposer de plus d'informations sur l'environnement en cherchant à obtenir au travers de la caméra des informations de distance. Nous pouvons par exemple à partir d'une image et des éléments clés qui la composent connaître l'orientation et la position de la caméra dans l'espace [Xie+09] et ainsi faire une estimation de distance.

Dans tous les cas, quelle que soit la méthode qui sera employée, il est donc important d'avoir connaissance que les éléments observés et identifier dans les différentes images ne seront pas tous à la même échelle. De plus, au vu du nombre de paramètres qui peuvent changer (fréquence d'enregistrement, résolution de l'image, exposition de la caméra...), nous nous sommes demandé comment simplifier l'acquisition de données afin d'obtenir un format exploitable. Le scénario idéal serait une vision du dessus des différentes scènes afin de ne pas être impacté par des effets de déformation dus aux distances.

Ne pouvant toutefois pas disposer d'une scène observée du dessus et au vu du nombre de contraintes grandissantes (d'une part sur l'anonymisation des données et d'autre part sur les différents paramètres faisant que l'espace observé par les caméras de vidéosurveillance ne sera jamais parfait). Il nous a fallu imaginer un format de données suffisamment générique pour nous permettre d'appliquer des filtres en fonction des différents besoins. La stratégie employée dans cette thèse à ce sujet a donc consisté à transformer chacune des images à analyser en nuage de point. En effet, les systèmes de détection sont suffisamment puissants pour détecter des individus à partir d'une caméra, et ce, quelle que soit son orientation [AAA19]. Ainsi en disposant de la position et de la taille qu'occupe un individu dans l'image, nous sommes alors capables d'en extraire un nuage de points. Il est alors envisageable de disposer d'une suite de nuages de points correspondant à la suite d'images qui a été analysée.

Le fait de disposer de nuages de points signifie de fait que nous disposons de données étant, par exemple, visualisable dans un espace orthonormé. Nous ouvrons donc sur cette thèse la possibilité d'analyser de manière statistique les informations issues de vidéos. Cette réflexion nous mène alors à rechercher des méthodes permettant d'exploiter pleinement les informations de ces différents nuages de points.

Nous sommes par exemple capables d'identifier certains éléments clés tels que la formation d'un groupe. Dans ce domaine, il existe plusieurs méthodes permettant d'effectuer des regroupements de points dans une masse de points afin d'identifier les groupes. Parmi elles, nous retrouvons des méthodes comme K-Means ou encore DBSCAN, qui sont des méthodes de regroupement de points présentant chacune leurs avantages et leurs inconvénients [Don+19; ZDL17]. Il est alors possible d'étudier à l'aide de calculs simples les mouvements de foule en se soustrayant de l'image à la différence des méthodes traditionnelles où l'étude du mouvement de foule est portée sur l'évolution globale de l'image [WM10; RML17].

Un des objectifs de cette thèse sera donc de déterminer de quelle manière l'étude des nuages de points permettra de détecter les comportements anormaux. Cet aspect novateur permet de se détacher de la vidéo afin de travailler sur un ensemble de données anonymes. La réussite de ces objectifs amènerait donc deux avantages :

- Les données sur lesquelles les analyses sont effectuées sont complètement anonymes dans la mesure où il est impossible d'identifier un individu à partir de ces données.
- Le système d'analyse comportementale n'est pas dépendant d'un système d'analyse vidéo spécifique. À partir du moment où le système d'analyse vidéo permet l'extraction de données de position nous pourrions alors réaliser les analyses.

Ces avantages impliquent d'une part le respect des consignes imposées par le projet, mais d'autre part une ouverture à d'autres axes de recherches dans la mesure où les données de position que nous utilisons n'ont pas besoin d'être issues de vidéos. En effet, nous pourrions imaginer effectuer nos analyses sur une suite de nuages de points issues de tous autres types d'environnement.

Ces réflexions nous ont menées à nous demander à quel point nous pouvions pousser les analyses de ces différentes séquences de nuages de points. En ayant à disposition une multitude de données de position, il devient alors possible d'effectuer un suivi basé sur des nuages de points. L'ajout d'un suivi à nos méthodes permettrait d'ouvrir les perspectives d'analyse. Les premières recherches sur ce genre de suivi nous ont permis d'envisager une méthode utilisant des filtres de Kalman [Sun+14; QY20].

Les filtres de Kalman permettent notamment de prédire un état futur à partir d'un état passé. Nous pouvons utiliser ce filtre dans plusieurs domaines d'application, comme le domaine maritime par exemple [Hon+19; FF18], afin d'estimer la trajectoire d'un véhicule marin ou même dans le domaine aérien qui montre une forte inertie dans les déplacements [HJW20; SNB18]. De plus, les filtres de Kalman pouvant être utilisés à partir de position et de vitesse, certains travaux proposent une approche dans le domaine aérien, permettant de naviguer en l'absence de GPS ou autre technologie de navigation par satellite [KYK19a]. Nous retrouvons également les filtres de Kalman dans le domaine du traitement de signal où il permet notamment de lisser les signaux ou d'en estimer la forme [GCN07].

Les champs d'application sont vastes et montrent dans la plupart des cas de bons résultats c'est pourquoi nous nous proposons d'étudier une méthode de suivi anonymisé basé sur les filtres de Kalman. Notre objectif serait alors d'utiliser notre séquence de nuages de points afin de prédire l'évolution de ce nuage de sorte à obtenir un suivi individualisé pour chacun des points composant ce nuage.



# BASE DE DONNÉES ET SIMULATEUR

---

## 2.1 Préambule

Le projet SafeCity est à la base d'une grande partie des réflexions de cette thèse. La bonne réalisation de ce projet et des recherches dépendent de la qualité de l'analyse qui sera effectuée sur les caméras de vidéosurveillance. Or pour disposer d'une qualité suffisante d'analyse, il faut également disposer d'informations à analyser de qualité. Cette qualité dépend principalement du réalisme des données et de leurs quantités. Afin de disposer de ces informations de qualité, deux stratégies sont proposées. La première stratégie consiste à rechercher des vidéos proches de ce que nous souhaitons observer afin de nous en servir de base de réflexion. Cette stratégie repose donc uniquement sur le fait d'établir une base de données à partir d'éléments réels. La seconde stratégie mise en place consiste à générer des données réalistes et proches de ce que nous souhaitons observer. Cette stratégie ne repose donc plus sur le besoin d'établir une base de données, mais sur la possibilité d'obtenir autant de données que souhaité à tout moment du temps en les générant. Ce chapitre présente ces 2 stratégies, leurs avantages, leurs inconvénients et les choix qui ont dû être faits à l'issue de leur réalisation.

## 2.2 Base de données

### 2.2.1 Introduction

Une base de données est un élément sur lequel les différentes applications peuvent se baser pour disposer d'informations. Une base de données peut donc prendre plusieurs formes et peut contenir une quantité plus ou moins importante

d'éléments. Dans notre cas, ce qui sera considéré comme une base de données, est la composition globale de toutes les informations issues de la réalité.

### 2.2.2 Le besoin

La première façon de constituer notre base de données consiste à utiliser les caméras de la ville afin d'extraire les informations nécessaires à la détection des différents comportements. Ce point présente toutefois 2 problèmes :

- Dans le cas très spécifique du projet SafeCity, nous avons dû faire face à une restriction due à la confidentialité des images afin de respecter le RGPD. Ce faisant, nous n'avons pas pu travailler avec les images directement issues des caméras de vidéosurveillance de la ville.
- Même si nous avions pu disposer de ces vidéos, il n'est pas certain qu'elles auraient pu nous servir à enrichir notre base de données.

En effet bien que les images auraient été réelles elles n'auraient pas nécessairement montré de comportements dangereux ou suspects. Pour rappel, les cas que nous cherchons à identifier sont :

- La détection de dispersion,
- la détection de rassemblement,
- la détection de maraudage.

Si les vidéos obtenues ne présentent pas ces scénarios, elles n'apporteraient que le seul intérêt de pouvoir valider qu'il n'y ait pas de faux positif dans l'analyse. Il a fallu trouver une solution pour se procurer des images mettant en avant les différents scénarios. Il a d'abord été envisagé d'utiliser les vidéos publiques des caméras de vidéosurveillance disponibles sur internet, mais après quelques recherches, il s'avère que le nombre de caméras disponibles est relativement limité et l'origine du flux vidéo ou même le droit d'exploitation sont contestables. Des sites comme [**Insecam**] par exemple se connectent sur des caméras IP ouvertes sans indiquer qui est l'auteur de la source ou le droit d'utilisation de la vidéo. De plus, la qualité des vidéos est souvent très mauvaise et comme pour les vidéos issues de caméras de vidéosurveillance de la ville de Nice, elles ne présenteront heureusement que très rarement des événements anormaux.

### 2.2.3 Alternative aux vidéos de surveillance

Une autre solution est de se pencher sur l'utilisation des vidéos d'événements sportifs, notamment en ce qui concerne la dispersion et le rassemblement. En effet, des sports tels que du rugby ou du volley-ball peuvent généralement mettre en avant des individus cherchant à se disperser où se rassembler. Cette solution théoriquement viable ne s'est toutefois pas révélée idéale pour plusieurs raisons :

- La plupart des événements sportifs ne sont pas filmés en caméra fixe. Il est donc difficile de se procurer des images qui pourraient être assimilées à de la vidéosurveillance.
- Dans la plupart des cas, l'événement est filmé de trop loin ce qui empêcherait de détecter les différents individus présents sur l'image.
- Les individus qui figurent dans ce genre de vidéo n'adoptent pas un comportement de marche normal.

Le meilleur moyen d'obtenir des vidéos mettant en œuvre l'ensemble des scénarios que nous souhaitons étudier a finalement été de penser nos scénarios et de les tourner dans les mêmes conditions que celles obtenues par les caméras de vidéosurveillance de la ville de Nice.

### 2.2.4 Choix du matériel

Afin de réaliser notre étude, nous avons choisi de nous rapprocher au mieux de l'architecture matérielle déjà en place et de choisir initialement des caméras IP de la marque Axis modèle M1125 (figure 2.1). Les caractéristiques du modèle sont disponibles en annexes A. Cette caméra permet l'obtention de vidéo en ré-



FIGURE 2.1 – Caméra AXIS M1125-E choisi dans le cadre des expérimentations.



solution 1920 par 1080 pixels et avec une fréquence d'image par seconde de 25 à 30. Le format de compression de vidéo étant libre, les formats retenus permettant l'utilisation du flux vidéo en direct sont le H.264 et le MJPEG (Motion Joint Photographic Experts Group) qui sont des standards de compression dans le domaine de la diffusion de flux vidéo [NA18; CSL12]. L'utilisation de ce modèle permet d'être complètement compatible avec les caméras actuellement en place sur lesquelles nous avons peu d'informations. Toutefois, en nous contentant de cette unique voie d'enregistrement, nous n'aurions pu proposer une solution ne répondant qu'à l'architecture du projet déjà en place. Étant donné que SafeCity est avant tout un projet de recherche, la contrainte matérielle n'a pas été totalement définie ni même imposée. Afin de nous permettre une plus grande flexibilité lors des différentes recherches, nous avons donc décidé d'utiliser des caméras plus performantes lors de nos phases d'enregistrement quitte à en dégrader l'image pour nous rapprocher de la qualité des caméras de vidéosurveillance.

L'utilisation de caméras IP amène toutefois une notion d'éloignement physique entre la partie acquisition de la caméra et notre système d'analyse. Cette notion devra être prise en compte pour la suite de l'étude notamment lors de la définition de l'architecture de la solution à mettre en œuvre. En effet, en utilisant une caméra plus performante, mais qui ne diffuse pas de flux vidéo, il faudra intégrer un composant supplémentaire permettant d'émuler une caméra IP ayant un comportement en tout point identique avec les paramètres cités précédemment.

### 2.2.5 Choix des scénarios

En parallèle de l'identification du matériel que nous allons utiliser, nous avons réfléchi aux différents scénarios pouvant être tournés. L'objectif est d'enregistrer plusieurs vidéos mettant en œuvre différents scénarios. Plusieurs scénarios ont donc été identifiés afin de disposer d'un ensemble de cas nominaux qui pourront servir de base d'études :

- Dispersion rapide,
- dispersion lente,
- dispersion progressive,
- rassemblement hétérogène,
- rassemblement progressif,

- maraudages

D'autres scénarios plus spécifiques ont également été identifiés afin de couvrir des cas plus particuliers pouvant se produire :

- dispersion rapide, avec tout les individus partant dans le même sens,
- dispersion lente avec tous les individus partant dans le même sens,
- dispersion lente en plusieurs petits groupes,
- rassemblement progressif suivi d'une dispersion rapide,
- rassemblement hétérogène suivi d'une dispersion rapide,
- maraudages puis rassemblement lent,
- maraudage et dispersion rapide,
- maraudage et assaillant provoquant une dispersion avec tout les individus partant dans le même sens,
- maraudage et dispersion rapide,
- arrivée d'un individu et départ simultané d'un autre,
- un individu seul quitte un groupe rapidement,
- un individu rejoint un groupe lentement,
- un assaillant arrivant en dehors du champ de vision de la caméra provoque une dispersion rapide avec tous les individus partant dans le même sens,

### 2.2.6 Contrainte de l'acquisition vidéo

La vidéo étant au cœur de la base de données, il a d'abord fallu définir les contraintes imposées par son utilisation. L'étude porte sur l'analyse comportementale par l'utilisation de caméra de vidéosurveillance. L'étude met donc plusieurs contraintes en avant :

- Les caméras doivent être d'une résolution suffisante pour pouvoir distinguer un individu.
- Les caméras doivent enregistrer un nombre d'images par seconde suffisant afin que l'analyse dispose de suffisamment d'images traitées.
- Les caméras doivent être à une distance suffisamment proche de la scène pour pouvoir distinguer un individu.
- L'angle de la caméra doit permettre de distinguer la scène dans son ensemble en évitant que les personnes au premier plan n'obstruent trop celles se trouvant derrière.
- Les vidéos devant servir à la validation des solutions et à la recherche, elles

devront être réalistes et disponibles en quantité suffisante.

- Les caméras utilisées devront permettre, à terme, une diffusion du flux vidéo en H.264 et Motion JPEG.

La résolution des caméras est un élément fondamental dans ce contexte d'utilisation. Les outils utilisés pour l'exploitation des données sont majoritairement entraînés à détecter des formes ou des objets à partir de bases de données de référence. Bien que ces bases de données comportent des éléments de qualité plus ou moins bonne, il est préférable que la résolution soit la plus haute possible. En effet, c'est en disposant d'une résolution la plus haute possible que les détections seront les plus fiables. Pour répondre à notre besoin, nous avons décidé d'utiliser la résolution actuellement standard des caméras de vidéosurveillance à savoir 1920 par 1080 pixels. Nous remarquerons toutefois que certaines caméras de vidéosurveillance peuvent disposer de résolution beaucoup plus importante comme 3840 par 2160 pixels. L'étude ne portant pas sur ce genre de caméra cette résolution n'a donc pas été retenue.

Aujourd'hui, il existe de nombreuses caméras de sécurité disposant de fréquence d'enregistrement d'image très variable. Cette fréquence reste toutefois généralement aux alentours de 30 images par seconde. Cette valeur évolue en fonction du besoin : en dessous de 24 l'œil humain commence à ressentir les saccades dans la vidéo et au-dessus de 24 la vidéo donnera une impression de plus grande fluidité [PK21]. Toutefois, le nombre d'images par seconde impactera surtout la quantité de données à transmettre entre la caméra et l'affichage de la vidéo. Bien que certaines caméras de vidéosurveillance puissent enregistrer à plus de 60 images par seconde, la ville de Nice n'utilise pas de ce genre de dispositif. En effet, les villes équipées de caméras de vidéosurveillance ont généralement un panel de caméras dont il faut ramener les flux dans un même centre de traitement.

Afin de nous donner une idée des limites de résolution et de fréquence d'enregistrement des caméras que nous pouvons nous permettre d'utiliser, nous pouvons nous baser sur un échantillon de vidéo au format MP4 : en considérant qu'une vidéo MP4 de 115 secondes, qui a une résolution de 1920 par 1080 pixels et une fréquence d'enregistrement de 30 images par seconde pèse environ 58 000 000 octets soit 55,3Mo. Cela signifie qu'un flux vidéo de même qualité nécessitera

un débit d'environ 0,5Mo par seconde pour s'afficher. Sur une connexion fibre optique classique limitée à 50Mo par seconde (il est difficile d'estimer le débit maximum d'une connexion fibre optique étant donné qu'elle dépend du fournisseur d'accès, nous fixerons donc une limite raisonnable par rapport aux capacités maximales d'une fibre optique [Ess+10]), nous pouvons dans un cas optimal recevoir le flux de 100 caméras en simultané. En enregistrant à 60 images par seconde, le nombre de caméras est divisé par 2 et en enregistrant en 3840 par 2160 pixels ce nombre est divisé par 4.

Afin de rester dans un contexte réaliste pour mener les recherches, il a été décidé d'utiliser une fréquence d'image de 25 à 30 images par seconde. L'étude nous montrera que ce choix n'est pas anodin, car bien qu'il n'est pas le plus élevé par rapport à ce qu'il était possible de sélectionner, il reste le choix permettant le plus de polyvalence et de flexibilité lors des analyses. La caméra cible devant être une caméra de surveillance, nous avons fait le choix d'utiliser des caméras à une hauteur supérieure à 3 mètres du sol. Cette hauteur permet notamment de se rapprocher de la réalité terrain où les caméras de la ville se trouvent sur des mâts ou hors de portée des passants.

L'environnement dans lequel seront appliqués les différents algorithmes sera à terme une place publique ouverte. La caméra filmant la scène forme un angle  $\theta$  avec le sol comme montré en figure 2.2. Cet angle  $\theta$  est primordial :

- Si  $\theta$  est trop faible, de  $0^\circ$  à  $30^\circ$ , la profondeur de champ sera trop importante et les détections s'appliqueront principalement sur les personnes les plus visibles et seulement parfois sur des personnes lointaines. Ce phénomène provoquerait l'apparition de détection intempestive pouvant générer de mauvaises alertes.
- Si  $\theta$  est trop important, de  $60^\circ$  à  $90^\circ$ , les détections risquent de ne pas se faire, car les algorithmes sont entraînés à détecter des personnes relativement de face et risquent de ne pas identifier les individus s'ils sont vus de dessus. L'autre problème pouvant survenir est le manque de profondeur de champ passé un certain angle. La profondeur de champ devient alors insuffisante et ne couvre pas une zone suffisamment importante pour rendre l'analyse utile.

L'angle choisi afin de réaliser cette étude doit s'approcher de celui utilisé par les caméras de surveillance fixe. Étant donné qu'il n'y a pas de réel standard, nous choisirons de travailler avec un angle  $\theta$  pouvant varier de  $30^\circ$  à  $60^\circ$  par rapport au sol.

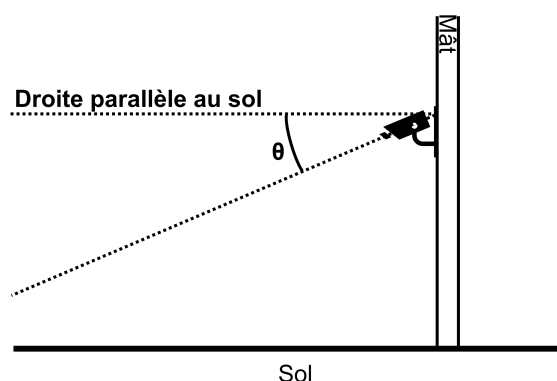


FIGURE 2.2 – Définition de l'angle  $\theta$  formé entre l'orientation du sol et de la caméra.

## 2.2.7 L'extraction d'informations

Initialement, le projet SafeCity devait reposer sur la présence de caméras dans des espaces publics (grands ou petits), mais également dans des lieux accueillant du public, dont l'angle de vue permettait d'identifier sans difficulté la présence de certains objets dans l'image, ou encore le visage des passants (c.f. 1.1.1). Nous avons dans un premier temps imaginé le type d'analyse envisageable dans ce contexte. Dans le domaine de l'analyse vidéo, plusieurs axes d'étude sont possibles. L'un des premiers axes consiste à tenter d'identifier sur une vidéo des objets particuliers tels que des sacs ou des armes. La détection d'un sac permet par exemple d'identifier si ce dernier était porté ou posé au sol et donc potentiellement abandonné. La détection de la présence d'une arme permet d'identifier une personne potentiellement dangereuse comme peut le montrer la figure 2.3.

Un autre axe consiste à identifier le visage d'une personne et étudier son comportement. Nous pouvons par exemple compter le temps de présence d'une personne dans un lieu ou encore le nombre de fois où elle y fait une apparition. La

première recherche d'outils existants permettant d'identifier des personnes et des objets en a mis en avant deux : OpenCV et YOLOv3[DZ21 ; Cha+18].

OpenCV est avant tout un outil de traitement d'image permettant d'effectuer des opérations de lecture et d'écriture sur ces dernières. Il permet également de faire de la détection de visage humain en utilisant les cascades de Haar[VJ04 ; IT17], ainsi que d'autres types de détections plus simples tels que des formes géométriques[KB]. OpenCV permet également de faire des traitements d'images nécessaires à d'autres outils utilisant des intelligences artificielles afin de reconnaître des objets[CB10 ; Far+19 ; KB]. Une utilisation courante d'OpenCV consiste à calculer l'histogramme orienté de gradient (HOG) d'une image contenant un élément spécifique[SAT11]. Cette technique permet de passer d'une image où chaque pixel la composant possède une information sur sa quantité de rouge, vert et bleu à un découpage par zone où chaque zone de l'image est représentée par 2 valeurs : un angle et une amplitude permettant de décrire un vecteur.

Quant à YOLOv3, il s'agit d'un réseau neuronal qui permet l'identification dans une image de certains éléments. Il permet entre autres la détection d'individus[TS18]. Son fonctionnement de base a pour but d'identifier dans une image les objets qu'il connaît en générant les coordonnées d'un rectangle les encadrant. YOLOv3 peut également être utilisé sur une vidéo, il faudra alors récupérer chacun des rectangles générés par YOLOv3 pour chacune des images de la vidéo.

### 2.2.8 Expérimentation sur l'extraction d'information

L'objectif derrière l'utilisation de ces outils est de pouvoir extraire des informations de position sur les objets et les personnes présentes dans l'image. YOLOv3 étant basé sur la technologie de réseau neuronal, il ne pourra identifier que les éléments pour lesquels il a été entraîné. Nous pouvons par exemple retrouver l'utilisation de YOLOv3 dans le domaine médical[Hos+21] comme dans le domaine de l'urbanisme ou de l'agriculture[Li+20]. Plusieurs réseaux existent et il est également possible de créer son propre réseau afin de faire apprendre des objets spécifiques. Pour ce faire, nous devons disposer d'une quantité importante d'images présentant les objets que nous souhaitons faire apprendre à YOLOv3[War+19]. Une fois ces images en notre possession, il faut les détourner et labelliser tous les

objets que nous souhaitons faire apprendre à YOLOv3. L'avantage de cet apprentissage est de nous permettre de faire apprendre au réseau ce qu'est par exemple une arme à feu, un couteau ou même un sac. Une première phase d'expérimentation a été menée afin de proposer une détection d'objets potentiellement dangereux. Nous avons préparé un ensemble d'environ 300 images de couteaux labellisé à l'aide de l'outil BBox-Label-Tool [Qiu14] afin d'apprendre au réseau ce que nous souhaitons détecter. Au plus le nombre d'images préparé est important et au plus les résultats seront pertinents et nous pourrons détecter l'objet appris comme le montre la figure 2.3. En revanche si le nombre d'images utilisé n'est pas suffisant ou si le réseau est aussi entraîné pour d'autres objets, il peut arriver que la détection ne soit pas certaine comme le montre la figure 2.4.

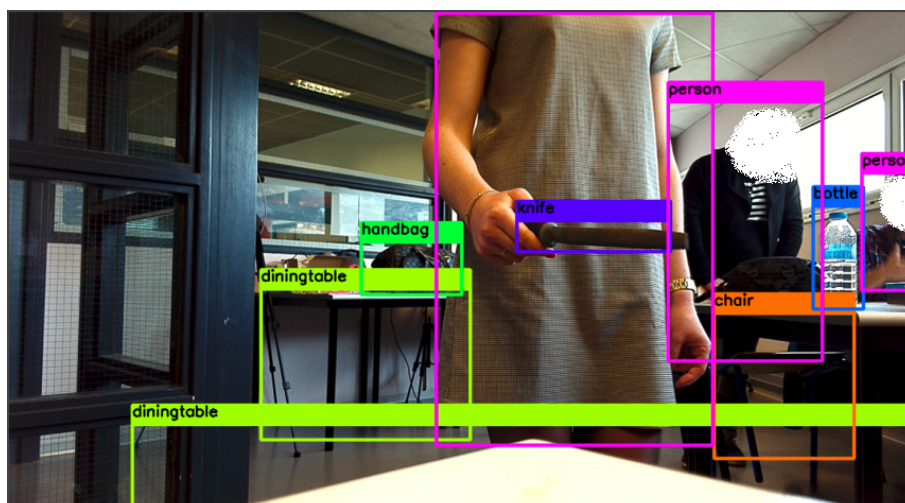


FIGURE 2.3 – Un ensemble de détection réalisé à l'aide de YOLOv3 sur l'ensemble d'une image issue d'une vidéo.

Si l'objectif est de détecter un objet unique, l'utilisation de 300 images pourra suffire. Certaines détections pourront ne pas avoir un taux de certitude élevé, mais nous pourrons choisir le seuil à partir duquel la détection aura bien lieu. Il restera préférable d'utiliser plus d'images pour améliorer la certitude des détections. Malgré tout, les premiers résultats ont permis de montrer que l'apprentissage, bien que rébarbatif, était réalisable et permettrait avec suffisamment d'entraînement de détecter tout type d'objets.

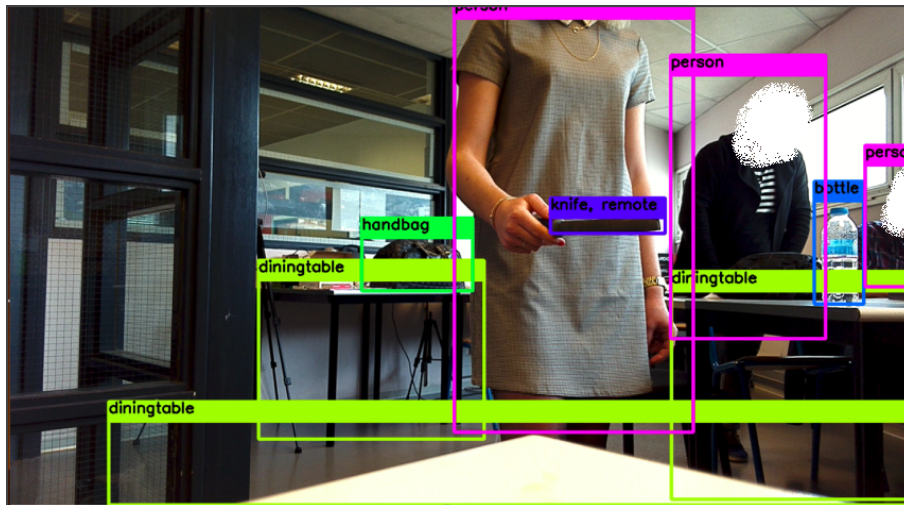


FIGURE 2.4 – Un ensemble de détection réalisé à l’aide de YOLOv3 sur l’ensemble d’une image issue d’une vidéo présentant une incertitude sur le type d’objet détecter (couteau ou télécommande).

## 2.2.9 Conclusions des expérimentations

Bien que ces premiers résultats étaient encourageants, ils restent problématiques. En effet, lors des différentes phases de réflexions autour du projet SafeCity, il est apparu que l’utilisation de caméras permettant l’identification d’objets ou de visage peut être problématique pour des raisons de confidentialité. Le projet SafeCity et notamment la partie concernant l’utilisation des caméras ont donc été revus par le porteur du projet. Nous n’avons dès lors plus la possibilité d’utiliser l’identification d’objet, car les seules caméras encore à notre disposition étaient celles de la ville. Après avoir testé YOLOv3 sur ces caméras, il s’avère qu’elles se trouvent alors trop éloigner de la scène pour pouvoir identifier clairement les objets souhaités.

Les travaux menés sur la détection d’objet ont tout de même pu être réutilisés pour la détection d’individu. En effet, pour obtenir les données de vérité terrain, il a fallu extraire les informations de position des personnes présentes dans les vidéos afin de pouvoir les envoyer à l’analyseur. YOLOv3 permet de détecter la présence d’un individu dans l’image même si cette dernière est à plusieurs mètres de la caméra. En faisant cette analyse sur un maximum d’images composant une vidéo nous pouvons donc obtenir la position de tous les individus présents dans



ces images. YOLOv3 a pour vocation de labelliser les images en y ajoutant les informations de détection, mais en modifiant un peu le code, il est alors tout à fait possible d’obtenir les informations de position des différentes personnes et de les envoyer sur le réseau. Le fait de procéder de cette manière nous permet deux choses :

- la première est que le format de données peut être réfléchi et encapsuler d’une certaine manière afin d’être standardisé entre la sortie de YOLOv3 et celle du simulateur
- la seconde permet une anonymisation des données, car aucune information autre que les positions et zones occupées par les personnes présentes dans l’image ne circule sur le réseau.

L’anonymisation est un point clé du projet. Bien que le système d’analyse se trouvera sur un réseau sécurisé, le fait de ne disposer que des informations de position permet d’éviter de diffuser des informations sensibles. Nous évitons ainsi de faire transiter sur le réseau le visage d’une personne par exemple.

### 2.2.10 Estimation de l’environnement

L’estimation de l’environnement est dans la littérature un moyen d’utiliser les informations disponibles dans une image comme points de repère pour établir une base de réflexion. Dans notre contexte, il peut être intéressant de disposer d’informations sur l’environnement observé. Le fait de pouvoir connaître par exemple les distances entre les différents éléments de la vidéo permettrait d’avoir une meilleure idée des vitesses et des distances entre les individus.

Un des axes de recherche menée a consisté à essayer d’établir, à partir des données issues de la caméra, quels éléments pouvaient révéler des informations sur la disposition du terrain. L’idée initiale était de considérer que la scène à analyser pouvait être transposable en deux dimensions. Cette transposition peut également être faite en vues de dessus si nous sommes capables de connaître les distances réelles à différentes profondeurs de champ.

Une telle transposition est théoriquement faisable, mais devient difficilement réalisable dès lors que la scène observée se complexifie : différents espaces à des

hauteurs différentes ou encore des surfaces irrégulières. L'objectif de cette méthode est de fixer un ou plusieurs points de référence et de déterminer la distance entre ces derniers. Il existe plusieurs méthodes qui permettent de définir l'environnement, dans notre cas, deux d'entre elles ont fait l'objet d'une étude plus approfondie :

- L'utilisation de caméras spécifiques à cet usage comme la stéréovision,
- Utilisation d'éléments de référence dans l'environnement observable de l'image pouvant permettre de points de repère .

Dans le cas de la stéréovision, l'objectif est d'utiliser des caméras spécifiques dotées de plusieurs objectifs afin d'ajouter une notion de perspective à la scène au travers d'une fusion des deux images obtenues. Cette méthode permet notamment d'améliorer l'appréhension de l'environnement en robotique[MJ97] ou plus généralement de contribuer à l'appréhension d'un environnement 3D au travers de caméras[Ned+07]. Cet usage nous permettrait notamment d'estimer les différentes distances de la scène observée directement à partir du point de vue de la caméra.

En ce qui concerne l'utilisation d'éléments de référence, l'objectif de cette méthode est d'utiliser dans le champ de vision de la caméra un élément dont les dimensions sont connues. Cet élément peut prendre plusieurs formes, l'objectif étant de pouvoir mesurer cet élément en pixel afin de faire correspondre sa dimension réelle connue à sa dimension en pixels observés. Cette méthode permet de créer un système de caméra calibré à son environnement[JKH08].

### 2.2.11 Expérimentations sur les mires

Bien que l'idée de l'utilisation de caméras stéréovision semblait être intéressante dans notre cas d'utilisation, les contraintes projet ne nous permettent pas de modifier les caméras déjà en place. La solution finalement retenue est celle impliquant l'utilisation d'éléments de référence. Une première série d'expérimentations a été menée afin de déterminer si à l'aide de mires dans l'image il était envisageable d'obtenir une estimation des distances. La figure 2.5 montre les premiers résultats obtenus dans le cadre d'une détection de mires circulaires orange.

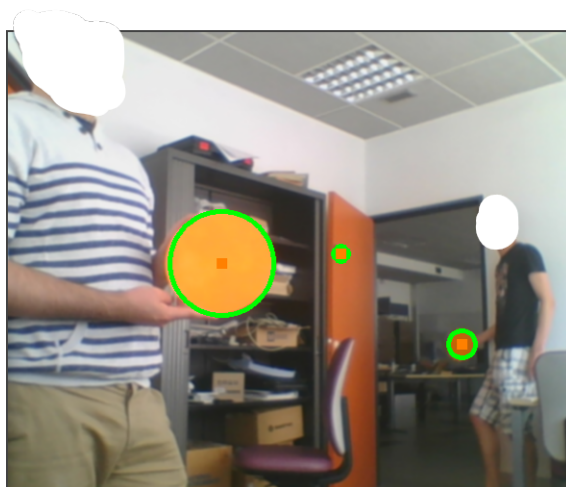


FIGURE 2.5 – Détection de mires à différentes distances de la caméra.

Cette expérimentation a été effectuée grâce à l'utilisation de la bibliothèque OpenCV présentée précédemment. Les expérimentations ont mis en avant que la détection de mires était possible comme le montre la figure 2.5. Toutefois, plusieurs éléments problématiques ont pu être observés :

- L'utilisation des mires à une distance élevée se révèle peu efficace, car la taille de ces dernières en pixels devient trop faible pour être significative.
- La simplicité de conception des mires peut révéler de mauvaise détection comme le montre la figure 2.5. En effet, les mires détectées par OpenCV sont entourées en vert, nous comptons donc trois détections dans l'image, or il n'y a que deux mires.
- La détection des mires n'est pas parfaite.
- Certaines caméras de surveillance telles que la AXIS M1065-L W (figure 2.6) par exemple peuvent présenter des lentilles particulières permettant un plus grand champ de vision.

En ce qui concerne la problématique d'utilisation de mires à distance et de leurs simplicités, la solution la plus simple a été d'agrandir les mires afin d'essayer d'augmenter la distance de détection et de complexifier la mire en ajoutant un autre cercle de couleur verte à l'extérieur de la mire comme le montre la figure 2.7. La problématique la plus préoccupante est le manque de précision dans la détection comme le montre la figure 2.8, il existe une distance (2.8.a) entre le cercle de détection de OpenCV et la taille réelle de la mire en pixels. L'apparition de ce



FIGURE 2.6 – Caméra secondaire AXIS M1065-L W.



FIGURE 2.7 – Mires complexifiées avec l'ajout d'une couleur verte autour de l'orange déjà existante.

décalage est relativement aléatoire, mais est toutefois liée à l'orientation de la cible par rapport à la caméra qui doit lui faire parfaitement face pour que le cercle de détection soit le plus précis possible. La seconde problématique la plus préoccu-

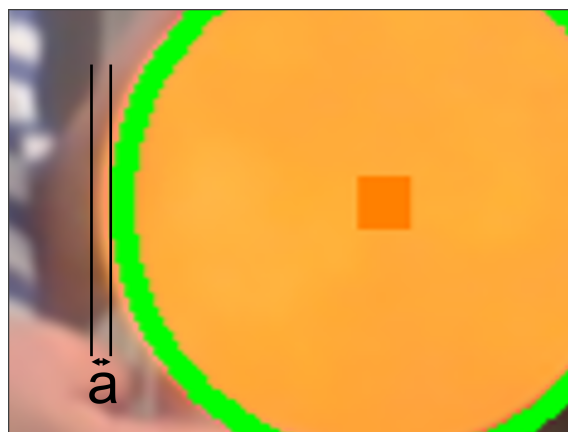


FIGURE 2.8 – Observation du décalage dans la détection de mire.  $a$  : Distance observable entre le bord de la détection de OpenCV et le bord réel de la mire.

pante  $a$  a été découverte lors d'expérimentation avec la caméra secondaire (figure 2.6). La caméra est équipée d'une lentille permettant un plus grand champ de vision. L'utilisation d'une lentille permettant un plus grand champ de vision permet notamment de voir plus de choses dans l'image, mais va provoquer une distorsion de cette dernière [Buq+21]. Or en combinant ces deux dernières problématiques, il s'avère que la calibration de la caméra par mire dépend de 2 nouveaux paramètres :

- La précision de l'orientation de la mire par rapport à la zone d'observation de la caméra.
- L'utilisation d'une caméra avec un champ de vision ne déformant pas l'image.

Ne parvenant pas à établir de résultats cohérents entre la mesure logicielle et les mesures réelles, aucune solution suffisamment développée n'a pu aboutir. Toutefois en reprenant les réflexions à leur base nous sommes aperçus que bien qu'il était intéressant de savoir quelles étaient les distances approximatives de la scène observée, elle n'était pas forcément nécessaire dans le cadre de l'étude. Nous verrons dans la suite de ce chapitre comment l'extraction des données a eu lieu et de cette manière comment nous pouvons faire abstraction des mesures réelles au travers de l'utilisation des mesures relatives en pixels.

### 2.2.12 L'anonymisation des données

La conversion des données doit être simple et facilement réutilisable c'est pourquoi les données issues soit de la simulation soit de la réalité ont le même format JSON. Ce format permet de mettre facilement en avant les informations essentielles. Dans notre cas, il s'agira des informations concernant la zone dans laquelle une personne apparaît dans l'image. En ayant ces informations, nous sommes alors capables de virtualiser l'environnement d'étude et de nous en servir de base pour l'analyse. Une trame type aura donc le format suivant :

```
1 {
2     timecode: int,
3     individuals: [
4         { x: int, y: int, width: int, height: int },
5         ...
6     ]
7 }
```

L'avantage de ce format est qu'il est évolutif et facilement compréhensible. Dans notre cas d'utilisation, il est régulièrement apparu qu'il fallait pouvoir modifier ou supprimer certaines valeurs aberrantes. Ces modifications pouvaient alors être faites avec un simple éditeur de texte par exemple. L'évolutivité a montré son avantage surtout pendant la phase de recherche et développement où il a régulièrement fallu tester de nouveaux paramètres. En effet, étant donné que la problématique sous-jacente reste le suivi, certaines pistes sur des paramètres de forme ont été envisagées et ont donc été testées. Le format de données proposé se compose, par trame, de 2 éléments : un Timecode et un tableau.

Le Timecode sert de repère temporel et permet notamment de savoir si la donnée qui vient d'être reçue est bien temporellement après la dernière donnée traitée. Le tableau contient un ensemble de quadruplets de valeur. Il y a un nombre de quadruplets égal à celui du nombre de personnes détecté dans l'image. Chaque quadruplet contient : la position en abscisse de la personne ( $x$  : un entier, car il s'agit de la coordonner en pixels à partir du coin haut gauche de l'image), la position en ordonnée de la personne ( $y$  : un entier pour les mêmes raisons que l'abscisse), la hauteur du rectangle de détection dans laquelle se trouve la personne

(height : un entier, car la valeur est en pixel) et la largeur du rectangle de détection dans laquelle se trouve la personne (width : un entier, pour les mêmes raisons que la hauteur).

Cet encapsulage de données permettrait à terme d'ajouter de nouveaux paramètres pour chaque individu. Nous avons imaginé pouvoir ajouter les données liées à des repères graphiques des individus identifiés. Certaines méthodes de suivi reposent sur la correspondance physique d'un même élément entre 2 images[Hus+06]. Le principe de cette méthode repose sur l'identification d'un paramètre de forme tel que la couleur de l'objet que nous souhaitons traquer ou la forme de ce dernier . Une fois cette forme identifiée dans la première image , l'algorithme utilisé va chercher la zone de l'image s'approchant le plus de la forme recherchée.

Nous pensons que l'ajout d'un paramètre de forme lors de l'identification des individus est un élément permettant de préserver l'anonymat de la personne (si ce dernier est basé sur une valeur caractérisant une forme ou une couleur) et pourrait ainsi servir de référence dans un système de suivi individualisé. Nous aborderons plus précisément ce point dans le chapitre 4.

## 2.3 Simulateur

### 2.3.1 Le besoin

L'extraction d'informations issues de vidéos est au centre des recherches effectuées dans cette thèse. L'intérêt de ces vidéos est de pouvoir alimenter l'analyseur en données et par conséquent nous permettre de valider que les détections correspondent à ce qui était attendu. Toutefois, même en disposant de ces vidéos, il est impossible de couvrir tous les cas de comportements anormaux qui peuvent se produire dans une situation réelle. Cette problématique peut être résolue de deux façons :

- En tournant des vidéos supplémentaires qui nécessiteraient un temps de tournage considérable et ne permettraient que d'augmenter le nombre de cas couverts.

- En disposant d'un système capable d'alimenter l'analyseur avec des données générées tel que pourrait le faire un simulateur.

La mise en place d'un simulateur permettrait de ne plus être exclusivement dépendant de vidéos afin de vérifier le bon fonctionnement de l'analyseur. En revanche, le fait de simuler des éléments se rapprochant de la réalité implique que ce dernier soit robuste et réaliste. De plus, afin de s'assurer de pouvoir couvrir d'éventuels comportements tiers qui n'auraient pas été imaginés, il faudra s'assurer que ce simulateur soit évolutif dans son architecture.

L'étude des 3 scénarios présenter en 2.2.2 met en avant un certain nombre de paramètres communs. Ces paramètres ne peuvent être utilisés que s'ils sont directement obtenus à partir d'une vidéo. Étant donné que les recherches se basent sur des caméras de surveillance de jour, il faut que les paramètres soient exploitables à partir de caméras filmant dans le domaine visible. Les paramètres exploitables qui ont été retenus sont :

- La présence d'individus dans l'image. Ce paramètre, pouvant sembler évident, met toutefois en avant le fait que le simulateur puisse parfois ne pas montrer d'individus. De manière générale, dans une scène, il peut arriver que personne ne se trouve sur l'image pendant un certain temps.
- La notion de position. Dès qu'un individu se trouve dans l'image, il est important de connaître sa position dans l'image et sa position par rapport aux autres individus.
- La distance entre chaque individu. Le fait de connaître la position de chaque personne dans l'image permet également d'obtenir les positions relatives entre chaque individu.

Avec ces 3 paramètres, il est possible de réfléchir à l'ensemble des solutions à mettre en œuvre pour la résolution des 3 scénarios. D'autres paramètres peuvent aussi avoir leur intérêt comme la vitesse d'un individu, sa trajectoire, sa taille, des paramètres de forme (comme la couleur des vêtements par exemple), etc.

Il a été expliqué précédemment que les vidéos mettant en œuvre les différents scénarios ont été tournées afin de proposer une vérité terrain dans les analyses. Les réflexions menées autour du simulateur ont permis de déterminer que ce der-



nier devait être capable de générer des scénarios à la volée. La génération de ces scénarios doit pouvoir être effectuée de plusieurs manières :

- Le simulateur doit d’abord être capable de compléter les scénarios qui ont été tournés en vidéo.
- Le simulateur doit être capable de générer ses propres scénarios intégralement et de les transmettre comme si les données étaient extraites d’une caméra de vidéosurveillance.

L’intérêt de compléter des scénarios déjà tournés en vidéo est qu’ils peuvent ainsi être modifiés. Nous obtenons alors un scénario initialement réel où la présence de comportements anormaux peut être modifiée en fonction du besoin. Nous pouvons ainsi avec un minimum de scénarios réels obtenir une multitude de scénarios dérivée présentant plus ou moins d’anomalies.

En ce qui concerne la possibilité de générer l’intégralité d’un scénario, l’intérêt réside dans la possibilité de création de scénarios à la demande. Le fait de pouvoir disposer de scénarios entièrement fabriqués permet notamment de tester les limites de l’analyse. Nous pouvons par exemple générer des scénarios extrêmes avec de nombreuses anomalies ou a contrario en présentant très peu. Il est également possible de générer des scénarios réalistes sur une durée beaucoup plus longue que celle tournée en vidéo. Nous disposerions alors d’un système ayant une capacité de génération continu. Cette notion est importante étant donné que l’objectif initial est l’utilisation d’un système d’analyse sur des caméras de vidéosurveillance. Les caméras de vidéosurveillance de la ville étant capables de filmer en continu, un tel système permettrait donc de s’approcher du même mode de fonctionnement. La base de ce simulateur se compose donc de 2 modules : le premier permet de générer des trajectoires et le second de les envoyer au système d’analyse.

En ce qui concerne le premier module, la génération de trajectoires peut se faire de plusieurs manières. Le simulateur doit être capable de générer de nouveaux marcheurs se déplaçant dans un environnement 2D. La notion de hauteur/déni-velé n’étant pas abordée au vu du contexte d’études, nous nous sommes orientés sur des solutions de génération en environnement 2D, comme si la scène était observée de dessus. Une phase de recherche importante a été menée sur la ma-

nière de générer de nouveaux marcheurs et afin de les intégrer, le simulateur a également été réfléchi et conçu de manière à pouvoir utiliser plusieurs solutions permettant la génération. Le simulateur a donc connu plusieurs versions utilisant parfois plusieurs de ces solutions de génération seules ou en combinant certaines. Parmi les principales méthodes utilisées, nous retrouvons :

- Le rejeu. Le simulateur doit être capable de rejouer les données issues d'une vidéo dans les mêmes conditions que si les données avaient été directement extraites des vidéos.
- Le suivi de chemin. C'est une des premières méthodes qui a été imaginée. Son objectif est qu'au travers de l'utilisation de points de passage, un utilisateur puisse créer un chemin qui sera suivi par un marcheur simulé. Cette méthode est notamment utilisée pour le contrôle de véhicule autonome afin de les faire passer par des points de passage bien spécifiques [Yua+06; Yan+16]. Le chemin est généré au travers d'une séquence de points que l'utilisateur aura définie. La complexité et la forme du chemin peuvent être définies par différents algorithmes.
- La génération autonome. Cette méthode consiste à générer des individus sans qu'ils soient issus de vidéos ou sans qu'ils soient préalablement paramétrés par un utilisateur. Cette méthode est de loin la plus complexe des trois, car elle implique que le simulateur soit autonome dans sa manière de générer les informations.

Le second module quant à lui est un simple module de communication. L'objectif de ce dernier est de pouvoir faire transiter les informations du simulateur à l'analyste. Pour ce faire, il utilisera le format de données JSON présenté précédemment, il les enverra sous forme d'une chaîne de caractères à l'analyste qui utilise le même format de données. Bien que ce module ne présente que peu d'intérêt scientifique, il est important de souligner qu'il a tout de même été pensé pour pouvoir être facilement modifié. L'objectif étant de pouvoir répondre aux besoins du projet sans ajouter de nouvelles contraintes dans les phases de recherche. En effet, l'objectif du simulateur est de se rapprocher du fonctionnement d'une caméra dont les informations importantes auraient été extraites et envoyées sur le réseau. Or, le fait de faire transiter les données sur le réseau ralentit le processus d'analyse ce qui, dans le cadre de la recherche, est une perte de temps que nous

pouvons éviter en remplaçant ce module par un système de partage de mémoire.

En ce qui concerne l’environnement de développement du simulateur, le choix a été fait en concordance avec les décisions qui ont été prises pour l’analyseur, sa réalisation se fera donc en Python3. L’utilisation de ce langage le rend compatible avec la plupart des systèmes informatiques et permet également de faciliter la conversion des données, car même si le format de données choisi est utilisable par de nombreux langages de programmation le fait d’utiliser un même langage rend l’exploitation des données beaucoup plus accessibles. Le système d’exploitation utilisé est Windows 11, toutefois le lancement du programme se fera à l’aide de WSL2 sur un sous-système Ubuntu 20.04. Les caractéristiques matérielles principales sont un Processeur Intel I7 de 6 cœurs physiques cadencé à 2,60 GHz, une mémoire RAM de 16 Go et d’une carte graphique NVIDIA RTX 3070.

### 2.3.2 Le Rejeu

Dans le cadre du simulateur, le rejeu prend une place essentielle. L’objectif d’un rejeu est d’offrir la possibilité de rejouer une séquence réelle dans les mêmes conditions que lors de sa première acquisition. Dans notre cas, le rejeu offrira notamment la possibilité de tester les différents algorithmes d’analyse sur la même base de données. Le rejeu peut être effectué de plusieurs manières :

- Le premier moyen étant de simplement rejouer la vidéo en simulant le flux vidéo et en envoyant ce dernier sur le système d’extraction d’informations tel que YOLOv3. Cette solution implique toutefois de devoir utiliser une simulation de caméra IP et une analyse de flux vidéo.
- Le second moyen consiste à ne conserver du rejeu que l’extraction d’informations et de rejouer cette extraction d’informations afin de l’envoyer à l’analyseur. Étant donné que le format de données issues de la partie extraction d’informations vidéo est le même que celui issu du simulateur, il n’est pas nécessaire de rejouer la vidéo.

Dans un premier temps, le module d’extraction d’information a été adapté de sorte à ne plus envoyer directement les données sur le réseau, mais à les stocker dans des fichiers sur la machine. Pour les besoins du projet, la fonctionnalité d’envoi sur le réseau a bien entendu été conservée, mais désactivée dans le cadre de

la recherche. Les fichiers ainsi créés disposent de deux informations fondamentales : l'ensemble des informations extraites de la vidéo et un Timecode permettant d'identifier à quel moment de la vidéo ont été détectées ces informations. L'intérêt du Timecode est de pouvoir déclencher l'envoi des informations au même rythme que leurs détections dans la réalité. C'est un point important du rejeu, il faut s'assurer que les informations sont envoyées à la même vitesse que lors du scénario initial réel ou simulé.

Cette solution de rejeu permet donc de disposer d'une simulation dans les mêmes conditions que la réalité. Un autre avantage de cette solution est le fait de pouvoir analyser des vidéos de manière plus précise. En effet, l'extraction d'informations peut prendre un certain temps à s'exécuter. Ce temps dépend essentiellement de la puissance de calcul utilisée afin d'extraire les informations dans l'image. La vitesse d'extraction des données ne fait pas partie de l'étude toutefois, cette vitesse d'extraction dépend de nombreux critères tels que la résolution de l'image, la puissance de calcul disponible ou encore la finesse de détection préalablement paramétrée. Le système mis en place dans le cadre du projet permet l'extraction de données de 20 images par seconde parmi les 25 à 30 disponibles dans le flux vidéo.

En revanche, dans le cadre de la recherche, rien ne nous empêche d'extraire les informations de la totalité des images disponibles dans la vidéo. Dans ce cas, le rejeu présente un autre avantage, car même s'il est impossible en temps réel de traiter plus de 20 images par seconde, il est possible de faire un traitement de toutes les images et de les stocker dans le fichier. Dans ce cas, il suffira de renseigner comme Timecode non pas par le temps de traitement, mais par le temps de l'image dans la vidéo. Le fait d'avoir ajouté cette notion de temps au simulateur nous permet de rejouer dans des conditions optimales, normales ou dégradées en simulant une extraction d'information plus ou moins rapide.

### 2.3.3 Le Suivi de chemin

Afin de compléter le rejeu, il a fallu trouver des solutions permettant d'ajouter des marcheurs virtuels au scénario réel. La première solution imaginée était d'offrir la possibilité à l'utilisateur du simulateur de créer ses propres chemins

qui seront suivis par un marcheur. L'objectif de cette solution est de permettre à l'utilisateur, au travers d'une série de clics dans l'environnement de simulation, de définir les différents points de passage par lesquelles le marcheur devra passer. L'utilisateur pourra alors définir un ou plusieurs chemins qui pourront être joués avec ou sans rejou2.9.

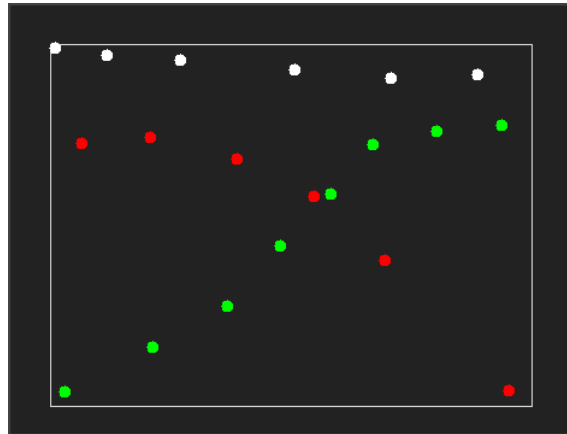


FIGURE 2.9 – Trois chemins définis par des points de passage chaque point appartient au chemin dont il a la couleur vert, blanc ou rouge.

L'utilisateur est alors dans la capacité de créer une multitude de chemins et donc une multitude de marcheurs suivant ces différents chemins. Toutefois, les marcheurs suivront le chemin préétabli de manière très linéaire 2.10. Les points définissant le chemin étant considéré comme des points de passage, le marcheur cherchera à rejoindre chaque point au travers d'une ligne droite. Afin de permettre à l'utilisateur de disposer du plus de flexibilité possible, plusieurs fonctionnalités ont été imaginées et implémentées :

- L'utilisateur peut choisir s'il souhaite effectuer un départ en simultané de tous les marcheurs, auquel cas tous les marcheurs démarreront en même temps en suivant leurs chemins respectifs
- L'utilisateur peut choisir de créer un départ en différé où le moment de départ de chaque marcheur est déterminé aléatoirement.
- L'utilisateur peut choisir le temps qu'un marcheur mettra à parcourir la distance entre deux points.
- L'utilisateur peut choisir le temps qu'un marcheur mettra pour parcourir la

distance totale. Si cette solution est privilégiée, l'utilisateur ne pourra plus choisir le temps de parcours entre deux points du chemin.

La mise en œuvre de cette solution pour enrichir le jeu, bien que rudimentaire, a permis de valider que l'utilisation d'un simulateur était une véritable force. Cela permet effectivement de générer facilement une quantité de nouveaux scénarios considérable, il ne reste alors plus qu'à envoyer de manière périodique les données générées au système voulu, dans notre cas il s'agira de l'analyseur.

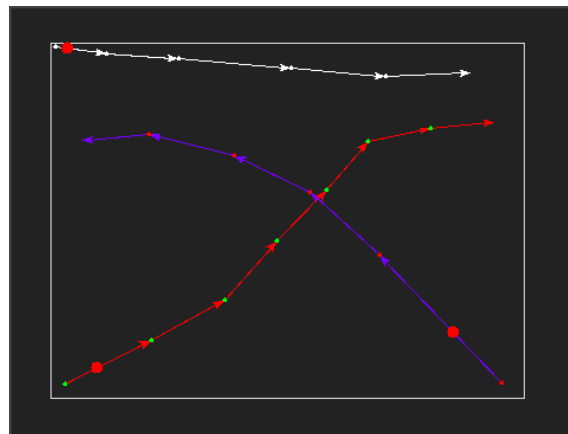


FIGURE 2.10 – Trois marcheurs suivent trois chemins qui ont été définis en utilisant des points de passage.

En revanche, cette première solution a également mis en avant une première série de problématiques :

- Les chemins sont diversifiés, car ils sont générés par l'utilisateur, toutefois, ils sont suivis par les marcheurs de manière très robotique. Lorsqu'un marcheur suit son chemin, il part d'un premier point pour rejoindre un second point en ligne droite. Une fois atteint, le marcheur adopte un virage brutal afin de rejoindre le point suivant toujours en suivant une ligne droite, et ce jusqu'à ce qu'il n'y ait plus de points.
- Il a fallu se poser la question de la vitesse, est-ce que le marcheur doit parcourir le chemin à la même vitesse, quel que soit le nombre de points ou est-ce que le marcheur doit parcourir le chemin entre deux points en respectant un même temps donné? Dans le premier cas, le marcheur se déplacerait toujours à la même vitesse tout au long du chemin sans montrer de signes

d'accélération ou de ralentissement. Dans le second cas, le marcheur aurait une vitesse qui serait définie par la distance qu'il y a entre deux points : si la distance est grande, le marcheur sera rapide et si la distance est courte le marcheur sera lent entre ces points.

- Dans notre cas d'étude, les caméras sont inclinées par rapport à la scène ce qui veut dire que la scène n'est jamais observée complètement de dessus où complètement de face il y a donc un effet de perspective qui entre en jeu. Si le chemin d'un marcheur se révèle être plutôt vertical par rapport à la scène filmée, cela veut donc dire que le marcheur est censé s'éloigner de la caméra. Dans ce cas, la vitesse du marcheur devrait normalement réduire en fonction de sa distance à la caméra. A contrario si le marcheur se rapproche de la caméra sa vitesse doit augmenter et s'il se déplace uniquement de manière horizontale la vitesse devrait être relativement constante.

Afin d'amener un comportement plus humain aux différents marcheurs, il a fallu identifier ceux qui pouvaient être améliorés dans les problématiques précédentes afin de proposer une nouvelle version. L'objectif de cette nouvelle version est le même que précédemment, mais doit intégrer l'idée de devoir se rapprocher au plus d'un comportement humain. Une nouvelle version de cette solution a donc été réfléchi et développée afin de répondre aux différentes problématiques en s'assurant que :

- Le suivi du chemin doit se faire de manière moins robotique, en adoptant des courbes dans les virages plutôt que des angles bruts.
- L'utilisateur peut modifier la vitesse du marcheur sur l'ensemble du chemin.
- La perspective doit être intégrée en modifiant la vitesse du marcheur en fonction de sa position sur l'axe Y (les ordonnées).

En ce qui concerne la première problématique, la trajectoire humaine ayant une allure moins robotique, avec des courbes plus délicates que de simple changement d'angle. Il a fallu trouver une solution permettant d'adoucir les angles empruntés par les marcheurs. Il s'avère qu'une trajectoire humaine lorsqu'elle n'est pas influencée par l'environnement va avoir tendance à être courbée avant et après le point où il est nécessaire de changer d'angle. Dans la figure 2.3.3.a, nous pouvons observer le parcours que suivrait un marcheur si nous lui donnons 3 points de passage. Or la trajectoire qu'un humain aurait pu emprunter s'approcherait plus

de la figure 2.3.3.b.

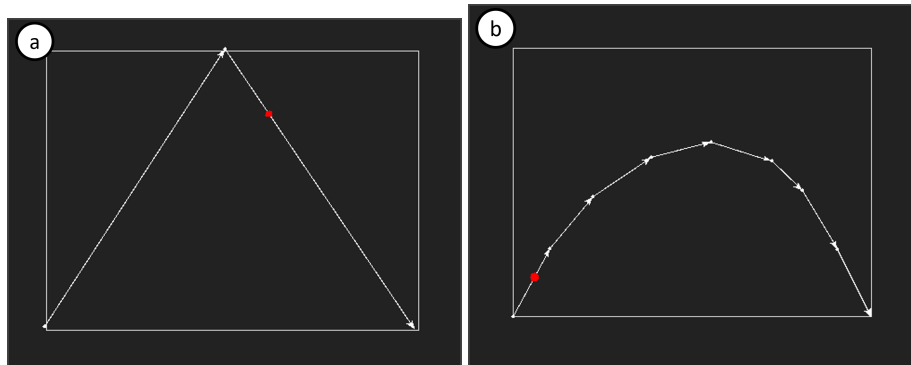


FIGURE 2.11 – (a) La trajectoire obtenue à l’aide de 3 points de passage et (b) une trajectoire plus humaine permettant de partir et d’arrivée des mêmes points.

Notre volonté est donc, à partir d’un ensemble de 3 points, d’adoucir le chemin qui pourrait être tracé entre ces points en ajoutant des points de passage supplémentaire dans le chemin. Étant donné que ces points sont dans un espace deux dimensions, les recherches ont été menées afin de déterminer comment tracer une courbe à partir de cet ensemble de points. Plusieurs solutions peuvent être envisagées, les plus évidentes sont :

- L’utilisation des courbes de Bézier dans leur forme polynomiale. Étant donné que les points ne sont pas pondérés pour faciliter la compréhension du simulateur, il n’est pas nécessaire d’utiliser des courbes de Bézier rationnel.
- L’utilisation des courbes d’interpolation de Lagrange.

Dans les deux cas, les courbes sont des courbes polynomiales ce qui permet leur obtention à partir d’un ensemble de points dans un espace 2D. Dans le cas de la courbe d’interpolation de Lagrange, nous nous assurons que le marcheur suivra un chemin passant par tous les points de passage définis par l’utilisateur [Luo+15]. Toutefois, l’utilisation de ces courbes implique un fort degré d’oscillation dans l’éventualité où le chemin présente des points de passage éloignés les uns des autres comme le montre la figure 2.12. De plus, l’interpolation de Lagrange permet de définir une courbe selon un axe ne pouvant évoluer que de manière croissante ou décroissante. Ce qui implique dans notre cas que, quel que soit l’ordre dans lequel les points de passage ont été créés par l’utilisateur, la courbe passera d’abord par le point ayant la plus petite abscisse et terminera par le point ayant



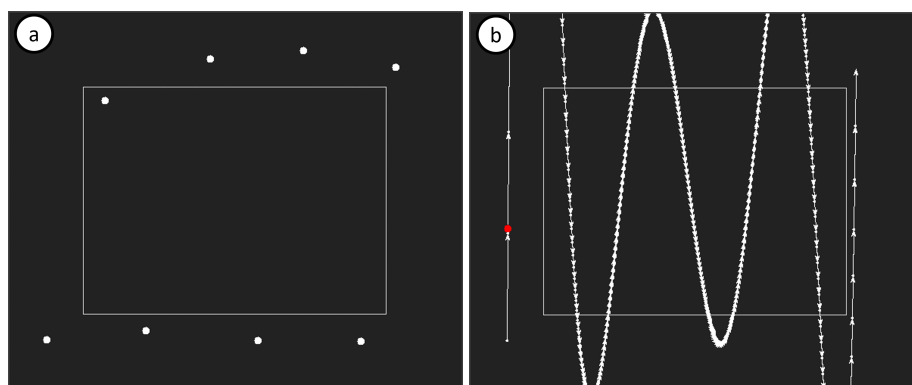


FIGURE 2.12 – (a) Une série de points de passage et (b) l’interprétation d’un chemin passant par ces points de passage en utilisant les courbes d’interpolation de Lagrange.

la plus grande abscisse ou inversement. Il n’est alors pas possible par exemple de proposer un chemin où le marcheur se déplace jusqu’à une certaine distance puis fait demi-tour pour revenir sur ses pas.

Ces contraintes ont impliqué de mettre de côté l’utilisation des courbes d’interpolation de Lagrange, car elles apporteraient trop de contraintes à l’utilisateur lors de leurs utilisations. En revanche, l’utilisation des courbes de Bézier dans leur forme polynomiale montre un intérêt certain. En effet, bien que la courbe ne passe pas par les points de passage définis par l’utilisateur, leur utilisation est beaucoup plus intuitive et permet de faire des chemins suivant la chronologie des clics effectués par l’utilisateur. Cette méthode permettant d’adoucir les angles et s’approchant le plus possible de la démarche humaine. Les courbes de Bézier sont notamment utilisées afin de définir des courbes évitant les changements d’angle brusque[JKV09; SK07].

À l’aide des courbes de Bézier, nous pouvons par exemple, à partir de trois points, créer une courbe commençant d’un premier point, terminant au troisième point et passant plus ou moins proche du second. Afin de faciliter le stockage et de ne pas surcharger en calcul le processeur, il a été choisi de faire une conversion des chemins avant de démarrer la simulation. En effet, si le calcul se fait après le démarrage de la simulation, cela veut dire que pour chaque nouvelle coordonnée il faudra calculer la courbe de Bézier associée et la position du marcheur sur cette courbe. La solution retenue consiste donc à transformer un ensemble de trois

points ou plus représentant le chemin brut en un ensemble de  $N$  points. Ces  $N$  points représentent les points de passage par lesquels passe à la courbe de Bézier ainsi généré. Considérons que nous souhaitons reproduire un équivalent de la figure 2.3.3.b avec les courbes de Bézier. Il suffira alors de reproduire la figure 2.3.3.a et d'utiliser les points composants ce chemin pour former notre courbe de Bézier comme le montre la figure 2.13.

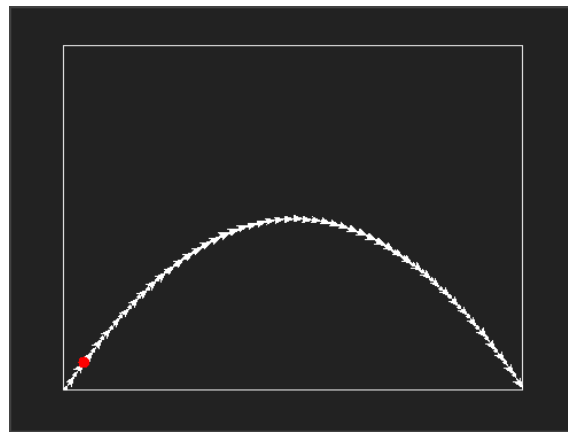


FIGURE 2.13 – Utilisation des courbes de Bézier sur les points de passage qui ont été définis en 2.3.3.a.

D'un point de vue de la simulation, l'utilisation des courbes de Bézier permet également l'avantage d'exclure les comportements aberrants. En effet, il devient alors impossible de faire un changement de trajectoire instantané qui devrait être physiquement impossible dans la réalité, comme le montre la figure 2.14.a. À la place, nous obtiendrons une courbe dont le rapprochement des points permet d'abord au marcheur de ralentir avant d'accélérer de nouveau dans le sens opposé comme le montre la figure 2.14.b.

Nous sommes ainsi capables, à partir de trois points, de générer une courbe de Bézier composé du nombre de points voulu, il a donc fallu régénérer la courbe en fixant des limites de granularité, en ajustant le nombre de points la composant. En effet, la quantité de points utilisés pour tracer la courbe va impliquer la prise en compte de nouvelles contraintes :

- L'utilisation des ressources disponibles.

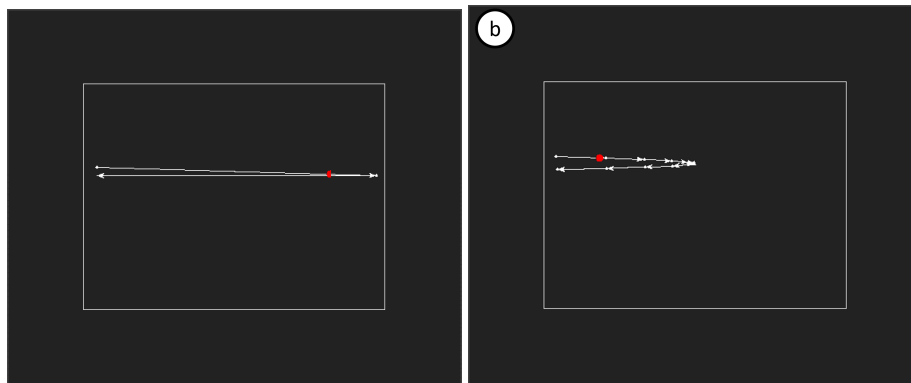


FIGURE 2.14 – (a) Une trajectoire définie par trois points de passage fortement éloignés les uns des autres et (b) Trajectoire modifiée par les courbes de Bézier réduisant la distance parcourue et augmentant le nombre de points de passage.

— L’adaptation de la vitesse du marcheur.

En ce qui concerne la contrainte liée aux ressources, chaque point composant la courbe doit être calculé. Si le nombre de points augmente, le temps de calcul nécessaire à l’obtention de la courbe augmente proportionnellement. De plus une fois que tous les points sont calculés il faut les stocker en mémoire. Bien que le stockage en soi ne présente pas de problème au vu de la quantité de mémoire disponible sur les appareils modernes, c’est l’accès à cette ressource qui est problématique. En effet, chaque traitement impliquant l’utilisation de ces données en mémoire sera ralenti par leur quantité. En ce qui concerne l’adaptation de la vitesse, dans le cas de la solution où le marcheur parcourt le chemin à une vitesse constante, le marcheur arrivant dans un virage important ne montrera pas de signes de ralentissement, nous faisant ainsi nous éloigner de la réalité. En revanche s’il a été choisi que le marcheur doit rejoindre chaque point composant son chemin en ayant le même temps de parcours entre chaque point. Nous pourrions observer, dans le cas où ce nombre de points est trop important, que le marcheur qui suit ce chemin se trouvera dans les courbes à être beaucoup plus lent (figure 2.15.b) qu’un marcheur se trouvant sur les extrémités ou sur les parties les plus droites de la courbe (figure 2.15.a).

Ces observations n’indiquent toutefois pas qu’il y ait une solution à privilégier plutôt qu’une autre. En effet, afin de permettre à l’utilisateur une exploitation optimale du simulateur, le choix de la vitesse et du nombre de points composant la



FIGURE 2.15 – (a) Zone de déplacement à vitesse constante et (b) zone de déplacement ralenti dans le virage.

courbe sera paramétrable. Cela permettra de produire des courbes plus ou moins complexes avec des temps de parcours liés à ces courbes plus ou moins importants. Le fait de pouvoir modifier ces paramètres permet donc une plus grande flexibilité dans les scénarios.

### 2.3.4 Les Boids

Bien que cette version du simulateur pourrait permettre de valider le fonctionnement attendu de ce dernier, il reste séquentiel. Le fait que le système soit séquentiel pose un problème dans la mesure où une marche doit pouvoir être interrompue à tout moment par un événement anormal, c'est un des grands objectifs du simulateur. Intégrer dans le déplacement du marcheur un événement aléatoire est compliqué et contre-intuitif. En effet, les trajectoires étant calculées à l'avance, l'insertion d'un élément aléatoire imposerait alors de devoir régénérer toutes les trajectoires pour le prendre en compte. Cette régénération de toutes les trajectoires si elles sont trop importantes nécessitera une puissance de calcul importante en très peu de temps et provoquera une latence dans le système.

Afin de gérer ce problème, il faut fatalement disposer d'un système capable de s'adapter à une situation aléatoire. Pour ce faire, il existe dans la littérature des modèles capables de s'autogérer dans leurs déplacements en répondant à des

règles simples. La solution retenue se base sur les Boids. L'objectif d'un Boid est d'évoluer dans un environnement en respectant un nombre de règles qui lui ont été fixées[LZ19]. Il s'agit d'une adaptation des méthodes multiagents dont l'objectif est de faire coopérer plusieurs systèmes autonomes afin d'atteindre un objectif spécifique[OFM07]. Chacune de ces règles a une incidence plus ou moins forte sur son comportement, il ne peut en aucun cas déroger à ces règles et ces règles peuvent être plus ou moins complexes et en plus ou moins grande quantité. Dans notre cas, un Boid n'a que peu d'intérêt à exister seul, notre objectif étant de simuler une place potentiellement remplie d'individus, il est préférable que les Boids évoluent en groupe. Les règles définies influencent sur l'ensemble des comportements des Boids, il a donc fallu les réfléchir de sorte qu'elles soient les plus cohérentes possibles :

- la première règle est celle qui a le poids le plus fort sur l'ensemble des règles, elle consiste à empêcher les Boids d'entrer en collision les uns avec les autres.
- La seconde règle consiste à essayer de maintenir un groupe avec d'autres Boids,
- la troisième règle impose à chaque Boid d'essayer de suivre le mouvement de son groupe,
- la quatrième impose de fuir certains types de Boids qui représentent un danger potentiel.

À ces règles vont s'ajouter des règles de déplacement telles que ne pas sortir d'une certaine zone et essayer d'avoir un comportement le plus humain possible en évitant d'avoir des changements de trajectoire trop brusque.

L'utilisation de cette technologie permet de disposer d'une infinité de scénarios qui vont pouvoir se jouer de manière autonome. Le fait d'utiliser un système de règles permet notamment d'en instaurer de nouvelles si la situation le nécessite. Nous pouvons par exemple imaginer que notre simulateur dispose de plusieurs types de Boids. Ceux évoqués pour l'instant sont les Boids classiques et les Boids présentant un danger, mais nous pouvons également imaginer des Boids "enfants", dont le comportement peut varier d'autres Boids en faisant par exemple en sorte qu'un Boid "enfant" soit toujours à proximité d'un Boid classique.

L'utilisation de ces règles permet également une forte interactivité avec la population de Boids. Étant donné que chaque Boid effectue ses déplacements de manière quasi aléatoire, il n'y a pas besoin de calculer l'ensemble des trajectoires des marcheurs. Nous pouvons alors imaginer insérer des événements en plein milieu d'une simulation.

Pour compléter certains scénarios, la simulation a également été pensée de sorte à pouvoir fusionner l'utilisation des Boids et des scénarios séquentiels. L'utilisation des deux méthodes combinées ne permet pas d'améliorer l'efficacité du scénario séquentiel, car ce dernier ne sera pas réceptif aux événements extérieurs. Toutefois, si nous programmons un Boid dangereux de manière séquentielle, ce dernier suivra une trajectoire quoiqu'il arrive, ce qui forcera le comportement des Boids à s'adapter à la situation.

Dans le cadre de notre simulateur, la problématique liée à la génération de trajectoires n'est bien sûr pas la seule. En effet, le simulateur doit non seulement générer des trajectoires humaines, mais il doit également permettre d'imiter le comportement de marche. Lors des analyses et des études sur les vidéos, il a été observé un phénomène d'oscillation dans la détection lorsque les personnes marchent. Comme le montre la figure 2.16, cette oscillation est due à la fenêtre de détection qui est apposée par le système de détection d'individus. La fenêtre de détection englobe la totalité de l'individu si ce dernier est en train de marcher, l'espace entre les pieds évolue constamment, ce qui provoque ce phénomène d'oscillation. Nous observons sur la figure 2.16.a une image où l'individu a les pieds relativement proches, la fenêtre de détection et donc petite. En revanche, nous observons sur la figure 2.16.b un individu ayant les pieds relativement éloignés l'un de l'autre ceux qui générera une fenêtre de détection plus grande. Ce phénomène d'oscillation impacte donc trois paramètres dont nous pouvons initialement considérer leurs évolutions comme étant linéaires :

- La hauteur de l'individu,
- la largeur de l'individu,
- la position du centre de gravité de l'individu.

Étant donné que les Boids sont régis par des règles, il est tout à fait envisageable d'appliquer ces règles de manière à intégrer l'oscillation.



FIGURE 2.16 – (a) Un individu durant la marche ayant les jambes rapprochées et (b) le même individu durant la marche ayant les jambes éloignées.

Dans le cas de la hauteur et de la largeur de l'individu, ces derniers évoluent de manière linéaire en fonction de la distance à la caméra. Si l'individu est proche de la caméra, ces paramètres seront plus grands que lorsqu'il est éloigné de la caméra. En ce qui concerne l'oscillation, la solution imaginée consiste à agrandir ou diminuer l'espace occupé par l'individu simulé en fonction de deux paramètres à savoir la distance l'individu par rapport à la caméra et la vitesse de déplacement de l'individu. La distance de l'individu impactant la quantité de pixels qui seront ajoutés ou retirés à l'espace occupé par l'individu. En ce qui concerne la vitesse, elle impactera la fréquence à laquelle les modifications de taille seront effectuées. Cette règle devra aussi prendre en compte la direction dans laquelle se dirige l'individu. En effet, si l'individu suit une trajectoire où ce dernier est soit complètement de face soit complètement de dos par rapport à la caméra, nous pourrions alors observer que le phénomène d'oscillation est très important sur la hauteur de l'individu comme le montre la figure 2.17. A contrario, le phénomène d'oscillation sera beaucoup plus présent sur la largeur de l'individu dans l'éventualité où celui-ci traverse l'image de gauche à droite comme le montre la figure 2.18.

Dans le cas de la position du centre de gravité, la gestion est beaucoup plus simple. Le centre de gravité est déplacé par les règles déjà en place. En effet, les règles qui régissent actuellement les Boids permettent le déplacement de ces derniers, ce faisant le centre de gravité et simplement déplacer. Le fait d'appliquer des règles qui modifient la hauteur et la largeur de l'individu impacteront directement le centre de gravité qui est calculé par rapport à la position de l'individu et la taille de sa zone de détection définie par la hauteur et la largeur de l'individu.

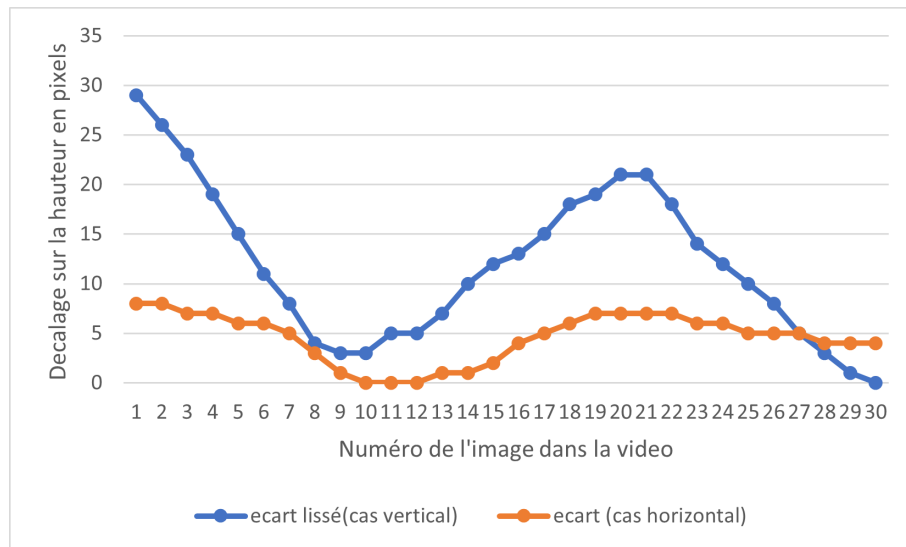


FIGURE 2.17 – Oscillation dans la hauteur sur la fenêtre de détection d'un individu ayant une trajectoire verticale à l'axe de la caméra en bleu et horizontal à l'axe de la caméra en orange.

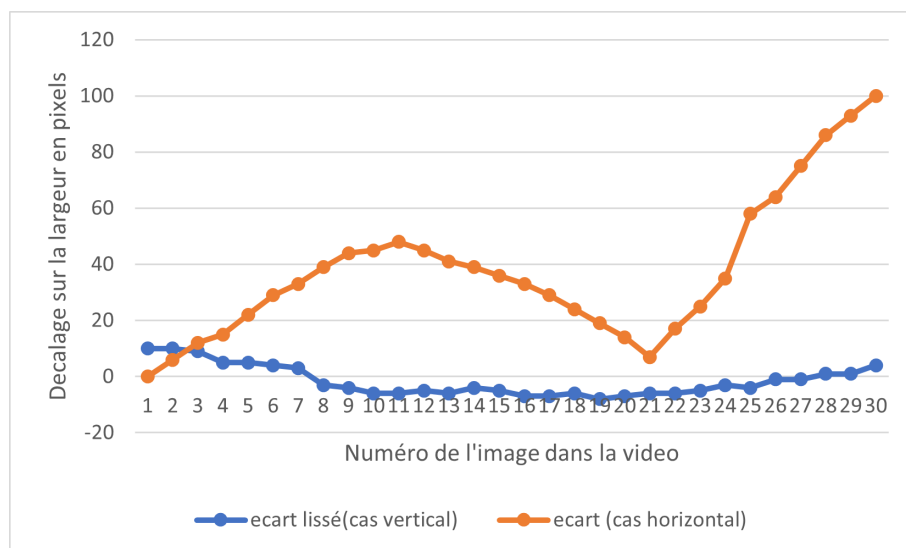


FIGURE 2.18 – Oscillation dans la largeur sur la fenêtre de détection d'un individu ayant une trajectoire verticale à l'axe de la caméra en bleu et horizontal à l'axe de la caméra en orange.



Le centre de gravité se trouvera donc toujours au centre du rectangle de détection de l'individu, quelle que soit sa position ou sa taille.

L'avantage de l'utilisation d'un système à règles s'est d'autant plus justifié, car nous pouvons facilement utiliser les règles développées précédemment afin d'influer sur les caractéristiques du Boid. Les règles citées jusqu'ici n'influent pas sur la taille de la fenêtre de détection, toutefois, nous pouvons intégrer une règle qui fait osciller ces paramètres afin de se rapprocher au plus près de la réalité. Une autre règle intervient dans la prise en compte de la perspective. Dans la réalité, la caméra ne filme pas en vue de dessus, il y a un angle à prendre en compte qui va influencer sur la perspective de l'image et donc provoquer une estimation des distances différentes d'une personne qui se trouve près de la caméra ou d'une personne qui s'en éloigne. La section 2.3.7 aborde ce sujet plus en profondeur.

L'ensemble de ces règles permet donc de disposer d'un simulateur performant, mais aussi disposant d'une grande diversité dans ses scénarios. L'ensemble de ces règles peut être modifié et amélioré afin de correspondre au mieux à la vérité terrain. Les règles pourront alors être validées si les résultats obtenus à l'analyse de la vérité terrain et de la simulation sont suffisamment proches.

### **2.3.5 L'optimisation du temps de traitement**

Lors du développement du simulateur, il s'est avéré que les scénarios impliquant une présence d'individus importante (de l'ordre de la centaine), le simulateur ralentissait au point de ne plus pouvoir générer suffisamment d'information pour alimenter l'analyseur en temps réel. Afin de contourner ce problème, il était envisageable de simuler les scénarios intégralement et de les analyser a posteriori une fois que tout le scénario était généré. Toutefois, les temps observés pour la génération d'un scénario peuvent s'avérer être considérablement élevés comme le montre la figure 2.19.

Un des objectifs fondamentaux du simulateur est de disposer de la même capacité de génération d'information qu'une caméra de vidéosurveillance. Notre simulateur doit donc pouvoir générer une quantité d'informations suffisamment

importante dans un délai le plus court possible afin de restreindre le moins possible la génération de scénarios. Cependant, cet axe ne présentait pas au départ une importance capitale pour le projet étant donné que ce dernier devait être réalisé à titre démonstratif. Il a tout de même été imaginé une manière d'optimiser le temps de traitement par rapport au contexte dans lequel le simulateur a été développé.

Le simulateur est un élément qui a été conçu et pensé de manière modulaire, l'ensemble des briques logicielles qui le composent doivent pouvoir être interchangeables afin de s'adapter au mieux aux différents scénarios. Dans le cadre du projet, les trois scénarios ont été identifiés et les briques logicielles nécessaires à la bonne détection de ces scénarios ont été développées. Toutefois, le développement du simulateur a révélé un potentiel intéressant pour le domaine de la recherche. En effet, l'aspect modulaire du simulateur le rend donc adaptable à de nombreuses autres situations que la simulation d'individus dans le cadre de l'analyse comportementale. À l'issue du livrable du projet, nous avons pu mettre en évidence que le simulateur pouvait être amélioré, notamment au niveau du temps de traitement.

Au moment de l'étude, le simulateur a été développé en Python3, les composants logiciels étaient donc tous développés dans ce langage. Bien que ce langage s'avère être de plus en plus efficace en termes de temps d'exécution dans certains cas d'utilisation [JC19]. Python3 reste souvent assez lent à l'exécution et présente une gestion de la mémoire moins efficace que des langages compilés [Pre00; Sha+18]. La vitesse du simulateur s'avère être très dépendante de la manière dont le langage est exécuté et afin de se permettre la plus grande flexibilité possible et le meilleur temps de traitement, sa conception a été imaginée au travers de l'utilisation de Multiprocessing.

Le Multiprocessing se révèle être d'une efficacité considérable lorsqu'il s'agit d'améliorer les performances d'un programme exécutant des tâches pouvant être réalisées en parallèle. Cela va bien sûr dépendre de la possibilité de sous-division des instructions du programme afin de les répartir entre différents processus [Oza18]. L'équivalent informatique d'un processus est un sous-programme capable de tra-

vailler en parallèle du programme principal afin d’alléger la charge de travail du programme principal. La figure 2.19 présente les résultats obtenus lors de la simulation de scénario de la version du simulateur utilisant le Multiprocessing. Les tests ont été effectués en se basant sur une quantité de nuées et un nombre de Boids les composants. Initialement l’observation attendue lors de l’exploitation

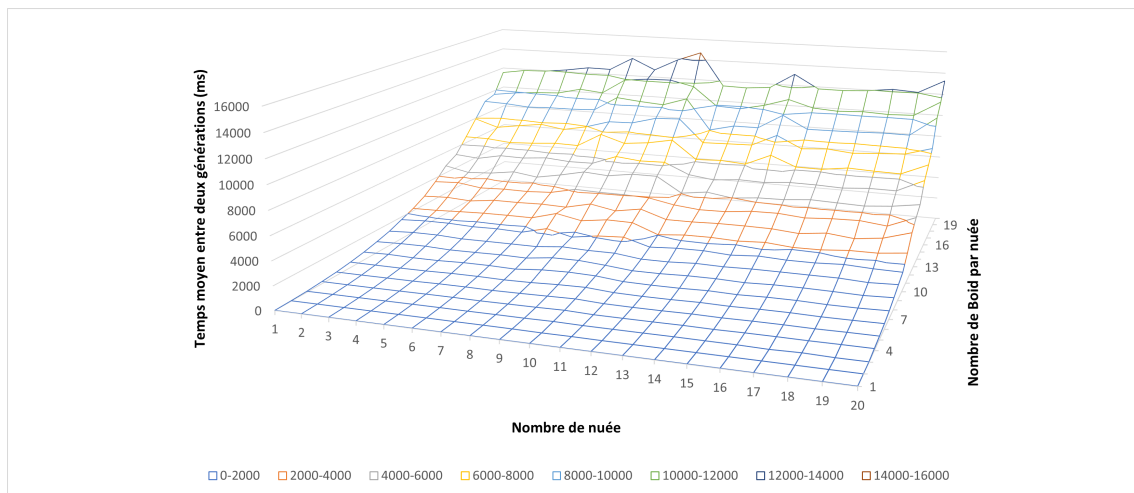


FIGURE 2.19 – Temps de traitement du simulateur pour un nombre donné de nuées et de Boids par nuée.

des processus aurait dû être une courbe ou le temps de traitement augmente avec la complexité de calcul nécessaire à la génération d’un mouvement. En d’autres termes, le temps de traitement le plus important devait être observé lorsque le nombre de nuées et le nombre de Boids composants ces nuées étaient le plus élevé. De plus, le temps de traitement d’une nuée de 20 Boids ou de 20 nuées d’un Boid devrait être sensiblement le même, car dans les deux cas la quantité de calcul nécessaire pour effectuer un mouvement reste la même :

- Dans le cas d’une nuée de 20 Boids, nous exécutons une boucle parcourant les 20 individus afin de les mettre à jour.
- Dans le cas de 20 nuées de 1 Boid, nous exécutons 20 boucles sur 1 Boid par boucle.

En considérant que la création des différents processus a été faite en amont, la complexité des deux cas devrait donc être la même.

Or, dans le cas du simulateur qui a été utilisé et validé pour le projet, nous avons pu observer des résultats inattendus. La figure 2.19 montre que le nombre de nuées influe légèrement sur le temps de traitement là où le nombre de Boids composant la nuée agit de manière quadratique sur le temps de traitement. Cette observation révèle donc que le nombre de nuées n'impacte que peu les performances, c'est la quantité de Boids qui va augmenter le temps de traitement. La première conclusion que nous pouvons tirer de cette observation est que la quantité de nuées n'impacte pas le temps de traitement, car la charge de calcul est répartie de manière équitable entre chaque processus.

L'objectif du simulateur est de pouvoir simuler potentiellement une quantité importante d'individus. Ces individus peuvent dans les cas extrêmes composer des groupes de plusieurs dizaines d'individus. Nous pouvons par exemple faire face à une foule composée d'une quantité importante d'individus lors d'un événement sportif ou d'une manifestation. Or, dans le cas de 20 nuées composées de 20 boids, nous pouvons observer un temps de traitement moyen entre chaque mouvement d'environ 13 secondes.

### 2.3.6 Identification des causes du temps de traitement

L'étude du fonctionnement du langage de programmation choisi a révélé que les processus n'étaient pas réellement indépendants les uns des autres et continuaient d'exister dans un environnement partagé. En effet, Python3 utilise dans le cas du découpage par processus une bibliothèque appelée Multiprocessing[Pyt22a]. Cette dernière est censée permettre l'exécution de tâches en parallèle. Toutefois, il s'avère que cette librairie met à disposition des processus un système de partage de données. Ce système de partage de données est en réalité un autre processus qui est parfois obligé de mettre en attente les processus l'utilisant afin de permettre l'accès à la mémoire qu'il propose[Pyt22b]. Ce faisant, nos processus sont alors ralentis par le gestionnaire de mémoire ce qui nous empêche de tirer pleinement parti de la puissance du système sur lequel était exécuté le simulateur.

En recherchant les causes de ce phénomène de manière plus profonde, nous avons constaté que le temps de traitement était principalement impacté par deux paramètres :

- Le premier est lié au partage de la mémoire. En effet, parmi les règles que les Boids doivent respecter, il en est une qui nécessite que chaque Boid ait connaissance des Boids des autres nés afin de pouvoir éviter une éventuelle collision. Or, l'accès en mémoire se fait au travers de la bibliothèque Manager qui fait également partie de la bibliothèque Multiprocessing. Cette bibliothèque se révèle être très efficace dans les cas où les accès à la mémoire sont ponctuels. Il s'avère que dans notre cas certains traitements peuvent, en fonction du nombre d'individus à traiter, être inférieurs à 10ms. Ce temps de traitement empêche la librairie de fonctionner correctement et provoque un blocage de l'exécution de manière complètement aléatoire.
- Le second est un problème lié au langage. En Python3, les processus ne sont pas de réels processus. Ils sont simulés de sorte qu'ils cohabitent dans un même environnement et effectuent leurs tâches en montrant une impression de parallèle. En réalité, comme montré en figure 2.20 les processus ont plus tendance à s'exécuter les uns après les autres afin de pouvoir accéder à la mémoire ce qui implique que le résultat ne sera exploitable que lorsque tous les processus auront terminé. Il peut arriver que les processus fonctionnent en parallèle, mais cette gestion étant propre au langage Python3 il est difficile de le contrôler.

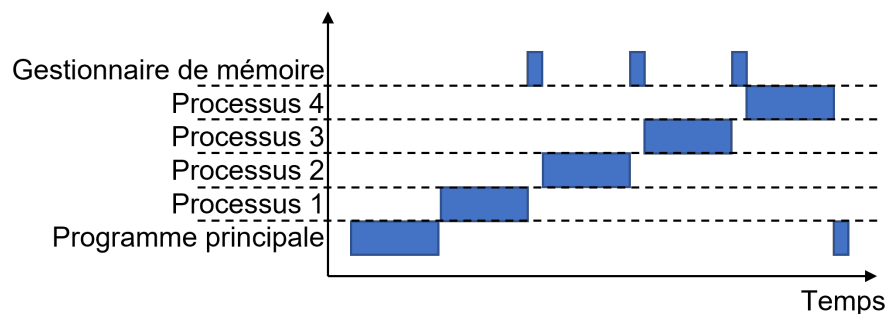


FIGURE 2.20 – Observation de l'exécution des différents processus de la première version du simulateur en fonction du temps.

Afin de résoudre le premier problème, la solution la plus évidente à mettre en œuvre est un système de synchronisation de la mémoire qui permettrait d'éviter la plupart des conflits. La bibliothèque Multiprocessing dispose d'un système permettant de bloquer l'accès aux ressources partagées. L'utilisation de ce système

permettrait de gérer les accès à la mémoire de manière ordonnée, ce qui permettrait de résoudre la cause principale du ralentissement. En effet, nous avons pu observer qu'en l'état, les processus les plus lents n'ont plus accès à la mémoire dès lors qu'un autre processus est suffisamment rapide pour pouvoir y accéder en continu. En revanche, la mise en place de cette solution va restreindre l'accès à la mémoire à chaque processus pendant un certain temps. Cette restriction permet donc d'un côté de ne plus avoir de blocage du système, mais va d'un autre côté ralentir considérablement ce dernier en forçant l'attente.

En ce qui concerne le second problème, il a fallu trouver une solution permettant de créer des processus réels. Pour ce faire, il faut faire en sorte que le système exécutant le simulateur reconnaisse les processus que nous créons comme de réels processus. En effet, si nous sommes capables d'exécuter de réel processus informatique nous pouvons alors exploiter entièrement la puissance de notre système. De plus, nous ne serions plus dépendants d'un outil de gestion de mémoire, car il faudrait trouver une solution permettant de s'acquitter complètement de la mémoire partagée. Étant contraint par le langage de programmation, les solutions permettant de régler le problème sont limitées :

- La première solution consisterait à changer de langage de programmation afin de disposer d'un langage permettant une gestion des processus comme s'il s'agissait de processus informatique. Un langage comme le C par exemple permettrait de répondre à ce besoin.
- La seconde solution consisterait à exécuter à partir du programme principal d'autres sous-programmes sur le système effectuant les opérations souhaitées. Dans ce cas, chaque autre sous-programme serait considéré comme un processus indépendant, car il n'aurait pas été créé dans l'environnement du simulateur, mais bien dans l'environnement du système.

La première solution impliquant un changement complet de stratégie en changeant de langage de programmation, c'est la seconde solution qui a été privilégiée. Il a donc fallu trouver une manière de permettre au programme principal d'exécuter un ensemble de sous-programmes tout en s'assurant que le partage de données entre les différents sous-programmes était toujours possible. En explorant les différents besoins de partage de mémoire il s'est avéré que cette dernière n'était pas totalement indispensable.

En effet, il faut que toutes les nuées aient une connaissance de la position des Boids composant les autres nuées, mais en aucun cas une nuée n'est censée modifier la position des autres nuées. Cette notion est importante, car dans ce cas il n'est pas réellement nécessaire de partager la donnée pour ce qui est de la connaissance des autres Boids. Il est par exemple envisageable de communiquer à un sous-programme la position des Boids des autres nuées avant que le sous-programme n'ait effectué son calcul. Il en est de même pour l'obtention du résultat du sous-programme, il n'est pas nécessaire que ce dernier accède à la mémoire pour transmettre son résultat. Nous pouvons par exemple récupérer le résultat de chaque sous-programme de manière indépendante afin de reconstituer l'ensemble des nuées calculé.

Le langage Python3 comme de nombreux autres langages mettent à disposition de l'utilisateur des solutions permettant d'exécuter des instructions sur le système d'exploitation. Dans notre cas, le système utilisé est un Ubuntu 20.04 LTS. En utilisant une de ces solutions, nous pouvons imaginer pouvoir exécuter les sous-programmes directement en ligne de commande à partir de notre programme principal. La solution retenue est l'utilisation de la bibliothèque Popen qui permet de répondre à ce besoin[Pyt22c]. Étant donné que l'architecture initiale du simulateur a été pensée de manière modulable, les librairies gérant les Boids ont pu être retravaillées de sorte à pouvoir s'exécuter de manière indépendante.

En ce qui concerne la problématique de gestion de la mémoire, comme il a été expliqué précédemment il n'est pas nécessaire d'avoir un système de partage de mémoire dès lors que nous sommes capables de communiquer dans les deux sens avec les sous-programmes. Il a donc fallu imaginer un protocole de communication permettant de transmettre les informations à chaque sous-programme sur la position des différents individus déjà généré par les autres sous-programmes. Cette communication prend la forme d'un échange de données au format JSON.

Le format JSON se révèle de nouveau efficace dans ce projet et aucun traitement supplémentaire n'est nécessaire étant donné qu'il est déjà au format texte. Une fois ce protocole de communication établi entre le programme principal et les sous-programmes, nous avons pu nous assurer que les sous-programmes, bien

qu'ils sont exécutés par le programme principal, soient considérés par le système comme des programmes indépendants. Il s'est avéré que dès leurs exécutions, chaque sous-programme dispose des mêmes caractéristiques qu'un programme exécuté par le système. Le cycle de vie des processus se rapproche désormais plus de la figure 2.21.b que de la figure 2.21.a.

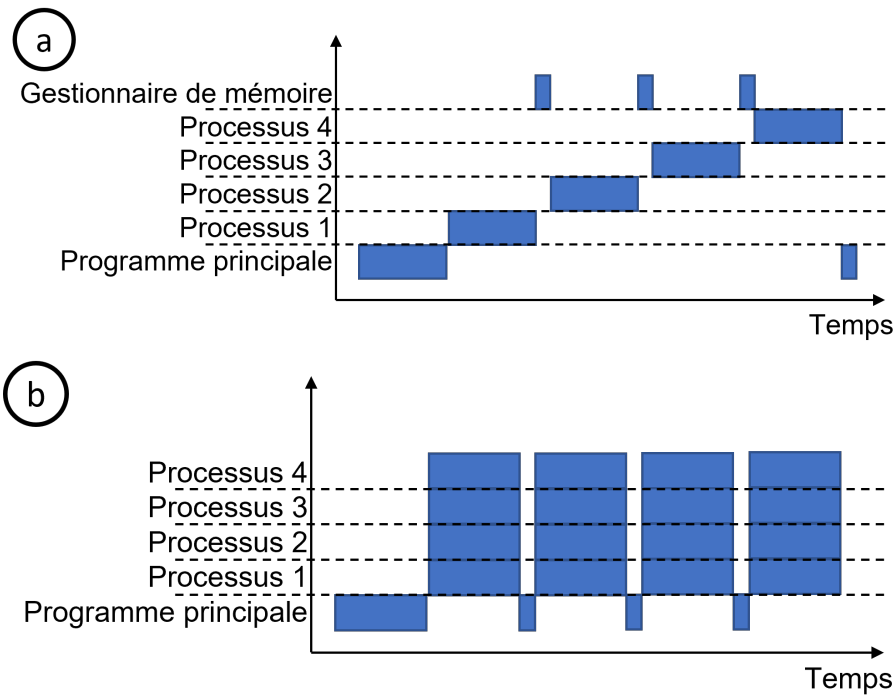


FIGURE 2.21 – (a) Observation de l'exécution des différents processus de la première version du simulateur en fonction du temps et (b) observation de l'exécution des différents processus de la nouvelle version du simulateur en fonction du temps.

En disposant de cette modification dans la manière de gérer les Boids, une nouvelle séquence de tests a été effectuée afin de comparer l'efficacité de la nouvelle solution à l'ancienne méthode. En observant les résultats en figure 2.22 dans les mêmes conditions et avec les mêmes paramètres de test, nous pouvons nous apercevoir de deux choses :

- Le temps d'exécution global est, au plus long de l'exécution, 10 fois inférieur à l'exécution la plus rapide de la première méthode.
- L'allure du graphique a complètement changé. Il a non seulement la forme attendue lors de la première expérimentation, mais aussi un temps entre



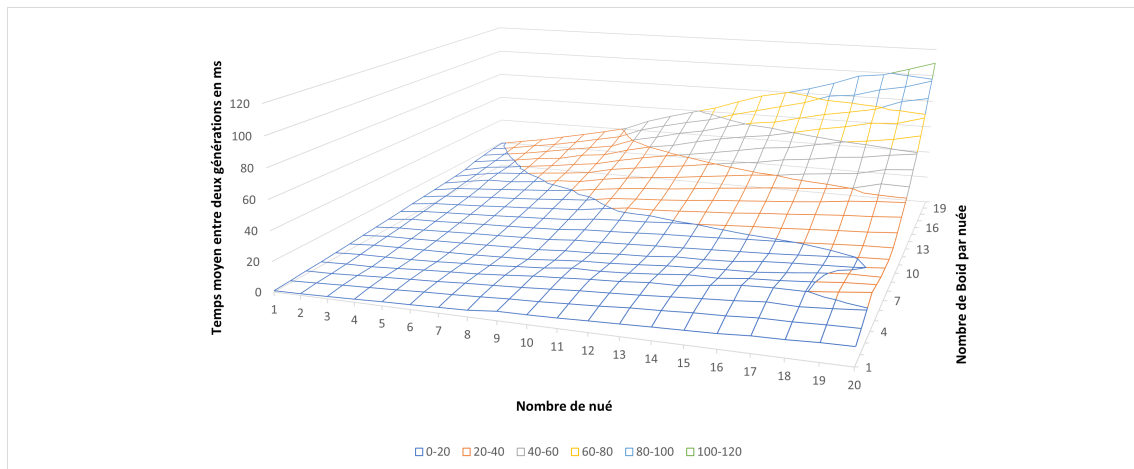


FIGURE 2.22 – Temps de traitement du simulateur pour un nombre donné de nuées et de Boids par nuée dans sa nouvelle version.

deux générations environ 100 fois moins important à la première expérimentation.

Dans la première méthode, le temps d'exécution dépendait du nombre de Boids plus que du nombre de nuées. Dans la nouvelle méthode, nous pouvons nous apercevoir que le temps d'exécution augmente à mesure que le nombre de Boids et de nuées augmente. Ce phénomène s'explique par l'exécution de réels processus. En effet, en comparant la méthode d'exécution des processus de Python3 et la méthode d'exécution de processus que nous venons de mettre en œuvre, nous pouvons nous rendre compte en figure 2.21.b que les processus peuvent s'exécuter en simultanée la ou en figure 2.21.a les processus n'avaient pas la possibilité de le faire.

### 2.3.7 L'optimisation de la perspective

L'objectif principal de notre simulateur est de pouvoir simuler un comportement humain au travers des différents algorithmes de génération d'individus et bien que cet objectif est primordial, il n'est pas le seul. En effet, l'autre objectif que doit pouvoir accomplir le simulateur et de retranscrire les données à l'analyseur dans les mêmes conditions que celle obtenue à l'aide d'une caméra de vi-

déosurveillance . Ces conditions ne peuvent être obtenues que si le simulateur applique à l'ensemble des données un effet de perspective. En effet, comme cela a été expliqué dans la section 2.2.6, les caméras de vidéosurveillance filment la scène avec un certain angle ce qui a pour effet d'appliquer un effet de perspective à l'image. Prenons l'exemple d'une place publique carrée sur laquelle des individus sont identifiables. Sans prendre en compte la perspective, nous pourrions imaginer qu'en vue de dessus nous observerions la figure 2.23.a, en réalité la caméra permettrait l'obtention une image ressemblante plus à la figure 2.23.b. Nous observons donc une déformation de la place qui passe d'une forme carrée à une forme trapézoïdale. Dans les explications précédentes, la gestion se fait principa-

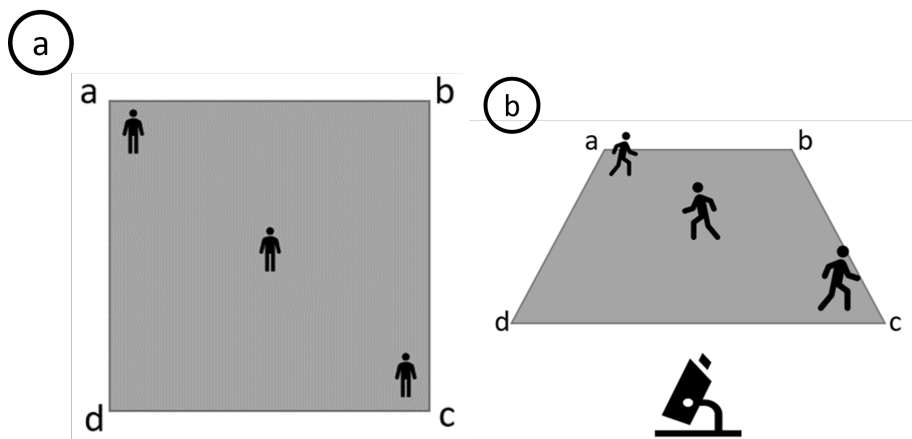


FIGURE 2.23 – (a) Vue de dessus de la zone observée et (b) Zone observée par la caméra appliquant un effet de perspective.

lement par l'ajout de règles au niveau des Boids sur lesquels nous appliquons une modification de la vitesse en fonction de leur position dans la simulation. Cette solution présente l'intérêt d'être facilement implémentable puisqu'il suffit d'ajouter de nouvelles règles que les Boids pourront suivre. L'inconvénient majeur de cette solution est qu'elle doit être implémentée de la même manière sur tous les éléments de simulation. Dans le cas où le simulateur utilise plusieurs nuées, toutes ces nuées doivent répondre de la même manière à la règle de la perspective. Il n'y a pas de réelle problématique sur ce point-là dans la mesure où nous pouvons facilement imposer une même règle à toutes les nuées. En revanche, si nous prenons le cas du suivi de chemin, au vu des contraintes de cette méthode sur la gestion du temps de trajet l'implémentation de la solution peut s'avérer être plus difficile. Il

en serait d'ailleurs de même pour toute autre méthode pouvant être implémentée sur le simulateur.

Des réflexions ont donc été menées sur la manière dont nous pouvons appliquer une notion de perspective à l'ensemble des éléments de la simulation. Étant donné que l'ensemble des éléments de la simulation évolue dans un environnement en deux dimensions (qui pourrait être assimilable à une observation d'une scène vue de dessus), nous pouvons envisager d'appliquer la perspective de manière globale à l'ensemble des éléments de simulation sans que ceux-ci aient besoin d'en avoir connaissance. En effet, à ce moment de l'étude, le simulateur est capable de générer une multitude d'individus répondant uniquement à des règles de déplacement. Une fois ces individus générés, les informations de position et de taille des différents individus sont envoyées à l'analyseur qui les gère en l'état.

L'autre solution permettant d'appliquer une perspective à l'ensemble de la simulation serait alors de considérer que les données générées par le simulateur sont effectivement issues d'un environnement en deux dimensions en vues du dessus sur lequel sera appliquée une transformation. Cette transformation agira comme une passerelle utilisant comme données d'entrée les données issues du simulateur et générant en sortie les données transformées à destination de l'analyseur. Une telle solution se révélerait alors efficace, quel que soit le type d'algorithme utilisé pour la simulation puisque la transformation serait appliquée de la même manière à tous les individus simulés.

Afin d'implémenter cette solution, il faut d'abord comprendre comment peut être effectuée la transformation. Pour ce faire, nous pouvons par exemple filmer une même scène avec une caméra orientée à différents angles. Cette solution semble toutefois laborieuse et les sources d'erreurs peuvent s'avérer nombreuses. En effet, cette prise de mesure se révélerait être très dépendante de l'objectif utilisé, de l'environnement, de la hauteur et de la précision de l'angle lors de l'acquisition. De plus, la littérature révèle que le calcul de perspectives à partir d'une caméra est possible, mais s'avère relativement complexe [Wol+91 ; Mua+04].

Toutefois, notre objectif final n'est pas nécessairement de convertir un environnement issu d'une caméra en environnement virtuel. En effet, l'objectif du simulateur est de générer un environnement virtuel s'approchant de la réalité. Or, étant donné que les scènes qui seront analysées ne sont pas préétablies et peuvent changer, nous pouvons tout à fait imaginer pouvoir simuler un environnement où l'angle de la caméra et le type de terrain peuvent être paramétrés.

En nous basant sur la littérature [Bar83 ; BV00], nous constatons qu'un environnement en perspective du point de vue d'une caméra est défini par deux éléments :

- Un point de fuite,
- une profondeur de champ.

En prenant en compte ces 2 éléments il est tout à fait possible de se rapprocher de la figure 2.23.b à partir de la figure 2.23.a. Le bord [AB] de la figure 2.24.a représente la limite éloignée de notre simulation, ce qui est équivalent à notre profondeur de champ. Le point de fuite F sur la figure 2.24.a doit quant à lui être positionné au centre de l'axe horizontal. La position du point F va permettre de tracer les droites partant de la base de la zone de simulation (le bord le plus proche de la caméra représentée ici par le bord [DC]). Nous pouvons alors transformer notre carré de simulation en trapèze en nous assurant que les bords [AD] et [BC] suivent les droites passants par (DF) et (CF) afin d'obtenir notre nouvelle zone de simulation localiser en [A'B'], [B'C'], [CD] et [D,A'] comme le montre la figure 2.24.b. Cette transformation permet de passer sur une vue en perspective toutefois l'angle d'orientation de la caméra n'est pas pris en compte, pour ce faire, il faut ajuster la distance du bord [A'B'] afin de leur approcher du bord [DC] et ainsi créer un effet d'écrasement de la zone de simulation comme le montre la figure 2.24.c. Si le bord [A''B''] est proche de [DC] cela signifie que la caméra est plutôt orientée vers l'horizon et si le bord [A''B''] est éloigné de [DC], cela signifie que la caméra est plutôt orienté vers le sol.

En appliquant cette notion supplémentaire à notre simulateur, nous disposons alors de tous les éléments qui ont pu être identifiés afin d'obtenir une bonne simulation de la réalité. C'est au travers de ce simulateur et des vidéos que nous avons à notre disposition que nous avons pu tester et validé les solutions d'analyse proposée dans la suite de cette thèse. De plus, la réalisation de ce simulateur a

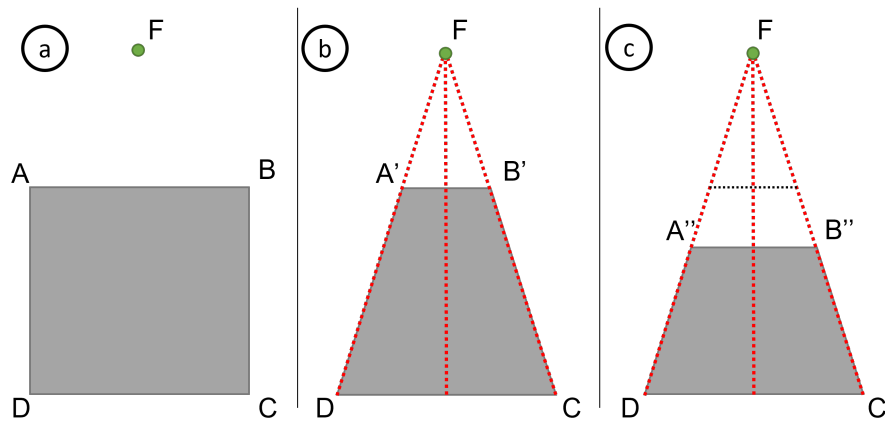


FIGURE 2.24 – (a) Zone de simulation sans perspective, (b) zone de simulation adaptée à un point de fuite F et (c) zone de simulation adaptée à un point de fuite F et à une profondeur de champ.

été pensée de sorte qu'ils puissent être réutilisés dans d'autres cas d'utilisation. Les bibliothèques qui ont été développées afin de réaliser le simulateur et plus particulièrement celle gérant le déplacement des différents individus peuvent totalement être modifiées afin de répondre à d'autres besoins. L'objectif fondamental du simulateur étant de gérer l'évolution de points dans un environnement de deux dimensions, en retirant la transformation permettant de passer la simulation en perspective, nous pouvons tout à fait imaginer déployer le simulateur dans d'autres domaines comme le déplacement d'objets maritimes et aériens ou encore dans des domaines plus microscopique.

# ANALYSE COMPORTEMENTALE DE FOULE

---

## 3.1 Préambule

La seconde partie abordée dans cette thèse concerne l'analyse comportementale au travers de différents dispositifs. Afin d'aborder ce chapitre dans de bonnes conditions, il est important de définir ce que nous considérons comme une analyse comportementale et ce que cela représente dans le cas d'une foule. L'analyse comportementale de foule suit généralement une méthode de traitement séquentielle [RML17; MH16] :

- Prétraitement d'une vidéo
- Détection des éléments souhaités
- Reconnaissance du comportement

L'analyse comportementale est un sujet vaste et peut s'appliquer à de nombreux domaines (c.f. : 1.2). Toutefois, lors des réflexions menées sur le projet SafeCity, cette notion s'est vue définie à plusieurs reprises :

- Dans le cadre de la surveillance d'une école primaire, l'analyse comportementale devait consister à identifier les comportements à l'intérieur d'une zone sécurisée afin de déceler d'éventuels problèmes.
- Une fois ce cas d'usage écarté, l'analyse comportementale devait constituer l'élément clé permettant de sécuriser l'école au travers d'une surveillance des environs de cette dernière.

En admettant ainsi que l'analyse comportementale repose sur la surveillance des extérieurs d'un lieu, il est alors possible d'adopter plusieurs stratégies :

- Analyser de manière globale les extérieurs de l'école.
- Analyser de manière spécifique le comportement des individus extérieurs de l'école.

L'objectif de la première stratégie est de pouvoir analyser rapidement l'extérieur de l'école afin de déceler des comportements potentiellement anormaux. Cette stratégie doit être rapide, efficace et permettre des déductions logiques face à une situation concrète telle qu'un humain pourrait le faire. L'objectif de la seconde stratégie est de pousser l'analyse plus en profondeur afin de garantir une plus grande fiabilité dans les résultats. Cette stratégie sera alors plus coûteuse en calcul, et plus complexe en élaboration. Ces deux stratégies seront abordées ici. En effet, la première pose les bases de l'analyse comportementale proposée dans cette thèse et la seconde permet d'affiner un des axes d'amélioration pouvant être suivis afin de perfectionner l'analyse comportementale proposée.

L'objectif de ce chapitre est donc de permettre une identification d'un comportement à risque de manière globale au travers de l'utilisation de différentes méthodes. L'étude de comportement anormal appliquée à la vidéosurveillance est un sujet vaste qui peut être effectué de plusieurs manières en fonction du comportement que l'on souhaite détecter [MZ18]. L'ensemble des réflexions menées dans ce chapitre a donc pour but de répondre à la demande de détecter les comportements préalablement définis à savoir :

- La détection de dispersion,
- la détection de rassemblement,
- la détection de maraudage.

La stratégie abordée dans ce chapitre consiste ainsi à étudier une scène dans sa globalité afin d'en extraire les informations significatives. Pour ce faire, il a d'abord fallu identifier au travers des détections que nous souhaitons effectuer les différentes technologies permettant d'atteindre cet objectif.

En ce qui concerne la détection de dispersion, définissons d'abord ce que signifie une dispersion dans notre cas d'étude : une dispersion se définit par l'apparition significative d'une distance entre des personnes formant initialement un groupe. L'apparition de cette distance peut se faire de manière rapide et spontanée ou peut se faire de manière plus lente et éparse. Une fois cette définition établie il apparaît alors qu'une dispersion peut être détectée si nous sommes capables d'identifier dans un ensemble de points les points pouvant former un groupe et si de fait nous sommes capables de calculer une distance caractérisant l'écart entre

chacun. Dans ce cas, une distance significative correspondra à la distance à partir de laquelle nous pouvons considérer que l'individu ne fait plus partie du groupe. Cette distance devra être établie au travers des différents cas d'études que nous rencontrerons dans la suite de ce chapitre.

En ce qui concerne la détection de rassemblement, définissons également ce que signifie un rassemblement dans notre cas d'étude : un rassemblement se définit par la présence d'individus dans une zone de l'écran pouvant être assimilée à un groupe. L'apparition de ce groupe peut se faire rapidement ou alors progressivement dans le temps. Cette définition met alors en avant le fait qu'un rassemblement doit être détecté si nous décelons la présence de plusieurs individus ayant une tendance à se rapprocher dans une même zone de l'écran. Un élément supplémentaire à prendre en compte dans cette détection est la notion d'heure. En effet, le cas d'étude étant appliqué aux alentours d'une école, il est normal qu'aux heures d'accueil des élèves des groupes se forment. De la même manière, il sera anormal de détecter un rassemblement en dehors de ces heures d'accueil et plus particulièrement dans les heures sensibles à savoir lorsque l'établissement est fermé ou lorsque les enfants sont dans la cour de récréation par exemple. Dans le cas de la détection de rassemblement, un groupe sera caractérisé par la présence de plusieurs individus se trouvant à une distance spécifique les uns des autres. Nous verrons dans la suite de ce chapitre que cette distance n'est pas le seul paramètre qu'il faudra définir afin de révéler l'appartenance d'un individu à un groupe.

Enfin en ce qui concerne la détection de maraudage, nous devons établir la définition d'un maraudage dans notre cas d'étude : un maraudage se définit par la présence d'un individu dans l'image de manière récurrente. La présence de cet individu peut être localisée dans une zone de l'image ou aux alentours d'une zone. Suite à des discussions autour du respect du RGPD, nous n'avons pas eu la possibilité de sauvegarder des images afin d'identifier formellement une personne, ce cas d'étude se limitera à l'étude d'un individu ne quittant pas le champ de la zone surveillée. La détection de maraudage devra également se faire en dehors des heures d'accueil des élèves et plus particulièrement dans les heures d'ouverture de l'établissement.



La définition des différents besoins permet de mettre en avant 3 axes de recherche :

- Le regroupement de points,
- L'analyse d'écart,
- Les heatmaps.

Ces 3 axes contribuent à différents degrés à la détection de l'ensemble des comportements préalablement définis :

- Le regroupement de points interviendra dans la détection du rassemblement et de la dispersion en permettant la validation de la présence ou non d'un groupe dans l'image,
- L'analyse d'écart interviendra dans le cadre du rassemblement et de la dispersion afin de détecter lesdits phénomènes,
- Les heatmaps interviendront dans le cadre de la détection de maraudage en permettant à moindre coût de détecter les zones de l'image dans lesquelles il y a le plus d'activités humaines.

La suite de ce chapitre met donc en avant les différentes recherches menées autour de ces 3 axes ainsi que la manière dont elles ont été utilisées afin de répondre à nos besoins.

## 3.2 Le regroupement de points

### 3.2.1 Introduction

Le regroupement de points est une technique devenue indispensable à de nombreux domaines tels que la représentation 3D [SLL11; Zha+18] ou encore dans le domaine de la classification de données[Zha+16]. Ces méthodes permettent d'identifier parmi une quantité importante de données celles qui sont proches par leurs valeurs ou leurs caractéristiques. Si nous considérons que chaque donnée d'un ensemble est identifiable par ses caractéristiques, cette donnée peut alors être d'une multitude de types. Si nous prenons l'exemple de notre cas, une donnée est un point identifiable par une position en abscisse et ordonnée à un temps spécifique. Toutefois, cette donnée peut être définie par d'autres éléments, nous pouvons par exemple imaginer qu'une donnée soit représentative d'un individu identifiable par la taille de la personne et son poids. Dans les deux cas, nous sommes

alors capables d’interpréter ces données au travers d’un graphique où seront utilisés en abscisse et en ordonnée les éléments permettant leur identification.

En partant du postulat que nous disposons d’une vidéo (figure 3.1.a) dans laquelle chaque image la composant peut être analysée (figure 3.1.b), nous disposons alors, pour cette vidéo, d’une suite d’images sur laquelle des individus sont potentiellement présents (figure 3.1.c). Nous sommes donc en mesure d’identifier les zones de l’image dans lesquelles se trouvent les individus (figure 3.1.d) afin d’obtenir pour chaque image, la position en abscisses et en ordonnée ainsi qu’une taille en hauteur et en largeur de chaque individu comme nous l’avons expliqué en 2.2.8. Nous disposons alors, pour chaque image de la vidéo, de la localisation des individus et de la taille qu’ils occupent dans l’image. Étant donné que dans cette partie nous n’utiliserons pas la hauteur et la largeur des individus, nous pouvons alors considérer que nous disposons pour chaque image de la vidéo d’un ensemble de points où chaque point représente un individu. Nous utiliserons donc le terme de nuages de points afin de désigner l’ensemble des individus identifiés dans une image.

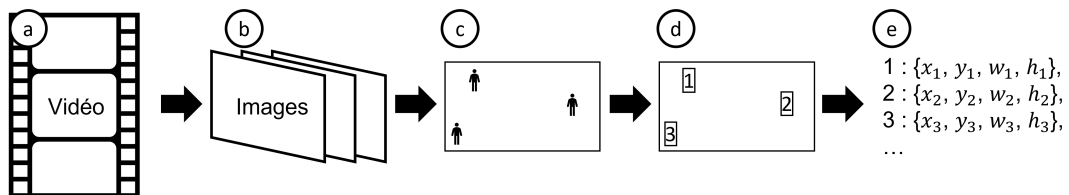


FIGURE 3.1 – Chronologie d’acquisition des données : (a) observation de la scène vidéo, (b) extraction des images, (c) analyse des images, (d) identification des individus dans les images et (e) extraction de la position et de la taille des individus au format JSON.

La problématique fondamentale de cette partie est de déterminer comment nous pouvons identifier les groupes d’individus présents dans l’image au travers de leurs positions dans un nuage de points. Étant donné que nos points sont caractérisés par des informations de position, les premières recherches ont consisté à établir quelles étaient les manières permettant de faire du regroupement dans un ensemble de données partageant les mêmes types d’information. De nos jours, cette problématique est commune, en effet, nous pouvons de plus en plus obser-

ver la présence d'intelligence artificielle ou de systèmes de prédiction dans la vie de tous les jours. Ces systèmes, s'ils sont basés sur l'apprentissage, auront besoin d'un accès important à des données qui leur serviront de référence. C'est de cette manière que ces systèmes sont entraînés et c'est la quantité de données utilisées pendant l'entraînement qui permettra d'obtenir un système fiable[Zhu2015]. Il est donc important d'une part de disposer de données en quantité suffisante, mais d'autre part il faut être assuré que ces données soient correctement interprétées. Nous pouvons par exemple regrouper ces données à l'aide de paramètres qui seraient proches, auquel cas il est intéressant d'utiliser des algorithmes de classification automatique.

### 3.2.2 Classification automatique

La classification automatique est un domaine vaste qui permet la catégorisation de donnée au travers de nombreux algorithmes. L'objectif de la classification automatique réside dans la possibilité de regrouper des données au travers de paramètres communs. C'est, par exemple, aujourd'hui un des buts que nous souhaitons atteindre en utilisant des réseaux de neurones[AA15]. Étant donné que nous cherchons à établir des groupes en nous basant sur la proximité des points dans un nuage de points, il est alors tout à fait envisageable de faire la corrélation entre ce type de classification et notre volonté d'identifier des groupes.

En observant les résultats obtenus dans le domaine, nous nous apercevons qu'il est tout à fait imaginable de transposer des méthodes de classification à notre situation dans la mesure où l'espace d'étude est un espace 2D dans lequel ont été positionnés des points. Chaque point est donc caractérisé par des valeurs spécifiques qui lui sont propres et qui peuvent s'approcher ou non des valeurs d'un autre point. En cherchant à établir des liens entre notre cas d'étude et la classification automatique, il est apparu que des méthodes de regroupement hiérarchique permettraient potentiellement d'atteindre notre objectif. Il existe deux grands domaines permettant de faire du regroupement hiérarchique :

- La classification descendante hiérarchique,
- La classification ascendante hiérarchique.

Dans le cas de la classification descendante, la stratégie est d'exclure d'un groupe initial toutes les données qui en sont trop éloignées afin de ne plus se focali-

ser que sur le groupe de données les plus proches de ce que nous souhaitons conserver. Cette méthode est relativement utile dans le cadre d'un filtrage où nous souhaitons identifier un groupe spécifique dans un ensemble de données [SF09; Nak+17]. Dans le cas d'une classification ascendante, la stratégie est de partir d'un ensemble anonyme et de classer ces éléments anonymes en décelant des similitudes dans leurs paramètres [VMS06; NNK19]. Dans ce cas, nous avons une classification permettant l'apparition de plusieurs groupes d'individus partageant des paramètres communs ou proches, mais où ces paramètres diffèrent d'un groupe à l'autre.

Étant donné que nous disposons d'un nuage de points représentant les différents individus, le postulat de départ est de considérer que les images traitées présenteront parfois des personnes appartenant à un même groupe. Notre objectif serait d'identifier la présence d'un ou plusieurs groupes dans le nuage de points. Une classification descendante hiérarchique ne paraît donc pas idéale sachant qu'elle ne nous permettrait d'identifier qu'un seul groupe à la fois dans le nuage de points. Nous nous orienterons donc vers une classification ascendante.

Il existe dans la littérature de nombreuses méthodes permettant une classification ascendante hiérarchique, par exemple :

- DBSCAN [Den20]
- K-Means [ASI20]
- Mean-shift [Car15]
- Spectral clustering [Lux07]

Afin de présélectionner certaines méthodes, il a fallu identifier au travers des paramètres qu'elles utilisent pour fonctionner, ceux qui sont exploitables dans le contexte de notre étude. Dans le cas de K-Means et de Spectral Clustering, les méthodes ont besoin de connaître à l'avance le nombre de groupes que nous souhaitons identifier. Dans le cas de Mean-shift, nous avons besoin de connaître la densité limite permettant d'établir qu'il s'agit d'un groupe. Enfin dans le cas de DBSCAN, il est nécessaire d'indiquer la distance maximum entre 2 individus pour que ces derniers soient considérés comme faisant potentiellement partie du même groupe.

Ces méthodes font partie d'un ensemble de méthodes permettant d'effectuer du regroupement. Parmi l'ensemble des méthodes étudiées, elles ont été retenues, car elles présentaient un intérêt au travers des paramètres nécessaires à leur fonctionnement. En effet, il était nécessaire de disposer de méthodes pouvant fonctionner à partir de nuages de points issus des caméras de vidéosurveillance. Nous devons donc identifier des méthodes utilisant la position des points dans un nuage ou se basant sur toute donnée en étant issues directement, comme la distance moyenne entre deux points par exemple.

Si les méthodes utilisées nécessitent pour fonctionner de connaître le nombre de groupes que nous souhaitons identifier, nous pouvons alors imaginer une solution qui analyse le nuage de points pour un nombre de groupes donné et qui, si nécessaire, réanalyse le nuage de points si le nombre de groupes donné ne permet pas d'obtenir un résultat cohérent. Cette méthode est toutefois hasardeuse et peut potentiellement nécessiter de nombreux calculs afin d'établir un résultat stable.

Dans le cas du Spectral clustering, nous nous basons sur la notion de densité afin de déterminer dans un ensemble de points la fin d'un groupe par rapport à la dispersion des membres se trouvant aux extrémités de ce groupe. C'est au travers de l'étude de cette méthode qu'il a été observé qu'une méthode ne nécessitant pas de connaître à l'avance le nombre de groupes à identifier était fonctionnelle et envisageable. La recherche des différents algorithmes disponibles permettant d'identifier les groupes au travers de la proximité des membres qui le composent a permis de mettre en avant l'algorithme DBSCAN.

Parmi les méthodes citées précédemment, deux ont particulièrement été étudiées K-Means et DBSCAN. Le premier algorithme de regroupement à avoir été testé est K-Means. K-Means est sûrement l'algorithme de regroupement le plus utilisé et le plus connu aujourd'hui. Cet algorithme est notamment utilisé en Data Science et en Machine Learning afin de disposer d'une classification des données efficace. L'objectif dans ces domaines ressemble sur plusieurs critères à ce que nous cherchons à faire. En effet, en Data Science et en Machine Learning, l'algorithme peut être confronté à un ensemble de points dans un espace orthonormé qu'il va devoir regrouper en fonction de leur emplacement dans un espace où ce

dernier est défini par des valeurs caractéristiques[Zha+22; Naw+22].

### 3.2.3 Expérimentation préliminaire

Nous avons donc essayé d'appliquer l'algorithme de regroupement K-Means à nos nuages de points où le nombre de groupes était préalablement connu. Le principe de l'algorithme K-Means est le suivant :

- Nous définissons un nombre  $n$  de points qui représenteront la position du barycentre des points du groupe à former. Nous nommerons ces points les « points centraux »
- Nous attribuons des valeurs aléatoires à la position des « points centraux ». Leurs positions doivent toutefois rester proches des points que nous souhaitons regrouper (figure 3.2.a).
- Nous associons aux « points centraux » tous les points qui leur sont le plus proches (figure 3.2.b).
- Nous repositionnons les « points centraux » en calculant la position moyenne des points qui leur ont été associés (figure 3.2.c).
- Nous recommençons les étapes 3 et 4 jusqu'à ce qu'un équilibre se fasse ce qui révélera alors la position finale des « points centraux », mais surtout les groupes auxquels appartient chaque point (figure 3.2.d).

Cette méthode permet d'obtenir des résultats satisfaisants dans le domaine du regroupement[Nal+11], mais présente un problème dans notre cas. En effet, un des éléments primordiaux de cette méthode est de définir à l'avance le nombre de « points centraux » que nous souhaitons trouver. Or dans notre cas d'étude, cette valeur est inconnue, car elle devra être idéalement trouvée lors de l'analyse. De plus, il s'avère que cette méthode ne permet pas d'avoir un résultat reproductible dans la mesure où le positionnement aléatoire de départ des points centraux peut révéler un résultat différent pour un même nuage de point.

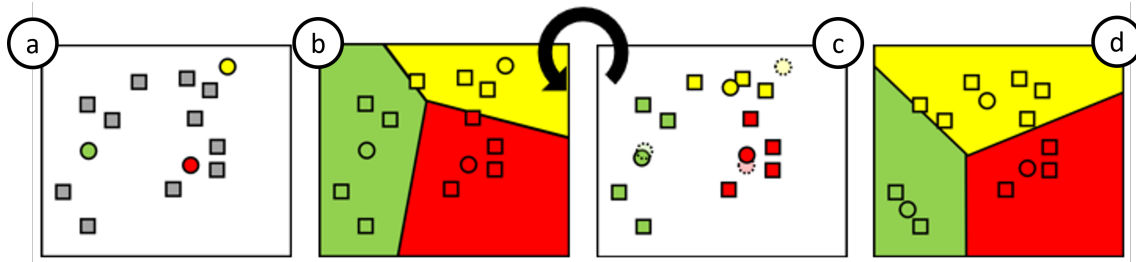


FIGURE 3.2 – Principe de fonctionnement de K-Means : (a) positionnement des points centraux, (b) association des points à grouper aux points centraux, (c) calcul de nouvelles positions des points centraux et (d) obtention d'un résultat stable.

Cette méthode n'a toutefois pas été complètement écartée, car en utilisant d'autres algorithmes de regroupement nous serions potentiellement capables de définir le nombre de groupe d'individus dans le nuage de point. En combinant alors ces méthodes, il serait alors possible d'avoir d'une part un précomptage permettant d'indiquer le nombre de groupes présents dans le nuage de points et d'autre part une confirmation au travers de l'algorithme K-Means par exemple si aucun de nos autres algorithmes n'est suffisamment autonome pour faire lui-même cette déduction.

### 3.2.4 Amélioration de la méthode

Une méthode de regroupement qui ne nécessite pas d'avoir la connaissance du nombre de groupe à former avant de l'utiliser serait dans notre cas DBSCAN. La méthode DBSCAN se base sur deux paramètres afin de former les groupes : la distance entre les points sous forme de rayon et le nombre minimum de points devant se trouver dans ce rayon. Cette méthode se base donc sur une notion de densité, si le nombre d'individus composant le groupe est insuffisant ce dernier n'est alors pas détecté comme un groupe. Étant donné qu'une dispersion ou un regroupement ne peut avoir lieu que si au moins 2 personnes sont présentes dans l'image, nous avons donc adapté cette méthode afin de considérer que le nombre minimum d'individus pouvaient être de deux afin de répondre le plus largement possible à notre problématique de formation de groupe. L'algorithme une fois adapté, fonctionne alors de la sorte :

- Un nombre  $n$  de points a été obtenu à l'issue de la détection (figure 3.3.a).

- pour chaque point, nous vérifions le nombre d'autres points entrant dans son rayon (figure 3.3.b).
- si un ou plusieurs points entrent dans le rayon d'un même point et réciproquement, cela signifie que les points appartiennent au même groupe (figure 3.3.c).

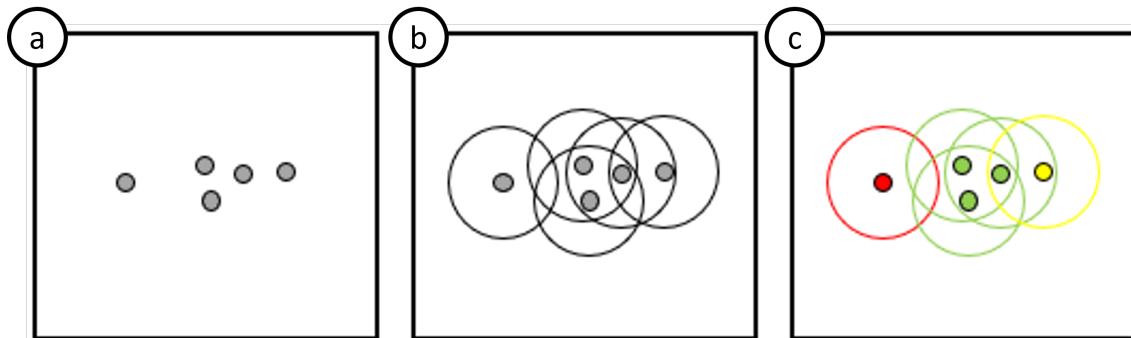


FIGURE 3.3 – Principe de fonctionnement de DBSCAN : (a) points anonymes positionnés dans un espace en deux dimensions, (b) détection des points positionnés dans un certain rayon autour de chaque point et (c) définition de l'appartenance des points à un groupe.

Nous pouvons également ajouter une notion de certitude en adaptant l'étape 3 de cet algorithme. Étant donné que chaque point est analysé afin de déterminer tous les autres points à l'intérieur de son rayon de détection, nous pouvons attribuer une appartenance plus ou moins forte à un groupe pour chaque point. Cette notion d'appartenance permet de mettre en avant que l'appartenance à un groupe va dépendre du cas d'usage dans lequel nous nous trouvons. Étant donné que dans notre cas chaque point est la représentation d'un individu, il y a énormément de variables qui peuvent jouer sur la présence d'une personne à l'endroit où elle a été détectée. Une personne peut se trouver à proximité d'un autre groupe d'individus sans que cela signifie qu'elles appartiennent à ce groupe. L'appartenance au groupe que nous cherchons à détecter dans notre cas est une appartenance sociale à un groupe ce qui ne se traduit pas uniquement par des notions de distance. Le contexte ainsi que la scène observée peuvent restreindre physiquement l'accès à certaines zones ce qui impose à des individus qui ne se connaissent pas de rester proches sans que cela implique qu'ils forment un groupe. La notion d'appartenance abordée précédemment permet donc de faire la distinction entre des



groupes formés par choix et ceux formés par contrainte. Cette déduction peut se faire de manière assez simple en se basant sur les constats suivants :

- Le point possède dans son rayon de détection plusieurs autres points, il a donc une appartenance forte avec les membres de son groupe.
- Le point possède dans son rayon de détection un seul autre point, il appartient donc au même groupe que ce point, mais son appartenance sera considérée comme faible, car il pourrait s'agir d'une personne indépendante au groupe qui se trouve juste à proximité de ces derniers.
- Le point ne dispose d'aucun autre point dans son rayon de détection, dans ce cas le point peut être considéré comme seul et n'appartenant à aucun groupe.

En effet, si un point possède dans son rayon plusieurs points cela ne veut toutefois pas dire que ces autres points possèdent plusieurs points dans leur rayon. Un point peut alors appartenir à un groupe, car il est dans le rayon d'un autre point, mais peut ne pas être considéré comme faisant partie du groupe au sens social du terme, car il ne révèle que très peu d'affinité avec ce dernier en étant par exemple à proximité d'un des membres du groupe, mais relativement éloigné des autres membres du groupe.

### **3.2.5 Conclusion des expérimentations**

L'utilisation de DBSCAN permet d'effectuer des regroupements de points cohérents permettant une identification des différents groupes présents dans la scène. Nous disposons ainsi d'un système de regroupement de points pouvant servir de base à une analyse plus poussée. En effet, le regroupement de points seul ne permet pas d'établir des anomalies dans le comportement. Afin de percevoir une anomalie il faut qu'en plus de la présence des différents groupes que nous soyons capables d'identifier quelle est la tendance de l'évolution des individus dans ces groupes. Pour ce faire, la méthode employée consiste à comparer les paramètres statiques disponibles lors de la détection tout au long du temps. L'analyse la plus simple et la plus évidente que nous pouvons mettre en œuvre est de comparer la distance entre les individus d'un même groupe ou encore entre les groupes eux-mêmes.

## 3.3 Analyse d'écart

### 3.3.1 Introduction

Dans le cadre des différents scénarios qu'il était nécessaire d'identifier, l'analyse d'écart est une analyse qui a semblé évidente du fait de sa simplicité de mise en œuvre et son faible coût en calcul. L'objectif de cette méthode est d'apporter des informations supplémentaires sur la formation ou la dispersion des différents groupes. En effet, la présence ou non d'un groupe implique que ce groupe a été formé ou dispersé à un moment de l'observation. En ajoutant une analyse de la formation ou de la dispersion de ces groupes, nous sommes alors capables d'établir si la vitesse à laquelle ces groupes se sont formés ou dispersés est cohérente. Elles seront considérées comme cohérentes si elles ne sont pas anormales et inversement.

Nous pouvons alors imaginer appliquer l'analyse d'écart de 2 manières :

- Une analyse d'écart globale
- Une analyse d'écart par secteur

Dans le cas de l'analyse d'écart "global", l'objectif est d'identifier à partir de la position de chaque point dans le nuage quel est l'écart "global" de l'ensemble de ces points et d'observer son évolution dans le temps. L'écart "global" serait dans ce cas une valeur significative telle que la distance moyenne ou encore l'écart type entre les individus. La mise à disposition de cette information permettra quant à elle de déceler si la variation dans l'écart révèle un comportement normal ou anormal. Cette étude permet à moindre coût de disposer d'une information sur un mouvement de foule par exemple.

Dans le cas de l'analyse d'écart par secteur, l'objectif est d'analyser la variation de l'écart au sein du groupe afin de déceler un comportement anormal. Dans ce contexte, il est nécessaire d'établir des zones d'analyse dans lesquelles l'analyse d'écart sera effectuée. Afin de définir ces différentes zones, nous pouvons utiliser les méthodes de regroupement expliquées précédemment.

### 3.3.2 Cas d'études

Une analyse très basique du comportement de groupe consiste à identifier les évolutions de l'écart type. L'écart type étant la mesure de la dispersion des valeurs, il est représentatif de l'écart entre des individus d'un groupe à un instant  $t$ . Les images étant traitées de sorte que tous les individus présents dans l'image soient identifiés, nous pouvons alors calculer l'écart type entre ces individus de chaque image. Nous obtenons alors, pour chaque image, une valeur représentative de l'écart global entre les individus présents dans la zone observée. Nous obtenons alors 3 possibilités de scénarios :

- L'écart type augmente,
- l'écart type diminue,
- l'écart type reste stable ou n'évolue pas.

Le cas de l'augmentation ou de la diminution de l'écart type permettrait d'identifier logiquement la présence d'une dispersion en cas d'augmentation de ce dernier ou d'un rassemblement en cas de diminution. Dans le cas où l'écart type reste stable ou s'il est nul, cela indique qu'il n'y a pas d'individu observé ou qu'il n'y en a qu'un seul. Nous pouvons aussi avoir un écart type stable lorsque le peu d'individus présents dans l'image reste à une position relativement statique. Ce phénomène peut arriver par exemple, lorsque deux personnes sont assises dans l'image et ne bougent pas pendant un certain temps ou lorsque deux ou plusieurs personnes discutent sans se déplacer.

### 3.3.3 Cas particuliers

Il existe toutefois des cas particuliers dans lesquels l'écart type peut augmenter de manière significative sans que cela représente une dispersion et a contrario l'écart type peut diminuer de façon très significative sans que cela signifie qu'il y ait eu un rassemblement. En ce qui concerne le premier cas, nous pouvons imaginer le schéma en figure 3.4 :

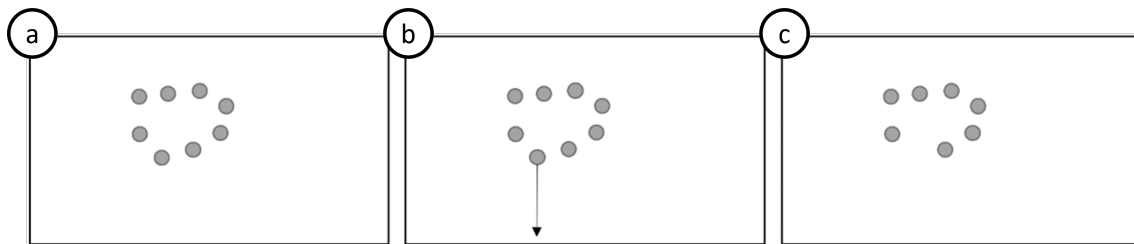


FIGURE 3.4 – Cas particulier : (a) un groupe est déjà formé, (b) un membre du groupe commence à s'en éloigner jusqu'à (c) sortir de l'image.

Une personne faisant partie d'un groupe déjà formé (figure 3.4.a) le quitte et se déplace jusqu'à sortir de l'image (figure 3.4.b). Durant le temps où cette personne s'écarte du groupe, l'écart type augmente ce qui laisse présager une dispersion. Toutefois dès lors que la personne a quitté l'image (figure 3.4.c), l'écart type se réduit de nouveau de manière rapide ce qui pourrait faire croire à un rassemblement. Dans le second cas, nous pouvons imaginer le schéma en figure 3.5 :

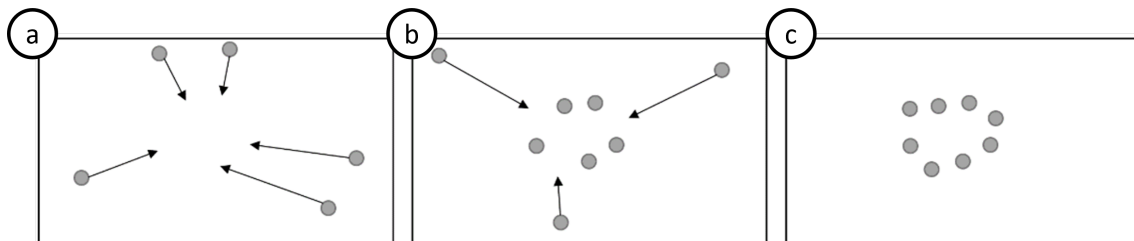


FIGURE 3.5 – Cas particulier : (a) un groupe se forme, (b) d'autres personnes rentrent dans l'image en direction du groupe jusqu'à (c) former un autre groupe.

Un premier groupe d'individus se forme (figure 3.5.a), ce qui permettra de détecter un rassemblement. Si ce groupe d'individus est déjà formé et qu'un autre nombre d'individus rentre dans l'image pour se greffer aux groupes déjà formés nous observerons d'abord une dispersion (figure 3.5.b). En effet, avant l'entrée dans l'image des nouveaux individus, l'écart type a une certaine valeur qui va forcément augmenter dès lors qu'une ou plusieurs personnes entrent dans l'image. Une fois la dispersion détectée et les valeurs de nouveau stabilisées, les personnes se rapprochant du groupe vont provoquer la détection d'un rassemblement (figure 3.5.c). Ce qui, à ce moment-là du scénario, correspond bien à ce qu'il fallait

détecter.

### 3.3.4 Gestion des cas particuliers

La solution mise en œuvre pour corriger ces faux positifs est une étude non pas de l'écart type, mais de la variation de la tendance de l'écart type. Prenons le tableau 3.1 suivant de valeur d'écart type donné pour 18 images :

Image	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Écart	30	32	32	29	31	29	30	35	41	48	54	60	69	76	36	36	37	36

TABLE 3.1: Évolution de l'écart type sur une suite de 18 images issues d'une vidéo.

Dans les premières versions de l'analyseur, un tel relevé de valeur va provoquer la détection d'une dispersion à partir de l'image 8 et ce jusqu'à l'image 14. L'analyseur détecterait ensuite un rassemblement à partir de l'image 15, car la variation de l'écart type est très importante et va en diminuant. Pour régler ce problème, la solution qui a été réfléchiée et mise en œuvre consiste à stocker l'évolution de la valeur de l'écart type sans considérer l'importance de variation entre 2 valeurs. Pour ce faire, nous créons un tableau dans lequel seront stockés  $X$  éléments où  $X$  représente le nombre d'éléments souhaité pour adoucir la détection. Ce nombre pourra évoluer en fonction de l'environnement observé. Ce principe repose sur une fenêtre glissante qui permet de conserver les  $X$  dernières valeurs permettant ainsi de s'assurer une continuité dans le flux de données qui est analysé[Dat+12]. En utilisant une fenêtre glissante, nous nous assurons aussi que le traitement des valeurs n'intégrera pas de valeurs aberrantes supplémentaires, car il s'agira forcément du traitement de  $X$  valeurs issu du traitement de  $X$  images se suivant dans la vidéo. Dans notre cas, nous avons choisi d'utiliser 10 éléments, car après plusieurs tests il s'avère que l'angle de vue de notre caméra et la distance à laquelle se trouvait la scène ne nécessitent pas de stocker plus d'éléments. Ces 10 éléments pourront contenir 3 valeurs possibles : -1, 0 et 1. Les éléments représentent la variation qu'il y a entre la valeur d'écart type de l'image  $n-1$  et de l'image  $n$ . Si cette variation est positive, nous stockons la valeur 1, si la variation est négative nous stockons la valeur -1 et s'il n'y a pas eu de variation nous stockons la valeur 0.

Reprenons le tableau 3.1 en admettant cette fois qu'il y a eu 10 images avant l'image 1 où l'écart type n'a pas évolué et avait la valeur 30. Nous établissons pour chaque image la somme des valeurs du buffer. Pour ce faire, le tableau 3.2 met en avant l'évolution du buffer à mesure de la réception des écarts par image. À l'image 1, le buffer recevra dans la case B1 la valeur -1 si l'écart a diminué depuis la dernière image, 0 si l'écart n'a pas évolué depuis la dernière image et 1 si l'écart a augmenté depuis la dernière image. De même, les valeurs se déplaceront de leur position actuelle dans le buffer à la position suivante, ainsi la valeur de B1 passera en B2, celle de B2 en B3...

Image	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Écart	30	32	32	29	31	29	30	35	41	48	54	60	69	76	36	36	37	36
B1	0	1	0	-1	1	-1	1	1	1	1	1	1	1	1	-1	0	1	-1
B2	0	0	1	0	-1	1	-1	1	1	1	1	1	1	1	1	-1	0	1
B3	0	0	0	1	0	-1	1	-1	1	1	1	1	1	1	1	1	-1	0
B4	0	0	0	0	1	0	-1	1	-1	1	1	1	1	1	1	1	1	-1
B5	0	0	0	0	0	1	0	-1	1	-1	1	1	1	1	1	1	1	1
B6	0	0	0	0	0	0	1	0	-1	1	-1	1	1	1	1	1	1	1
B7	0	0	0	0	0	0	0	1	0	-1	1	-1	1	1	1	1	1	1
B8	0	0	0	0	0	0	0	0	1	0	-1	1	-1	1	1	1	1	1
B9	0	0	0	0	0	0	0	0	0	1	0	-1	1	-1	1	1	1	1
B10	0	0	0	0	0	0	0	0	0	0	1	0	-1	1	-1	1	1	1
Som.	0	1	1	0	1	0	1	2	3	4	5	5	6	8	6	7	7	5

TABLE 3.2: Utilisation d'un buffer de lissage de dix valeurs nommées de B1 à B10 et mise en avant de la somme des valeurs du buffer en ligne "Som.".

Faisons la somme de toutes les valeurs du tableau, nous obtenons alors pour chaque image une valeur entre  $-X$  et  $X$ . Les valeurs sont cette fois représentatives de la tendance globale de la variation de l'écart type. Si cette valeur est négative, la tendance indique un regroupement et si cette valeur est positive, la tendance indique une dispersion, mais nous pouvons encore aller plus loin. En effet, dans ce tableau nous remarquerons que les valeurs d'écart type dans les phases de stabilité enregistrent malgré tout des valeurs différentes de 0 dans le tableau. Or, une variation de l'écart type faible peut simplement indiquer que la détection des individus ne s'est juste pas faite au même endroit sur l'image à quelques pixels

près, mais n'indique pas que les personnes sont en train de se déplacer. Il faut alors intégrer la possibilité que la détection vidéo ne soit pas parfaite et qu'entre deux images la détection d'une personne puisse évoluer de quelques pixels. Pour ce faire, il faut également intégrer un seuil à partir duquel la variation de l'écart type doit être prise en compte entre 2 images. Reprenons le tableau précédent en fixant ce seuil à la valeur 4 :

Image	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Écart	30	32	32	29	31	29	30	35	41	48	54	60	69	76	36	36	37	36
B1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	-1	0	0	0
B2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	-1	0	0
B3	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	-1	0
B4	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	-1
B5	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
B6	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
B7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
B8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
B9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
B10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Som	0	0	0	0	0	0	0	1	2	3	4	5	6	7	6	6	6	5

TABLE 3.3: Utilisation d'un buffer de lissage de dix valeurs nommées de B1 à B10 et mise en avant de la somme des valeurs du buffer en ligne "Som.". L'intégration des valeurs dans le buffer ne se fait que si la variation a dépassé le seuil établi (ici 4).

Les sommes des buffers sont différentes :

Som. t1	0	1	1	0	1	0	1	2	3	4	5	5	6	8	6	7	7	5
Som. t2	0	0	0	0	0	0	0	1	2	3	4	5	6	7	6	6	6	5

TABLE 3.4: Comparaison de la somme des buffers avec (Som. t2) et sans seuil d'intégration (Som. t1).

La détection d'une dispersion peut quand même se faire, car la somme du buffer augmente, mais nous remarquons que dans les phases de stabilité le buffer est resté stable, il n'a pas enregistré d'évolution. Une fois le nombre de valeurs du buffer établi ainsi que la valeur de seuil permettant d'enregistrer les valeurs,

il est alors nécessaire d'établir le seuil de détection du comportement souhaité. Étant donné que cette valeur doit être comprise entre 0 et  $X$ , nous pouvons fixer les valeurs de seuil à  $X/2$  qui représente une vitesse de détection moyenne par rapport aux possibilités que nous avons. Si la valeur est proche de 0 la détection sera plus rapide et si elle s'en éloigne, elle sera plus lente. Nous pouvons faire varier cette valeur en fonction de la vitesse à laquelle nous souhaitons effectuer les détections. Le fait de fixer cette valeur à la moitié de la taille du buffer permet une détection à 5 images du moment de l'événement. En analysant 15 images par seconde, nous pourrions bénéficier d'une détection en 0,33 seconde. Les valeurs obtenues ici sont alors dépendantes de plusieurs critères notamment la distance et le nombre d'images traitables par seconde. Il est difficile d'établir un lien entre ces paramètres, car ils dépendent à la fois de la puissance de calcul disponible, du type de caméra utilisé, de l'angle et de la distance de la caméra qui observent la scène. Les valeurs ont donc été fixées dans le contexte des expérimentations pour obtenir de bons résultats, mais devront être ajustées en fonction de la situation pour obtenir des résultats cohérents.

### 3.3.5 Résultats et discussions

Les observations, suite aux résultats des différentes analyses, montrent que la détection se fait sans aucun problème dans le cas d'un des deux scénarios parfaits à savoir

- Tous les individus entrent en même temps dans l'image et se regroupent en un même point.
- Un groupe déjà formé se disperse de manière brutale.

Et dans les scénarios issus de la réalité (ceux qui sont issus de vraies vidéos), nous obtenons un succès dès lors que :

- Le buffer contient suffisamment d'éléments,
- le seuil à partir duquel les valeurs sont prises en compte et enregistrées dans le buffer est correctement calibré,
- le seuil de détection établi par rapport au nombre d'éléments que peut contenir le buffer est à la bonne valeur.

Nous avons donc dans cette expérimentation trois variables, dépendantes de l'environnement, qui doivent être ajustées en fonction des situations :

- La taille du buffer : le nombre d'éléments stockés dans le buffer où chaque



élément représente la comparaison entre la valeur de l'écart type de l'image n-1 et n.

- La sensibilité à la variation : la valeur minimum qu'il doit y avoir entre la valeur de l'écart type à l'image n-1 et n pour considérer qu'il y a eu une réelle évolution.
- Les seuils de détection : les seuils à partir desquels nous considérons que la détection peut avoir lieu. Par défaut, ce seuil dans le cas de la dispersion vaut :  $\text{taille du buffer} / 2$  et dans le cas de rassemblement vaut :  $-(\text{taille du buffer} / 2)$

Cette phase permet de valider la détection de comportement de rassemblement et de dispersion dans les situations particulières présenté précédemment à savoir lors de l'arrivée d'autres personnes dans la scène ou la présence d'autres individus extérieurs à la situation.

### 3.3.6 Amélioration de l'analyse

La détection peut encore être améliorée en faisant une analyse par secteur de la zone surveiller. En analysant non pas l'écart type global, mais l'écart type propre à chaque groupe identifié dans l'image. Cette identification peut se faire à l'aide des algorithmes de regroupement expliqué précédemment. En procédant de la sorte, un groupe peut se disperser pendant qu'un autre se rassemble sans que cela ne pose de problème. L'analyse est également beaucoup moins sensible à l'apparition d'un individu qui n'a rien à voir avec le groupe en train de se disperser.

Cette version de la solution en complément des méthodes de regroupement permet de déterminer la présence ou non d'un phénomène anormal. Toutefois, sa sensibilité fait de cette solution un outil qui ne serait pas adapté à l'analyse de plusieurs scénarios en simultané. Un axe d'étude a donc été mené afin de déterminer de quelle manière il était envisageable de créer des secteurs à analyser.

Un découpage par secteur d'une zone d'étude permettrait de limiter les calculs à une zone spécifique et d'avoir moins de calcul à réaliser. Un découpage permet de se focaliser sur le secteur présentant un intérêt et donc de ne pas avoir à faire de calculs inutiles. Cette technique est notamment utilisée dans le cadre de l'apprentissage des réseaux neuronaux où il sera essentiel de créer un découpage

dans l'image de ce que nous souhaitons faire apprendre au réseau[MB15]. C'est en nous basant sur cette réflexion que nous avons décidé de découper notre image en secteurs d'intérêt.

### 3.3.7 Division en secteur d'intérêt

Définissons d'abord ce qu'est un secteur d'intérêt. Dans notre cas, les études porteront sur la présence d'individus et leurs comportements. Nos secteurs d'intérêt sont donc les zones de l'image dans lesquels nous pouvons détecter un ou plusieurs individus. En ce qui concerne la détection de dispersion et de rassemblement, les secteurs d'intérêt seront ceux pour lesquels des groupes d'individus ont été identifiés. Étant donné que DBSCAN nous indique l'appartenance de chaque individu à un groupe comme le montre la figure 3.6, l'idéal serait d'utiliser un découpage autour des différents groupes identifiés. Après plusieurs recherches, il se trouve que les diagrammes de Voronoï [Aur91] peuvent le mieux répondre à notre besoin.

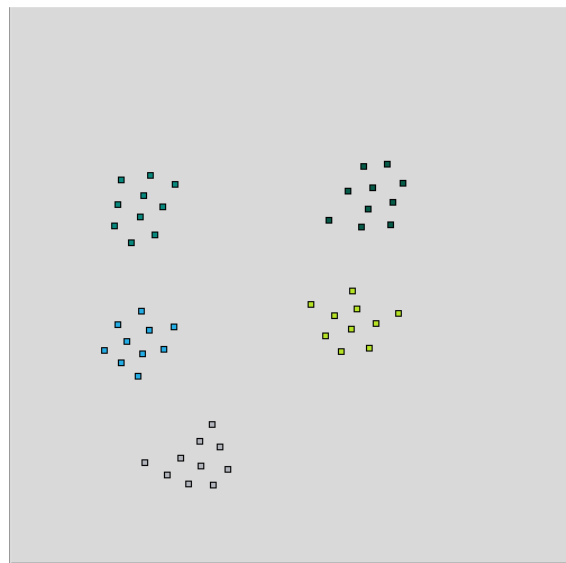


FIGURE 3.6 – Image provenant de l'analyseur ayant fait un regroupement de points en utilisant DBSCAN. Chaque couleur correspond à un groupe, ici il y a 5 groupes de 10 individus.

Toutefois, sans regroupement préalable ce genre de diagramme n'aide pas réellement au découpage. Si nous prenons l'exemple présenté en figure 3.6, il s'agit de



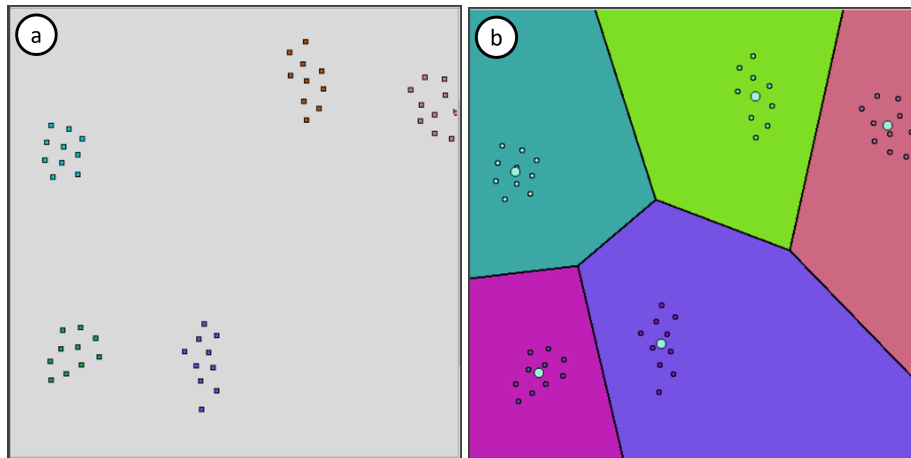


FIGURE 3.8 – Comparaison entre (a) La position des points envoyée par le simulateur où chaque couleur représente un groupe d'individus qui a été simulé et (b) le résultat obtenu après regroupement où chaque point représente un individu et chaque couleur d'individu représente son appartenance à un groupe. Chaque groupe est également dans une zone établie par les diagrammes de Voronoï où chaque couleur représente un espace associé à un groupe de points

### 3.3.8 Conclusion

À l'aide de ce découpage par zone nous disposons alors de sous-secteurs dans lesquels nous serons capables d'analyser de manière spécifique chaque groupe. Toutefois, plusieurs problèmes se posent :

- À quel moment doit-on recalculer chaque sous-secteur ?
- Les points identifiés comme formant un groupe dans une image sont anonymes, comment les réidentifier dans l'image suivante ?
- Les points étant anonymes, comment savoir quand un point passe d'un groupe à un autre ?

La mise en œuvre de cette solution permet de visuellement identifier les groupes dans l'image, mais pour une seule image. Si les groupes restent relativement statiques et ne cherchent pas à se déplacer sur l'ensemble de l'image, il est alors envisageable de faire une analyse par secteur de la même manière que lors de l'analyse globale. Malheureusement, chaque point représentant une personne, il est impossible de prévoir le comportement qui sera adopté. Cette stratégie reste toutefois intéressante, mais doit être complétée à l'aide d'autres technologies afin de pou-

voir identifier un groupe et en étudier son évolution. Nous devons être capables à tout moment du temps de pouvoir identifier chaque membre qui le compose. Il est donc nécessaire de disposer d'un système de suivi permettant à tout moment du temps de suivre chaque membre du groupe.

L'utilisation d'un système de suivi, même partiel, permettrait d'obtenir des informations sur le déplacement d'un point par rapport aux autres points. Dès lors, la sectorisation pourra être faite sur l'ensemble des groupes identifiés dans le nuage et si un point de ce nuage adopte un comportement particulier, il pourra être identifié et le découpage sera réajusté. En combinant la méthode présentée dans cette section avec un suivi, nous serions alors capables de renforcer la détection des dispersions et des rassemblements. Nous verrons dans la suite de cette thèse certaines des méthodes présentées 3.2 et 3.3 renforcées avec du suivi.

Deux des trois scénarios à identifier sont partiellement couverts à l'aide des deux méthodes précédentes. Toutefois, en ce qui concerne la détection de maraudage il a fallu trouver une méthode permettant d'identifier la présence récurrente d'un individu. Étant donné que l'identification des points ne permet pas d'identifier formellement une personne, de plus, ne disposant pas de système de suivi permettant d'identifier un point à tout moment du temps, nous avons dû imaginer une solution permettant de s'accommoder de ces contraintes en utilisant notamment les heatmaps.

## **3.4 Les heatmaps**

### **3.4.1 Introduction**

Le dernier type de détection à effectuer autour du projet SafeCity était la détection de maraudage. Comme il l'a été expliqué précédemment la détection d'un maraudage repose sur le fait qu'un individu reste à proximité du lieu surveillé dans des heures inadéquates ou de manière récurrente. Afin de procéder à cette détection, il a d'abord fallu considérer les contraintes auxquelles nous étions déjà soumis. En effet, la méthode la plus efficace pour détecter la présence récurrente d'un individu serait de pouvoir l'identifier. Étant donné que dans le cadre du pro-

jet il nous est impossible de procéder à l'identification formelle d'un individu au travers des vidéos, il est apparu autour de cette détection des problématiques liées au contexte du projet.

- Comment identifier un individu sur deux images différentes sans l'identifier formellement ?
- Comment faire la différence entre une personne malveillante et une personne sans mauvaises intentions ?

L'authentification est impossible dans la mesure où il ne nous est pas autorisé d'identifier une personne formellement dans une image. Ce faisant, nous ne sommes alors pas en mesure de comparer deux images afin de dire si la personne se trouvant en image 1 se trouve également en image 2. Il n'est toutefois pas exclu d'utiliser une méthode de suivi basé sur de l'identification d'un individu et non pas sur l'identification formelle d'une personne. Une autre problématique est de pouvoir détecter la présence d'un individu à proximité d'un lieu. Étant donné que nous disposons de caméras fixes, il a fallu imaginer une solution permettant de faire de cette contrainte un avantage. Des recherches dans la littérature ont mis en avant des méthodes permettant, au travers de caméras, de détecter les zones d'une image dans lesquelles il se passe le plus d'activités[Asa17; RWM11]. Cette méthode repose sur les heatmaps.

Le principe de base d'une heatmap est de représenter sur une matrice à deux dimensions, dans une palette de couleur spécifique, l'intensité d'une grandeur par case de la matrice. Cette technique est utilisée dans de nombreux domaines, il peut totalement être utilisé dans notre cas d'utilisation. En effet, une heatmap est une matrice permettant une représentation 2D d'un ensemble de valeurs. Étant donné que nous utilisons une caméra dans le cadre de nos analyses on peut totalement imaginer appliquer cette matrice sur l'ensemble de l'image de la caméra. Cette matrice représentée par deux axes peut par exemple correspondre à une zone géographique ou dans notre cas à l'image entière de la caméra avec : en abscisse la largeur de l'image en pixel et en ordonnée la hauteur de l'image en pixels. Cette matrice dont toutes les cases sont initialisées à la même valeur va donc pouvoir chauffer ou refroidir en fonction de la variation de la grandeur à analyser.

### 3.4.2 Principe de fonctionnement

Une heatmap, peut être remplie de plusieurs manières : elle peut être remplie a posteriori c'est-à-dire une fois que toutes les données ont été obtenues et elle peut être également remplie en direct, c'est-à-dire à mesure que les données lui parviennent. Le cas qui nous intéresse ici est le remplissage en direct qui devra être réalisé en parallèle de l'analyse de la vidéo. Dans le cas d'une heatmap qui se remplit en direct, les valeurs vont évoluer à l'aide de deux phénomènes qui s'opposent : la chauffe et le refroidissement.

### 3.4.3 Expérimentation

Une expérimentation a donc été menée afin de déterminer si en partant de la simple base de la vidéo et de l'extraction des individus s'y trouvant nous serions capables d'alimenter une heatmap de la présence des individus se trouvant dans la scène. La méthode employée dans notre cas a consisté à définir une matrice de même résolution que la caméra soit de 1920 colonnes par 1080 lignes. En ce qui concerne le remplissage de la matrice, il faut définir comment seront interprétés la chauffe et le refroidissement. Ces deux phénomènes peuvent être modulés selon quatre paramètres :

- L'intensité de chauffe
- l'intensité de refroidissement
- la fréquence de chauffe
- la fréquence de refroidissement

La valeur de ces paramètres dépendra de la donnée utilisée pour remplir la heatmap qui dans notre cas doit être une grandeur extractible de la vidéo. Comme il a été expliqué en 2.2.6 nous, disposons de caméras de vidéosurveillance de résolution 1920 par 1080 pixels. De plus, nous disposons pour chaque image d'une vidéo de la position en abscisse et en ordonnée de chaque individu s'y trouvant ainsi que de la hauteur et de la largeur du rectangle les recouvrant. Il est indispensable de bien comprendre ce que nous souhaitons utiliser comme valeur pour remplir la matrice.

Le postulat de départ de cette expérimentation est de considérer que la heatmap sera chauffée par la présence d'individus et refroidit par l'absence d'individu. De

ce postulat on comprend donc que :

- La fréquence de chauffe sera définie par la présence d'un individu dans la zone de la matrice correspondant à la zone de l'image dans laquelle se trouve l'individu. L'événement se produira donc pour chaque image analysée de la vidéo. Dans un scénario normal, la fréquence de chauffe correspond donc à la fréquence d'acquisition d'image de la caméra de vidéosurveillance.
- La fréquence de refroidissement sera également définie par la fréquence d'acquisition d'images de la caméra de vidéosurveillance, toutefois elle ne sera pas dépendante de la détection dans ses vidéos, en effet le refroidissement devra se faire de manière périodique, mais surtout récurrente.

En ce qui concerne les paramètres liés à l'intensité de la chauffe ou du refroidissement, la valeur est moins évidente à établir. En effet, l'intensité de la chauffe représente la valeur qui sera ajoutée aux cases de la matrice dans lesquelles un individu a été détecté. Elle s'oppose donc directement à l'intensité du refroidissement qui aura pour but de faire refroidir l'ensemble de la matrice de manière périodique. Prenons l'exemple en figure 3.9 :

- La matrice est initialement remplie d'une même valeur dans notre cas 0 (figure 3.9.a).
- Dès qu'un ou plusieurs individus sont détectés dans l'image, on reporte leurs présences dans une matrice temporaire en ajoutant l'intensité de chauffe dans les cases où ce dernier a été détecté (figure 3.9.b).
- La matrice initiale et la matrice temporaire sont ensuite additionnées (figure 3.9.c).
- La nouvelle matrice est alors refroidie. L'objectif du refroidissement dans ce cas est de diminuer toutes les valeurs qui ne sont pas à la valeur initiale de l'intensité de refroidissement (figure 3.9.d).

Une fois ce cycle terminé il se répète tant qu'il y a des images à analyser. Nous comprenons alors l'importance du choix des valeurs d'intensité de refroidissement et de chauffe, car si la valeur d'intensité de refroidissement est supérieure à l'intensité de la chauffe, la matrice aura plus tendance à refroidir qu'à chauffer ce qui empêchera d'obtenir une détection.



Dans l'exemple présenté en figure 3.9, notre objectif est de dépasser une valeur de seuil fixé à 3 (valeur utilisée à titre d'exemple, elle n'est pas représentative de la réalité). Cet objectif est alors atteint si dans l'image suivante des positions d'individu se superposent aux zones déjà chaudes de la matrice principale. Afin de déterminer quelles valeurs étaient les plus adéquates à une bonne détection, plusieurs tests ont été effectués.

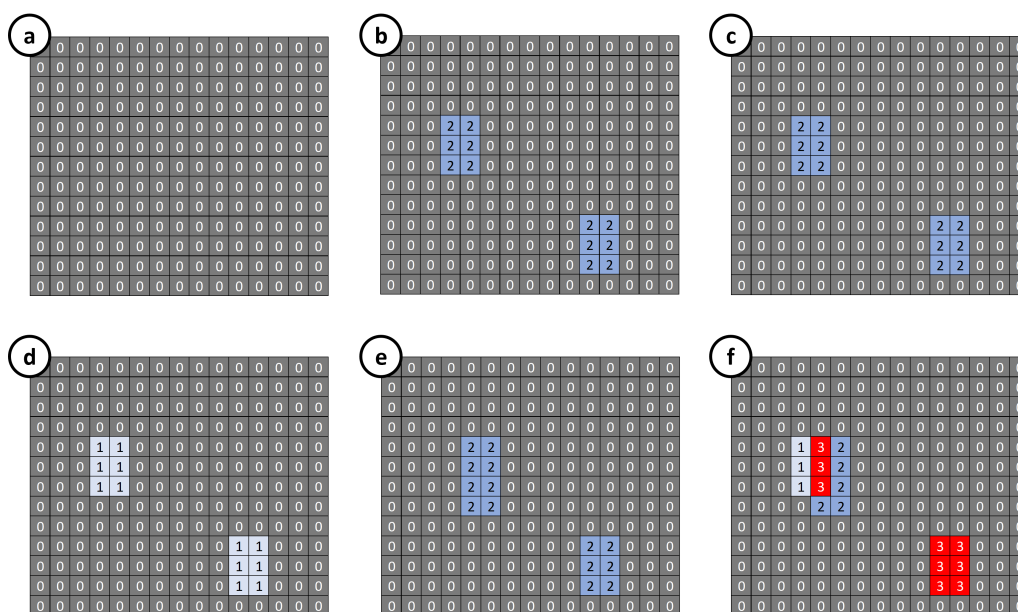


FIGURE 3.9 – Représentation visuelle de la matrice caractérisant la heatmap : (a) Matrice principale vierge. (b) Matrice issue de la détection vidéo. (c) Ajout de la matrice (b) en matrice (a). (d) Matrice principale refroidie. (e) Seconde matrice issue de la détection vidéo. (f) Ajout de la matrice (e) en matrice (d).

### 3.4.4 Résultat et discussions

Les expérimentations montrent que cette méthode n'est efficace que si l'individu reste au même endroit durant plusieurs secondes. Comme le montre la figure 3.10.b, l'analyseur dispose alors d'une retranscription de la scène observée sous forme de heatmaps dans laquelle les zones les plus chaudes représentent la présence d'un individu récurrent. Le temps à partir duquel la détection s'effectue est quant à lui ajustable en fonction du besoin. En revanche, si l'individu reste

dans la zone tout en se déplaçant de manière régulière la heatmap n'a alors pas le temps de chauffer suffisamment pour détecter l'individu. En effet, afin que la heatmap ne surchauffe pas, nous sommes obligés de la refroidir de manière régulière afin d'éviter les faux positifs. Il apparaît alors évident au vu des résultats que la méthode permet de répondre à un besoin immédiat de détection d'individus statiques dans la zone, mais ne permet pas de détecter un individu qui aurait connaissance de la faiblesse du système de détection et qui viendrait à se déplacer continuellement.

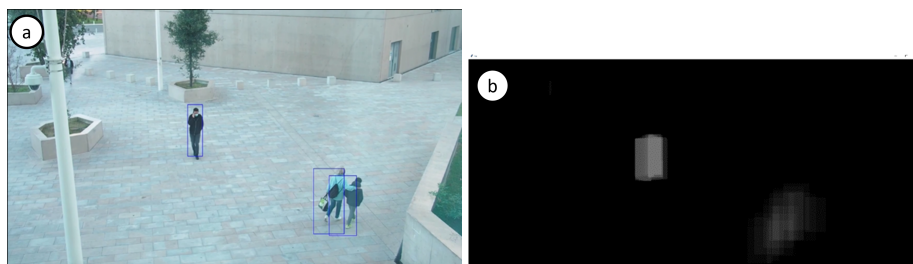


FIGURE 3.10 – Transposition de la réalité dans une heatmap : (a) Représentation d'une image issue de la vidéosurveillance sur laquelle ont été ajoutés les rectangles de détection vidéo et (b) représentation visuelle de la matrice servant de heatmap permettant la détection du maraudage où chaque zone claire représente la présence d'un individu.

Malgré cette faiblesse dans la méthode, les résultats sont suffisants pour répondre aux besoins initiaux de détection de maraudage statique. Cette méthode nécessite alors également une amélioration afin d'être pleinement fonctionnelle. En effet, comme pour les scénarios de dispersion et de rassemblement, l'utilisation d'un dispositif de suivi individualisé permettrait d'attribuer à l'individu détecté des paramètres supplémentaires :

- L'heure à laquelle l'individu est apparu dans l'image,
- Depuis combien de temps ce dernier y est présent,
- Quelle distance l'individu a parcourue dans l'image depuis qu'il a été détecté,
- À quelle vitesse se déplace-t-il.

### **3.4.5 Conclusion**

Cette dernière partie de l'analyse comportementale globale a permis de valider l'ensemble des demandes du projet SafeCity dont l'objectif était de prouver la faisabilité. En revanche, comme ont pu le montrer les différentes expérimentations, les méthodes ne sont pas optimales et ne permettent pas une grande finesse de détection. La suite des travaux abordés dans cette thèse aura pour but de compléter les différentes méthodes présentées dans ce chapitre afin d'améliorer la détection de comportements anormaux au travers de l'implémentation d'un suivi individualisé. Il est apparu dans ce chapitre que l'ensemble des comportements que l'on souhaite détecter pourrait être amélioré en ayant à disposition des informations permettant d'identifier les individus présents dans l'image. Une des principales directives de ce projet étant l'anonymisation des données, le prochain chapitre présentera les recherches menées afin d'effectuer un suivi individualisé anonymisées à partir des seules informations de position et de taille des individus présents dans la scène.

# SUIVI INDIVIDUALISÉ (ANALYSE INDIVIDUELLE)

---

## 4.1 Préambule

L'ensemble des travaux menés dans les deux chapitres précédents permettent de faire des déductions comportementales sur la scène observée en se basant sur des données statique et anonyme. L'ensemble de ces données est obtenu à partir de caméra de vidéosurveillance ou par la simulation. Nous avons pu observer dans le chapitre 2 de quel format sont ces données et de quelle manière nous pouvions les générer de sorte à nous rapprocher de la réalité. Nous nous apercevons que dans le format de données proposées il n'y est pas indiqué d'informations permettant d'identifier formellement un individu. Les seules informations dont nous disposons sont la position et la taille occupées par les individus détectés par le système vidéo.

Ces informations nous ont permis dans le chapitre 3 d'identifier par des calculs simples les comportements de dispersion, de rassemblement et de maraudage d'une scène. En revanche, nous nous apercevons que les méthodes ne permettront de faire des détections que très basiques. Dans le cas de la dispersion et du rassemblement, les seuls éléments réellement observables permettant de formuler la présence d'un de ces événements se basent uniquement sur une notion de distance entre les différents points ce qui dans certains cas peut poser problème. L'utilisation d'une seule notion de distance pour effectuer nos différentes détections implique notamment de ne pouvoir effectuer ces dernières que si la présence de groupe peut être avérée. En effet dans la mesure où il n'y a pas de groupe présent dans l'image, il devient alors difficile d'identifier les comportements de rassemblement et de dispersion . A contrario, le fait qu'il y ait peu d'individus dans

l'image facilite les détections de maraudage (détection qui devient plus difficile dans le cas où plusieurs personnes sont présentes dans l'image).

Ces problématiques mettent en évidence que la détection de comportement au travers de données anonymisées n'est peut-être pas la seule solution envisageable. L'objectif de ce chapitre est donc de mettre en avant l'utilisation de solutions permettant d'effectuer un suivi individualisé en utilisant les mêmes données anonymes. En effet, la mise en place d'un outil de ce type permettra de résoudre plusieurs problèmes rencontrés dans les chapitres précédents :

- En disposant d'un suivi individualisé, nous sommes en mesure d'identifier tout au long d'une séquence la présence d'un individu bien spécifique .
- Nous pouvons associer une vitesse et une trajectoire à une entité, ce qui permet de disposer d'information individuelle supplémentaire susceptible d'être analysée.

Les travaux effectués dans ce chapitre se présentent sous forme de 2 axes de réflexion :

- Comment pouvons-nous effectuer un suivi individualisé au travers des méthodes déjà employées et par conséquent de données anonymisées ?
- Est-il possible de prédire la trajectoire d'un individu au travers de données anonymisées ?

Afin de nous assurer de l'efficacité des méthodes abordées dans ce chapitre et de pouvoir les confronter, nous utiliserons un jeu de vidéos présentant différents scénarios. Parmi l'ensemble des vidéos tournées présentées en chapitre 2, nous avons sélectionné un ensemble de 7 vidéos présentant des caractéristiques différentes. Cette sélection a été établie de sorte à présenter des vidéos réalistes ne présentant donc pas toujours des trajectoires idéales. Les critères de sélection qui ont permis la sélection des vidéos sont donc :

- Le nombre de fois ou un individu passe derrière un obstacle
- Le nombre de fois ou des individus se croisent
- Le nombre d'individus dans la vidéo
- La durée de la vidéo

Étant donné que les vidéos que nous utiliserons sont déjà traités par le système de détection d'individus nous utiliserons aussi comme autre critère le nombre de fois

où des individus ont été fusionnés . En effet, comme il a été expliqué en chapitre 1, les algorithmes de détection utilisés ne sont pas parfaits et peuvent parfois compter une personne lorsque deux personnes sont trop proches l'une de l'autre et ont tendance à se superposer dans l'image. Nous appellerons donc ce phénomène une fusion qui sera prise en compte dans la sélection des vidéos. Le tableau 4.1 présente, pour chaque vidéo sélectionnée, les différents critères que nous venons de définir.

Numéro de la vidéo	Obstacles	Croisements	Fusions	Nombres d'individus	Durée de la vidéo
1	3	2	12	10	7
2	2	0	4	7	15
5	6	0	6	9	22
7	2	1	4	7	18
12	27	13	14	33	103
17	4	0	4	9	26
19	1	0	0	1	115

TABLE 4.1: Récapitulatif des différents événements se produisant dans les vidéos sélectionnées.

Nous pouvons nous apercevoir que l'ensemble des événements se produisant dans les vidéos sont susceptibles d'interrompre un suivi du fait de la disparition d'un individu dans le champ de vision de la caméra. Dès lors que l'algorithme ne sera pas capable de raccorder une piste passée à sa position future, nous parlerons alors de décrochage. L'efficacité des méthodes que nous verrons dans ce chapitre pourra donc être évaluée au travers de la bonne gestion de ces décrochages et de leurs capacités à s'en accommoder ou à les corriger.

Les différents décrochages pourront être comptabilisés par le biais de graphiques générés à partir des données issues des différents algorithmes. Comme le montre la figure 4.1.

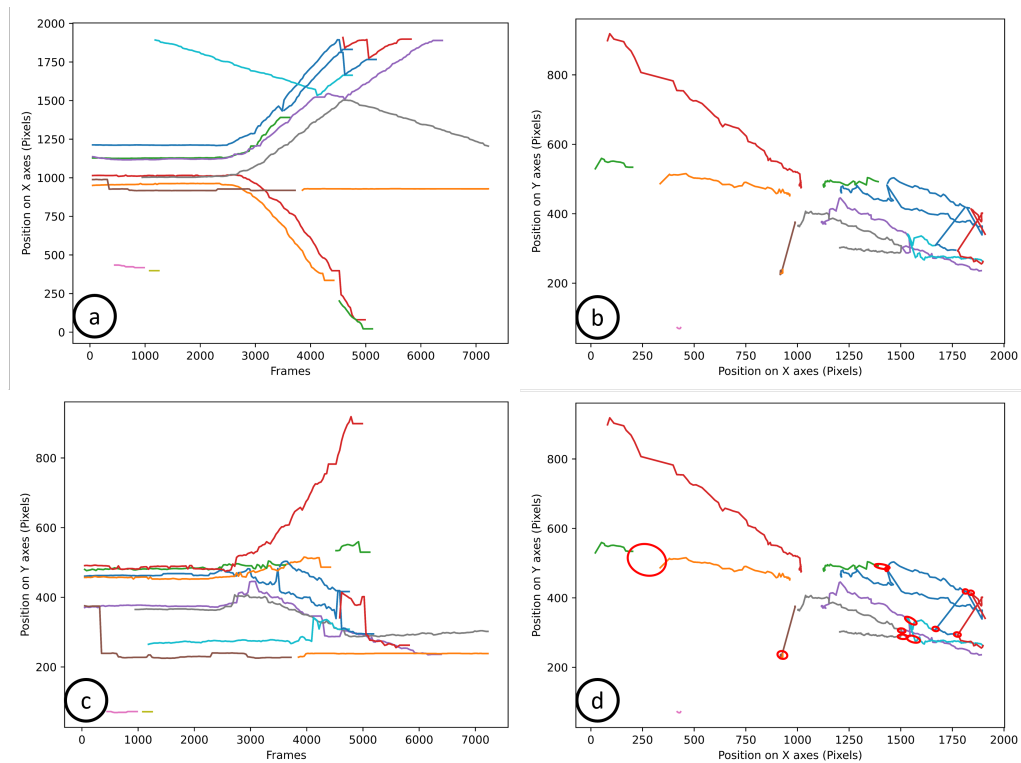


FIGURE 4.1 – Suivi en fonction du temps sur les axes x et y, chaque couleur représente un point suivi (a) suivi sur X de la vidéo n°1 à 30IPS avec la première méthode, (b) suivie sur X et Y de la vidéo n°1 à 30IPS avec la première méthode, (c) suivi sur Y de la vidéo n°1 à 30IPS avec la première méthode, (d) mise en évidence des décrochages de la vidéo n°1 à 30IPS avec la première méthode, chaque décrochage est entouré en rouge.

Les figures 4.1.a, 4.1.c et les figures 4.2.a, 4.2.c représentent la position de chaque individu pour chaque image de la vidéo, chaque ligne colorée représente le suivi potentiel d'une personne. La figure 4.1.a et la figure 4.2.a représentent leurs évolutions sur l'axe des abscisses de la caméra en fonction du temps écoulé. La figure 4.1.c et la figure 4.2.c représentent leurs évolutions sur l'axe des ordonnées en fonction du temps écoulé. La figure 4.1.b et la figure 4.2.b sont une représentation de toutes les trajectoires des personnes présentes dans la vidéo entière. Les lignes sont de couleurs différentes, car chaque couleur correspond à un point connu (avec un identifiant) ce qui signifie que le suivi a été interrompu si pendant la marche d'un individu la ligne change de couleur. La figure 4.1.d est le même graphique que la figure 4.1.b, mais ce qui est considéré comme un "décrochage" est

entouré en rouge. La vidéo n°1 présentée dans la figure 4.1 est une dispersion rapide d'individus initialement groupés. La figure 4.1.d montre 12 décrochages pour la vidéo n°1 à 30IPS (Image par seconde) traitée avec la première méthode.

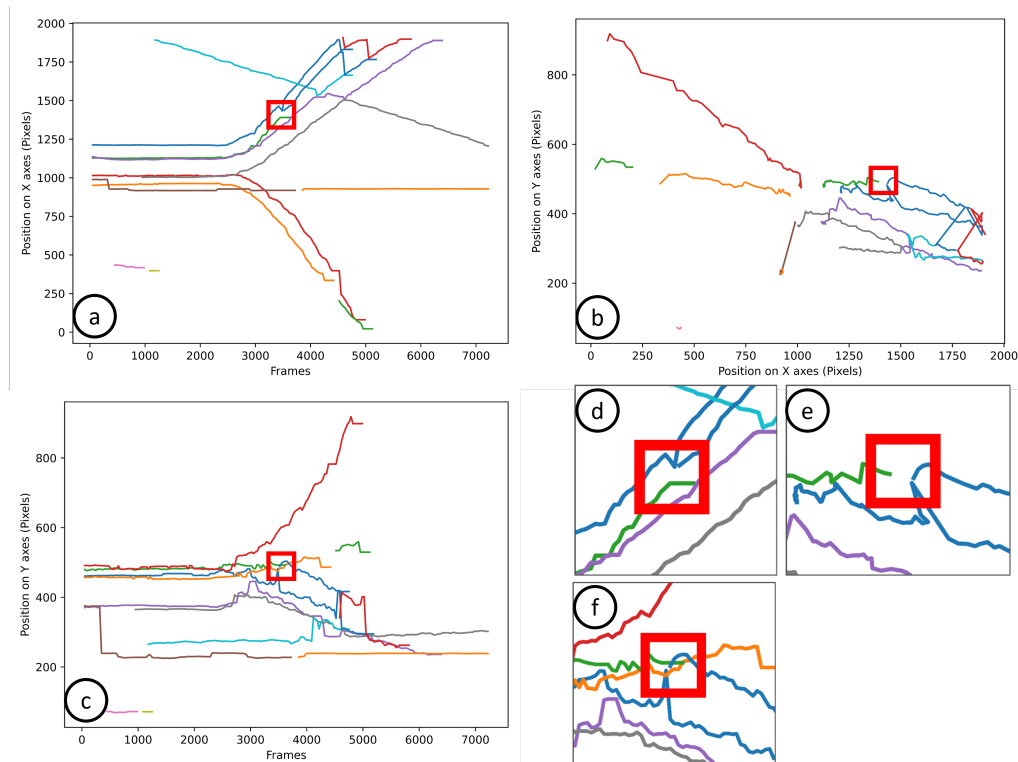


FIGURE 4.2 – Suivi en fonction du temps sur l'axe des x et l'axe des y, chaque couleur représente un point suivi avec mise en évidence d'un décrochage (a) suivi sur X de la vidéo n° 1 à 30IPS avec la première méthode, (b) suivie sur X et Y de la vidéo n° 1 à 30IPS avec la première méthode, (c) suivie sur Y de la vidéo n°1 à 30IPS avec la première méthode, (d) mise en évidence du décrochage sur (a), (e) mise en évidence du décrochage sur (b), (f) mise en évidence du décrochage sur (c).

Dans la plupart des cas, les décrochages sont pris en compte sur les graphiques tels que présentés dans la figure 4.1.b, mais si le nombre de pistes est trop important, il est parfois nécessaire d'utiliser les graphiques tels que présentés dans les figures 4.1.a, 4.1.c. La figure 4.2 montre comment le même décrochage est identifié sur les 3 types de graphiques (a), (b), et (c) ainsi qu'un zoom sur les zones où le décrochage se produit : (d) zoom de (a), (e) zoom de (b), et (f) zoom de (c).



Pour tester les deux algorithmes dans les mêmes conditions, les algorithmes ont reçu les mêmes données. Pour valider la robustesse des algorithmes, l'expérimentation a également été menée en retirant méthodiquement certaines données, augmentant ainsi le temps entre la réception des nuages de deux points. L'objectif de ce retrait est de voir comment les algorithmes se comportent s'ils sont alimentés en données par une détection plus ou moins lente, c'est-à-dire 10, 15, 20, 25 et 30 traitements d'images par seconde. Chaque vidéo a été traitée par les deux algorithmes pour obtenir l'ensemble des nuages de points étiquetés avec un identifiant pour chaque point de chaque nuage reçu.

## 4.2 Le suivi

### 4.2.1 Introduction

L'approche la plus basique a été de développer un algorithme dont le but est de trouver une corrélation entre deux nuages de points issus de deux images se suivant dans une vidéo. En effet, comme expliqué dans le chapitre 2, les données de la vidéo et plus particulièrement les positions et tailles des personnes présentes dans chaque image de la vidéo ont été extraites afin d'obtenir un nuage de points pour chaque image de la vidéo où chaque point représente une personne dans l'image à laquelle nous attribuons également une information sur la taille de la détection. L'objectif de cette méthode est de rechercher les points les plus proches en considérant que l'évolution d'une personne entre deux images dépend de trois paramètres :

- la distance à laquelle se trouve la personne par rapport à la caméra ;
- la vitesse à laquelle la personne se déplace ;
- la vitesse à laquelle la caméra enregistre.

Nous souhaitons donc former des paires de points basés sur la proximité des individus entre deux images de la vidéo. Nous évaluons d'abord la position des points dans le nuage en utilisant uniquement leurs coordonnées X et Y. Cependant, pour procéder à cette évaluation, nous devons d'abord connaître les trois situations observables au départ :

- le nombre de points entre les deux nuages de points est le même ;
- le nombre de points du nuage 1 est inférieur à celui du nuage 2 ;

— le nombre de points du nuage 1 est supérieur à celui du nuage 2.

Chacun de ces cas entraîne une particularité que l’algorithme doit être capable de gérer. Considérons le cas nominal où le nombre d’individus reste inchangé (figure 4.3), la solution la plus logique est de regarder entre les deux nuages de points pour voir quels sont les points les plus proches et ainsi former des paires de points (figure 4.3.d).

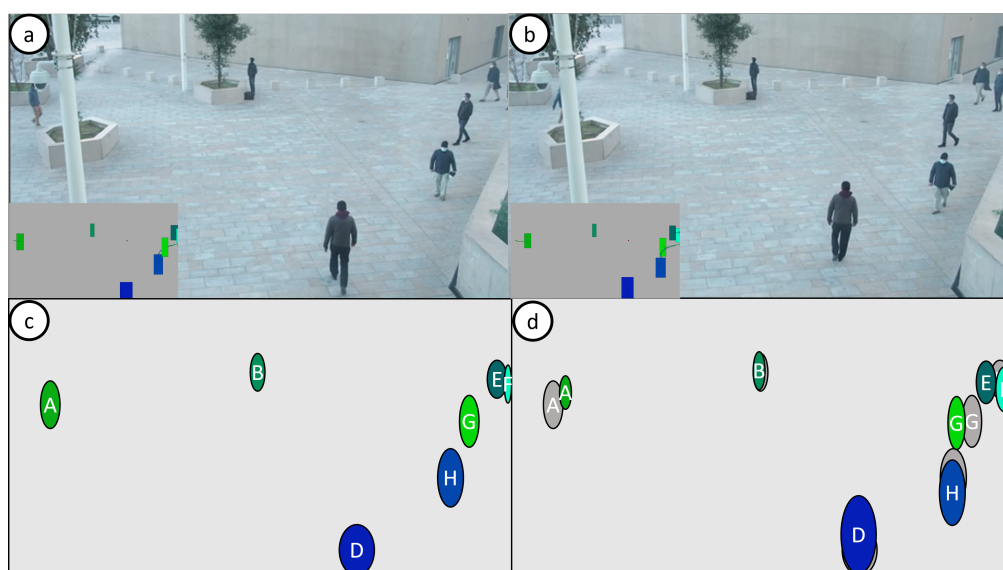


FIGURE 4.3 – Visualisation des détections vidéo et des données anonymisées : (a) Première image traitée par la détection vidéo et (b) seconde image traitée par la détection vidéo. (c) Première image anonymisée et (d) seconde image anonymisée avec association des nouvelles positions aux anciennes.

## 4.2.2 Le regroupement

DBSCAN ainsi que d’autres algorithmes de regroupement permettent de regrouper des points dans un ensemble de points en fonction de paramètres spécifiques comme nous avons pu le voir en 3.2. Le principal paramètre utilisé ici est la distance, car notre objectif principal est de regrouper les points les plus proches. L’autre paramètre qui est important est la quantité de points à regrouper, dans

notre cas le regroupement ne peut se faire que sur deux points maximum. Nous n'utilisons pas l'ensemble de l'algorithme, nous n'utilisons que la partie consistant à former des groupes. Cette partie est ensuite modifiée de manière à ne pas former de groupes dans le même nuage de points, mais à former des groupes entre deux nuages de points. Tout point appartenant au premier nuage ne peut être groupé qu'avec un seul autre point appartenant à un second nuage. Cette méthode fonctionne si le nombre de points entre le premier et le second nuage reste inchangé, si les points qui sont présents dans le premier nuage sont également présents dans le second nuage et si seule leur position a changé. En dehors de ce cas très spécifique et idéal, d'autres existent.

Pour effectuer les regroupements avec DBSCAN, nous attribuons à chaque point du nuage A un rayon de détection. Il suffit ensuite de rechercher dans le nuage B tous les points qui rentrent dans les cercles de détection précédemment établis. Comme le montre la figure 4.4, dans une utilisation normale, plusieurs points peuvent être regroupés s'ils sont à l'intérieur de plusieurs rayons de détection. Dans notre cas, nous allons essayer d'avoir un seul point dans chaque rayon de détection. 4.4.

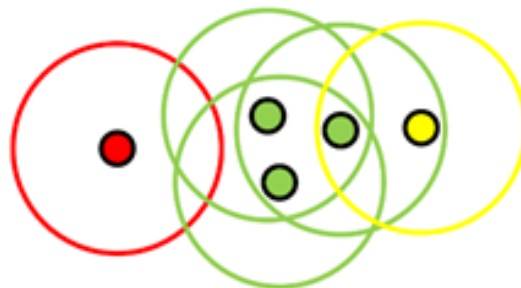


FIGURE 4.4 – Classification par DBSCAN : (Vert) Points classifiés. (Jaune) Points d'extrémité. (Rouge) Points uniques.

Comme le montre la figure 4.5, l'algorithme de traitement de proximité est le suivant :

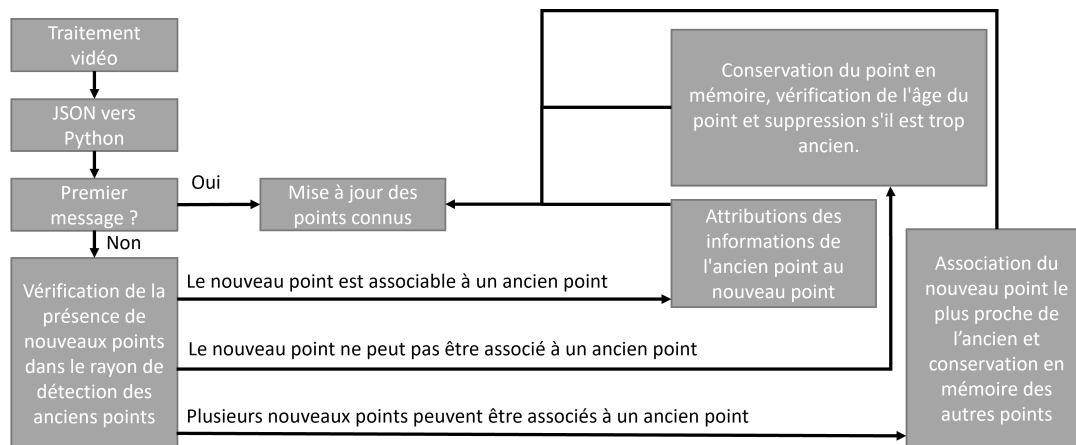


FIGURE 4.5 – Fonctionnement de l’algorithme de suivi de proximité.

### 4.2.3 Interprétation des données

Comme expliqué dans le chapitre 2, le flux vidéo est traité indépendamment des systèmes d’analyse. Une fois traité, le résultat est stocké dans un fichier JSON. La première étape consiste à convertir les données du fichier JSON en données pouvant être traitées par le langage utilisé (dans notre cas python3). S’il s’agit alors des premières données du fichier, elles sont automatiquement intégrées dans les points connus. Les points connus dans notre cas sont tous les points ayant un identifiant. S’il ne s’agit pas du premier message, la première étape consiste à former les paires possibles entre les anciens points connus et les nouveaux points reçus. Trois résultats sont alors possibles :

- Le nouveau point est associable à un ancien point ;
- Le nouveau point n’est pas associable à un ancien point ;
- Plusieurs points sont associables à un ancien point.

Dans le premier cas, la solution est simple, il existe un ancien et un nouveau point parfaitement associable, le nouveau point se voit donc attribuer les informations de l’ancien. Dans le second cas, si le nouveau point ne peut être associé qu’à un ancien point, cela peut signifier soit que le nouveau point est une mauvaise détection, soit que la personne représentée par ce point est passée derrière un obstacle ou a quitté le champ de vision de la caméra. Dans les deux cas, le point est en-

registré en mémoire pour être utilisé ultérieurement. Cependant, ce stockage en mémoire doit être temporaire, car s’il ne s’agit pas d’une disparition temporaire, le point ne doit pas être réaffecté. Dans le troisième cas, si plusieurs nouveaux points sont associable à un ancien point, nous associons simplement le point le plus proche de l’ancien point et les autres points sont gardés en mémoire de la même manière que pour le deuxième cas. Cependant, cette solution est problématique si entre deux images un des individus a disparu et un autre est apparu ailleurs dans l’image. En effet, nous avons le même nombre de points entre les deux nuages (figure 4.6.a), nous pouvons donc former nos couples, cependant, la distance entre certains couples sera trop grande pour qu’il s’agisse effectivement de la même personne (figure 4.6.b).

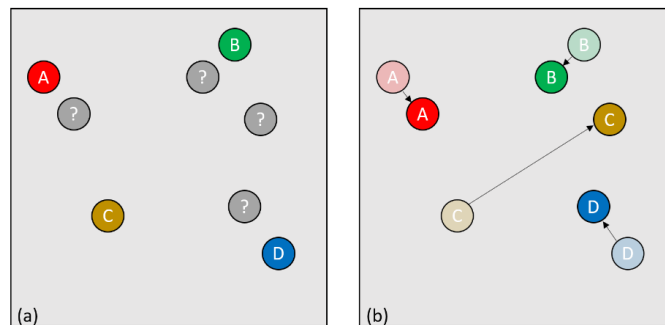


FIGURE 4.6 – (a) Quatre points identifiés et quatre nouveaux points reçus et non identifiés, dont un disparaît et un autre apparaît ailleurs (b) les points non identifiés sont associés aux points identifiés.

Pour corriger ce problème, une solution simple consiste à fixer un rayon autour de chaque point. Dans notre cas ce rayon a été ajusté par l’observation de la scène et de l’environnement pendant plusieurs heures. Une fois le rayon établi, si un point apparaît en dehors de tous les rayons présents dans le nuage, il s’agira très probablement d’un nouvel individu, auquel cas nous pouvons conclure qu’une autre personne a disparu de l’image. Cette technique suit la logique d’une méthode de clustering DBSCAN [Den20] qui permet à la fois de regrouper et d’identifier des individus isolés. Dans le cas où le nombre de points du premier nuage est inférieur à celui du second nuage, nous sommes confrontés à la disparition d’un individu. Cette disparition peut être due soit à une sortie de la zone filmée par la caméra, soit à un passage derrière un obstacle (figure 4.7.a à figure 4.7.b) qui empêcherait

la caméra d'identifier l'individu (figure 4.7.c). Dans le premier cas, la solution la plus simple est de vérifier où la disparition a eu lieu. Si nous sommes proches du bord de la zone surveillée, nous considérerons que la personne a effectivement quitté le champ, en veillant à ce que la valeur de proximité du bord utilisée soit la plus petite possible pour que la personne continue à être suivie si elle passe derrière un obstacle proche d'un bord par exemple. Dans le second cas, si la personne disparaît ailleurs qu'à proximité d'un des bords de la zone surveillée, cela signifie qu'elle est très probablement passée derrière un obstacle et qu'elle peut potentiellement réapparaître. Dans ce cas, il faut former les paires évidentes et enregistrer la dernière position connue de la personne qui a disparu. En gardant la trace de toutes les positions, il sera possible de retrouver la personne si elle réapparaît. Si la personne est perdue pendant trop longtemps et réapparaît ensuite (figure 4.7.d à figure 4.7.e), elle sera identifiée comme une nouvelle personne (figure 4.7.f).



FIGURE 4.7 – (a) Un individu entre dans l'image, (b) l'individu passe derrière un obstacle, (c) l'individu n'est plus caché, (d) le point représentant l'individu de l'image (a) est identifié, (e) l'individu identifié en (d) est perdu, (f) l'individu identifié en (d) est toujours perdu et un autre individu est identifié dans l'image (c).

Pour que cette méthode fonctionne, nous devons fixer un nombre d'images pendant lequel les points perdus seront gardés en mémoire, car si nous gardons ces points en mémoire trop longtemps, nous pourrions alors détecter un nouvel individu qui apparaîtrait au même endroit qu'une disparition et donc l'algorithme de suivi peut considérer qu'il s'agit de la même personne.

Enfin, dans le cas où le nombre d'individus de la première image est supérieur à celui de la deuxième image et que nous n'avons pas réussi à former une paire à partir des points qui étaient en mémoire, cela signifie qu'un individu est apparu dans le champ de vision de la caméra. Dans ce cas, il faut vérifier si l'apparition a eu lieu près d'un des bords, car cela signifierait simplement qu'il y a un nouvel individu dans l'image, il n'y aura donc pas de couples associés à l'ancienne image pour ce point. Dans le cas où le nombre d'individus dans la première image est supérieur à celui de la deuxième image (figure 4.8), nous sommes alors dans un cas d'apparition qui peut être dû au fait qu'une personne entre dans le champ de vision de la caméra ou qu'une personne était cachée derrière un obstacle et est à nouveau observable. Nous allons donc d'abord établir toutes les paires évidentes entre les deux images en utilisant la méthode de regroupement expliquée précédemment et nous considérons ensuite que la personne qui vient d'apparaître est une nouvelle personne à suivre (figure 4.8.d).

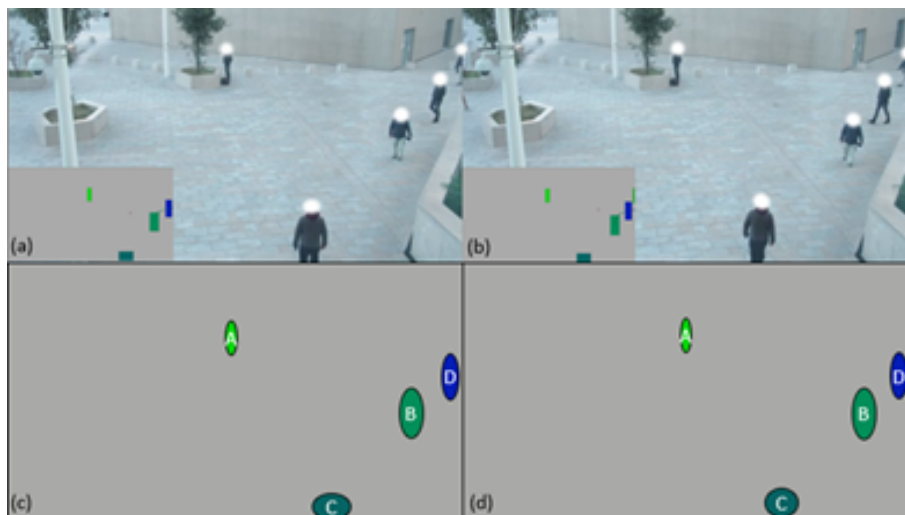


FIGURE 4.8 – Passage de quatre à cinq personnes dans le champ de vision de la caméra. (a) Représentation graphique de la vue algorithmique à un instant  $t$  avec 4 personnes, (b) représentation graphique de la vue algorithmique à un instant  $t + 1$  avec 5 personnes, (c) 4 points identifiés, (d) les points non identifiés sont associés aux points identifiés ; les points seuls reçoivent un nouvel identifiant.

#### 4.2.4 Résultats

Les résultats montrent que la méthode fonctionne dans le cas idéal où le nombre d'images traitées est suffisamment grand (idéalement supérieur à 20), le champ de vision de la caméra est suffisamment clair et les individus présents dans la scène n'ont pas de trajectoires qui se croisent. Dès que ces paramètres sont réunis, nous disposons d'un algorithme efficace et rapide. Il est efficace, car le traitement se fait uniquement entre deux nuages de points et chaque point aura un identifiant le plus probable qui s'avère souvent être le bon. L'algorithme est également rapide, en effet, une fois que toutes les comparaisons de points sont faites, l'algorithme est capable de donner un identifiant à tous les points. Cependant, l'algorithme est sensible au croisement d'individus et au passage derrière des obstacles comme le montre le tableau 4.2.

IPS	Décrochage pour 7 vidéo						
	n°1	n°2	n°3	n°4	n°5	n°6	n°7
10	12	3	5	6	24	18	0
15	12	2	5	7	18	14	0
20	12	5	5	6	16	14	0
25	12	6	5	7	15	12	0
30	12	7	5	7	22	12	1

TABLE 4.2: Décrochage total du suivi de proximité sur un ensemble de sept vidéos avec un nombre prédéfini d'IPS.

Notre étude vise à réaliser le suivi dans un environnement urbain, le franchissement et le passage derrière un obstacle sont donc réguliers. Nous devons donc trouver une solution permettant de gérer ces décrochages.

### 4.3 Prédiction de trajectoire

#### 4.3.1 Introduction

Afin de corriger la méthode présentée dans le paragraphe précédent, il était nécessaire de trouver un moyen d'estimer l'évolution probable entre deux nuages de points par leurs distances, mais aussi par leur comportement passé. Plusieurs



méthodes, notamment dans le domaine maritime [FF18; Hon+19], permettent par exemple d'estimer la trajectoire de véhicules marins ou même aériens [SNB18; PZS20] qui ont une forte inertie en utilisant des filtres de Kalman. Puisque les filtres de Kalman peuvent être utilisés à partir de données de position et de vitesse, ils peuvent également être utilisés pour le guidage aérien en l'absence de GPS ou d'autres technologies de navigation par satellite [KYK19b]. Les filtres de Kalman peuvent également être utilisés dans le traitement du signal pour le lissage ou l'estimation [GCN07].

Le champ d'application est donc très large, c'est pourquoi l'utilisation d'un tel filtre est devenue évidente. Une telle méthode pourrait nous permettre de faire des prédictions sur la position des futurs points de notre nuage de points, car même si l'inertie d'un véhicule est plus importante que celle d'un piéton qui marche, du fait de la différence de masse, l'évolution de ce dernier n'est pas aléatoire et conserve une trajectoire assez régulière avec des angles naturels. Cependant, il sera important d'intégrer l'idée qu'une trajectoire peut être perturbée et potentiellement révéler une erreur de trajectoire ou un comportement anormal. En utilisant les filtres de Kalman, nous sommes capables d'estimer la trajectoire future d'un objet se déplaçant dans un espace 2D ou 3D et ne subissant pas un changement soudain de son inertie.

Notre cas d'étude s'applique au suivi urbain par le biais de caméras de vidéo-surveillance, nous pouvons donc utiliser ces filtres. Nous utiliserons les filtres de Kalman dans un contexte discret. Dans ce contexte, nous n'aurons besoin que de 2 images pour mettre en place nos systèmes de filtres  $N-1$  et  $N$ . Les données de  $N-1$  nous permettront de faire une prédiction de la position future des points de  $N$  et  $N$  nous permettra de valider les différentes prédictions faites. De plus, l'utilisation des filtres de Kalman reste compatible avec la méthode de suivi de proximité. Nous pouvons également utiliser l'effet d'association par la proximité pour renforcer le suivi par Kalman.

En plus des domaines mentionnés ci-dessus, des utilisations des filtres de Kalman en suivi sont possibles pour permettre, par exemple, dans le domaine du sport d'identifier un joueur à tout moment [QY20] ou dans le domaine de la sécurité de

combler les lacunes dans le mouvement d'une personne qui passerait derrière un obstacle par exemple [HEW19]. Cependant, en raison de la puissance de calcul nécessaire pour effectuer à la fois la détection et le suivi, il existe très peu de travaux sur le sujet dans le domaine du suivi urbain. La plupart des travaux utilisant les filtres de Kalman dans le domaine du suivi vidéo sont directement appliqués sur cette dernière.

La méthode que nous proposons permet de se détacher complètement de la partie vidéo et se concentre uniquement sur le suivi de points anonymes. Ce faisant, le travail effectué ici peut être appliqué dans différents domaines en dehors du suivi urbain. En effet, cette méthode peut être utilisée pour suivre n'importe quelle population d'objets du moment qu'ils sont identifiables par des paramètres qui évoluent sans variations brusques dans le temps.

### 4.3.2 Principe de fonctionnement

Comme pour la méthode de suivi de proximité, nous avons la même série de nuages de points comme données d'entrée où chaque élément de la série est associé à une image de la vidéo. Cependant, comme la taille de détection est importante dans cette méthode, nous utiliserons pour chaque point l'information sur sa hauteur et sa largeur en pixels. Ainsi, nous pouvons associer chaque point à une surface. Ces nuages de points seront ensuite traités indépendamment les uns des autres, mais tout nuage de points aura un impact sur l'analyse des points suivants. Pour effectuer l'analyse, nous considérons que nous avons deux types de points : des points dits "anonymes détectés", car ils proviennent directement des nuages de points, et des points dits "identifiés par suivi basé sur le filtre de Kalman" générés et gérés par nos filtres de Kalman. L'objectif est d'associer un point "détecté anonyme" à un point "identifié par suivi basé sur le filtre de Kalman"; nos points "identifiés par suivi basé sur le filtre de Kalman" sont gérés par l'algorithme qui créera un de ces points dès qu'un point "détecté anonyme" ne pourra être associé à aucun point "identifié par suivi basé sur le filtre de Kalman" existant. Nous nous retrouvons avec un tableau de points "identifiés par un suivi basé sur le filtre de Kalman" qui évolue au fur et à mesure que nous recevons des nuages de points à traiter. Prenons l'exemple suivant :

- L'algorithme reçoit un nuage de points N1 qui contient X points (figure 4.9.a);
- L'algorithme regarde si l'un des points "détectés anonymement" reçus partage une surface suffisante (figure 4.9.c) avec celle de l'un des points de "suivi basé sur le filtre de Kalman" (figure 4.9.b);
- Si oui, l'identifiant du point "identifié par suivi par filtre de Kalman" est donné au point "détecté anonyme" qui devient un point identifié (figure 4.9.d) et les données de position du point "détecté anonyme" sont intégrées au point "identifié par suivi par filtre de Kalman" associé (figure 4.9.f) de sorte que la trajectoire du point "identifié par suivi par filtre de Kalman" continue d'évoluer en fonction de son point "détecté anonyme" (figure 4.9.e);
- sinon, un point de "suivi basé sur le filtre de Kalman" est créé avec les données du point "détecté anonymement" (figure 4.9.e).

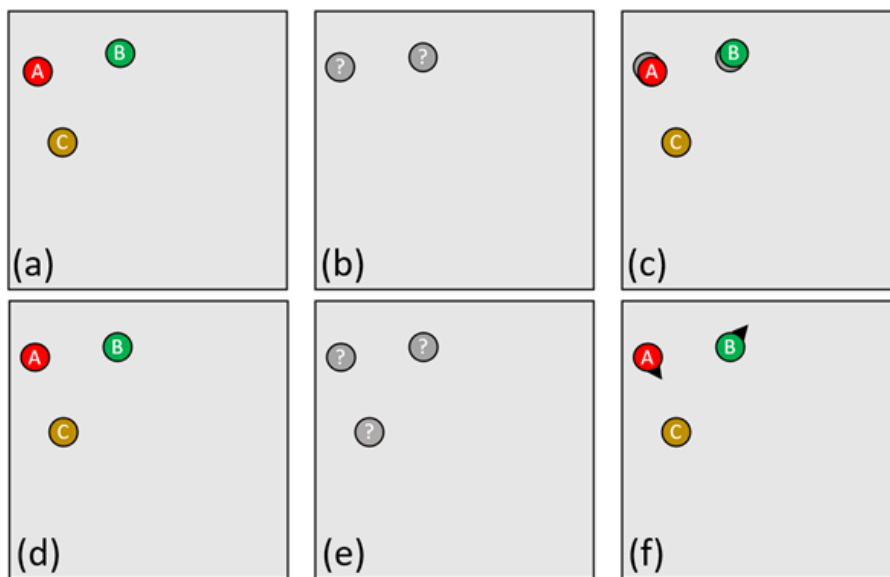


FIGURE 4.9 – (a) Trois points sont identifiés, (b) deux points "identifiés par le suivi par filtre de Kalman" existent en mémoire, (c) calcul des surfaces communes et création de points "identifiés par le suivi par filtre de Kalman" pour les points n'ayant pas de paire, (d) trois points sont identifiés, (e) trois points "identifiés par le suivi par filtre de Kalman" existent en mémoire, (f) le calcul de la surface commune permet maintenant de connaître la trajectoire de A et B.

Comme le montre la figure 4.10, l'algorithme de traitement utilisant les filtres de Kalman est le suivant :

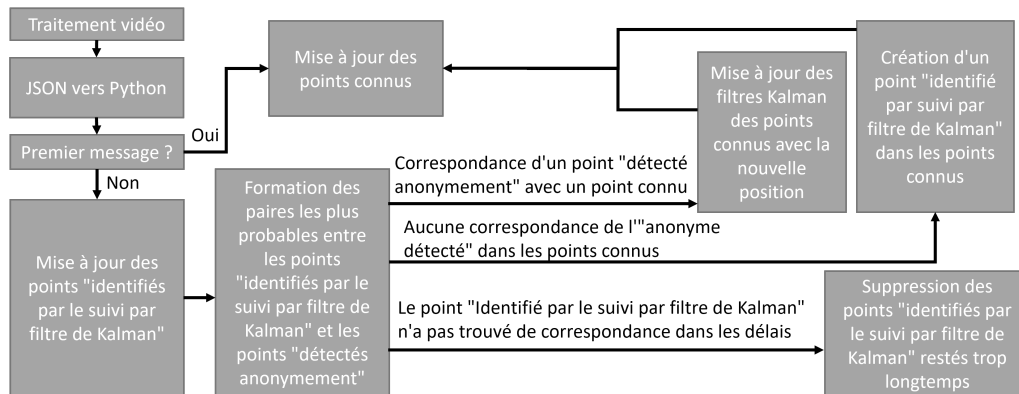


FIGURE 4.10 – Fonctionnement de l'algorithme de suivi avec les filtres de Kalman.

S'il s'agit de la première donnée du fichier, elle est automatiquement intégrée aux points connus. Les points connus dans notre cas sont tous les points ayant un identifiant. S'il ne s'agit pas du premier message, les points "identifiés par le filtre de Kalman" sont mis à jour. Il est alors possible de comparer la nouvelle position des points "identifiés par le filtre de Kalman" et les nouveaux points reçus. Chaque nouveau point reçu sera comparé à l'ensemble des points "filtrés par Kalman" et donnera l'un des résultats suivants :

- Le point correspond à l'un des points connus ;
- Le point ne correspond à aucun point connu ;
- L'un des points "identifiés par le filtre de Kalman" ne trouve pas de correspondance.

Dans le premier cas, les filtres des points "identifiés par les filtres de Kalman" sont mis à jour tout comme les points connus. Dans le deuxième cas, un nouveau point "identifié par le filtre de Kalman" est créé dans les points connus et les points connus sont mis à jour. Enfin, dans le troisième cas, si le filtre est resté trop longtemps sans correspondance, il est supprimé. Une fois que l'algorithme a vérifié chaque point "détecté anonyme", il attribue un nouvel identifiant aux points restés "détectés anonyme", met à jour la position des filtres, et supprime les filtres restés trop longtemps non re-identifiés ou dont les coordonnées ne sont plus dans la zone surveillée. La prédiction effectuée par les filtres de Kalman est basée sur

deux fonctions effectuant une série de calculs matriciels. La première fonction est une fonction de mise à jour qui permet au filtre de mettre à jour les informations de son point connu avec les informations du point qui lui a été réassocié. La seconde fonction est une méthode de prédiction qui à partir de la dernière information connue va prédire l'information du futur point qui pourrait lui être associé. Afin de respecter la méthode utilisée par les filtres de Kalman, nous disposons de six matrices principales :

- E, la matrice représentant le vecteur d'état initial contenant les informations de position, de vitesse et de taille du point ;
- A, la matrice de transition permettant de passer de l'état de l'image I-1 à I ;
- O, la matrice d'observation permettant de définir les paramètres à surveiller, dans notre cas il s'agit des informations connues dans le nuage de points, à savoir la position et la taille ;
- Q, la matrice de covariance représentant le vecteur de bruit du processus ;
- R, la matrice de covariance représentant le vecteur du bruit de mesure ;
- P, la matrice de prédiction évoluant avec chaque nouvelle information ;
- Les signes +, -, . sont utilisés pour les opérations matricielles de base ;
- Le signe (-1) est utilisé pour inverser les matrices.

Dans le cas de la mise à jour, nous recevons également la matrice J représentant le vecteur des nouvelles informations du point et qui permet de renforcer la prédiction à venir ainsi que G une matrice identité de même dimension que E. Dans un premier temps, nous mettons à jour les informations de E :

$$E = E + ((P.O^T).((O.(P.O^T) + R))^{(-1)}).(J - (O.E))$$

Cette équation permet de mettre à jour les informations de position :

$$P = (G - ((P.O^T).(((O.(P.O^T)) + R))^{(-1)})).O).P$$

Cette équation permet de mettre à jour la matrice de prédiction.

Dans le cas de la prédiction, le calcul est plus simple :

$$E = A.E$$

Pour prédire la position future :

$$P = ((A.P).A^T) + Q$$

Cette équation permet de calculer la covariance de l'erreur. Le tableau 4.3 montre, pour une piste, son état initial avec ses coordonnées sur X et Y en pixels, sa vitesse sur X et Y en pixels/seconde, sa largeur et sa hauteur en pixels. Nous obtenons alors une prédiction de la position future de la piste puis l'état réel de la piste

qui a été associé à la prédiction, et ce, pour 17 images successives. Dans le cas du tableau 4.3, les résultats ont été obtenus à l'aide d'un suivi par filtre de Kalman, mais ont également été vérifiés manuellement afin de s'assurer que l'extrait de données présenté ne comporte aucune perte. Il ne s'agit que d'un extrait, car le fichier original compte plus de 3000 images.

	Initial	P. 1	R. 1	P. 2	R. 2	P. 3	R. 3	P. 4	R. 4	P. 5
X Coordinate	1680	1680	1694	1701	1706	1718	1715	1725	1728	1739
Y Coordinate	454	454	449	447	456	463	453	454	445	441
X Speed	0	0	7	7	12	12	10	10	11	11
Y SPEED	0	0	-2	-2	7	7	1	1	-4	-4
Width	85	85	92	92	111	111	113	113	111	111
Height	210	210	213	213	218	218	222	222	219	219
	R. 5	P. 6	R. 6	P. 7	R. 7	P. 8	R. 8	P. 9	R. 9	P. 10
X Coordinate	1766	1778	1780	1793	1787	1787	1795	1804	1811	1823
Y Coordinate	425	419	422	418	426	426	424	423	420	417
X Speed	12	12	13	13	10	10	9	9	12	12
Y SPEED	-6	-6	-4	-4	0	0	-1	-1	-3	-3
Width	85	85	83	83	98	98	109	109	96	96
Height	201	201	207	207	224	224	222	222	216	216
	R. 10	P. 11	R. 11	P. 12	R. 12	P. 13	R. 13	P. 14	R. 14	P. 15
X Coordinate	1820	1830	1831	1841	1840	1850	1850	1860	1855	1862
Y Coordinate	418	415	416	413	414	411	409	405	406	402
X Speed	10	10	10	10	10	10	10	10	7	7
Y Speed	-3	-3	-3	-3	-3	-3	-4	-4	-4	-4
Width	90	90	87	87	8	89	91	91	89	89
Height	207	207	200	200	198	198	197	197	204	204
	R. 15	P. 16	R. 16	P. 17	R. 17					
X Coordinate	1862	1869	1870	1877	1875					
Y Coordinate	402	398	398	394	396					
X Speed	7	7	7	7	6					
Y Speed	-4	-4	-4	-4	-3					
Width	87	87	85	85	81					
Height	198	198	199	199	200					

TABLE 4.3: Échantillon de 17 comparaisons entre la prédiction et la réalité. (P. : Prédiction, R. : Réel)

### 4.3.3 Correspondance réalité et prédiction

Nous constatons alors que les prédictions sont très proches de la réalité. Le tableau 4.4 montre toutes les surfaces communes entre les prédictions et la réalité. Cet exercice a été réalisé sur plusieurs échantillons afin de valider l'efficacité du suivi d'une part, mais aussi de déterminer le seuil à partir duquel une prédiction peut être rattachée à une piste réelle. Le pourcentage de surface commune choisi pour la comparaison a d'abord été fixé arbitrairement à 70%. Il a ensuite été abaissé à 50%, car, après plusieurs tests, il a été constaté qu'un taux supérieur à 50% était indicatif d'une bonne correspondance comme le montre le tableau 4.4 et qu'à l'inverse, les correspondances inférieures à 50% sont le plus souvent des chevauchements dus à des individus qui se croisent ou marchent à proximité. De plus, étant donné que chaque point "identifié par le suivi par filtre de Kalman" ne peut être associé qu'à un seul point "détecté anonyme", nous constatons que les points "identifiés par le suivi par filtre de Kalman" qui se chevauchent à moins de 50% avec un point "détecté anonyme" ont dans la plupart des cas un autre point "détecté anonyme" sur lequel plus de 50% de la surface est commune. Cet effet permet dans tous les cas de ne sélectionner que les correspondances avec plus de 50% de la surface et de valider cette valeur.



	Numéro de l'image								
	n°1	n°2	n°3	n°4	n°5	n°6	n°7	n°8	n°9
Surface commune en Pixels	14,768	17,748	23,320	23,100	15,540	16,434	16,517	19,800	20,448
Prédiction + Surface actuelle	22,678	26,046	25,964	26,295	25,854	17,832	22,616	26,350	24,486
Surface commune en %	65.12	68.14	89.82	87.85	60.11	92.16	73.03	75.14	83.51

	Numéro de l'image							
	n°10	n°11	n°12	n°13	n°14	n°15	n°16	n°17
Surface commune en Pixels	18,009	17,400	17,226	17,355	16,464	17,226	16,830	15,563
Prédiction + Surface actuelle	21,357	18,630	17,796	18,194	19,619	18,156	17,311	17,552
Surface commune en %	84.32	93.40	96.80	95.39	83.92	94.88	97.22	88.67

TABLE 4.4: Zone commune entre la réalité et la prédiction sur un échantillon de 17 images.

Nous avons donc deux cas possibles pour chaque point "détecté anonyme" : soit le point "détecté anonyme" peut être associé à un point "identifié par le suivi basé sur le filtre de Kalman", cette association se fait si la surface commune entre les deux points est suffisante. Dans le cas où plusieurs points "détectés anonyme" ont une surface commune suffisante, c'est le point ayant la plus grande surface commune qui sera conservé. Soit le point "détecté anonyme" n'est associable à aucun point "identifié par suivi par filtre de Kalman" et dans ce cas un point est créé en utilisant les informations de position et de taille associées au point "détecté anonyme". Une fois cette vérification effectuée, notre nuage de points n'est plus "détecté anonyme" et nous pouvons attendre le nuage suivant. Cette méthode re-

lève également des cas de décrochage lorsque le nombre d'images par seconde n'est pas suffisant ou si la trajectoire d'un individu n'est pas assez "naturelle"; le nombre de décrochages est présenté dans le tableau 4.5.

IPS	Décrochage pour 7 vidéo						
	n°1	n°2	n°3	n°4	n°5	n°6	n°7
10	8	2	2	2	4	3	1
15	4	0	2	2	2	1	2
20	3	2	3	2	4	0	1
25	3	2	3	1	3	0	1
30	1	3	2	3	4	0	1

TABLE 4.5: Décrochages totaux dans le suivi avec le filtre de Kalman sur un ensemble de sept vidéos avec un nombre prédéfini d'IPS.

Nous pouvons constater que le tableau 4.5 présente de meilleurs résultats, car le nombre de décrochages est moins important. Nous pouvons donc comparer les deux méthodes à l'aide de ces résultats.

#### 4.3.4 Discussions et conclusion

Tous les résultats sont présentés et comparés dans le tableau 4.6. Le tableau 4.6 indique le nombre de décrochages pour chaque méthode. En général, la méthode la plus fiable est celle qui utilise les filtres de Kalman. Cette efficacité peut être observée en faisant la moyenne des décrochages (toutes vidéos confondues) de chaque fréquence IPS, ce qui montre un minimum de 209% de décrochage en plus avec la première méthode (Tableau 4.7).

IPS	Décrochage V1							Décrochage V2						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
10	12	3	5	6	24	18	0	8	2	2	2	4	3	1
15	12	2	5	7	18	14	0	4	0	2	2	2	1	2
20	12	5	5	6	16	14	0	3	2	3	2	4	0	1
25	12	6	5	7	15	12	0	3	2	3	1	3	0	1
30	12	7	5	7	22	12	1	1	3	2	3	4	0	1

TABLE 4.6: Comparaison du nombre de décrochages entre les deux versions de l'algorithme.

IPS	Erreur moyenne IPS (proximité)	Erreur moyenne IPS (Kalman)	Différence
10	10	3	+209.09%
15	8	2	+346.15%
20	8	2	+286.67%
25	8	2	+338.46%
30	9	2	+371.43%

TABLE 4.7: Comparaison des décrochages à une fréquence variable d'IPS, toutes vidéos confondues.

En revanche, il a été identifié que dans certains cas de vidéos, la première méthode s'avérerait plus efficace. En effet, la vidéo n°19 montre la présence d'une seule personne tout au long de la vidéo. Il s'avère que l'avantage des filtres de Kalman de pouvoir anticiper la trajectoire d'un individu peut devenir un inconvénient si ce dernier s'arrête brusquement et ne reprend pas sa marche. Le filtre permettant de prédire la position future du point, ne peut pas anticiper un arrêt brutal qui a pour effet de faire décrocher le suivi. Cet effet peut également être observé si le point disparaît parce qu'il est obstrué par un obstacle trop longtemps. De plus, la première méthode consistant à comparer les points les plus proches pour les relier effectuera nécessairement un suivi parfait s'il n'y a qu'un seul point dans chaque image, car il n'y a pas d'autre possibilité (tableau 4.8). Cependant, ces problèmes ont été identifiés à l'avance et n'affectent pas suffisamment les résultats pour révéler un réel avantage à utiliser la méthode de suivi de proximité seule.

Vidéo n°	Erreur moyenne (proximité)	Erreur moyenne (Kalman)	Différence
1	12	4	+215.79%
2	5	2	+155.56%
3	5	2	+108.33%
4	7	2	+230.00%
5	19	3	+458.82%
6	14	1	+1,650.00%
7	0	1	-83.33%

TABLE 4.8: Comparaison des décrochages sur toutes les vidéos avec tous les IPS.

Les résultats expérimentaux montrent que la méthode du filtre de Kalman est efficace, dans la plupart des cas, un suivi beaucoup plus efficace que la méthode de proximité. Ce résultat provient de l'intégration d'une estimation de la trajectoire. Cette intégration dans un environnement urbain est très efficace, car elle permet d'anticiper l'évolution de la position d'une personne marchant normalement. Cependant, dès que le mouvement d'un individu est plus aléatoire ou brusque, l'algorithme ne sera plus capable d'estimer les trajectoires. La combinaison des deux méthodes nous permet d'être le plus efficaces possible en ayant à la fois une prédiction à court terme qui fait généralement correspondre l'image future avec la prédiction faite, mais aussi un système de correspondance par proximité. En effet, dans certains cas, les prédictions peuvent ne pas correspondre à la réalité, soit parce que la piste a disparu et dans ce cas, une prédiction future devrait pouvoir retrouver la piste; soit parce que les informations utilisées pour faire la prédiction ne sont pas nécessaires, dans ce cas, l'association par proximité permettra de retrouver la piste et de renforcer la prédiction future. Néanmoins, il y a encore quelques décrochages à la fin des expériences. Dans la plupart des cas, il s'agit de décrochages dus au passage derrière un obstacle ou lorsque plusieurs personnes se croisent. Plus rarement, il peut arriver que le décrochage soit lié à une mauvaise détection et dans ce cas, il serait nécessaire d'améliorer le système de détection.

### 4.3.5 Conclusion

Les expérimentations mettent en évidence l'avantage d'utiliser les filtres de Kalman pour le suivi des personnes dans les environnements urbains. Les résultats sont disponibles en ligne pour les vidéos utilisées pour obtenir les données de ces expérimentations : <https://youtu.be/R-NkqDYEAjM>. Les méthodes proposées montrent qu'un suivi image par image ne peut être efficace que si une trace des positions passées est conservée. Une méthode simple de comparaison de positions permet déjà d'obtenir des résultats efficaces dans un environnement contrôlé : sans obstruction et avec un nombre contrôlé d'individus se déplaçant à distance les uns des autres. L'utilisation de cette même méthode avec l'association d'une prédiction des positions futures rend le système de suivi beaucoup plus efficace et lui permet de réussir à suivre des individus en conditions réelles : avec des disparitions derrière des obstacles, des croisements d'individus, des changements brusques de trajectoires. Cependant, le système n'est pas infallible, en effet

il est possible que le suivi s'arrête lorsqu'il y a trop d'individus dans l'image qui se déplacent trop rapidement. Pour pallier ce problème, plusieurs solutions sont possibles. Dans le cas présent, la solution mise en œuvre est celle présentée dans [Aug+21]. L'objectif est de se concentrer sur le mouvement d'un groupe à travers leur évolution dans le temps en étudiant par exemple la variation de l'écart-type et du barycentre. Cette méthode permet de réaliser des conclusions beaucoup plus rapides lorsque le nombre d'individus est trop important. Une autre solution serait l'intégration d'un facteur de forme fourni lors de la détection vidéo et qui serait calculé de la même manière pour tous. Cette ou ces valeurs pourraient être calculées en utilisant des informations qui ne sont pas encore utilisées par les méthodes précédentes comme une valeur liée à la couleur de la détection ou à la morphologie de la détection par exemple en utilisant la méthode présentée dans [Goo+21]. Ce facteur, s'il évolue de manière douce, pourrait même être intégré dans la méthode à l'aide de filtres de Kalman et peut-être même utilisé directement dans le filtre. Une autre solution possible serait d'utiliser les résultats de la méthode par filtres de Kalman pour entraîner une intelligence artificielle par des simulations. En effet, la méthode fonctionnant déjà correctement, les résultats de cette dernière pourraient nous permettre d'avoir suffisamment de données pour entraîner un réseau de neurones, par exemple.

# PERSPECTIVES SIMULATEUR ET ANALYSEUR ET LIVRABLE

---

## 5.1 Livrable

L'ensemble des travaux de cette thèse avait pour but de contribuer à l'implémentation d'un système de sécurité urbain. Dans ce contexte, cette thèse ne s'insère que dans une des problématiques proposées dans le projet SafeCity. La problématique était initialement de déterminer si et comment nous pouvions assurer la sécurité des individus en permettant d'alerter les autorités compétentes le plus rapidement possible. La thèse a ainsi pu disposer d'un moyen de validation des résultats au travers de la présence d'une école primaire parmi les cas d'études.

Une grande flexibilité dans les réflexions autour de solutions répondant à la problématique a été nécessaire pour pouvoir faire face au différent changement de stratégie du projet. Bien que le cœur du problème est resté le même tout au long du projet à savoir la sécurisation des espaces urbains, les moyens permettant d'attendre cet objectif ont énormément évolué. En effet, la sécurisation devait initialement se faire au travers de caméra ajouter dans les lieux participant à l'étude et plus particulièrement dans l'ensemble des pièces de l'école.

Les différentes prises de décisions et réflexion menée autour du projet nous ont imposés à revoir l'entièreté des solutions que nous voulions mettre en place. La seule solution envisageable à l'issue des dernières restrictions était une étude approfondie non pas de l'intérieur des lieux à surveiller, mais de l'extérieur des lieux en utilisant le matériel de surveillance urbain déjà en place. C'est en cherchant à atteindre cet objectif que les solutions imaginées et mises en œuvre dans cette thèse ont été réalisées. Le projet SafeCity n'est toutefois pas uniquement centré autour

de l'analyse vidéo , nous retrouvons notamment dans ce projet la réalisation de plusieurs éléments à savoir :

- Un dispositif de contrôle des entrées sous forme d'un portail intelligent,
- un dispositif portatif permettant le passage ou non au travers de ce portail ainsi que d'autres fonctionnalités de géolocalisation en cas d'évacuation de l'établissement
- un dispositif d'analyse sonore sous forme d'une "alerte box" permettant aux personnes présentes dans les salles de rapidement être informées d'une situation dangereuse ou de pouvoir facilement indiquer la présence d'une situation dangereuse
- un dispositif de surveillance vidéo séparé en 3 modules : Un module de reconnaissance faciale, un module de détection de franchissement des barrières et un module d'analyse comportementale.

L'ensemble des éléments ont été réalisé par différents prestataires et a pu être validé par les différents demandeurs à l'issue du projet.

Il est ressorti de ce projet que les résultats obtenus sur l'analyse comportementale permettent une détection suffisamment efficace pour permettre la levée d'une alerte rapide auprès des autorités compétentes. Le projet n'ayant été réalisé qu'à titre démonstratif aucune suite n'a été exigée ou envisagée par les demandeurs. Tous les travaux présentés, en dehors du chapitre 3, ont été réalisés de sorte à améliorer le démonstrateur qui a été validé par le client. Toutefois, l'ensemble de ces autres travaux nous ont permis de mettre en avant la possibilité d'une méthode de suivi individualisé utilisable en environnement urbain au travers d'identifications vidéo anonymes.

Les résultats de ces différentes recherches et expérimentations mettent donc en avant la possibilité de créer un système d'analyse comportementale individualisé dans un contexte de surveillance urbain. Les différentes conclusions de cette thèse mettent en avant qu'il est possible d'identifier des individus dans une image et de n'en conserver que leurs tailles et positions afin de faire un suivi. Nous pouvons également, au travers de ces mêmes conclusions, envisager d'utiliser les informations de ce suivi pour alimenter un système d'analyse comportemental urbain encore plus performant que celui présentait dans le cadre du projet

SafeCity. L'amélioration des performances étant dépendante de l'intégration des données de suivi individualisé, de la puissance de calcul disponible pour la détection et l'analyse, de l'amélioration du système de détection vidéo qui au moment de l'étude utilisait YoloV3. Aujourd'hui, ce système d'analyse vidéo pourrait être amélioré grâce à l'utilisation de YoloV5 qui, même s'il est légèrement plus lent que YoloV3, à la possibilité d'effectuer de meilleures déductions [KBM22]. Au moment de l'écriture de cette thèse, une version YoloV7 a vu le jour et propose déjà de premiers résultats [WBL22].

## 5.2 Perspectives simulateur et analyseur

les recherches menées lors de cette thèse sur les différents composants de notre système de détection de comportement anormal nous ont menés à développer deux éléments autour desquels les travaux de recherche ont principalement gravité. Lors du développement de ces deux éléments, seul un certain nombre de fonctionnalités ont pu être développées. En effet en établissant ce qu'il était possible de faire au travers de la littérature, certains axes de recherche ont pu être envisagés, mais n'ayant pas eu suffisamment de temps pour les réaliser certains d'entre eux ont été mis en attente.

### 5.2.1 Simulateur

Dans sa version la plus finalisée, le simulateur permettait d'accomplir des simulations d'individus autogérées grâce à un certain nombre de règles. L'ensemble des règles qui constituent l'autogestion de ces individus ne sont pas strictes et disposent d'un coefficient qui permettra d'appliquer avec plus ou moins d'intensité la règle voulue. Ces coefficients s'apparentent donc à une sorte de contrainte psychologique de l'individu évoluant dans la simulation. En effet, le fait d'ajuster ces différents coefficients peut rendre l'individu plus ou moins indépendant de son groupe ou encore plus ou moins craintif des individus des autres groupes. Ce coefficient est défini arbitrairement en fonction de la simulation voulue afin de faire respecter de manière plus ou moins forte les règles établies. Dans la mesure où ce paramétrage est un paramétrage préalable, ces coefficients sont donc fixes et ils resteront ainsi tout au long de la simulation. Un des axes d'amélioration du simu-



lateur serait donc la possibilité d’ajuster les différents coefficients tout au long de la simulation afin de créer des comportements plus aléatoires et moins robotiques.

Nous pouvons envisager plusieurs solutions afin de répondre à ce besoin :

- Faire un ensemble de tests sur des valeurs de coefficients proposés aléatoirement sous le contrôle d’un individu qui serait capable d’identifier les valeurs pour lesquelles les comportements semblent les plus humains possibles.
- Faire évoluer l’ensemble des coefficients de manière logique à l’aide de fonctions permettant des variations douces et naturelles.

Dans le premier cas, nous serions contraints de créer une base de données de valeur intéressante à exploiter qui servirait de base au simulateur pour générer ces scénarios. Cette solution augmenterait le nombre et la diversité des scénarios possibles, mais poserait une contrainte fondamentale. En effet afin de sélectionner les différents scénarios il faut donc envisager la présence d’une personne ou d’une équipe capable d’identifier les comportements semblant naturels au travers de leur propre opinion. C’est un travail rébarbatif et long qui ne permettrait de n’isoler qu’une quantité limitée de scénarios.

Dans le second cas, les scénarios restent aléatoires dans la mesure où les valeurs sont modifiées de manière aléatoire. Toutefois, cette solution implique que les scénarios observés peuvent ne plus être cohérents. Même si les valeurs sont bornées afin de rester dans des plages de valeurs qui ont pu sembler cohérentes à l’observation, nous ne pourrions pas nous assurer au vu de la quantité importante de possibilités que toutes les combinaisons de valeur paraîtront naturelles.

Une solution idéale serait d’avoir la possibilité de constituer des scénarios de manière aléatoire tout en conservant une évolution humaine et naturelle dans les déplacements.

### **5.2.2 Analyseur**

Tout comme le simulateur, lors du livrable projet, l’analyseur faisait encore l’objet de développements dans le cadre de la thèse. Les travaux qui ont été me-

nés durant la thèse ont principalement eu pour objectif de déterminer comment l'analyseur pouvait être renforcé à l'aide d'un suivi individualisé. Ce dernier dont l'objectif était initialement de pouvoir identifier un comportement anormal au travers d'un ensemble de points évoluant à travers le temps s'est vu renforcé à l'aide de notions de suivi individualisé.

Parmi l'ensemble des méthodes existantes afin de réaliser du suivi, les deux présentées dans cette thèse ont été traitées dans le but de déterminer leur efficacité. Elles ne sont pas les seules méthodes existantes aujourd'hui pour faire du suivi, mais elles sont toutefois innovantes dans leur manière de traiter un suivi urbain dans la mesure où ces derniers sont généralement réalisés directement sur la vidéo.

Le fait d'être indépendant de la vidéo peut rendre l'analyseur compatible à une multitude de cas d'étude. Son objectif étant d'associer les points d'un premier nuage de points à second nuage de points de sorte à créer un suivi, il est alors possible d'utiliser cette méthode pour d'autres cas que celui du suivi urbain. Il est déjà apparu dans la littérature que des méthodes apparentes peuvent être utilisées dans le cadre de la prédiction de mouvement de bateaux [AH83]. L'application de cette méthode dans le suivi urbain permet toutefois d'envisager des perspectives afin d'améliorer l'analyse comportementale. En effet, le fait de disposer d'un suivi fiable permet également de bénéficier d'informations sur la trajectoire et la vitesse des différents individus. Dans le cas de l'analyse comportementale, nous avons à traiter trois cas bien spécifiques :

- La dispersion,
- le rassemblement,
- le maraudage.

Les expérimentations mettant en avant la détection de dispersion et de rassemblement montrent que l'efficacité du système peut être mise en cause dans les cas où plusieurs événements anormaux se produisent en simultanément. Le fait de disposer d'un suivi individualisé permet d'ajouter des éléments supplémentaires à nos analyses. En effet, les points composant le nuage de points peuvent désormais être potentiellement suivis à l'aide des méthodes de suivi présentées dans cette thèse. Ce suivi amène ainsi des caractéristiques de vitesse et de trajectoire à chacun des

points composant les nuages.

En prenant en considération la possibilité d'utiliser pour chaque point ces deux nouveaux paramètres, nous pouvons alors mener des études plus complexes sur la trajectoire emprunter par les différents individus. Certaines méthodes, déjà existantes, permettent de faire de la prédiction de trajectoire [Ala+16], le fait d'utiliser une de ces méthodes où d'adapter notre propre méthode basée sur les filtres de Kalman par exemple, nous permettrait de détecter de nouveaux cas.

Dans le cas d'une fuite unidirectionnelle, notre analyseur serait incapable de détecter la dispersion si l'ensemble des individus part dans la même direction. En effet, ce dernier étant basé sur l'évolution de la distance entre chacun des membres des différents groupes, si cet écart n'évolue pas ou peu, le seuil de détection peut ne pas être franchi. En revanche, dans le cas où nous avons associé à chaque point une vitesse de déplacement cela implique alors que nous sommes également capables de connaître l'accélération de chaque individu. Auquel cas, nous sommes alors capables de déceler la dégradation d'une situation au travers de l'accélération des différents individus. Si cette accélération est trop importante et appliquée à de trop nombreuses personnes, cela pourra traduire le comportement de fuite d'une situation ou d'un individu.

Un autre cas particulier de dispersion que notre analyseur n'est actuellement pas capable de détecter peut se produire si une dispersion hétérogène a lieu en même temps qu'un rassemblement. Il est parfois impossible pour l'analyseur de détecter les deux événements, car la variation de l'écart type s'équilibre entre son augmentation due à la dispersion et sa diminution due au rassemblement. Le fait de disposer des informations d'accélération et donc de détecter un comportement issu des accélérations permettrait également de corriger ce cas.

En ce qui concerne le cas d'étude liée à la détection de rassemblement, il s'est vite avéré que ce dernier n'avait pour but de ne détecter que la présence d'individus durant des plages horaires spécifiques. Il n'y aurait eu dans ce cas que peu d'intérêt à améliorer cette méthode. Toutefois si nous le souhaitions nous pourrions par exemple envisager de détecter la variation de la vitesse de X individus à

l'approche d'un groupe ou d'un autre individu dans l'image. Auquel cas, si nous décelons un ralentissement, cela voudra potentiellement dire que les individus chercheront à intégrer le groupe ou les individus présents dans l'image.

Le cas que nous cherchions à détecter pour lequel l'utilisation de paramètres de vitesse et d'accélération a une importance capitale serait le maraudage. En effet, dans le cadre du maraudage nous utilisons une méthode se basant sur les heatmaps afin de déceler les zones sur lesquelles un individu est resté un certain temps. Si cet individu a tendance à ne pas rester en place et à occuper l'espace, il est alors difficile pour l'analyseur de détecter le maraudage. Toutefois, en étudiant non plus les positions occupées dans l'espace, mais le déplacement des individus nous sommes alors capables de dire si la position de l'individu au cours du temps a évolué de manière suffisamment significative pour ne pas être considérée comme une menace ou si ce dernier est resté trop statique par rapport à la zone observée et dans ce cas détecter une anomalie. En procédant ainsi, nous pourrions également empêcher une problématique que nous avons pu identifier, mais que nous n'avons pas pu observer lors des phases de test : si plusieurs individus empruntent un même chemin dont l'espace de marche est restreint et oblige donc tous les individus à chauffer la même zone de la heatmap, il est en théorie possible, si le nombre d'individus est suffisant, de détecter un maraudage juste parce que ce chemin serait abondamment emprunté.

De manière générale, l'utilisation de paramètres permettant la détection de la vitesse et de l'accélération permet également d'amener de nouveaux cas d'étude. Nous pouvons par exemple imaginer détecter les changements de trajectoire brusques ou potentiellement détecter des changements de trajectoire inhabituels pour un humain. De nombreuses études montrent que le déplacement humain, même s'il est régi par la décision de ce dernier, a une tendance à être influencé par l'environnement d'une part, mais aussi par la volonté de ce dernier à parcourir la distance la plus courte possible pour atteindre son objectif [BJ03 ; Zho+20 ; KG14]. Dans ce cas, nous pouvons imaginer qu'une personne qui ne cherche pas à avancer selon un schéma spécifique n'est peut-être pas en train de simplement se déplacer, mais peut-être en train d'essayer d'observer quelque chose. Il est alors envisageable d'utiliser ce genre de système de détection dans des lieux sensibles tels que les

écoles, jardin d'enfants ou tout autre lieu pouvant être la cible d'acte malveillant.

### 5.2.3 Intelligence Artificielle

Comme le montrent de nombreuses références de cette thèse, de nos jours énormément de solutions informatiques tournent autour de l'intelligence artificielle. Certaines des méthodes abordées dans cette thèse sont également utilisées dans le domaine de l'intelligence artificielle.

En cherchant à améliorer les différentes méthodes proposées dans cette thèse, il paraît évident que certains éléments peuvent être améliorés en utilisant des systèmes d'intelligence artificielle. Notre volonté initiale n'était pas d'utiliser ce genre de système qui pouvait être plus coûteux, car nous devions pouvoir nous plier aux différentes exigences du projet et notamment à une architecture dont les performances n'étaient que peu connues. Toutefois, avec une meilleure connaissance du système disponible pour faire les différents calculs nous pouvons envisager d'utiliser des systèmes du domaine de l'intelligence artificielle.

L'intelligence artificielle est aujourd'hui employée dans de nombreux cas, il est donc important de déterminer celle qui nous permettrait d'améliorer nos méthodes actuelles. L'aspect intéressant de l'intelligence artificielle pourrait être, dans notre cas, tout ce qui se base sur les apprentissages automatiques ou aux réseaux de neurones. Ces méthodes d'intelligence artificielle consistent au travers d'un apprentissage, à faire des déductions sur une donnée qui n'a pas encore été apprise. C'est une méthode qui impose un certain nombre de contraintes, mais qui nous permettrait avec une certaine approche d'obtenir des déductions supplémentaires dans nos travaux.

L'utilisation d'une méthode RNN (réseau de neurones reculant) pourrait être une première base de réflexion. L'objectif est de prédire un événement à venir en se basant sur les informations passées [Fuj+21]. Nous pourrions alors essayer de déterminer la position à venir des individus dans l'image à l'aide des informations de vitesse et de trajectoire ou directement à partir des informations de position. Nous retrouvons alors une corrélation avec la méthode présentée dans cette thèse utilisant le filtre de Kalman. Il pourrait donc être intéressant de comparer sur le

plan de la vitesse et de la performance c'est deux méthodes afin de voir si l'une des deux se distingue. Un autre axe intéressant pourrait être une approche hybride où une des deux méthodes alimente l'autre afin de potentiellement réduire le temps de traitement et d'augmenter la précision des résultats.

Un dernier axe de recherche envisageable serait la possibilité d'identifier un ensemble de scénarios anormaux. Nous pouvons par exemple imaginer :

- La détection de changement de trajectoire brusque,
- la détection d'accélération brusque,
- la détection de deux personnes courant l'une derrière l'autre...

En identifiant cet ensemble de scénarios, nous pouvons ensuite imaginer utiliser le simulateur proposer dans cette thèse afin de générer une multitude de simulations pour chaque scénario et ainsi composer une base de données de comportements anormaux. En effet, une des premières phases de réflexion du projet nous a amenés à imaginer un système se basant sur les réseaux neuronaux qui seraient entraînés à détecter les comportements dangereux. Toutefois, ce système a rapidement été écarté des solutions, car il nous a paru trop complexe de déterminer un ensemble de comportements à détecter.

En revanche, il est possible d'imaginer d'utiliser une telle solution en ciblant des cas très spécifiques, la problématique serait alors de constituer une base de données suffisamment importante en événements normaux et anormaux, car un réseau de neurones aura besoin d'être entraîné pour tous les cas que l'on souhaitera lui faire déterminer. La décision finale du réseau pourra être binaire (il y a un événement anormal ou il n'y a pas d'événement anormal) où plus nuancer si l'entraînement a été suffisamment fin, nous pourrions obtenir des réponses précises sur le type d'anomalie qui est en train de se produire. Cette méthode est à la base de ce que l'on appelle la maintenance prédictive où l'objectif est de détecter une anomalie avant qu'elle se produise en se basant sur les paramètres normaux de cette dernière [Cho+20; FZS19]. Une autre problématique à résoudre ici sera de savoir s'il est possible de constituer une base de données suffisamment générique en se basant sur les données dont nous disposons pour pouvoir faire de la prédiction dans les comportements humains.



# CONCLUSION GÉNÉRALE

---

La sécurisation des citoyens demeure un élément essentiel dans la volonté de développement des villes. Cette sécurisation peut se faire par le renforcement de ressources humaines, mais aussi par la bonne utilisation de dispositifs numériques innovants. Il demeure toutefois un aspect important dans cette quête de la sécurité qui est le respect de la vie privée de chacun. Le fait de faire transiter des informations permettant d'identifier un individu à tout moment du temps demeure un acte sensible présentant dans les cas les plus extrêmes une mise en danger de ce dernier.

Les travaux menés dans cette thèse permettent d'envisager des solutions de détection des dangers préservant l'anonymat de tous. Ces solutions accompagnées d'équipements suffisamment performants pourront potentiellement être améliorées de sorte à pouvoir interpréter l'activité de tout individu étant détectable par une caméra de vidéosurveillance.

Les travaux menés durant cette thèse ont permis d'explorer plusieurs outils se basant sur des domaines variés. La conclusion globale de leur utilisation montre qu'il est possible d'envisager un suivi anonyme en environnement urbain. Il est de plus en plus certain que de nombreuses villes vont tendre à s'équiper de systèmes de vidéosurveillance et le respect de cet anonymat demeurera une volonté de l'amélioration continue de la sécurité. L'usage des données personnel et le traitement de ces dernières sont protégés par le règlement général sur la protection des données (RGPD) afin d'assurer qu'une personne physique ne puisse être identifiée formellement contre sa volonté. Cette thèse permet donc de respecter cette obligation.





# **FICHE TECHNIQUE AXIS M1123-E NETWORK CAMERA**

---

## AXIS M1125-E Network Camera

<b>Caméra</b>		<b>Ressources</b>	Compteur de pixels
<b>Capteur</b>	Capteur CMOS RVB à balayage progressif 1/3"	<b>intégrées d'aide à l'installation</b>	
<b>Objectif</b>	Vari focal, correction infrarouge, monture CS, diaphragme DC, 3-10,5 mm Champ de vision horizontal : 91°-32° Champ de vision vertical : 49°-18°	<b>Général</b>	
<b>Jour et nuit</b>	Filtre IR à retrait automatique	<b>Boîtier</b>	Boîtier certifié IK10 en polymère résistant aux chocs, conforme aux normes IP66 et NEMA 4X Couleur : blanc NCS S 1002-B
<b>Éclairage minimum</b>	Couleur : 0,25 lux F1.4 N/B : 0,05 lux F1.4	<b>Développement Durable</b>	Sans PVC Circuit imprimé exempt de brome
<b>Durée d'obturation</b>	1/66500 à 1 s	<b>Mémoire</b>	RAM de 512 Mo, mémoire Flash de 256 Mo
<b>Vidéo</b>		<b>Alimentation</b>	8-28 V CC ou alimentation par Ethernet (PoE) IEEE 802.3af Classe 2, max. 5,0 W, type 3,6 W
<b>Compression vidéo</b>	Profil Baseline, Main et High H.264 (MPEG-4 Part 10/AVC) Motion JPEG	<b>Connecteurs</b>	RJ45 10BASE-T/100BASE-TX PoE Bloc terminal pour une entrée et une sortie (sortie 12 V CC, charge max. 15 mA) Entrée CC, bloc terminal
<b>Résolutions</b>	1920x1080 à 160x90	<b>Stockage</b>	Compatible avec les cartes microSD/microSDHC/microSDXC Cryptage de la carte SD Prise en charge de l'enregistrement sur une unité de stockage réseau (NAS) Pour des recommandations sur les cartes SD et le stockage NAS, voir <a href="http://axis.com">axis.com</a>
<b>Fréquence d'image</b>	Jusqu'à 25/30 ips dans toutes les résolutions	<b>Conditions d'utilisation</b>	De -20°C à 50°C (-4°F à 122°F) Humidité relative 10 à 100 % (avec condensation)
<b>Flux vidéo</b>	Flux multiples, configurables individuellement en H.264 et Motion JPEG Technologie Axis Zipstream en H.264 Fréquence d'image et bande passante contrôlables, VBR/ABR/MBR H.264	<b>Conditions de stockage</b>	De -20°C à 50°C (-4°F à 122°F) Humidité relative de 5 à 95 % (sans condensation)
<b>Réglages de l'image</b>	WDR – Forensic capture : jusqu'à 120 dB selon la scène, réglage manuel de la vitesse d'obturation, compression, couleur, luminosité, netteté, contraste, balance des blancs, contrôle d'exposition (y compris contrôle du gain automatique), zones d'exposition, réglage de précision du comportement en faible éclairage, rotation : 0°, 90°, 180°, 270°, correction des rapports d'aspect, texte et images en surimpression, masque de confidentialité, duplication des images	<b>Homologations</b>	<b>CEM</b> EN 55022 Classe A, EN 61000-6-1, EN 61000-3-2, EN 61000-3-3, EN 61000-6-2, EN 55024, EN 50121-4, FCC Partie 15 Sous-partie B Classe A, ICES-003 Classe A, VCCI Classe A, RCM AS/NZS CISPR 22 Classe A, KCC KN22 Classe A, KN24 <b>Sécurité</b> IEC/EN/UL 60950-22 IS 13252 <b>Environnement</b> EN 50581, IEC 60068-2-1, IEC 60068-2-2, IEC 60068-2-14, IEC 60068-2-30, IEC 60068-2-78, IEC/EN 60529 IP66, NEMA 250 Type 4X, IEC/EN 62262 IK10, IEC 60721-4-4 Classe 4M4 <b>Réseau</b> NIST SP500-267
<b>Panoramique/inclinaison/zoom</b>	PTZ numérique	<b>Dimensions</b>	222 x 137 x 448 mm (8 3/4 x 5 3/8 x 17 5/8 po), support mural compris
<b>Réseau</b>		<b>Poids</b>	1,6 kg (3,5 lb), support mural compris
<b>Sécurité</b>	Protection par mot de passe, filtrage d'adresses IP, cryptage HTTPS <sup>®</sup> , cryptage, contrôle d'accès réseau IEEE 802.1X (EAP-TLS) <sup>®</sup> , authentification digest, journal d'accès utilisateur, protection contre les attaques par force brute, firmware signé	<b>Accessoires fournis</b>	AXIS T94Q01A Wall Mount, outil Resistorx T20, kit de connexion, Guide d'installation, 1 licence utilisateur décodeur Windows <sup>®</sup>
<b>Protocoles pris en charge</b>	IPv4, IPv6, USGv6, HTTP, HTTPS <sup>®</sup> , SSL/TLS <sup>®</sup> , QoS Layer 3 DiffServ, FTP, SFTP, CIFS/SMB, SMTP, Bonjour, UPnP <sup>™</sup> , SNMP v1/v2c/v3 (MIB-II), DNS, DynDNS, NTP, RTSP, RTP, TCP, UDP, IGMP, RTCP, ICMP, DHCP, ARP, SOCKS, SSH, LLDP, MQTT v3.1.1	<b>Accessoires en option</b>	AXIS T94R01P Conduit Back Box, AXIS T98A16-VE Surveillance Cabinet Series, AXIS T91A47 Pole Mount, AXIS T90B Illuminators
<b>Intégration système</b>		<b>Logiciel de gestion vidéo</b>	AXIS Camera Companion, AXIS Camera Station, logiciel de gestion vidéo des partenaires de développement d'applications d'Axis disponibles sur <a href="http://axis.com/vms">axis.com/vms</a>
<b>Interface de programmation</b>	API ouverte pour l'intégration logicielle, avec VAPIX <sup>®</sup> et AXIS Camera Application Platform ; caractéristiques sur <a href="http://axis.com">axis.com</a> AXIS Video Hosting System (AVHS) avec connexion en un clic Profil 8 ONVIF <sup>®</sup> , Profil S ONVIF <sup>®</sup> et Profil T ONVIF <sup>®</sup> . Caractéristiques disponibles sur <a href="http://onvif.org">onvif.org</a>	<b>Langues</b>	Anglais, allemand, français, espagnol, italien, russe, chinois simplifié, japonais, coréen, portugais, chinois traditionnel
<b>Vidéo intelligente</b>	<b>Fournis</b> AXIS Vidéo Motion Detection, alarme de sabotage <b>Compatibles</b> Prise en charge de AXIS Camera Application Platform permettant l'installation d'applications tierces, voir <a href="http://axis.com/acap">axis.com/acap</a>	<b>Garantie</b>	Garantie Axis de 3 ans et option de garantie prolongée d'AXIS, voir <a href="http://axis.com/warranty">axis.com/warranty</a>
<b>Déclencheurs d'événements</b>	Outils d'analyse, événements de stockage local, entrée externe	a. Ce produit inclut un logiciel développé par le projet OpenSSL pour une utilisation dans la boîte à outils OpenSSL ( <a href="http://www.openssl.org">www.openssl.org</a> ), ainsi qu'un logiciel de cryptographie développé par Eric Young ( <a href="http://ey@cryptsoft.com">ey@cryptsoft.com</a> ).	
<b>Actions sur événement</b>	Chargement de fichiers par FTP, HTTP, HTTPS, SFTP, partage réseau et e-mail Notification par courrier électronique, HTTP, HTTPS, TCP enregistrement vidéo sur stockage local, mémorisation d'images pré/post-alarme, PTZ préréglé, tour de garde, envoi de clip vidéo, envoi d'un message trap SNMP, mode de vision jour/nuit, mode WDR, voyant d'état, port de sortie	Responsabilité environnementale : <a href="http://axis.com/environmental-responsibility">axis.com/environmental-responsibility</a>	
<b>Flux de données</b>	Données d'événements		

# LISTE DES TABLEAUX

---

1.1	Objectifs fonctionnels des expérimentations . . . . .	15
3.1	Évolution de l'écart type sur une suite d'images . . . . .	92
3.2	Utilisation d'un buffer de lissage sans seuil dans l'étude de la variation de l'écart type . . . . .	93
3.3	Utilisation d'un buffer de lissage avec seuil dans l'étude de la variation de l'écart type . . . . .	94
3.4	Comparaison des buffers avec et sans seuil d'intégration . . . . .	94
4.1	Caractéristiques des vidéos . . . . .	109
4.2	Nombre de décrochage par vidéo . . . . .	119
4.3	Comparaisons entre la prédiction et la réalité . . . . .	126
4.4	Zone commune entre la réalité et la prédiction . . . . .	128
4.5	Décrochages totaux dans le suivi avec le filtre de Kalman . . . . .	129
4.6	Comparaison du nombre de décrochages entre deux versions . . . . .	129
4.7	Comparaison des décrochages à une fréquence variable d'IPS . . . . .	130
4.8	Comparaison des décrochages toutes vidéos . . . . .	130

# TABLE DES FIGURES

---

1.1	Entités de la sécurité . . . . .	11
1.2	Les grandes fonctions . . . . .	12
1.3	Architecture générale . . . . .	13
1.4	Architecture générale . . . . .	16
1.5	Architecture Analyse comportementale . . . . .	17
2.1	Camera cible . . . . .	31
2.2	Représentation de l'angle $\theta$ . . . . .	36
2.3	Détection couteau YOLOv3 . . . . .	38
2.4	Détection couteau ou télécommande YOLOv3 . . . . .	39
2.5	Utilisation de mires . . . . .	42
2.6	Caméra secondaire . . . . .	43
2.7	Mires complexifié . . . . .	43
2.8	Décalage détection de mires . . . . .	44
2.9	Définition de trois chemins en point de passage . . . . .	52
2.10	Suivi de chemin point de passage . . . . .	53
2.11	Trajectoire pour trois points . . . . .	55
2.12	Définition d'un chemin Avec les courbes d'interpolation de Lagrange . . . . .	56
2.13	Trajectoire utilisant les courbes de Bézier . . . . .	57
2.14	Exclusion des comportements aberrants . . . . .	58
2.15	Ralentissement dans les courbes . . . . .	59
2.16	Comparaison des fenêtres de détection . . . . .	62
2.17	Oscillation dans la hauteur . . . . .	63
2.18	Oscillation dans la largeur . . . . .	63
2.19	Temps de traitement du simulateur en première version . . . . .	66
2.20	Gestion des processus dans la première version du simulateur . . . . .	68
2.21	Comparaison de la gestion des processus . . . . .	71
2.22	Temps de traitement du simulateur dans sa nouvelle version . . . . .	72

---

2.23	Comparaison de dessus et en perspective . . . . .	73
2.24	Ajout de perspective à la zone de simulation . . . . .	76
3.1	Chronologie d'obtention des données . . . . .	81
3.2	Principe de fonctionnement de K-Means . . . . .	86
3.3	Principe de fonctionnement de DBSCAN . . . . .	87
3.4	Cas particulier départ du groupe . . . . .	91
3.5	Cas particulier groupement en plusieurs temps . . . . .	91
3.6	Illustration de l'utilisation de DBSCAN dans l'analyseur . . . . .	97
3.7	Utilisation des diagrammes de Voronoï . . . . .	98
3.8	Illustration de l'utilisation DBSCAN et de Voronoï . . . . .	99
3.9	Visualisation de la matrice heatmap . . . . .	104
3.10	Comparaison entre réalité et la vision heatmap . . . . .	105
4.1	Suivi en fonction du temps sur les axes x et y . . . . .	110
4.2	Suivi en fonction du temps sur les axes x et y (agrandi) . . . . .	111
4.3	Visualisation détection vidéo et données anonymisées . . . . .	113
4.4	Rappel classification par DBSCAN . . . . .	114
4.5	Fonctionnement suivi par proximité . . . . .	115
4.6	Cas particulier d'identification . . . . .	116
4.7	Décrochage du suivi . . . . .	117
4.8	Augmentation du nombre d'individus détecté . . . . .	118
4.9	Illustration de suivi avec filtre de Kalman . . . . .	122
4.10	Fonctionnement suivi par Kalman . . . . .	123

# BIBLIOGRAPHIE

---

- [AA15] Ruben ARMANANZAS et Giorgio A. ASCOLI, « Towards the automatic classification of neurons », in : *Trends in Neurosciences* 38 (5 mai 2015), p. 307-318, ISSN : 0166-2236, DOI : 10.1016/J.TINS.2015.02.004.
- [AAA19] Misbah AHMAD, Imran AHMED et Awais ADNAN, « Overhead View Person Detection Using YOLO », in : *2019 IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2019* (oct. 2019), p. 0627-0633, DOI : 10.1109/UEMCON47517.2019.8992980.
- [AH83] Vincent J. AIDALA et Sherry E. HAMMEL, « Utilization of Modified Polar Coordinates for Bearings-Only Tracking », in : *IEEE Transactions on Automatic Control* 28 (3 1983), p. 283-294, ISSN : 15582523, DOI : 10.1109/TAC.1983.1103230.
- [AK21] Gokulnath ANAND et Ashok Kumar KUMAWAT, « Object detection and position tracking in real time using Raspberry Pi », in : *Materials Today : Proceedings* (juill. 2021), ISSN : 2214-7853, DOI : 10.1016/J.MATPR.2021.06.437.
- [Ala+16] Alexandre ALAHI et al., « Social LSTM : Human trajectory prediction in crowded spaces », in : *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-December* (déc. 2016), p. 961-971, ISSN : 10636919, DOI : 10.1109/CVPR.2016.110.
- [Asa17] Akinori ASAHARA, « Decomposition of pedestrian flow heatmap obtained with monitor-based tracking », in : IEEE, 2017, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/8115869/>.
- [ASI20] Mohiuddin AHMED, Raihan SERAJ et Syed Mohammed Shamsul ISLAM, « The k-means Algorithm : A Comprehensive Survey and Performance Evaluation », in : *Electronics 2020, Vol. 9, Page 1295* 9 (8 août 2020), p. 1295, ISSN : 2079-9292, DOI : 10.3390/ELECTRONICS9081295, URL :

---

<https://www.mdpi.com/2079-9292/9/8/1295/htm%20https://www.mdpi.com/2079-9292/9/8/1295>.

- [Aug+21] Amaury AUGUSTE et al., « Implementation of a behavioral analysis method of crowd movement in the service of video surveillance », in : (avr. 2021), p. 18, ISSN : 1996756X, DOI : 10.1117/12.2586946, URL : <https://hal.archives-ouvertes.fr/hal-03207774>.
- [Aur91] Franz AURENHAMMER, « Voronoi diagrams—a survey of a fundamental geometric data structure », in : *ACM Computing Surveys (CSUR)* 23 (3 jan. 1991), p. 345-405, ISSN : 15577341, DOI : 10.1145/116873.116880.
- [Bar83] Stephen T. BARNARD, « Interpreting perspective images », in : *Artificial Intelligence* 21 (4 nov. 1983), p. 435-462, ISSN : 0004-3702, DOI : 10.1016/S0004-3702(83)80021-6.
- [BJ03] D. C. BROGAN et N. L. JOHNSON, « Realistic human walking paths », in : *Proceedings - IEEE Workshop on Program Comprehension 2003-January (2003)*, p. 94-101, ISSN : 10928138, DOI : 10.1109/CASA.2003.1199309.
- [Buq+21] Julie BUQUET et al., « Evaluating the impact of wide-angle lens distortion on learning-based depth estimation », in : *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (juin 2021)*, p. 3688-3696, ISSN : 21607516, DOI : 10.1109/CVPRW53098.2021.00409.
- [BV00] Christian BRÄUER-BURCHARDT et Klaus VOSS, « Automatic Lens Distortion Calibration Using Single Views », in : (2000), p. 187-194, DOI : 10.1007/978-3-642-59802-9\_24, URL : [https://link.springer.com/chapter/10.1007/978-3-642-59802-9\\_24](https://link.springer.com/chapter/10.1007/978-3-642-59802-9_24).
- [Car15] Miguel A. CARREIRA-PERPINAN, « A review of mean-shift algorithms for clustering », in : (mars 2015), DOI : 10.48550/arxiv.1503.00687, URL : <https://arxiv.org/abs/1503.00687v1>.
- [CB10] Zenon CHACZKO et Robin BRAUN, « Teaching computer vision for telemedicine systems using openCV », in : *2010 9th International Conference on Information Technology Based Higher Education and Training, ITHET 2010 (2010)*, p. 17-20, DOI : 10.1109/ITHET.2010.5480069.



- 
- [Cha+18] G. CHANDAN et al., « Real Time Object Detection and Tracking Using Deep Learning and OpenCV », in : *Proceedings of the International Conference on Inventive Research in Computing Applications, ICIRCA 2018* (déc. 2018), p. 1305-1308, DOI : 10.1109/ICIRCA.2018.8597266.
- [Cho+20] Yulim CHOI et al., « Method of predictive maintenance for induction furnace based on neural network », in : *Proceedings - 2020 IEEE International Conference on Big Data and Smart Computing, BigComp 2020* (fév. 2020), p. 609-612, DOI : 10.1109/BIGCOMP48618.2020.00021.
- [CSL12] Lei CHEN, Narasimha SHASHIDHAR et Qingzhong LIU, « Scalable secure MJPEG video streaming », in : *Proceedings - 26th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2012* (2012), p. 111-115, DOI : 10.1109/WAINA.2012.163.
- [Dat+12] Mayur DATAR et al., « Maintaining Stream Statistics over Sliding Windows », in : <https://doi.org/10.1137/S009753970139836331> (6 fév. 2012), p. 1794-1813, ISSN : 00975397, DOI : 10.1137/S0097539701398363, URL : <https://epubs.siam.org/doi/10.1137/S0097539701398363>.
- [Den20] Dingsheng DENG, « DBSCAN Clustering Algorithm Based on Density », in : *Proceedings - 2020 7th International Forum on Electrical Engineering and Automation, IFEEA 2020* (sept. 2020), p. 949-953, DOI : 10.1109/IFEEA51475.2020.00199.
- [Don+19] Gangsong DONG et al., « DB-Kmeans :An Intrusion Detection Algorithm Based on DBSCAN and K-means », in : IEEE, 2019, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/8892910/>.
- [DZ21] Zoubaydat DAHIROU et Mao ZHENG, « Motion Detection and Object Detection : Yolo (You only Look Once) », in : *Proceedings - 2021 7th Annual International Conference on Network and Information Systems for Computers, ICNISC 2021* (2021), p. 250-257, DOI : 10.1109/ICNISC54316.2021.00053.
- [Ess+10] Rene Jean ESSIAMBRE et al., « Capacity Limits of Optical Fiber Networks », in : *Journal of Lightwave Technology* 28 (4 2010), p. 662-701, ISSN : 15582213, DOI : 10.1109/JLT.2009.2039464.

- 
- [Far+19] Xurshedjon FARHODOV et al., « Faster RCNN Detection Based OpenCV CSRT Tracker Using Drone Data », in : *International Conference on Information Science and Communications Technologies : Applications, Trends and Opportunities, ICISCT 2019* (nov. 2019), DOI : 10.1109/ICISCT47635.2019.9012043.
- [FF18] Sindre FOSSEN et Thor I. FOSSEN, « Extended kalman filter design and motion prediction of ships using live automatic identification system (AIS) Data », in : *Proceedings - 2018 2nd European Conference on Electrical Engineering and Computer Science, EECS 2018* (déc. 2018), p. 464-470, DOI : 10.1109/EECS.2018.00092.
- [Fuj+21] Ryo FUJII et al., « A Two-Block RNN-Based Trajectory Prediction from Incomplete Trajectory », in : *IEEE Access* 9 (2021), p. 56140-56151, ISSN : 21693536, DOI : 10.1109/ACCESS.2021.3072135.
- [FZS19] Vika FARIZA, Zulkarnain ZULKARNAIN et Isti SURJANDARI, « Comparing Artificial Neural Network and Failure Distribution Methods for Maintenance Scheduling : A Case Study of Wooden Door Industry », in : *Proceedings - 2018 5th International Conference on Information Science and Control Engineering, ICISCE 2018* (jan. 2019), p. 161-165, DOI : 10.1109/ICISCE.2018.00043.
- [GAM15] Souhail GUENNOUNI, Ali AHAITOUF et Anass MANSOURI, « Multiple object detection using OpenCV on an embedded platform », in : *Colloquium in Information Science and Technology, CIST 2015-January* (January jan. 2015), p. 374-377, DOI : 10.1109/CIST.2014.7016649.
- [GCN07] János GAL, Andrei CÂMPEANU et Ioan NAFORNITA, « Estimation of chirp signals in gaussian noise by kalman filtering », in : *ISSCS 2007 - International Symposium on Signals, Circuits and Systems, Proceedings 1* (2007), p. 299-302, DOI : 10.1109/ISSCS.2007.4292711.
- [Goo+21] Ehsan Sedgh GOOYA et al., « A human body morphology detector : a distinctive filter to differentiate the image-based process depending on whether the person is a child or an adult », in : (avr. 2021), p. 23, ISSN : 1996756X, DOI : 10.1117/12.2586783, URL : <https://hal.archives-ouvertes.fr/hal-03209362>.

- 
- [HEW19] Mark HEIMBACH, Kamak EBADI et Sally WOOD, « Improving Object Tracking Accuracy in Video Sequences Subject to Noise and Occlusion Impediments by Combining Feature Tracking with Kalman Filtering », in : *Conference Record - Asilomar Conference on Signals, Systems and Computers* 2018-October (fév. 2019), p. 1499-1502, ISSN : 10586393, DOI : 10.1109/ACSSC.2018.8645175.
- [HJW20] Peng HAOJIE, Zhang JIE et Shi WEISHENG, « An Adaptive Cubature Kalman Filter Algorithm for Passive Location », in : IEEE, 2020, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/9317500/>.
- [HMT15] Omar El HARROUSS, Driss MOUJAHID et Hamid TAIRI, « Motion detection based on the combining of the background subtraction and spatial color information », in : *2015 Intelligent Systems and Computer Vision, ISCV 2015* (mai 2015), DOI : 10.1109/ISACV.2015.7105548.
- [Hon+19] Jianqing HONG et al., « Straight track tracking of paddle boat based on kalman filter », in : *Proceedings - 2019 4th International Conference on Mechanical, Control and Computer Engineering, ICMCCE 2019* (oct. 2019), p. 260-265, DOI : 10.1109/ICMCCE48743.2019.00066.
- [Hos+21] Amran HOSSAIN et al., « A YOLOv3 Deep Neural Network Model to Detect Brain Tumor in Portable Electromagnetic Imaging System », in : *IEEE Access* 9 (2021), p. 82647-82660, ISSN : 21693536, DOI : 10.1109/ACCESS.2021.3086624.
- [Hus+06] Mustafa HUSAIN et al., « Dynamic object tracking by partial shape matching for video surveillance applications », in : *Proceedings - International Conference on Image Processing, ICIP (2006)*, p. 2405-2408, ISSN : 15224880, DOI : 10.1109/ICIP.2006.312947.
- [IT17] Ahmad Puad ISMAIL et Nooritawati Md TAHIR, « Human Gait Silhouettes Extraction Using Haar Cascade Classifier on OpenCV », in : IEEE, 2017, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/8359051/>.

- 
- [JC19] David JACKSON et Gary CLYNCH, « An investigation of the impact of language runtime on the performance and cost of serverless functions », in : *Proceedings - 11th IEEE/ACM International Conference on Utility and Cloud Computing Companion, UCC Companion 2018* (jan. 2019), p. 154-160, DOI : 10.1109/UCC-COMPANION.2018.00050.
- [JKH08] Erno JEGES, István KISPÁL et Zoltán HORNÁK, « Measuring human height using calibrated cameras », in : *2008 Conference on Human System Interaction, HSI 2008* (2008), p. 755-760, DOI : 10.1109/HSI.2008.4581536.
- [JKV09] K. G. JOLLY, R. Sreerama KUMAR et R. VIJAYAKUMAR, « A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits », in : *Robotics and Autonomous Systems* 57 (1 jan. 2009), p. 23-33, ISSN : 0921-8890, DOI : 10.1016/J.ROBOT.2008.03.009.
- [KB] Adrian. KAEHLER et Gary R. BRADSKI, « Learning OpenCV 3 : computer vision in C++ with the OpenCV library », in : ( ).
- [KBM22] Aicha KHALFAOUI, Abdelmajid BADRI et Ilham E.L. MOURABIT, « Comparative study of YOLOv3 and YOLOv5's performances for real-time person detection », in : *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2022* (2022), DOI : 10.1109/IRASET52964.2022.9737924.
- [KG14] Christoph G. KELLER et Dariu M. GAVRILA, « Will the pedestrian cross ? A study on pedestrian path prediction », in : *IEEE Transactions on Intelligent Transportation Systems* 15 (2 2014), p. 494-506, ISSN : 15249050, DOI : 10.1109/TITS.2013.2280766.
- [KYK19a] S. Varun KUMAR, P. V. YOKESHRAJ et R. KOTTEESWARAN, « Optimized Inertial Navigation System with Kalman Filter based altitude determination for aircraft in GPS Deprived Regions », in : IEEE, 2019, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/8819646/>.
- [KYK19b] S. Varun KUMAR, P. V. YOKESHRAJ et R. KOTTEESWARAN, « Optimized inertial navigation system with kalman filter based altitude determination for aircraft in GPS deprived regions », in : *Proceedings of the 3rd*

- 
- International Conference on Computing Methodologies and Communication, ICCMC 2019* (mars 2019), p. 672-678, DOI : 10 . 1109 / ICCMC . 2019 . 8819646.
- [Li+20] Min LI et al., « Agricultural Greenhouses Detection in High-Resolution Satellite Images Based on Convolutional Neural Networks : Comparison of Faster R-CNN, YOLO v3 and SSD », in : *Sensors 2020, Vol. 20, Page 4938* 20 (17 août 2020), p. 4938, ISSN : 1424-8220, DOI : 10 . 3390 / S20174938, URL : <https://www.mdpi.com/1424-8220/20/17/4938/> htm%20https://www.mdpi.com/1424-8220/20/17/4938.
- [Luo+15] Lu Ping LUO et al., « Trajectory planning for energy minimization of industry robotic manipulators using the Lagrange interpolation method », in : *International Journal of Precision Engineering and Manufacturing 2015 16 :5 16* (5 mai 2015), p. 911-917, ISSN : 2005-4602, DOI : 10 . 1007 / S12541 - 015 - 0119 - 9, URL : <https://link-springer-com.scd-proxy.univ-brest.fr/article/10.1007/s12541-015-0119-9>.
- [Lux07] Ulrike Von LUXBURG, « A tutorial on spectral clustering », in : *Statistics and Computing 2007 17 :4 17* (4 août 2007), p. 395-416, ISSN : 1573-1375, DOI : 10 . 1007 / S11222 - 007 - 9033 - Z, URL : <https://link-springer-com.scd-proxy.univ-brest.fr/article/10.1007/s11222-007-9033-z>.
- [LZ19] Yewei LIU et Qingrui ZHOU, « Study on group motion based on boids model », in : *Chinese Control Conference, CCC 2019-July* (juill. 2019), p. 6275-6280, ISSN : 21612927, DOI : 10 . 23919 / CHICC . 2019 . 8865691.
- [MB15] Sneha MURMU et Sujata BISWAS, « Application of Fuzzy Logic and Neural Network in Crop Classification : A Review », in : *Aquatic Procedia 4* (jan. 2015), p. 1203-1210, ISSN : 2214-241X, DOI : 10 . 1016 / J . AQPRO . 2015 . 02 . 153.
- [MH16] Saurabh MAHESHWARI et Surbhi HEDA, « A review on crowd behavior analysis methods for video surveillance », in : *ACM International Conference Proceeding Series 04-05-March-2016* (mars 2016), DOI : 10 . 1145 / 2905055 . 2905258, URL : <http://dx.doi.org/10.1145/2905055.2905258>.

- 
- [MJ97] Don MURRAY et Cullen JENNINGS, « Stereo vision based mapping and navigation for mobile robots », in : *Proceedings - IEEE International Conference on Robotics and Automation 2* (1997), p. 1694-1699, ISSN : 10504729, DOI : 10.1109/ROBOT.1997.614387.
- [Mua+04] Anuar Mikdad MUAD et al., « Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system », in : *IEEE Region 10 Annual International Conference, Proceedings/TENCON A* (2004), DOI : 10.1109/TENCON.2004.1414393.
- [MZ18] Amira Ben MABROUK et Ezzeddine ZAGROUBA, « Abnormal behavior recognition for intelligent video surveillance systems : A review », in : *Expert Systems with Applications* 91 (jan. 2018), p. 480-491, ISSN : 0957-4174, DOI : 10.1016/J.ESWA.2017.09.029.
- [NA18] Anif NURROHMAN et Maman ABDUROHMAN, « High performance streaming based on H264 and Real Time Messaging Protocol (RTMP) », in : *2018 6th International Conference on Information and Communication Technology, ICoICT 2018* (nov. 2018), p. 174-177, DOI : 10.1109/ICoICT.2018.8528770.
- [Nak+17] Felipe Kenji NAKANO et al., « Top-down strategies for hierarchical classification of transposable elements with neural networks », in : *Proceedings of the International Joint Conference on Neural Networks 2017-May* (juin 2017), p. 2539-2546, DOI : 10.1109/IJCNN.2017.7966165.
- [Nal+11] M. C. NALDI et al., « Efficiency issues of evolutionary k-means », in : *Applied Soft Computing* 11 (2 mars 2011), p. 1938-1952, ISSN : 1568-4946, DOI : 10.1016/J.ASOC.2010.06.010.
- [Naw+22] Marriam NAWAZ et al., « Skin cancer detection from dermoscopic images using deep learning and fuzzy k-means clustering », in : *Microscopy Research and Technique* 85 (1 jan. 2022), p. 339-351, ISSN : 1097-0029, DOI : 10.1002/JEMT.23908, URL : <https://onlinelibrary.wiley.com/doi/full/10.1002/jemt.23908>, <https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/jemt.23908>, <https://onlinelibrary.wiley.com/doi/10.1002/jemt.23908>.

- 
- [Ned+07] Sergiu NEDEVSCHI et al., « Online extrinsic parameters calibration for stereovision systems used in far-range detection vehicle applications », in : *IEEE Transactions on Intelligent Transportation Systems* 8 (4 déc. 2007), p. 651-660, ISSN : 15249050, DOI : 10.1109/TITS.2007.908576.
- [NNK19] Zahra NAZARI, Masooma NAZARI et Dongshik KANG, « A bottom-up hierarchical clustering algorithm with intersection points », in : *International Journal of Innovative Computing, Information and Control* 15 (1 fév. 2019), p. 291-304, ISSN : 13494198, DOI : 10.24507/IJICIC.15.01.291.
- [OCS11] Éldman de O. NUNES, Aura CONCI et Ángel SÁNCHEZ, « Robust background subtraction on traffic videos », in : *2011 18th International Conference on Systems, Signals and Image Processing*, juin 2011, p. 1-4.
- [OFM07] Reza OLFATI-SABER, J. Alex FAX et Richard M. MURRAY, « Consensus and cooperation in networked multi-agent systems », in : *Proceedings of the IEEE* 95 (1 jan. 2007), p. 215-233, ISSN : 00189219, DOI : 10.1109/JPROC.2006.887293.
- [Oza18] Sukru OZAN, « Increasing system performance in machine learning by using multiprocessing », in : *26th IEEE Signal Processing and Communications Applications Conference, SIU 2018* (juill. 2018), p. 1-4, DOI : 10.1109/SIU.2018.8404280.
- [PK21] Farid PAZHOHI et Alan KINGSTONE, « The Effect of Movie Frame Rate on Viewer Preference : An Eye Tracking Study », in : *Augmented Human Research 2021* 6 :1 6 (1 jan. 2021), p. 1-5, ISSN : 2365-4325, DOI : 10.1007/s41133-020-00040-0, URL : <https://link.springer.com/article/10.1007/s41133-020-00040-0>.
- [Pre00] Lutz PRECHELT, « Empirical comparison of seven programming languages », in : *Computer* 33 (10 oct. 2000), p. 23-29, ISSN : 00189162, DOI : 10.1109/2.876288.
- [Pyt22a] PYTHON, *multiprocessing — Process-based parallelism — Python 3.10.5 documentation*, <https://docs.python.org/3/library/multiprocessing.html>, 2022.

- 
- [Pyt22b] PYTHON, *multiprocessing — Process-based parallelism — Python 3.10.5 documentation - Managers*, <https://docs.python.org/3/library/multiprocessing.html>, 2022.
- [Pyt22c] PYTHON, *subprocess — Gestion de sous-processus — Documentation Python 3.10.5 - popen-objects*, <https://docs.python.org/fr/3/library/subprocess.html>, 2022.
- [PZS20] Haojie PENG, Jie ZHANG et Weisheng SHI, « An Adaptive Cubature Kalman Filter Algorithm for Passive Location », in : *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2020* (déc. 2020), p. 420-423, DOI : 10.1109/ICCWAMTIP51612.2020.9317500.
- [Qiu14] Shi QIU, *BBox-Label-Tool*, <https://github.com/puzzledqs/BBox-Label-Tool>, 2014.
- [QY20] Gui QUANAN et Xia YUNJIAN, « Kalman Filter Algorithm for Sports Video Moving Target Tracking », in : *Proceedings - 2020 International Conference on Advance in Ambient Computing and Intelligence, ICAACI 2020* (sept. 2020), p. 26-30, DOI : 10.1109/ICAACI50733.2020.00010.
- [Rei+21] Matthias A. REINHARD et al., « Cognitive behavioral analysis system of psychotherapy reduces loneliness in patients with persistent depressive disorder », in : *Journal of Affective Disorders Reports* 5 (juill. 2021), p. 100171, DOI : 10.1016/J.JADR.2021.100171.
- [RF18] Joseph REDMON et Ali FARHADI, « YOLOv3: An Incremental Improvement », in : (avr. 2018), DOI : 10.48550/arxiv.1804.02767, URL : <https://arxiv.org/abs/1804.02767v1>.
- [RK22] Rohit RANJAN et Shashi Shekhar KUMAR, « User behaviour analysis using data analytics and machine learning to predict malicious user versus legitimate user », in : *High-Confidence Computing* 2 (1 mars 2022), p. 100034, DOI : 10.1016/J.HCC.2021.100034.
- [RML17] Krutika ROHIT, Kinjal MISTREE et Jenish LAVJI, « A review on abnormal crowd behavior detection », in : IEEE, 2017, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/8275999/>.



- 
- [RWM11] Scott D. ROTHENBERGER, John E. WENSKOVITCH et G. Elisabeta MARAI, « Pexel and heatmap visual analysis of multidimensional gun/homicide data », in : *VAST 2011 - IEEE Conference on Visual Analytics Science and Technology 2011, Proceedings* (2011), p. 297-298, DOI : 10.1109/VAST.2011.6102482.
- [SAP20] Pertiwang SISMANANDA, Maman ABDUROHMAN et Aji Gautama PUTRADA, « Performance Comparison of Yolo-Lite and YoloV3 Using Raspberry Pi and MotionEyeOS », in : *2020 8th International Conference on Information and Communication Technology, ICoICT 2020* (juin 2020), DOI : 10.1109/ICoICT49345.2020.9166199.
- [SAT11] Yahia SAID, Mohamed ATRI et Rached TOURKI, « Human detection based on integral Histograms of Oriented Gradients and SVM », in : IEEE, 2011, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/6031422/>.
- [SF09] Carlos N. SILLA et Alex A. FREITAS, « Novel top-down approaches for hierarchical classification and their application to automatic music genre classification », in : *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics* (2009), p. 3499-3504, ISSN : 1062922X, DOI : 10.1109/ICSMC.2009.5346776.
- [Sha+18] Ali SHATNAWI et al., « A comparative study of open source deep learning frameworks », in : *2018 9th International Conference on Information and Communication Systems, ICICS 2018* 2018-January (mai 2018), p. 72-77, DOI : 10.1109/IACS.2018.8355444.
- [SK07] Igor SKRJANC et Gregor KLANCAR, « Cooperative collision avoidance between multiple robots based on Bézier curves », in : *Proceedings of the International Conference on Information Technology Interfaces, ITI* (2007), p. 451-456, ISSN : 13301012, DOI : 10.1109/ITI.2007.4283813.
- [SLL11] Bao Quan SHI, Jin LIANG et Qing LIU, « Adaptive simplification of point cloud using k-means clustering », in : *Computer-Aided Design* 43 (8 août 2011), p. 910-922, ISSN : 0010-4485, DOI : 10.1016/J.CAD.2011.04.001.

- 
- [Sme+19] Kirill SMELYAKOV et al., « Local feature detectors performance analysis on digital image », in : *2019 IEEE International Scientific-Practical Conference : Problems of Infocommunications Science and Technology, PIC S and T 2019 - Proceedings* (oct. 2019), p. 644-648, DOI : 10 . 1109 / PICST47496 . 2019 . 9061331.
- [SNB18] M. S. SELEZNEVA, K. A. NEUSYPIN et A. V. BABICHENKO, « Modification of non-linear kalman filter in correction scheme of navigational systems of carrier-based aircraft », in : *2018 International Russian Automation Conference, RusAutoCon 2018* (oct. 2018), DOI : 10 . 1109 / RUSAUTOCON . 2018 . 8501670.
- [Sun+14] Yiou SUN et al., « A modified marginalized Kalman filter for maneuvering target tracking », in : *IEEE, 2014*, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/7105582/>.
- [TS18] Paulius TUMAS et Arturas SERACKIS, « Automated image annotation based on YOLOv3 », in : *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering, AIEEE 2018 - Proceedings* (déc. 2018), DOI : 10 . 1109 / AIEEE . 2018 . 8592167.
- [VJ04] Paul VIOLA et Michael J. JONES, « Robust Real-Time Face Detection », in : *International Journal of Computer Vision 2004 57 :2 57* (2 mai 2004), p. 137-154, ISSN : 1573-1405, DOI : 10 . 1023 / B : VISI . 0000013087 . 49260 . FB, URL : <https://link.springer.com/article/10.1023/B:VISI.0000013087.49260.fb>.
- [VMS06] P. A. VIJAYA, M. Narasimha MURTY et D. K. SUBRAMANIAN, « Efficient bottom-up hybrid hierarchical clustering techniques for protein sequence classification », in : *Pattern Recognition 39* (12 déc. 2006), p. 2344-2355, ISSN : 0031-3203, DOI : 10 . 1016 / J . PATCOG . 2005 . 12 . 001.
- [War+19] Arif WARSİ et al., « Gun Detection System Using Yolov3 », in : *2019 IEEE 6th International Conference on Smart Instrumentation, Measurement and Application, ICSIMA 2019* (août 2019), DOI : 10 . 1109 / ICSIMA47653 . 2019 . 9057329.

- 
- [WBL22] Chien-Yao WANG, Alexey BOCHKOVSKIY et Hong-Yuan Mark LIAO, « YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors », in : (juill. 2022), DOI : 10.48550/arxiv.2207.02696, URL : <https://arxiv.org/abs/2207.02696v1>.
- [WM10] Shu WANG et Zhenjiang MIAO, « Anomaly detection in crowd scene », in : IEEE, 2010, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/5655356/>.
- [Wol+91] William J. WOLFE et al., « The Perspective view of Three Points », in : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (1 1991), p. 66-73, ISSN : 01628828, DOI : 10.1109/34.67632.
- [Xie+09] Wenhan XIE et al., « Accurate pose and location estimation of uncalibrated camera in urban area », in : *International Geoscience and Remote Sensing Symposium (IGARSS)* 4 (2009), DOI : 10.1109/IGARSS.2009.5417399.
- [Yan+16] Xiao YANG et al., « Path following for an autonomous underwater vehicle using GP-LOS », in : *OCEANS 2016 - Shanghai* (juin 2016), DOI : 10.1109/OCEANSAP.2016.7485596.
- [Yao+20] Rui YAO et al., « Video Object Segmentation and Tracking », in : *ACM Transactions on Intelligent Systems and Technology (TIST)* 11 (4 mai 2020), p. 36, ISSN : 21576912, DOI : 10.1145/3391743, URL : <https://dl.acm.org/doi/abs/10.1145/3391743>.
- [Yua+06] Jing YUAN et al., « Path following control for tractor-trailer mobile robot based on virtual desired path », in : *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)* 2 (2006), p. 8871-8875, DOI : 10.1109/WCICA.2006.1713715.
- [ZDL17] Liping ZHANG, Song DENG et Shiyue LI, « Analysis of power consumer behavior based on the complementation of K-means and DBSCAN », in : IEEE, 2017, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/8245490/>.
- [Zha+16] Zhenxin ZHANG et al., « A Multilevel Point-Cluster-Based Discriminative Feature for ALS Point Cloud Classification », in : *IEEE Transactions*

---

on *Geoscience and Remote Sensing* 54 (6 juin 2016), p. 3309-3321, ISSN : 01962892, DOI : 10.1109/TGRS.2016.2514508.

- [Zha+18] Ke ZHANG et al., « Point Cloud Attribute Compression via Clustering and Intra Prediction », in : *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB 2018-June* (août 2018), ISSN : 21555052, DOI : 10.1109/BMSB.2018.8436850.
- [Zha+22] En ZHANG et al., « Practical multi-party private collaborative k-means clustering », in : *Neurocomputing* 467 (jan. 2022), p. 256-265, ISSN : 0925-2312, DOI : 10.1016/J.NEUCOM.2021.09.050.
- [Zho+20] Zijun ZHOU et al., « Pedestrian Navigation Method Based on Machine Learning and Gait Feature Assistance », in : *Sensors 2020, Vol. 20, Page 1530* 20 (5 mars 2020), p. 1530, ISSN : 1424-8220, DOI : 10.3390/S20051530, URL : <https://www.mdpi.com/1424-8220/20/5/1530/htm%20https://www.mdpi.com/1424-8220/20/5/1530>.
- [ZL10] Lijing ZHANG et Yingli LIANG, « Motion Human Detection Based on Background Subtraction », in : t. 1, IEEE, 2010, URL : <https://ieeexplore-ieee-org.scd-proxy.univ-brest.fr/document/5459067/>.





---

**Titre :** Déploiement d'un système de tracking par vidéo et analyse comportementale : application à la sécurisation d'espaces urbains.

**Mot clés :** Suivi, Analyse comportementale, Filtre de Kalman, Détection vidéo, Boids, Anonymat

**Résumé :** Les villes œuvrent de sorte à pouvoir rendre la vie de ses citoyens la plus sécurisée possible. Cette volonté s'accompagne de la mise à disposition de nouveaux dispositifs tels que des caméras de vidéosurveillance en quantité. Toutefois, le trop grand nombre de ces dispositifs peut être contre-productif. Une trop grande quantité d'informations peut empêcher d'identifier l'information importante parmi la totalité des informations disponibles. L'objectif de cette thèse est d'abord de trouver des solutions permettant d'indiquer concrètement le type de situation anormale se déroulant afin de focaliser l'attention de l'opérateur sur cette situation. Cette partie se base sur

des analyses statiques extraites des flux vidéo des caméras de vidéosurveillance. Le second objectif de cette thèse a pour but d'améliorer le premier grâce à un suivi individualisé permettant de conserver l'anonymat des personnes suivies. Cette solution se base sur de la prédiction de déplacement à l'aide de filtres de Kalman. Enfin, le dernier objectif de cette thèse est directement lié au manque d'informations disponibles pour tester l'efficacité de nos différentes méthodes. Cet objectif repose donc sur la mise en place d'un système de simulation permettant de se rapprocher au mieux d'une situation réelle en utilisant des entités virtuelles autonomes.

---

**Title:** Deployment of a video tracking and behavioral analysis system: application to the security of urban spaces.

**Keywords:** Tracking, Behavioral analysis, Kalman filter, Video detection, Boids, Anonymity

**Abstract:** Cities are working to make the lives of their citizens as safe as possible. This desire is accompanied by the availability of new devices such as video surveillance cameras in quantity. However, too many of these devices can be counterproductive. Too much information can prevent the identification of important information among the totality of available information. The objective of this thesis is first to find solutions to indicate concretely the type of abnormal situation taking place in order to focus the operator's attention on this situation. This part is based on static analyses

extracted from video streams of video surveillance cameras. The second objective of this thesis is to improve the first one thanks to an individualized monitoring allowing to keep the anonymity of the monitored persons. This solution is based on the prediction of displacement using Kalman filters. Finally, the last objective of this thesis is directly related to the lack of information available to test the efficiency of our different methods. This objective is based on the implementation of a simulation system allowing to get closer to a real situation by using autonomous virtual entities.